Modeling Noisy Texture for Component Based Video Coding

Fabian JÄGER[1], Johannes BALLÉ[1]

[1]Institut für Nachrichtentechnik
RWTH Aachen University
Aachen, Germany

{jaeger,balle}@ient.rwth-aachen.de

Abstract. *One of the problems in modern video coding is the processing of noisy textures. Established video coding algorithms perform reasonably well when applied to source material, which contains little additive noise or noisy textures. These components of images and videos are not well suited for decorrelation, most contemporary coding approaches are based on. In this paper a technique for efficient processing of noisy textures is introduced. As a preprocessing step, to split the source material into an "exact" structure component and a "statistical" noise component, a conservative image decomposition by means of a denoising algorithm is used. The structure component is then coded with conventional video coding algorithms as it is well suited for decorrelation techniques. For the noise component a statistical representation based on a customized, non-stationary ARX process is presented and evaluated. Additionally this paper describes the decomposition of the source video material into three components, which yields in even better coding results because of the characteristics of each component.*

Keywords

video coding, noise, texture, statistical model, autoregressive process, decomposition, denoising, multilayer

1. Introduction

Most modern video coding techniques are based on decorrelation. Per video frame this is achieved by representing the signal with linear combinations of basis functions followed by a quantization of the resulting coefficients according to the targeted quality prerequisites. Another decorrelation method used in digital video coding is spatial and temporal prediction of the video signal. Both approaches do not perform well on material containing noise or noisy texture as these components have statistical characteristics and therefore are not suited for conventional coding methods. To achieve transparent encoding of such signals higher bit rates are required.

Preceding studies of the human perception showed that exact representation of noise in images is not required. In fact, the human visual system cannot distinguish between two noise signals that are based on the same statistical process [4]. Ballé et al. presented an approach for modeling and coding noisy texture in natural images, which assumes that noisy texture in images can be represented by a statistical process [1] . He demonstrated that a non-stationary ARX process can describe noisy textures in natural images and therefore can lead to more efficient compression of this image component.

The concepts introduced in [1] form the foundation for the methods presented in this paper. Therefore a short overview of the methods Ballé introduced are given before their extension for the processing of digital video signals is discussed. Finally some exemplary results are presented followed by an outlook on further research topics in this area.

2. Modeling Noisy Texture in Images

In this section the concept of modeling noisy texture as a stochastic process is described. A critical prerequisite for the modeling is the decomposition of the input signal into the structure component and the noise component. In contrast to other approaches in this field, which use segmentation for the decomposition, denoising algorithms are used for this task. This decision is justified with the fact that denoising can be used without requiring perceptual measures or semantic analysis. For the subsequent modeling process the texture component should only contain structureless noise and can therefore be represented by a statistical model. For this sort of decomposition denoising seems to be the more appropriate choice.

2.1. Decomposition

As decomposition of the input image is a decisive part of the whole modeling method described in this paper, a recently proposed [2] technique, the Non-Local-Means (NLM) algorithm, appears to be a logical choice. It estimates the original uncorrupted pixel value by a weighted average of pixel values with similar neighborhoods. The estimated pixel value at position x in image u is then defined as

$$\hat{u}(x) = \frac{1}{C(x)} \sum_{y_i \in \mathcal{W}} w(x, y_i) \cdot u(y_i) \tag{1}$$

with $\mathcal{W}$ representing a search window given as a set of displacement vectors y_i. A schematic illustration of the algorithm is shown in Figure 1. The normalizing factor $C(x)$ makes sure that the sum of all weights $w(x, y_i)$ amounts to 1. The weights for each neighborhood are themselves defined as

$$w(x, y_i) = exp\left(-\frac{\| \mathbf{G}(u_{\mathcal{N}}(x) - u_{\mathcal{N}}(x + y_i))\|^2}{2h^2}\right) \tag{2}$$

with $\mathcal{N}$ defining the neighborhood to compare. The function $\mathbf{G}$ weights the pixel differences with a Gaussian mask, which is also normalized to make sure the overall signal energy is not modified. The strength of the denoising filter is parametrized with the parameter h. It has to be adjusted proportional to the variance of the given noise process. As this variance is not known for an image beforehand there have to be some more sophisticated approaches to estimate h. For now it is sufficient to perform a rather conservative decomposition by using a low enough value. After the decomposition with the NLM algorithm the input signal is split into a structure component $\hat{u}$ and a noise component $n = u - \hat{u}$.

