

Correction of Insertions and Deletions in Selective Watermarking

M. Schlauweg, D. Pröfrock, and E. Müller

*Institute of Communications Engineering,
Faculty of Computer Science and Electrical Engineering, University of Rostock,
Rostock 18119, Germany*

Abstract

Common forward error correction is designed to correct substitution errors only. There is no detection of insertions/deletions. Such systems are usually employed in digital watermarking schemes. If watermark data is embedded selectively based on content features where de-synchronization can occur during extraction, common error correction can result in bursts of errors. But there are a few techniques that are able to handle this kind of error. We analyze these techniques, which can be separated into three approaches, namely concatenated coding, dynamic programming, and punctured channel coding. As demonstrated, the latter one fails to correct de-synchronization in second generation watermarking schemes, if the number of selected embedding locations is much smaller than the number of host signal samples. In this paper, we propose a new method that outperforms all other methods presented so far concerning insertion/deletion error correction in second generation watermarking schemes.

1. Introduction

Digital watermarking is a young technology that aims to solve the problems of (among others) illegal distribution of multimedia content and integrity verification. During watermark embedding, the multimedia content (e.g., audio, image, video) is slightly changed. In this way, additional data like ownership information, a user fingerprint, or a signature can be integrated in the content. At watermark extraction side, these slight host signal changes can be detected and thus the embedded information can be retrieved.

The goal is to embed watermark data as robust as possible without causing perceivable distortions. For that reason, newer content adaptive methods consider human perceptual models and embed the data selectively within host signal locations where the perceiv-

able distortion is low. Perceptually sensitive regions are left undisturbed. That means, if the host signal $x = \{x_n \in \mathbb{R} : 1 \leq n \leq N\}$ consists of N signal samples, watermark embedding takes place only for a subset of K locations. Examples for the field of selective image or audio watermarking can be found in [1] - [5], and [9]. Apart from perceptual optimization efforts, selective watermarking is also applied when it is impossible to use the full set of host signal samples. At content-dependent embedding approaches as for instance in [6], some host signal samples don't have enough content to embed data. In addition to that, in medical or cartographic image applications some selected image regions are of high interest and must remain unchanged. Further, at so called second generation watermarking methods, e.g., in [1] and [5], only a small subset of content feature points is selected for watermarking. For these systems, the number of selected embedding locations K is much smaller than N .

To embed watermark data selectively the host signal has to be separated into subsets, where we assume that this separation is only based on the content of the original signal. During extraction, the same separation has to be determined from the watermarked signal without any side information transmitted from the embedding process. Discrepancies yield errors, even if no attack has been applied. A host sample x_n toggling from "used" to "not used" or vice versa results not only in one common substitution error (binary: $0 \rightarrow 1$ or $1 \rightarrow 0$) but 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. In conjunction with common substitution errors this kind of data transmission disturbances are known as IDS-channel errors (insertions/deletions/substitutions). For instance, in Figure 1 and Figure 2, the embedding as well as extraction masks are shown for a selective watermarking scheme proposed in [1]. White

regions indicate host signal locations where to embed watermark data. If the separation mask slightly changes due to signal processing or attacks, insertions and deletions occur. For the example image, there are 7 insertions and 2 deletions if the decoder assumes to extract data only from white blocks.



Figure 1. Original image (left) and texture-feature-based embedding mask (right).

