Digital Watermarking: From Concepts to Real-Time Video Applications

Christoph Busch, Wolfgang Funk, and Stephen Wolthusen
Fraunhofer Institute for Computer Graphics

The digital watermarking field was born around the same time that the World Wide Web started its phenomenal growth. In recent years, digital watermarking has become increasingly applied to hide information in digital multimedia data, thus enlisting the watermarking technology in the difficult fight against intellectual property rights infringement.

Watermarking fundamentals

Digital watermarking evolved from the ancient technique of *steganography,* which is arguably older than the ciphers used in cryptographic systems. Steganography can be defined as hiding messages in plain sight—that is, rather than encrypting the message, it is embedded in a larger volume of data that doesn't require secret transmission; this volume of data is the message carrier. The ultimate goal is to hide messages in such a way that only the recipient, knowing what to look for, can extract them. Before the availability of computer systems, this was done manually and evolved almost into an art all its own. Popular examples include adding minor, coded variations to drawings and paintings; slightly modifying certain words or letters in handwritten material so that the recipient could pick them out to decipher the hidden message, or, as a boundary case, the classic medieval palimpsest written in invisible ink on a normal letter. Obviously, manual steganography is a rather tedious and time-consuming task at best and can be used only for small amounts of information.

Historically, transmission media was limited to canvas and paper, as our examples show. However, steganography's applicability is not limited to visual media. With the advent of digital multimedia data and transmission—which includes still-image, video, audio, and geometry data among others—the fundamental concept of steganography can be transferred from the field of analog data to the digital world using modern cryptography concepts and digital signal processing (DSP) technology.

Digital watermarks began humbly around 1993 with the exploration of simple least-significant bit (LSB) hiding schemes;[1] most of the seminal works appeared two years later, including articles by Ingmar Cox, Johannis Pitas, and Eckhard Koch and Jian Zhao.[2-4] Since that time, both the research and literature on the subject have grown beyond the boundaries of this introduction. We therefore refer you to survey articles such as the one by Ingmar Cox and Matt Miller,[5] and the digital-watermarking bibliography maintained online and in print by Petitcolas (http://www.cl.cam.ac.uk/~fapp2/steganography).

We have developed a secure, robust watermarking algorithm and applied it in digital streaming MPEG-2 format video—the format of choice in the broadcast and video stock industry.

There are many possible ways to define desirable digital-watermarking traits, depending on the application. Some requirements conflict, such as visibility versus robustness. Here we offer one definition; we later offer a brief rationale for its use as a yardstick.

Definition 1. *Digital watermarking embeds information into digital data (the carrier) in a secret and inconspicuous way. The embedded information is denoted as a digital watermark. The digital watermark shall be robust against distortions that do not significantly degrade the receiver's perception of the carrier signal.*

Suitable applications

Different kinds of digital data are available on a multitude of storage media types (or storage representations). Obviously, watermarking must be independent from the representation in which the digital media data is stored, as otherwise, removing the watermark would be as simple as converting the data from one representation to another. (We assume here that data are stored in digital form; converting digital watermarked data and storing them in analog can be considered an *attack* on the watermark.)

0272-1716/99/$10.00 © 1999 IEEE

The watermarking algorithms apply to individual signal domains (audio, video, and so on) and must be designed specifically for each domain. It's insufficient to simply adapt algorithms from one domain to accept signal data from another because the robustness criterion would not be met for distortions in the new domain. A robust watermarking algorithm, however, must adapt itself to expected attacks within the specified domain and must be tunable to allow a balancing of robustness and visibility (audibility, and so on). Safe conversions between multimedia data types are therefore possible only within one such media domain.

Generally, because covert messages have some noise robustness, they can be integrated into all multimedia data types and later retrieved from the manipulated data. But is this really watermarking? Certainly hiding and retrieving messages is not enough. Following definition 1 above, the watermark also must be *robust*. Resistance to manipulation is the discerning feature of a digital watermark.

The key to a robust watermark is to introduce it with a high degree of redundancy. This gives us a criterion to decide if a medium is well suited for watermarking. Data inherently high in redundancy can carry a more robust watermark because a highly redundant watermark can be embedded without being noticeable. Besides redundancy, the remaining prerequisite for a suitable carrier signal is that the signal must tolerate at least small, well-defined modifications without changing its semantics.

Real world still-image, video, and audio data are therefore well suited for watermarking. Problems arise with some artificially generated data (such as raytraced scenes) because, unlike samplings from real-world data, they do not contain noise unless they have been subjected to lossy compression algorithms, which mask the watermarking effects and make it possible for the human eye to detect irregularities at lower insertion levels than with "natural" sources. We discuss this type of processing in more detail below.

The most difficult media for watermarking are plain text and executable files. Although binary executables may contain some redundancy, introducing random modifications into such data is discouraged. Even when the modifications are safe, attacks are trivial since any executable file can be converted into a canonical format—and the watermark thus removed—without losing any information needed for execution.

The question with textual data itself is, "How do you introduce additional information without rendering the data useless?" There are schemes for solving this problem, such as encoding the secret message as the first letter of each word in the text,[1] but such schemes are easily broken. A newer technique for text watermarking is to introduce slight format changes to the text,[6] such as line and word shifts, to encode a message. Obviously, such a watermark will not survive format conversion. Because text does not contain any natural noise, you can extract its contents and easily remove all formatting redundancy. For example, a commercial optical-character-recognition package operating on a scanned printout or other bitmapped representation would easily eliminate all text watermarks.

Because of these difficulties, we deal here only with the more promising watermarking candidates: in particular, image and video data. With the advent of Digital Versatile Disk (DVD) as a serious distribution medium, the intellectual property of the video community is now exposed to the same dangers that have plagued digital imaging and digital audio interest groups for at least a decade.

Limitations

There are two important issues here that must be understood. First, digital watermarking for copyright protection comes into play only after offenses have been committed (disregarding for the moment the deterrent effect some proponents claim). Second, detecting secret watermarks is a computationally complex operation, limiting the practicality of spidering the Web for watermarked data.

Bearing this in mind, digital watermarks should be used as the method of last resort for intellectual property rights problems, after more conventional means have failed. Conventional mechanisms include those used for control and auditing as well as copy protection schemes. These mechanisms are valid in their own right and cannot be fully replaced by digital watermarks, even though the intrusiveness and limitation of system usability such mechanisms bring often causes potential users to reject them. In any case, these methods and their implementation issues exceed the scope of this article.