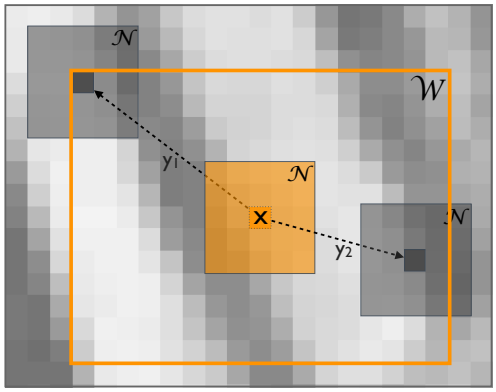

Fig. 1. Schematic illustration of the Non-Local-Means algorithm. In this example the difference between the neighborhood of the current pixel x and the two neighborhoods at the displacements y_1 and y_2 define the particular weights $w(x, y_i)$.

2.2. Noise Model

To model the noise component with a statistical process, some assumptions have to be made. First it can be assumed that the noise component has approximately zero mean, while it does not behave like a simple white Gaussian noise process. Pixels of noisy textures show dependencies on their surrounding, which is called noise color. In this approach this is modeled in form of linear dependencies. Linearity is generally not given in natural images, but it is a good approximation in terms of complexity and performance. A statistical process taking the noise color into account can be described with the following equation.

$$n(x) = A^T(x) \cdot n_{\mathcal{N}_A}(x) + \varepsilon_{\sigma(x)}(x) \tag{3}$$

If the neighborhood $\mathcal{N}_A$ is defined as a semi-causal window, the model is equivalent to a 2-dimensional autoregressive (AR) process with the parameter vector A and variance σ. To model instationarities A and σ are allowed to be position dependent. Unfortunately noisy texture mostly is not fully represented by the noise component due to imperfect decomposition. Furthermore textures consist of a regular part, which resided in the structure component and a noisy part. As a result texture features in the noise component also depend on information in the structure component. To model this effect the noise model has to be extended to also incorporate external information. This so-called ARX process is then defined as

$$n(x) = A^T(x) \cdot n_{\mathcal{N}_A}(x) + X^T(x) \cdot \hat{u}_{\mathcal{N}_X}(x) + \varepsilon_{\sigma(x)}(x). \tag{4}$$

The ARX process also models the given dependency of the noisy texture on the local neighborhood in the structure component $\hat{u}$. A and X represent the allowed linear dependencies between the pixel values of both components around pixel position x.

2.3. Quantization

The introduced statistical model for representing noisy textures alone does not yield better coding efficiency. Lossy image compression techniques achieve this goal primarily by quantization. [1] presents a simple way of quantizing the computed model parameters of the noise component, which is inspired by a block-based approach utilized in many image coding methods.

The basic idea is to split the image into blocks (partitioning) and compute the model parameters $A(i)$, $X(i)$ and $\sigma(i)$ for each block i. This technique presumes that these parameters are constant within each block, which is a sound assumption as textures normally don't change within a certain spatial area. To further improve the coding efficiency

for the model parameters it is possible to cluster the computed parameter vectors $A(i)$ and $X(i)$ into a given number of groups. The idea behind this improvement is that within a natural image there is a limited number of distinct types of textures and therefore each block can be assigned to one of these texture types (clusters). The mentioned technique corresponds to a vector quantization of the parameters $A(i)$ and $X(i)$ and can be realized by an iterative algorithm similar to the Generalized Lloyd Algorithm (GLA) for vector quantization of linear predictive coding (LPC) coefficients described in [8].
This quantization step results in an appropriate block mapping $B(i)$, which assigns all blocks to a specific cluster k resembling a certain type of texture. For each cluster there are two corresponding parameter vectors A_k and X_k modeling the linear dependencies of the particular texture type.

3. Modeling Texture in Video

The preceding section describes the general concept of a texture model and explains how this approach can improve image quality in low bitrate scenarios. To use this method for video signals some modifications have to be made. Due to the increased amount of redundancy within a video stream, the whole modeling procedure should incorporate neighboring frames to benefit from this information gain.

