IMPROVED ENTROPY CODING FOR COMPONENT-BASED IMAGE CODING

*Christian Feldmann, Johannes Ballé**

Institut für Nachrichtentechnik
RWTH Aachen University
Aachen, Germany

ABSTRACT

In this paper, we improve on our previous work regarding component-based image coding, a hybrid transform-based/perceptual image coding scheme based on a decomposition of the image into structure and texture characterized by a Gaussian Markov random field. The 2D Itakura distance allows us to evaluate the performance of our texture model in terms of rate vs. distortion. A minimal quantization step size for near-lossless coding of model parameters is determined. Furthermore, we show that texture contrast can be efficiently coded using transform-based techniques.

***Index Terms*—** perceptual coding, Gaussian Markov random field, entropy coding, texture synthesis

1. INTRODUCTION

We have recently proposed the concept of a hybrid conventional/perceptual image coding scheme based on a decomposition of an image into two independent components – a *structure* and a *texture* part, where the texture part is characterized by a locally homogeneous Gaussian Markov random field (GMRF) [1, 2]. This class of texture is generally described as *random* or *noise-like*. The proposed coding scheme exploits the visual irrelevancy of texture by discarding Fourier phase of the texture component. Such a method appears attractive both because the decorrelating properties of orthogonal transforms diminish for noise-like texture and because Fourier phase is perceptually irrelevant for GMRF texture.

The texture part is encoded using estimated model parameters, while the structure (deterministic) part is encoded in a conventional way. At the decoder side, the texture is synthesized by sampling from the GMRF with the estimated model parameters. In many cases, such texture is observed in superposition with deterministic structure, which is our justification for the decomposition approach.

Unfortunately, the peak signal-to-noise ratio (PSNR) or structural similarity index (SSIM) are inappropriate measures for visual quality of the texture component, as two instances of an identical GMRF model may exhibit a large difference in pixel value (or low cross-correlation). However, spectral measures such as the Itakura distance [3] commonly known from speech coding can be used to quantify differences in the power spectrum if appropriately extended to two dimensions. The Itakura measure is attractive for our application as it is simply a function of the Maximum Likelihood (ML) estimate of the model standard deviation and therefore allows complexity-efficient vector quantization. As the power spectrum completely determines a GMRF, the measure also allows to assess reconstructed quality of texture given that the decomposition is correct.

*This work is supported by DFG under grant OH 50/13.

The "decomposition by denoising" method we use is unfortunately not optimal for this application, as denoising methods are commonly optimized for detection of Gaussian white noise, which merely represents a small subset of GMRF models. Better decompositions need to be found. However, in spite of this open question, we could demonstrate that our method can already improve visual quality of some texture [1] compared to transform-based codes since the model parameters represent a more compact representation of a GMRF. This holds asymptotically with growing extents and with increasing spectral flatness of the random field (i.e. with increasing "noisyness" of the texture). The question to be answered is at what point this representation becomes more efficient than conventional methods.

In order to find an empirical answer to this question, the encoding of the model parameters should be made as efficient as possible. In our previous work, we used an ad-hoc scheme of encoding the model parameters based on the Context-Adaptive Binary Arithmetic Coding (CABAC) engine [4]. In this paper, we evaluate several alternative techniques for improving the rate requirements of our model.

2. TEXTURE MODEL

The model describing the texture component consists of an i.i.d. Gaussian white noise process $\sigma w(x)$ with variance σ^2 and filtered by an "all-pole" filter $a(x)$:

$$t(x) = \sigma w(x) - a(x) * t(x) \tag{1}$$

The 2-vector x denotes spatial locations; $*$ denotes 2D convolution. This representation is commonly called 2D autoregressive (AR) or one-sided (unilateral) GMRF. The power spectrum of $t(x)$ is given by:

$$\phi_{tt}(f) = \frac{\sigma^2}{|1 + A(f)|^2} \tag{2}$$

where $A(f)$ is the 2D Fourier transform of $a(x)$. It can be shown that this random field is completely determined by its power spectrum, rendering the phase of $A(f)$ only relevant for implementation purposes. The 2D Itakura distance is given by:

$$D_{\mathrm{I}}(\tilde{\mathbf{a}}, \mathbf{a}) = \ln\left(\frac{\hat{\sigma}^2(\tilde{\mathbf{a}})}{\sigma^2}\right) \tag{3}$$

