Resilient Transmission of H.264/AVC Video Sequences using Probabilistic Neural Networks

Reuben A. Farrugia and Carl J. Debono
Department of Communications and Computer Engineering
University of Malta
Msida MSD2080, Malta
{rrfarr, cjdebo}@eng.um.edu.mt

***Abstract*—H.264/AVC is expected to become an essential component in the delivery of wireless multimedia content. While achieving high compression ratios, this codec is extremely vulnerable to transmission errors. These errors generally result in spatio-temporal propagation of distorted macroblocks (MBs) which significantly degrade the perceptual quality of the reconstructed video sequences.**

This paper presents a scheme for resilient transmission of H.264/AVC streams in noisy environments. The proposed algorithm exploits the redundant information which is inherent in the neighboring MBs and applies a Probabilistic Neural Network (PNN) classifier to detect visually impaired MBs. This algorithm achieves Peak Signal-to-Noise Ratio (PSNR) gains of up to 14.29 dB when compared to the standard decoder. Moreover, this significant gain in quality is achieved with minimal overheads and no additional bandwidth requirement, thus making it suitable for conversational and multicast/ broadcast services where feedback-based transport protocols cannot be applied.

Keywords-error-resilient video coding; H.264/AVC; learning systems;wireless video transmission;

I. Introduction

Recent advances in wireless communication systems have made location-independent access to multimedia content possible. The H.264/AVC standard is expected to become an essential component in emerging wireless multimedia applications such as multimedia telephony, streaming and multicast/broadcast applications due to its high compression efficiency and network friendly design [1] – [3]. However, like all other video compression standards, H.264/AVC is extremely susceptible to transmission errors where it only manages to detect 57% of the corrupted macroblocks (MBs) [4]. This generally results in spatio-temporal propagation of the undetected visual artifacts which significantly degrade the quality of the reconstructed video sequence.

The error detection capabilities of H.264/AVC were investigated in [4], where slice level concealment (SLC) was adopted. This method achieves its goals with a pessimistic approach where all the MBs present in a corrupted slice, including uncorrupted MBs, are concealed. The solution proposed in [5] requires more bandwidth and thus reduces the compression efficiency of the codec. Relation-based fragile watermarking was successfully adopted in [6], however, the method is non-standard compatible and includes additional complexities in both the encoder and decoder. The authors in [7] – [8] adopt Sequential Decoding in order to derive the most-likelihood bitstream to be decoded. Although these methods provide additional robustness to the transmitted bitstream, they do not discriminate between annoying and imperceptible visual artifacts, and therefore these artifacts have the same probability of being detected.

Different image-level error-resilient mechanisms optimized to detect major visual distortions for other video coding systems are present in literature [9] – [12]. However, the authors in [9] achieve a substantial gain in quality at the expense of a significant increase in complexity which makes it unsuitable for real-time applications, while the methods presented in [10] – [12] are not robust enough to guarantee an acceptable level of quality when transmitting over a noisy channel.

This paper presents an adaptation of the error-resilient mechanism previously presented in [13], for H.263 coded video streams, to H.264/AVC coded sequences. Due to the differences between codecs, a different set of 8 image-level features were adopted in this work. These features are then used by the Probabilistic Neural Network (PNN) classifier to detect corrupted MBs which are subsequently concealed. The proposed algorithm manages to detect most of the major visual distortions and Peak Signal-to-Noise Ratios (PSNR) gains of up to 14.29 dB were achieved.

This paper is organized as follows. The proposed error detection algorithm is presented in section II while the simulation results which highlight the gain in quality provided by the proposed system are presented in section III. The final comments and conclusion are presented in section IV.

II. Proposed Error Detection Algorithm

The proposed error detection algorithm is incorporated within the modified decoder, as shown in Fig. 1. All transport protocols adopted in conversational and multicast/ broadcast services such as H.324M and RTP/UDP/IP employ bit-level error checksums to detect corrupted packets [1]. These error checksums are used to label as potentially corrupted those MBs present within a corrupted packet. A set of features

978-1-4244-1688-2/08/$25.00 ©2008 IEEE

which exploit the redundant information inherent within neighboring MBs are then used by the PNN classification module to detect those MBs which provide major visual distortions and appropriately concealing them. The proposed algorithm is applied only on those MBs which were labeled as being potentially corrupted, thus minimizing the processing required by the error detection process.

