2002

Spread spectrum-based video watermarking algorithms for copyright protection

Serdean, Cristian Vasile

Chapter

5

"I have the terrible feeling that, because I am wearing a white beard and am sitting in the back of the theatre, you expect me to tell you the truth about something. These are the cheap seats, not Mount Sinai."

George Orson Welles (1915-1985)

Watermarking in the DCT Domain

The literature largely agrees that watermarking in the transform domain offers higher capacity and increased robustness compared to the spatial domain. This chapter presents the case of watermarking in the DCT domain, together with several methods of increasing the capacity/robustness of the system. To achieve this goal, the system uses both advanced HVS models for watermark embedding and state-of-the-art FEC (Turbo codes) in order to protect the watermark. The casting of the watermark and other alternative modulation techniques are also analysed. A description of the DCT transform and its properties, together with many other references can be found in [Jain, 1989].

In order to improve the system even further, 3-D marking replaces the usual frame by frame approach (2-D marking) by taking into account the temporal dimension. This increases the "local" chip rate leading to better cross-correlation results (wider cross-correlation area) and caters for frame dropping/duplication attacks.

5.1 Watermark Embedding in the DCT Domain

The DCT based watermark embedding is presented in **Figure 5-1**. The scheme is similar to the spatial domain approach, but it has several differences due to the particularities of the DCT domain marking.

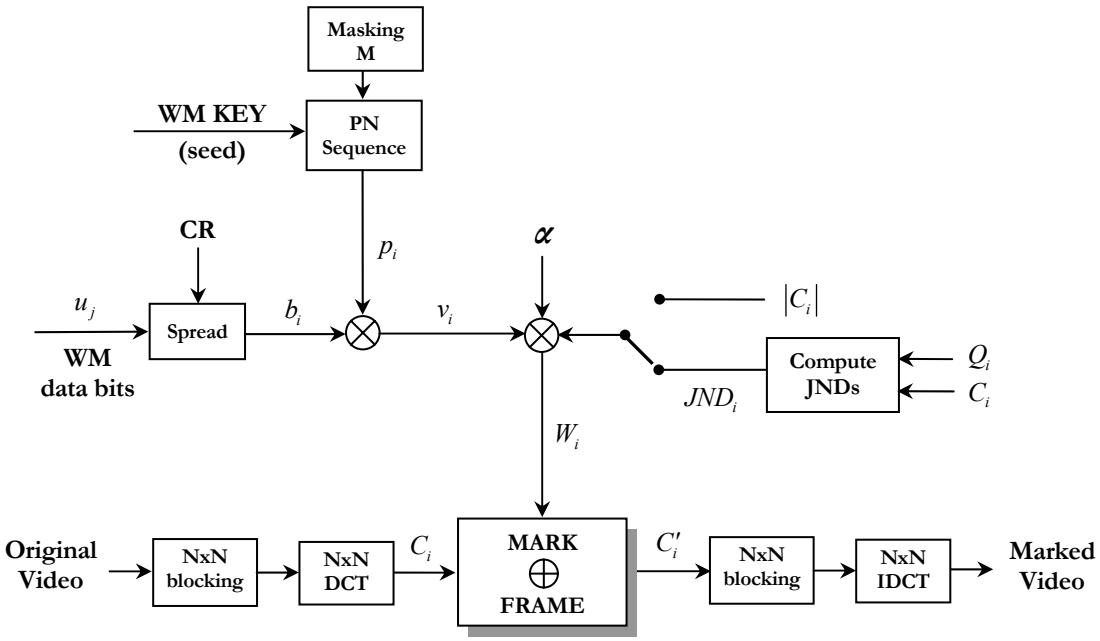


Figure 5-1 Watermark embedding in the DCT domain

The strength of marking is given for each DCT coefficient by an advanced visual model represented in **Figure 5-1** as "Compute JNDs". This block calculates the so called *Just Noticeable Difference* (JND) measure which represents the maximum value which can be added or subtracted from the given DCT coefficient, without leading to perceptual artefacts in the marked sequence. In other words, the HVS model keeps the strength of the marking just below the visibility threshold and ensures that the sequence is marked with the maximum energy and yet the invisibility requirement is still satisfied. The factor α is used as a global adjusting factor for the entire frame, either to attenuate or to amplify the value given by the HVS model for some difficult sequences.

There is also the possibility of using a "heuristic" marking as well, where the amplitude of the watermark is directly proportional to the amplitude of the DCT coefficient. The results in this case are much worse.

The watermark generation takes into account the masking matrix M, which allows one to select which coefficients within the DCT block will be marked and which skipped. The matrix M has the same dimension to the DCT block and can only have two values: zero and one. Value zero signifies that the respective coefficient is skipped (not marked) and value one that the corresponding coefficient is marked. Some authors suggests that only the medium frequency DCT coefficients should be marked since the low frequency coefficients lead to visibility artefacts and the high frequency coefficients are not robust to compression attacks. The experiments show that in fact is actually better to watermark all DCT coefficients rather

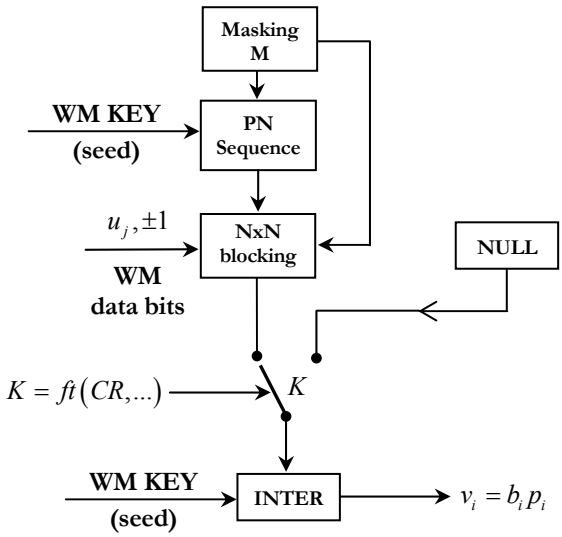


Figure 5-2 Watermark spreading detail

than excluding some of them. So, although this feature could be useful for the "heuristic" schemes, the use of advanced perceptual models (as the JND is) makes it rather inutile. In fact when is used together with the JND model, the performance of the system decreases. The HVS model inserts so much energy in the high frequency coefficients that those coefficients are quite robust to compression attacks and is a waste not to mark them. This is the reason for setting all the values within the matrix M to one.

More details about the casting of the watermark are illustrated in **Figure 5-2**. Several things can be observed. There is the possibility of watermarking only a desired number of blocks. In this case, depending of the desired percentage of marked blocks (which gives the actual chip rate) some blocks can be skipped (e.g. "marked" as null blocks).

In order to save time and memory resources, the PN sequence is generated only for the valid positions within the block (where the value of matrix M is one). Also, the PN sequence is generated only for the valid blocks (not for the null blocks). The PN sequence is generated using the same multiplicative congruential generator described in section 3.1.1.

Since the security of the algorithm is very important, the PN sequence is generated according with a secret key. The security of the algorithm is further improved by using an interleaver block INTER, also dependent on a secret key (it can be a different key or the same key used for generating the PN sequence). The use of the interleaver ensures a pseudo random distribution of the watermark data bits within the frame and within the consecutive frames as well. Each NxN DCT block corresponds to one input watermark data bit.

The watermark is embedded according to the following equation

$$C'_{n,i} = C_{n,i} + w_{n,i} = C_{n,i} + \alpha b_{n,i} p_{n,i} JND_{n,i} \tag{5.1}$$

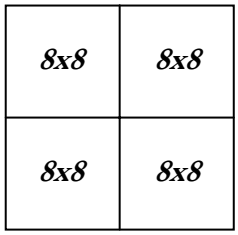


Figure 5-3 Structure of the macro-block

where $C'_{n,i}$ represents the $i'th$ coefficient from the marked block n, $C_{n,i}$ represents the original coefficient, α is an amplitude adjusting factor, $b_{n,i}$ represents the spread input data bit corresponding to block n, $p_{n,i}$ is the pseudo-random sequence corresponding to this block and finally $JND_{n,i}$ contains the HVS values associated with the block n.