$$= \ln\left(\int_{\square} \frac{\phi_{tt}(f)/\sigma^2}{\phi_{\tilde{t}\tilde{t}}(f)/\tilde{\sigma}^2} df\right) \tag{4}$$

where $\hat{\sigma}(\tilde{\mathbf{a}})$ represents the ML estimate of σ conditioned on $\tilde{\mathbf{a}}$, a distorted set of model coefficients, and $\mathbf{a}$ represents the ML optimal model parameters (the vectorized coefficients of $a(x)$). The integral is taken over the unit square ($\square$). This measure captures the similarity of the gain-normalized power spectra of the ML optimal model

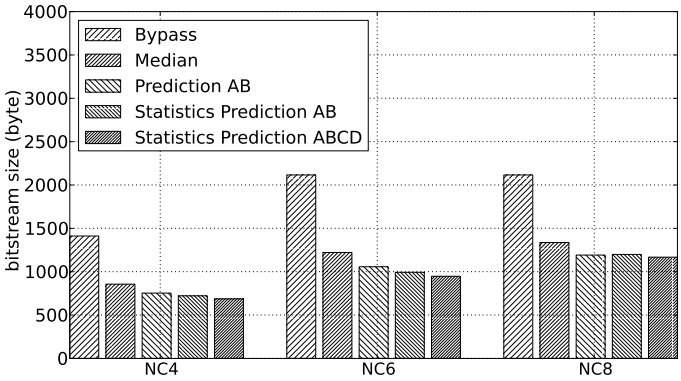

Fig. 1. Codebook index coding results averaged over the test set for three different codebook sizes (4, 6, and 8).

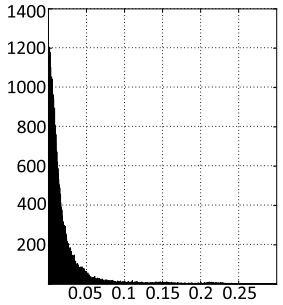

Fig. 2. Frequency of the absolute value of ARX model coefficients.

process $\phi_{tt}(f)$ vs. the process characterized by the distorted coefficients ($\phi_{\tilde{t}\tilde{t}}(f)$). It can be used to measure the distortion introduced by quantization of the model parameters.

Similarly to our previous work, we use a decomposition based on the Non-Local Means (NLM) filter [5]. The denoised version of the image corresponds to the structure component $s(x)$, while the difference image between denoised and original versions amounts to the texture component $t(x)$. The model parameters are estimated locally on a block partitioning of the texture component and the coefficients $\mathbf{a}$ are vector quantized. The codebook indices and standard deviations are encoded block-wise, and the codebook vectors are transmitted using scalar quantization.

To compensate for the remaining dependency between structure and texture parts left by the denoising, we previously extended the model equation (1) by an additional term,

$$t(x) = \sigma w(x) - a(x) * t(x) - c(x) * s(x) \tag{5}$$

where $c(x)$ represents the filter modeling linear dependency between the two components [1]. This equation corresponds to a modified model known as ARX (autoregressive with external input). In this case, the ML optimality degrades to least-squares optimality and the equality of Eqs. (4) and (3) does not hold any longer. However, Eq. (3) is still a complexity-efficient measure for distortion introduced by coefficient quantization.[1] We expect our conclusions to be transferrable to the AR case.

3. CODING SCHEMES

In [1] we proposed a scheme for entropy coding of the ARX model parameters. In this section, we present different approaches for improved coding efficiency. We use an independent implementation of the Context-Adaptive Binary Arithmetic Coding (CABAC) engine. The model comprises three types of parameters to be signalled.

3.1. Codebook indexes

The indexes $B(i)$ map each block position i to a centroid vector numbered k. Since texture is usually attached to objects in a scene that span multiple blocks, the indexes tend to form areas of blocks that are associated with the same centroid. This behavior tends to be stronger when the vector quantizer is regularized using a Gaussian convolution [2], which is necessary for very small block sizes.

[1]It is always non-negative and $D_{\mathrm{I}}(\mathbf{a}, \mathbf{b}) = 0$ implies $\mathbf{a} = \mathbf{b}$.

