A QUANTIZATION SCHEME FOR MODELING AND CODING OF NOISY TEXTURE IN NATURAL IMAGES

Johannes Ballé and Mathias Wien
Institut für Nachrichtentechnik
RWTH Aachen University
Aachen, Germany
email: {balle,wien}@ient.rwth-aachen.de

ABSTRACT
Noisy textures in images pose a particular problem to image compression as they are not well suited for decorrelation. Established image compression methods thus need to be driven at relatively high bit rates in order to achieve visually transparent compression of this type of signal. At the same time, it is an intuitive observation that pixel-wise exactness of the reconstruction of noisy texture is not necessary in order for it to be transparent. Thus, it is desirable to find a more efficient representation of such texture. In this paper, we suggest the use of a denoising algorithm to achieve a conservative decomposition of the image signal into an "exact" structure component and a "statistical" noise component. We develop a noise model which allows us to represent and parameterize noisy textures as a non-stationary ARX process. Furthermore, we present an estimation algorithm for the model parameters and evaluate results on a number of well-known images.

KEY WORDS
image coding, noise modeling, autoregressive process, compression by synthesis, multi-layer image model

1 Introduction

Decorrelation is the basis for the vast majority of image coding techniques. Most commonly, the signal is represented by a linear combination of basis functions of a discrete transform and then quantized according to the quality requirements. Another common method is spatial or (in the case of video) temporal prediction. Both of these methods do not work efficiently with noisy signals, as they are designed to exploit statistical dependencies between pixel values. Hence, transparent coding typically requires a high bit rate for this type of signal.

On the other hand, it is intuitively clear that for visually plausible results, the exact representation of noise in images is not a requirement. Generally, the human visual system does not make a difference between two noise signals, given that they are simply two realizations of the same underlying statistical process. We thus expect to achieve greater compression efficiency by relaxing the restrictions imposed by the PSNR measure.

Recent work [1, 2] has explored ways of incorporating exemplar-based texture synthesis algorithms [3, 4] into image coding frameworks. This is basically achieved by identifying segments in the original signal which are deemed to be reconstructed by a synthesis algorithm as opposed to conventional coding methods. This is an effective way to loosen the restrictions of the PSNR measure. However, such schemes make it necessary to evaluate synthesis results using perceptual measures, which may be unreliable due to the semantic gap [5]. In this paper, we consider a different, more conservative approach, which may be practical without requiring perceptual measures or semantic analysis. Instead of segmentation, we consider *decomposition* as a means of distinguishing between image content that needs to be reconstructed with pixel fidelity and content which may be transparently reconstructed by means of a statistical model. Another important difference is that our model captures different types of texture than exemplar-based synthesis methods, as these are not well suited for noisy signals[1].

For a moment, let us assume that noisy textures in natural images could be characterized by a stationary white Gaussian noise process. It would then be reasonable to assume that a well-chosen denoising algorithm ideally succeeds to remove all noisy texture from the image while preserving all of image structure. We may then refer to the denoised version of an image as the "noiseless" *structure component* and to the difference image as the *noise component*. Note that the noise component would then only contain structureless noise, which may be represented by a statistical model.

In general, natural noisy textures do not satisfy the constraints of an additive, white, stationary Gaussian random process. Firstly, they are non-stationary, as textures tend to belong to objects and may thus apply only to certain segments of an image. Secondly, they are not additive, as for instance, film grain (which may for artistic reasons be a desired effect and is consequently treated as an example of a noisy texture present in natural images) is known to be dependent on the signal. And thirdly, they are not white, as noisy textures often contain repetitive elements. However, we have found that for coding purposes, using a recent de-

[1] The two mentioned synthesis methods do not model *innovation* such as the Gaussian random term in an AR process, but merely seek to minimize an error criterion involving adjacent image patches.

noising algorithm such as the Non-Local Means algorithm (NLM) [6] is a justifiable approximation. We give a more detailed explanation in the following sections.

