

# Soft Feature-Based Watermark Decoding with Insertion/Deletion Correction

Mathias Schlaueg, Dima Proefrock, and Erika Müller

Institute of Communications Engineering,  
Faculty of Computer Science and Electrical Engineering,  
University of Rostock,  
Rostock 18119, Germany,  
{mathias.schlaueg, dima.proefrock, erika.mueller}@uni-rostock.de

**Abstract.** Hard decisions constitute the major problem in digital watermarking applications, especially when content adaptive embedding methods are used. Soft-decision decoding, on the other hand, has proved to be performance gaining, often realized during forward error correction. But despite this insight, no adaptive watermarking approach integrates the adaptation process into soft decoding, up to now. Further, insertion/deletion errors can occur in a content dependent watermarking system due to hard decisions if data is embedded only in some selected regions of the host signal. This kind of error usually desynchronizes the decoder and disables the correct watermark extraction. In this paper, we work out three fundamental properties of content dependent quantization-based watermarking. We show how the coupling between these properties and common soft-decision forward error correction decoding can be used to build up an overall soft processing watermarking. No pre-distortion has to be used, and hence, the image quality is not degraded. Even adaptation techniques can be used where it is computational infeasible to project a pre-distortion back onto the host image. Afterwards, we describe how to modify a common Viterbi decoder to enable the correction of insertion/deletion errors combined with our new soft decoding approach and hence improve the overall performance.

**Key words:** Digital watermarking, image segmentation, adaptive decoding, soft-decision forward error correction, re-synchronization.

## 1 Introduction

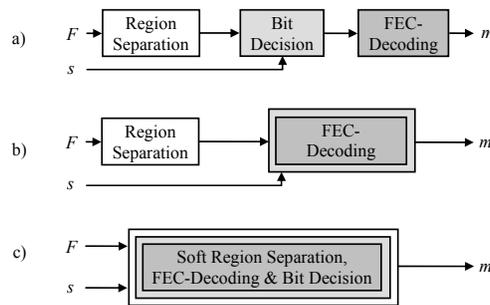
After several years of watermarking research, more and more sophisticated methods have been developed to embed additional data into multimedia content. All these methods have in common that they strive after a good compromise between invisibility and robustness of the embedding. In image watermarking, invisibility means that degradations of the perceptual quality of the host image must be avoided. Hence, newer content adaptive watermarking methods take advantage of the property that the HVS (human visual system) is less sensitive to changes

in textured regions than in smooth regions of an image [1]. Data can be embedded with higher robustness against watermark attacks in stronger textured regions without being visible, commonly known as *perceptual shaping* [2].

Basically, there are two kinds of adaptive watermarking outlined in literature. The first embeds data into all regions. Less as well as more perceptually significant areas are used, whereas the embedding strength is adapted to the image content, as in [1], [2]. The second approach is based on the idea only embedding in regions where the perceivable distortion is low and leaving perceptually sensitive areas undisturbed [3], [4], [6]. In both cases, the image has to be separated into regions with different HVS properties, and during the extraction process, the same separation has to be determined from the host signal. Discrepancies yield errors, even if no attack has been applied [5]. While the former approach has to deal with common substitution errors (binary:  $0 \rightarrow 1$  or  $1 \rightarrow 0$ ), the later would result in catastrophic de-synchronization. That means a deletion error occurs when an embedded symbol is not detected by the receiver and an insertion error occurs when the receiver detects a symbol that was not transmitted.

The most often applied technique to circumvent discrepancies between both separation feature maps is to form a gap around the separation threshold. In other words, the used feature is pre-distorted to leave a margin. As a consequence the image quality is degraded. Furthermore, there are separation approaches where it is computational infeasible or even impossible to project the pre-distortion back onto the host image. For example, the separation feature could have been calculated using *higher order statistics* from extensive data. In this case, errors are ignored or tried to be corrected by FEC (forward error correction), usually with moderate success, because common forward error correction schemes are designed to correct substitutions. They cannot detect or correct insertions/deletions. In such systems, synchronization loss will result in a sequence or burst of errors until synchronization is re-established.

In this paper, we propose to integrate commonly used hard region separation into an overall soft processing framework, as in Fig. 1 c). Three interesting key properties deviated from content dependent quantization-based watermarking

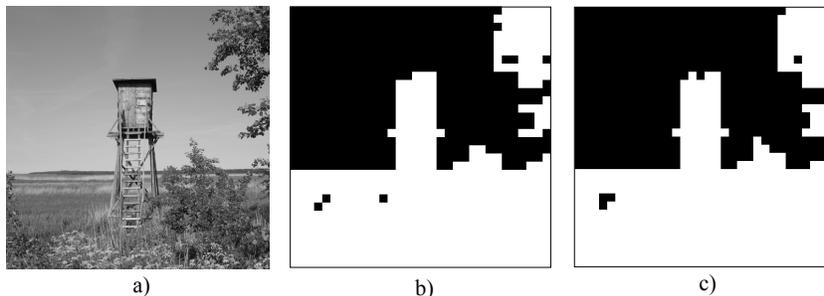


**Fig. 1.** Overall hard processing a), hard region separation with soft bit decoding b), overall soft processing c).  $F$  = separation feature,  $s$  = host signal,  $m$  = watermark.

