Transparent Video Watermarking

Luiz Eduardo Sauerbronn
Departament of Mechanical Engineering, PUC-Rio
email: sauer@mec.puc-rio.br

Bruno Feijó
email: bruno@inf.puc-rio.br

Marcelo Dreux
Departament of Mechanical Engineering, PUC-Rio
email: dreux@mec.puc-rio.br

PUC-RioInf.MCC26/00 July.2000

Abstract: This paper describes the watermarking procedure to embed copyright protection into generic signals. A brief explanation concerning the Discrete Cossine Transform (DCT) is presented, showing how can it be obtained from a FFT (Fast Fourier Transform) algorithm. Finally, it applies its use into image and video sequences and presents two embeding methods. One intends to ilustrate what has been explained herein and the other presents the state of the art in video watermarking.

Keywords: Watermark, FFT, DCT, JPEG compression

Resumo: Este artigo descreve o procedimento de impressão de direitos autorais em sinais digitais genéricos. É dada uma breve explanação sobre a Transformada Cosseno, mostrando como ela pode ser obtida a partir de algoritmos de FFT (Transformada Rápida de Fourier). Ao final, são ilustradas duas formas de impressão dos direitos autorais (marca d’água digital), sendo a primeira, puramente ilustrativa, enquanto que a segunda procura descrever o estado da arte no assunto.

Palavras chave: Marca d’água, Transformada de Fourier, Transformada Cosseno, Compressão JPEG

1. Introduction

Digital data (such as music, images and video) are readily reproduced and distributed over information networks. However, these attractive properties lead to problems enforcing copyright protection. As a result, creators and distributors of digital data are hesitant to provide access to their digital intellectual property. Digital watermarking has been proposed as a means to identify the owner and distribution path of digital data. Digital watermarks address this issue by embedding owner identification directly into the digital data itself. The information is embedded by making small modifications into the data itself. When the ownership of a video (for example) is in question, the information can be extracted to completely characterize the owner or distributor of the data.

In order to be useful, a watermark must be perceptually invisible, statistically undetectable, robust to distortions applied to the host video, and able to resolve multiple ownership claims. Many watermark algorithms have been proposed. Some techniques modify spatial/temporal data samples, e.g., [Schy94],[Bend94], [Pita95], while others modify transform coefficients, e.g., [Bend94], [Cox95], [Giro96]. Similar to early work in image and audio watermarking [Swan96], [Bone96], the watermarking procedure presented here employs *perceptual models* to embed a robust and invisible watermark into video signals. Other watermarking schemes utilize the fact that digital media contain perceptually insignificant components which may be replaced or modified to embed copyright protection. However, most techniques do not *directly* exploit perceptual masking. The watermark desbribed in [Swan97a] adapts to each individual host video. In particular, the temporal, spatial and frequency distributions of the embedded watermark are dictated by the masking characteristics of the host signal. As a result, the strength of the watermark is maximized according to the host, e.g., higher amplitude in regions of the host with more textures, edges, and motion. Furthermore, to combat possible pirate attacks that attempt to exploit redundancy and motion present in the video stream, it has been created watermarks for individual objects within the video streams. The object-based watermark prevents removal based on statistical analysis and averaging of successive video frames.

This work starts by introducing a technique to calculate the DCT of a generic signal. It is, then, discussed how DCT can be used to embed copyright protection into generic signals and some different techniques to achieve this task.

2. How to calculate DCT?

What is the DCT?

DCT is a special case of FFT where the resulting signal (in the frequency domain) has only real coefficients, i.e. sin coefficients in the frequency domain are always null. In order to achieve it a smart mirroring is applied to the original signal. This mirroring obeys the following procedure:

$$
\begin{aligned}
&f(0) = f(N) \\
&for(i = 1; i < N; i++) \\
&f(2N - i) = f(i)
\end{aligned}
$$