5.1.1 Block Sizes and Macro-Blocks

It is well known that the cross-correlation process gives better results for larger cross-correlation areas. In the case described above, the cross-correlation area is relatively small: NxN. Typical values for N are 8, 16, 32, 64 and 128. Unfortunately N cannot be too large because of the desired resilience to attacks like line or column cuts. On the other hand the JND model works well only for small blocks, giving the best results for 8x8 blocks. To overcome at least partially this problem, one can introduce the concept of *macro-block*.

As **Figure 5-3** shows a macro-block is composed of four additional 8x8 DCT blocks coupled together. Using 8x8 blocks ensures that the HVS model works at its full potential, and by connecting four of these blocks together the effective chip rate increases four times and therefore the cross-correlator works better (the cross-correlation window is in this case 16x16). In other words, the marking is done on 8x8 blocks and the recovery of the watermark on 16x16 blocks. A macro-block corresponds to one input data bit.

5.1.2 PN Sequence Arrangement

The idea is to use the same PN value for a group G of additional DCT coefficients, in the hope that this kind of arrangement will be more robust to geometric attacks and it will improve the overall performance of the system. For the typical case described in the thesis (8x8 blocks) four additional DCT coefficients were used (in a square shape). For each of these four coefficients has been assigned the same PN sequence.

The experimental results show that by using such an arrangement, the performance of the system is slightly better (the SNR of the peaks is slightly bigger). In fact it has been observed that the variance of the peaks tends to decrease but the mean of the peaks decreases as well, although not as quickly as the variance. Generally speaking the difference between this case and normal marking is only about 1%. Although this difference is small, since this technique does not increase the complexity of the algorithm, it still constitutes a gain.

5.1.3 Alternative Modulation Techniques

The modulation used until now can be described as "amplitude" modulation since the process involves a simple addition of the watermark. On the other hand, it is well known from communication theory that differential modulation is superior to amplitude modulation. Based on this assumption one could try to implement a "differential" modulation technique for watermarking and profit from its superior noise immunity. Such an idea called "cocktail watermarking" was described in [Lu et al, 1999 and 2000] for a non-blind system. As usual, for blind systems the problem is always more difficult but the technique could be adapted to the requirements of a blind watermarking system.

Taking advantage of the macro-block structure already defined in section 5.1.1, one could adapt it for differential modulation. As **Figure 5-3** shows, there are 4 blocks available within a macro-block, and one macro-block has associated one watermark data bit.

One idea could be to divide the macro-block into two or four regions as **Figure 5-4** suggests. In the first two cases the macro-block is divided in two areas. Lets assume that the watermark data bit corresponding to this macro-block is u_j. Then the first half of the macro-block can be marked with u_j and the second half with $-u_j$ (or the other way around). At the retrieval two distinct correlations are required for each macro-block: one for the first half of the macro-block and the other one for the second half. The decision is taken by comparing the

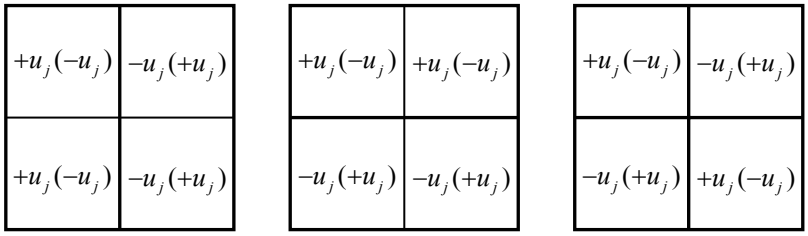


Figure 5-4 Differential modulation

signs of those two regions. The amplitude of the peak can be used as well, as a confidence measure. Assuming no errors, the signs of those two regions are always different. If the signs are not different or if the peaks are not high enough one could decide to discard the cross-correlation result entirely as unreliable.

The third case illustrated in **Figure 5-4** can be interpreted either as in the first two cases, considering in this instance a diagonal splitting of the block, or one could consider the block divided into four regions. Similar principles can be applied in this case too; the only difference is that now, four distinct correlations are required for each macro-block.

Using this kind of differential marking has however an important drawback: the effective chip rate decreases two (or respectively 4) times; in other words, the cross-correlation area is two or four times smaller, which has a negative effect on the cross-correlator.

The experiments carried out for the various arrangements described in **Figure 5-4** show that overall the scheme performs marginally worse than the normal scheme (with amplitude modulation). This is due to the smaller effective cross-correlation area. As a conclusion, taking into account that the scheme is slightly more complex and performs slightly worse, one should stick with the normal "amplitude" modulated scheme.

5.2 The Just Noticeable Difference

A very successful application for perceptual models has been proven to be image/video compression [Jayant, 1993-1 and 1993-2]. Perceptual models allow one to take advantage of the characteristics of the HVS in order to remove irrelevant and redundant information whilst keeping the compression artefacts as low as possible.

One of the most advanced HVS models, the JND model, was developed by Watson [Peterson et al, 1993], [Ahumada et al, 1992], [Watson, 1993], [Watson et al, 1994]. The aim of this model is to provide a (down to the coefficient) adaptive quantisation matrix for a JPEG based encoder. The JND algorithm is superior to many other HVS models already mentioned in section 2.4.2, due to its highly adaptive nature. This HVS model supplies a (different) JND value for each DCT coefficient.

The perceptual model used in this chapter is based on a simplified form of this HVS model. Using the idea presented in [Kim et al, 1999] this algorithm is extended to account for yet another masking effect of the HVS. The algorithm is described below.

5.2.1 Modulation Transfer Function

The model starts by computing first the *frequency sensitivity* of the eye as described by the *modulation transfer function* of the eye (MTF). The MTF describes the human eye's sensitivity to sine wave gratings at various frequencies and provides only a basic approximation of the visual model, that depends only on the viewing distance, equipment and other viewing parameters and it is independent of the image content. The result can be interpreted as a static JND threshold for each frequency band. An example could be the classical quantisation matrix from the JPEG standard.

The model developed by Watson computes this threshold using a complex formula which involves several parameters dependent on the viewing conditions. Since for watermarking the goal is only to compute the JND threshold rather than the perceptual quantisation matrix as in Watson's case, it is possible to simplify this step significantly

$$T_F(i) = \frac{Q_i}{2} \tag{5.2}$$

where Q_i is the standard quantization matrix of the JPEG standard (or any other quantization matrix developed for JPEG). This simplification affects only marginally the performance of the algorithm.

5.2.2 Luminance Masking

The next step is to compute the *luminance masking (sensitivity)* threshold. Luminance sensitivity is a way to measure the effect of the detectability threshold of noise on a constant background. This phenomenon depends on the average luminance value of the background as well as the luminance level of the noise. It basically suggests that the noise is more visible on a low intensity constant background than a high intensity contrast background. For the HVS system this is a nonlinear function. Since luminance sensitivity takes advantage of the local luminance levels from the image/video it is important that the size of the block is small enough. The luminance sensitivity is defined in [Watson, 1993] as follows

$$T_L(i,k) = T_F(i) \cdot \left[\frac{C_{0,k}}{\overline{C}_0} \right]^{0.649} \tag{5.3}$$

where $C_{0,k}$ represents the DC coefficient within block k and $\overline{C}_0$ corresponds to the mean DC coefficient over a frame.