The watermarking process

The steps for implementing a watermarking system are straightforward and exploit the properties of data that include a natural noise margin. We now review issues to consider.

Basic considerations

Watermarking naturally splits into two rather different areas. The first area is the security of a watermarking system: We must decide where to embed the watermark in the data and how to restrict access to the information. The location may refer to either the spatial or frequency domain and should be determined using some randomness source. The second area is signal processing: We must decide how to modify the original data to embed the watermark.

Security

The design criteria for a secure watermarking algorithm are the same as for a cryptographic algorithm. In fact, the same criteria the National Bureau of Standards (now the National Institute of Standards and Technology) specified in 1973 for a proposed standard cryptographic algorithm[7] also apply to a watermarking algorithm.

Definition 2. *To be considered secure, the watermarking algorithm must*

- provide a high level of security,
- be completely specified and easy to understand,
- rely on a key for security rather than on the algo-

rithm's secrecy (this was already a requirement of Kerckhoffs in 1883)[8]

- be available to all users,
- adapt to diverse applications,
- be economically implementable in electronic devices,
- be efficient, and
- permit validation.

A watermark can be implemented using a secret key to restrict access to the locations where it is embedded. A secret key regulates access to the watermark.

The security of any watermarking algorithm can be increased easily by encrypting the watermark before it is embedded. Any standard cryptographic algorithm can be used for this purpose. However, the encrypting approach has an important drawback: All message bits must be retrieved correctly or the original message cannot be decrypted. While it is possible to include error-correcting code, this increases the size of the message, which should be as small as possible.

Some commercially available algorithms for image watermarking forego the secure-watermarking principles in their search for higher robustness and ease of use and distribution. Such algorithms use a single, common secret key to embed information (usually the data creator's identity compressed to a short string), which can then be read by anyone with the detection program; no key is required beyond a common one used for all embeddings and extractions. As we explain below, all other parameters being equal, the amount of data that can be embedded is inversely proportional to the watermark's robustness. This requires a mapping of more detailed information in a central database onto a compact identification string.

The danger in this case is that the entire algorithm, including the secret key, must be embedded in the code for both embedding and detecting the watermark. By disassembling the code, you can modify the algorithm's behavior. For example, removing the block against over-watermarking allows attackers to simply render the original watermark unreadable and even claim ownership themselves. Another potential attack is to embed a third person's (publicly available) author-identification string into material and thus create ownership that could be used as fraudulent evidence for criminal activities.

When watermarking schemes use a single secret key for all embeddings and extractions, there is no way to determine who watermarked data first unless you expand the scheme significantly by introducing a central trusted registration database. This can provide a time-stamp facility for thwarting attacks like those described above. However, such registrations must be accomplished over an authenticated and encrypted channel to ascertain the claimant's identity. This helps prevent injection attacks, in which a third party inserts material on behalf of a legitimate claimant through a preestablished, authenticated connection.

In essence, this scheme leaves no way to resolve an argument involving multiple claimants, reducing the evidentiary value to zero. Given the fact that schemes for breaking the program code (though not the algorithm) of such commercial packages have been published on the Internet, it should be very difficult to even use such systems for prima facie evidence.

The first requirement for a secure digital marking system therefore must be that, while the algorithm for embedding is itself publicly known, the key used to determine the actual positions or frequencies of the embedded mark (which is different from the optional encryption key) will never be disclosed to parties other than the data owner and possibly law enforcement authorities. Such a marking has evidentiary value, but by itself does nothing to inform the user of copyrighted data. It is therefore often advisable to embed a watermark with both a publicly known key and a secret watermark. This public mark provides embedded, auxiliary information—such as the contact person for licensing the material—which does not alter the data's appearance and will survive most processing steps.

The second important requirement for the secret watermark is that it must be impossible to distinguish from the original data's white noise. If some patterns remain characteristic of an embedded watermark regardless of its key (even if this occurs only after elaborate transformations), the watermark can be removed or destroyed without knowledge of the secret key.

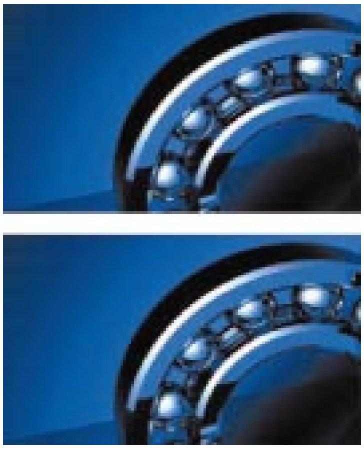
Media modification

The second building block of a watermarking algorithm is the signal-processing step, in which the original data is modified to enable embedding of the watermark. This step can be considered as adding a secondary signal into a primary one; assuming this message has been encrypted, it will appear as white noise. Although robustness increases with the strength of the added signal, the initial signal's quality degrades. In most cases, you must subjectively determine the point at which the noise degrades the original signal too much.

During normal processing, digital data are subject to various manipulations that weaken the embedded signal's strength or skew it in a way that makes reconstruction and eventual discovery more difficult (such as rotation, cropping, shearing, and color adjustments). Obviously, such processing can also be used intentionally to attack a watermark. The threshold for the marking signal's maximum strength depends on several factors. One is the subjective level of degradation the marker can accept. Another factor is a perceptual issue: People are more forgiving of noise levels when the data is noisy to begin with, such as in still photographs, which are noisy due to imperfections in the signal-acquisition mechanism (such as a charge-coupled device sensor).

If, on the other hand, data is used that does not contain "natural" noise—such as computer- generated imaging—people notice a much smaller addition of noise. In this case, you can sometimes completely reconstruct the original data by performing a spectral analysis over it and eliminating all frequencies with intensities below a certain threshold. As an alternative, you can use the noise signal distribution by averaging the data points presumed to be of equal value in the original data set and using them to interpolate the remaining data points.

