LOSSLESS IMAGE COMPRESSION USING CAUSAL BLOCK MATCHING AND 3D COLLABORATIVE FILTERING

Robert Crandall, Ali Bilgin

University of Arizona

ABSTRACT

Predictive coding has proven to be an effective method for lossless image compression. In predictive coding, untransmitted pixels are predicted based on the pixels already available at the decoder. Prediction errors are then compressed by entropy coding, and the original image can be reconstructed exactly at the decoder. More accurate prediction decreases the entropy of the prediction error, allowing for increased compression. Conventional image prediction methods rely on information from the immediate local neighborhood of each pixel. We introduce a novel predictor that leverages non-local structural similarities which have been shown to be effective in image denoising and deblurring applications. Experimental results show that the proposed method achieves state-of-the-art compression performance.

Index Terms— Lossless compression, causal prediction, block matching, collaborative filtering

1. INTRODUCTION

Over the last few decades, substantial progress has been made in estimation methods that adapt to the local structure in image and video data. While conventional regression methods were developed based primarily on the spatial Euclidean distance between pixels, several recent approaches have illustrated that stronger signal adaptivity can be achieved by going beyond spatial proximity. The non-local means (NLM) algorithm [1], which created significant interest in the signal processing community, considers photometric distance between image patches and, for the most part, ignores spatial proximity. The locally adaptive regression kernel (LARK) method [2] uses the photometric differences based on gradients to adapt the shape and size of the regression kernel. The block-matching and three-dimensional filtering (BM3D) method [3] forms 3D clusters of patches based on photometric distance and processes each cluster using 3D filtering. For an excellent perspective on recent methods for adaptive processing of multidimensional data, the interested reader is referred to [4]. While these methods have led to state-of-the-art results in many applications including image and video denoising [3] [5], deblurring [6], and compressive sensing [7], they have not seen wide adoption in image compression applications.

In this work, we propose a method that extends the block-matching and 3D collaborative filtering ideas of BM3D to image prediction and compression. Conventional prediction methods such as Gradient-Adjusted Predictor (GAP) [8] and Median Edge Detector (MED) [9] employ edge-detection based local contexts to adapt between smooth regions and edge areas. More recent adaptive prediction methods optimize prediction coefficients inside a local window [10], [11], [12] using least-squares. Recent literature suggests that stronger adaptivity can be achieved by non-local models. In [13] the authors propose a least-squares predictor based on minimum-description-length (MDL) principles that combines an adaptive local support with nonlocal matching to achieve state-of-the-art results, but with very high computational complexity. Motivated by these recent results, an adaptive predictor which uses sparse representations of 3D clusters of non-local image patches is proposed.

Unlike previous utilization of the BM3D framework in image denoising, deblurring, or compressive sensing applications, image compression requires causality to ensure synchronization between the encoder and the decoder. Thus, we introduce concepts such as *causal* block matching and *time-dependent* 3D collaborative filtering to address the unique requirements of image compression. A novel adaptive threshold selection method designed for 3D collaborative filtering in image compression applications is also introduced. Our method is of lower complexity and thus more practical than the algorithm proposed in [13], but is able to similarly exploit non-local image structure to improve prediction performance. Experiments on test images illustrate that the proposed predictor provides noticeable improvement over existing predictors in the literature. This predictor is then combined with context modeling and arithmetic coding to create a complete lossless image coder. Experimental results show that this coder outperforms conventional lossless image compression techniques such as EDP [12], JPEG-LS [9], JPEG-2000 [14], and CALIC [8]

The remainder of this paper is organized as follows: The causal block matching procedure is described in Section 2. In Section 3, we briefly describe the least-squares procedure used to form an initial prediction and how it is used to form the rectangular blocks that will be used in the 3D groups of Section 4. Section 4 describes the 3D collaborative filtering

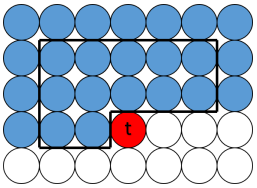

Fig. 1. Current pixel x_t to be predicted (red) and its causal neighborhood $\mathcal{N}_t$ (outlined in black). Blue pixels are known; white pixels have not been transmitted yet and cannot be used in the prediction step.

procedure and how it is used to improve the initial prediction by exploiting non-local structure. Section 4.1 describes a method for adaptive selection of the 3D group threshold parameter. Finally, in Section 5 we demonstrate the improved results of our predictor using a lossless compression codec on a set of standard test images. Conclusions are provided in Section 6.