5.2.3 Contrast Masking

A further refinement can be achieved by extending the visual model to include *contrast masking*. Contrast masking refers to the detectability of one signal in the presence of another signal (noise, artefacts). The effect is strongest when both signals are of the same spatial frequency, orientation and location [Legge et al, 1980]. More complex regions can tolerate more distortion than a smooth region or a region containing a simple sharp edge. The contrast masking is computed as in [Watson, 1993]

$$T_C(i,k) = T_L(i,k) \cdot \max\left[1\ ,\ \left(\frac{|C_{i,k}|}{T_L(i,k)}\right)\right]^w \tag{5.4}$$

where $C_{i,k}$ represents the $i'th$ DCT coefficient from block k and $w = 0$ for the DC coefficient and 0.7 elsewhere.

Equation (5.4) accounts for three important components of the human visual system: frequency, luminance and respectively contrast sensitivity. It can be seen that each new sensitivity threshold depends on the previous one.

5.2.4 Lateral Inhibition Masking

Using the same idea as in [Kim et al, 1999], the model can be extended by incorporating another masking effect, the *lateral inhibition masking*

$$T_{LI}(i,k) = \left\{ \begin{array}{l} T_C(i,k) \ \text{ if } \left(T_C(i,k) > \mu(N_{i,k}) \ \text{ or } \ (T_C(i,k) - \sigma(N_{i,k})) < \dfrac{Q_i}{32} \right) \\ T_C(i,k) - \ \sigma(N_{i,k}) \ \text{ otherwise} \end{array} \right\} \tag{5.5}$$

where $\sigma(N_{i,k})$ and $\mu(N_{i,k})$ are the standard deviation and respectively the mean for the eight neighbours of $T_C(i,k)$ and can be calculated as in **Figure 5-5**.

In the HVS, the horizontal and amacrine cells transmit signals to the neighbouring bipolar and ganglion cells, which inhibit their responses. Lateral inhibition model simulates this characteristic of HVS. In this way, the use of equation (5.5) ensures that greater marking energy will be assigned to those coefficients that are susceptible to the inhibitory effect of their neighbours. The condition from equation (5.5) was obtained from limited subjective visibility tests carried out under this project.

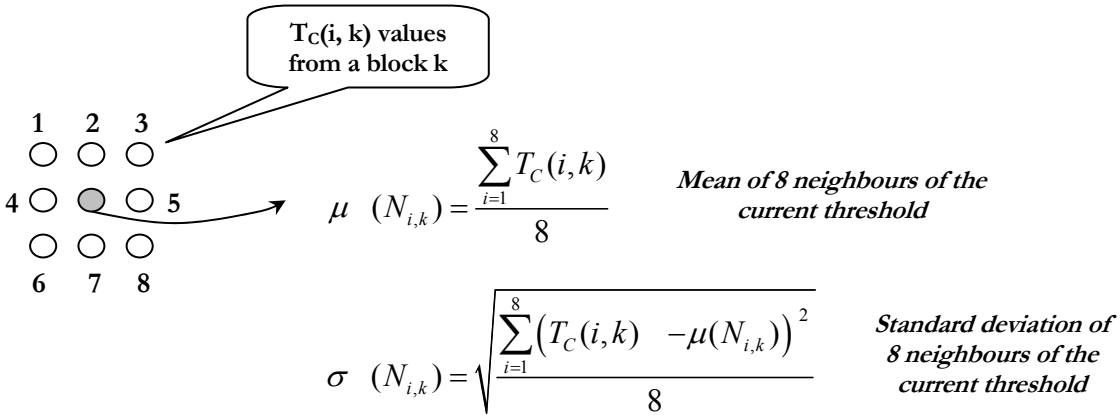


Figure 5-5 Computing the parameters of lateral inhibition masking

5.2.5 JND Threshold

Other authors [Podilchuk et al, 1997-1, 1997-2 and 1998], [Wolfgang et al, 1999], [Kim et al, 1999] use $T_C(i,k)$ or a form of $T_{LI}(i,k)$ directly as JND values, and marking is *conditional* i.e. a condition based on the original frame dictates which coefficient can be marked and which

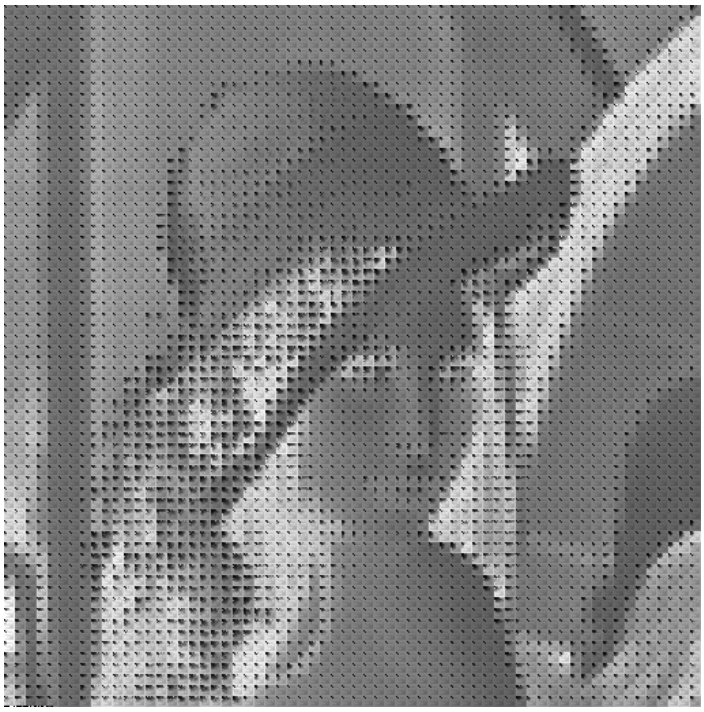


Figure 5-6 The JND map (profile) of the Lena image

cannot. In these schemes the original frame can be used to determine the marked coefficients (they are non-blind) and so watermark retrieval is possible, but this approach is not feasible for video watermarking. Moreover, neither $T_C(i,k)$ or $T_{LI}(i,k)$ are strictly speaking JND values in the sense given by the definition of the JND.

Following the basic JND definition given in Watson's paper, leads to *unconditional* marking suitable for a blind watermarking system. Moreover this approach is less empirical than the previous method and gives better results. The JND values can be computed as

$$JND_{i,k} = \frac{Q_i}{2T_{LI}(i,k)} \tag{5.6}$$

Clearly, JND values are both HVS and media dependent. In practice, the theoretical JND values supplied by equation (5.6) are within a factor 2 or 3 below of the actual perceptual threshold, and this is accounted for by the factor α in equation (5.1).

The JND map of the well known image Lena is presented in **Figure 5-6**.

5.2.6 Advantages of the JND Model

Using a good HVS model constitutes a requirement for any watermarking system; the use of such a model significantly improves the robustness and the capacity of a watermarking system. This is particularly true for highly adaptive HVS models as JND model is, and can lead to optimal embedding strength. Summarising, one can say that the JND model:

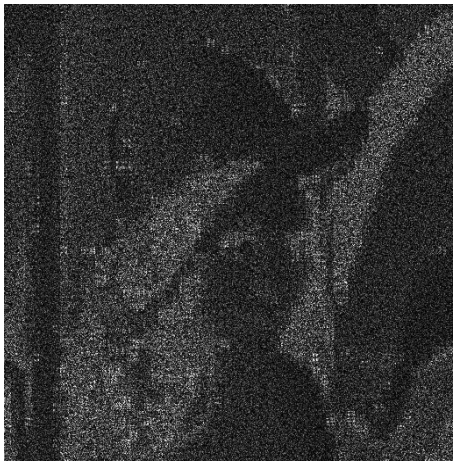
- Exploits various properties of the HVS and adaptively controls the amount of watermark energy to be embedded into each transform coefficient of the image/video; in other words the algorithm is both *HVS dependent and media dependent*.
- *Provides an upper bound* on the amount of modification one can make to the content of original image/video without incurring perceptual differences. The algorithm proves to be reasonably *accurate* for a variety of images/video sequences.
- Provides the maximal strength of the watermark which can be embedded into an image/video, leading to *maximal robustness, capacity and invisibility*.

