COLOR PREDICTION IN IMAGE CODING USING STEERED MIXTURE-OF-EXPERTS

Ruben Verhack[⋆†], *Simon Van De Keer*[⋆], *Glenn Van Wallendael*[⋆], *Thomas Sikora*[†], *and Peter Lambert*[⋆]

[⋆]Ghent University - iMinds - Data Science Lab, Ghent, Belgium
[†]Technische Universität Berlin - Communication Systems Lab, Berlin, Germany

ABSTRACT

We propose a novel approach for modeling and coding color in images and video. Luminance is linearly correlated with chrominance locally, as such we can predict color given the luma value. Using the *Steered Mixture-of-Experts* (SMoE) approach, the image is viewed as a stochastic process over 5 random variables including the 2-D pixel locations, 1 luminance and 2 chrominance values. We model this process as a continuous joint density function by fitting a K-modal 5-D *Gaussian Mixture Model* (GMM). As such, the chroma values are predicted as the expectation of the conditional density. To validate, the technique was integrated within JPEG showing PSNR gains in the lower bitrate regions. A deeper analysis of the tolerance of the activation function is given through recycling color models in video sequences, yielding a high quality reconstruction over a considerable range of frames.

Index Terms— Image coding, inter-channel prediction, color prediction, Gaussian Mixture Model, Mixture-of-Experts

1. INTRODUCTION

Over the last few decades, image and video coding have been a very active field of research. The abundance of images is ever increasing, given the popularity of social media platforms and on-demand services. The proportional growth in computing power has opened the path for exploring computationally heavier techniques aiding in compression [1]. In this work we design a novel approach using modern machine learning methods for color modeling in image and video coding. The main goal being to find a luma to chroma predictor that is able to reconstruct the color components efficiently.

The human vision is much less susceptible to nearby changes in color compared to changes in luminance [2]. In order to treat luma and chroma differently, a transformation from RGB to the YCbCr color space is often used. This splits the components in a luminance component Y and the chroma components Cb and Cr. Based on the previous observation, chroma components are often subsampled [3] and quantized more coarsely [4]. Although the YCbCr-transform decorrelates the channels globally, in practice correlation between the luma channel and the chroma channels still exists locally.

Research has shown that within small regions of an image, linear inter-channel correlation exists between the luminance and chrominance channels [5] [6]. This lead to a proposal for an integration within HEVC [7], but was rejected because of non-negligible overhead and the loss of luminance-chrominance plane parallelism. Color modeling is also inherent to the field of colorization, as such our approach was inspired by the work of Cheng et al [8].

All modern and well established compression schemes are based on block-based transform coding and DPCM-like prediction methods. In this work we take a completely different approach, which was largely motivated by the recently introduced *Steered Mixture-of-Experts Regression* (SMoE) methodology [9][10], combined with ideas from image colorization [11]. We present a color modeling scheme using *Mixture-of-Experts* to model a non-linear predictor $F(\boldsymbol{x}, Y) \rightarrow (C_\text{b}, C_\text{r})$ over the whole image, where Y is the luminance, $\boldsymbol{x}$ is the 2-dimensional pixel location, and C_b and C_r are the chrominance values. This approach moves away from the usual block based techniques central to many modern compression schemes, e.g. JPEG and JPEG2000 for image and HEVC for video.

The underlying stochastic process of the amplitudes are modeled as 5-D (2-D location and 3 color channels YUV) multi-modal *Gaussian Mixture Model* (GMM). As such a space-continuous internal representation of the image is obtained. The GMM models the joint probability density function, which contains all the necessary and sufficient statistics to perform the chroma regression. The decoder then performs the chroma estimation based on this model. Every Gaussian kernel is considered as an expert and all experts collaborate toward the chroma reconstruction given a location and a luma value. Given the softmaxed support of the experts, the model yields a continuous, smoothed piecewise regression function over the whole domain.

The research activities described in this paper were funded by the Data Science Lab (Ghent University - iMinds), Communication Systems Lab (Technische Universität Berlin), Flanders Innovation & Entrepreneurship (VLAIO), the Fund for Scientific Research Flanders (FWO Flanders), and the European Union. The computational resources (STEVIN Supercomputer Infrastructure) and services used in this work were kindly provided by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation, and the Flemish Government department EWI.

