Combined video and audio watermarking: Embedding content information in multimedia data

Jana Dittmann[a], Martin Steinebach[a], Ivica Rimac[b], Stephan Fischer[b], and Ralf Steinmetz[a,b]

[a]GMD - German National Research Center for Information Techology, Darmstadt, Germany
[b]Darmstadt University of Technology, Merckstr. 25, 64283 Darmstadt, Germany

ABSTRACT:

Audio and video watermarking enable the copyright protection with owner or customer authentication and the detection of media manipulations. The available watermarking technology concentrates on single media like audio or video. But the typical multimedia stream consists of both video and audio data. Our goal is to provide a solution with robust and fragile aspects to guarantee authentication and integrity by using watermarks in combination with content information.

We show two solutions for the protection of audio and video data with a combined robust and fragile watermarking approach. The first solution is to insert a time code into the data: We embed a signal as a watermark to detect gaps or changes in the flow of time. The basic idea uses numbers increasing by one. If in the verification process the next number is smaller than the last one or the step is greater than one, the time flow has been changed. This is realized without the combination of video and audio data. But we can synchronize the two data streams. A time signal is only valid if the combination of audio and video signals satisfy a certain attribute. To keep the basic example: If we embed an increasing number in the audio and a decreasing number in the video, we could test if the combination of both always equals zero.

The second solution is more complex: We use watermarks to embed information in each media about the content of the other media. With the help of speech recognition technology it is possible to embed the spoken text, the content, of an audio file in the video. With an algorithm previously developed in [1] we extract video content representation which is embedded in the audio stream.

In our paper we present the problem of copyright protection and integrity checks for combined video and audio data. We show our two solutions and discuss our results.

Keywords: Collusion secure fingerprinting, watermarking, copyright protection, customer copy identification

1. MOTIVATION

The expansion of digital networks all over the world allows extensive access on, and reuse of, visual material. Problems include unauthorised taping, reading, manipulating or removing of data, which might lead to financial loss or legal problems of the producers and creators. Thus, designers, producers and publishers of digital data like images, video or multimedia material are seeking technical solutions to the problems associated with copyright protection of multimedia data. Systems are required which provide environments where digital data can be signed by authors or producers as their intellectual property, i.e. by embedding private or public information into the video data, to ensure and proof ownership rights on the produced video material during its distribution.

Digital watermarking is recognized as a possible solution to this problem. It is the enabling technology to prove of ownership on copyrighted material, detect the originator of illegally made copies, monitor the usage of the copyrighted multimedia data and analyze the spread spectrum of the data over networks and servers.

There is one major difference between multimedia content and watermarking technology: Watermarking always concentrates on one type of media, multimedia combines different kinds of media, like images and audio. In this paper we introduce a way to connect the different media types to increase security for authors and users. Our goal is to provide a solution with robust and fragile aspects to guarantee authenticity and integrity by using watermarks in combination with content information. We combine audio and video watermarking, with the media referring to each other, building a linked structure with mutual information about the content of the linked parts.

In section 2 we present a synchronization based approach, where time codes are inserted in the media. In section 3 an advanced approach is introduced: content fragile watermarking, using feature information instead of time codes. First test results are given in section 4, in section 5 we describe open problems and subjects of further research. The conclusion in section 6 summarizes the results and perspectives of this paper.

2. SYNCHRONISATION BASED APPROACH

Inserting a time code into multimedia data is a efficient way to ensure integrity. If, for example, a part of an audio clip is erased, a gap in the retrieved time signals occurs, and the change is detected.

The most basic attack against the integrity of a multimedia stream or a combination of different multimedia is the removal of an unwanted small part. Another basic attack is the change of positions of parts of one stream while leaving the other as is, thereby attacking the synchronization of the streams. These attacks can be defeated by inserting time code information inside both media. Any removal of a frame or position changes can be used as structural attacks. They are detected because the flow of the timing signal is distorted.

The first design decision concerning synchronization schemes is to choose between absolute and relative time codes. An absolute time is a complete time stamp repeated over the signal, e.g. (00:00:00;00:00:01;...23:59:59) while a relative time code is a counter like on a tape deck using increasing numbers to generate specific accessible points in the recordings, e.g. (000,001,.., 999).

A well-know absolute synchronization time code is SMPTE (Society of Motion Picture and Television Engineers) [2], which is used in video and music production. It offers a timing signal for a whole day (hh:mm:ss:ff, where f stands for frame) with a resolution down to single video frames. If a time code like SMPTE could be embedded in a multimedia stream as a watermark, every position in the stream would include a complete timing information. Checking time integrity would be a simple task.