3.1. Decomposition

Buades et al. showed [3] that the Non-Local-Means denoising algorithm can be easily adapted to process video signals by comparing the current neighborhood not only with neighborhoods within the same frame but also within adjacent frames. This modification provides even better denoising results because of the increased number of observations for the estimation of the uncorrupted pixel value $\hat{u}(x)$. To take into account that in a video stream objects and their textures move between frames, the search area $\mathcal{W}$ has to be adapted accordingly.

3.2. Modeling and Quantization

The already mentioned gain of information that can be used to improve the texture model's coding efficiency is not only usable for the decomposition, but also for the rest of the described modeling procedure. Especially the quantization step can benefit from the information available in adjacent frames. The Generalized Lloyd Algorithm, which is used for the quantization of the model parameters $A(i)$ and $X(i)$, requires a preliminary initialization. For single images this initialization is done randomly, because there is no a priori knowledge available. Now with video signals this restriction does not hold any longer and the initialization of the clustering algorithm can either be done with the block mapping $B(i)$ of the previous frame or by using the parameters of the previous frame to compute an initial optimal mapping of blocks to texture clusters, which minimizes the prediction error $\sigma_k(i)$ for each block. The latter approach proves to be more reasonable as it allows for textures to move from one frame to the other, which is absolutely common in video streams. In contrast, if the clustering was initialized with the previous block mapping, then only those textures that did not move beyond their previous block borders would be initialized correctly.

With this initialization based on a priori knowledge, the clustering procedure converges after less iterations as shown in Figure 2. Another effect resulting from this simple modification is the stability of the cluster mapping $B(i)$ over time. As all computed model parameters have to be encoded in some way in a subsequent step, it was one design goal to make sure that these parameters can be encoded efficiently. With the presented initialization method for the clustering algorithm this can be achieved for the block mapping as presented in Figure 3. If the mapping is initialized randomly it is more likely that blocks are assigned to a different cluster from one frame to the other, because the parameters of each cluster differ in a way leading to another optimal mapping. With the described initialization this can be avoided, because it makes sure that textures occurring in consecutive frames get assigned to the same cluster.

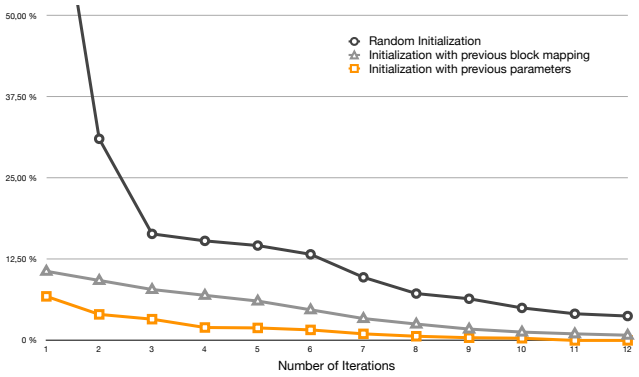

Fig. 2. Average percentage of blocks changing their assigned cluster in the last iteration. Those blocks have not found their optimal parameter cluster to minimize the estimation error σ. Initialization with a priori knowledge yields quicker convergency of clustering process.

3.3. Three-Component Decomposition

Throughout the analysis of video signals, their decomposition and modeling, it became obvious that there are two different types of noise in a video that have to be treated differently. One the one hand there is noisy texture, which is highly correlated in the spatial and temporal domain. This is the sort of noise the texture model is designed for. In addition there is another noise type, which can be observed only

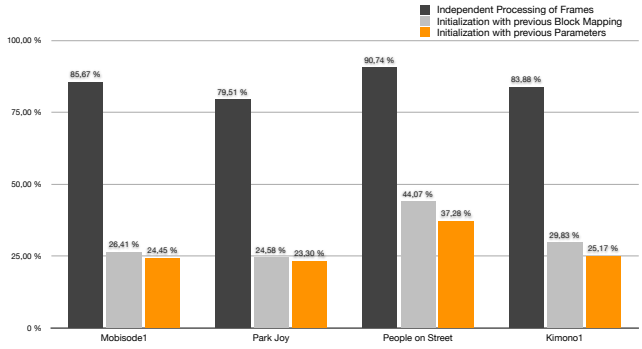

Fig. 3. Average percentage of blocks changing their assigned cluster from one frame to the other. These changes in the block mapping cannot be encoded with simple differential approaches and could increase the required bit-rate for the encoding of the texture model.