Let us now reconsider our initial problem. After decomposition of the image, we represent the structure component by means of conventional coding techniques. The noise component, however, needs to be represented by a statistical model. This model must capture the visual characteristics of the component while having a small number of parameters. In this paper, we develop a noise model and quantization scheme that attempts to solve this problem.

The rest of this paper is organized as follows. In the next section, we shortly review the NLM algorithm used for decomposition. We proceed with a mathematical definition of our noise model in Section 3. Then, we present details on our quantization scheme and estimation algorithm (Section 4). Finally, we present some experimental results (Section 5) and conclude the paper with Section 6.

2 Decomposition

A recently proposed and yet well-adopted denoising method is the Non-Local Means (NLM) algorithm [6]. It estimates the pixel values of the original, uncorrupted image based on a weighted average of pixel values with similar neighborhoods. For each pixel position x in the image u, the estimated pixel value is defined as

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

with $\mathcal{W}$ representing a search window given as a set of displacement vectors y and a normalizing factor $C(x) = \sum_{y \in \mathcal{W}} w(x,y)$ chosen such that the sum of weights amounts to 1. The weights are then defined by

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

with $u_{\mathcal{N}}(x)$ representing the vectorized pixel values of a neighborhood $\mathcal{N}$ centered at position x and G a vector of weights chosen such that the positions in $\mathcal{N}$ are weighted according to a Gaussian mask normalized to a sum of 1. h acts as a filtering parameter. Throughout the rest of this paper, we shall refer to $\hat{u}$ as the structure component and to $n = u - \hat{u}$ as the noise component.

It was demonstrated in [6] that the NLM algorithm quite impressively outperforms many other well-known denoising methods such as bilateral filtering [7] and diffusion-like processes [8]. Particularly, it was observed that the noise component (referred to as "method noise" in [6]) contains less image structure than with the other evaluated methods if the filtering parameter h is chosen carefully.

Note that the objective of the denoising algorithm is not exactly a decomposition of the image as we seek it. Particularly, the algorithm was designed with the assumption that the noise component is approximately stationary, white, and Gaussian. As we need to adjust h proportionally to the variance of the noise process, which generally is non-stationary, we may not be able to find the optimal value for it. However, it is possible to perform a *conservative* decomposition by setting it to a low enough value. Empirical results show that it is practical to use the NLM filter in this way, although further research is needed to automate the selection of h and to exploit non-stationary texture.

The advantages of the non-local approach are as follows. Firstly, it delivers competitive denoising results and preserves "regular" texture and repetitive structure. Thus, texture containing repetitive elements as well as noise is actually split up between both components, leaving us with a nearly-white texture in the noise component that can actually be modeled by the process described in Section 3. Repetitive structure is better represented by conventional (predictive) coding techniques such as [9]. Secondly, we may use h to directly control the balance of signal energy across both components, allowing the method to be fully adjusted to the desired amount of "pixel-exactness". Additionally, it is easy to implement the algorithm and take advantage of its parallel structure [10]. Thus, we currently consider the NLM algorithm the most promising candidate for our experiments.

3 Noise Model

As we may assume that the noise component has approximately zero mean, the simplest approach would be to represent the entire component using a single parameter, the standard deviation σ of a white, zero-mean Gaussian noise process $\varepsilon_\sigma(x)$.

This process, however, does not capture non-stationarity and noise color. Thus, we extend the model by allowing linear dependencies between neighboring pixels. Linearity can generally not be assumed here, but it is a good choice in terms of complexity and performance with respect to noisy signals. We may describe such a process using the following recursive 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 restricted to the non-symmetric half plane (NSHP), our model is equivalent to a 2-dimensional autoregressive (AR) process with the parameter vector A and variance σ^2. Here, we also allow A and σ to be dependent on x to model instationarities.