We use a time signal with dual synchronization properties. Figure 1 shows a video and an audio stream with embedded time codes. The embedded watermark works as an absolute time code like the one in the upper right corner of figure 1, while the data the watermark consists of is used as a relative time code. The absolute time code is used to synchronize the different media streams. It is encoded as a watermark using a secret key. No third party must be able to rewrite a time code after changing the content thereby obscuring attacks.

As robust watermarking technology is used to embed the absolute time code, very small parts of a media stream could be deleted without destroying the watermark. Very small changes can not be detected. The second feature offers protection against this. Figure one shows that an absolute time code is embedded as a sequence of bits. The encoding parameters for these bits can be used as a relative time code if they are changed every time a bit is embedded. Figure 3 shows an example further described in section 2.2. When the time code is read, synchronization in the single media is checked for every bit of the watermark, and after retrieving the complete watermark, synchronization of the different media is checked.

The data rate for a SMPTE time code is up to 2400 bits per second, much more then current watermarking technology can offer. Therefor we use a smaller time code, designed to give high resolution timing information at low data rates. A time code consisting of hour, minute and second (hh:mm:ss) has a range of {0-23}, {0-59}, {0-59} and can be encoded with {4} + {6} + {6} Bit. 16 bps is a data rate most watermarking algorithm can provide.

Section 2.1 and 2.2 describe how this synchronization concept is realized for video and audio data. Two approaches are used. The time code for the video stream is used to initialize the watermarks to be embedded, the time code for the audio is the content of the watermark. The later one is a direct method while the first is an indirect one.

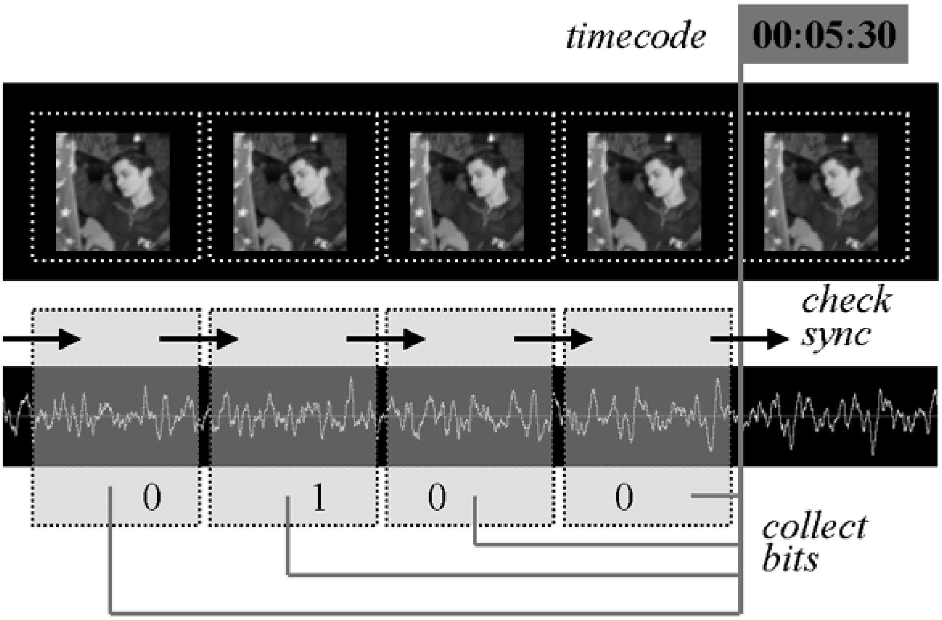

Figure 1: Video and audio data is checked for synchronous timing

2.1. Synchronisation Authentication with robust Video Watermarking

Our used watermarking method in the video domain is based on overlaying a pattern with its power concentrated mostly in low frequencies. The pattern is created using a pseudo random number generator and a cellular automaton with voting rules. The basic ideas has been shown by Fridrich, [3] and we have extended his work to blind watermarking techniques which can embed binary information [4].