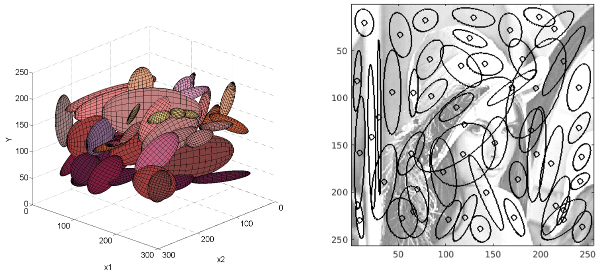

Fig. 1. Gaussian mixture model with 50 Gaussians for a subsampled version of size 256×256 of *Lena* (left) with a top view showing the location in the image (right).

2. STEERED MIXTURE-OF-EXPERTS

In the *Steered Mixture-of-Experts* (SMoE) framework, the underlying stochastic process of the amplitudes are modeled as an N-D multi-modal Mixture Model with K modes. The parameters are estimated using e.g. the *Expectation-Maximization* (EM) algorithm [12]. The mixture describes segments of pixels by local N-D Gaussian steering kernels with global support. As such, each component in the Mixture-of-Experts steers along the direction of highest correlation. The conditional density then serves as the regression function. Consequently, we arrive at a closed form continuous analytical model. Previous work has illustrated this idea for image and video luma coding, where the joint probability density function was modeled as respectively 3-D and 4-D *Gaussian Mixture Models* (GMMs) [9][10].

In this paper, color images are modeled as 5-D GMMs (2-D position and one dimension per color channel), in which the known random 3-D variable X contains position and luma value, while the random 2-D variable Z holds the two chroma components. To avoid confusion, the variable name Y is reserved for the luma values. In case of GMMs, the regression function $m(\mathbf{x})$ is given by

$$m(\mathbf{x}) = E[Z|X = \mathbf{x}] = \sum_{j=1}^{K} w_j(\mathbf{x}) m_j(\mathbf{x}) \tag{1}$$

with

$$w_j(\mathbf{x}) = \frac{\pi_j \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{X_j}, R_{X_j X_j})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{X_l}, R_{X_l X_l})} \tag{2}$$

$$m_j(\mathbf{x}) = \boldsymbol{\mu}_{Z_j} + R_{Z_j X_j} R_{X_j X_j}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{X_j}) \tag{3}$$

and with π_j being the prior/mixing weight, $\boldsymbol{\mu}_j$ the center, R_j the 5×5 covariance matrix, and $\mathcal{N}(\cdot)$ the Gaussian distribution. The subscript X denotes the first 3 dimensions, Z the last 2 dimensions. The parameters $(\pi_j, \boldsymbol{\mu}_j, R_j)$ are estimated using the EM algorithm [12]. Its application in mixture models is well known [13]. Note that although the algorithm is said to converge, it only converges to one of many local optima [12]. As such, the algorithm is sensitive to initialization.

The resulting function (Eq. 1) has the form of a kernel estimator. Formally said, kernel estimators smooth out the contribution of each observed data point over a local neighborhood of that data point [14]. But whereas normally the weighting function is determined by local structure in the data (because the kernel function takes only the local neighborhood into account), the weighting terms $w_j(x)$ here are determined by the global density (Eq. 2). These terms can be seen as window or activation functions determining the amount of influence the component has on a given point. They are actually equal to the posterior probabilities that a certain Gaussian was responsible for generating **x**, given that we have seen **x**. The terms $m_j(\mathbf{x})$ (Eq. 3) are the outputs of the components, they describe how the data behaves around the center of the component. Note that our approach is a global semi-parametric method with the flexibility of a non-parametric one.

3. COLOR CODING

3.1. Modeling

We are estimating a function $F(\boldsymbol{x}, Y) \rightarrow (C_\mathrm{b}, C_\mathrm{r})$ over the whole image, where Y are the luma values, $\boldsymbol{x} = [x_1, x_2]$ the positions of the pixels, and C_b and C_r are the chroma values. This readily translates to a 5-dimensional Mixture-of-Gaussians to model the joint density of these variables, from which we can derive the regression function.

A visual representation of the model can be found in Fig. 1. All 5 dimensions are visualized. The input variables (x_1, x_2, Y) are on the three axes, on which we plotted their 3-D Gaussians, while the output variables $(C_\mathrm{b}, C_\mathrm{r})$ are represented as the color of the ellipsoids, determined by the YCbCr values of their centers.