The disadvantage of the JND model is its relative complexity, but its use can be justified by the good performance of the algorithm.

(a)

(b)

(c)



(d)

(e)

Figure 5-7 Lena image: **(a)** the original, **(b)** JND based watermarked version, **(c)** "heuristically" watermarked version, **(d)** the watermark corresponding to image (b) and **(e)** the watermark corresponding to image (c).

5.2.7 Examples of Watermarked Images

To illustrate the power of the visual model **Figure 5-7** shows the exaggeratedly marked image Lena and the corresponding watermark for both JND based marking and "heuristic" marking for similar performance results (BER). For the heuristic case, the factor α was chosen to be $\alpha = 0.06$ and for the JND marking case this factor was set to $\alpha = 6$. Both images representing the watermark were amplified 8 times in order to see the watermark properly.

One can see that the level of distortion is much higher and the artefacts much more annoying for the "heuristic" marking case. The JND marking instead leads to a noise like type of the visual artefacts which is much easily tolerated by the human visual system. By analysing the watermark itself, defined as the difference between the marked image and the original image, it can be easily seen that the "heuristic" marking leads to a much coarser watermark which does not account for the specifics of the HVS.

5.3 Watermark Recovery in the DCT Domain

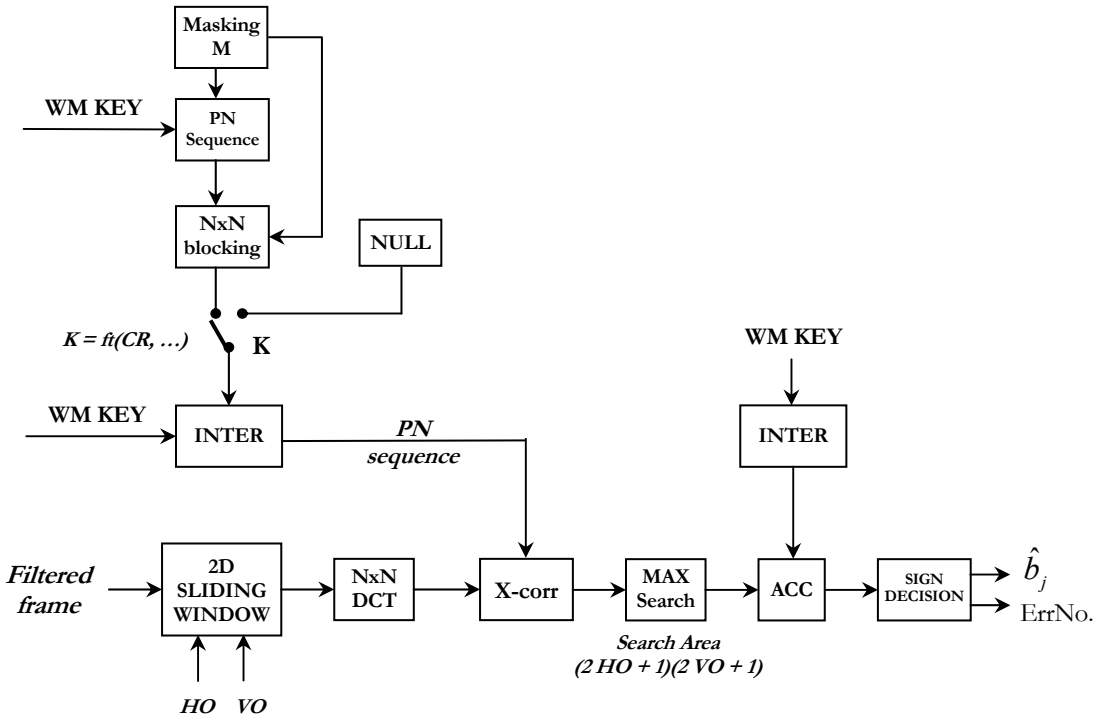


Figure 5-8 DCT watermark retrieving

The recovery of the watermark is very similar to the spatial domain technique described in section 3.2.2. The schematic of the recovery process is presented in **Figure 5-8**.

The frame is first filtered using a Laplacian filter. Then the 2-D sliding window block reads the appropriate block (macro-block) from the frame and the DCT transform of this block (macro-block) is performed obtaining the DCT coefficients.

The cross-correlation between these coefficients and the same PN sequence used for embedding (given by the watermarking key) is computed and compared to the other partial cross-correlation peaks obtained for all the other possible sliding positions.

When all these partial sliding results corresponding to one block (macro-block) are computed the maximum value is delivered to the accumulator ACC which adds this value to the corresponding previous value for that particular input bit. It can be noticed that the accumulator buffer has the same length as the number of input watermark data bits.

After all the blocks within a frame and all the frames were processed in this way, the accumulator will contain the final cross-correlation peaks for al the input data bits. Finding the (estimated) value of the input bit involves a simple sign decision (with the threshold set to zero).

The role of the second interleaver is to communicate to the accumulator the correct position of the current bit in the ACC buffer. In fact the second interleaver together with the accumulator acts as a "deinterleaver".

5.4 Temporal Dimension: 3-D Sliding Correlator

5.4.1 Temporal Macro-Blocks

The advantage of having a bigger cross-correlation area was already discussed in section 5.1. Until now, the solution was to use the macro-block as the smallest unit corresponding to (containing) one watermark data bit. Such a spatial macro-block composed by 4 additional 8x8 DCT blocks increases the "local" effective chip rate by a factor of four and therefore the cross-correlator works better.

This approach can be further extended to the temporal dimension. **Figure 5-9** illustrates this concept. By using time as the third dimension, one could extend the concept of macro-block to account for this dimension as well.

Therefore a temporal macro-block is composed from a "marking depth" number of spatial macro-blocks as defined in **Figure 5-9**. The "marking depth" can be chosen as a compromise between the size of the cross-correlation window and resilience to time synchronisation attacks. Experiments show that a marking depth of four frames is a good compromise. In this case, when 2-D sliding is performed, the effective block size is 16 times bigger compared to the case of a single block, or 4 times bigger compared to the case of a spatial macro-block which translates to increased performance for the cross-correlator.

5.4.2 Temporal Sliding Correlator

One disadvantage of the scheme described until now is the lack of robustness to time synchronisation attacks like frame dropping or frame duplication. This flaw of the existing scheme can be addressed by using a 3-D cross-correlator rather than the 2-D correlator described before. In this case the search is carried out in 3 dimensions: both in space (2-D) and in time.

The temporal sliding is illustrated in **Figure 5-10**. In this case becomes possible to perform the temporal sliding by moving all the blocks within the frame which correspond to the same watermark data bit in the same time. In this way the effective size of the cross-correlation window is much bigger compared to the case of spatial (2-D) sliding, where this technique cannot be successfully applied because of the "discrete" geometric attacks like line and column cuts. Because of this technique the frame dropping attack can be recovered with minimal loss and therefore the results of temporal sliding are much better compared with those obtained for spatial sliding.