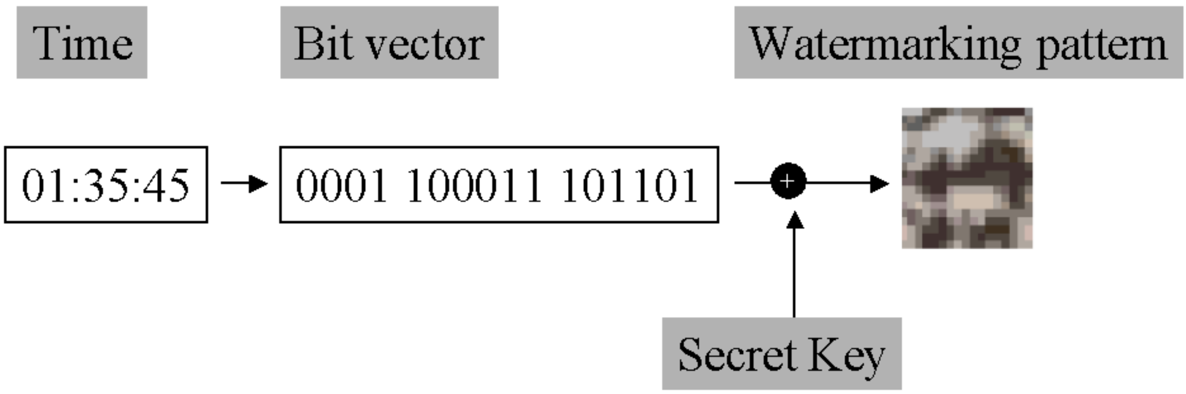

Figure 2: A time code in transformed in a binary vector coded in a watermarking pattern using a secret key

We start by creating a 8x8 pseudo random pattern M with the time code and user key as a seed. To eliminate the high frequencies in this pattern a cellular automaton with simple voting rules is used. Every position in the 8x8 random pattern is tested on the number of '1' in the eight co-sited positions. If the number exceeds five the actual position is set to '1' too, if the number is less than 3 it is set to '0', see the marked rectangle for an example. By applying these rules several times on the whole 8x8 block we obtain a pattern M with less high frequencies. Now a correlation between the pattern and the luminance block has to be inserted, we add the pattern to the luminance blocks of the image. We generate an 8x8 block to avoid visual distortions using a smooth-block/edge detector, described in [4]. Due to the fact that we use 8x8 patterns, we can embed the time code watermarking pattern redundantly to improve robustness against content-preserving operations. This redundancy can be used as a relative synchronisation tool, as we will show in section 2.2.

In the retrieval process the same 8x8 pattern M has to be generated as in the embedding process. To test the correlation between the luminance block and the pattern M the average luminance value *av1* of positions with a corresponding '1' and the average luminance value *av0* (sum0 div #0) with a corresponding '0' in the pattern M is produced. If the luminance block and the pattern M would be uncorrelated the difference of both values should be near zero. But due to the embedding process sum1 of these values should be significantly higher (around 2*k) than sum0. Thus we estimate an embedded pattern if *av1>av0*. Otherwise we estimate that this pattern was not embedded. With this statistical analysis we avoid using the original frames and can decide if the synchronization is correct or not to detect manipulations.

2.2. Synchronization Authentication with robust Audio Watermarking

With a data rate of 16 bit per second we can embed a complete time code with a resolution of one second into the media. We introduce a method to embed a watermark with the dual synchronization properties mentioned above using our mp2-algorithm. It embeds watermarks into MPEG audio Layer 2 files using patterns in scale factors [5].

Given a MPEG-file, a bit vector to embed and a secret key we encode the binary data into a sequence of patterns in the scale factors of the frames of the MPEG-file. Difference patterns based on this scale factors are calculated and the central algorithm changes these patterns until a sufficient number matches our desired sequence of patterns. The whole watermark is inserted in this way, if there are more frames than needed the watermark is inserted multiple times. Then the new scale factors are written in the source file, overwriting the old ones and so creating a watermarked MPEG-file.

We add another bit to the 16 time code bits for synchronization purposes, the sync-bit. This makes is easy to resynchronize when only a part of the watermarked file is available. During the retrieval of the watermark, the data bits are dropped until a synchronization bit is found. Our watermark to be embedded into one second of data will be w = $(\text{sync},\{0,1\}^{16})$. As we embed a watermark every second, this is the maximum time until a synchronization point is found. The pattern for the sync-bit stays the same over the whole file. The sequence of the following bits is encoded using a sequence of patterns instead of a fixed pattern as in the basic algorithm.

Figure 3 shows how the algorithm distributes the patterns over the frames. The watermark W consists of a synchronization bit (S) and a binary vector. The algorithm distributes it over the frames (F) of the MPEG file (mp2). Three different pattern types are used, {sync}, {p0} and {p1}, the later two to embed the binary data. An index is given with the patterns to show how the algorithm uses different patterns {p0} and {p1} while embedding the complete time code. Even very small changes, e.g. dropping or swapping of frames, can now be detected as the index is corrupted.

