DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071

Maximal stable extremal regions for robust video watermarking

Waldemar Berchtold[1]***, Marcel Schäfer***[2]***, Martin Steinebach***[3]

Fraunhofer-Institute SIT, Darmstadt, Germany

[1] ***waldemar.berchtold@sit.fraunhofer.de,*** [2] ***marcel.schaefer@sit.fraunhofer.de,*** [3] ***martin.steinebach@sit.fraunhofer.de***

Abstract

This paper presents a video watermarking algorithm that is designed to resist the analog gap beside other known attacks. The analog gap, i.e. re-recording e.g. with a camcorder from a screen, poses a huge challenge for digital video watermarking applications. A satisfactory solution is not known yet. In this work we propose a novel transparent and blind video watermarking algorithm that uses so called maximal stable extremal regions (MSER) for identifying regions of the video, in which a watermark is capable to survive many attacks, even the analog gap. Therefore, for embedding as well as detecting, each frame of the video has to be analyzed and stable regions ought to be found. For the embedding, all selected regions are approximated by circles. By means of the orientation of the MSER-Region the preprocessed pattern are adjusted, scaled and added to the content. The MSER itself is amplified to increase the recognition on the detector side. To keep the transparency high the Laplacian matrix is used as psycho visual model as well as the scene detection to reduce the flickering effect.

1 Introduction

Unauthorized video recording and distribution e.g. of movies is claimed to be the cause of significant losses of rights owners. Identification of the source of such unauthorized uploaded video content by means of watermarking methods is not new [8],[17],[5]. However, re-recording video content still poses a huge challenge with respect to robustness. In fact, re-recording – also known as the analog gap – unites a magnitude of different kind of attacks and crucial operations into one attack. These are for example scaling, rotating, desynchronizing, changing the resolution, changing the frame rate, perspective displacement, changing the frames, lens distortions and changing contrast and brightness [1], [12]. Even if solutions exist against some of these attacks/operations, a satisfactory solution against the union is not known [2].

The well known video-watermarking approach by Kalker et al. [8] provides resistance against re-recording, but shows weaknesses against rotation, scaling and de-synchronization. A commercial solutions by Civolution[1] (formerly CineFence by Philips[2]) is said to be applied in the movie industry, however the functionality and its robustness is not publicly available and therefore unknown. Other approaches that might provide resistance against re-recording are *Coded Anti-Piracy CAP* by *Kodak* and a product by *Deluxe Laboratories*. However, these approaches earned much criticism for the corresponding watermarking algorithm regarding its transparency. Moreover CAP is notorious for being susceptible to manipulation.

The present work proposes a video-watermarking approach that promises transparency, robustness and security against the analog gap. We based our work on the MSER algorithm introduced by Matas et al. [9]. The authors state that the regions are invariant against several image transformations and other operations. In their original work the authors used this approach as feature for object recognition. There are several implementations to calculate the MECs, for instance the work by Badoiu-Clarkson [4]. We solve this task relying on the work of Gärtner [6] as it is optimized regarding performance and accuracy.

Our approach is similar to the work in Su et al. [15]. Su et al. propose to use feature points as footprints but they did not state which feature they extract. Further, the generation of the pattern is not described. The work of Brisbane et al. [3] and Nikolaidis and Pitas [10] use segmentation to extract regions of interest out of images and to embed the watermark in the spatial domain into these regions by adding predefined patterns. However, the approaches still lack robustness and transparency.

To that effect, our approach analyzes each frame in order to extract the maximal stable extremal regions (MSER) and approximates them by their corresponding minimal enclosing circle (MEC). These MSERs and the uniformly enlarged MECs (eMECs) are taken as synchronization method between the video and a certain key-depending watermark-pattern, that is generated in a pre-process where the high frequencies are eliminated. Before embedding, for each MSER the watermark-pattern is scaled and rotated such that it fits the corresponding eMEC and so that its orientation is congruent to the orientation of the MSER. Contrary to Su et al. [15], our work does not embed the watermark into the extracted MSERs, instead we add the watermark-pattern to the area inside the eMECs but around the MSERs. Moreover, we add certain steps to amplify the MSER feature.

The paper is structured as follows, section 2 presents the proposed video-watermarking algorithm. The aproach is evaluated in section 3 and a final statement is given in the conclusion section 4

2 Video Watermarking Scheme

In this section we present the video-watermarking algorithm. First the general structure of our scheme is sketched 2.1, afterwards we describe the various steps that prepare for the actual watermark embedding and detecting 2.2. The processes watermark generation, synchronization as well as watermark message embedding and detecting close this section 2.3.