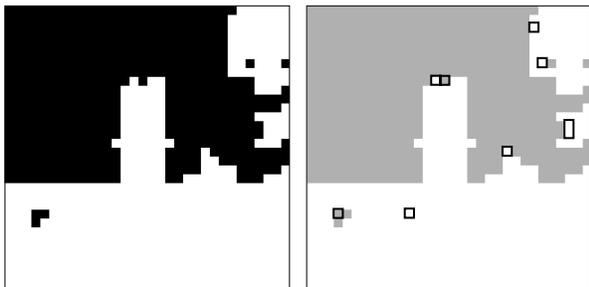


Figure 2. Slightly changed separation mask due to signal processing

Common forward error correction (FEC) is designed to correct substitution errors only. There is no detection of insertions/deletions. In such systems, synchronization loss will result in a burst of errors until synchronization is re-established. But there are a few techniques that are able to handle this kind of error. In this paper, we analyze these techniques (in section 2), which can be separated into three approaches, namely *concatenated coding*, *punctured channel coding*, and *dynamic programming*. As we demonstrate, the IDS correction approach that uses punctured channel coding fails if $K \ll N$ or, in other words, if the ratio K/N is too small. This case can occur for second generation watermarking schemes. In section 3, we propose a new method based on extended dynamic programming, similar to the system presented by Swart *et al.* in [10]. It is independent of the ratio K/N . This new IDS error correction method outperforms all other methods presented so far, as we show in section 4. Using our extended error correction method second generation watermarking, for instance the one in [5], is able to handle host signal sample insertions/deletions.

2. Insertion/deletion error correction

The term watermark de-synchronization is used twice in literature. The term sometimes 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 symbol insertions and deletions is the major problem in content adaptive watermarking. Known approaches such as ESD (*exhaustive search detection*) or TMD (*template matching detection*) usually applied to re-synchronize the whole message [7] are computational infeasible in case of IDS.

Most error correction schemes are designed to correct substitution errors. It is assumed that the decoder knows the block boundaries in case of *block codes* or the message length in 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 [8] and [10], the authors give an overview of recent approaches to channel re-synchronization. One technique is, for example, the successive application of FEC codes with different properties in a concatenated coding scheme. Using marker sequences the message to be transmitted is packetized. Burst errors due to synchronization loss within a packet are passed to a decoder able to correct these errors. Another technique treats the host signal locations where no data is embedded as erasures. Similar to the idea of punctured channel coding, the encoder simply discards the code symbols at these locations. Afterwards, the decoder inserts don't-care states for the assumed locations knowing only the host signal separation 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. With reduced computational effort for all insertion/deletion locations the decoder checks the possibilities in a forward-backward algorithm and determines the most probably correct message. All these techniques will be discussed and compared in the following subsections.

2.1. Concatenated coding

One possibility of IDS correction, employed for selective watermarking by Davey and McKay in [8], is the application of concatenated codes, commonly used for satellite data transmission. As shown in Figure 3, message $m = \{m_l \in \{0,1\} : 1 \leq l \leq L\}$ is encoded successively using two or more FEC codes with different properties. During extraction, first, an inner code is applied to infer the locations of insertion/deletion errors and an outer code afterwards corrects these errors, as in [8] or [9]. Thereby, carrier signals (marker) are provided for the outer burst-error-correcting decoder. If these markers are not in their expected locations, the outer decoder can determine the insertion/deletion locations.

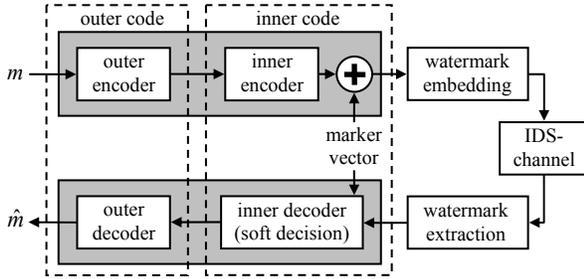


Figure 3. Concatenated coding using inner/outer codes and marker vectors for IDS correction [9].

2.2. Punctured channel coding

Another IDS error correction approach is known from channel coding. $F = \{F_n \in \mathbb{R} : 1 \leq n \leq N\}$ is the separation criterion and the embedding locations are p .

$$p = \{p_k \in \mathbb{N}^+ : \tau \leq F(p_k), 1 \leq p_k \leq N, 1 \leq k \leq K\} \quad (1)$$

After encoding the message resulting in the sequence $w = \{w_n \in \pm 1 : 1 \leq n \leq N\}$ the encoder discards w_n at host signal locations \bar{p} where no data is embedded.

$$\bar{p} = \{\bar{p}_n \in \mathbb{N}^+ \setminus p_n : 1 \leq n \leq N\} \quad (2)$$

This so called puncturing of code word symbols is usually employed to adapt the length of a coded message to a given code rate or frame length. At the receiver side, the decoder must know the locations of the symbols not transmitted. By inserting don't-care symbols (usually value 0) at these locations and applying FEC decoding that is able to handle erasures the message can be retrieved. By this strategy the error correction efficiency of an employed $1/r$ -rate code is

lowered, but the punctured code is as good as not punctured codes of rate $(N-K)/r$ in general.