For example, Figure 1a shows a computer-generated image; Figure 1b shows the same image after noise



1 **An unmarked computer generated image (a) and a watermarked computer generated image (b).**

insertion. Some artifacts appear in both Figure 1a and 1b. These result from a fairly strong JPEG compression, which is typically used for low-bandwidth transmission media. Nonetheless, the noise introduced by the watermarking process is still plainly visible. Some watermarking systems produce less noticeable artifacts. If, for example, they operate on the same block size elements as JPEG or MPEG, the distinction between compression-induced errors and those caused by the watermarking algorithm is less clear. On the other hand, the inverse holds true for other compression algorithms such as iterated function systems (fractal compression systems) or wavelet-based algorithms.

Lossy compression is often a necessary processing step for multimedia data. These mechanisms take advantage of the human perceptual system to eliminate certain data that is least significant for data perception. (This complicated issue is covered in more detail elsewhere.[9]) Spectral energy distribution is a good indicator for the significance of individual frequencies. Other effects that can be considered include auditory masking, which lets a strong signal make an entire range of weaker signals in its spectral neighborhood inaudible; and the perceptual system's limited sensitivity to different frequencies (this applies to both sound waves and colors). Watermarks are problematic here, since they themselves strive to be perceptually invisible (inaudible, and so on) and thus are likely victims of lossy compression algorithms; they must strive to remain at the borderline of visibility.[2] Most modern watermarking algorithms prove resilient against common lossy compression algorithms.

In the case of other manipulations, the main problem is recovering the watermark signal without the original (unmodified and unmarked) data. This is often a requirement, as the modified data would otherwise need to be matched manually against their likely originals. (The alternative would be a reliable metric for similar multimedia data sets, but the current state of the art does not supply this.) This manual matching is unacceptable in commercial settings, which require automated procedures.

In the case of audio and video streams, another possible modification or attack is to cut or resample the data streams. Since the watermark detection algorithm requires some form of synchronization, it's possible to circumvent it without degrading the signal quality (unless more elaborate mechanisms or manual correlations are used). This is particularly significant when it's undesirable (or even impossible) to store potentially pirated data for analysis (such as in broadcast video). Further discussion of attack schemes on digital and video watermarks is available in the literature.[10,11]

Copyright protection of video

Digital video streams in MPEG-2 format[9] form the basis of emerging television standards like the European DVB project,[12] the DVD video standard, and video stock archives for instant random access to digital video streams. Providers of digital contents and services need copyright protection mechanisms including digital watermarking to track the dissemination of their digital products and detect large-scale commercial piracy and illegal copying of their data.

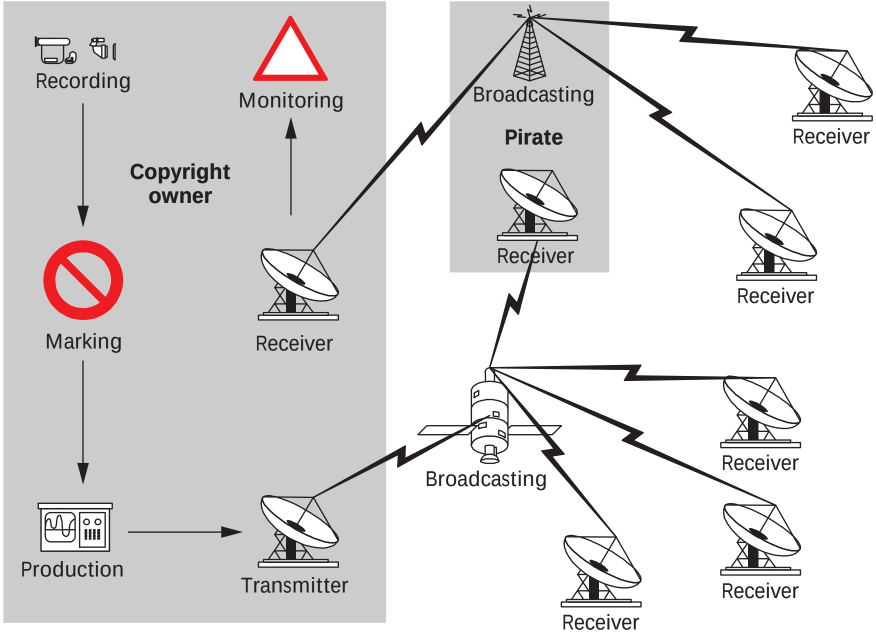
Here we present an algorithm for digital watermarking that has been integrated in a system for real-time watermarking of video streams. We designed and tuned the algorithm to meet the requirements of the production chain in TV studios and to enable the watermark to overcome the lossy MPEG-2 compression scheme. The complete system has been set up within the project Talisman (Tracing Authors' rights by Labelling Image Services and Monitoring Access Network),[13] that was funded by the European "Advanced Communications Technologies & Services" program. Talisman commenced in September 1995 and was completed by August 1998. The system embeds watermarks into video streams, monitors for labels in MPEG-2 bit streams, and integrity checks in video, all in real time. We restrict our discussion here to the core watermarking algorithm.

At the moment, special hardware is needed to achieve real-time watermark embedding for video streams of TV studio quality. Our Talisman partners, Thomson CSF (France) and CSELT (Italy), managed the extremely important hardware-related aspect of video watermarking, including hardware design and DSP-unit programming.

Application scenario

Figure 2 sketches the basic scenario for which the watermarking algorithm was designed. A video stream acquired using a digital device is subsequently processed in the TV production chain and finally broadcast as an MPEG-2 bit stream.

The digital data is marked directly after recording and before entering the production chain. The data is not necessarily recorded in a closed studio environment; it can be recorded in the field and subsequently transmitted to the broadcaster's headquarters. The raw material is cut and assembled during the production process. The final version of the video stream is encoded according to the MPEG-2 standard, and the compressed video stream is broadcast. Other parties can digitally record the data and broadcast it again instantly, or with some delay, as a digital video stream without any loss of quality. The watermark monitoring process can check for the watermark and automatically detect unauthorized use of copyrighted material.



2 Broadcasting environment for the watermarking algorithm.

Requirements

The application scenario outlined above imposes some restrictions on the watermarking algorithm:

- Watermark embedding and retrieval must be performed in real time to avoid slowing down TV production (recording, viewing, archiving, and so on).
- Watermarks must be of "transparent quality"; that is, they must be invisible in the high-quality original digital material used in TV production.
- The MPEG-2 compression decreases the digital video stream's data rate by reducing the noise level within the data. Watermarks must survive this degradation.
- Slight translations of the video frames may occur during production and transmission. Horizontal shifts mainly appear during analog-to-digital conversion, and a vertical shift of one line may occur in video mixers and digital video effects. Both problems are restricted to older installations and to new units improperly installed.