are presented in section 2. These properties can be used to combine HVS-based region separation and watermark decoding with impressive improvements in error correction capability. Afterwards in section 3, we give an overview of recent insertion/deletion/substitution error correction approaches and present a new solution to the problem of bit re-synchronization in adaptive watermarking. Experimental results for the combination of both new techniques are shown in section 4 and conclusions will be given in section 5.

## 2 Soft-Decoded Adaptive Watermarking

As mentioned earlier, if there are discrepancies of the region separation during adaptively embedding and extracting the watermark  $m$ , as in Fig. 2 b) and c), then wrong parameters are used and errors can occur. If both less as well as more perceptually significant areas are used, for example, with different embedding strengths, “only” single bit errors occur. In the case of bit sensitive, signature-based applications such as image authentication [1] these bit errors are problematic but can be corrected using common forward error correction. But if data are only embedded in regions where the perceivable distortion is low, long bursts of errors occur until re-synchronization. For example, the nine mask differences in Fig. 2 c) would result in three insertions and six deletions anywhere in the extracted watermark stream if the decoder assumes that data is only embedded in the white blocks.

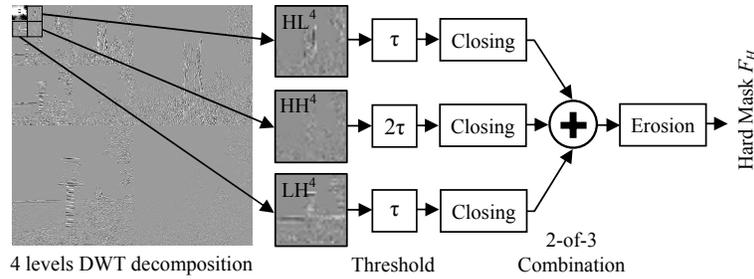


**Fig. 2.** Original image a). Hard embedding mask b), hard extraction mask after JPEG compression with quality factor  $QF = 50$  c) using the DWT-based segmentation ( $\tau=2$ ).

Thus, the first goal must be to find a segmentation scheme that yields less decision errors. Furthermore, we require the segmentation feature map  $F$  to consist of real values,  $F \in \mathbb{R}$ . These soft values, the soft mask, will be used during our new soft-decision decoding for weighting the input signal  $s$ . In section 2.1 and 2.2, we propose two separation approaches that fulfill these requirements. Afterwards in 2.3, three interesting properties of content adaptive quantization-based watermarking are presented. We show how these properties can be combined and integrated into a soft-decision decoding in section 2.4.

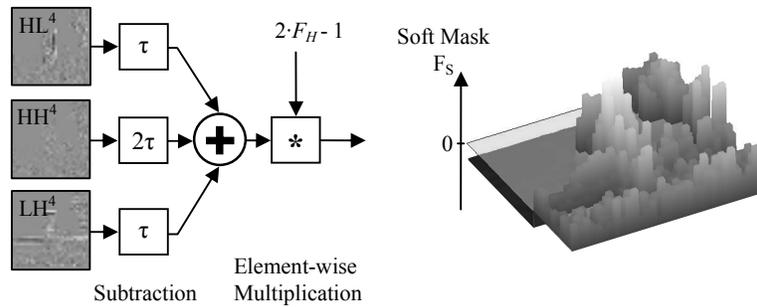
### 2.1 DWT-based Soft Mask Generation

In [1], we applied texture segmentation in the DWT (discrete wavelet transform) domain for an image authentication scheme. Large DWT-coefficients indicate image positions with strong texture, whereas small ones stand for homogenous regions. As in Fig. 3 except for the  $LL^4$ -subband, all coefficients of the fourth decomposition level are compared to a threshold. Afterwards, the known morphologic operations *closing* and *erosion* are used to eliminate small gaps and to refine the separation. The binary masks are combined logically and used during embedding as well as extraction to select between two different watermarking strengths  $\Delta_1$  and  $\Delta_2$ .



**Fig. 3.** Hard mask generation for adaptive embedding as described in [1].

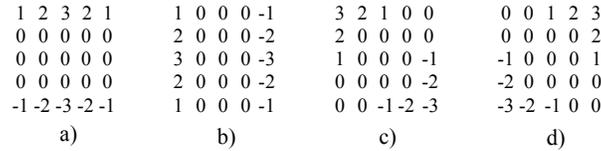
But the described DWT segmentation approach was originally designed for a hard region separation and hard FEC-decoding. Now, in this paper we modify it to a soft version, whereas the texture threshold  $\tau$  is subtracted rather than being used for binarization. As shown in Fig. 4, the resulting three matrices are summed and multiplied element-wise by the signed hard mask values  $F_H$ , which are calculated as described above. Compared to the original image the homogenous sky yields negative values, whereas for stronger textured regions the feature  $F_S$  is positive.