It is important to notice that, when performing the DCT, due to the problem of mirroring, the original signal (represented by **N** spatial/temporal values) becomes a “new” signal (represented by **2N** spatial/temporal values). The FFT of this new signal is, then, performed. To reconstruct the original signal from the DCT, it is necessary to apply the inverse FFT and, then, get just the first half of the resulting signal. Why just the first half? Because the reconstructed signal will also be mirrored by the same procedure presented above. So, DCT becomes a particular case of FFT, where there are no imaginary component. Why does it happen? The best way to check the statement above may pass by the calculation of the Discrete Fourier Transform (DFT) of a one-dimensional signal, that belongs to the procedure showed above and conclude that it really happens. As an example, the calculation of the DCT for a signal f(t), defined for t=0..2 is presented below:

First, the signal is mirrored, by transforming it to the signal f(t), defined for t=0..5;

The FFT of a generic signal is given by: FFT = A + Bi

Where: A: represents the real component

B: represents the imaginary component

The imaginary components will be calculated as follows:

$$B(0) = \frac{1}{6}\left[f(0)\cdot\sin\left(\frac{0p}{6}\right) + f(1)\cdot\sin\left(\frac{0p}{6}\right) + f(2)\cdot\sin\left(\frac{0p}{6}\right) + f(3)\cdot\sin\left(\frac{0p}{6}\right) + f(4)\cdot\sin\left(\frac{0p}{6}\right) + f(5)\cdot\sin\left(\frac{0p}{6}\right)\right]$$
$$B(1) = \frac{1}{6}\left[f(0)\cdot\sin\left(\frac{0p}{6}\right) + f(1)\cdot\sin\left(\frac{2p}{6}\right) + f(2)\cdot\sin\left(\frac{4p}{6}\right) + f(3)\cdot\sin\left(\frac{6p}{6}\right) + f(4)\cdot\sin\left(\frac{8p}{6}\right) + f(5)\cdot\sin\left(\frac{10p}{6}\right)\right]$$
$$B(2) = \frac{1}{6}\left[f(0)\cdot\sin\left(\frac{0p}{6}\right) + f(1)\cdot\sin\left(\frac{4p}{6}\right) + f(2)\cdot\sin\left(\frac{8p}{6}\right) + f(3)\cdot\sin\left(\frac{12p}{6}\right) + f(4)\cdot\sin\left(\frac{16p}{6}\right) + f(5)\cdot\sin\left(\frac{20p}{6}\right)\right]$$
$$B(3) = \frac{1}{6}\left[f(0)\cdot\sin\left(\frac{0p}{6}\right) + f(1)\cdot\sin\left(\frac{6p}{6}\right) + f(2)\cdot\sin\left(\frac{12p}{6}\right) + f(3)\cdot\sin\left(\frac{18p}{6}\right) + f(4)\cdot\sin\left(\frac{24p}{6}\right) + f(5)\cdot\sin\left(\frac{30p}{6}\right)\right]$$
$$B(4) = \frac{1}{6}\left[f(0)\cdot\sin\left(\frac{0p}{6}\right) + f(1)\cdot\sin\left(\frac{8p}{6}\right) + f(2)\cdot\sin\left(\frac{16p}{6}\right) + f(3)\cdot\sin\left(\frac{24p}{6}\right) + f(4)\cdot\sin\left(\frac{32p}{6}\right) + f(5)\cdot\sin\left(\frac{40p}{6}\right)\right]$$
$$B(5) = \frac{1}{6}\left[f(0)\cdot\sin\left(\frac{0p}{6}\right) + f(1)\cdot\sin\left(\frac{10p}{6}\right) + f(2)\cdot\sin\left(\frac{20p}{6}\right) + f(3)\cdot\sin\left(\frac{30p}{6}\right) + f(4)\cdot\sin\left(\frac{40p}{6}\right) + f(5)\cdot\sin\left(\frac{50p}{6}\right)\right]$$