[1] http://www.civolution.com/about-us/audio-video-watermarking-and-fingerprinting/video-watermarking-products/

[2] http://www.business-sites.philips.com/shared/assets/global/Downloadablefile/CineFence-13275.pdf

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071

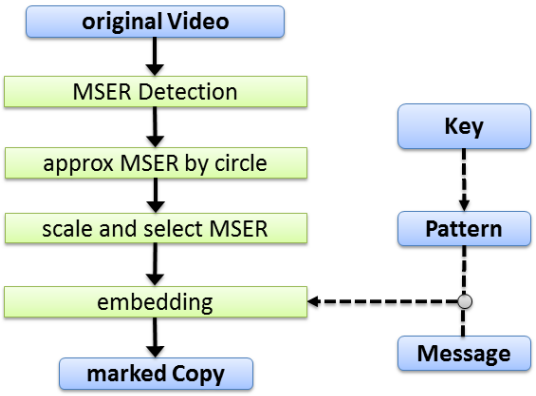

Figure 1: Rough structure of the presented video-watermarking approach

2.1 General Structure of the Scheme

The algorithm is separated into several steps as depicted in figure 1. The video is decoded in order to allow the algorithm to access each single frame. The frames are analyzed separately to determine those regions suitable for embedding. Suitable regions are maximal stable extremal regions (MSER), which are detected applying the algorithm proposed in [9]. MSERs are found to be suitable because they allow detection even after the video has been manipulated. Next, each found MSER is approximated by its smallest possible circle enclosing the corresponding region. Afterwards, the circles are expanded by a fixed factor, the corresponding expanded circles are denoted as *eMEC*s. In case of overlapping eMECs, all of those but the one providing the most stable MSER are deleted. Afterwards each of the resulting MSERs is amplified first by enhancing the pixels around the border of the MSER, and second by additionally adding a certain pattern acting as a high-pass or low pass-filter. For each of the corresponding eMECs we calculate its orientation according to the gradient of the regression line of the corresponding inner MSER. A watermark-pattern is then embedded into the area inside the eMEC but not belonging to the MSER. This area is thus referred to as embedding-area. The orientation is used as synchronization between the embedding-area and the watermark-pattern. In figure 2 these steps are illustrated according to the known *Lena* image (a). First the MSERs are detected (b) and the circles are placed around the corresponding MSER (c). Afterwards the algorithm filters only the circles that are fully inside the image. Of the overlapping circles only those are selected that are the most stable (d).

The watermark-pattern is a circle consisting solely of values -1 or 1 generated according to a secret key. For each embedding-area, the watermark-pattern is scaled accordingly and rotated according to the orientation of the corresponding eMEC and then added to this embedding-area.

The process of adding the watermark-pattern modifies the brightness values of the corresponding embedding-areas according to the watermark message bit that is to be embedded ('0' or '1'). To ensure transparency, we apply a Laplacian high-pass filter as introduced in [8] in combination with a scene detection algorithm [16].

For detection the algorithm extracts the MSERs and approximates these with the corresponding MECs and scales these circles to become eMECs. Overlapping circles are all deleted but the ones that enclose the most stable MSER, in order to follow the structure of the embedding process. Afterwards each potential embedding-area is analyzed. The algorithm determines the correlation between the accordingly scaled and rotated watermark-pattern and the corresponding embedding-area. To do so, we calculate the mean of the brightness values of those positions for which the watermark-pattern has positive entries. The same is done to the those positions of the watermark-pattern that have negative values. Depending on the ratio of the two mean values, the algorithm decides whether a '0' or a '1' has been embedded.

2.2 Preparation Stage

Several steps are required before the actual embedding of the watermark. These steps are explained in the following subsections.

Adjusted MSER detection

MSERs were introduced by Matas et al. [9]. The regions are invariant against continuous transforming of image coordinates or monotonous transforming of an image, such as rotation, scaling, distortion, brightness adjustment. This means the MSERs can be detected again after any of those operations. Matas et al. make use of the properties of the MSERs in order to find the same object in images from different perspectives. However, the invariance of the MSER attracts attention for further applications. They form the base of the watermarking algorithm presented in this work. Finding the MSERs was detailed by Nistér and Stewénius [11]. The term maximal stable extremal region suggests that each MSER is very stable. This is not always the case. The algorithm solely ensures, that the detected regions – hence denoted as MSERs – provide more stability than the neighboring larger or smaller extremal regions close by.

The algorithm to detect the MSERs is provided via the OpenCV[3] library. However, we had to do some modifications to the program code to work as desired. For once, the code had to be adjusted to receive information about the variance as well as information it being a minimal or maximal region. Moreover, the MSER detection needed some modifications such that it actually correlates to the original definition from [9], which actually enabled improved MSER detection rates compared to the publicly available implementation. Finally, the runtime was optimized by modifying the MSER algorithm inside the OpenCV implementation. For instance, in case the algorithm finds an MSER that is completely enclosed by a larger MSER, the variance decides which MSER to take for the further process. Because a smaller variance means more stability for the MSER, only the corresponding MSER will be used any further. This modification decreases the complexity of the process in which an MSER is approximated with a circle and leads to a reduction of the subsequent selection of circles, see section 2.2. It is ensured that in case of overlapping circles only the circle approximating the most stable MSER is selected. This way multiple embedding in some parts of the frame will not occur.