Given that the resulting model is continuous. Any resolution can be sampled from this function, which means that predictions are readily available given any scale. This allows for easy resampling and super-resolution for the color model. The smoothing properties of the model can also implicitly denoise the chrominance components, similar to [15].

3.2. Image coding

Considering the Eq. 1, 2, and 3, the parameters of the model needed are the mixing coefficients π_j, the centers $\boldsymbol{\mu}_j$, and parts of the covariance matrices: $R_{X_j X_j}$ and $R_{Z_j X_j}$. Note that $R_{X_j X_j}$ is 3-D and defines the weighting/window function, whereas $R_{Z_j X_j}$ defines the linear correlation between the luma and location with the color channels. As X is 3-D and Z is 2-D, these matrices have respectively 9 and 6 values. Because the covariance matrix is symmetrical by nature, only 6 of the 9 components are needed for $R_{X_j X_j}$. In total: 5 mean values, 12 covariances and 1 mixing coefficient

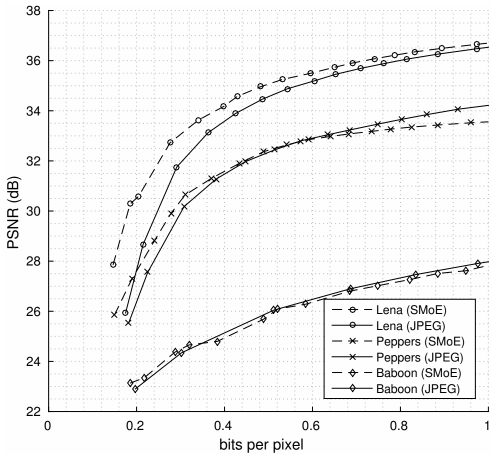

Fig. 2. Rate-distortion curves for three reference images *Lena*, *Mandrill* and *Peppers* (PSNR)

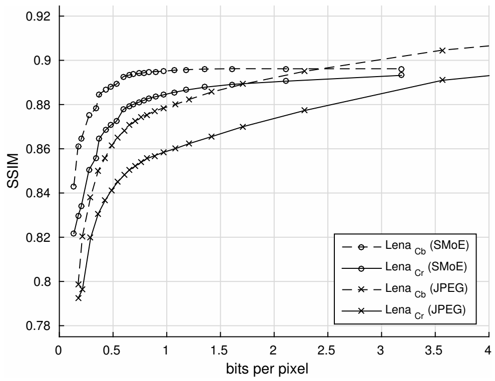

Fig. 3. Rate-distortion curves for *Lena* (SSIM)

are stored per Gaussian. These are all floating point numbers. The center values and mixing coefficients are quantized to 8 bit and the covariances are quantized to 10 bit, bringing the total number of bits per mixture component to 168. No other form of coding (e.g. entropy coding) is performed to the values in this work. More elaborate methods exist using difference coded components, eigen decompositions, and arithmetic coding [9].

An encoder thus exists of a luma-encoding part and a color modeling part, in which the parameters are estimated. Consequently, the parameters are stored together with the compressed luminance plane. A decoder would need to decode the luma plane first, to feed it together with the model parameters to a regressor component, which produces the chroma that can be recombined with the luminance values to reconstruct the image. It should be noted that the decoder can be highly parallel, as each chrominance value can be computed independently.

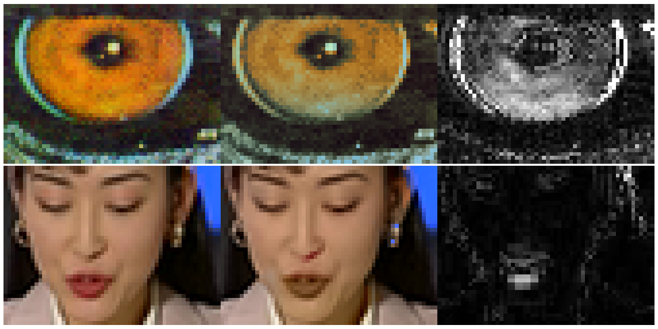

Fig. 4. Two types of artifacts (left: original, middle: reconstruction, right: residual). Above: bad color estimation due too little components (underfit). Below: bad color estimation due to motion (*akiyo* sequence).