Evaluating the expressions, the following relations are obtained:

$$B(0) = 0$$
$$B(1) = \frac{1}{6}\left[f(1)\cdot\sin\left(\frac{2p}{6}\right) + f(2)\cdot\sin\left(\frac{4p}{6}\right) + f(4)\cdot\sin\left(\frac{8p}{6}\right) + f(5)\cdot\sin\left(\frac{10p}{6}\right)\right]$$
$$B(2) = \frac{1}{6}\left[f(1)\cdot\sin\left(\frac{4p}{6}\right) + f(2)\cdot\sin\left(\frac{8p}{6}\right) + f(4)\cdot\sin\left(\frac{16p}{6}\right) + f(5)\cdot\sin\left(\frac{20p}{6}\right)\right]$$
$$B(3) = 0$$
$$B(4) = \frac{1}{6}\left[f(1)\cdot\sin\left(\frac{8p}{6}\right) + f(2)\cdot\sin\left(\frac{16p}{6}\right) + f(4)\cdot\sin\left(\frac{32p}{6}\right) + f(5)\cdot\sin\left(\frac{40p}{6}\right)\right]$$
$$B(5) = \frac{1}{6}\left[f(1)\cdot\sin\left(\frac{10p}{6}\right) + f(2)\cdot\sin\left(\frac{20p}{6}\right) + f(4)\cdot\sin\left(\frac{40p}{6}\right) + f(5)\cdot\sin\left(\frac{50p}{6}\right)\right]$$

Note that, for all values of t, B(t) must be zero. The situation that leads to this result implies the following relations:

$$f(0) = f(3)$$
$$f(1) = f(5)$$
$$f(2) = f(4)$$

Note that this relation satisfies the mirroring algorithm given above.

The DCT described here is also known as DCT1, because there is only a single mirror from the given signal. The well known DCT4 (used in MPEG compression), uses another mirroring process that also results in no imaginary componentes. The mirroring process used on DCT4 reflects the original signal four times, instead of only once, as in the DCT1.

Thus, the conclusion is that every DCT is a special case of the DFT without imaginary components on the frequency domain representation.

As an explanation to the beginners, FFT (Fast Fourier Transform) is just a fast way of computing DFT (Discrete Fourier Transform). Both offers the same result, but FFT do it faster. As DCT (Discrete Cossine Transform) uses DFT (or FFT) into its calculation, it is suggested a thorough research into the FFT process before analyzing the DCT one.

Why use the DCT?

The watermarking process must be robust to some signal distortions, such as the JPEG compression. The JPEG compression performs the DCT1 of the given signal, discards the least significant coefficients and stores the signal information as a frequency combination of the most siginificant coefficients of the DCT1 process. Therefore, if the watermark has been embeded in the least significant coefficients, it will be simply discarded by the JPEG compression.

The Source Code Provided

The FFT algorithm from [fftw99] has been downloaded and a library has been created (which is included on the source code with the name of fftw.lib). In the source code provided in [dct99], the bi-dimensional FFT is evaluated as a multiple one-dimensional FFT. Analogously, the n-dimensional DCT source code is given as a set of multiple one-dimensional DCT. It offers the robustness of the DCT calculation, no matter the dimensions you are dealing with.

To deal with images, the Image Library IM [im99], developed by TeCGraf, has been used.

Every code has been generated and compiled using Microsoft Visual C++ 5.0.

3. A Simple Method to Embed the Watermark

After obtaining the DCT of a bi-dimensional signal (an image), alter the coefficients to achieve a new behavior that represents your watermark. Which behavior could it be? For example, in the first tests the coefficient (9,9) has been set with the same values of the coefficient (8,8). After applying the JPEG compression, the watermark is still there and, as a matter of surprise, its importance has even grown since jpeg compression has discarded all the least significant coefficients and, therefore, highlighted the importance of the most significant coefficients.

It is worth remembering that, as the DFT, the DCT also stores a lot of information on the coefficients close to the (0,0) coefficient. The coefficient (0,0) itself stores the luminance information of the entire signal. Therefore, coefficients at (8,8) and (9,9) will be probably always preserved by the JPEG compression.

Other behaviors can be set. This is just a sample.

4. A State-of-The-Art Method to Embed the Watermark

Here follows an algorithm that represents the state-of-the-art on watermarking of three-dimensional digital signal. The information given above is partly extracted from [Swan97a].