Solanki *et al.* use this idea of puncturing in a watermarking system [3]. The authors treat the host signal locations where no data is embedded as erasures. The decoder inserts don't-care symbols for the assumed locations knowing only the host signal separation criterion but not the explicit locations. As in Figure 4, insertions now become errors, and deletions become additional don't-care states that can be corrected by the decoder. Solanki *et al.* suggest employing either *Reed-Solomon* (RS) codes or *Repeat-Accumulate* (RA) codes, where the first are only preferable to RA codes if IDS errors are expected to occur burst-wise. In addition to that RS codes in contrast to RA codes usually cannot benefit from *soft decoding*.

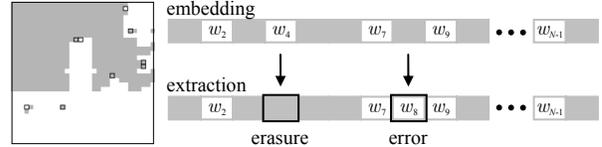


Figure 4. IDS error correction using erasures during watermark extraction.

2.3. Dynamic programming

Even a few years before Solanki *et al.*, another approach for the purpose of IDS error correction in a watermarking system with selective embedding was proposed by Mansour and Tewfik [4]. Their technique, based on *dynamic programming*, recovers the correct message if extra bits (false alarms) are added to the body of the message at random locations. The authors modify the *Viterbi algorithm* used to decode an 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 τ for selecting embedding locations at the decoder is chosen to be less than the threshold at the

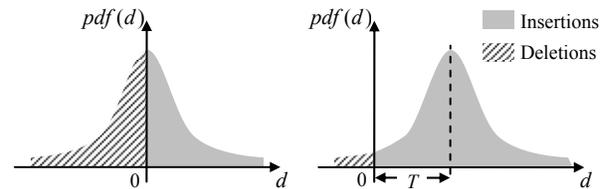


Figure 5. Probability density function of the occurrence of threshold transitions and hence insertions/deletions at the decoder (left). Threshold shifted at decoder - most deletions become insertions (right).

embedding side. The new threshold is $\tau - T$. Hence, the probability of occurrence of deletion errors is lower than for insertion errors (see Figure 5). But in this case, all embedding locations 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.

Recently, another IDS 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 the sequence of symbols $w_k = x(p_k): 1 \leq k \leq K$ 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. Each of the parallel-interconnected Viterbi decoders is one bit out of sync with the others (see Figure 6). By monitoring the rate of change for the accumulated error metrics, one is able to ascertain which of the decoders is in synchronization. Based on the idea in [11], where several standalone decoders were used, Swart *et al.* integrated the framework into one larger Viterbi decoder and called it super trellis decoder.

At every observation step one sub decoder is lagging by one bit, denoted by -1 , and one is leading by one bit, denoted by $+1$. If for an observation step one of the sub decoders out of sync results in a more probable metric the decoding process changes to this decoder. In that case, an input sample will be skipped or repeated before the decoder continues.

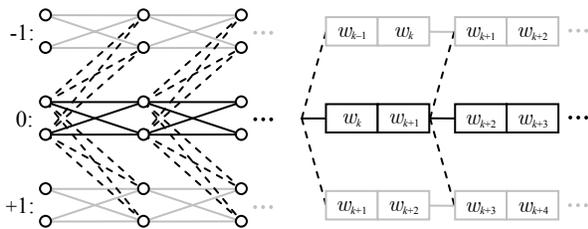


Figure 6. Representation of the super trellis [10] (left). Another illustration to visualize the indices of the input samples for an $1/2$ -rate code (right).