One disadvantage of the technique is complexity. The problem was hard enough for

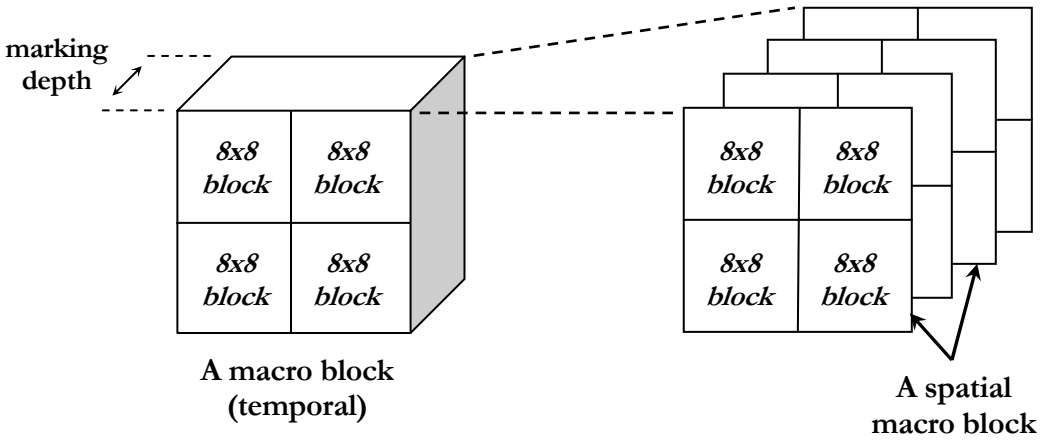


Figure 5-9 Structure of temporal macro-block

the 2-D case, as equation (3.11) shows, but now is even more complex, once the third dimension is added to the equation

$$NC = (2 \cdot ho + 1)(2 \cdot vo + 1)(2 \cdot to + 1) \tag{5.7}$$

where NC represents the number of cross-correlations and ho, vo and to are the horizontal, vertical and respectively the temporal offsets. If for a 2x2 spatial sliding, the 2-D cross-correlator has to perform $NC = 25$ cross-correlations for each block, in this case, assuming a 2x2x2 sliding, the 3-D correlator has to perform $NC = 125$ cross-correlations.

5.5 Performance of the DCT Scheme

The performance of the 2-D DCT scheme with and without sliding is illustrated in **Figure 5-11**. **Figure 5-11(a)** shows the performance of the system for different test sequences,

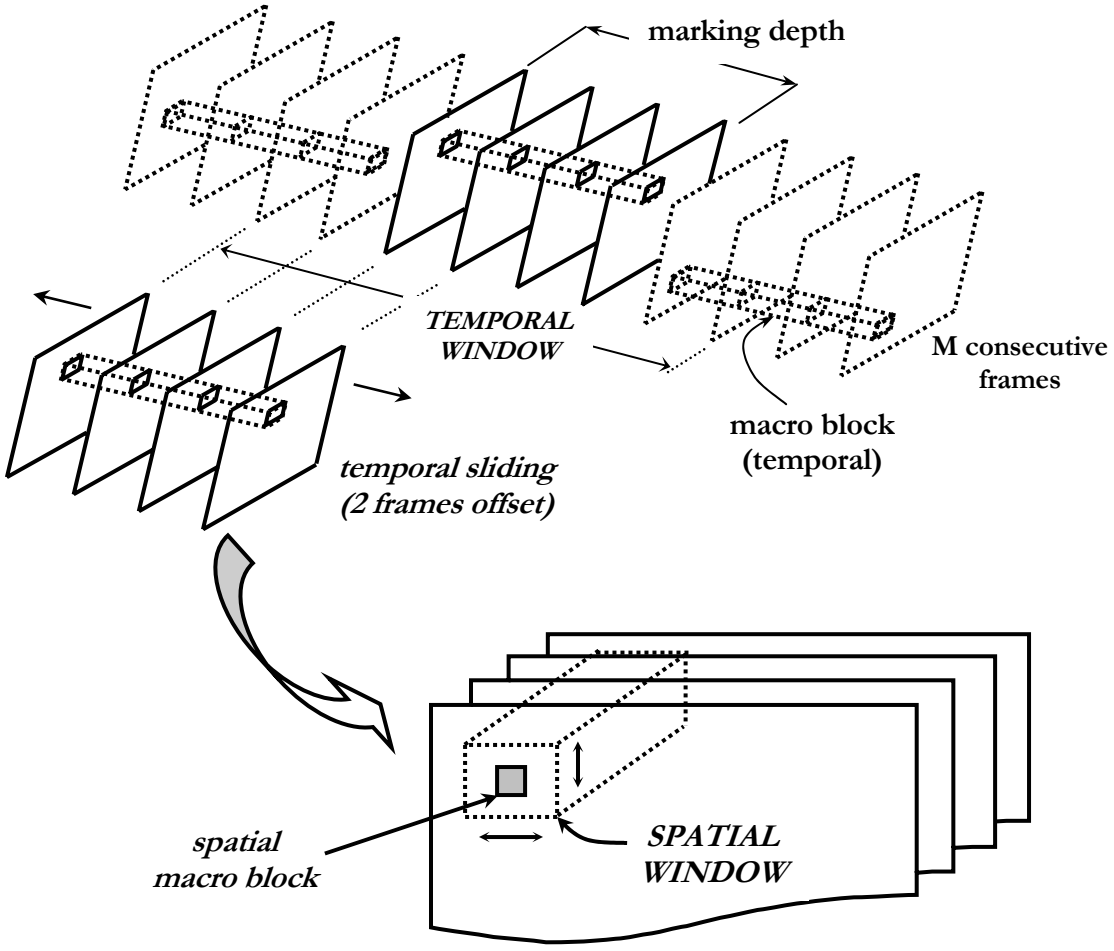
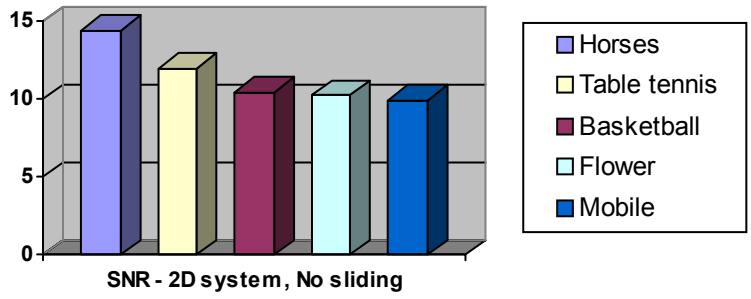


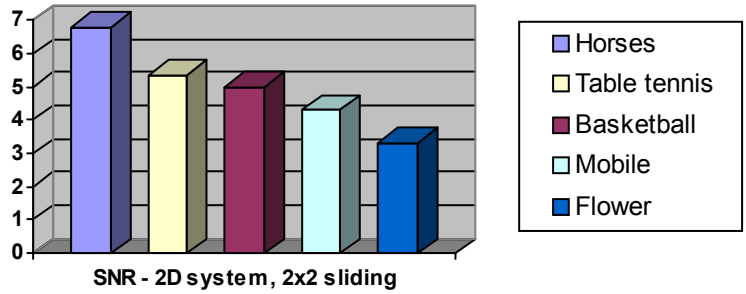
Figure 5-10 Temporal sliding

without sliding. The results for the same sequences but with 2x2 sliding are presented in **Figure 5-11(b)**. The difference between these two cases is shown in **Figure 5-11(c)**.

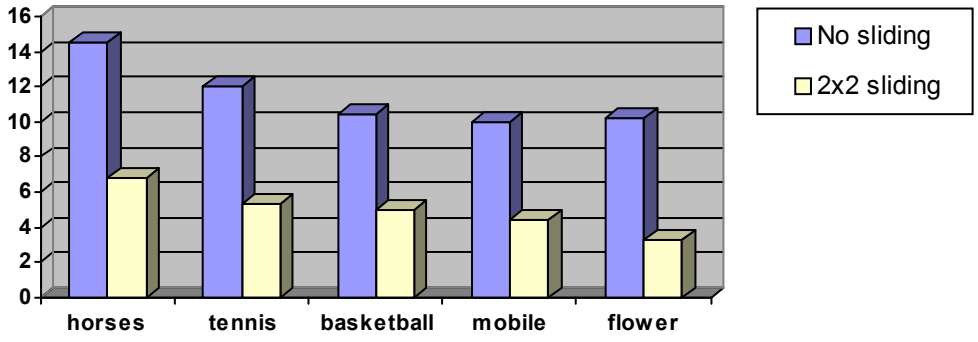
These results were obtained for watermarking 24 video frames with 1024 watermark data bits, for a factor $\alpha = 2$ and without any DCT coefficient masking. The chip rate in this case was 9720, corresponding to 100% block marking percentage (all the blocks were marked). The same parameters were used for **Figure 5-12**. For the 3-D scheme the marking depth was set to 4. **Figure 5-12(a)** shows the effect of block size on the performance of the 2-D sliding correlator for 2x2 sliding. As expected, the 16x16 macro-block gives better results.



