A TWO-PART PREDICTIVE CODER FOR MULTITASK SIGNAL COMPRESSION

Scott Deeann Chen and Pierre Moulin

University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering
405 North Mathews Avenue, Urbana, IL 61801 USA

ABSTRACT

Traditional compression techniques optimize signal fidelity under a bit rate constraint. However, signals are often not only reconstructed for human evaluation purposes but also analyzed by machines. This paper introduces a two-part predictive (2PP) coding architecture intended for signal compression with the dual purposes of preserving signal fidelity and feature fidelity. First we introduce the architecture of the 2PP coder, then we apply and evaluate it on two problems: scene classification and pedestrian detection. Tradeoffs between compression rate, mean-squared reconstruction error, and classification accuracy, are explored.

***Index Terms*—** compression, predictive coding, quantizer learning, scene classification, pedestrian detection

1. INTRODUCTION

The vast amount of image and video data produced by surveillance and related applications presents critical challenges in terms of storage, transmission, processing, and interpretation – especially when the image sensors operate in mobile and bandwidth-constrainted environments. While traditional compression methods such as JPEG2000 (for still images) and H.264 (for video) attempt to maximize visual quality under a rate constraint, they are not ideal for other tasks such as target identification, detection, and localization. In particular, the features extracted from the compressed images or video might be substantially degraded versions of the original ones. This has negative consequences in terms of performance for the aforementioned tasks. For example, when detecting pedestrians in compressed video, false positives and misses increase sharply. As illustrated in Fig 1, the state-of-the-art FPDW pedestrian detection algorithm performs well on the uncompressed image but poorly on the JPEG2000 compressed image.

The basic question then, is how to compress signals when multiple evaluation criteria are relevant. Interest in theoretical and practical aspects of this problem began in the 1990s. Baras *et al.* [2] and Perlmutter *et al.* [3] designed vector quantizers for the problem of joint compression and classification, and Jana and Moulin [4] optimized transform coders for such problems. However these papers use fairly simple surrogate functions for coder design and do not provide means to exploit the latest advances in image/video compression and classification. Hence, we propose a two-part predictive (2PP) coder which integrates state-of-the-art compression and classification building blocks and aims at providing good visual quality as well as high-quality image features. Our 2PP coder uses compressed signals as predictors for features. Related work includes scalable coding [5], where a low-resolution video is used to predict a high resolution version. The 2PP coder is described in Section 2 and applied to scene classification in Section 3 and to pedestrian detection in Section 4.

Fig. 1. Detection results of the state of the art pedestrian detector, the Fastest Pedestrian Detector in the West (FPDW) [1] are shown in green bounding boxes. The left is an uncompressed frame and the right is a JPEG 2000 compressed frame at compression ratio of 1000.

2. THE 2PP CODER

The 2PP coder is diagrammed in Fig. 2. Its key components are a lossy codec, feature extractors, and quantization functions. These components are integrated into a predictive coding framework as diagrammed in Fig. 2(a). The choices of the codec and the feature extractor depend on the type of signal and on the content analysis task at hand. The codec is a state-of-the-art system such as JPEG 2000 for images, H.264 for videos, and MP3 for audio. The feature extractor captures essential information for content analysis, such as spectrograms for speech recognition, dense SIFT visual-word histograms for scene classification [6], and integral channel features [7] for pedestrian or object detection. The quantizers will be discussed later in this section.

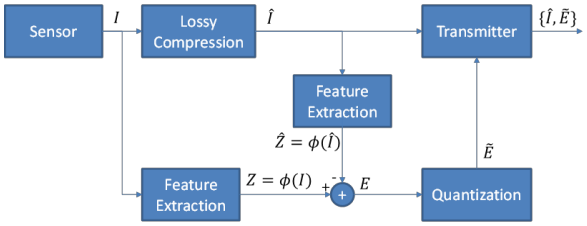
The 2PP coder outputs two parts: content bits and feature bits. The aforementioned lossy codec generates the content

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

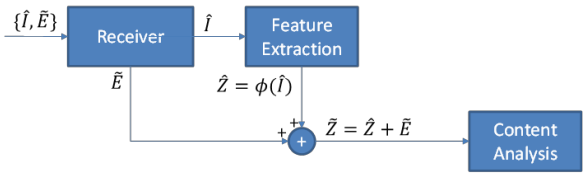
bits. The feature extractor computes the features for both the original and compressed signals. The difference (prediction error) is then quantized and encoded into feature bits which will be used to mitigate the information loss due to compression.