2.4. Properties of the correction techniques

If ratio K/N is very small (or in other words $K \ll N$) the technique proposed by Solanki *et al.* fails, as dem-

onstrated in what follows. For example, in [5], we proposed a second generation watermarking scheme based on embedding at robust image features points. As shown in Figure 7, the algorithm selects, e.g., $K = 32$ content feature points to embed a watermark in a 512×512 pixel image. Since these feature points could be located at any pixel location, the overall host signal length is $N = 262144$. As explained in [5], insertion and deletion errors can occur for this watermarking scheme, which have to be handled by IDS error correction.

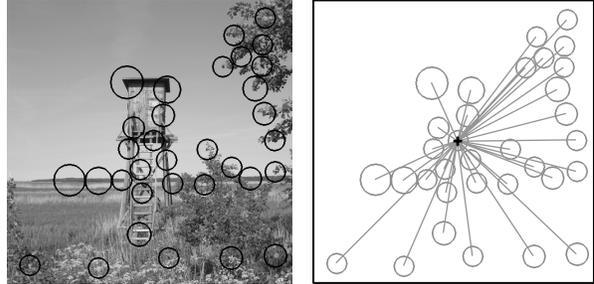


Figure 7. Example image with $K = 32$ content feature points selected for data embedding using a second generation watermarking [5].

From channel coding it is known that a given (η, κ) -RS code with rate $r = \eta/\kappa$ can correct up to f errors and e erasures as long as equation $e + 2 \cdot f \leq \eta - \kappa$ holds (upper bound). But, in a punctured coding application with RS codes all non-embedding host locations must be seen as erasures. That means a basic number of $\tilde{e} = N - K$ erasures has to be corrected by the decoder resulting in Eq. (3) representing the correction capability left.

$$N - K + e + 2 \cdot f \leq \eta - \kappa \quad (3)$$

Similarly, for punctured $1/r$ -rate RA codes we found out that approximately following equation must be hold:

$$5 \cdot r \leq K/N \quad (4)$$

For the above example ($N = 262144$, $K = 32$), both punctured coding-based IDS correction approaches are computational infeasible. But otherwise, the approach is useful as presented in [3].

As opposed to Solankis punctured channel coding, for the techniques by Davey and MacKay as well as Swart *et al.* the ratio K/N doesn't matter. This turns out to be a basic requirement for second generation watermarking schemes as demonstrated above.

3. The modified super trellis decoder

In [1], we implemented the method proposed by Swart *et al.* and made some modifications on their super trellis decoder. For example, Swarts decoder simply repeats the previous signal sample, e.g., w_{k+1} , as shown in Figure 6, if a deletion is assumed for a symbol in the range $]p_k, p_{k+1}[$. But since the decoder has knowledge about all samples, we propose to integrate a host signal sample delivered by the watermark extractor that is assumed to be deleted. Here, we denote e the deleted symbol, where $1 \leq k \leq K$ and $1 \leq n \leq N$.

$$e = \{e_k \in x(\bar{p}_n) : \arg \min_n (|F_n - \tau|), p_k < \bar{p}_n < p_{k+1}\}$$

If there are more than one deletion candidates in the range $]p_k, p_{k+1}[$, we suggest to chose the one closest to the separation threshold. Further, considering Figure 6, the super trellis decoder from [10] is not able to deduce the actual insertion/deletion location within the pair of coded symbols. As can be seen in Figure 8, our super trellis extension checks for all possible combinations of insertion/deletion locations and hence, finds as well as corrects the actual error location. This gains the overall error correction capabilities of the decoder. Additionally, we apply a weighting to the pair of samples within the window for the trellis paths out of sync. For this, we use the distance of F_n to the separation threshold for the host signal sample location \bar{p}_n assumed to be inserted or deleted, respectively.

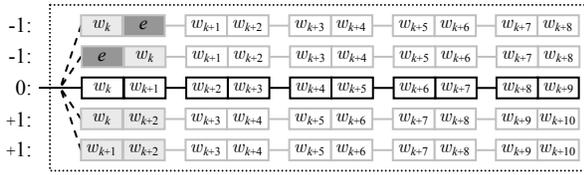


Figure 8. Modified super trellis decoder for a code of rate 1/2, where 2 input samples form a pair of symbols. Window length is 5.