We observe that noisy textures are often not fully represented in the noise component. Moreover, textures are split into a "regular" part and a "noisy" part, such that the spatial position of texture features in the noise component depends on texture features in the structure component[2]. In order to adapt our model to this effect, we extend it further to a so-called ARX process, given by

[2] This may vary with different decomposition processes.

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

In this model, the distribution of noise is also dependent on the local neighborhood in the structure component $\hat{u}$ (see previous section). In analogy to A, X represents linear dependencies between the pixel values of both components around pixel position x. As we may freely assume that the structure component is already decoded before decoding of the noise component takes place, $\mathcal{N}_X$ is not restricted by causality.

4 Quantization Scheme

Lossy image compression schemes primarily use quantization as a means of achieving coding efficiency. In this paper, we explore a simple but efficient quantization scheme of the model parameters inspired by the block-based approach taken by many popular image coding methods.

Given a digital image of $M \times N$ pixels x in the domain $\mathcal{D} = \{0, \ldots, M-1\} \times \{0, \ldots, N-1\}$, we restrict A, X, and σ to be constant for each rectangular block of $b \times b$ pixels. Thus,

$$A(x) = A(i) \quad \text{for } i = \left\lfloor \frac{1}{b} \cdot x \right\rfloor \quad (5)$$

and likewise for $X(i)$ and $\sigma(i)$. We need to estimate these parameters for each of the $\frac{MN}{b^2}$ blocks. This is achieved by standard least-squares optimization methods.

In order to exploit the fact that the information about texture is mainly contained in the linear model coefficients $A(i)$ and $X(i)$, we further demand that each $A(i)$ and $X(i)$ take the value of one of a number K of representative model parameters A_k and X_k, with $k \in \{1, \ldots, K\}$. This has three consequences for our model. Firstly, given an appropriate block mapping, we may now estimate A_k and X_k taking advantage of a greater number of observations, thus decreasing the variance of our parameter estimators. Secondly, by doing this, we assume that the image contains only a limited number of different *types* of noise color. This corresponds to the intuitive idea that we may only have a distinct number of textured objects in a scene. And finally, we reduce the number of needed parameters for the model. This scheme corresponds to a vector quantization of $A(i)$ and $X(i)$.

What remains is finding an appropriate mapping of image blocks to block clusters $B(i) \in \{1, \ldots, K\}$, as well as the optimal parameters A_k and X_k for each block cluster and $\sigma(i)$ for each block. We employ an iterative algorithm similar to the Generalized Lloyd Algorithm (GLA) for vector quantization of linear predictive coding (LPC) coefficients outlined in [11].

Let us first consider the estimation algorithm for the ARX parameters. Using a simple extension of the "covariance method" [12], we obtain estimates for A_k and X_k by means of building the empirical correlation matrix for an entire cluster of image blocks. This corresponds to the first step of the iteration in Algorithm 1. After solving the linear equation system, we estimate $\sigma_k^2(i)$ as the mean squared prediction error for each of these K sets of coefficients and each block. This is either achieved by explicitly calculating the prediction error signal and estimating its power, or – equivalently – by using the Itakura–Saito distance measure [13], which straight-forwardly extends to 2D. The latter is computationally more efficient for large block sizes.

Given $\sigma_k(i)$, we can select $B(i)$ such that $\sigma(i)$ is minimized for each block, which concludes the iteration of the algorithm. However, in the case of small block sizes such as 4×4, the estimator of $\sigma_k(i)$ tends to be too unreliable for the algorithm to converge.

Noting that texture characteristics often change slowly across image space, we allow the selection to also depend on the local neighbors of i. This can be achieved conveniently by completely evaluating $\sigma_k(i)$ and then basing the selection of $B(i)$ on the result of a convolution of $\sigma_k^2(i)$ with a Gaussian kernel G_B instead of $\sigma_k^2(i)$ itself. With this modification in place, the algorithm needs significantly less than 10 iterations to get very close to convergence on our test material. The "smoothness" of the clustering result can be adjusted by varying the variance of the kernel.