Implementation issues

We took a frame-based approach to video watermarking. During watermark embedding and retrieval, the algorithm separately processes each frame of the uncompressed video stream. Jana Dittmann and her colleagues[14] proposed a similar approach. Other techniques for video watermarking work on longer sequences and can process compressed bit streams.[15,16]

The original video data stream comes from a digital camera or a digital video player. The standard digital equipment in professional TV studios implements the International Telecommunications Union (ITU) 601 sampling standard. Digital equipment connects via the ITU-R 656[12] serial digital interface. The data is stored in digital betacam format, which provides lossless compression of the video streams, resulting in a compression ratio of 1.77:1.

The interface between the video system and the watermarking algorithm receives ITU-R 601 digital component signal as input and redirects the luminance component of each original frame to the watermarking algorithm. The watermark is embedded and the modified data returned to the interface, which inserts the modified luminance component in the ITU-R 601 output stream. When running in monitoring mode, the algorithm receives the decoded MPEG-2 stream from the interface and performs the watermark retrieval process for the luminance component of each frame of the decoded stream (see Figure 2). The interface provides the algorithm with consecutive frames of the digital video stream and directs the watermarked data to the output device. Interfacing and watermark processing must not take longer than 40 ms per frame to achieve the PAL frame rate of 25 frames per second.

The modified Koch-Zhao algorithm

At first glance, watermarking single video-stream frames seems similar to watermarking still images. This is because our approach to video watermarking modifies an algorithm designed for still pictures and applies it to each video-sequence frame. Our starting point was the algorithm proposed by Koch and Zhao.[4]

The most important modifications we introduced to the Koch-Zhao algorithm are new discrete cosine transform (DCT) and inverse DCT routines, which we optimized for real-time implementation with hardware support and additional checks of edges and textures prior to watermark embedding and retrieval to avoid artifacts in the watermarked video sequences. Here we restrict our detailed discussion to the visibility problem.

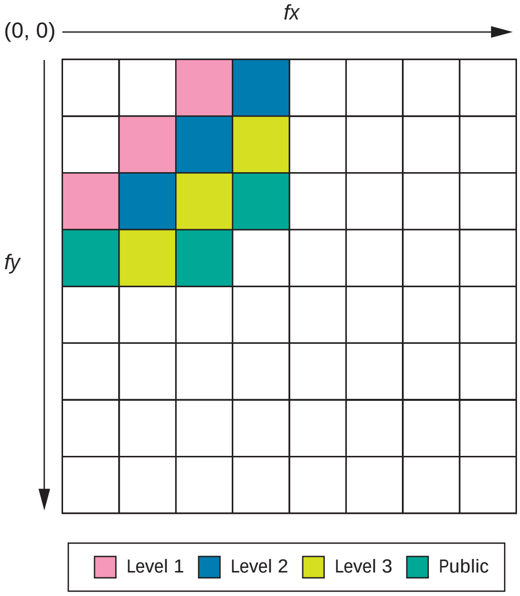
The algorithm is block based and shares some features with the JPEG standard for still-image compression.[9] The image's luminance component is divided into 8 × 8 pixel blocks. The algorithm selects a sequence of blocks and applies the DCT to each of the selected blocks. The transformed blocks are quantized with the luminance quantization table proposed in the JPEG

(0, 0) fx

fy

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

3 Luminance quantization table proposed by the JPEG standard.



4 Sub-bands used for watermarking,

standard and shown in Figure 3. The quantization step divides a DCT coefficient's value by an integer number that depends on the coefficient's position within the block matrix. Generally speaking, high-frequency coefficients are divided by higher quantization values than low-frequency coefficients. The integer values forming the quantization table can be multiplied or divided by a constant value to allow scaling of the quanitzation's impact on the coefficients.

Two components of the algorithm must be considered:

1. The position for the watermark embedding must be generated. To do this, we use a key to initialize a pseudo-random-number generator that determines the order of block processing and the coefficient to be modified within the block. The key may be public or secret, leading to a public or secret watermark.
2. We must embed the watermark. To do this, we must choose a method to modify the coefficients selected during the position-generation step. Even though the number of digital-watermark bits is not restricted by the algorithm, but rather by the number of suitable blocks in the frame, we will assume a fixed length of 64 information bits. This lets us redundantly embed the watermark, which is necessary for it to survive the MPEG-2 compression scheme.

The algorithm can embed up to four different, noninterfering watermarks in each frame. This is accomplished by dividing the frequency range for watermarking into four sub-bands, as Figure 4 shows. The sub-bands denoted as levels 1 through 3 are used for secret watermarking; the fourth band is used for public watermarking. We choose one sub-band prior to embedding or reading the watermark and use it throughout the video stream.

For each selected block, we encode one bit as follows (we use t_i to denote the absolute value of quantized DCT coefficients' absolute value; the subscript i identifies the coefficient position within the sub-band):

1. The pixel values within the block area are transformed using DCT.
2. A mechanism for detecting edges is applied that takes advantage of the block's DCT representation.
3. A pair of DCT coefficients (t_k, t_l) is selected from the appropriate sub-band of the transformed block. Each sub-band consists of three coefficients, leading to six possible coefficient pairs.
4. The selected coefficient set is quantized.
5. The set's t_i values are used to determine if the block is well textured and suited for watermark embedding.
6. Depending on the bit value to be embedded, t_k and t_l must hold a predefined relationship. The condition for encoding a "1" bit is $t_k > = t_l + d$; the relationship for encoding "0" is $t_k + d < = t_l$. If the required relationship does not already occur naturally, the coefficients are changed accordingly. This is equivalent to overlaying a 2D cosine pattern on the original block data. The impact of the pattern on the block's visual quality in the pixel domain can be scaled by adjusting the noise level d (the difference).
7. The changed coefficients are multiplied by the quantization value at the corresponding position of the quantization matrix and embedded into the DCT transformed block. The block data is inverse DCT transformed to the pixel domain, and the altered block is put back to its original position inside the image matrix.

To increase the watermark's robustness against the lossy MPEG-2 compression, the watermark is embedded with maximum redundancy. All blocks available in the video frame are subjected to the watermarking procedure.