2. CAUSAL BLOCK MATCHING

At the core of the BM3D algorithm is a block-matching and 3D grouping procedure that assembles groups containing nonlocal structural information about an image into a highly redundant frame representation [3], [15]. In lossless compression applications, only a portion of the image is available for processing at a given time; all pixels that have not yet been processed are unknown to the decoder and cannot be used in prediction. To deal with this limitation, we introduce a *causal* block matching and 3D-grouping procedure, which gives a *time-dependent* representation that contains nonlocal structural information available only from the previously transmitted pixels.

For each pixel to be processed, we compute a k-nearest neighborhood of similar patches in a *causal* way; that is, each patch can only be matched with patches that have already been transmitted to the decoder. Let X be an image of size $N_1 \times N_2$, arranged in raster (row-major) order. The pixel $x_{i,j}$ at row i and column j is indexed by $t = j + N_2 i$, and we identify $x_t := x_{i,j}$. For a given pixel x_t, the causal neighborhood of size T is defined as

$$\mathcal{N}_t = \{x_{t'} | t' < t \text{ and } max(|i - i'|, |j - j'|) \leq T\}$$

That is, $\mathcal{N}_t$ contains all pixels within a square of side length $2T + 1$ around x_t that appear before x_t in raster order. Each neighborhood contains $2T(T + 1)$ pixels, and $\mathcal{N}_t(i)$ gives the i^{th} pixel in a given neighborhood in raster order. The pixel x_t (shown in red) and its causal neighborhood $\mathcal{N}_t$ of size $T = 2$ (outlined in black) are illustrated in Figure 1.

Given two neighborhoods $\mathcal{N}_{t_1}$ and $\mathcal{N}_{t_2}$, let $d(\mathcal{N}_{t_1}, \mathcal{N}_{t_2})$ be some metric measuring distance between the two neighborhoods; here we will use the standard squared Euclidean distance which we label d_2.

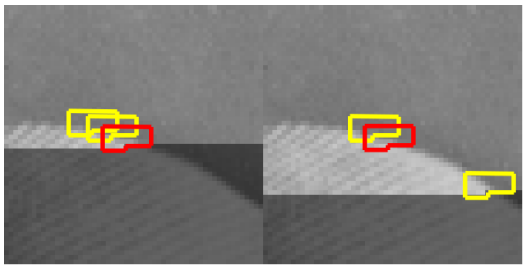

Fig. 2. Causal block-matching process on closeup of Lena image. Neighborhood $\mathcal{N}_{t_1}$ (red patch in both images) and two of its nearest-neighbors at time t_1 (yellow patches in left image) and at a later time t_2 (yellow patches in right image). The available matching region is larger at t_2, and the group $K_{t_1}(t_2)$ is updated to contain this new information.

For each pixel x_τ with $\tau \leq t$, we compute the current k-nearest-neighbors group $\mathcal{K}_\tau(t) = \{\mathcal{N}_{\tau_1}, ..., \mathcal{N}_{\tau_k}\}$, which is the set of patches in the causal region at time t that have minimal distance from the patch $\mathcal{N}_\tau$ under the metric d. These nearest-neighbor groupings change with time, since we may find better matches for a given patch as the causal region expands as illustrated in Figure 2. It is important to emphasize this time-dependent nature of the neighborhood groups as this represents a major difference between our work and the grouping used in earlier BM3D approaches.

3. EXTENSION TO RECTANGULAR PATCHES

The neighborhoods $\mathcal{N}_t$ are non-rectangular. To allow the use of standard rectangular transforms that exploit spatial correlation, we predict the current pixel x_t as well as the "future" pixels $x_{t+1}, ..., x_{t+T}$, so that we obtain a rectangular neighborhood $\mathcal{N}'_t$ containing $(T + 1) \times (2T + 1)$ pixels. The first $2T(T+1)$ pixels of $\mathcal{N}'_t$ are known, while the remaining $T+1$ need to be predicted.

For each non-rectangular neighborhood $\mathcal{N}_\tau$ in the group $\mathcal{K}_t$, we have the corresponding rectangular neighborhood $\mathcal{N}'_\tau$ obtained by adding the next T pixels to $\mathcal{N}_\tau$. Note that many of the pixels needed to form rectangular neighborhoods in group $\mathcal{K}_t$ are within the causal window and can be used directly. In fact, at time t there are exactly $T + 1$ unknown pixels appearing in the neighborhoods $\{\mathcal{N}_\tau : \tau < t\}$, namely the pixels $x_t, ..., x_{t+T}$. We fill the unknown $T + 1$ pixels using a causal predictor function P_t. $P_t(t_0) = \tilde{x}_{t_0}$ is a prediction of the pixel x_{t_0} that depends only on the causal neighbors x_τ with $\tau < t$.