The 2PP decoder is diagrammed in Fig. 2. First the content bits are used to decompress and display the signal. Second, the (degraded) features are computed from this decompressed signal. Third, they are refined using feature bits, and input to the content analysis algorithm.

For a given bit budget, the 2PP coder allows a tradeoff between visual quality and content analysis through allocation of bits to content and to features. One extreme of the tradeoff is to allocate all bits to content (as is done in conventional coders). The other extreme is to allocate most bits to features. In practice, a suitable operating point can be selected that provides satisfactory visual quality and content analysis performance.



(a) Encoder of 2PP coder.



(b) Decoder of 2PP coder.

Fig. 2. 2PP coder architecture

Formally, we denote by $I \in \mathbb{R}^n$ the uncompressed signal, by $\hat{I} \in \mathbb{R}^n$ the compressed version of I, by $\phi(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^d$ the d-dimensional feature extractor that maps I to $Z = \phi(I)$ and $\hat{I}$ to $\hat{Z} = \phi(\hat{I})$, by $E = Z - \hat{Z}$ the feature prediction error, and by $\tilde{E}$ the lossy compressed version of E. The content bits describe the compressed signal $\hat{I}$. The feature bits describe the lossy compressed feature prediction error, $\tilde{E}$.

Receiving $\hat{I}$ and $\tilde{E}$, the 2PP decoder approximates the original feature vector Z by $\tilde{Z} = \phi(\hat{I}) + \tilde{E}$ which is input to the content analysis module.

The 2PP coder allows a tradeoff between visual quality and analysis performance. Denote by B_1 and B_2 the number of content and feature bits, respectively. According to bandwidth and performance requirements, a user chooses a bit budget $\bar{B} \geq B_1 + B_2$ and determines the number of content bits, B_1, and feature bits, B_2.

To control B_1, the user tunes settings of the compression scheme, such as the compression ratio of JPEG2000 or bit rate of H.264. To control B_2, the user selects the number of bits assigned per feature. Of the original d scalar features, only a subset (of size $d' \leq d$) will be allocated bits.

Precisely, the number of feature bits, B_2, depends on d' as well as the statistics of feature prediction error vector $E = \{E_j,\ 1 \leq j \leq d\}$. We quantize each E_j into k levels by a quantizer $q_j : \mathbb{R} \mapsto \{q_{j1}, \ldots, q_{jk}\}$. The quantized feature prediction error vector is then $\tilde{E} = \{q_j(E_j),\ 1 \leq j \leq d\}$. The number of bits required to encode $\tilde{E}$, assuming an entropy encoder and statistically independent components, is $B_2 = \sum_{j=1}^{d} H(\tilde{E}_j)$, where $H(\tilde{E}_j)$ denotes the entropy of $\tilde{E}_j$. Hence, $B_2 \leq d \log_2 k$, where the upper bound holds when $\{E_j\}_{j=1}^{d}$ are uniformly distributed. If $d' < d$ we have $B_2 \leq d' \log_2 k$.

The design of quantizers $q_j(\cdot)$ affects B_2 in two ways. First, B_2 grows logarithmically with the number of levels k. Second, the quantization levels $\{q_{j1}, \ldots, q_{jk}\}$ affect the distribution of $\tilde{E}$. The quantizers could be designed heuristically or learned from training data. Heuristic designs require prior knowledge of the distribution. Instead, we learn the quantization levels from training data as described below.

For each element I_i, $1 \leq i \leq p$ of a set of p training data, we first compute its features $Z_i = \phi(I_i)$, its compressed version $\hat{I}_i$, the features of its compressed version $\hat{Z}_i = \phi(\hat{I}_i)$, and its prediction errors $E_i = Z_i - \hat{Z}_i$. We propose two quantizers:

1. A simple 3-level quantizer with levels $\{\mu_j - \sigma_j, \mu_j, \mu_j + \sigma_j\}$, where μ_j and σ_j denote the empirical mean and standard deviation of $\{E_{ij}\}_{i=1}^{p}$ and E_{ij} denotes the jth component of E_i.