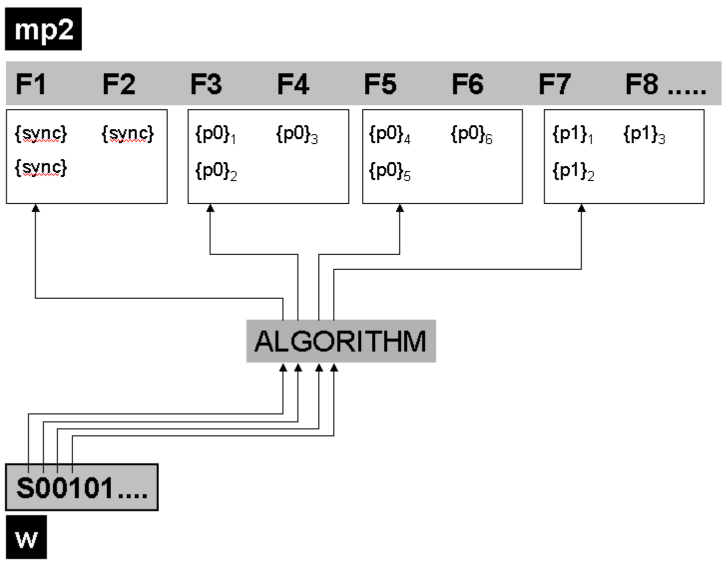

Figure 3: A time code is distributed to include relative timing information

3. FEATURE BASED APPROACH

3.1. Content-fragile Watermark

The concept of a content-fragile watermark [1] can be described as using a robust watermark to embed content information for integrity verification. This information can be compared with the actual content. If changes have been made, content and watermark differ, and a warning message is prompted. Fragility is about losing equality of extracted and embedded content in this case. The idea of content-fragile watermarking is based on the knowledge that we have to handle content-preserving operations, manipulations which do not manipulate the content.

The well-known problem of "friendly attacks" occurs here as in any watermarking scheme: Some signal manipulations must be allowed without breaking the watermark. In our case, every editing process that does not change the content itself is a friendly attack. Compression, dynamics, A/D/D/A-conversion and many other operations that only change the signal but not the content described by the signal should not be detected content manipulation. The idea is to use content information as an indicator for manipulations. The main problem is, which media features are appropriate to distinguish between content-preserving and content-changing manipulations.

We describe a content extraction algorithm for image data in detail in 3.2. Additionally we describe an algorithm for speech in section 3.3. The watermarking algorithms used to embed the content information are the same as in section 2. Only the embedded data differs from the synchronization approach.

Content fragile watermarking is the combination of two technologies: Robust watermarking for embedding and retrieval and feature detection for information extraction. In the following two sections we describe how content can be detected and extracted of video and audio data. The watermarking algorithms we use are described in section 2. For video watermarking, we use a indirect method where watermarking patterns are initialized using content information and a secret key. For audio, we use a direct method where content data is embedded as a watermark using a secret key.

3.2. Video Feature

The main concept of content detection for images is to extract the image characteristics of human perception, called content, which will be used for watermarking. The idea is to determine the edge characteristics of the image or single video frame and transform them into a feature code for the content-fragile digital watermark [1]. The edge characteristics of an image give a very good reflection of the image content, because they allow the identification of object structures and homogeneity of the image. We are using the canny edge detector described as the most efficient edge separator in [6].