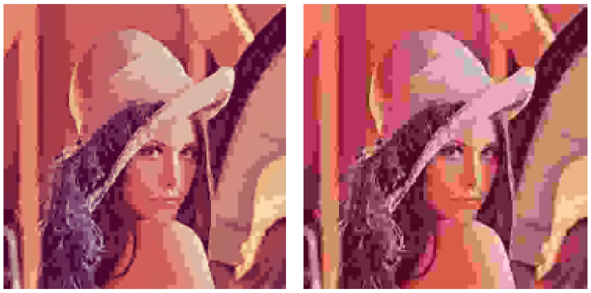

Fig. 5. Image reconstruction at extremely low bitrate (0.17 bpp). Left: GMR, right: JPEG

4. EXPERIMENTAL RESULTS

4.1. Experimental setup

For our experiments, we initialized the EM algorithm using a 2-D Sobol sequence for choosing the initial position quasi-uniformly [16]. The covariance matrices were initialized as diagonal matrices with a bandwidth of 0.001 for the pixel locations and $(C_\mathrm{b}, C_\mathrm{r})$ values and a bandwidth of 0.01 for the Y values.

4.2. Image coding - Integration with JPEG

In order to evaluate the viability of our technique in image coding, we integrated our technique into the JPEG standard. We consider this work to be a proof-of-concept, as our model is only coded very roughly and is not considered to be optimized. As such, more highly optimized state-of-the-art is not considered for comparison. The GMM is trained on the original luminance values, and the reconstruction is done on the JPEG reconstructed luminance values. Note that adding more components to the mixture increases the quality, but also increases the bitrate. Optimization between of the quantization strength and the amount of components was done by a simple grid search.

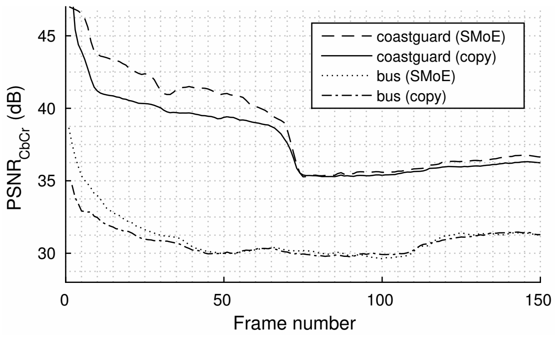

Fig. 6. PSNR$_{\mathrm{UV}}$ per frame for the reference video sequences *bus* and *coastguard* using SMoE vs. copying the chroma values

The rate-distortion curves can be found in Fig. 2 for three reference images of 512×512 pixels. *Baboon* contains mostly very high frequency content. *Lena* has softer color changes, but also contains some highly textured content as well as very low textured content. *Peppers* has overall low frequency content, but contains hard differences between colors with edges in between. At bitrates lower than 0.25 bpp, we can see improvements in terms of PSNR over JPEG. This proves that a gain in compression is possible at the lowest bitrates. If we look at the rate distortion in terms of SSIM for *Lena* (Fig. 3), we can see reductions up to 50% in bpp in the ranges of 1.0 - 2.0 bpp for the Cb and Cr components. SSIM is meant to assess quality in a more perceptual fashion, although that is debatable [17]. SSIM favors more continuous structural properties [18], which is exactly what our technique does well. However at the higher bpp ranges, this is also a flaw, as the model can not capture detailed structures in parts of the image.

The typical loss of detail due to model underfitting is shown on top of Fig. 4. One can see that for *Baboon*, the details around the eye are excessively blurred. A reconstruction at the lowest bitrate for *Lena* is shown in Fig. 5, where it is clear that JPEG distorts the colors whereas our technique yields a smooth and visually more accurate representation.

4.3. Activation/Windowing function

In this section we take a closer look to the activation function $w_j(\boldsymbol{x})$ for each component. This function is activated by the position and the luminance of a certain position. As such, our model is assumed to handle relatively small displacements which allows for relatively coarse modeling.

We evaluate this assumption by taking a look at consecutive frames of video sequences. We assume that for fairly static content a model estimated for a given frame will most likely hold for consecutive frames as well. Luminance values are locally mapped to chrominance values. When a segment of luminance values slightly shifts away from its original position, the location will not match that in the model anymore. The luma values however will still be close to the center of a nearby component, which will contribute to it still getting the correct color.