in video signals, because it is primarily characterized by its temporal uncorrelateness. Because of the characteristics of the two types of noise it appears obvious to decompose the input signal into three distinct components.
This paper describes a modification of the Non-Local-Means denoising algorithm, which provides the required three component decomposition. This is achieved with the assumption that noisy texture can be found in adjacent frames with a very high probability. Therefore only the best matches in neighboring frames are incorporated for the estimation of $\hat{u}(x)$ and there is no explicit weighting of those best matches. By subtracting $\hat{u}(x)$ from the original frame the uncorrelated noise component can be computed. Performing the original NLM filtering on $\hat{u}(x)$ yields the decomposition into structure and texture components.

3.4. Noise Modeling

With the introduced decomposition of the input video signal into the three components structure, texture and noise the previously described model has to be adapted to the new circumstances. While the structure component can still be encoded by conventional means, the texture component appears to be much more stable over time as the temporally uncorrelated noise is removed. The presented model for noisy texture performs much better on this component. Through the prior removal of the image noise the observations needed for a sufficient accurate estimation of the uncorrupted signal become much more reliable.

The newly described noise component is characterized by a random variation of brightness in natural images generated by the sensor and circuitry of a scanner or digital camera. Furthermore it can also contain artifacts introduced by film grain or artistic effects by the film maker. Analysis of the component's characteristics showed that it can be modeled by a statistical model similar to the one used for the texture component.

The foundation of the noise model is another autoregressive (AR) model, which is able to describe dependencies between neighboring pixels. These dependencies can be assumed to be stationary as they result primarily from the technical conditions when taking the video and therefore impact the signal as a whole. An initial draft of the noise model can thus be described by the following equation.

$$n(x) = A^T \cdot n_{\mathcal{N}_A}(x) + \varepsilon_{\sigma(x)}(x) \tag{5}$$

The parameter vector A does not need to be dependent on the position x due to the stationarity of the signal. In comparison the the model for noisy texture there does not need to be a model parameter X, which described the influence of the regular component of a texture residing in the structure component. As the noise component is not correlated with the image content, the describing model can be a simpler AR model. While the parameter A, which describes the noise characteristics, can be modeled stationary, the intensity (variance) of the noise signal is still position dependent. Boie and Cox [6] proved that the variance of noise produced by sensors and circuitry of digital cameras mostly depends on the intensity of the image itself in the surrounding of the current pixel. Therefore the model parameter $\sigma(x)$ itself can be modeled by incorporating the intensity of the structure component. This is easily done by generating a function $\sigma(x) = f(I(x))$, where $I(x)$ is the intensity of the image at position x. The resulting model for the noise component can then be described by the following equation.

$$n(x) = A^T \cdot n_{\mathcal{N}_A}(x) + \varepsilon_{\sigma(I)}(x) \tag{6}$$

The function $\sigma(x) = f(I(x))$ is computed once for every frame and can be approximated by a polynomial. This reduces the amount of data per frame for the noise model to a single parameter A and some coefficients for the intensity-variance function $\sigma(I)$. An exemplary function and its approximating polynomial is plotted in Figure 4.

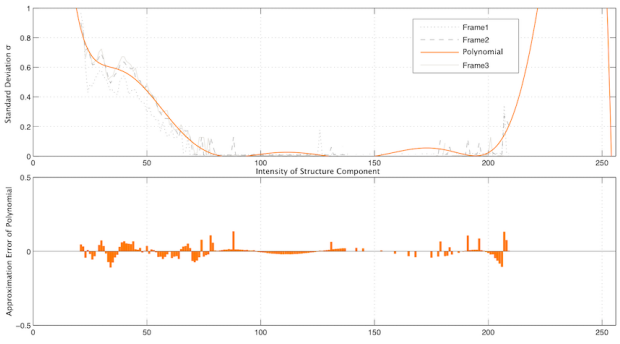

Fig. 4. Example of the approximation of the intensity-variance function with a polynomial of degree 9. The frame does not contain intensities less than 20 and higher than 210 and therefore these regions can be ignored.