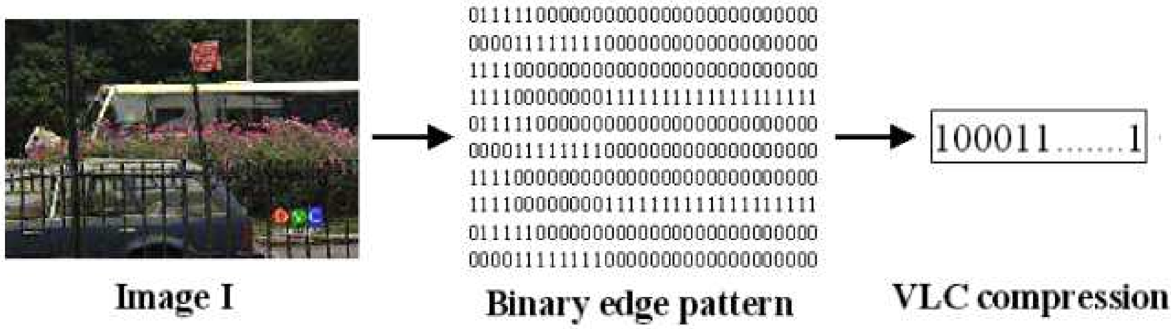

Figure 4: Feature extraction by edge detection

Canny has defined three criteria to derive the equation of an optimal filter for step edge detection: (1) good detection, (2) good localization, and (3) low-responses multiplicity. The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries. In his report [7] Canny developed mathematical formula for a one-dimensional edge profile in order to facilitate the analysis. The result of the edge detection process is a new image consisting of gray and black values for the edges and white for the background, called edge characteristic C. The algorithm does not give contours as result. Our task is now to transform this image into a binary contour characteristic: the so called binary edge pattern EP. The edge image is processed in the following way:

1. Processing the image edges: The Canny algorithm produces edges with different gray values, background is white. Values close to the color black represent more intensive edges and are replaced by black. Edges close to the value white are replaced by white. With the parameter *detail [0...1]* we can influence the recognized edge details, default is 0.5. As the result we get a new black/white edge image with the most relevant edges represented by black.
2. The black/white edge image is now scanned:. If the pixel value is black then we store a '1' else the value is white and we store a '0'. Finally we get the binary edge pattern *EP* of the image which will be used in the following algorithm for content based signatures. This post processing is intended to provide a conversion from the edge detection result image to almost binary representation of edges. The ideal result should be a sequence of points along with topological connections indicating the path of edges. Here we provide only a binary scheme for edge location.

To generate the feature code we extract the edge characteristics C_I of the image I or video frame with the Canny edge detector E (feature extraction). Second we transform it into a binary edge pattern EP_{CI} introduced in the section above. Third we integrate a tolerance threshold to improve the recognition of content-preserving changes like quantization and scaling without loosing the sensibility of the manipulation recognition. Based on the edge characteristic we build a binary block edge pattern over 8x8 blocks of the image. If the 8x8 block is crossed by an edge, then the binary block edge pattern contains a "1" else a "0". To recognize manipulations, we can also count the crossing edges. We get another abstraction level based on 8x8 blocks. To determine the acceptance rate of content-preserving and content-changing modifications block size can be variably chosen. Forth we make a VLC(Variable Length Code) for data reduction to produce the feature code.

1. *Feature extraction:* $C_I = E(I)$
2. *Binary edge pattern:* $EP_{CI} = f(C_I)$
3. *Block analysis:* EP_{CIB}
4. *Feature code:* $VLC(EP_{CIB})$

The feature code will be used in section 3.3 as integrity information.

A major problem of the calculated feature code not addressed in our current approach are color manipulations. If the edge characteristic remains the same (same object), it is possible to change the content by manipulating the color impression, for example the structure of a flag (stripes) could be the same but the color could be manipulated to confuse the observer in a news channel. Furthermore there are problems with images without edges where we have only slight color effects. Changes of the color could not be detected.

3.3. Audio Features

Similar to the previous subsection, features for audio content description are needed to ensure the integrity of the audio stream. The extracted feature data has to be robust against all allowed attacks, such as filtering, compression or MP3 encoding, and it has to be coded with significantly less data then the original signal. The latter constraint is important since we want to embed the audio code into the video as a watermark.

To avoid high computational costs our approach is based on a low-level feature. Our idea is to determine signal sequences of equal sign in the audio track of a video and to transform them into a feature code representing the content of the signal. It is obvious that the inversion of the samples of an audio signal does not change the content but is performed easily. Thus, we rather chose the changing of the signs of the sequences as the feature for speech content verification, instead of the sign of each signal sequence in between two consecutive zero crossings.

Our algorithm can briefly be described as follows:
1. The signal is filtered through a bandpass with edge frequencies of 500Hz and 3kHz.
2. The filtered signal is parsed and all zero crossings are detected.
3. The number of samples between two consecutive zero crossings are counted.
4. The length of equal signed sequences *ess* is compared to the given threshold th_{ess}. Sequences, which are shorter than the minimal value th_{ess} are skipped, and only those sequences which exceed the threshold produce an output:
   a) a “0” if the sign of a previous sample sequence equals the sign of the current sequence,
   b) a “1” if the signs of the consecutive sequences differs.