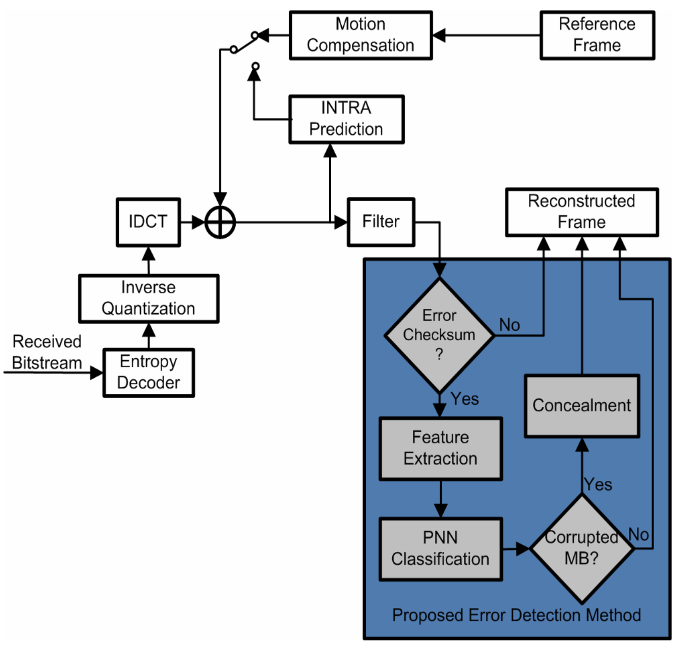


Fig. 1. Proposed Error Detection Algorithm

A. *Feature Extraction*

The feature extraction module extracts eight features which exploit both color and texture consistency. These features were found to provide enough information to discriminate between corrupted and uncorrupted MBs and are the 1) Average Inter-Sample Difference across Boundaries (AIDB), 2) Mean and 3) Standard Deviation of the Internal AIDB of a 4x4 block, 4) vertical and 5) horizontal Internal AIDB, 6) Mean and 7) Standard Deviation of the Average Internal Difference between Subsequent Blocks (AIDSB) and 8) Texture Consistency (TC). These features were computed in the perceptually uniform CIELUV color space model, where the distance in this space generally corresponds to the difference in color sensation [14].

Average Inter-sample Difference across Boundaries (AIDB)

The AIDB feature is a measure of the spatial fitness of the MB. As illustrated in Fig. 2a, the dissimilarity across each MB boundary *AIDB(M:X)*, is computed using:

$$AIDB(M:X)=\begin{cases}\frac{1}{K}\left\|p^{in}-p^{out}\right\|_2 & \text{if X } \textit{is available}\\ 0 & \text{Otherwise}\end{cases} \qquad (1)$$

where $X\in\{N,S,E,W\}$, K = 16, and $\|\square\|_2$ is the L^2 norm computed in the CIELUV color space model. The *AIDB* is then derived by averaging the *AIDB(M:X)* over the available neighboring MBs.

Internal AIDB of a block (IAIDB$_{Block}$)

Each MB is divided in a grid of 16 4x4 blocks. As shown in Fig. 2b, the dissimilarity across each block boundary *IAIDB$_{Block}$(M:X)* is computed using (1) with K = 4. The *IAIDB$_{Block}$* feature is then derived by averaging the *IAIDB$_{Block}$ (M:X)*.

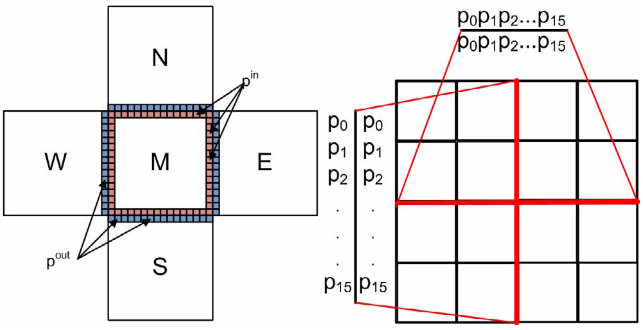


Fig. 2 a) AIDB and IAIDB$_{Block}$ features b) IAIDB feature

Internal AIDB

The *IAIDB* feature is a measure of the dissimilarity across the internal horizontal/vertical boundary as illustrated in Fig. 2b and is computed using:

$$IAIDB=\sum_{i=0}^{K-1}\left\|p_i^{in}-p_i^{out}\right\|_2 \qquad (2)$$

where K = 16 and $\|\square\|_2$ is the L^2 norm computed in the CIELUV color space model.