Algorithm 1 VQ parameter estimation

```
Initialize B(i) randomly
m ← 0
repeat
  for k = 1 to K do
    estimate A_k, X_k
  end for
  for i ∈ D do
    estimate σ_k(i)
  end for
  m ← m + 1
  if m = number of iterations then
    break
  end if
  for i ∈ D do
    B(i) ← arg min_k (σ_k^2(i) * G_B)(i)
  end for
until false
```

5 Experimental Results

Figures 1 and 2 illustrate experimental results for the well-known images "Lena" and "Baboon" (gray-scale versions), respectively. The upper left image represents the structure component (i.e. the denoised version of the original image). Note that we did not use the most conservative setting of the NLM algorithm to demonstrate the effects of parameter selection. The upper right image represents the noise component. Due to the stationarity assumption, some low-

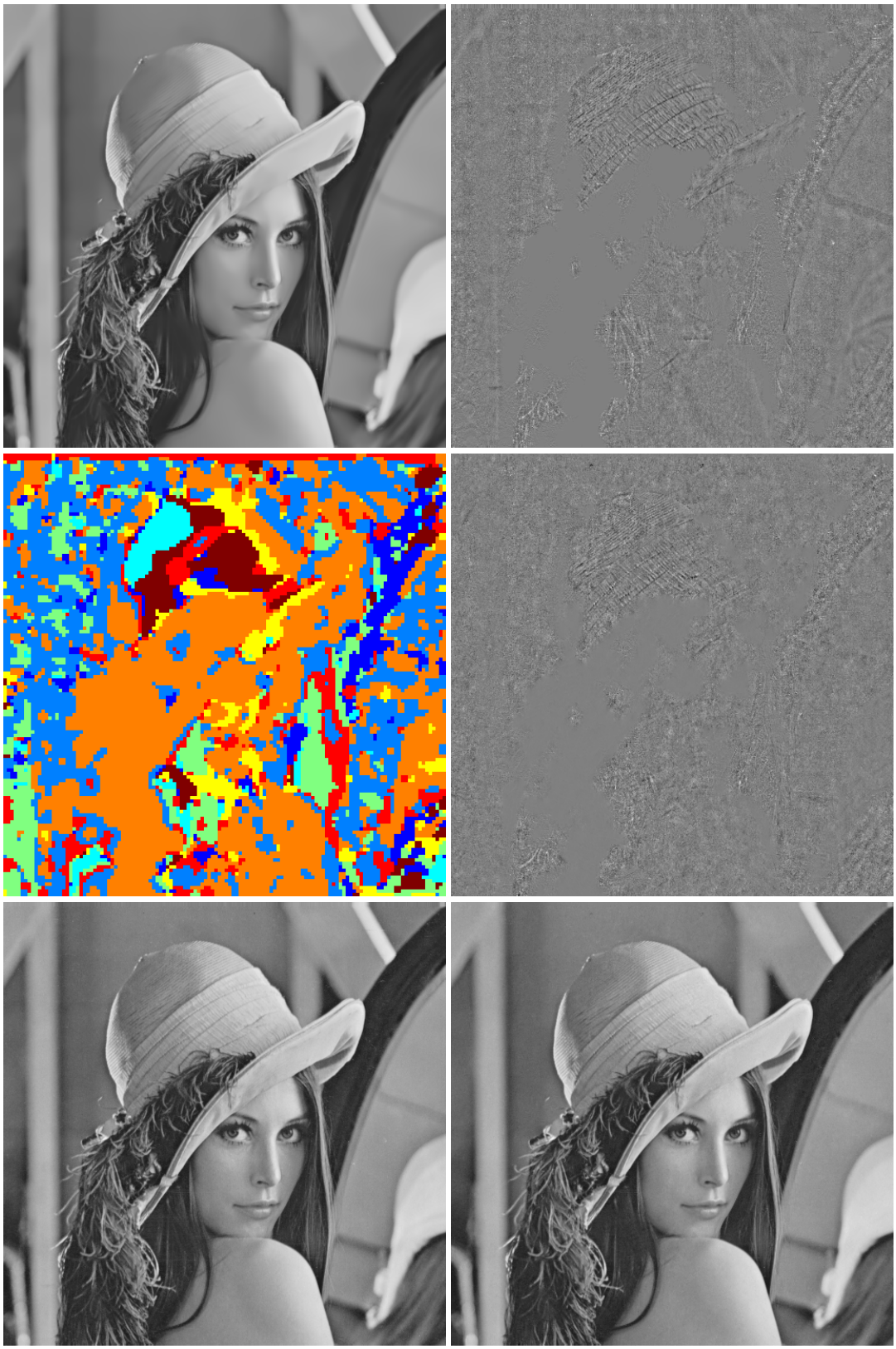

Figure 1. Results for image “Lena”. Top left: structure component; top right: noise component (amplified by factor 4); middle left: visualization of block mapping; middle right: reconstructed noise component (amplified by factor 4); bottom left: original; bottom right: reconstruction (addition of structure component and reconstructed noise component). Parameters: $\mathcal{N} : 17 \times 17$, $\mathcal{W} : 25 \times 25$, $h = 4$, $\mathcal{N}_A : 17 \times 9$ (NSHP mask), $\mathcal{N}_X : 5 \times 5$, $b = 4$, $k = 8$