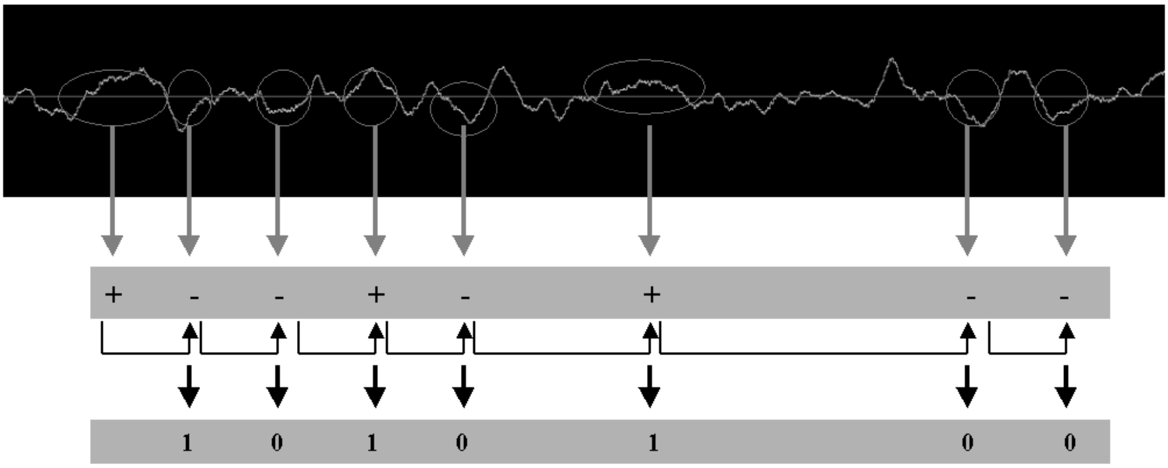

Figure 5: Sings in the audio data are coded in a binary vector

Note that since short sequences are skipped it is possible and likely to extract a code with multiple “0” or “1” in a row, e.g. “...0001011...”.

In our initial version of the algorithm we work with the PCM file format for simplicity reasons. We implemented an algorithm to extract the *ess* out of an PCM audio file and gained first experience with our approach applying it to audio data of recorded lectures. We found out that the algorithm performed well with a minimum number of *ess* of 40-50 for this material, but tests with non speech audio data confirmed our assumption, that the threshold th_{ess} strongly depends on the underlying data. Thus a very important task is the estimation of the minimal length of equal signed sequences.

In an extended version of our algorithm we introduced a parameter to set the wanted number of coding bits per second. Starting with an initial threshold, iteratively the value of th_{ess} is modifies until the amount of coding bits reaches the specified number.

3.3. Combining audio and video features for content-fragile watermarking

Both video and audio features we extracted in 3.1. and 3.2. are used in our combined media watermarking scheme. We calculate a binary vector based on the retrieved content features of both media. This vector is used as the content of the watermarks embedded in both media. Figure 7 illustrates this process.

To combine the content features we can use binary operations, as the output format of both content detectors is a binary vector, as shown in figure 6.

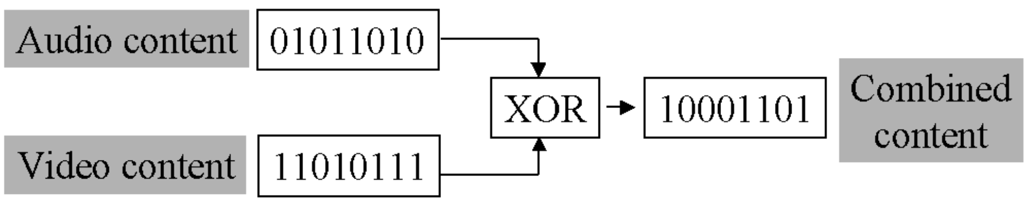

Figure 6: combined content description with binary operators

To embed the watermarks we use a direct method for audio, where content data is inserted with an secret key, and an indirect one for video, where patterns are initialized with the content data and a secret key.

With this mutual watermarking scheme we ensure integrity and synchronization. The video stream watermark describes the video stream and the belonging audio stream and the audio stream describes itself and the video stream. Every time a image or a significant amount of audio information is deleted or changed, the other media will point out this modification.