(a)



(b)



(c)

Figure 5-11 Performance of the 2-D DCT watermarking scheme for several video sequences: **(a)** without sliding, **(b)** with 2x2 sliding and **(c)** comparison between these two cases.

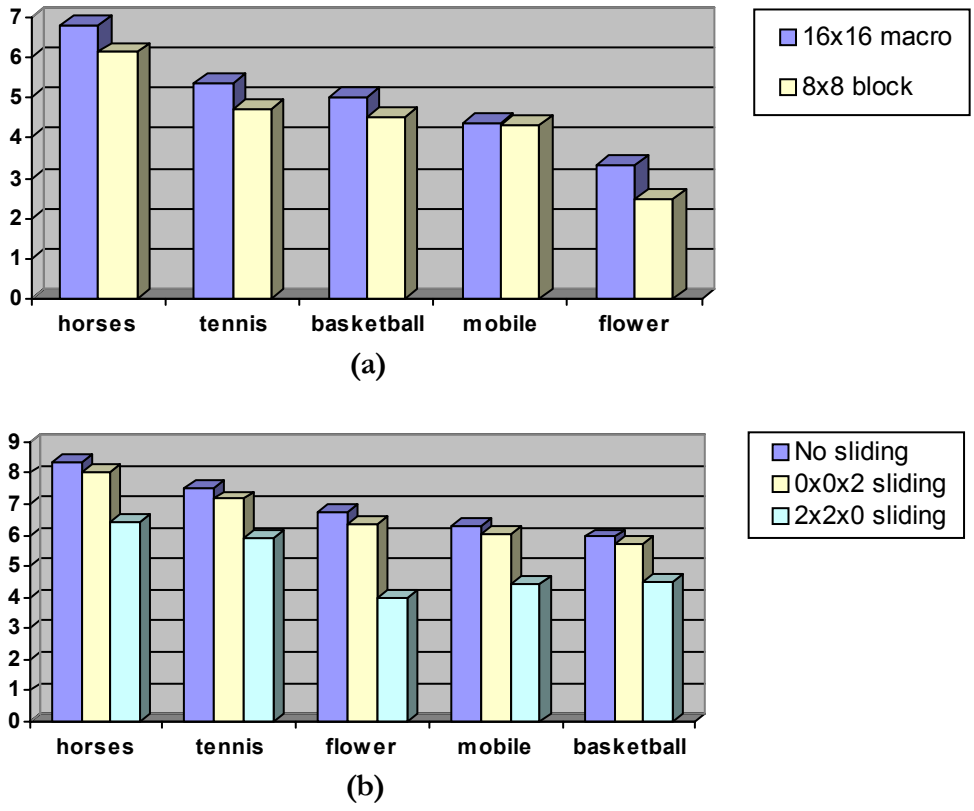


Figure 5-12 DCT domain watermarking: **(a)** the effect of block dimension on the 2-D sliding correlator, for 2x2 sliding and **(b)** the effect of spatial and temporal sliding on the performance of the 3-D system.

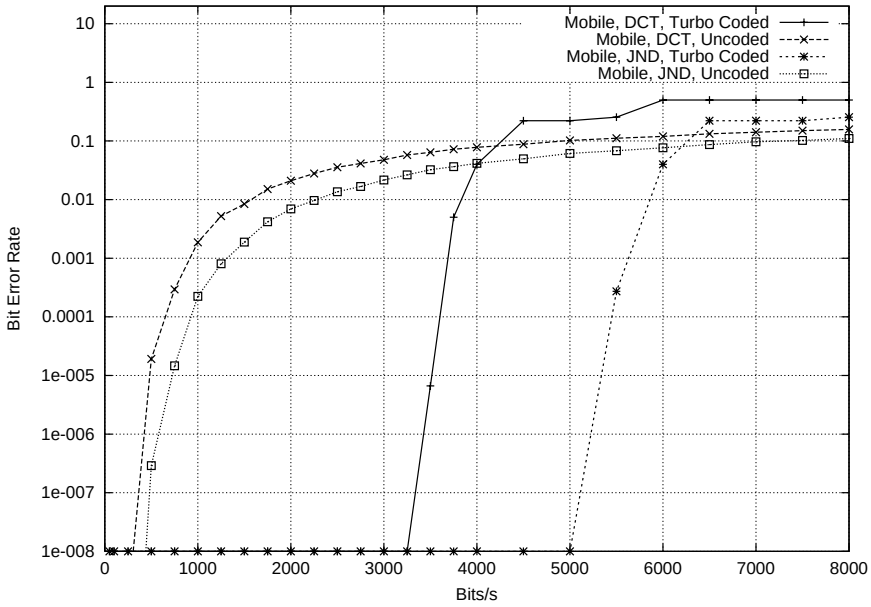


Figure 5-13 The capacity of the JND-based system compared with the "heuristic" marking, under 6Mbps MPEG2 attack.

For the same parameters of the scheme as in the 2-D case, **Figure 5-12(b)** compares the case of temporal sliding with the spatial sliding (for the 3-D scheme). It can be seen that due to the higher effective cross-correlation area available for temporal sliding (all the blocks corresponding to one bit are moved together) the loss in this case is minimal compared to the non sliding case. This difference is obvious when temporal sliding is compared with spatial sliding (for the same 3-D correlator). The SNR is much smaller for this later case, due to the much smaller effective cross-correlation area.

The performance of the JND-based system compared to the "heuristic" marking system (where instead of the JND value one uses the magnitude of the DCT coefficients and an appropriate scaling factor α) is shown in **Figure 5-13**, for a MPEG2 attack at 6Mbps. As it can be easily remarked, the JND scheme is net superior to the heuristic marking, since allows one to embed a much higher energy (close to the maximum possible limit) into the video, while maintaining the invisibility of the watermark. This is true for both Turbo coded and uncoded cases. The gain of the Turbo coded system for the JND case is much higher (almost double) because the channel is "better" in this case (the SNR is higher, because the watermark energy is higher).

The performance of the system for multiple line cuts is illustrated in **Figure 5-14**. The figure illustrates 3 distinct cases: first, only one line (line 288) is cut and the watermark is recovered without performing any sliding at al; in the second case, two lines are cut (line 192 and line 384) and the watermark is recovered still without any sliding at al; and finally, 2x2 spatial sliding is employed in order to recover the watermark from the attacked image.