Given a causal predictor P_t, we predict $\tilde{x}_t = P_t(t)$ using the known pixels for $\tau < t$. We then predict $\tilde{x}_{t+1} = P_{t+1}(t + 1)$ using the known pixels in the region $\tau < t$ and the prediction for the current pixel $\tilde{x}_t$. We continue in this fashion until we have the predictions for $\tilde{x}_t$ through $\tilde{x}_{t+T}$. These predicted values are used to extend the non-rectangular

neighborhoods $\{\mathcal{N}_t\}$ to the rectangular neighborhoods $\{\mathcal{N}'_t\}$.

While any causal predictor can be used during this procedure, we have selected to use a standard least-squares predictor for P_t as in [10], [11], [12]. The current pixel x_t is predicted as a weighted sum of the pixels in its causal neighborhood $\mathcal{N}_t$:

$$\tilde{x}_t = P_t(t) = \sum_{i=1}^{2T(T+1)} y(i)\mathcal{N}_t(i).$$

The weighting vector y is formed using least squares. For each of the $2T(T+1)$ pixels in $\mathcal{N}_t$, we form a causal training window of size $2K(K+1)$ centered at that pixel, where we assume $K > T$. These training patches are of different size than the neighborhoods $\mathcal{N}_t$ since $K > T$. We vectorize these training patches and store them in a matrix A. We store the $2K(K+1)$ causal neighborhood around the current pixel x_t in a vector b. We then solve for the minimum-norm solution to the least squares problem $\min \|Ay - b\|_2^2$, which is given by

$$y = (A^T A)^{-1} A^T b.$$

4. REGULARIZATION BY COLLABORATIVE FILTERING

When x_t is the current pixel to predict, we extend its neighborhood $\mathcal{N}_t$ to the rectangular neighborhood $\mathcal{N}'_t$ as described above. For each pixel x_τ with $\tau \leq t$, we have the current k-nearest-neighbors group $\mathcal{K}_\tau(t) = \{\mathcal{N}_{\tau_1}, ..., \mathcal{N}_{\tau_k}\}$, which is the set of patches in the causal region at time t that have minimal distance from the patch $\mathcal{N}_\tau$. If this group $\mathcal{K}_\tau(t)$ does *not* contain the current patch $\mathcal{N}_t$, then we ignore it on the current processing step. If it *does* contain the current patch, i.e. $\tau_j = t$ for some $j \in 1, 2, .., k$, then we extend each patch in the group to a rectangle to get the updated nearest-neighbors group $K'_\tau(t) = \{\mathcal{N}'_{\tau_1}, ..., \mathcal{N}'_{\tau_2}\}$.

By stacking the rectangular nearest-neighbors patches $\mathcal{N}'_{\tau_1}$ through $\mathcal{N}'_{\tau_k}$ and the patch $\mathcal{N}'_\tau$, we obtain a 3D group of pixels of size $(T+1) \times (2T+1) \times (k+1)$, which we label $\mathcal{G}_\tau(t)$. Since the blocks are selected by finding nearest-neighbors to the current patch under the d_2 metric, the 3D groups we build tend to have strong correlations along the third dimension as well as the 2D spatial correlations present in most natural images. We exploit the correlations in all three dimensions to enhance the sparsity of our representation, leading to a better estimate of the current unknown pixel.

For each group $\mathcal{G}_\tau(t)$, we apply a three-dimensional sparsifying transform F, then apply a hard thresholding operator T_λ which sets all elements of the transformed group with magnitude less than λ to zero. Finally, we invert the transform F to obtain the regularized estimate for the group, $\tilde{\mathcal{G}}_\tau(t)$:

$$\tilde{\mathcal{G}}_\tau(t) = F^{-1}(\mathcal{T}_\lambda F(\mathcal{G}_\tau(t)))$$

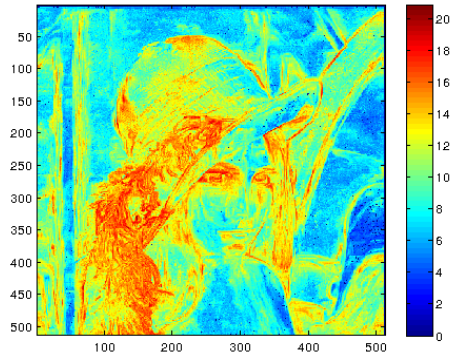

Fig. 3. Map of adaptive thresholds used for Lena image.