Due to the design of the NLM denoising filter there are some areas in a frame where it is not possible to de-

compose the original signal. As a result these areas are empty in the texture and the noise component although they should show characteristics similar to the rest of the image. To make sure that these areas in the noise component are modeled correctly, they have to be excluded when generating the intensity-variance function. It would be possible to add a binary mask for areas, which should not be modeled by the noise model, but that would add a lot of additional data to be transmitted. A simple but efficient solution is to incorporate not only the structure component into the intensity-variance function, but also the texture component's variance. As mentioned before, areas that cannot be decomposed are approximately zero in both, the texture and the noise component. Taking this into account the function describing the variance in the noise component can be extended to $\sigma(x) = f(I(x), \sigma_T(x))$, where $\sigma_T(x)$ corresponds to the variance at position x in the texture component. The result of synthesized noise using the presented noise model are shown in Figure 7.

4. Coder Integration

During the development of the presented models their integration into an existing coding architecture was always a crucial design aspect. As already mentioned before, the structure component is intended to be encoded conventionally. The other two components have to be estimated by the corresponding model and afterwards transmitted independently. An efficient entropy encoding scheme for the produced model data is not yet included in this early coding architecture.

Conventional video coders, like the JM reference software, process a video frame block-wise. These blocks can either be encoded "intra" or "inter". Inter-coded blocks use motion vector information referencing adjacent frames utilizing temporal redundancy to increase coding efficiency. The motion estimation should be done on both, the structure and the texture component to ensure that the estimated motion vectors are sufficiently accurate. Depending on the decision of the encoder how to encode the structure component, the texture component is handled accordingly. If the structure should be encoded "intra", the texture is modeled with the presented model for noisy textures. Inter-encoded blocks use the same motion vectors as the structure component. Hence those blocks in the texture component are just shifted blocks from previous frames. As a result inter-encoded blocks in the texture component are encoded at zero cost as the motion vectors are already encoded in the structure component. Considering that about 80-90% of the blocks in a video signal are generally encoded in "inter"-mode, this yields a tremendous reduction in data to be transmitted.

Another important aspect of the presented integration of the texture and noise models into an existing coding architecture is the fact that most of the required computation is done at the encoder. The decoder just needs to do some simple matrix multiplications for the AR process and a final addition of all three components. This concept is illustrated in Figure 5 for the encoder and in Figure 6 for the decoding unit.

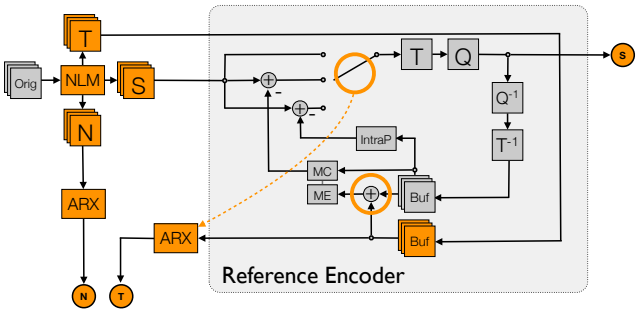

Fig. 5. Integration of the presented models for the texture and noise components into a existing coding architecture, like the JM. Highlighted in orange are those parts that have to be modified.

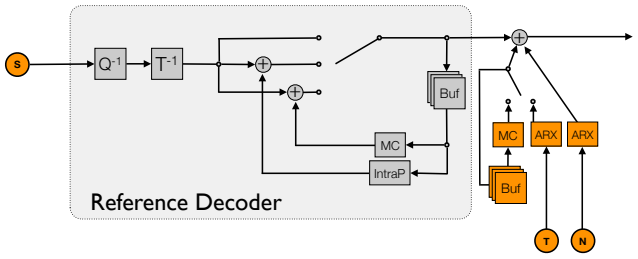

Fig. 6. Schematic illustration of the video decoder. The required modifications (highlighted in orange) have no impact on the original decoder implementation as the texture and noise components can be computed separately and added afterwards.

5. Experimental Results

Figure 7 illustrates exemplary results of a complete encoding-decoding process using the presented integration of the two models into the existing JM coder software. The structure component is encoded conventionally. In the shown frame about 86% of the blocks are encoded in inter mode, which means that estimated motion vectors are used to predict the blocks based on their position in previous frames. These inter coded blocks are treated similarly in the texture component and only the remaining intra-coded blocks use the presented texture model. The second row of Figure 7 shows that there are no visible artifacts and that the characteristics of the noisy texture can be described by the presented statistical model. The same holds for the noise component, where only one type of noise is to be modeled for the whole frame. The variance of the noise process in this component is defined by the approximated intensity-variance function, which works very well in most scenes as can be seen in the last row of Figure 7.