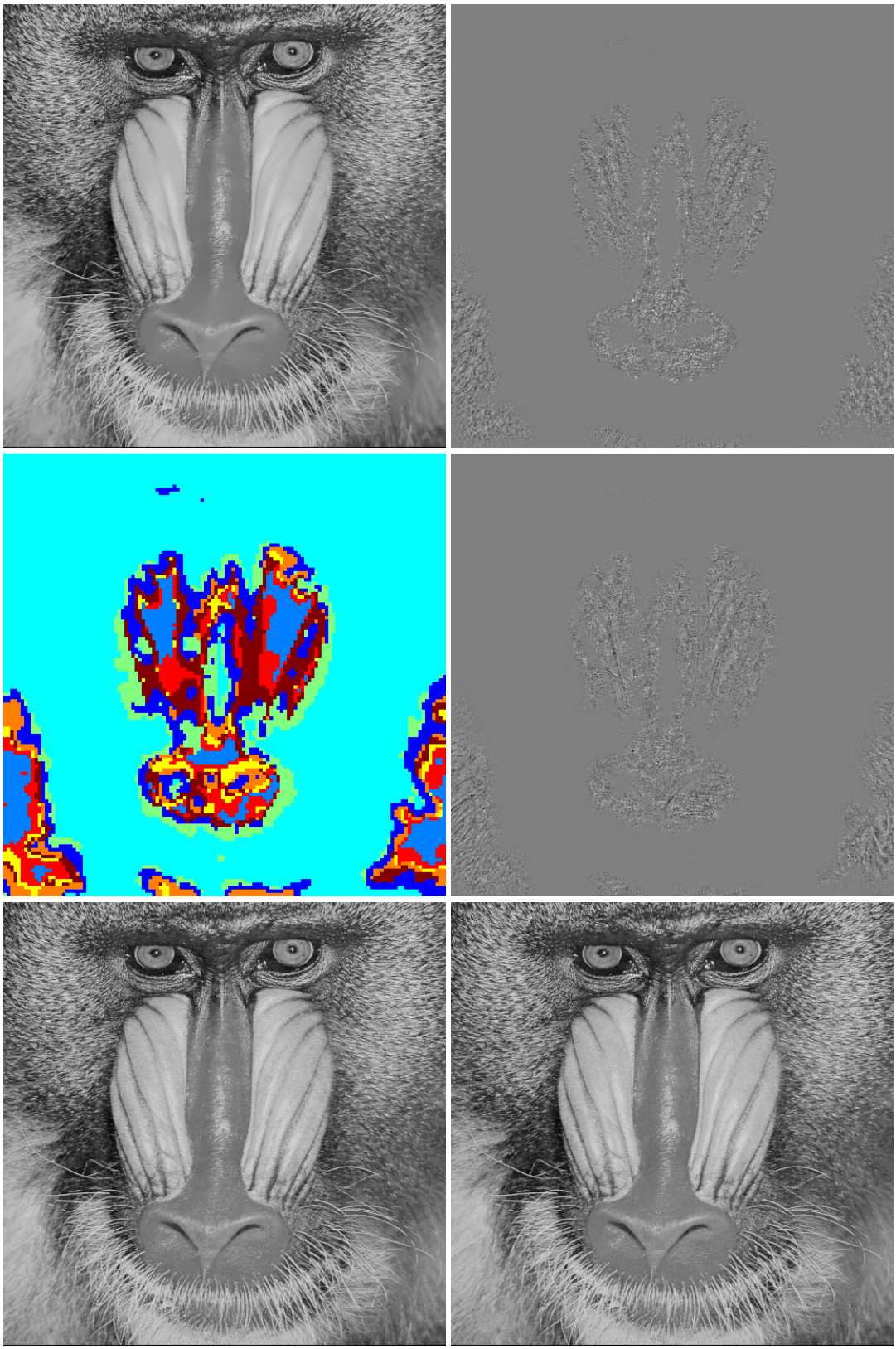

Figure 2. Results for image “Baboon”. Top left: structure component; top right: noise component (amplified by factor 4); middle left: visualization of block mapping; middle right: reconstructed noise component (amplified by factor 4); bottom left: original; bottom right: reconstruction (addition of structure component and reconstructed noise component). Parameters: $\mathcal{N} : 17 \times 17$, $\mathcal{W} : 25 \times 25$, $h = 4$, $\mathcal{N}_A : 17 \times 9$ (NSHP mask), $\mathcal{N}_X : 5 \times 5$, $b = 4$, $k = 8$

Figure 3. Coding results for image "Kodak 04" (cropped). Left: JPEG2000 reconstruction. Right: reconstruction using our scheme. Parameters: $\mathcal{N} : 17 \times 17$, $\mathcal{W} : 25 \times 25$, $h = 2$, $\mathcal{N}_A : 11 \times 6$ (NSHP mask), $\mathcal{N}_X : 1 \times 1$, $b = 8$, $k = 4$. Total size of the bit stream is 38.3 kibytes for both figures. For the right figure, the model parameters take 8% of the total bit rate.

contrast edges are removed (for instance, in Figure 1, the frame of the mirror in the background of the image), while some noisy texture with high contrast is not (Figure 2, the fur right and left of the mandrill's face). Consequently, some content in the noise component cannot sufficiently be modeled by the ARX process, while some noise remains in the structure component that could be. An improvement of this behaviour could be achieved by applying local estimation of noise variance, thus adapting h to the local texture.

Nonetheless, it is important to note that noisy texture that was correctly decomposed is successfully captured by the ARX model. This is evident in the middle right image, which depicts the reconstruction of the noise component by means of the estimated parameters A_k, X_k, $\sigma(i)$, and $B(i)$. The middle left image is a visualization of $B(i)$, where each color represents a different block cluster. In the bottom row, we directly compare the original image (left) to the reconstruction (addition of structure component and reconstructed noise component). Although the presented model does not guarantee stability of the synthesis filter, we observed stability problems only in rare cases. Improved models are being investigated to guarantee stability.