For the extraction of the MSERs the following parameters are chosen:

- $\delta = 4$

[3] OpenCV (Open Source Computer Vision), http://opencv.org

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071

(a) original (b) MSER detection (c) overlapping eMECs

(d) selected eMECs (e) orientation of the eMECs

Figure 2: The different steps of the preparation stage according to the 'Lena' image

- minimal size: 0.1%
- maximum size: 10%
- maximum variance: 0.4
- minimal diversity: 0.5

These parameters prescribe which of the detected MSERs are sorted into the selected set. The parameters are obliged to the following parameters:

- δ: This parameter is used to classify the found extremal region and to verify if it is maximal stable. To this respect may $Q_1, Q_2, ..., Q_n$ an increasing sequence of nested minimal or maximal extremal regions for which holds $Q_1 \subset Q_2 \subset ... \subset Q_n$. To this respect, if holds $q'(i) = 0$ und $q''(i) > 0$ mit $q(i) = \frac{|\frac{Q_{i+\delta}}{Q_{i-\delta}}|}{|Q_i|}$, then Q_i is considered an MSER.
- minimal size: The minimal size marks a threshold. MSERs smaller than this value are not taken for further consideration. The parameter correlates percentaged to the size of the image.
- maximal size: The maximum size also marks a threshold. MSERs exceeding this value are not taken for further consideration. Analogously, the parameter correlates percentaged to the size of the image.
- maximum variance: If the MSER's variance exceeds this parameter value, the MSER is not taken for further consideration. The parameter ensures that only sufficiently stable MSERs are taken.
- minimal diversity: This parameter prevents from taking MSERs into further consideration that are very alike others already taken. Many images contain nested MSERs that only differ by few pixels. Only if two nested MSERs differ by at least the size of the parameter, both are considered for further consideration.

Circle selection

After the extraction of the MSERs, its smallest enclosing circles (MECs) are determined and enlarged by a fixed factor. Afterwards, of overlapping circles those are selected that approximate the corresponding most stable MSER.

To determine the smallest enclosing circles – in literature denoted as *minimal enclosing circle* or *MEC* – two class of algorithms are available. One could possibly calculate the solution with precisely calculating algorithms, or one could determine solutions via heuristic algorithms that may deviate from the correct solution. The latter demand significantly lesser effort than precise algorithms. Depending on the required accuracy in the actual application scenario, one can apply different algorithms. In this work we make use of an implementation by Gärtner and Schönherr [7], that is based on heuristics and thus finds the MECs very fast and sufficiently precise, i.e. in some cases the selected circles deviate from what would actually be the MECs by definition. For this reason, the algorithm investigates two parameters: accuracy, providing the maximal distance of the approximated circle to the correct MSER, and slack, denoting the deviation from it. Gärtner and Schönherr suggest accepting the approximated circle as MEC if $slack = 0$ and $accuracy \leq 10^{-15}$.

After approximating the MSERs with circles, their radii are enlarged. This improves the robustness compared to the original MECs, as it allows a broader utilization of the content. Moreover, the enlarged circles allow to amplify the MSERs in the embedding stage and solely the parts out of the range of the MSER are taken for embedding the watermark. In case of such an amplification of the MSER, the probability to reliably find it again at the

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071


Figure 3: Exemplary pattern used for amplification of the MSERs

detection stage increases. Note that amplification is only possible to a certain level, otherwise the corresponding parts become too large which increases the probability of overlapping parts or even one huge part consisting of the whole frame used for embedding. Empirical examination lead to a suitable amplification factor of 1.4. Hence this factor is taken ever on in this work. Afterwards, all circles that do not lie completely inside the frame are discarded. Besides, in case of groups of overlapping circles, for each group only the one circle is taken that approximates the most stable MSER. The MSER with the smallest variance is the most stable one.

MSER amplification

To enhance robustness, i.e. to increase the probability that the detection process finds the correct MSERs, the *boundary line* of the MSER, that is the *line* of outermost pixels of the MSER, is considered further: The pixels outside the MSER that have at least one adjacent pixel that belongs to the boundary line, are amplified by three brightness values. Further, all pixels adjacent to these *first outer line* of pixels just amplified are themselves amplified by two brightness values. Finally, all pixels adjacent to these *second outer line* of amplified pixels are yet amplified by one brightness value. The inverse is done to the corresponding three inner lines: All pixels inside the MSER adjacent to the boundary, denoted as *first inner line* of the MSER, are amplified by three brightness values in the inverse direction. The *second inner line*, i.e. all pixels adjacent to this *first inner line* from inside the MSER are inversely amplified by two brightness values and the corresponding *third inner line* is inversely amplified by one brightness value. In case of a minimal region the MSER is darker than its environment and hence the amplification is done in the other direction. After these amplification steps, the MSER is further modified by adding a certain pattern to the maximal regions and subtracting this pattern to the minimal regions. To do so the pattern is scaled to the size of the MSER and arranged according to the center point of the circle prior to adding or subtracting it. Noteworthy, the transparency is not degraded by this modification, since the MSERs stand out from their environments. The pattern is depicted in figure 3, it must not be confused with the watermark pattern as described in section 2.3.