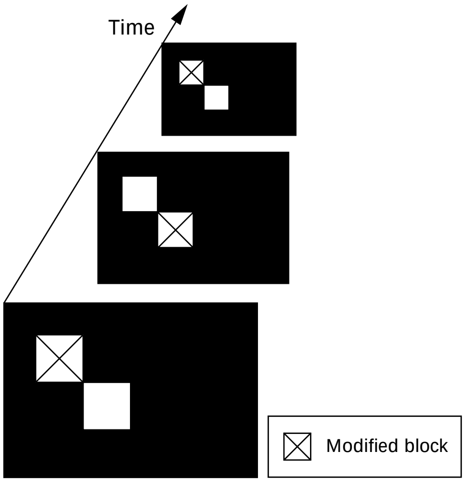
The retrieval process is symmetrical to the embedding procedure. The secret key is needed to find the correct sequence of blocks and the coefficients within them. These coefficients are evaluated, and if a pattern from the predefined set is found, the corresponding bit value is recorded. If no valid pattern is found ($t_k = t_l$) the bit is marked as not readable. Sixty-four "slots" are set up (one for each code bit), and each retrieved bit is put into the appropriate slot. After processing all blocks, the corresponding bits are used to reconstruct the original bits of the code word. The decision in each slot is based on its majority entries.

The visibility problem

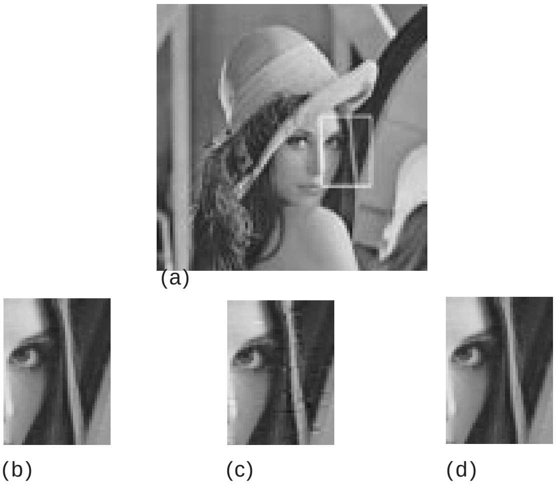
The transition from watermarking for still images to video sequences revealed properties of the original Koch-Zhao algorithm that escaped notice before. The most prominent feature of video sequences is the increased sensitivity to changes introduced by the watermarking process. Even for parameter settings that minimize the watermark's strength, watermarking artifacts are visible in high-quality digital betacam video sequences.

Figure 5 illustrates the visibility problem. The block position is no longer restricted to a single image (*x* and *y* axes) but extends to the time axis, *t*. The modification of blocks that are close to each other in *x* and *y* as well as in *t* can result in flickering effects. To avoid such degradation, video streams must be processed more carefully than still images.

Homogenous areas within frames are particularly sensitive to this type of degradation as are regions containing sharp edges. Two criteria for checking blocks before actually embedding the watermark information have been introduced: *edge detection* and *plain area detection* mechanisms. Figure 6 shows DCT artifacts in edge and homogenous areas. The effects are exaggerated to make the problem visible using a still image. Figure 6a shows the original clipped image; 6b shows the clip marked without the check algorithms. Here, artifacts in both homogenous and edge regions are clearly visible. Figure 6c shows the same part marked and checked for edge and homogenous blocks.



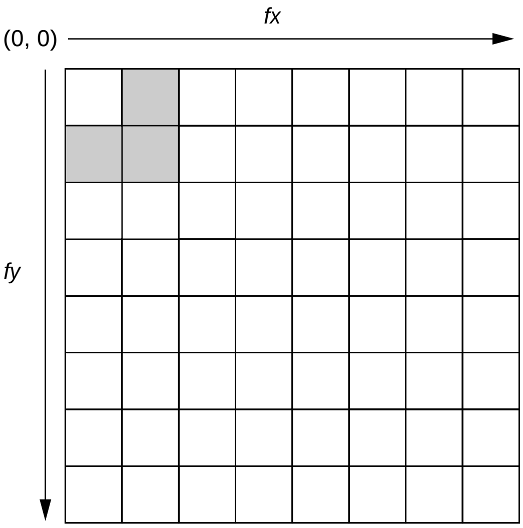
5 Visibility problem in consecutive video frames.



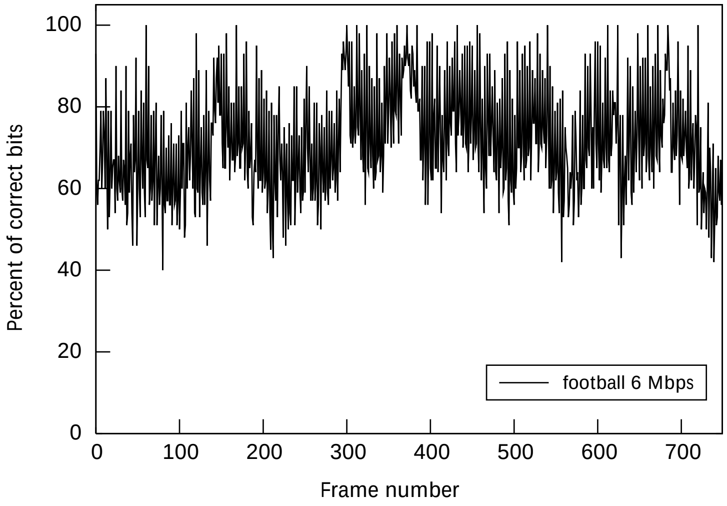
6 Visibility of edge blocks. The original image (a), the unmarked clipped image (b), and the image marked without the check algorithms (c). Artifacts are visible in both edge and homogenous regions. Once the check algorithm is applied, the artifacts are no longer visible (d).

Edge detection. Numerous edge-detection algorithms are available, ranging from simple ones like the Sobel operator to more sophisticated ones, such as wavelet-based schemes.[17] Because our algorithm aims at real-time capabilities, tools we use for edge detection must work fast. Even simple schemes like the Sobel operator have a computational complexity too high to be applied in real time.