Note that in this case, the model can be built at decoder side. As such, the color of the first frame needs to transmitted on which the model is trained. Also note that we do not incorporate any form of motion compensation. In practice, it is possible to update the centers of the components by following the direction of the motion. Or better, time could be added as a 6th dimension. As such, the components move along the motion [10], the model should then be trained on a set of frames.

Two test sequences were tested: (1) *bus*, which contains relatively large motion with a camera following a bus through traffic and (2) *coastguard*, which shows two boats crossing each other, and contains a large sudden camera motion. Both models were trained using 200 components on the first frame. This model is then reused on all consecutive frames. Please note that no motion compensation is taken into account. The goal is to illustrate that relative small motion is captured by incorporating the luminance into the weighting/gating function for each expert. As such, we compare with simply copying over the chroma values of the first frame to the consecutive ones.

Fig. 6 shows the average PSNR of the Cb,Cr channels of the reconstruction (PSNR$_{\mathrm{CbCr}}$), compared with simply copying the chroma values from the first frame. For *bus*, although there is much motion, the color model is locally stable for a considerable number of frames. As such, it performs better than simply copying the values. Secondly, for *coastguard*, the PSNR is gradually dropping because of the movement of the boats. A sudden drop appears when the camera moves upward, which makes the color model inconsistent with the data. Both examples have the inherent disadvantage compared to simply copying the chroma values from the first frame, that they inherently lose quality because the chroma are approximated, not copied.

We can conclude that our model is able to capture relatively small motion changes. However, the model should be retrained when it becomes inconsistent, or the centers should move along the motion if the number of frames increases.

5. CONCLUSION

In this paper, we explored the possibility of using *Steered Mixture-of-Experts* for color modeling for image and video coding. Firstly, it was shown that compression gains are possible for low bitrates by to incorporate our technique in JPEG, even with very crude coding of the color model. Secondly, our experiments have shown that the model allows for some displacement of the luma values. Finally, future work involves exploring the time dimension in video, and improving the coding of the model.

6. REFERENCES

[1] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, 2005.

[2] Z.-N. Li, M. S. Drew, and J. Liu, *Fundamentals of multimedia*, Springer, 2004.

[3] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," *IEEE transactions on consumer electronics*, vol. 46, no. 4, pp. 1103–1127, 2000.

[4] M. T. Orchard and C. A. Bouman, "Color quantization of images," *IEEE transactions on signal processing*, vol. 39, no. 12, pp. 2677–2690, 1991.

[5] S. H. Lee and N. I. Cho, "Intra prediction method based on the linear relationship between the channels for YUV 4:2:0 intra coding," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 1037–1040.

[6] L. F. Lucas, N. M. Rodrigues, S. M. de Faria, E. A. da Silva, M. B. de Carvalho, and V. M. M. da Silva, "Intra-prediction for color image coding using YUV correlation," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 1329–1332.

[7] Xingyu Zhang, C. Gisquet, E. Francois, Feng Zou, and O. C. Au, "Chroma Intra Prediction Based on Inter-Channel Correlation for HEVC," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 274–286, jan 2014.

[8] L. Cheng and S. Vishwanathan, "Learning to compress images and videos," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 161–168.

[9] R. Verhack, T. Sikora, L. Lange, G. Van Wallendael, and P. Lambert, "A universal image coding approach using sparse steered mixture-of-experts regression," in *Proceedings IEEE International Conference on Image Processing 2016 (ICIP '16)*, 2016.

[10] L. Lange, R. Verhack, and T. Sikora, "Sparse steered mixture-of-experts regression for universal video coding," in *Submitted to 2016 Picture Coding Symposium (PCS '16)*, 2016.

[11] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics (TOG)*. ACM, 2004, vol. 23, pp. 689–694.

[12] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[13] G. McLachlan and D. Peel, "Finite mixture models," 2004.

[14] M. P. Wand and M. C. Jones, *Kernel smoothing*, Crc Press, 1994.

[15] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on image processing*, vol. 16, no. 2, pp. 349–366, 2007.

[16] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[17] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on image processing*, vol. 15, no. 11, pp. 3440–3451, 2006.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.