**Fig. 4.** Soft mask generation based on DWT coefficient amplitude.

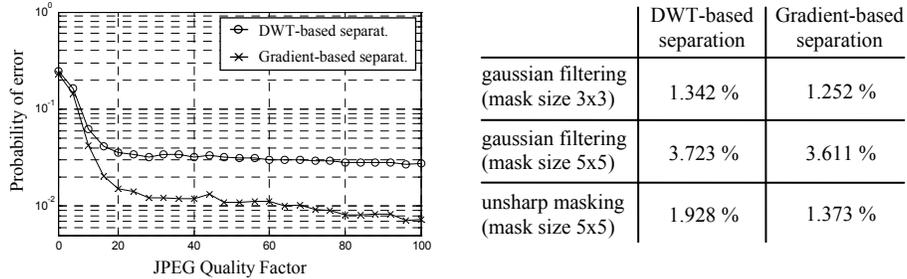
**2.2 Gradient-Based Soft Mask Generation**

For the proposed DWT segmentation approach it would be easy to pre-distort the feature to leave a margin to lower the number of segmentation errors at the decoder. But for another approach that turns out to be even more robust to signal processing a feature pre-distortion is computational more complex. In this second, gradient-based segmentation we propose to detect horizontal, vertical and diagonal edges using the *Sobel* operator. That means the image is filtered using the Sobel filter matrices from Fig. 5 at first. Afterwards, a Gaussian low pass filtering is applied and the four feature maps are compared to a threshold and summed as in the DWT-based approach. Again, morphologic operations are used for the binary hard mask to refine the segmentation, whereupon the filtered edge masks are multiplied element-wise by the hard mask similar to Fig. 4.



**Fig. 5.** Sobel filter matrices for horizontal a), vertical b), diagonal c-d) edge detection.

The following figure indicates the robustness of both described segmentation methods determined from several tests on natural images. By the term robustness here the probability of occurrence of a feature threshold transition is meant, which would yield additional errors to be corrected by the FEC-decoder.



**Fig. 6.** Probability of occurrence of a feature threshold transition during region segmentation at the receiver site due to non-malicious signal processing.

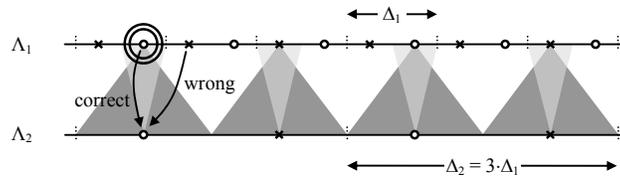
**2.3 Three Key Properties towards Soft Decoding**

For watermark embedding we use QIM (*quantization index modulation*) in the DWT domain as it is described in [1]. But for the extraction site, we found

three new interesting properties to be used for soft-decoding during nearest-neighbor quantization. Thereby,  $\Lambda_1$  and  $\Lambda_2$  denote the two quantization lattices that each consist of sub-lattices marked with  $\times$  and  $\circ$  accordingly. Although here we describe the embedding for the DWT domain, other domains, e.g., DCT (discrete cosine transform) or pixel domain could be used as well.

Firstly, we assume that data is embedded in both less as well as more textured regions. Later on, in section 3, we consider the case that data is embedded only in strong textured regions but not in homogenous parts of the image. We show that this so-called selective watermarking is a special case of adaptive watermarking where the embedding strength  $\Delta_1$  is zero at the encoder.

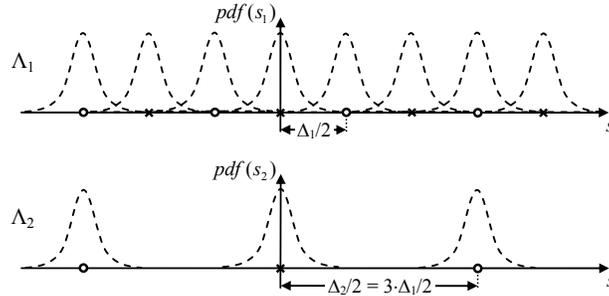
**Property I - Lattice Point Coverage.** In Fig. 7, the “natural” covers of  $\Lambda_1$  and  $\Lambda_2$  are shown for the case of lattice crossings due to false feature separation. The shaded areas indicate “natural positive crossings”. For example, suppose  $\Lambda_1$  has been used during watermark embedding and the host signal was quantized to the highlighted point  $\circ$ . If afterwards the mask changes slightly as in Fig. 2 c), then  $\Lambda_2$  would be used during extraction. In this case the nearest-neighbor quantization would yield the correct bit decision as long as the sample keeps inside the shaded area, even if the separation feature has passed over the decision threshold. But if the point  $\times$  right beside it has been used, it would be falsely decided to a point  $\circ$  in lattice  $\Lambda_2$ . In this case, a bit substitution error would occur.