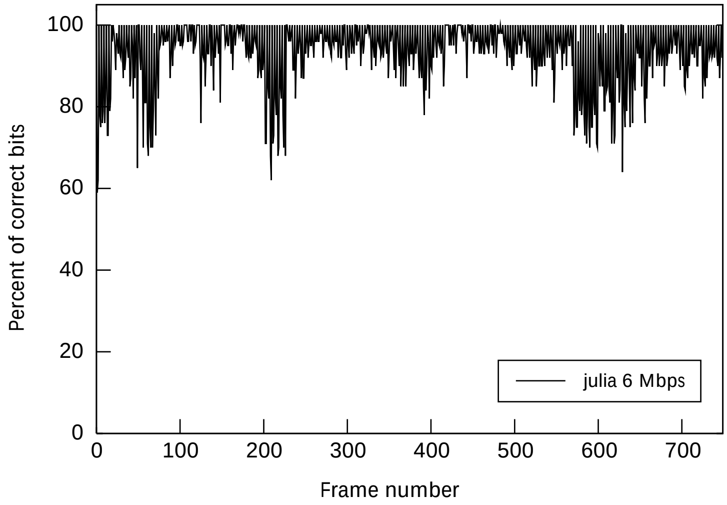
To minimize the overhead for edge detection, we use features computed within the basic algorithm cycle. The lowest frequency DCT coefficients of a transformed block can be used to decide whether the block contains an edge. We can illustrate this by looking at the DCT transform of a step function, where the lowest frequency terms have very high amplitudes compared to the transform of smoother functions. To separate edged blocks from textured blocks, we apply the following criterion: If the absolute value of one of three lowest AC coefficients exceeds a predefined static threshold, the block is classified an "edge block." Figure 7 shows the coefficients used for edge detection. We determined a threshold value of 40 through numerous heuristic experiments. Our goal was to detect all edge blocks without catching too many textured blocks well suited



7 Coefficients used for edge detection.



8 **Retrieval results for “football” at 6 Mbps.**



9 **Retrieval results for “Julia” at 6 Mbps.**

for watermark embedding. However, the threshold value depends on the specific implementation of the DCT transform used.Plain-area detection. The detection of smooth blocks is equivalent to a block’s texture analysis. To avoid computational overhead, we use a criterion based on the block’s quantized DCT coefficients. Instead of counting the number of non-zero coefficients in a transformed block’s predefined region, as proposed by Benham et al.,[18] we look at the set of quantized coefficients selected for modifications. If one of the set’s quantized coefficients equals zero, the block is classified as a “plain block.”

Treatment of invalid blocks. You can follow two basic strategies after detecting a block that may be unsuitable for watermarking. First, you could skip all invalid blocks, losing one code bit per block. The second strategy is an adaptive technique that reduces the amount of changes to the DCT coefficients to minimize the impact on the block’s visual quality. We propose a third strategy that attempts a compromise between these approaches. For blocks classified as plain, we adjust the modifications to lower values—that is, we reduce the changes to the coefficient amplitudes. Blocks classified as edge blocks are always skipped.

The watermark-reading process has to apply the same mechanisms as the embedding process. If an edge block is detected, it is ignored during watermark retrieval. For plain area blocks, the algorithm uses a higher reading sensitivity to evaluate the current bit value encoded in the block. The criteria for block checking prior to watermark embedding and retrieval must be stable. Changes introduced by watermark embedding or operations on the watermarked frame must not affect the criteria; it they do, block misclassifications significantly decrease the bit-retrieval rate.

Real-time considerations

Two components prove crucial to achieving real-time watermark embedding and retrieval:

- the interface between the algorithm and the luminance component of the digital video signal, and
- the signal processor that executes the algorithm’s core (that is, the DCT and IDCT).

We rewrote the DCT and IDCT to work on 16-bit integer values. The DCT only computes the quarter of DCT coefficient actually needed for watermarking (as shown in Figure 4). The IDCT has been restricted to transforming back only the changes introduced to the selected coefficient pair.

Shift resistance

Like other block-based watermarking methods, our algorithm is not resistant to geometrical transformations unless these transformations are inverted before watermark detection. Shifts can occur in some video installations, so we have to apply a mechanism for synchronization before monitoring the watermark. The Talisman system implemented a special watermarking algorithm for detecting a watermarked frame’s origin. This algorithm neither interferes with the core watermarking algorithm nor significantly degrades image quality.

Robustness against MPEG-2 encoding

Despite the transparency demands in watermarked video streams, the watermark must be robust against digital TV’s MPEG-2 encoding. The watermark information must survive MPEG-2 encoding, which is applied immediately before transmission.

Two features of the MPEG-2 standard are very important (and challenging) for watermarking algorithms:

- The MPEG-2 encoder is very effective at removing spatial and temporal redundancy from the video stream.
- The MPEG-2 encoder keeps the video stream's data rate constant. The data are compressed as much as needed to reach this goal. In rare cases, the encoder will skip complete frames.

We tested detection performance and invisibility of the watermarked video streams on 12 digitized sequences of digital betacam quality (ITU-R 601 format). Each sequence was 30 seconds long, corresponding to 750 video frames. The sequences were very different in character, ranging from feature films to synthetic clips.

We used the following test setup for each sequence:

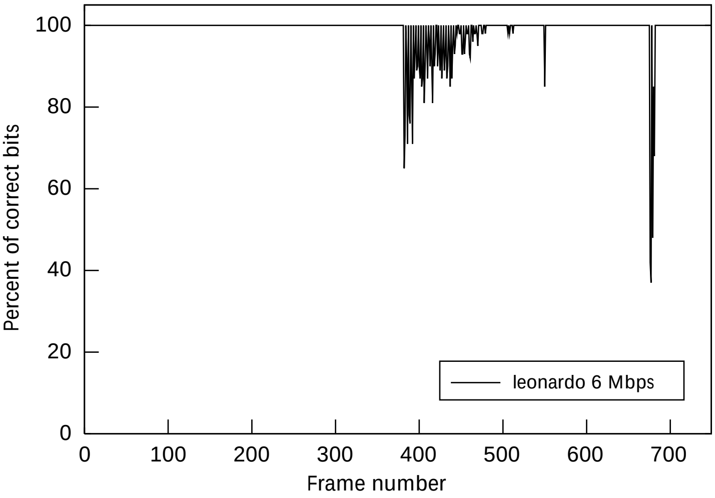
- First, we fixed the algorithm's parameters and watermarked the single frames, then stored them on disk.
- Next, we animated the frames using a high-quality M-JPEG codec (compression ratio 1:2 to 1:3) and estimated the perceptual quality of the watermarked sequence. If artifacts were visible, we adjusted the algorithm's parameters and generated a new sequence of watermarked frames. We repeated this procedure until we found no more artifacts.
- We input the watermarked frames into the software MPEG-2 encoder/decoder of the MPEG Software Simulation Group. The data were encoded at Main Profile and Main Level. Two data rates were tested, 4 Mbps and 6 Mbps.
- We decoded the data, stored it on disk as single frames, and read the watermark from each frame.
- Finally, we repeated all the steps until we found the maximum watermark retrieval rate without introducing visible artifacts into the original sequences. The resulting watermarked sequences were submitted to TV professionals, who checked for watermark visibility under studio conditions (digital betacam recorders and digital monitors). If they objected to the quality of the watermarked sequences, we modified the algorithm and repeated the test cycle.