Average Internal Difference between Subsequent Blocks (AIDSB)

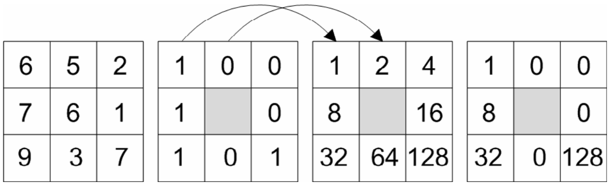
The *AIDSB* feature provides a measure of the temporal consistency between the current MB under test and the corresponding MB in the previous frame. The MB is dissected into 16 4x4 blocks, where the dissimilarity between each block b_t in the current MB and the corresponding block in the previous frame b_{t-1} are computed using:

$$AIDSB=\frac{1}{K^2}\left\|p_t-p_{t-1}\right\|_2 \qquad (3)$$

where p correspond to the pixels in block b, and K = 4.

Texture Consistency(TC)

The Local Binary Pattern (LBP) operator [15] is a powerful grey-scale invariant texture measure. The original 3x3 neighborhood (Fig. 3a) is thresholded by the value of the center pixel. The values of the pixels in the thresholded neighborhood (Fig. 3b) are multiplied by the binomial weights given to the corresponding pixels (Fig. 3c). The result in this example is illustrated in Fig. 3d. Finally, the values of the eight pixels are summed to obtain the texture unit.



6	5	2
7	6	1
9	3	7

(a)

1	0	0
1		0
1	0	1

(b)

1	2	4
8		16
32	64	128

(c)

1	0	0
8		0
32	0	128

(d)

Fig. 3. Computation of the Local Binary Pattern

The LBP histograms of the current MB h_t and the corresponding MB in the previous frame h_{t-1} were first computed. The *TC* is then computed by evaluating the histogram insertion method given by:

$$TC = \sum_{i=0}^{B} \min\left(h_t, h_{t-1}\right) \tag{4}$$

where B is the number of bins.

B. Probabilistic Neural Networks (PNN)

The PNN classifier [16] illustrated in Fig. 4 was trained by connecting the normalized training pattern *x* on the input units. The weights linking the input units and the pattern units are set such that $w_k = x_k$ where k = 1, 2, … , n (where in this case n = 8). Then, a single connection from the first pattern unit is made to the pattern unit corresponding to the known class of that pattern.

The trained network is used for classification where the test pattern *x* is first normalized. Each pattern unit computes its non-linear activation function given by:

$$\mathbf{z} = \exp\left(\frac{\langle x \square w\rangle - 1}{\sigma^2}\right) \tag{5}$$

where σ is the smoothing parameter and $\langle \square \rangle$ represents the inner dot product. The summation units provide the Parzen window estimate of the underlying distribution. The decision unit then discriminates the class that the test vector belongs to.

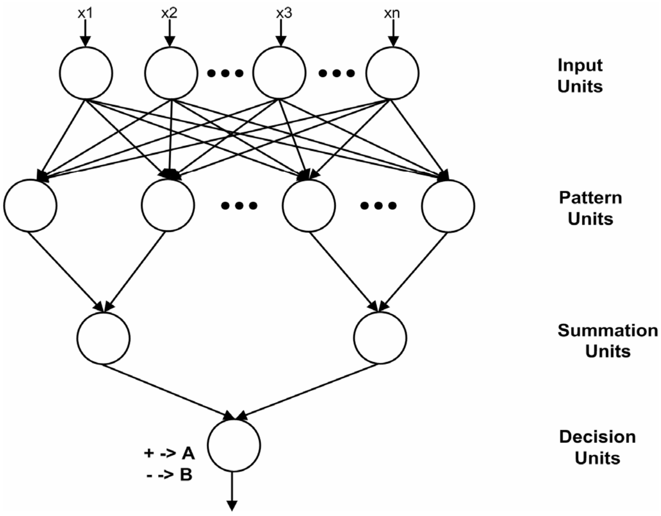


Fig. 3.Architecture of the PNN Classifier

III. Simulation Results

The visually impaired MBs generally contain different distortion levels (DL), where some of the impairments are imperceptible by the user while others provide significant visual distortions. These distorted MBs were scaled according to Table I. The aim of the proposed algorithm is to maximize the detection of annoying artifacts while being more lenient with imperceptible artifacts.