It is presented a watermarking procedure to embed copyright protection into video sequences. To address issues associated with video motion and redundancy, individual watermarks are created for objects within the video. Each watermark is created by shaping an author and video dependent pseudo-random sequence according to the perceptual masking characteristics of the video. As a result, the watermark adapts to each video to ensure invisibility and robustness. Furthermore, the noise-like watermark is statistically undetectable to prevent unauthorized removal. The watermark also resolves multiple ownership claims. It is demonstrated the robustness of the watermarking procedure to video degradation and distortions, e.g., those that result from additive noise, MPEG compression, cropping, printing and scanning.

Owner Representation, Deadlocks, and Visual Masking

Swanson reviews a few of the basic components of the proposed watermarking algorithm. In [Crav96], the authors investigate the problem of how to resolve rightful ownership of digital data when multiple ownership claims are made. It was shown that most current watermarking schemes are unable to resolve an ownership deadlock. To resolve deadlock Swanson uses a two step approach: dual watermarks and a video dependent author representation. The details of the two step approach are described in [Swan97]. With the video dependent author representation, the video owner is represented by a pseudo-random sequence. The pseudo-random sequence is created using a pseudo-random generator and *two* keys. One key is assigned to (or chosen by) the owner. The second key is computed from the video signal which the owner wishes to watermark. The pseudo-random sequence, after perceptual processing, is the actual watermark hidden into the video signal.

Swanson's watermarking procedure is based on spatial and frequency masking models, commonly used in high quality, low bit rate coders. Masking refers to a situation where a signal raises the visual threshold for other signals around it. The suggested frequency and spatial masking models [Zhu95] provide a tolerable error level (TEL) for each pixel and transform coefficient in a video frame. For each coefficient, the models predict that an error less than the TEL is invisible to human eyes. The TEL's provided by the proposed masking models are particularly useful for watermarking digital data. For example, for a pixel $p(r,c)$ located at (r,c) in a video frame, the spatial masking model provides a TEL $e(r,c)$. An error in $p(r,c)$ less than $|e(r,c)|$ is considered perceptually insignificant. As a result, a pixel $p(r,c)$ may change by plus or minus $|e(r,c)|$ to hold the watermark. Our watermarking scheme exploits this knowledge by pseudo-randomly changing every pixel in the frame $p(r,c)$, over all (r,c), by $y(r,c)e(r,c)$. The term $y(r,c)$ is the pseudo-random noise sequence which represents the author and video (see above). Thus, each pixel in the video is invisibly modified according to a pseudo-random (i.e., author and video based) pattern. The proposed watermarking algorithm modifies all of the video *transform coefficients* as well.

Object-based Video Watermarking

Video watermarking introduces issues that generally do not have a counterpart in images and audio. Video signals are highly redundant by nature, with many frames visually similar to each other. Due to large amounts of data and inherent redundancy between frames, video signals are highly susceptible to pirate attacks, including frame averaging, frame dropping, interpolation, statistical analysis, etc. Many of these attacks may be accomplished with little or no damage to the video signal. A video watermark must handle such attacks. Furthermore, a video watermark should identify any image created from one or more frames in the video. This is particularly important for watermarking news clips.

Currently, two watermarking strategies exist. An owner may apply an *identical watermark* to each frame in the video. However, such an approach is necessarily video independent, as the watermark is fixed. The other approach is to apply an *independent watermark* to each frame. However, similar regions and objects in successive video frames may be statistically compared or averaged to remove independent watermarks.

We propose an object-based watermarking procedure to address these issues. In the proposed watermarking procedure, a segmentation algorithm (e.g., [Chal96]) is employed to extract objects from the video. Each segmented object is embedded with a unique watermark according to its perceptual characteristics. In particular, each object in the video has an associated watermark. As the object experiences translations and transformations over time, the watermark remains embedded with it. An interframe transformation of the object is estimated and used to modify the watermark accordingly. If the object is highly modified or if the watermark exceeds the TEL of the object pixels, a new object and a new watermark are defined.

Objects defined in the video are collected into an *object database*. As new frames are processed, segmented objects may be compared with previously defined objects for similarity. Objects which appear visually similar use the same watermark (subject to small modifications according to affine transformations).