The de-synchronization error correction capability of the decoder presented in [1] (denoted MSTrellis1) is limited to one de-synchronization error per window length. That means, if there are more than one insertion/deletion errors within the observed message sequence the decoder decision probably fails.

A further extension that we propose for the decoder can handle more than one de-synchronization errors per window. For this purpose, we implement the forward algorithmic part of the new modified super trellis decoder as a recursive tree data structure (see Figure 10).

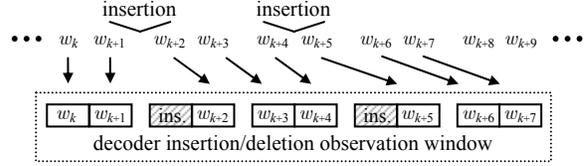


Figure 9. Example: 2 insertions within the insertion deletion observation window (window length is 5).

As opposed to the MSTrellis1 decoder, here, for every node of the tree all possibilities of insertions and deletions are checked. By this strategy, the new algorithm clearly outperforms the decoder by Swart *et al.* as well as the decoder presented in [1]. We denote this new modified super trellis decoder MSTrellis2.

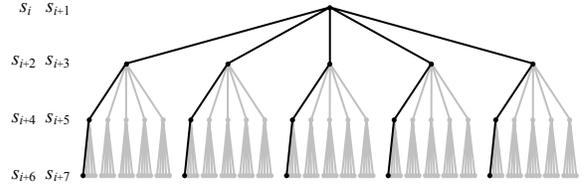


Figure 10. Tree structure of our new super trellis decoder for a code rate 1/2. Window length is 4.

For a 1/2-rate code, at every node of the tree structure representing an input symbol pair there are five branches. The left-most branch at every node represents the case of no symbol insertion/deletion. The other four branches are all possible insertion/deletion locations, checked by the new decoder. The black branches in Figure 10 visualize the functionality of algorithm MSTrellis1, which only checks for an insertion/deletion at the first symbol pair within the observation window. Since the input sample constellations for different branches are equal for a lot of nodes, we can reuse several trellis path metrics and hence lower the computational effort.

Finally, we found out that the overall correction performance can be gained if the decoder is applied a second time for the same input message using information about insertion/deletion locations generated during the first run. We call this version of our new algorithm MSTrellis3.

4. Experimental results

For performance examination purpose we define a binary symmetric channel model similar to the one applied in [8] - [10]. Instead of using a probability value for substitution errors, P_f , we define this error