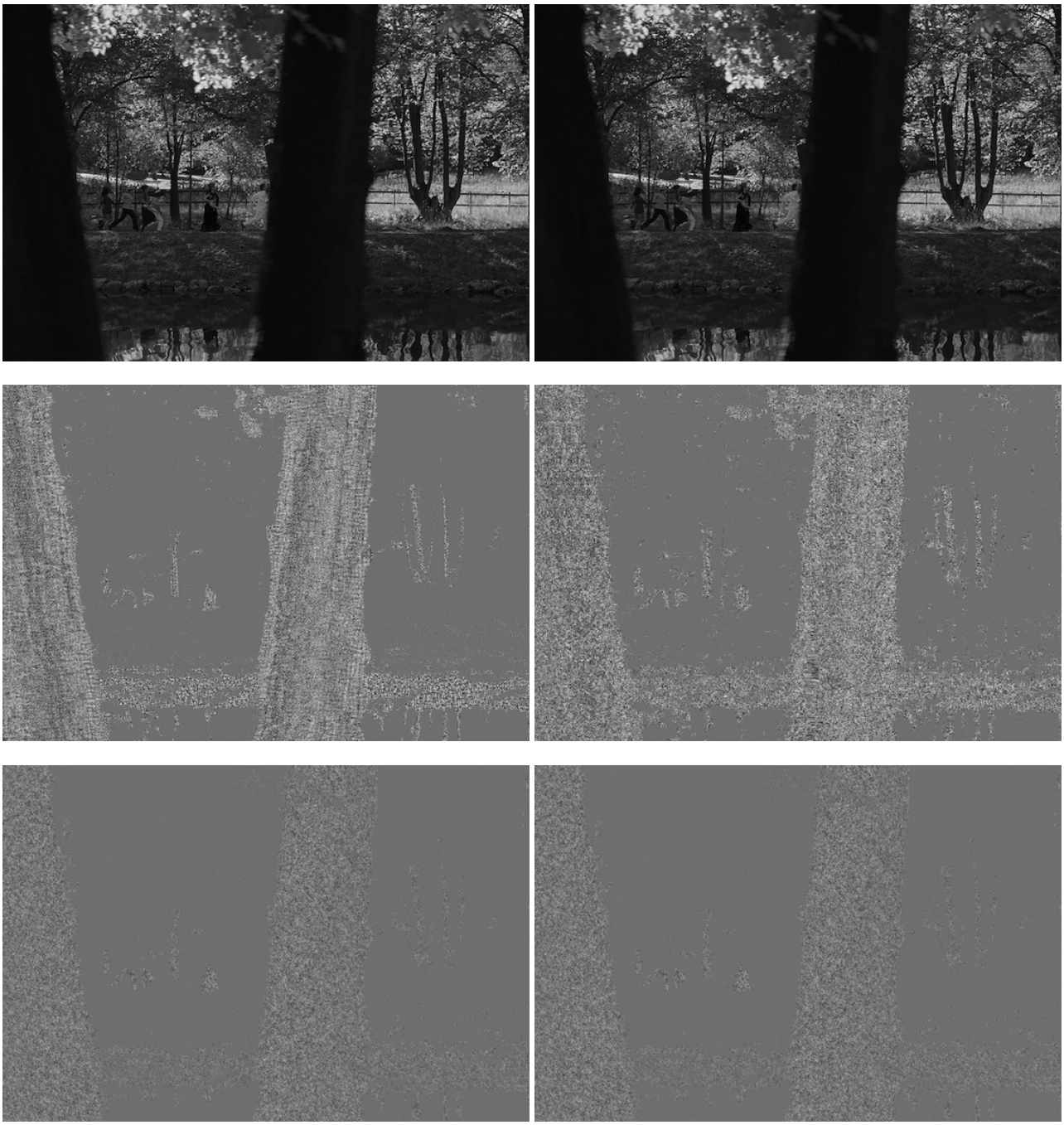

Fig.7. Coding results for structure, texture and noise component (from top to bottom) of a video frame. On the left side there is the uncompressed and on the right the completely encoded and afterwards decoded component using the presented integration of the models into the coder. Texture and noise are amplified by a factor of 10 for

6. Conclusion and Prospect

In this paper a new approach for modeling noisy texture and also image noise by means of specially adapted autoregressive models is presented. The two models are capable of modeling specific characteristics of the given component of the input video. This is achieved by initially decomposing the video into three components with a customized version of the Non-Local-Means algorithm. The described models for noisy textures and noise signals may provide opportunities to improve coding efficiency for low and medium bitrate scenarios.