**Fig. 7.** Covers of two different QIM lattices ( $\Delta_2 = 3 \cdot \Delta_1$ ).

However, the possibility is highest that no error occurs during a transition  $\Lambda_1 \leftrightarrow \Lambda_2$  in the case of an exactly three times larger quantization interval  $\Delta_2$ . Thereby, in at least 50 percent of all cases there is no problem if a transition occurs if the distortion to the quantized signal is less than  $\Delta_1/2$ .

**Property II - Distortion Probability Function.** In [11], Vila-Forcén *et al.* studied additive attacks against quantization-based watermarking. For attacks such as lossy compression, noise adding or filtering the distortion to the quantized signal can be expected to be Gaussian distributed. Since the variance of this distribution is the same for both lattices  $\Lambda_1$  and  $\Lambda_2$ , following distortion probability density functions  $pdf(s_i)$  can be expected (see Fig. 8). Due to the nearest-neighbor quantization  $s_i = \{-1 \leq s_i \leq +1 : s_i \in \mathbb{R}\}, \forall i \in \{1, 2\}$  for one periodical quantization bin, if we consider  $s_1 = s/\Delta_1$  and  $s_2 = s/\Delta_2$ .



**Fig. 8.** Probability density functions of the disturbed signal ( $\Delta_2/\Delta_1 = 3$ ).

Both probability density functions are drawn one upon the other in Fig. 8 to visualize that there are spaces at the second lattice where it is unlikely that a signal sample is located. In other words, if the feature is close to the decision threshold and the signal sample is somewhere in the space where  $pdf(s_2)$  is small, it is more likely that the sample was originally embedded using lattice  $\Lambda_1$ .

**Property III - Certainty of Decision.** We define  $C$  as certainty of how close the input feature  $F$  is to the selected feature threshold  $\tau$ . That means if the feature is close to the decision threshold, it is uncertain which QIM lattice has to be used. In this case the certainty is zero. If the feature is far from the threshold and it is obvious which lattice was chosen during embedding, the certainty is high. Using this certainty we propose two weighting functions,  $f_1$  and  $f_2$ , for the input signals in  $\Lambda_1$  and  $\Lambda_2$ .

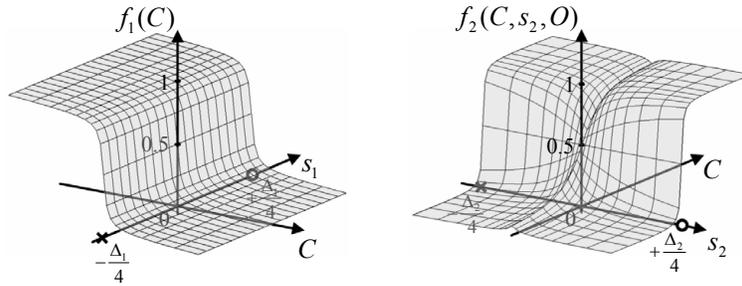
$$C = F - \tau \quad (1)$$

$$f_1(C) = \frac{1}{2} - \frac{\arctan(C)}{\pi} \quad (2)$$

$$f_2(C, s_2, O) = 1 - f_1(C \cdot (|s_2| + O)) \quad (3)$$

$C_x$	$f_1(C)$	$f_2(C, s_2, O)$
0	0,5	0,5
$-\infty$	1	0
$+\infty$	0	1

The weighting function  $f_2$  for lattice  $\Lambda_2$  depends on the absolute value of input signal  $s_2$  to implement Property II. The value  $O$  should be set to the ratio  $\Delta_1/\Delta_2$ , e.g.,  $O = 1/3$ . For one periodical quantization bin the weighting functions are visualized in Fig. 9. Considering the input signal both functions are drawn opposed to each other.



**Fig. 9.** Soft feature input signal weighting functions.

Since data embedded using the weaker embedding strength  $\Delta_1$  is less robust to attacks or common signal processing than data embedded using  $\Delta_2$  afterwards we stretch the signal  $s_w$  by the factor  $\Delta_2/\Delta_1$  at all mask positions where the separation feature is larger than the threshold  $\tau$ .

$$s_w = \frac{s_1 \cdot f_1(C) + s_2 \cdot f_2(C, s_2, O)}{2} \quad (4)$$

$$s_w^* = \begin{cases} s_w & C < 0 \\ \Delta_2/\Delta_1 \cdot s_w & C \geq 0 \end{cases} \quad (5)$$

#### 2.4 Soft-Decision Decoding of the Weighted Signal

After the weighing process, the FEC-decoder receives a stream containing information about the reliability of each received symbol. For example, the common *Viterbi decoder* with soft-decision input used to decode *convolutional codes* can be employed to determine the embedded message  $m$ . By observing a sequence of symbols this decoder determines multiple paths of message states in a kind of *state machine*. Each of these states is valuated by a path metric, representing the reliability of each bit-decision. Afterwards, in a trace back process the maximum likelihood path survives. The weighting approach directly influences this metric.