We evaluated several predictive lossless coding methods. $B(i)$ is scanned in raster scan order, and the already decoded neighboring blocks A (left), B (top), C (top right) and D (top left) are used to predict the index of the current block X. For all methods, if prediction fails or no surrounding blocks are available, the index is encoded in bypass mode using a fixed-length binarization.

Bypass mode: All indices are encoded using a fixed-length binarization and the bypass engine.

Median Prediction: The median value of A, B and C is used for the prediction. If C is not available, it is replaced by D. One bit in one CABAC context encodes if the prediction is correct. The idea is that the median will act as a majority decision choosing the index that occurs more than once.

Prediction AB: This is the method proposed in [1] that uses the blocks A and B to predict the mapping of X. The first bit encodes whether X has the same index as either A or B. One of two contexts are chosen for encoding this bit depending on whether A and B share the same index or not. If the first bit is true but A and B do not share the same index, a second bit, encoded in bypass mode, indicates which block yields the correct prediction.

Statistics Prediction AB: In order to further improve the prediction efficiency, we measured the relative frequencies of the four neighbors yielding the correct prediction, conditioned on the neighborhood constellation (for example, A, B and C may share the same index, while B has a different index). The coding method was the same as the above (Prediction AB), but the two bits were encoded in contexts based on the relative frequency. We used 10 contexts to bundle bits with approximately the same probability.

Prediction Statistic ABCD: The relative frequencies can also be used to extend the possible predictions to the neighboring blocks C and D. For a given neighborhood constellation, the statistic can provide a most probable prediction and a second most probable prediction in A, B, C and D. In the first bit, we encode the decision whether either one of these predictions is correct and in the second bit we signal which one of these two it is. If there is no second best prediction, the second bit can be omitted. Like before the context will be selected depending on relative frequencies.

3.2. ARX coefficients

Analysis of the coefficient distribution shows that there is a strong concentration at 0 that can be exploited by entropy coding (Fig. 2). The sign values are equiprobable, so if the quantized value is not equal to 0 they are encoded using the bypass engine.

Exp-Golomb, $k = 0$: The values are encoded using an Exp-Golomb code with $k = 0$. This is the scheme proposed in [1].

Exp-Golomb, adaptive k: In most cases, $k = 0$ yields no good

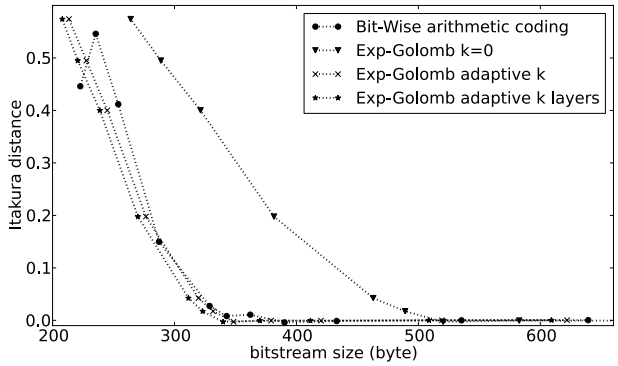

Fig. 3. Bitstream size vs. D_{I} for image "Lena" and a varying scalar quantization of $\mathbf{a}$. The depicted quantization step size ranges from 0.01 to 0.00001.

approximation to the distribution of the model parameters. Thus, a better value is determined by encoding the model parameters using values for k from 0 to 25 and choosing the value that yields the smallest bit rate. The optimal value of k is signalled to the decoder at the beginning of the bitstream.

Exp-Golomb adaptive k layers: Since large values tend to concentrate at the origin of $a(x)$, we group coefficients into "layers" of equal distance to the origin of the filter kernel. Each layer is then encoded like above and the k-value difference from each layer to the next is signalled to the decoder.

Bit-wise arithmetic coding: We adopt a method detailed in [6]. The quantized value is represented by its binary representation. Every bitlevel is encoded in a seperate CABAC context. This way, the most significant bit has a high probability of being 0 which decreases with decreasing significance. We then use the arithmetic coding engine to exploit the distribution of the model parameters.