Further research is being directed towards an efficient entropy encoding scheme for the produced model parameters. As the number of parameters to be encoded per frame is already reduced through partitioning and quantization, an appropriate coding system should yield very good compression rates for the texture model. Although the incorpora-

tion of motion vector information into the modeling of the texture component reduces the amount of data enormously, there are still some problems to solve in this field. Without a residual signal for the texture component, transformations within the texture from one frame to the other cannot be compensated and lead to visible artifacts as blocks are simply shifted based on the motion vectors for the structure component. There has to be a criteria for the texture component that decides when motion compensation is sufficient and in what cases it is more reasonable to re-synthesize the texture.

Acknowledgements

Research described in this paper was supervised by Prof. Dr.-Ing. J.R. Ohm and Dipl.-Ing. Johannes Ballé, Institut für Nachrichtentechnik, RWTH Aachen University.

References

[1] BALLÉ, J., WIEN, M. A Quantization Scheme for Modeling and Coding of Noisy Textures in Natural Images. In *Proc. of IASTED Conference on Signal and Image Pro- cessing SIP*, 2009.

[2] BUADES, A., COLL, B., MOREL, J.M., On image denoising methods, *tech. rep., Centre de Mathmatiques et Leurs Applications*, Cachan, France, 2004.

[3] BUADES, A., COLL, B., MOREL, J.M., Image and movie denoising by nonlocal means, *IJCV*, 2006.

[4] LANDY, M., GRAHAM, N.. Visual perception of texture. *The visual neurosciences*, Jan 2004.

[5] BIEDERMAN, I. Recognition-by-components: A theory of human image understanding. *Psychological review*, Jan 1987.

[6] BOIE, R., COX, I.J., An analysis of camera noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 671-674, 1992.

[7] MARZETTA, T. Two-dimensional linear prediction: autocorrelation arrays, minimum- phase prediction error filters, and reflection coefficient arrays. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(6):725733, 1980.

[8] GERSHO, A., GRAY, R.M., *Vector Quantization and Signal Compression*, Kluwer, 1992.

[9] NDJIKI-NYA, P., HINZ, T., WIEGAND, T. ,Generic and robust video coding with texture analysis and synthesis, in *Proc. of IEEE International Conference on Multimedia and Expo ICME*, pp. 14471450, July 2007.

[10] WEI, L.-Y., LEVOY, M. Fast texture synthesis using tree-structured vector quantization, in *Proc. of International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*, pp. 479488, July 2000.

[11] SMEULDERS, A.W.M., WORRING, M., SANTINI, S., GUPTA, A., JAIN, R., Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 13491380, Dec. 2000.

[12] MARAGOS, P.A., SCHAFER, R.W., MERSERAU, R.M., Two-dimensional linear prediction and its application to adaptive predictive coding of images, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, pp. 12131229, Dec. 1984.

[13] MARPE, D., SCHWARZ, H., WIEGAND, T., Context- based adaptive binary arithmetic coding in the H.264/AVC video compression standard, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 620636, July 2003.

About Authors...

Fabian JÄGER

was born in Berlin, Germany in 1984. He studied Computer Engineering at RWTH Aachen University. After his graduation in 2009 he decided to continue researching as a Ph.D. student at the Institut für Nachrichtentechnik at RWTH Aachen University. Fabian's research is focusing on algorithms and applications in 3D video coding.

Johannes BALLÉ

was born in Dortmund, Germany in 1980. He received the Dipl.-Ing. degree in Computer Engineering from RWTH Aachen University, Aachen, Germany, in 2007. Currently, he is pursuing the Ph.D. degree at the Institut für Nachrichtentechnik, RWTH Aachen University, focusing on image processing, signal theory, and data compression.