The improvement of the described input signal weighting approach concerning substitution error correction performance will be presented in section 4. Prior to this in the following section, we discuss the case that data is embedded exclusively in stronger textured regions, whereas perceptually sensitive areas are left undisturbed. This embedding can yield insertions as well as deletions at unpredictable locations within the watermarking sequence at the decoder.

### 3 Insertion/Deletion Error Correction

The term watermark de-synchronization is used twice in literature. The term sometimes often refers to the whole embedded message in consequence of geometric attacks such as shifting, rotation, or scaling. But it may also refer to a non-linear drift within the message sequence. This second kind of de-synchronization

caused by bit insertions and deletions is the major problem in content adaptive watermarking. Known approaches such as ESD (exhaustive search detection) or TMD (template matching detection), recently analyzed by Barni [7], usually applied to re-synchronize the whole message are computational infeasible in the case of IDS (insertions/deletions/substitutions).

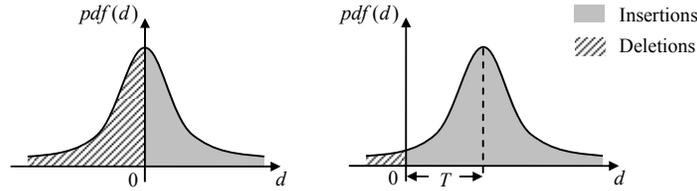
Most error correction schemes are designed to correct substitution errors. It is assumed that the decoder knows the block boundaries in the case of *block codes* or the message length in the case of *convolutional codes*. They can neither detect nor correct insertions/deletions. In such systems, synchronization loss will result in a sequence or burst of errors until synchronization is re-established.

Basically, there are two categories of re-establishing synchronization. While the first detects synchronization loss and discards incorrect code words to regain synchronization, the second is able to correct insertions/deletions and hence to recover the transmitted data. In [5] and [10], the authors give an overview of recent approaches to channel re-synchronization. One technique is, for example, to use an inner code to infer the positions of insertion/deletion errors and an outer code to correct errors in a *concatenated coding* scheme, as in [4], [5]. Thereby, carrier signals (marker) are provided for the outer burst-error-correcting decoder. If these markers are not in their expected positions, the outer decoder can infer the insertion/deletion positions. In this case, exhaustive search detection can be used for the inner code partition with reduced computational effort. Another technique, published by Solanki *et al.*, treats the host signal locations where no data is embedded as erasures [6]. Similar to the idea of *punctured channel coding*, the encoder simply drops the code symbols at these locations. Afterwards, the decoder inserts don't-care states for the assumed locations knowing only the selection criterion but not the explicit locations. Insertions now become errors, and deletions become additional don't-care states that can be corrected by the decoder. Both these solutions require extra coding effort at the sender. In contrast, a third technique that works only at the receiver is the use of extended *dynamic programming* during FEC-decoding. Here, existing codes and standard encoding methods are used. Only the decoder is slightly modified, as in [8] - [10].

### 3.1 Interconnected Viterbi Decoders

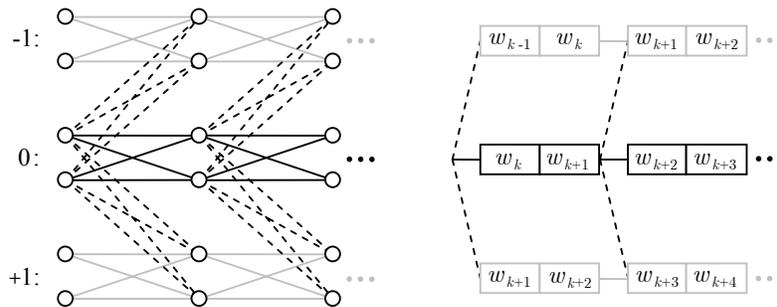
The first extended dynamic programming approach for the purpose of IDS-error correction in a watermarking system with selective embedding was proposed by Mansour and Tewfik [8]. Their technique recovers the correct message if extra bits (*false alarms*) are added to the body of the message at random locations. They modify the Viterbi algorithm used to decode the embedded watermark that was previously encoded with convolutional codes. The decoder is designed to handle substitution as well as insertion errors, but no deletions. To get only substitutions and insertions the threshold  $\tau$  for selecting embedding positions at the decoder is chosen to be less than the threshold at the embedding site. The new threshold is  $\tau - T$ . Hence, the probability of occurrence of deletion errors is lower than for insertion errors (see Fig. 10). But in this case all embedding positions with a feature value in the distance  $d \leq T$  to the threshold unavoidably

turn out to be insertions. As a consequence the FEC-decoder has to deal with an extra error rate that degrades the overall correction capability.