A set of 1000 feature vectors were used to train the PNN classifier, half of which represent corrupted MBs while the remaining half represent uncorrupted MBs. These were randomly selected from a set of five video sequences: *Foreman*, *Car-phone*, *Coastguard*, *Mobile*, and *News*. The smoothing parameter σ was set to 0.1, as after extensive testing it was found to offer an optimal solution. The recognition of the PNN was tested on another testing set of 1000 feature vectors and its performance is summarized in Table I.

TABLE I. Error Detection Rate of Distorted MBs

Distortion Level (DL)	*Description*	*Error Detection Rate*
0	Uncorrupted/Imperceptible	-
1	Perceptible but not annoying	63.01%
2	Slightly Annoying	92.11%
3	Annoying	100.00%
4	Very Annoying	100.00%

The PNN manages to detect 97.18% of those MBs which contain major visually annoying artifacts (DL4, DL3 and DL2). Those corrupted MBs which are not detected generally provide minimal distortions which are most of the time tolerated by the end-user.

The proposed error detection algorithm was then integrated within the Jointed Model (JM) software [17]. The syntax and semantic rules provided in [4] were used to detect syntax and semantic violations. Two raw QCIF video samples *Miss-America* and *Hall*, which were not used in the training phase, were encoded at 30 frames per second. In compliance with JM software, each slice was encapsulated within RTP/UDP/IP packets and transmitted over a binary symmetric channel (BSC) which yields the highest probability of error in the slice as compared to channels which provide burst errors [9].

Three different decoding methodologies were considered to decode the received corrupted bitstream:

1) Macroblock Level Concealment (MLC) which adopts the syntax and semantic violations described in [4] to detect a corrupted MB. This method will conceal the detected MB and the following ones until the end of the slice.

2) Slice Level Concealment (SLC) which applies the checksum of the UDP protocol to detect corrupted slices and conceals all the MBs contained within the corrupted slice.

3) The proposed algorithm (PNN) which applies the UDP checksum to identify potentially corrupted MBs and then applies the PNN classifier to detect visually distorted MBs which are then appropriately concealed.

The *Miss-America* and *Hall* sequences were corrupted with different error patterns at bit-error rates (BER) varying between 1.00E-005 and 5.00E-003. The performance of the considered decoding methodologies is illustrated in Fig. 4, where it can be immediately observed that the proposed algorithm outperforms the other solutions. This is mainly attributed to the fact that MLC does not manage to detect some major visual distortions which will provide significant degradation in quality of the reconstructed video sequence. On the other-hand the SLC algorithm conceals all MBs contained within a corrupted slice, even if only one corrupted MB is present within the slice. The SLC method generally results in over concealment thus reducing the perceptual quality of the reconstructed video sequence.

The proposed algorithm provides a good compromise between the two other methods, since it tries to localize those corrupted MBs which provide significant visual artifacts using the PNN classification module. The method used was found to detect most visual annoying artifacts which result in significant gains in PSNR which go up to 14.29 dB and 2.63 dB relative to the MLC and SLC methods. The subjective quality gain provided by the proposed solution is illustrated in Fig. 5, where the MLC algorithm does not manage to detect a number of major artifacts while SLC provides over-concealment reducing the quality of the recovered video sequence.