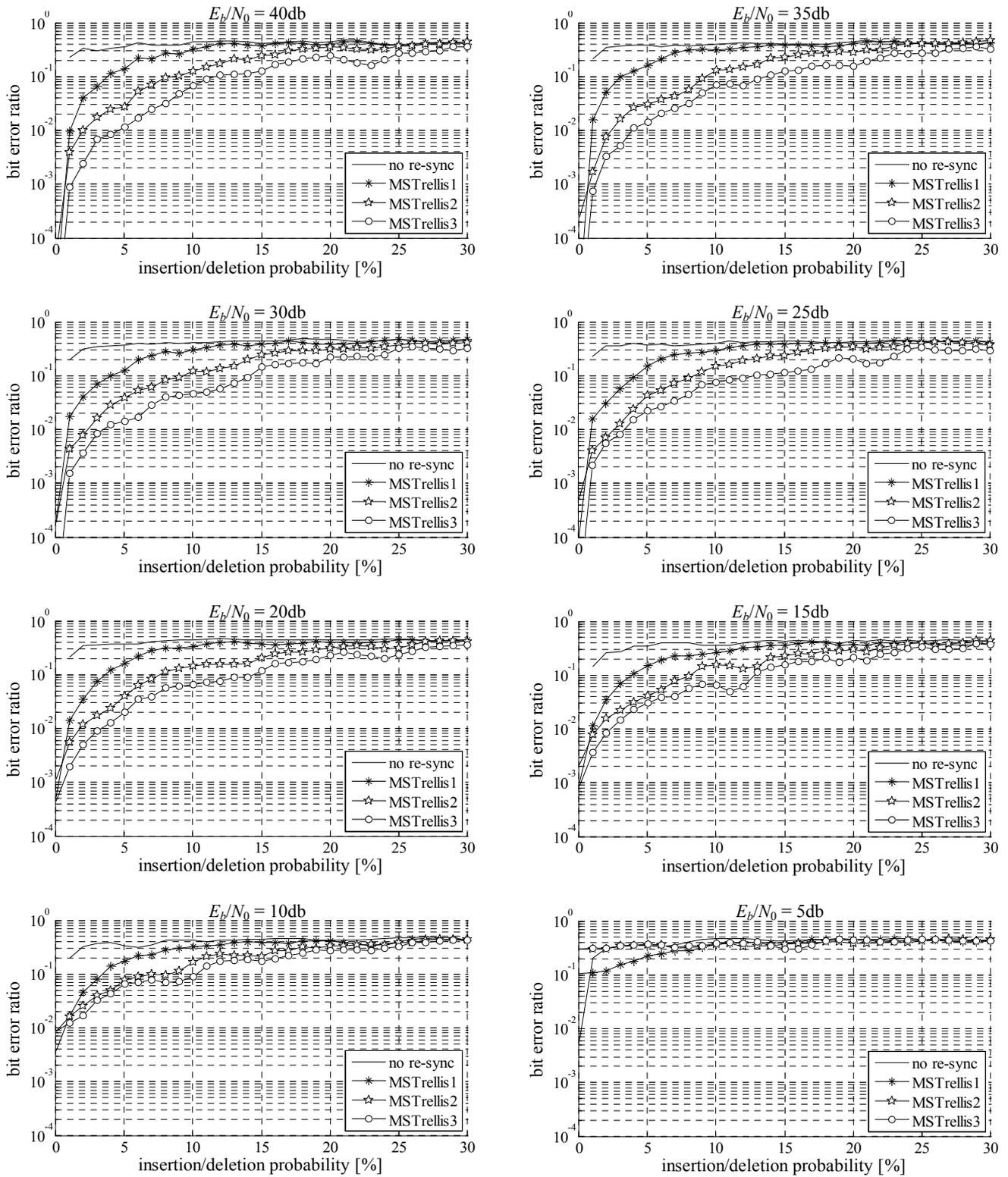


Figure 11. Simulation results of the performance analysis of our new IDS error correction method: bit error ratio vs. insertion/deletion probability. Simulation parameters: code rate $r = 1/2$, window length = 5.

rate by the *energy per bit to noise power spectral density ratio* E_b/N_0 known from channel coding. It is the normalized signal-to-noise ratio (SNR) measure, also known as the “SNR per bit”. It is useful when comparing the bit error rate performance of different digital modulation schemes. If $E_b/N_0 = \infty$, then $P_f = 0$, following Eq. (5).

$$P_f = 0.5 \cdot \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad (5)$$

Substitution errors result from signal disturbances that we model for *additive white Gaussian noise* with zero mean and variance σ^2 , following Eq. (6). The disturbed signal is $\hat{w} = w + z$, where z is the noise added.

$$\sigma^2 = 10^{-\left(\frac{E_b/N_0}{10}\right)} \quad (6)$$

To simulate symbol de-synchronizations we use the insertion/deletion probability P_d per symbol. For example, if $K = 1000$ and $P_d = 1\%$, then 10 insertions/deletions can occur within the coded message.

We simulated the bit error ratio vs. insertion/deletion probability on numerous random messages, encoded using standard convolutional codes ($r = 1/2$, generator polynomial = [7 5], constraint length = 3).

In Figure 11, the three versions of our new decoder (MSTrellis1, MSTrellis2, MSTrellis3) are tested for different AWGN parameters E_b/N_0 . As can be seen, if the added noise is too important, the decoders fail. But, in the same way the unmodified Viterbi decoder without re-synchronization fails as well. This decoder is the basis for our new IDS decoder. Here, the overall improvement can be seen very clearly.