Orientation determination

Synchronization between the watermark-pattern and the embedding-area during embedding as well as for detecting the watermark is achieved by the orientation of the eMACS. To determine the orientation, an euclidean regression line is constructed according to the pixels of the corresponding MSER. This is done according to Yaakov Stein in [14]. What is needed for determining the orientation is not the regression line per se, but its gradient. Alike common regression lines, the euclidean regression line is constructed such that the sum of the distances of the points to this line is minimal. Contrary to typical regression line calculations, the euclidean regression line minimizes the orthographic distance of the points to the line. Hence, rotating the points by an angle α, the angle between the corresponding new regression line and the original regression line will be α as well. In figure 4 this property is illustrated. Also one can see that other regression line constructions do not provide this property.

Given that, the embedding-area is parted in two halves along the regression line. The orientation of the embedding-area is then defined as the vector from the center of the eMAC to its boundary that goes through the one of the two halves that contains more pixels of the MSER and that is orthogonal to the regression line.

We illustrate the procedure with the following example: Given an embedding area such that a line with a gradient of $m = 1$ goes right through its MSER. The two possible orientations thus are $\alpha_1 = \arctan(m) = 45^\circ$ and $\alpha_2 = \alpha_1 + 180^\circ = 225^\circ$. We construct a normal with normal vector $n = (1,1)$. Assuming the semi circle with the normal n pointing at it contained more points, then the orientation of the area is equal to the orientation of the normal n, that is α_1. If we rotate the area by 90° counterclockwise, the new orientation should become 135°. After this rotation we have a new gradient of $m^* = -1$. The corresponding possible orientations thus are $\alpha_1^* = \arctan(m^*) = -45^\circ = 315^\circ$ and $\alpha_2^* = \alpha_1^* + 180^\circ = 495^\circ = 135^\circ$. The normal vector that helped dividing the area now changes to $n^* = (1,-1)$. Note that n will not become n^* after a rotation by 90° counterclockwise, but to $n = (-1,1)$. Consequently, n and n^* do not point into the same semi circle and hence the orientation of n^* is not what we desired. The majority of points now lies in the semi circle into which n^* does not point. For this reason the orientation is corrected and the corrected result becomes α_2^*.

2.3 Watermark Embedding and Detection

Circular Watermark Pattern

The algorithm starts generating a circled pattern. It consists of the values '-1' and '1' according to an a priori selected key. On the left side in figure 5 is depicted an exemplary pattern with enhanced values for which the values a projected to a scale of [0,255]. With this original pattern a high security level is hard to achieve, because a very precise synchronization is required to determine the correlation to the pattern, which is not realistic in case of an attack. Therefore larger areas of the same value within the pattern appear to be more appropriate. Hence the algorithm generates several patterns of varying diameters (16, 32, 64, 128 and 256 pixels). The procedure is as follows: First the original pattern is scaled to the corresponding size and transformed to the frequency domain by means of the Fourier Transform. There the high frequencies exceeding a selected threshold τ_m are set to 0. Afterwards the inverse Fourier transform takes them back to