We selected test results from four representative sequences to show the algorithm's performance and limits with respect to MPEG-2 compression. Figures 8 through 11 show the results with the frame number as the independent variable and the dependent variable as the percentage of bits retrieved correctly from a single frame. For each sequence, we tested 750 frames (30 seconds of video). The results are from watermarked sequences that professionals viewed as of acceptable quality. The data rate of the encoder was 6 Mbps.

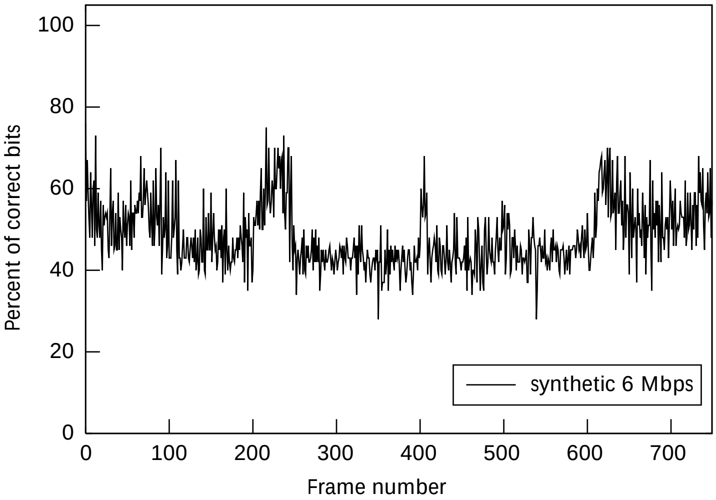
Figure 8 shows results from "football," a TV broadcast of a football match that had fast and medium movements. Figure 9 shows results from "Julia," an advertisement for a Hollywood movie that had fast and slow movement. Figure 10 shows results from "Leonardo," a high-quality TV production with slow movement. Figure 11 shows results from "synthetic," a rendered sequence of a truck with very fast movement.

The retrieval rate depends heavily on the sequence's characteristics. Two aspects of video watermarking must be considered here. First, sequences containing a natural noise level obviously suit watermarking well because more blocks pass the check for edges and texture. The same holds true for watermarking of still images.

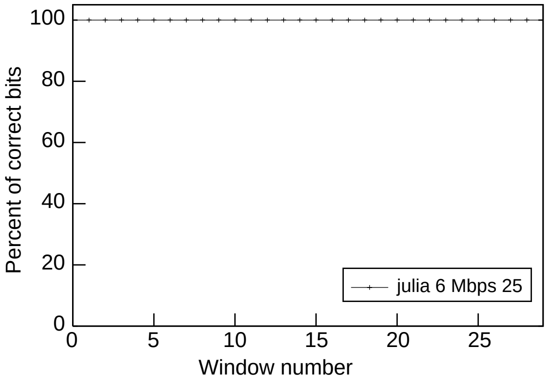
Second, as we noted above, MPEG-2 guarantees a constant data rate by exploiting the redundancy of videos in space and time. The sequence is decorrelated in time by



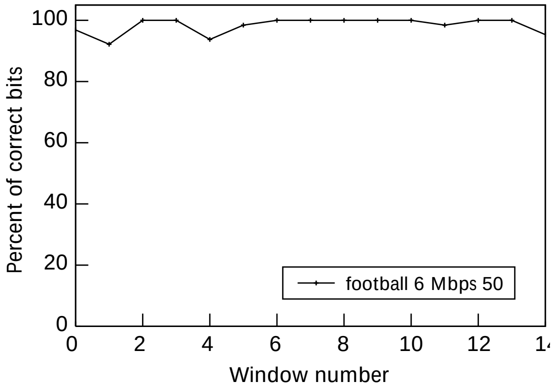
10 Retrieval results for "Leonardo" at 6 Mbps.



11 Retrieval results for "synthetic" at 6 Mbps.



12 Time integration of 25 frames for "Julia" at 6 Mbps.



13 Time integration of 50 frames for "football" at 6 Mbps.

a technique called *motion compensation*, which stores only the motion vectors that are needed to derive consecutive frames from each other. This mechanism works well as long as those frames don't differ too much. If the sequence is moving fast, MPEG-2 can't use motion compensation, and the frame-based lossy compression is enforced.

The test results reflect these issues. "Julia" is a real-world sequence with a noise margin sufficient for watermarking. Though the sequence contains some fast moving parts, the embedded watermark's redundancy enables it to survive the compression. The "football" sequence behaves very similarly, although the background in many frames is uniform and more blocks are classified as "plain." "Leonardo" is a very calm sequence; it has high visual quality and contains little noise. Nonetheless, the retrieval rate is very high because high compression can be achieved by motion compensation. "Synthetic" shows the worst case of video watermarking. The sequence contains no noise at all and changes very quickly. The retrieval rate is far from significant. (Retrieval rates below 50 percent are possible because blocks with $t_1 = t_2$ are classified as "not readable.")

Integrating the retrieval results for single frames can help overcome the encoder's influence without increasing the strength of the watermark insertion process. Essentially, we can exploit the retrieval results of 25 or 50 consecutive frames in the same way as single bits are read within a frame. This significantly increases the detection rate. Figures 12 and 13 show the time integration of two sequences; we used the integrated window as the independent variable and the percentage of correct bits as the dependent variable. Figure 12 shows that integrating 25 frames of the "Julia" result from Figure 9—an integration window of 25 frames, 1 second of PAL video—is sufficient to detect the watermark with very high accuracy. The result of integrating 50 frames of the "football" result appears in Figure 13.