Note that the results shown in Figures 1 and 2 do not include entropy coding or scalar quantization of A_k, X_k, and $\sigma(i)$. They simply demonstrate that the proposed vector quantization scheme is a valid model for noisy texture. Figure 3 illustrates preliminary coding results using the Context-Adaptive Binary Arithmetic Coding (CABAC) framework [14]. In this (ad-hoc) entropy coding scheme, A_k, X_k, and $\sigma(i)$ are coded using uniform quantization and an Exp-Golomb encoding, and $B(i)$ is encoded using fixed-length binary integers and context modeling. The

structure component, as well as the reference version of the image, is encoded using JPEG2000 [15]. The image is taken from the well-known Kodak set, available at [16]. The total bit rate of both versions of the image is 0.8 bpp. Note that while the skin texture appears oversmoothed in the JPEG2000 reconstruction, our scheme re-synthesizes a plausible texture in these regions.

The computationally most demanding part of the scheme is the decomposition algorithm, followed by the parameter estimation. Compared to the application of exemplar-based texture synthesis methods, little additional load is imposed on a decoder using our scheme, which may be a practical advantage. On the other hand, the structure of the NLM algorithm allows for easy parallelization [10], which may be implemented to reduce computation on the encoder side.

6 Conclusion

We present a novel scheme for modeling of noisy texture in still images with regard to image coding. Our model induces little computational load on the decoder, while being capable of capturing noisy texture present in natural images with a small number of parameters. Thus, our scheme may yield an efficient method of improving compression of noisy texture for low to medium bitrate coding.

Ongoing research is being directed towards two main subjects. Firstly, the extension to color and moving images is being studied, as the visual quality of the latter particularly suffers from the effects of camera noise and film grain. Secondly, we intend to validate alternatives and improvements to the denoising process, which may allow a less conservative decomposition and hence subject a greater proportion of noisy image texture to the modeling scheme.

References

[1] P. Ndjiki-Nya, T. Hinz, and T. Wiegand, "Generic and robust video coding with texture analysis and synthesis," in *Proc. of IEEE International Conference on Multimedia and Expo ICME*, pp. 1447–1450, July 2007.

[2] J. Byrne, S. Ierodiaconou, D. Bull, D. Redmill, and P. Hill, "Unsupervised image compression-by-synthesis within a JPEG framework," in *Proc. of IEEE International Conference on Image Processing ICIP*, 2008.

[3] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proc. of International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*, pp. 479–488, July 2000.

[4] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," in *Proc. of International Conference on Computer Graphics and Interactive Techniques SIGGRAPH*, pp. 277–286, July 2003.

[5] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1349–1380, Dec. 2000.

[6] A. Buades, B. Coll, and J. M. Morel, "On image denoising methods," tech. rep., Centre de Mathématiques et Leurs Applications, Cachan, France, 2004.

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. of IEEE International Conference on Computer Vision ICCV*, pp. 839–846, Jan. 1998.

[8] J. Weickert, *Anisotropic Diffusion in Image Processing*. PhD thesis, Universität Kaiserslautern, Kaiserslautern, Germany, Jan. 1996.

[9] J. Ballé and M. Wien, "Extended texture prediction for H.264/AVC intra coding," in *Proc. of IEEE International Conference on Image Processing ICIP*, vol. 6, pp. 93–96, 2007.

[10] A. Kharlamov and V. Podlozhnyuk, "Image denoising," tech. rep., NVIDIA, Inc., June 2007.

[11] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer, 1992.

[12] P. A. Maragos, R. W. Schafer, and R. M. Mersereau, "Two-dimensional linear prediction and its application to adaptive predictive coding of images," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, pp. 1213–1229, Dec. 1984.

[13] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23, pp. 67–72, Feb. 1975.

[14] D. Marpe, H. Schwarz, and T. Wiegand, "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. 620–636, July 2003.

[15] *ITU-T Rec. T.800 & ISO/IEC 15444-1: JPEG 2000 Image Coding System: Core Coding System*.

[16] "CIPR still images: Kodak." Downloaded from http://www.cipr.rpi.edu/resource/stills/kodak.html, June 2009.