**Fig. 10.** Probability density functions of the occurrence of threshold transitions (zero-crossings) and hence insertion/deletion errors at the decoder (left). If the threshold at the decoder is chosen to be less than the threshold at the embedding, most deletions turn out to be insertions (right).

Recently, another approach was described by Swart *et al.* [10]. Multiple parallel Viterbi decoders are used to correct IDS-errors. Each decoder receives a stream containing information about the reliability of each received symbol. By observing a sequence of symbols the common decoder determines several paths of message states in a kind of state machine. Each of these states is valued by a metric, representing the reliability of each bit-decision. Afterwards, in a trace back process the maximum likelihood path survives. Synchronization errors result in all metrics to have high rate of change for the decoders that are out of sync. Since each of the parallel-interconnected Viterbi decoders is one bit out of sync with the others (see Fig. 11), by monitoring the rate of change for the accumulated error metrics, one is able to ascertain which of the Viterbi decoders is in synchronization. Based on the idea in [9], where several standalone decoders were used, Swart *et al.* integrated the framework into one larger Viterbi decoder and called it super trellis decoder.



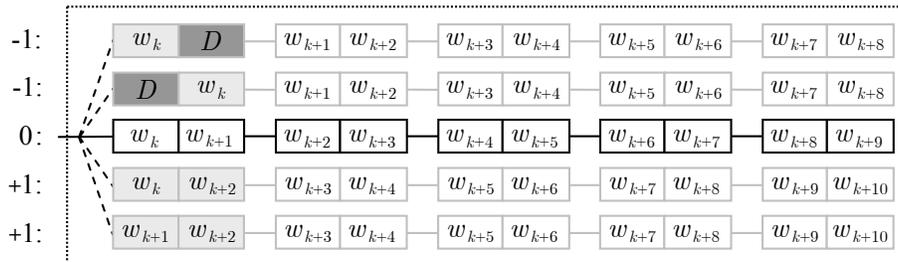
**Fig. 11.** Representation of the super trellis [10] (left). Another illustration of the super trellis to better visualize the indices of the input samples, for an 1/2-rate code (right). The sub decoder lagging by one bit is denoted by -1, the sub decoder leading by one bit is denoted by +1.

We implemented the proposed algorithm and made some modifications on the super trellis decoder. These modifications allow us to directly input the previously described soft-weighted samples  $s_w$  to the decoder stage.

A further input to the decoder must be the feature decision certainty  $C$  to infer the subset of  $s_w$  that is assumed to be the embedded watermark  $w_k$  at the host signal positions  $p_k$  prior re-synchronization. That means if  $N$  is the number of all possible candidates of embedding locations and  $K$  is the number of, e.g., coefficients for data embedding selected using the local perceptual criterion  $C$ ,  $w_k := s_w(p_k)$ , whereas  $p_k = \{p_k \in \mathbb{N} : 0 \leq C(p_k), 1 \leq k \leq K\}$ .

### Modifications:

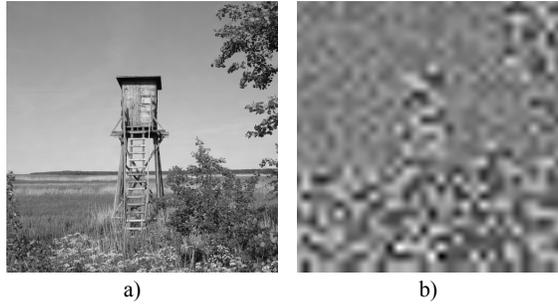
1. Deletions can only have been occurred at those positions where  $p_k < p_{k+1} - 1$ . This reduces the computational effort as well as the number of *false positives*.
2. Swartz decoder simply repeats the previous sample, e.g.,  $w_{k+1}$  in Fig. 11, if a deletion is assumed for a symbol in the range  $p_k \dots p_{k+1}$ . But since the decoder has knowledge about all samples, we propose to integrate the sample delivered by the watermark extractor that is assumed to be deleted. In the example in Fig. 12, such a deleted sample is labeled  $D$ . If  $p_k < p_{k+1} - n$ , where  $2 \leq n$ , or in other words if there are more than one deletion candidate, we suggest from all  $n$  samples  $D(p_k + 1) \dots D(p_k + n)$  to chose the one with the lowest absolute value  $|C|$ .
3. Considering Fig. 11, the old super trellis decoder is not able to deduce the actual insertion/deletion position within a pair of symbols, e.g., the tuple  $[w_k, w_{k+1}]$  in the case of a 1/2-rate code. As can be seen in Fig. 12, our trellis extension checks for all possible combinations of insertion/deletion positions and hence, finds as well as corrects the actual error location.
4. Additionally, we apply a weighting to the first pair of samples within the window for the decoders out of sync using the certainty of the sample positions assumed to be inserted or deleted, respectively.