3.3. ARX model standard deviations

The standard deviation $\sigma(i)$ of the driving noise signal $\sigma w(x)$ needs to be coded for every block i. In our experiments, it became obvious that the map of $\sigma(i)$ over the blocks shares certain properties with natural images. This is due to the local properties of texture and can be explained by the fact that σ^2 is directly related to the local energy (i.e., local contrast) in the corresponding block (Eq. (2)). An example can be seen in Fig. 4. Due to this relation, we measure distortion of σ using the MSE and examine the use of transform codes for encoding the model standard deviations.

CABAC binarization: We use a unary code up to quantization step index g. If the value is higher than this limit, the difference between the value and the limit is encoded with an Exp-Golomb code with $k = 0$. The unary coded bits are encoded in separate CABAC contexts in order to let CABAC approximate the distribution.

Context from codebook indexes: Blocks that are associated to the same centroid vector generally have a similar distribution of values of σ. The values are encoded like above, but with separate CABAC contexts for each centroid. This is the method used in [1].

JPEG-LS Prediction: JPEG-LS defines a prediction scheme that is based on an edge heuristic and uses the blocks A, B and D to get a prediction for the current block [7]. The prediction error is then encoded using the above CABAC unary/Exp-Golomb code.

Median Prediction: Another possible prediction for X is the median of A, B and C (if not available, D). As in the methods above, the prediction error is encoded using a unary binarization and an Exp-Golomb code.

Wavelet methods: As a readily available method, the SPIHT algorithm in combination with the Haar wavelet and the Daubechies 9/7 wavelet was implemented.

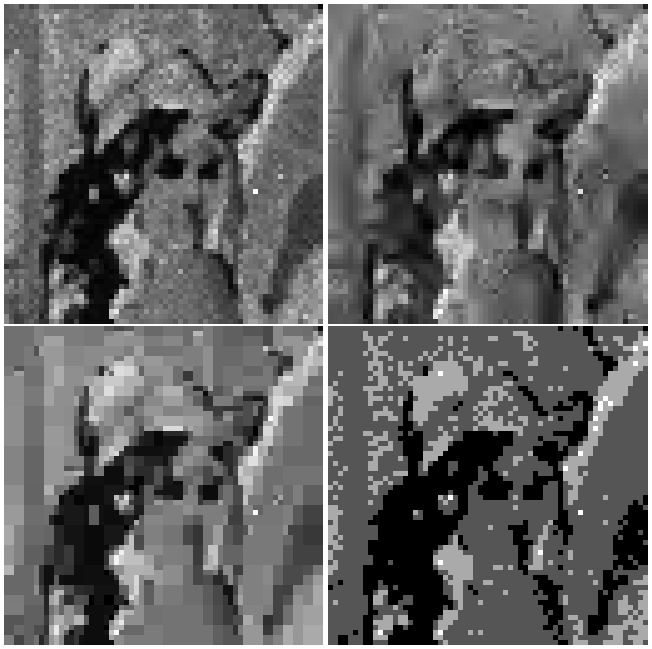

Fig. 4. Example of a map of standard deviations for "Lena." Top left: model estimates. Top right: lossy Daubechies wavelet version. Bottom left: lossy Haar wavelet version. Bottom right: lossy scalar quantization. All lossy methods are compared at the same bitrate.

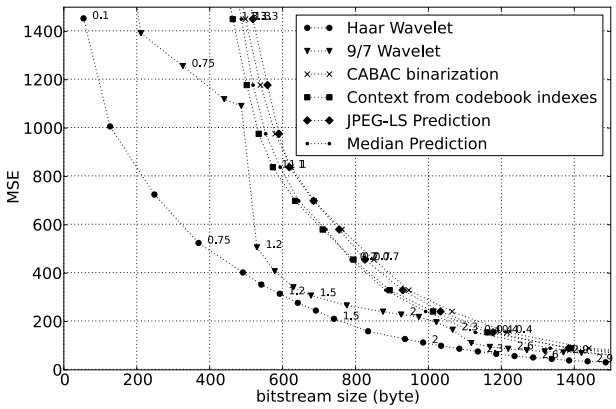

Fig. 5. MSE vs. bitstream size of $\sigma(i)$ for Kodim02.