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071

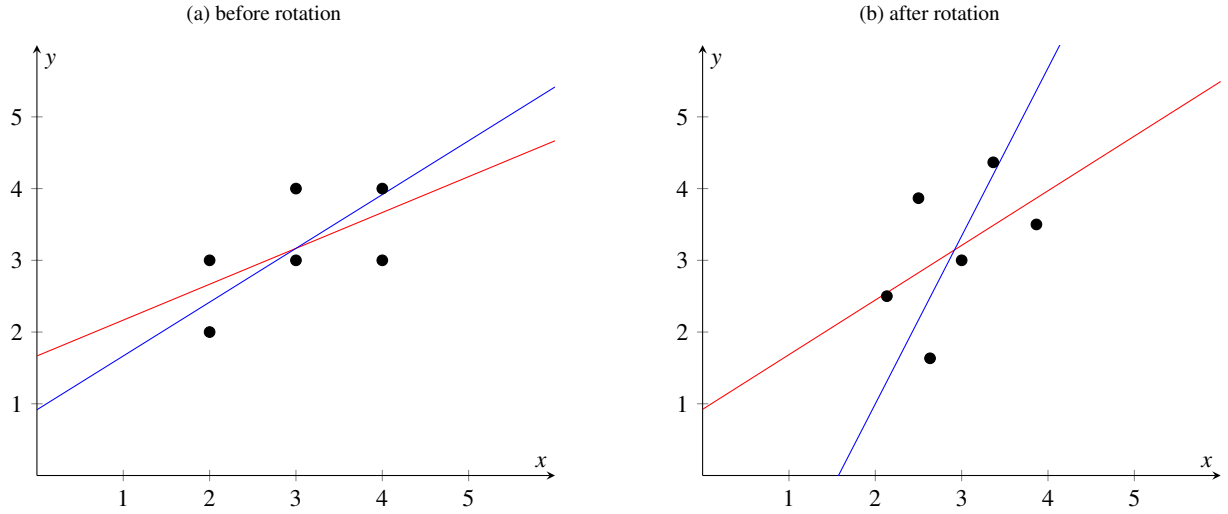

Figure 4: Comparison of euclidean (blue) and classical (red) regression lines

the spacial domain, where all values are projected to '-1' and '1' again. This way the pattern has some high frequency parts again, but its energy is very low thus preventing transparency lose for the concurring algorithm. Figure 5 exemplary shows two patterns with a diameter of 128 and 256. Note that the values have been enhanced and projected to [0,255]. The pattern that fits the circle best (in terms of size) is scaled to the corresponding size. In terms of embedding a '1', the pattern is added to the intensity of the pattern. Embedding a '0' means the inverse pattern is added.

Watermark synchronization

Contrary to other media types for which robust hashes can be used for synchronization purposes, no robust hash for video that is as well robust against re-recording was found. For this reason we apply a synchronization as follows: A fixed series of bits is put in front of the watermark message code. It is correspondingly embedded as the first part of the message. In the detection stage as soon as this fixed series is discovered, the actual message can be detected. Note that fixed message means, fixed for all messages to be embedded.

The bit sequence is arbitrary, as long as the sequence is always embedded to its full length, i.e. if it fits into all corresponding areas. However, especially at the beginning or at the end of videos the might occur scenes only proving a limited number of MSERs. For example front credits often contain plain areas, characters or fast cross-fading content. These parts of the video must not be used for embedding the synchronization sequence. Hence, the algorithm always searches for more appropriate parts to embed the synchronization to its full extend using the described watermark embedding process. In this work we applied the synchronization sequence 1110001110001110000.

Watermark message embedding

Required for the embedding process are the embedding areas of a frame, the pattern and the (binary) message code. Into each frame one bit of the message code is embedded. Depending on the message symbol – '1' or '0' – embedding means adding or subtracting the pattern. Note that this operation is only done to the embedding area, i.e. the interior of the MEC but without the MSER itself. The MSER though is amplified as described in section 2.2. To ensure that the operation of adding or subtracting the pattern, the algorithm makes use of the Laplacian filter matrix as introduced in [8]:

$$L = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \tag{1}$$

To enable the strength of the possible modification, the matrix is only applied to the embedding areas. Note that the original matrix L allows comparably strong modifications. For this reason we use an adjusted version $L* = min(20, 0.05 \cdot L)$. Obviously this results in embedding a weaker pattern, but the modifications inserted by applying L^* remain imperceptible. The so called *flickering effect* appears e.g. if in two consecutive frames of the same scene inverse orientated patterns are embedded at the same position causing a visual perceptible flickering. To prevent inversely embedded patterns, we apply a scene detection according to Trick and Thiemert [16].

Watermark message detection

The detection process is analog to the embedding process. First the MSERs are detected and approximated with the MECs. From these the eMACs are generated. Those eMACs lying over the edges of frames are discarded. In case of a group of overlapping eMACs, only the one that approximates the most stable MSER is taken, the others are discarded. From the resulting eMACs the potential embedding-area is calculated. Besides, the watermark-pattern is generated according to the secret key as described above. With it, for each embedding-area the watermark-pattern has to be scaled accordingly. The correlation between the watermark-pattern and the embedding-area is calculates as follows: Those brightness values of the embedding-area that are associated to a $'1'$ in the watermark-pattern are summarized in a group A. The same is proceed with the brightness values associated to $'-1'$, these are summarized in a group B. For both groups the mean values are calculated and its ratio is logarithmized. This