Figure 12 is generated using separate simulation results from [8] and [10]. By drawing these curves together with the simulation results of our new decoders we are able to directly compare the different methods. As can be seen, the decoder MSTrellis3 developed in this paper outperforms all other methods presented so far for the same code of rate 1/2. Especially, if the probability of occurrence of insertions/deletions is larger than 1%, our decoder performs clearly better. But, in the range $0 \leq P_d \leq 1$, the method by Davey and MacKay seems to be better.

It could be future work to also implement our modified super trellis decoder for other code rates and to test the decoder for other generator polynomials as well. Further, we will integrate the new algorithm into watermarking systems and test the performance concerning attacks like, for example, compression.

legend: Swart *et al.* [10] Davey *et al.* [8] Schlawweg *et al.* (code rate 1/2)
 -x- code rate 1/2 -●- code rate 0.71 * MSTrellis1
 + code rate 1/3 -□- code rate 1/2 -v- code rate 3/14 -○- MSTrellis3
 -△- code rate 1/20

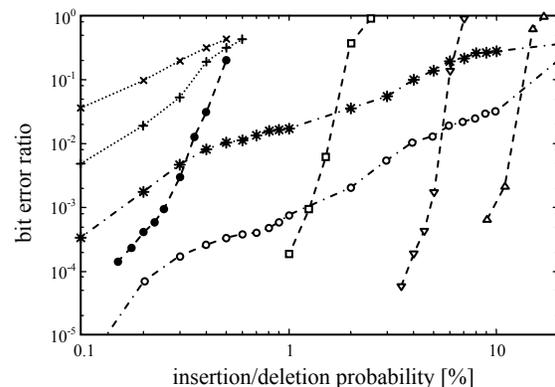


Figure 12. Bit error ratio vs. insertion/deletion probability. Comparison of our new IDS correction method to the approaches by Swart *et al.* as well as Davey and MacKay, where $P_f = 0$.

5. Conclusion

In this paper, we proposed a new error correction method that is able to detect and correct substitutions as well as insertion/deletion errors. Especially in the field of selective content-based watermarking where insertions and deletions can occur during extraction, common error correction often results in bursts of errors. But there are three techniques that are able to handle this kind of error. In this work, we analyzed these techniques and found out that only two of them are useful if the number of selected watermark embedding locations is much smaller than the number of host signal samples. We showed that our new method, which is independent of this ratio, outperforms all other methods presented so far.

6. References

- [1] Kutter, M., Bhattacharjee, S. K., and Ebrahimi, T., “Towards Second Generation Watermarking Schemes,” *In Proc. of IEEE International Conference on Image Processing*, Kobe, Japan, pp. 320-323, 1999.
- [2] Schlawweg, M., Pröfrock, D., and Müller, E., “Soft Feature-Based Watermark Decoding with Insertion/Deletion Correction,” *In Proc. of Information Hiding Workshop*, Saint Malo, France, pp. 236-250, 2007.
- [3] 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), pp. 1627-1639, 2004.

- [4] 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, pp. 523-528, 2001.
- [5] Schlauweg, M., Pröfrock, D., Zeibich, B., and Müller, E., "Self-Synchronizing Robust Texel Watermarking in Gaussian Scale-Space," *In Proc. of ACM Workshop on Multimedia and Security*, Oxford, UK, 2008.
- [6] Pröfrock, D., Schlauweg, M., Müller, E., "Content-Based Watermarking by Geometric Warping and Feature-Based Image Segmentation," *In Proc. of ACM/IEEE International Conference on Signal-Image Technology & Internet-Based Systems*, Hammamet, Tunisia, pp. 572-581, 2006.
- [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, pp. 106-113, 2005.
- [8] 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), pp. 687-698, 2001.
- [9] Coumou, D. J. and Sharma, G., "Insertion, Deletion Codes with Feature-Based Embedding: A New Paradigm for Watermark Synchronization with Applications to Speech Watermarking," *In IEEE Transactions on Information Forensics and Security*, Vol. 3 (2), pp. 153-165, 2008.
- [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, pp. 341-344, 2004.
- [11] 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, pp. 187-190, 2003.