4. RESULTS

We evaluated the methods on a test set comprising the Kodak set and the well-known test images Baboon, Barbara, Boat, Clown, Elaine, Lena, Peppers and Plane. Unless otherwise noted, we used an 11×6 support for $a(x)$, a 1×1 support for $c(x)$, a block size of 8×8 pixels, 4 centroid vectors, and a quantization step size of 0.00001

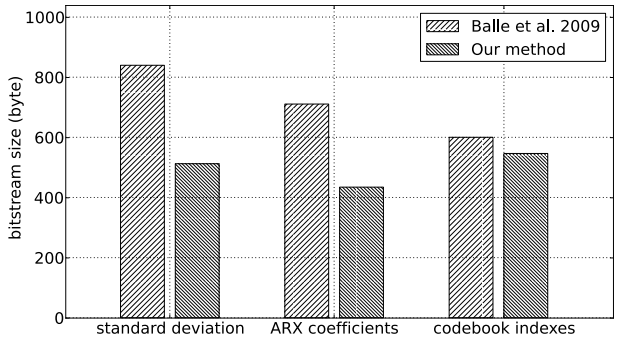

Fig. 6. Overall rate improvement compared to [1] at same settings, averaged over the test set.

Fig. 7. Reconstruction at 0.8 bpp, top: JPEG2000, bottom: our method

for the ARX model coefficients. This is the same setup as for our previous method [1], so results can be compared.

Codebook indexes: All predictive methods provide comparable results (Fig. 1). The median prediction is slightly worse than the method proposed in [1]. The results also show that the usage of an explicit statistic can further improve the efficiency at the additional cost of creating a statistic from a representative set of images.

ARX coefficients: The results show that the bit-wise arithmetic coding as well as the Exp-Golomb code with adaptive k perform significantly better than the Exp-Golomb code for $k = 0$ (Fig. 3). The results for the "layered" approach are even slightly better at almost no additional cost. For the other images in the test set, the results are very similar. We have observed that the quality of the reconstructed texture may sometimes decrease for higher bitrates (left side). This is due to the fact that there is no direct relationship between the quantization step size and the Itakura distance. However, for a quantization step size of 0.00001 or finer, D_{I} approaches 0 for all images in the test set. Generally, this result indicates that working with a fixed VQ codebook should be considered, such that scalar quantization of ARX model coefficients can be avoided entirely.

ARX model standard deviations: Obviously, the transform based methods yield a different reconstruction of the standard deviation map than the scalar quantization methods. In order to generate comparable results, we selected the rate point for the transform based methods in a way that the MSE equaled the one obtained for the scalar quantization, for each image. In Fig. 6, it can be seen that this yielded substantially better performance. A similar rate–distortion performance as in Fig. 5 could be observed for the entire test set.

Overall, the setup was adjusted to approximate visual quality of our previous method as close as possible. Compared to our previous work, we achieve similar visual results at lower rates. Fig. 7 represents an example. The skin texture is flattened by the JPEG2000 encoder, while our method obtains a better reconstruction.

5. CONCLUSION

We evaluated several alternative coding schemes for component-based image coding. Overall, the entropy coding performance was significantly improved. Results indicate that the standard deviations of the model can be efficiently encoded using transform-based techniques. Ideally, the model coefficients should be signalled using a fixed codebook, such that scalar quantization can be avoided. This, like an improved, model-aware decomposition and a better comparison to conventional methods by means of subjective testing, is a topic for future work.

6. REFERENCES

[1] J. Ballé, B. Jurczyk, and A. Stojanovic, "Component-based image coding using non-local means filtering and an autoregressive texture model," in *Proc. of IEEE International Conference on Image Processing ICIP '09*. Cairo, Egypt: IEEE, Piscataway, Nov. 2009, pp. 1937–1940.

[2] J. Ballé and M. Wien, "A quantization scheme for modeling and coding of noisy texture in natural images," in *Proc. of IASTED Conference on Signal and Image Processing SIP '09*. Honolulu, HI, USA: ACTA Press, Calgary, Aug. 2009.

[3] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23, no. 1, pp. 67–72, Feb. 1975.

[4] 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, no. 7, pp. 620–636, Jul. 2003.

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

[6] A. B. Kiely, "Bit-wise arithmetic coding for data compression," in *Proc. of IEEE International Symposium on Information Theory '95*, Sep. 1995, p. 394.

[7] D. S. Taubman and M. W. Marcellin, *JPEG 2000 – Image Compression Fundamentals, Standards and Practice*. Kluwer, Amsterdam, 2002.