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071

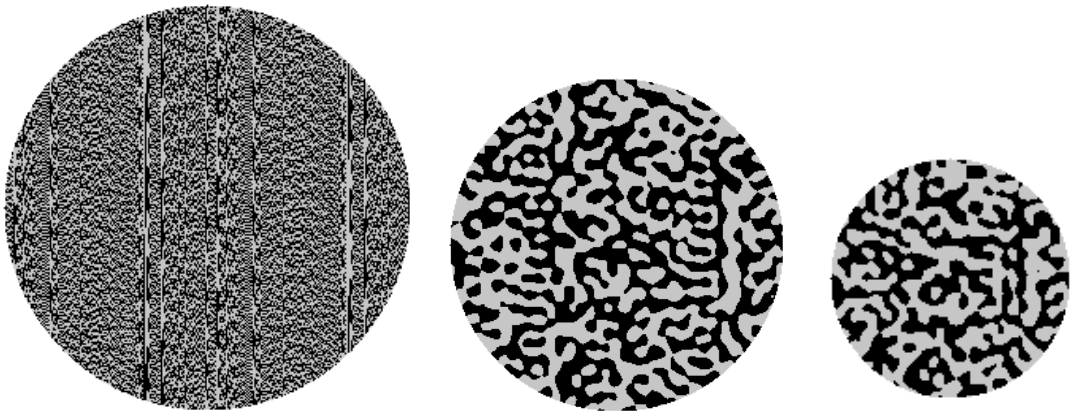

Figure 5: Original pattern (left), pattern with 256 pixels (center) and pattern with 128 pixels (right) as watermark-pattern examples taken for embedding the watermark message into the embedding-areas

way is determined the watermark message bit m_i.

$$m_i = \begin{cases} 1, & \text{if } \log_{10}\frac{A}{B} > \tau \\ 0, & \text{if } \log_{10}\frac{A}{B} < -\tau \\ ?, & \text{otherwise} \end{cases}$$

Prior to the watermark message detection the algorithm needs to find the synchronization sequence by the same means. As soon as this sequence is found, the algorithm starts the detection of the watermark. In case the algorithm finds other MSERs than used for embedding, a threshold τ ensures that no watermark bit is extracted from those MSERs. Analyzing the unmarked content confirmed that the logarithm of the correlation of the groups A and B as described above typically is close to 0. In case this value lies within the interval $[-\tau, \tau]$, the corresponding embedding-area is marked as $'?'$ and is no longer considered by the algorithm.

3 Evaluation

To evaluate the security of the proposed video watermarking algorithm we run several attacks on marked videos of various content. The videos taken for the testing are watermarked with four bits per frame. The testset contains movie trailers, nature sceneries, animations, motion pictures, TV-shows, sitcoms and videos of video games. The trailers consist of much and fast movement, frequent scene changes and plain frames. The nature sceneries provide only marginal movement and few changes. The class of animations contains animated music strips and typical animated videos that provide major plain colored regions. The video games videos contain recordings of different scene of the video game *The Elder Scrolls V: Skyrim*.

All tested videos come with a resolution of 1280×720, 30 frames per second and a bit-rate of 10,000kBit/s. Into each video we embedded both the message 100110010 and its inverse 011001101, in order to eliminate the probability that some symbols are by chance favored for embedding at specific position. The videos are trimmed to 30 seconds. Without the scene detection 12 frames form a sequence for embedding the watermark bits. That means, 3.4 seconds of the video is required for embedding a complete watermark message code. Activating the scene detection the required length for embedding a complete message varies according to the lengths of the scenes. Evaluating the security, we chose a watermark strength factor of 2 for embedding the watermark pattern. This way all brightness values were modified during embedding by -2 or 2.

	post-process/attack	BER
1	detect after embedding	0.89%
2	reduce the bit rate to 2000kBit/s	5.91%
3	reduce the bit rate to 1000kBit/s	24.66%
4	reduce the bit rate to auf 500kBit/s	30.64%
5	linear scaling 640x360	13.35%
6	linear scaling 1920x1080	1.68%
7	non-linear scaling 960x720	4.77%
8	crop to 960x720	3.55%
9	rotate 10	2.71%
10	horizontal mirroring	45.10%
11	convert from raw to h.264	0.87%
12	change frame rate to 25Frames/s	3.35%
13	change frame rate to 20Frames/s	5.75%
14	re-recording by Smartphone Camera (iPhone 4, Samsung Galaxy S4)	27.69%
15	simulate re-recording by CamMark[13]	12.27%

Table 1: Evaluation results of the bit error rate (BER) after typical post-processing operations and watermark attacks

3.1 Robustness evaluation

The robustness of the watermark against common post processing operations and watermark attacks have been evaluated as follows: Re-recording, linear and non-linear scaling, cropping, rotating, mirroring, format conversion, changing the frame rate and changing the bit-rate. We calculated the bit error rate (BER) for both, attacked and not attacked videos. The watermark of the watermarked video and of its attacked version was detected and compared to the watermark that has been embedded before. The results are listed in table 1.