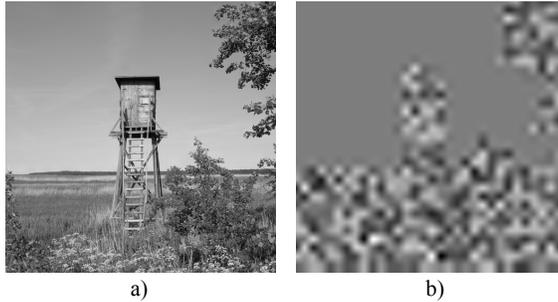
**Fig. 12.** Modified super trellis for a 1/2-rate code, where 2 input samples form a pair of symbols. Here, the decoder considers the input within a window of length 5.

## 4 Experimental Results

The probability of mask transitions using the described soft region segmentation is very low, as shown in Fig. 6. Thus, we had to perform a huge number of simulations to assess the improvement in bit error correction during watermark retrieval. To get an impression of how the image from Fig. 2 a) is changed due to the embedding process, in Fig. 13 and Fig. 14 the watermarked image is displayed. In the first image, less as well as more perceptually significant regions are used for data embedding. Here, only bit substitution errors have to be expected if the segmentation mask changes during watermark extraction. On the other hand, in Fig. 14, the host image is only watermarked in regions where the separation feature is larger than the threshold. That means for this approach also insertion/deletion errors have to be expected. As already mentioned, we call the second approach selective embedding and claim that selective embedding is a special form of adaptive embedding, where  $\Delta_1$  is zero.



**Fig. 13.** Adaptively watermarked image a). 1024 bits are embedded in less as well as more perceptually significant regions,  $\Delta_2 = 3 \cdot \Delta_1$ , where  $\Delta_1 = 2$ , PSNR = 45.91 dB. Difference to original b).

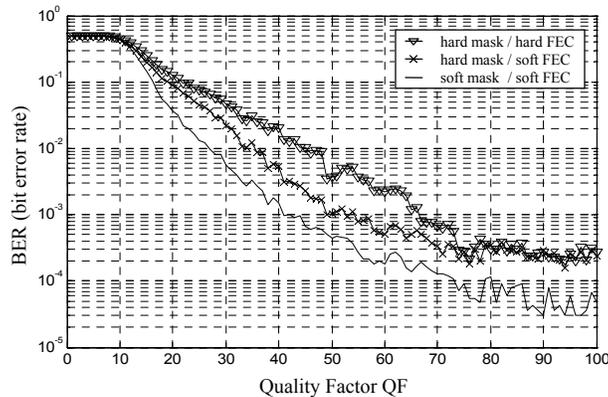


**Fig. 14.** Selectively watermarked image a). 541 bits are only embedded ( $\Delta_2 = 6$ ) in regions where the separation feature is larger than the threshold, PSNR = 46.32 dB. Difference to original b).

All results shown in this paper have been calculated for the same parameters for a set of 32 natural photos. We show the results of our new solution for JPEG compression. Other watermark attacks such as filtering or noise-adding have similar effect on the probability of mask transitions and hence, functionality of our approach. In this paper, malicious attacks such as *cropping* or *geometric attacks* are not considered since the original intention was to propose an extension for the image authentication watermarking system in [1].

#### 4.1 Adaptive Watermarking

Fig. 15 shows that our soft-mask/soft-FEC solution outperforms common adaptive watermarking using hard region separation combined with either soft or hard forward error correction. The soft-masking approach helps to lower the bit error rate approximately by a factor 5 for commonly used JPEG compression ( $QF \geq 10$ ) without any extra coding effort or quality degradations during watermark embedding.



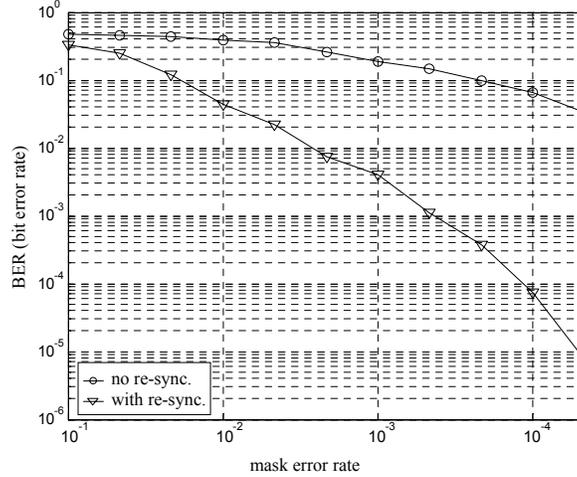
**Fig. 15.** Empirical bit error rate for lossy JPEG compression for different values of quality factor after forward error correction for the adaptive watermarking approach.