2. A Lloyd-Max quantizer of k levels trained from $\{E_{ij}\}_{i=1}^{p}$.

The compression ratio of the 2PP coder is given by

$$\text{compression ratio} \triangleq \frac{\text{original file size}}{B_1 + B_2}. \qquad (1)$$

We have selected scene classification and pedestrian detection as case studies. We have designed and evaluated 2PP coders for these tasks and investigated performance as a function of compression ratio for different 2PP settings, and the tradeoff between visual quality (PSNR) and classification accuracy at fixed compression ratios.

3. SCENE CLASSIFICATION

We describe natural scenes by dense SIFT visual-word histogram features [6][8], which is also popular for object classification [9]. Lloyd-Max quantizers are learned from training data and used in the predictive coding scheme. For evaluation, we used the Fifteen Scene Categories dataset [6], in

which each category contains pictures of the same type of scene. Note that the dataset is already slightly compressed (with compression ratio around 3) and has a few artifacts.

To control the 2PP output size, $B_1 + B_2$, we employed JPEG 2000 image coder to control B_1 and let the feature dimensions (number of visual words) range over $d = 25, 50, 100, 200, 400$ to control B_2. We also controlled B_2 by employing different quantization levels $k = 2, 4, 8, 16$.

Following [6], 100 images per category were used for training and the rest for testing, and the percentage of correctly classified images, or accuracy, was used as the performance metric.

3.1. Accuracy vs. Compression Ratio

Figs. 3 and 4 show classification accuracy vs. Compression Ratio for $d = 25$ and 200 features, respectively. Each figure has 5 curves, representing different PSNRs: 18.4, 20.2, 21.6, 23.6, 28.7. Each curve was obtained by fixing the number of quantization levels to $k = 2, 4, 8, 12, 16$.

We may view the slope of the curves as a measure of marginal classification accuracy acquired per bit. In Fig. 3 ($d = 25$), the slopes of the curves is approximately -0.32. However, in Fig. 4, ($d = 200$), the slopes are in the range -0.18 to -0.15. In general, the marginal return decreases as d increases.

This can be explained as follows. Since $B_2 = d \log_2 k$ grows faster with k when d is large, B_1 is smaller and the features extracted (predicted) from the compressed image are relatively poor. This makes the feature prediction errors larger and harder to quantize and encode. Therefore, the information loss due to low B_1 reduces the marginal benefits of extra feature bits.

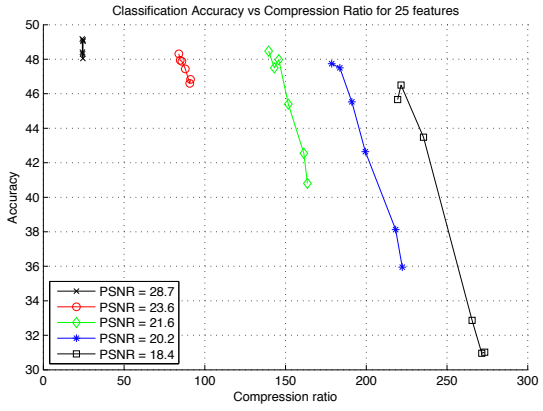


Fig. 3. Accuracy vs. Compression Ratio with Feature Dimension of 25. The performance on uncompressed data is around 49%.

3.2. Accuracy vs. PSNR

Fig. 5 shows the tradeoff between accuracy and PSNR at compression ratio 150. Each figure has 5 curves, representing

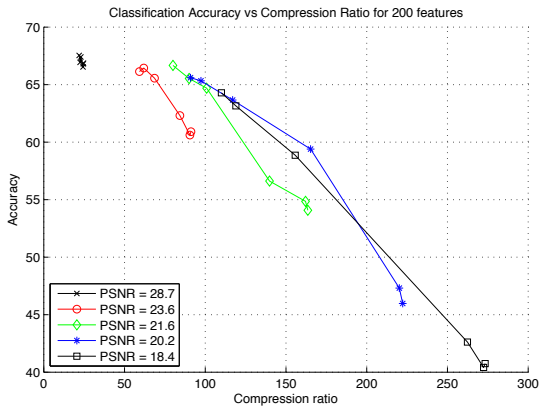


Fig. 4. Accuracy vs. Compression Ratio with Feature Dimension of 200. The performance on uncompressed data is around 67%.