Conclusions

Our algorithm for watermarking and monitoring video streams in a TV-broadcasting environment survives MPEG-2 compression of high-quality, real-world video sequences without degrading their quality. Applying the algorithm to fast-moving synthetic video sequences requires much longer time-integration intervals than the 50-frames-wide window we used in the tests presented here. The algorithm's current version is well suited for watermarking digital video streams such as movies or sporting events.

The format of digital video is restricted to some well-defined geometry. This makes geometric distortions detectable and removable. To introduce geometric distortions, an attacker would have to decode the video stream, process it, and encode it again.

Much video data is only valuable as a live event, such as a soccer match. This significantly reduces the value of the data when stored for later broadcast. Thus, because pirates must attack the watermark in real time, the cost of the attack increases.

Future work will focus on the reconstruction of the original geometry of distorted watermarked video frames and more elaborate time integration methods. Other topics to be considered are copyright protection schemes for MPEG-4 coded video data and synchronized watermarking of the video and audio channel of bit streams. ■

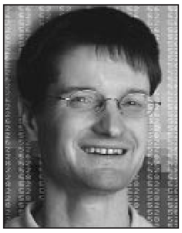
References

1. A. Tirkel et al., "Electronic Watermark," *Digital Image Computing, Technology and Applications—DICTA 93*, Maquarie University, Sidney, 1993, pp. 666-673.
2. I. Cox et al., "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on Image Processing*, Vol. 6, No. 12, Dec. 1997, pp. 1673-1687 (originally published in 1995 as NEC TR 95-10).
3. I. Pitas, and T.H. Kaskalis, "Applying Signatures on Digital Images," *IEEE Workshop on Nonlinear Image and Signal Processing*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 460-463.
4. E. Koch and J. Zhao, "Towards Robust and Hidden Image Copyright Labeling," *Proc. 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 452-455.
5. I. Cox and M. Miller, "A Review of Watermarking and the Importance of Perceptual Modeling," *Human Vision and Electrtonic Imaging II,* No. 3016, Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, Wash., 1997, pp. 92-99.
6. Low et Al., "Document Marking and Identification Using Both Line and Word Shifting," *Infocom 95*, IEEE Computer Society Press, Los Alamitos, Calif., 1995.

7. B. Schneier, *Applied Cryptography*, 2nd edition, John Wiley & Sons, New York, 1996.
8. A. Kerckhoffs, "La Cryptographic Militaire," *J. des Sciences Militaire*, Ninth Series, Feb. 1883, pp. 161-191.
9. K. Rao and J.J. Hwang, *Techniques & Standards for Image, Video and Audio Coding,* Ch. 11, Prentice Hall, Upper Saddle River, N.J., 1996, pp. 273-322.
10. F.A. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Attacks on Copyright Marking Systems," *Information Hiding: Second Int'l Workshop,* Vol. 1525 of LNCS, Springer-Verlag, Berlin, 1998, pp. 219-239.
11. S. Wolthusen, "On the Limitations of Digital Watermarks: A Cautionary Note," *Proc. of World Multiconference on Systemics, Cybernetics, and Informatics (SCI 98) / Int'l Conf. on Information Systems Analysis and Synthesis (ISAS 98)*, Vol. 4, Int'l Institute of Informatics and Systemics (IIIS), July 1998.
12. U. Reimers, "Concept of a European System for the Transmission of Digitized Television Signals via Satellite," *SMPTE J.*, Vol. 103, No. 11, 1994, pp. 741-747.
13. Talisman Project, 1998, http://www.tele.ucl.ac.be/TALISMAN.
14. J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust MPEG Video Watermarking Technologies," *ACM Multimedia 98*, ACM Press, New York, 1998, pp. 71-80.
15. F. Hartung and B. Girod, "Digital Watermarking of Raw and Compressed Video," *Proc. SPIE 2952: Digital Compression Technologies and Systems for Video Communication*, Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, Wash., 1996, pp. 205-213.
16. L. Qiao and K. Nahrstedt, "Watermarking Methods for MPEG Encoded Video: Toward Resolving Rightful Ownership," *IEEE Multimedia Computing and Systems Conf.*, IEEE Computer Society Press, Los Alamitos, Calif., 1998.
17. S. Mallat and W. Hwang, "Singularity Detection and Processing with Wavelets," *IEEE Trans. on Information Theory*, Vol. 38, No. 2, IEEE Computer Society Press, Los Alamitos, Calif., 1992.
18. D. Benham et al., "Fast Watermarking of DCT-based Compressed Images," *Proc. Int'l Conf. on Imaging Science, Systems, and Applications*, CSREA Press, Athens, Ga., 1997.

***Christoph Busch** is head of the Department of Security Technology for Graphics and Communication Systems at the Fraunhofer Institute for Computer Graphics, where he is responsible for the acquisition, management, and control of various applied research and development projects. He is also a lecturer on applied wavelet transforms in the educational program of the Computer Graphics Center and is currently a partner in several European projects, including ACTS' Okapi, Talisman, and Ocatlis projects and Esprit's Aimedia project, all of which deal with copyright protection and conditional access for interactive multimedia services.*

Busch studied geodetic sciences at the Technical University of Darmstadt, where he received a PhD in computer graphics in 1997.



***Wolfgang Funk** is a member of the research staff at the Fraunhofer Institute, where he works on digital watermarking, biometric authentication methods, and image and video processing. Funk received his diploma in physics in 1994 from the University of Würzburg, Germany.*

***Stephen Wolthusen** is an assistant scientist at the Fraunhofer Institute. His research interests include digital watermarking, host and network security, cryptography and cryptanalysis, and number theory. He is currently working on a distributed digital content security system that incorporates cryptography and watermarking. Wolthusen is also working toward a diploma in computer science at the Technical University of Darmstadt, Germany.*

Readers may contact the authors at the Fraunhofer Institute for Computer Graphics IGD, Department Security Technology for Graphics and Communication Systems, Rundeturmstr. 6, D-64283 Darmstadt, Germany; {busch, wfunk, wolt}@igd.fhg.de