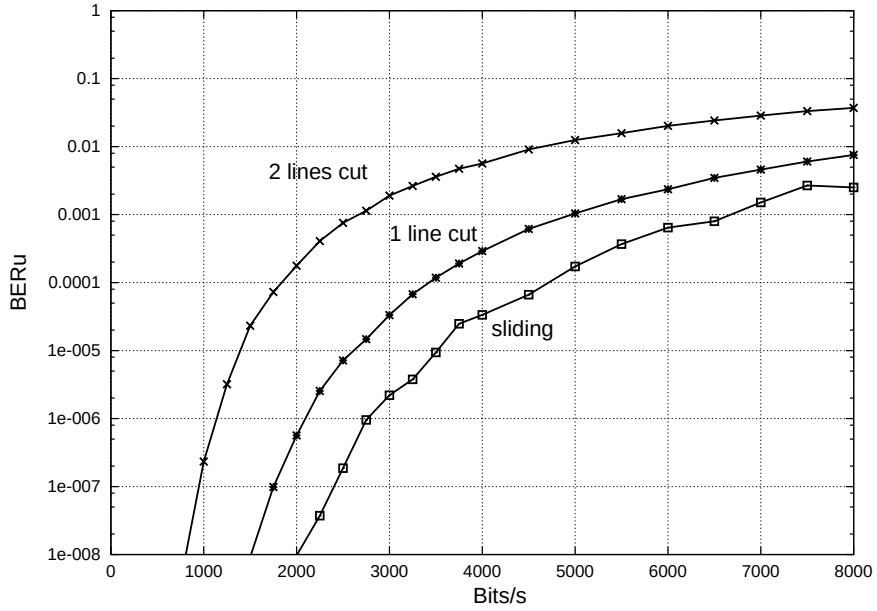


Figure 5-14 The performance of the JND-based system under multiple line cuts and the effect of sliding, for an uncoded system and typical video sequence "basketball".

Although the efficiency of the spatial sliding is not very high (when the video is not attacked the system's capacity is higher than 8000bps) even for the 3-D correlator, the capacity still improves significantly compared to the cases when no sliding is performed.

Using the same 3-D correlator, this time for frame cuts, leads to an entirely different situation. In **Figure 5-15** are illustrated again three distinct cases: first the frame number 5 (out of 25 frames) is cut and the watermark recovered without any temporal sliding; in the second case the frame number 10 is removed and again the watermark is recovered without any temporal sliding; finally temporal sliding is employed in order to recover both these attacks. One can easily see that cutting frame number 5 is a much worse attack than cutting frame number 10, because in this case the watermark is completely desynchronised starting with the frame number 6 rather than starting with the frame number 11. When temporal sliding is involved, the efficiency of the cross-correlator is very high, since moves together a much larger area compared to the case of temporal sliding. That's why the results for temporal sliding are very close to the un-attacked situation. In this case the marking depth was 2 frames.

The impact of the marking depth on the system's performance is illustrated in **Figure 5-16**, for an attack consisting in a line cut combined with 6Mbps MPEG2 compression. The diagram shows the results for a marking depth of 5 frames compared to the case of the "classical" 2-D correlator (which can be regarded as a 3-D correlator with a marking depth of one frame). The results are presented for both uncoded system and coded system, for a block length of the code N=128. As expected, the performance increases with the marking depth.

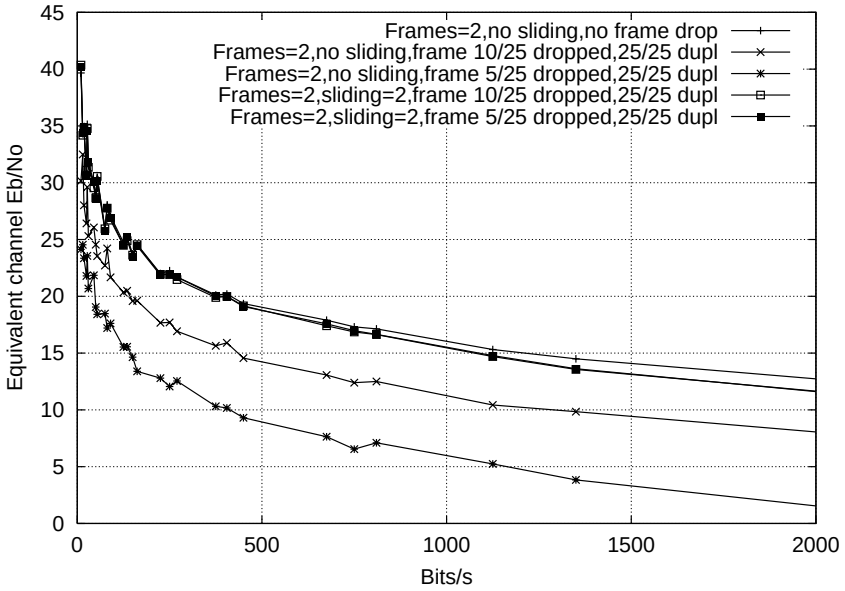


Figure 5-15 The performance of the JND-based system under frame cuts and the effect of sliding, for an uncoded system and typical video sequence "basketball".

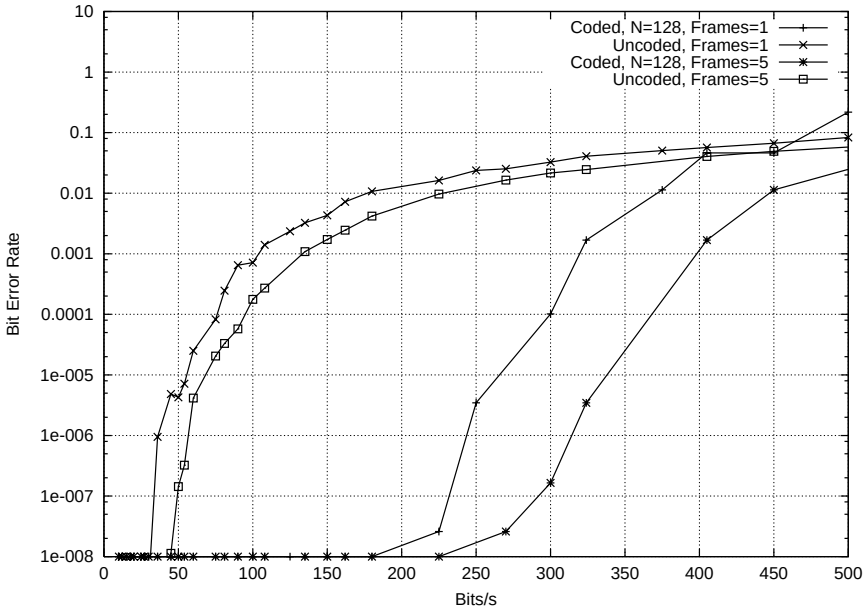


Figure 5-16 The influence of the marking depth on the system's performance under combined attack (line cut plus 6Mbps MPEG2 compression).

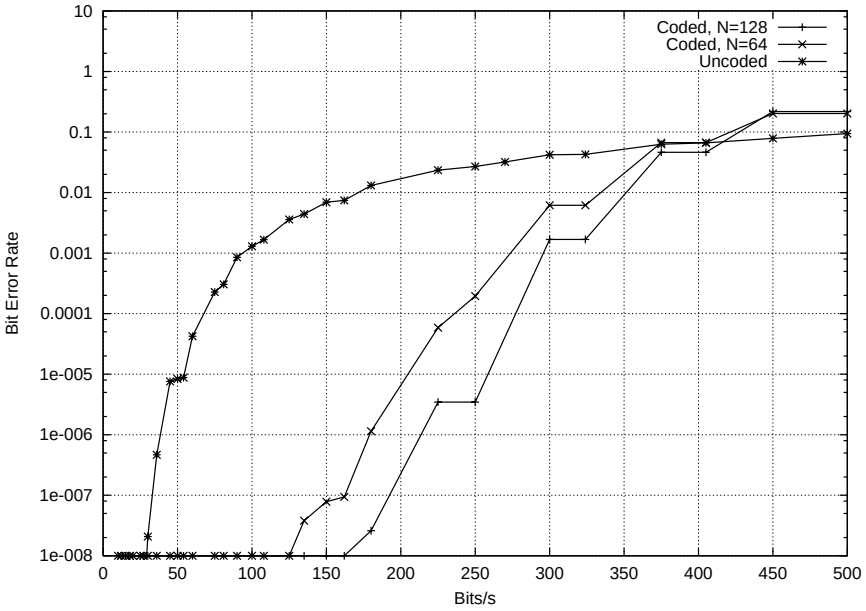


Figure 5-17 The influence of the Turbo code's block length on the system's performance under 3Mbps MPEG2 compression, for "basketball" video sequence.