different feature dimensions, $d = 25, 50, 100, 200, 400$. Each curve was obtained by fixing the number of quantization levels to $k = 2, 4, 8, 16$. Fig. 6 shows sample images whose PSNR ranges from 20 to 22.

As seen on the right of the figure, we have a 6% accuracy gain by trading off 0.8 of PSNR, from $d = 50$ and $k = 8$ to $d = 200$ and $k = 16$. In general, the 2PP allows a sharp tradeoff ratio (steep slope) between PSNR and accuracy. Note that at higher compression ratios, using $d = 400$ features the trade from PSNR to accuracy is more costly.

The 2PP coder presents significant advantages compared to the baseline, by substantially improving at a small PSNR loss. The operating point may be selected depending on the user's weighting of visual quality and accuracy.

We also experimented on lower compression ratios (≤ 20) which give higher PSNR (≥ 30) images. The graphs are omitted due to space limitations. In those experiments, classification accuracy does not drop as much ($\leq 1\%$) and the advantages of the 2PP architecture are marginal.

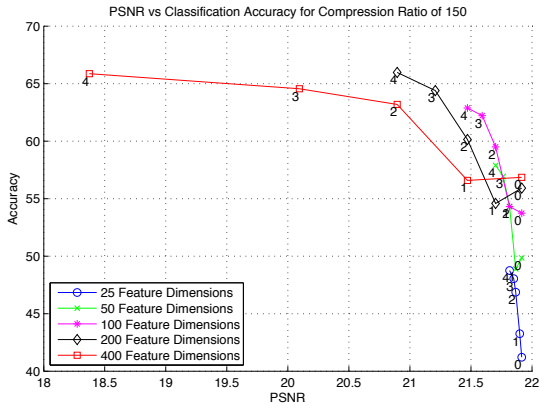


Fig. 5. Average peak noise-to-signal ratio (PSNR) vs. accuracy at compression ratio of 150. Number 0, 1, 2, 3, and 4 represent the baseline, using 2, 4, 8, and 16 quantization levels, respectively.

Fig. 6. Sample images from "bedroom" with PSNR 22, 21.5, 21, and 20, respectively.

4. PEDESTRIAN DETECTION

A pedestrian detection system analyzes video frames and locates pedestrians in the sequence. While pedestrian detection is actively researched in computer vision, all current approaches focus on accuracy but not on robustness to compression. We built an pedestrian detection system with the 2PP architecture based on the Fastest Pedestrian Detection in the West (FPDW) [1] and evaluated the system on the Caltech Pedestrian Dataset.

Following [1], we 1) use integral channel features, including histogram, gradient magnitude, and gradient histogram, 2) train and evaluate pedestrian detectors on every 30th frame, and 3) assess detection performance by log-average miss rate, which is the average of miss rates at nine false positives evenly spaced in log-space in the range from 10^{-2} to 10^{0}. We use H.264 as the baseline video encoder.

To control B_2, we allocate feature bits to the following feature subsets: 1) no features (baseline), 2) all features, 3) color features only, and 4) gradient histogram features only.

4.1. Log-Average Miss Rate vs. Compression Ratio

In this section, we compare log-average-miss-rate improvements between the settings. Log-average miss rate vs. Compression Ratio for the four settings are summarized in Fig. 7.

The log-average miss rate of uncompressed video is 44.5%. Remarkably, H.264 only suffers $5-10\%$ of log-average miss rate at compression ratios up to 900. Even though, the 2PP coder reduces miss rate up to 5% at compression ratios up to 600. Interestingly, while gradient histograms are the most informative integral channel features [7], sending feature bits for gradient histogram features gives the worst performance. One explanation is that different features benefits differently from feature bits. Gradient histogram features may be more robust to H.264, and therefore sending feature bits for histograms are less beneficial.