As a result, the watermark for each frame changes according to the perceptual characteristics while simultaneously protecting objects against statistical analysis and averaging. The proposed object-based video watermarking algorithm has several other advantages. As it is object based, the algorithm may be easily incorporated into the MPEG-4 object-based coding framework. In addition, the detection algorithm does not require information regarding the location (i.e., index) of the test frames in the video. It simply identifies the objects in the test frames. Once objects are identified, their watermarks may be retrieved from the object database and used to determine ownership rights.

In this paper, it is implemented a simplified block-based (MPEG). Rather than watermarking true objects with irregular boundaries, *blocks* are watermarked by using a modified form of MPEG motion tracking. Specifically, a frame-by-frame block tracking is performed in terms of translation, rotation, and scaling between the current reference block and candidate blocks in the next frame. Given a block in the current frame, an affine transformation vector is obtained by minimizing a cost function measuring the mismatch between the block and each predictor candidate. The range of predictor candidates are limited by scale, rotation, and translation. The error corresponding to the best matching candidate is compared to a similarity threshold. Candidate blocks with mismatches less than the threshold are signed with identical watermarks.

Embedding and Detecting the Watermark

The author representation, visual masking, and motion compensation are combined to form the proposed video watermarking algorithm shown in Figure 1.

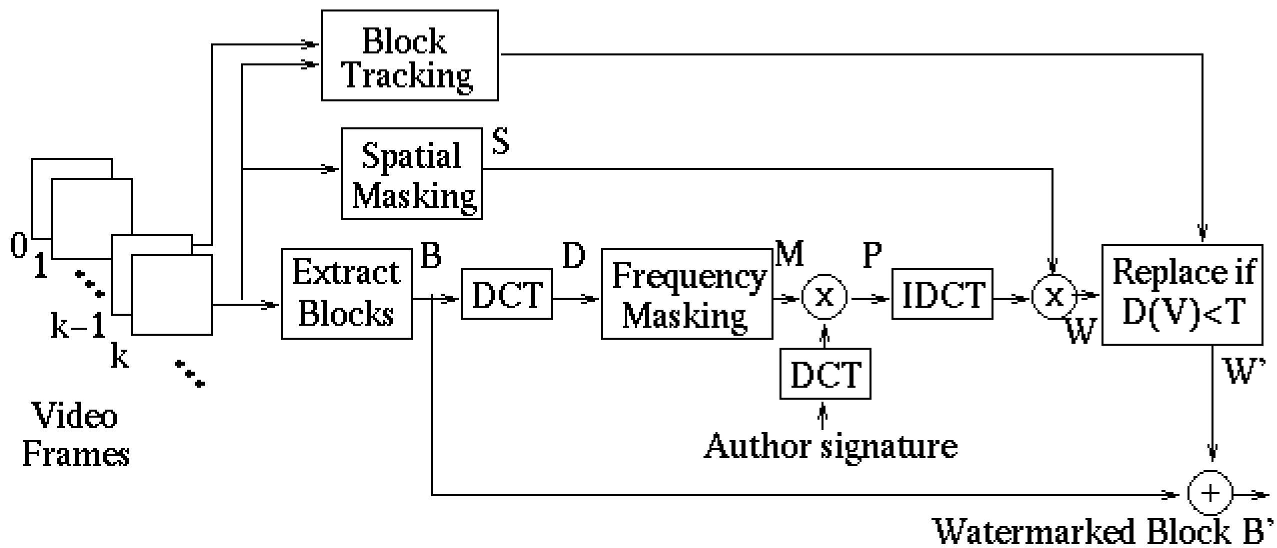

Fig. 1 - Diagram of video watermarking technique.

The watermark is computed frame-by-frame. Each of the following steps is repeated for each frame. Initially, the spatial *S* and frequency *M* masking values for the current frame are computed. The frequency masking values are obtained from the discrete cosine transform (DCT) coefficients *D* of *8x8* blocks *B* in the frame. Segmenting the frame into blocks ensures that the frequency masking estimates are localized. Each block of frequency masking values is then multiplied by part of the pseudo-random author representation. The inverse DCT of the product *P* is computed. The result is multiplied by the spatial masking values for the frame, creating the perceptually shaped pseudo-noise *W*.