For each such group we form a prediction for the current pixel, $x_{t,\tau}$, by extracting the pixel corresponding to x_t from the filtered group.

Since there are many such predictions (one for each pixel x_τ in the causal region whose nearest-neighbors group contains $\mathcal{N}_t$, and one for the current pixel x_t), we combine them by weighted averaging. We set the weight of a given prediction $x_{t,\tau}$, which we label $w_{t,\tau}$, to be inversely proportional to the number of nonzero elements in the thresholded group $\mathcal{T}_\lambda F(\mathcal{G}_\tau(t))$. If the current group $\mathcal{N}_t$ does not appear in $\mathcal{K}_\tau(t)$, we set $x_{t,\tau} = 0$ and $w_{t,\tau} = 0$. The final prediction for x_t is a weighted average of the individual estimates:

$$\tilde{x}_t = \frac{\sum_{\tau \leq t} w_{\tau,t} x_{\tau,t}}{\sum_{\tau \leq t} w_{\tau,t}}$$

4.1. Adaptive Threshold Selection

The optimal threshold λ used in the above procedure is highly dependent on image content and can significantly effect performance. Thus, we propose a simple context-based method for adaptively selecting the threshold parameter λ. Unlike in applications such as denoising and deblurring, in lossless compression we have the advantage that the exact value of each pixel before the current pixel x_t is known. This means that after forming our prediction of the current pixel, we can go back and retrospectively determine what the best threshold *would have been* for the 3D group used in that prediction. We can use previously computed values of this optimal threshold to estimate what the best threshold will be for the current pixel.

The optimal value is computed by sweeping through a predetermined set of values for λ, and selecting the one that gives the best prediction. We store off the transform-domain 3D group at each step for this purpose. In our experiments we use the training set $\lambda = \{0, 1, 2, ..., 20\}$. So, after completing the prediction of x_t, we determine which λ would have given the best prediction, by simply thresholding the saved

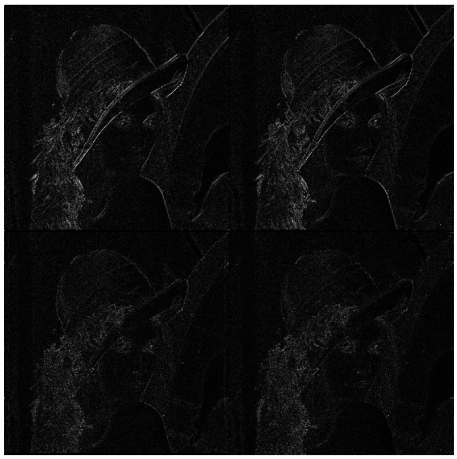

Fig. 4. Comparison of prediction errors from MED (upper left), GAP (upper right), EDP (bottom left), and proposed method (bottom right).

Image	MED	GAP	EDP	Proposed
Lena	4.546	4.387	4.245	**4.125**
Barbara	5.480	5.387	4.546	**4.377**
Boats	5.102	4.977	4.760	**4.670**
Peppers	4.939	4.726	4.511	**4.407**
Baboon	6.896	6.930	6.768	**6.746**
Average	5.393	5.281	4.966	**4.865**

Table 1. First order entropy of prediction errors (bits/pixel)

Image	CALIC	JPEGLS	JPEG2000	Proposed
Lena	4.097	4.236	4.308	**3.990**
Barbara	4.586	4.863	4.786	**4.206**
Boats	4.637	4.796	4.882	**4.539**
Peppers	4.407	4.506	4.626	**4.304**
Baboon	6.590	6.771	6.955	**6.511**
Average	4.863	5.034	5.111	**4.710**

Table 2. Lossless compression performance of different codecs (bits/pixel)

3D group $\mathcal{F}(G_\tau(t))$ and filtering with the hard thresholding operator $\mathcal{T}_\lambda$. The optimal values are saved in an image Λ of the same size as x.

Before performing the collaborative filtering to estimate the current pixel x_t, we estimate the threshold by simply taking the average of the optimal values in Λ corresponding to the causal neighborhood $\mathcal{N}_t$ (i.e. the pixels outlined in black in Figure 1). In Figure 3 we show a map of the estimated threshold values used in the prediction of each pixel. It is important to note the wide spatial variation in these thresholds throughout the image.