#### 4.2 Selective Watermarking

If in Equ. 3 (section 2.3) the step size  $\Delta_1$  is zero and no data is embedded in the perceptually sensitive regions, it follows that  $O$ , and  $s_1$  are zero too. Although this special form, the weighting function  $f_2(C, s_2)$  and hence the segmentation feature have still influence. That means if the feature is far from the threshold, the probability of occurrence of an insertion/deletion error is low. On the other hand, if the feature is close to the threshold the new super trellis decoder can benefit from this knowledge since the weighting process raises the error metric for all interconnected decoders that are out of sync.

In Fig. 16, the probability of occurrence of mask transitions, which we denote as mask error rate, is shown against the bit error rate (BER). The bit error

rate also includes burst errors that result from de-synchronization due to insertions/deletions.



**Fig. 16.** Empirical mask error rate against the bit error rate after forward error correction with and without re-synchronization for the selective watermarking approach.

## 5 Conclusions

In this paper, an integration of perceptual watermarking adaptation into the soft-decision decoding of common forward error correction was proposed. In prior watermarking approaches this adaptation process was always excluded from soft-decoding. We proved that a soft-integration yields impressive improvements in bit error correction during watermark retrieval. No pre-distortion has to be used to avoid adaptation errors during watermark extraction. Hence, the host image quality is not degraded. Furthermore, even adaptation techniques can be used where it is computational infeasible to project a pre-distortion back onto the host image. We showed that the performance can be further gained by combining our soft-decoding approach with a recently proposed channel coding technique capable of correcting insertion/deletion errors at the decoder site. We introduced some modifications for this insertion/deletion correction algorithm to improve the overall performance. Our proposed technique can be integrated very easily into numerous watermarking applications without loss of security, performance or functionality.

## References

1. Schlauweg, M., Pröfrock, D., Müller E.: JPEG2000-Based Secure Image Authentication. In Proc. of ACM Multimedia and Security Workshop, Geneva, Switzerland, (2006) 62-67

2. Wolfgang, R. B., Podilchuk, C. I., and Delp, E.J.: Perceptual Watermarks for Images and Video. In Proc. of IEEE - Special Issue on Identification and Protection of Multimedia, Vol. 87 (7), (1999) 1108-1126
3. Hernández, J. R., Delaigle, J.-F., and Macq, B.: Improving Data Hiding by Using Convolutional Codes and Soft-Decision Decoding, In Proc. of SPIE Security and Watermarking of Multimedia Contents, San José, USA, (2000) 24-47
4. Sharma, G., and Coumou, D. J.: Watermark Synchronization: Perspectives and a New Paradigm, In Proc. of Conference on Information Sciences and Systems, Princeton, USA, (2006) 1182-1187
5. Davey, M. C., and MacKay, D. J. C.: Reliable Communication over Channels with Insertions, Deletions and Substitutions, IEEE Transactions on Information Theory, Vol. 47 (2), (2001) 687-698
6. Solanki, K., Jacobsen, N., Madhow, U., Manjunath, B. S., and Chandrasekaran, S.: Robust Image-Adaptive Data Hiding Using Erasure and Error Correction, IEEE Transactions on Image Processing, Vol. 13 (12), (2004) 1627-1639
7. Barni, M.: Shedding Light on Some Possible Remedies Against Watermark De-Synchronization: A Case Study, In Proc. of SPIE Security, Steganography, and Watermarking of Multimedia Contents VII, San José, USA, (2005) 106-113
8. Mansour, M. F., and Tewfik, A. H.: Efficient Decoding of Watermarking Schemes in the Presence of False Alarms, In Proc. of IEEE Workshop on Multimedia and Signal Processing, Cannes, France, (2001) 523-528
9. Dos Santos, M. P. F., Clarke, W. A., Ferreira, H. C., and Swart, T. G.: Correction of Insertions/Deletions using Standard Convolutional Codes and the Viterbi Decoding Algorithm, In Proc. of IEEE Information Theory Workshop, Paris, France, (2003) 187-190
10. Swart, T. G., Ferreira, H. C., and dos Santos, M. P. F.: Using Parallel-Interconnected Viterbi Decoders to Correct Insertion/Deletion Errors, In Proc. of IEEE AFRICON Conference in Africa, Gaborone, Botswana, (2004) 341-344
11. Vila-Forcén, J. E., Voloshynovskiy, S., Koval, O., Pun, T., and Pérez-González, F.: Worst Case Additive Attack against Quantization-Based Data-Hiding Methods, In Proc. of SPIE Security, Steganography, and Watermark of Multimedia Contents VII, San José, USA, (2005) 136-146