Finally, motion compensation is included. The watermark for a macroblock in the current frame is replaced with the watermark for the macroblock from the previous frame (according to the offset by the motion vector) if the distortion is less than a threshold T. Thus, static regions (distortion $< T$) use a constant watermark, while motion regions use dynamic watermarks.

Detection of the watermark is accomplished via a generalized likelihood ratio test. For each frame or object R in the potentially pirated video sequence, the hypothesis test

- **H0**: $X = R - F = N$ (No watermark)
- **H1**: $X = R - F = W' + N$ (Watermark)

is performed. Here F may be the original frame closest to R or may be a composite frame of objects from the object database. In the case where R is an object, F is the corresponding object from the object database. The term W' is the potentially modified watermark, and N is noise. The correct hypothesis is obtained by measuring the similarity between the extracted signal X and original watermark W of F: $\mathrm{Sim}(X,W)=X*W/(W'*W)$, and comparing the result with a threshold. Similarity greater than a minimum threshold indicates the presence of the owner's watermark and copyright.

5. Results

For the simple method, shown above, here follow the visual results of embeding the watermark:

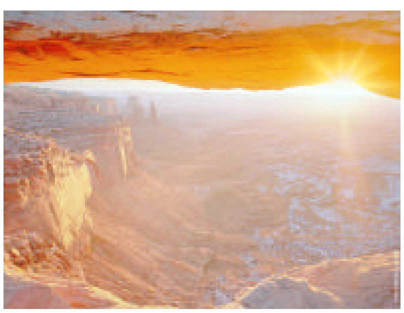

Fig.2 – Original Signal (no watermark)

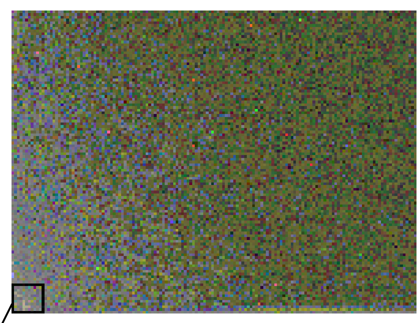

Fig.3 – Signal's DCT (inputing watermark)

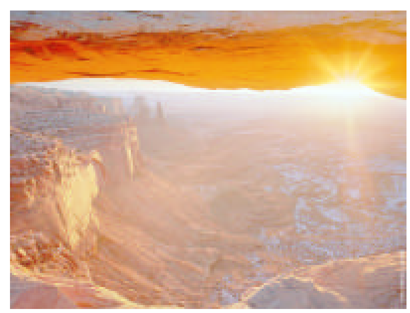

Fig.4 – Reconstructed Signal (with watermark)

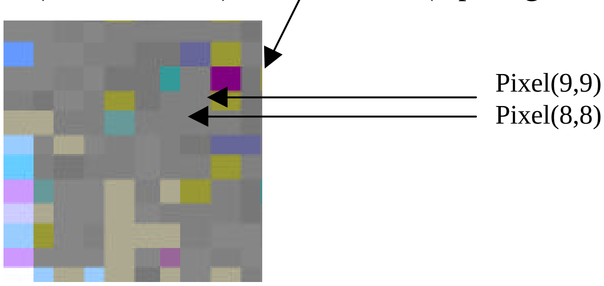

Fig.5 – Zoom of Fig. 3

For the advanced method, the visual results show that the watermark needs to be adjusted to remain transparent on the original signal:

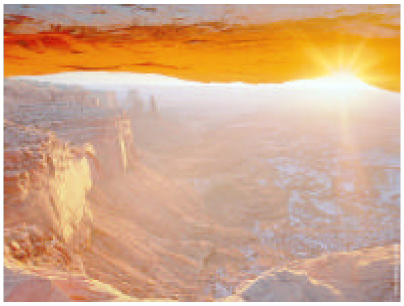

Fig.6 – Original Signal
(no watermark)

Fig.7 – Signal’s DCT
(inputing watermark)

Fig.8 – Watermark Simbol

Fig.9 – Watermark’s DCT

Fig.10 – Reconstructed Signal
(with watermark)
Note that the watermark shall be calibrated to remain transparent on the reconstructed Signal.