4.2. Log-Average Miss Rate vs. PSNR

Fig. 8 shows the tradeoff between log-average miss rate and visual quality (PSNR) at fixed compression ratios. We performed experiments on all four settings at compression ratios 300, 400, 500, and 600. The results are summarized in Fig. 8.

We make the following observations: 1) Sending feature bits for color features gives the best tradeoff. One gains 2% of log-average miss rate by paying merely 0.1, 0.3, 0.4, and 0.6 of PSNR at compression ratios 300, 400, 500, and 600, respectively. 2) Again, as discussed in the previous section, sending extra feature bits for color features is more beneficial than sending extra feature bits for histogram or all features.

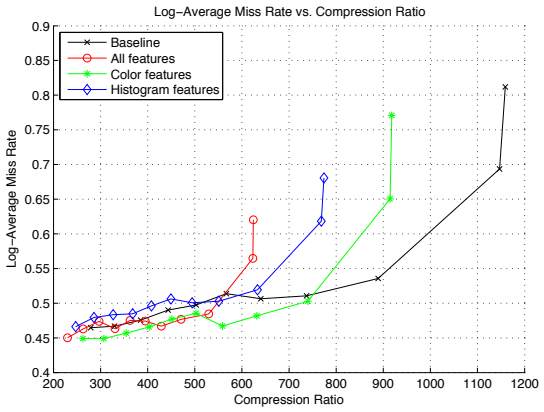


Fig. 7. Log-average miss rate vs. compression ratio for baseline (no feature bits), sending feature bits for all features, color features, and all features.

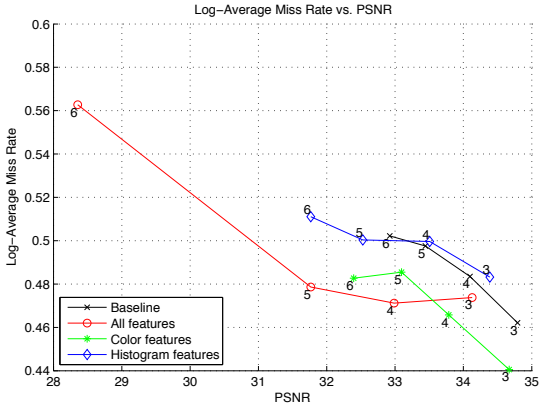


Fig. 8. Log-average miss rate vs. average PSNR at different compression ratios. Number 3, 4, 5, 6 denotes compression ratios 300, 400, 500, 600.

5. CONCLUSION AND FUTURE WORK

We have introduced the 2PP coding architecture, which allows users to pick an operating point and trade off signal reconstruction and content analysis performance according to their preference. We designed and evaluated 2PP coding systems for scene classification and pedestrian detection and demonstrated the merits of the 2PP coder.

For future work, one direction is to explore and design quantizers that exploit correlations between features, such as interframe correlation for videos and cross-feature correlations. Another direction is to compare the 2PP coder with post-processing techniques that remove compression artifacts from the signals or refine signal fidelity by features.

6. REFERENCES

[1] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the West." in *British Machine Vision Conference*, vol. 2, no. 3, Bristol, UK, 2010, pp. 68.1–11.

[2] J. S. Baras and S. Dey, "Combined compression and classification with learning vector quantization," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1911–1920, 1999.

[3] K. O. Perlmutter, S. M. Perlmutter, R. M. Gray, R. A. Olshen, and K. L. Oehler, "Bayes risk weighted vector quantization with posterior estimation for image compression and classification," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 347–360, 1996.

[4] S. Jana and P. Moulin, "Optimality of KLT for high-rate transform coding of Gaussian vector-scale mixtures: application to reconstruction, estimation, and classification," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4049–4067, 2006.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sept 2007.

[6] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. New York, NY, USA: IEEE, 2006, pp. 2169–2178.

[7] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features." in *British Machine Vision Conference*, vol. 2, no. 4, London, UK, 2009, pp. 91.1–11.

[8] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[9] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *2009 IEEE 12th International Conference on Computer Vision*. Kyoto, Japan: IEEE, 2009, pp. 221–228.