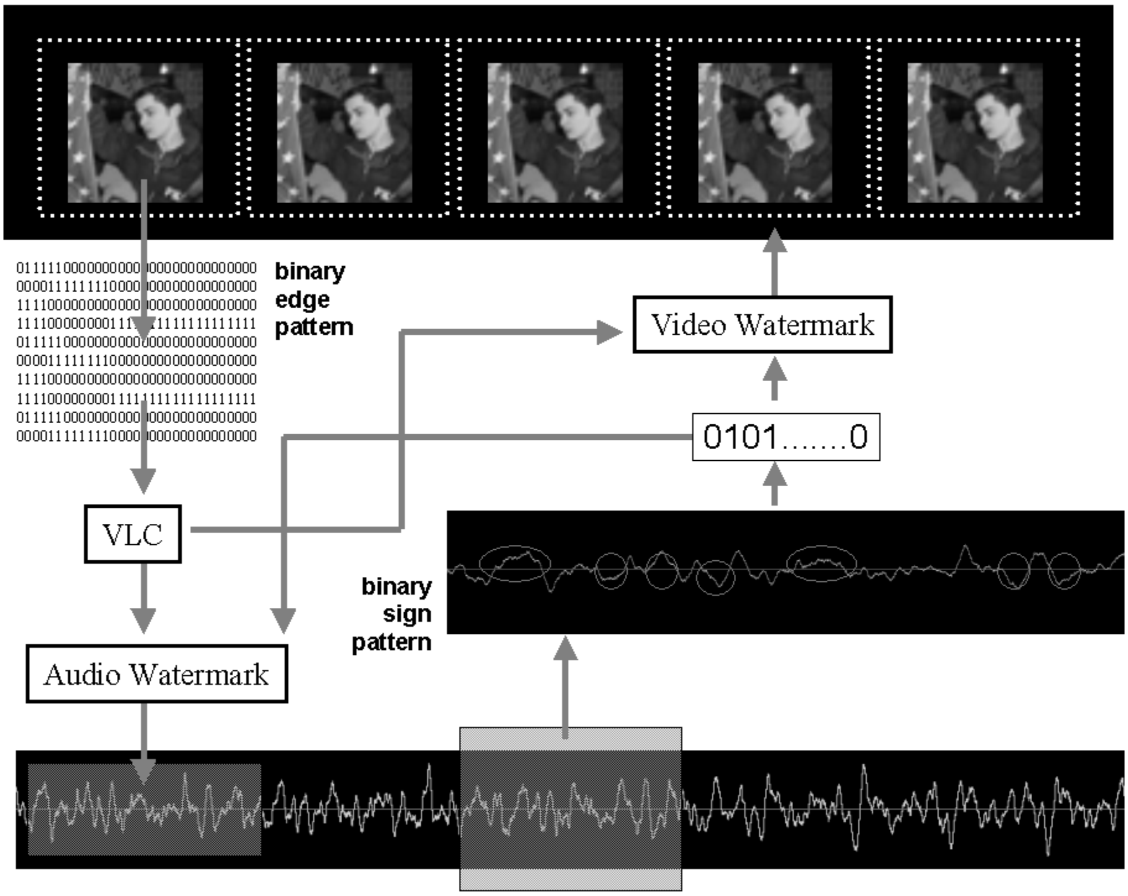

Figure 7: Content fragile watermarking scheme

Post production operations that do not change the content of the image will not destroy the robust watermarks. A producer can work with the watermarked material and can transmit it, a receiver can check for integrity with the help of a trust center providing the keys, for example. A possible application could be: A reporter records an interview with a video camera. He protects the resulting audio and video stream with a content fragile watermark, then converts it to MPEG. He sends the file to a news distributor. They sell it to a internet news web page. Several post production operations like scaling, cutting and normalization are performed for news production. The edited video placed within other news. A viewer is skeptical about the interview, trusts the reporter, but maybe does not trust the news provider. He sends the received file to a trust center to check if the content has been changed. The trust center sends back a positive or negative result of the content watermarking verification.

4. TESTRESULTS

In this section we describe first test results with the algorithms introduced in section 3. They have been implemented to verify our methods of content detection and to test the robustness of the extracted features against manipulations. We do not discuss test results for the watermarking algorithms. They can be found in [4] and [5].

4.1. Video Features

Our edge based approach recognizes all image manipulations except single pixel modifications, which do not effect edges. The approach is invariant to main content-preserving manipulations. Beside these advantages we have still some problems with high compression, quantization and scaling, because the edges are slightly moved. A solution for this problem is to use a tolerance threshold which can improve the recognition of content-preserving, using a binary block edge pattern detection and manipulation recognition [1]. Beside the introduction of a threshold, a major problem not addressed in our current approach are color manipulations. If the edge characteristic remains the same (same object) it is possible to change the content by manipulating the color impression, for example the structure of a flag (stripes) could be the same but the color could be manipulated to confuse the observer in a news channel. The edge detection approach offers first interesting results regarding the image manipulation detection. The development of attribute extraction will also be combined with object recognition. Another topic is the integration of color characteristics and the design of an intelligent cellular automaton for generation of watermarking patterns which are robust against quantization and scaling in addition to robust feature codes.