The impact of the Turbo code and the importance of the block length of the code are presented in **Figure 5-17** for a 3Mbps MPEG2 compression attack. Using a Turbo code with a block length of 64 bits improves the capacity of the system about four times, while for a length of 128 bits the capacity is five times higher. This fact can be explained by the fact that the performance of the Turbo code increases with the length of the block (i.e. the interleaver length) as was shown in **Figure 4-7**.

One concern of the content providers is that several users could collude their watermarked material (each copy of the same video sequence contains a different watermark) in order to "remove" the watermark. This is done by adding a number of watermarked copies together and then taking their average as the attacked video; doing this with a sufficiently high number of copies, will "disable" the watermark. Of course the attacker doesn't have a large number of sequences, so the collusion should be ineffective for a reasonable number of copies. The results for this attack are presented in **Figure 5-18**, for the uncoded case and for a different number of copies colluded together. One can see that this attack is quite mild. Even without Turbo coding and when 5 sequences are colluded together, the capacity is still around 1000bps. Applying coding to the scheme will result in capacities larger than 8000bps.

Finally, another potentially damaging attack is the VCR attack. In this case the digital programme (video sequence) is recorded to an analogue tape using a standard VCR. The content provider wants to be able to recover the watermark even in this case.

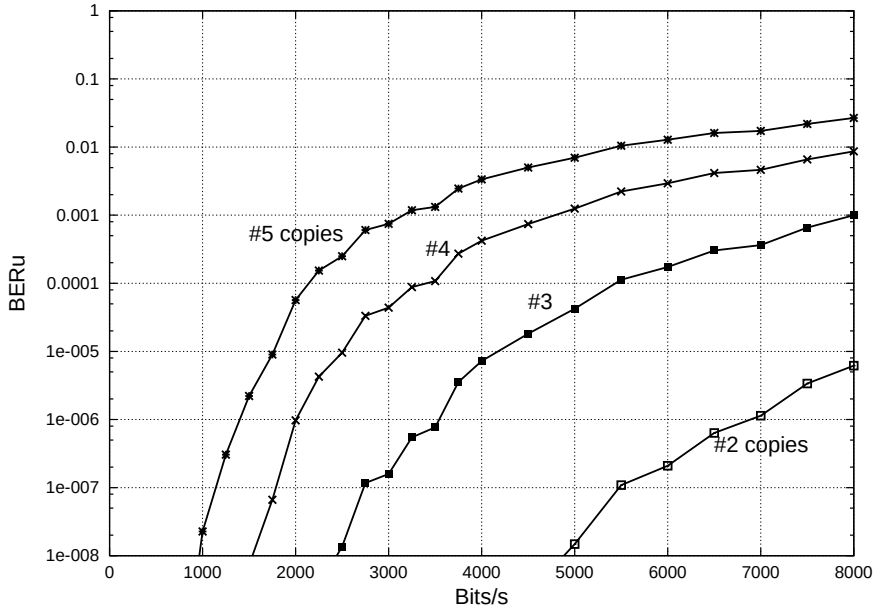


Figure 5-18 Collusion attack with a variable number of copies and its effect on system's performance.

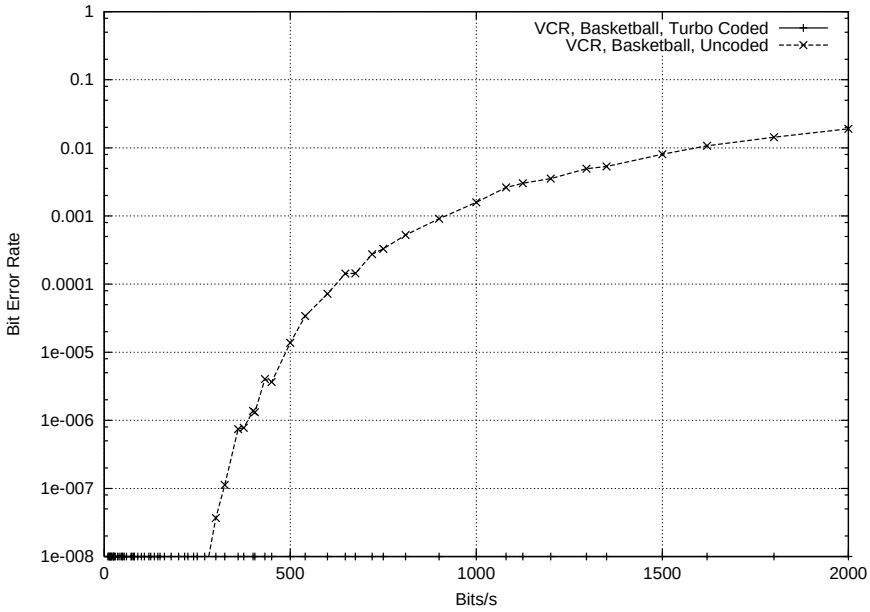


Figure 5-19 The VCR attack: the video is recorded on an analogue tape and then re-recorded in digital format using a specialised digital capture card.

Of course this can be done only after the analogue signal is transformed back to digital domain, for example by using a specialised video capture card. This attack affects the video in several ways: the signal is converted twice involving D/A ↔ A/D converters, the colour components could be slightly altered (not important in our case) and finally some jitter could be present due to the analogue recording process. In fact, this attack proves to be relatively mild as well, although is more damaging than the collusion attack. As **Figure 5-19** illustrates, even in the uncoded situation the capacity of the system is relatively large, around 350 bps. By using Turbo coding the capacity of the system exceeds 2000bps.

5.6 Conclusions

The frequency marking domain is known to give better results compared to the spatial domain watermarking techniques. The DCT in particular has also the advantage of being widely used in image processing, especially for compression. Many HVS models were developed in this context and there is relatively easy to adapt such a visual model to the requirements of watermarking.

Following this idea, this chapter presented one of the best HVS models available: the JND model and its application to the watermarking framework. This proves to be a very successful step in improving the watermarking system. With its highly adaptive nature, the JND model leads to a maximal robustness and maximal capacity watermarking system, while still preserving the invisibility of the watermark. As section 5.5 proved, the JND model almost doubles the capacity of the system.

Applying communication theory to watermarking becomes a more and more popular choice in watermarking community. By seeing the watermark channel as a communication channel, one could employ error correction codes to protect the watermark. Turbo codes are one of the best candidates for such a system, as already discussed in Chapter 4. The capacity of the watermarking system increases up to 4-5 times under 3Mbps MPEG2 compression attack, when Turbo codes are employed.

Although not as successful as the previously described methods, the system can be improved further by increasing the effective (local) cross-correlation area, using the macro-block concept and extending the system to account for the temporal dimension. Furthermore this extends the capability of the system to counteract time sync errors like frame dropping. The temporal sliding correlator proves itself to be highly efficient in combating frame dropping while achieving minimal loss compared to the non attacked case. Spatial shifts and line/column cuts can be handled without major problems. Alternative modulation techniques were also investigated leading to almost similar results (just slightly worse).

Cropping can be handled as well, giving quite good results. For example, even with an extreme attack like cropping a small 200x200 region from the video sequence, the system gives a capacity of some 1250bps (**Figure 6-9(a)**).

Attacks like collusion and VCR are relatively mild attacks and they are not posing a real threat to the system. Even without Turbo coding, the system can still achieve reasonable capacities, in fact much higher than the one required for broadcast monitoring [Cheveau et al, 2000].

Probably the most fearsome attack remains MPEG2 compression. While for a low compression at 6Mbps the system performs very well, giving a capacity of some 5000bps, this quickly drops to some 150bps for MPEG2 compression at 3Mbps while at 2Mbps the watermark cannot be retrieved at all, being completely lost.