3.2 Transparency evaluation

The SSI metric for videos by [16] is applied to evaluate the transparency of the proposed watermarking algorithm. To this respect five different tests with the whole test-set videos were con-

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071

Factor	2	3	4	5	6
SSIM	0.03	0.05	0.08	0.13	0.16

Table 2: Evaluation results of the video quality after embedding the watermark. We use the SSI metrik by Trick and Thiemert [16]

ducted. For each test we vary the watermark strength parameter. This means each pixel of the watermark pattern is multiplied by a fix factor – we chose 2, 3, 4, 5, and 6 – prior to adding it to the circle during embedding.

Quality evaluation using the SSIM requires both original and marked video. Table 2 shows the corresponding results. Proper video quality after embedding means values close to $'0'$. Larger values imply reduced quality. The existing implementation yield $'1'$ as a maximum value.

3.3 Discussion and assessment of the results

The results show a high detection rate even after various kinds of attacks. Obviously, the more degraded the resulting quality is after an attack, the higher will be the BER. The goal is to correctly detect the watermark as long as the quality of the attacked copy is *acceptable*. The point at which the video quality is no longer acceptable, however, depends on the application scenario and varies with it.

Reducing the bit-rate in the attacks 2 through 4 results in an incremental BER. This is reasoned by the weakened MSER and watermark pattern. Remember, during the watermark pattern generation high frequencies are eliminated in the frequency domain. Afterwards all values are rounded to -1 or 1. This step cause high frequencies in the watermark pattern again. Omit this step could lead to a more robust watermark pattern which mean a lower BER. Further, the weakened MSER imply that not all MSERs in their original form, position and with the correct orientation are found during the detection process. This leads to a deficient correlation between embedding pattern and marked video. A way to strengthen the MSER in order to better survive a bit-rate reduction could be to amplify the border of the MSER in a more extensive area.

Changing the solution is only a minor challenge for the proposed video watermarking algorithm, however a slight increment in the BER is noticeable. This is because of the resulting scaled embedding areas. In the detection process, orientation and size of the found regions might be deviating from those taken for the embedding process, which causes the errors. A solution to this could be to rely on a smaller threshold τ_m for the pattern generation. To this respect the pattern values should not be set to -1 and 1 again.

The attacks in 7, 8 and 9 have no significant impact on the BER, implying that the watermark is secure against these attacks. In some cases the embedded pattern is heavily weakened, for instance by cropping parts of the frame that contain the pattern, but due to redundancy in the embedding, the BER remains low.

Horizontal mirroring is not easily recognizable for the proposed watermarking algorithm. The human eye obviously detects the mirroring and after reversing it, detection of the watermark is again possible. For the watermark to be detected automatically in spite of mirroring attacks, there are two possibilities. First, the pattern correlation calculation during the detection process can also be done with the mirrored watermark pattern. Second, the watermark pattern could be designed rotation invariant. The second solution seems to be the more efficient way.

As the proposed watermarking algorithm was developed independent of a certain codec, it stays invariant against format conversions. The corresponding BER is comparable to the BER after no attack at all. In case the conversion induces a reduction in the bit-rate, an increment in the BER – as described for the attacks 2 through 4 – is expected.

Changing the frame rate poses only a minor challenge for the proposed scheme. Because four bits are embedded in a fixed sequence, the frame rate has no significant impact for the detection success. An increment of the BER is reasoned by the cross-fading of the frames, which weakens the MSERs and the embedded patterns. Hence, some MSERs might not be detected in their original form or orientation or the correlation between embedded watermark pattern and the detected region in the frames is decreased.

The resulting BERs after re-recording with smartphone cameras and CamMark [13] confirm the security of the approach against these attacks. The results with CamMark lead to a smaller BER, which we reason by the higher solution and quality of the videos that were attacked by CamMark. CamMark simulates the re-recording from a TV-screen by a camcorder. The recordings done by smartphones from a 24 inch screen with a frequency of 80 hertz reveal a slight but perceptible flickering as well as added noise on the whole video. This leads to a significantly increased BER compared to the re-recorded videos via CamMark. Improvement could be effected with the actions proposed for the attacks 2 through 6.

The BER can be reduced by adding more redundancy. The actual implementation of the scheme is not real-time capable, but it can be accelerated up via parallelization. We achieve good results regarding transparency due to values close to $'0'$, see table 2. The SSIM results show that the patterns used for embedding are not disturbing, only very few at a factor of 6 can be recognized.

Comparing the results to the state of the art one notices not only good results regarding robustness against re-recording, but as well against rotating, non-linear scaling and cropping.

4 Conclusion

In this work we present a video watermarking algorithm designed with a focus on robustness especially against video re-recording. Re-recording poses a huge challenge for video watermarking as it is often applied but hard to resist.