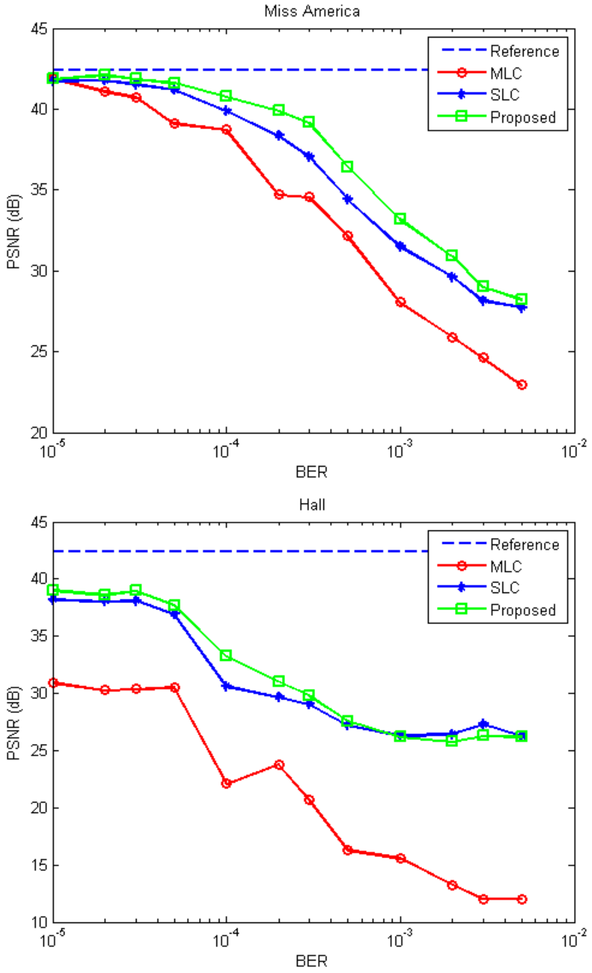


Fig. 4.PSNR gain of the three considered error detection algorithms (a) Miss-America, (b) Hall sequences at different BER

IV. Comments and Conclusion

This paper has provided an adaptation of the error-resilient method previously applied to H.263 encoded video sequences to H.264/AVC. Because of the differences between the coding architectures, a new set of features was adopted by the PNN classifier in order to detect visually impaired MBs for corrupted H.264/AVC video sequences. The proposed method manages to detect most of the artifacts, presenting significant gains in both objective and subjective quality.

The PNN-based algorithm is applied only when there are errors on the channel and therefore the complexity of the system increases with increasing BER. Through simulation, it was found that an increase of computational time of 2.11% is provided by this algorithm at a BER of 5.00E-003. Thus, the complexity added at the decoding side is minimal making the proposed system suitable for real-time applications such as video telephony and multicast/broadcast applications.

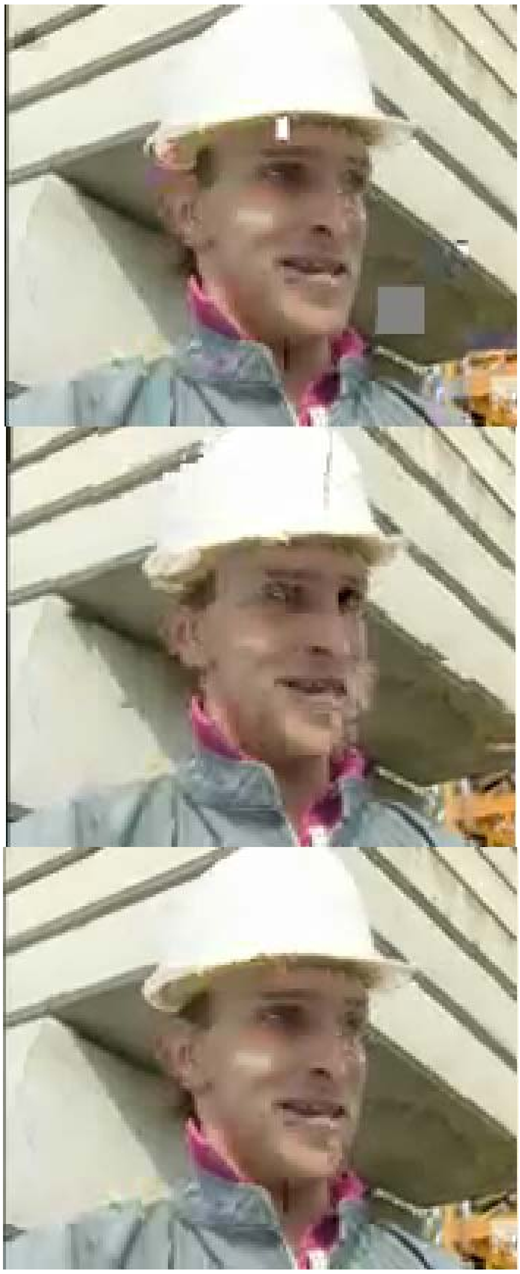


Fig. 5.Frame 228 of Foreman sequence using (a) MLC (b) SLC and (c) Proposed Algorithm