About the combination of the two DCTs, Swanson says that each DCT coefficient of the original signal is modified by adding a coefficient from the author signature scaled by an appropriate amount. The author signature can be considered a DCT coefficient, or a time/space coefficient in which case a DCT can be computed for DCT addition:

DCT(X')(i,j) = DCT(X)(i,j) + alpha(i,j)*AS(i,j)

Where: X is the image/object/video

X' is the modified X

alpha is a scaling factor

AS is the author signature

The same can be done for MPEG blocks, wavelet, FFT's, etc.

6. Conclusion

This paper describes the watermarking procedure to embed copyright protection into generic signals. A brief explanation concerning the Discrete Cossine Transform (DCT) is presented, by showing how it can be obtained from a FFT (Fast Fourier Transform) algorithm. At last, the use of DCT is applied into image and video sequences and two embeding methods are presented. One intends to ilustrate what has been explained herein and the other presents the state of the art in video watermarking.

7. References

[Bend94] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for Data Hiding." Tech. Rep., MIT Media Lab, 1994.

[Bone96] L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital Watermarks for Audio Signals," in *Proc. 1996 IEEE Int. Conf. on Multimedia Comp. and Systems*, (Hiroshima, Japan), pp. 473--480, 1996.

[Chal96] E. Chalom and V. M. Bove, Jr., "Segmentation of an Image Sequence Using Multi-Dimensional Image Attributes," in *Proc. 1996 IEEE Int. Conf. on Image Processing*, (Lausanne, Switzerland), pp. 525--528, 1996.

[Crav96] S. Craver, N. Memon, B.-L. Yeo, and M. Yeung, "Can Invisible Watermarks Resolve Rightful Ownerships?." IBM Research Technical Report RC 20509, *IBM CyberJournal*, July 1996.

[Cox95] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia." Tech. Rep. 95-10, NEC Research Institute, 1995.

[Giro96] F. Hartung and B. Girod, "Digital Watermarking of Raw and Compressed Video," in *Proc. of the SPIE Dig. Comp. Tech. and Systems for Video Comm.*, vol. 2952, pp. 205--213, Oct. 1996.

[Pita95] I. Pitas and T. Kaskalis, "Applying Signatures on Digital Images," in *Proc. 1995 IEEE Nonlinear Signal Processing Workshop*, (Thessaloniki, Greece), pp. 460--463, 1995.

[Swan97a]IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing June pp.23--25, 1997, Princeton, New Jersey, USA
Electronic Proceedings

[Swan96] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent Robust Image Watermarking," in *Proc. 1996 Int. Conf. on Image Proc.*, vol. III, (Lausanne, Switzerland), pp. 211--214, 1996.

[Schy94] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A Digital Watermark," in *Proc. 1994 IEEE Int. Conf. on Image Proc.*, vol. II, (Austin, TX), pp. 86--90, 1994.

[Swan97] M. D. Swanson, B. Zhu, and A. Tewfik, "Multiresolution Video Watermarking using Perceptual Models and Scene Segmentation," submitted to *IEEE Journal on Selected Areas in Comm.*, 1997.

[Zhu95] B. Zhu, A. Tewfik, and O. Gerek, "Low Bit Rate Near-Transparent Image Coding," in *Proc. of the SPIE Int. Conf. on Wavelet Apps. for Dual Use*, vol. 2491, (Orlando, FL), pp. 173--184, 1995.

On the internet:
[Dct99] Discrete Cossine Transform (DCT) Source Code:

http://www.mec.puc-rio.br/~sauer/trab/fdct_2d_com_reconstrucao/mitchell1.zip

[IM99] Image Library Reference: http://www.tecgraf.puc-rio.br/manuais/im

[fftw99] FFTW Homepage & Download Repository: http://www.fftw.org

[Scuri99] IM Developer: http://www.tecgraf.puc-rio.br/manuais/~scuri