The proposed approach is based upon the maximal stable extremal regions (MSER) algorithm[9], that extracts MSERs from every frame. To each of the MSERs the minimal enclosing circle (MEC), i.e. the smallest circle completely surrounding the corresponding MSER, is calculated and afterwards enlarged by a fixed factor. In case of overlapping circles only those are taken, that approximate the most stable MSER. In the embedding process, for each of the remaining circles a pre-generated pattern is scaled to the corresponding circle. Depending on the symbol to be embedded this scaled pattern or its inverse is added to the brightness values of the embedding area, that is all pixels within the enlarged circle but not belonging to the MSER itself. Moreover, the MSERs are enhanced in order to increase the probability to correctly detect them even after strong attacks. The detection process calculates the correlation between the pattern used for embedding and the corresponding part of the frame. This correlation determines which watermark symbol is detected. To ensure the visual

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-071

quality is not degraded, the algorithm uses a Laplacian filter [8] and a scene detection algorithm [16].

The results from the evaluation confirm a good detection rate of the watermark after various post-processing operations and attacks. For the attacks that resulted in a bit error rate (BER) exceeding 10% we discussed measures for optimization. We admit that we did not compare against approaches commercially applied that claim to be resistant against re-recording. The majority of those approaches however is not known to provide satisfying results regarding transparency. Compared to the state of the art our video watermarking scheme is not only resistant against re-recording, but also against common attacks such as rotating, non-linear scaling and cropping.

For future work different embedding patterns could be considered as well. Alternatives are mentioned in the corresponding paragraphs regarding discussion and assessment of the approach. In addition one could rely on rectangles for the approximation of the MSERs. To enhance speed and provide real-time capabilities, most processes could be parallelized. The evaluation of transparency should be reenacted via ABX tests, as the applied metric has not been considered to that effect.

ACKNOWLEDGEMENT

420 This work was supported by the *CASED Center for Advanced Security Research Darmstadt*, Germany (*http://www.cased.de*), funded by the German state government from Hesse under the *LOEWE* programme.

References

[1] C. Ben Zid, S. Baudry, B. Chupeau, and G. Dorr. A sneak peek into the camcorder path. *Proc. SPIE*, 8665:86650E–86650E–10, 2013.

[2] S. Bhattacharya, T. Chattopadhyay, and A. Pal. A survey on different video watermarking techniques and comparative analysis with reference to h.264/avc. In *Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on*, pages 1–6, 2006.

[3] G. Brisbane, R. Safavi-Naini, and P. Ogunbona. Region-based watermarking for images. In *Proceedings of the Second International Workshop on Information Security*, ISW '99, pages 154–166, London, UK, UK, 1999. Springer-Verlag.

[4] M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. *Comput. Geom. Theory Appl.*, 40(1):14–22, May 2008.

[5] F. Deguillaume, G. Csurka, J. J. O'Ruanaidh, and T. Pun. Robust 3d dft video watermarking. volume 3657, pages 113–124, 1999.

[6] B. Gärtner. Fast and robust smallest enclosing balls. In *Proceedings of the 7th Annual European Symposium on Algorithms*, ESA '99, pages 325–338, London, UK, UK, 1999. Springer-Verlag.

[7] B. Grtner and S. Schnherr. Smallest enclosing ellipses – fast and exact, 1997.

[8] T. Kalker, G. Depovere, J. Haitsma, and M. Maes. A video watermarking system for broadcast monitoring. In *Proceedings of IS&T/SPIE/EI25, Security and Watermarking of Multimedia Content*, pages 103–112, 1999.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, pages 36.1–36.10, 2002. doi:10.5244/C.16.36.

[10] A. Nikolaidis and I. Pitas. Region-based image watermarking. *Trans. Img. Proc.*, 10(11):1726–1740, Nov. 2001.

[11] D. Nistér and H. Stewénius. Linear time maximally stable extremal regions. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 183–196, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] X. Rolland-Nevire, B. Chupeau, G. Dorr, and L. Blond. Forensic characterization of camcorded movies: digital cinema vs. celluloid film prints. *Proc. SPIE*, 8303:83030R–83030R–11, 2012.

[13] P. Schaber, S. Kopf, C. Wesch, and W. Effelsberg. Camark: a camcorder copy simulation as watermarking benchmark for digital video. In *Multimedia Systems Conference 2014, MMSys '14, Singapore, March 19-21, 2014*, pages 91–102, 2014.

[14] Y. Stein. Two dimensional euclidean regression. In *Conference on Computer Mapping (Herzelia, Israel)*, 1983.

[15] K. Su, D. Kundur, and D. Hatzinakos. A novel approach to collusion-resistant video watermarking. In *Proceedings of the SPIE Security and Watermarking of Multimedia Contents IV*, pages 491–502.

[16] D. Trick and S. Thiemert. A new metric for measuring the visual quality of video watermarks. *Proc. SPIE*, 7880:78800D–78800D–13, 2011.

[17] R. Wolfgang, C. Podilchuk, and E. Delp. Perceptual watermarks for digital images and video. *Proceedings of the IEEE*, 87(7):1108–1126, Jul 1999.