References

[1] T. Stockhammer and M.M. Hannuksela, “H.264/AVC Video for Wireless Trnamission,” *IEEE Wireless Commun.*, vol. 12, no. 4, pp. 6 – 13, Aug. 2005.

[2] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, “H.264/AVC in Wireless Environments,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 647-673, Jul. 2003.

[3] S. Wenger, “H.264/AVC Over IP,” *IEEE. Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, Jul. 2003.

[4] L. Superiori, O. Nemethovam and M. Rupp, “Performance of H.264/AVC Error Detection Algorithm Based on Syntax Analysis,” in *Proc. of 4th Int. Conf. on Advances in Mobile Computing and Multimedia*, Yogyakarta, Indonesia, Dec. 2006.

[5] O. Nemethova, J.C. Rodriquez and M. Rupp, “Improved Detection for H.264 Encoded Video Sequences over Mobile Networks,” in *IEEE Proc. of 8th Int. Symp. on Communication Theory and Applications*, Ambleside, Lake District, UK, 2005.

[6] O. Nemethova, G.C. Forte and M. Rupp, “Robust Error Detection for H.264/AVC using Relation Based Fragile Watermarking,” in *IEEE Proc. of 13th Int. Conf. on Systems and Image Processing*, Budapest, Hungary, 2006.

[7] C. Weidmann, P. Kadlec, O. Nmethova and A. Al Moghrabi, “Combined Sequential Decoding and Error Concealment of H.264 Video,“ in *IEEE 6th Workshop on Multimedia Signal Processing*, Siena, Italy, Oct. 2004.

[8] C. Bergeron and C. Lamy-Bergot, “Soft-input Decoding of Variable-Length Codes applied to the H.264 standard,” in *IEEE 6th Workshop on Multimedia Signal Processing*, Siena, Italy, Oct. 2004.

[9] E. Khan, S. Lehmann, H. Gunji and M. Ghanbari, “Iterative Error detection and Correction of H.263 Coded Video for Wireless Networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 12, pp. 1294-1307, Dec. 2004.

[10] S. Ye, X. Lin and Q. Sun, “Content Based Error Detection and Concealment for Image Transmission over Wireless Channel,” in *IEEE Int. Symp. on Circuit and Systems*, Bangkok, Thailand, 2003.

[11] O. Lehtoranta, T.D. Hamalainen and V. Lappalainen, “Detecting Corrupted Intra Macroblocks in H.263 Video,” *IEEE Workshop on Multimedia Signal Processing*, St. Thomas, Virgin Islands, USA, Dec. 2002.

[12] H. Shyu and J. Leou, “Detection and Concealment of transmission Errors in MPEG-2 Images – A Genetic Algorithm approach,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 937-948, Sept. 1999.

[13] R.A. Farrugia and C.J. Debono, “Enhancing Error Resilience in Wireless Transmitted Compressed Video Sequences through a Probabilistic Neural Network Core,” in *IEEE Proc. of Picture Coding Sysmposium*, accepted for publication, Nov. 2007.

[14] L.V. Tran, “Efficient Image Retrieval with Statistical Color Descriptors,” Ph. D. dissertation, Dept. of Science and Tech., University of Linköping, Sweeden, 2003.

[15] T. Ojala, M. Pietikäinen and D. Harwood, “A Comparative Study of Texture Measures with Classification Based on Feature Distributions,” *Pattern Recognition*, vol. 29, no. 1, pp. 51-59, Jan. 1996.

[16] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification (Second Edition)*, John Wiley & Sons, USA, 2000.

[17] A.264/AVC Software Coordniation, “JM Software,” ver. 12.2, [Online], Available: http://iphone.hhi.de/suehring/tml..