4.2. Audio Features

We implemented our audio feature extraction algorithm described in section 3.3. It was applied on 1.5 hour of a recorded lecture and to some non speech audio data. To prove the robustness of our content feature we extracted the features of the original data and of manipulated data. The resulting equal sign sequences were compared. They remained the same after mp3 compression or equalizer application. We experienced better results with audio data with significant features. Depending on the examined audio data pitch shifting and time stretching produced errors.

The result of this tests is positive. We can extract features out of audio data by looking at the equal sign sequences and they are robust against manipulations that do not change content. Other manipulations distort the equal sign sequences and are detected.

5. OPEN PROBLEMS

The features for audio content extraction are a subject of further research. The sign detection is a first, simple approach. But as its reliability varies with different audio categories. Parameters are necessary to adapt it to different situations. The equal sign sequence length of music and speech is different, a working scheme for both cases has yet to be found. Also, testing with this feature has been done only for a few cases and examples.

There are a number of video and audio features described in publications [8], [9] which could be used as alternatives for content fragile watermarking feature detection. A combination of different features could also be a solution to the problems we experienced with different audio content types.

Types of attacks have to be investigated further. Detecting manipulations but ignoring allowed signal operations is essential for a appropriate content-fragile watermark. False alarms are as bad as failing to detect a manipulation. Both reduce the trust in the solution. Our goal is a feature that survives typical treatment in a post production studio, with operations like equalization, normalization, compression, limiting and denoising.

The final result of this research will be an environment to ensure trustworthiness of multimedia streams. Therefore an additional concept will be necessary to combine different feature detection algorithms with watermarking insertion, detection and a intuitive user interface.

6. CONCLUSION

In our paper we describe an approach to ensure the integrity of multimedia streams using a combination of robust watermarking and content extraction technologies. We discuss the weakness of existing watermarking solutions which usually concentrate on a single media type and explained the advantages of a mutual security concept.

Our concept of synchronization watermarking is a first solution to protect the integrity of multimedia streams. We show how existing robust watermarking algorithms can be used to embed a absolute time code and also how to use redundancy as a relative timing information.

We introduce the “content fragile watermark” where fragility is not about destroying the watermark but about loosing the identity between extracted and embedded information. A major problem is the robustness of the feature extractions against content-preserving post production operations. The content-fragile watermark must not be destroyed by these operation but has to detect content-changing operations.

We briefly discussed audio and video watermarking and feature extraction algorithms. Each of these algorithms can be improved. Robustness of the watermarking schemes and quality of the feature extraction are topics of further research.

Our future work will mainly address the improvement of feature extraction. Our intention is to verify our concepts in an application environment and developing further ways of mutual multimedia security.

REFERENCES

1. J. Dittmann, A. Steinmetz, R. Steinmetz*: Content-based Digital Signature for Motion Pictures Authentication and Content-Fragile Watermarkin*g, Inn Proc. of IEEE Multimedia Systems, Multimedia Computing and Systems, June 7-11, 1999, Florence, Italy, Volume1, pp. 574-579, 1999
2. Philip Rees, *Synchronisation and SMPTE timecode* ,1997, http://www.philrees.co.uk/articles/timecode.htm
3. Fridrich, J.: *Methods for data hiding*, Center for Intelligent Systems & Department of Systems Science and Industrial Engineering, SUNY Binghamton, Methods for Data Hiding", working paper (1997)
4. Dittmann,, Jana, Stabenau, Mark, Steinmetz, Ralf: *Robust MEG Video Watermarking Technologies*, Proceedings of ACM Multimedia’98, The 6th ACM International Multimedia Conference, Bristol, England, pp. 71-80
5. Dittmann, Jana, Steinebach, Martin, Steinmetz, Ralf, *Digital Watermarking for MPEG Audio Layer 2*, to appear in Proceedings of ACM Multimedia’99
6. Fischer, Stephan: *Indikatorenkombination zur Inhaltsanalyse digitaler Filme*, D 180 (Diss. Universität Mannheim), 1997, Shaker Verlag Aachen.
7. J. F. Canny, *A computational approach to edge detection*, IEEE Trans. on Pattern analysis and Machine intelligence. vol. PAMI-8, Nov. 1986.
8. Zhu Liu, Jincheng Huang, Yao Wang, Tsuhan Chen, *Audio feature extraction & analysis for scene classification*, IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing
9. Zwicker, E., Fastel, H., *Psychoacoustics*, Springer Verlag, Berlin Heidelberg, 1990