5. EXPERIMENTS

We applied the proposed algorithm to compress a test set of 8-bit grayscale images. In these experiments, we set the size of the nearest-neighbor groupings to $k = 45$, and the patch size is set to $T = 2$; thus, the 3D groups have size $3 \times 5 \times 46$. We use a separable 3-dimensional discrete cosine transform (DCT-II) as the sparsifying transform $\mathcal{F}$.

Table 1 presents first-order entropies of the prediction errors obtained using conventional prediction methods as well as the proposed method. The proposed method outperforms MED, GAP, and EDP predictors on all test images. The proposed method provides an average of 9.8%, 7.9%, and 2.0% decrease in the first-order entropy of the prediction errors with respect to MED, GAP, and EDP, respectively. Figure 4 illustrates the prediction error magnitudes obtained using different predictors.

The proposed prediction method was extended to an end-to-end codec by compressing the prediction errors using arithmetic coding with context-modeling. The codec software was written in C++. Table 2 presents a comparison of the lossless compression performance of the codec with other conventional lossless compression methods. As shown in the table, the proposed codec achieves an average of 3.1%, 6.4%, and 7.8% decrease in compressed file size with respect to CALIC, JPEG-LS, and JPEG-2000, respectively. The arithmetic coding and context modeling steps in the codec are similar to the ones used in CALIC. Thus, the improved performance of the proposed codec over CALIC can be attributed to the enhanced prediction performance of the proposed predictor over GAP.

The (unoptimized) codec takes roughly 300 seconds to compress the 512×512 Lena image on an Intel 1.70 GHz processor, or about 200 seconds without adaptive thresholding. CALIC, JPEG-LS, and JPEG-2000 take less than a second.

6. CONCLUSIONS

In this paper, we propose a causal predictor for lossless image coding based nonlocal collaborative filtering methods. Our contributions include the introduction of the time-dependent causal block matching procedure, the application of the 3D collaborative-filtering regularization to improve prediction performance, and the adaptive threshold selection procedure. The proposed predictor achieves state-of-the-art prediction performance. When combined with arithmetic coding and context modeling, the resulting codec provides excellent compression performance at the expense of increased computational complexity.

In future work, we plan to improve the performance of the codec using predictor blending and apply our nonlocal collaborative filtering methods to color images.

7. REFERENCES

[1] Antoni Buades, Bartomeu Coll, and J-M Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 2, pp. 60–65.

[2] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar, "Kernel regression for image processing and reconstruction," *Image Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 349–366, 2007.

[3] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] Peyman Milanfar, "A tour of modern image filtering: new insights and methods, both practical and theoretical," *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 106–128, 2013.

[5] Kostadin Dabov, Alessandro Foi, and Karen Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *Proc. 15th European Signal Processing Conference*, 2007, vol. 1, p. 7.

[6] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian, "BM3D frames and variational image deblurring," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1715–1728, 2012.

[7] Karen Egiazarian, Alessandro Foi, and Vladimir Katkovnik, "Compressed sensing image reconstruction via recursive spatially adaptive filtering," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*. IEEE, 2007, vol. 1, pp. I–549.

[8] Xiaolin Wu and Nasir Memon, "Context-based, adaptive, lossless image coding," *Communications, IEEE Transactions on*, vol. 45, no. 4, pp. 437–444, 1997.

[9] M.J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *Image Processing, IEEE Transactions on*, vol. 9, no. 8, pp. 1309–1324, Aug 2000.

[10] Xiaolin Wu, EU Barthel, and Wenhan Zhang, "Piecewise 2D autoregression for predictive image coding," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*. IEEE, 1998, pp. 901–904.

[11] G. Motta, J.A. Storer, and B. Carpentieri, "Lossless image coding via adaptive linear prediction and classification," *Proceedings of the IEEE*, vol. 88, no. 11, pp. 1790–1796, Nov 2000.

[12] Xin Li and Michael T Orchard, "Edge-directed prediction for lossless compression of natural images," *Image Processing, IEEE Transactions on*, vol. 10, no. 6, pp. 813–817, 2001.

[13] Xiaolin Wu, Guangtao Zhai, Xiaokang Yang, and Wenjun Zhang, "Adaptive sequential prediction of multidimensional signals with applications to lossless image coding," *Image Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 36–42, 2011.

[14] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image compression fundamentals, standards and practice*, Kluwer Academic Publishers, Boston, 2002.

[15] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian, "Image deblurring by augmented lagrangian with BM3D frame prior," in *Workshop on Information Theoretic Methods in Science and Engineering (WITMSE), Tampere, Finland*, 2010, pp. 16–18.