

# WEIGHTED UNIVERSAL TRANSFORM CODING: UNIVERSAL IMAGE COMPRESSION WITH THE KARHUNEN-LOÈVE TRANSFORM

*Michelle Effros*

Dept. of Electrical Eng. (116-81)  
California Institute of Technology  
Pasadena, CA 91125  
effros@systems.caltech.edu

*Philip A. Chou*

Xerox Palo Alto Research Center  
3333 Coyote Road  
Palo Alto, CA 94304  
chou@parc.xerox.com

## ABSTRACT

We introduce a two-stage universal transform code for image compression. The code combines Karhunen-Loève transform coding with weighted universal bit allocation (WUBA) [1] in a two-stage algorithm analogous to the algorithm for weighted universal vector quantization (WUVQ) [2, 3]. The encoder uses a collection of transform / bit allocation pairs rather than a single transform / bit allocation pair (as in JPEG) or a single transform with a variety of bit allocations (as in WUBA). We describe both an encoding algorithm for achieving optimal compression using a collection of transform / bit allocation pairs and a technique for designing locally optimal collections of transform / bit allocation pairs. We demonstrate performance using the mean squared error distortion measure. On a sequence of combined text and gray scale images, the algorithm achieves up to 2 dB improvement over a JPEG style coder using the discrete cosine transform (DCT) and an optimal collection of bit allocations, up to 3 dB improvement over a JPEG style coder using the DCT and a single (optimal) bit allocation, up to 6 dB over an entropy constrained WUVQ with first- and second-stage vector dimensions equal to 16 and 4 respectively, and up to 10 dB improvement over an entropy constrained vector quantizer (ECVQ) with vector dimension 4.

## 1. INTRODUCTION

Universal source coding is the branch of source coding theory that describes the existence and performance of data compression algorithms that are effective not just on a single source, but on a class of sources. The literature of universal source coding has recently yielded a variety of techniques for asymptotically optimal lossy compression, or quantization, of sources with unknown or varying statistics. These schemes yield optimal performance as the data sequence length and quantizer vector dimension grow without bound. Unfortunately,

the computation associated with using a vector quantizer grows exponentially with the vector dimension, and thus in practice, “universal” vector quantizers are typically implemented at very small vector dimensions.

Transform coding provides one method of achieving reasonable complexity codes at high effective vector dimensions. In transform coding, the data is transformed prior to coding. The transformation used is designed to decorrelate the signal and to compact most of its energy into a small number of transform components. By decorrelating the signal, we remove much of the benefit of coding vectors rather than coding scalars. Thus most transform codes follow the transform step by scalar rather than vector quantization. Since the transform components are typically of varying magnitudes and importances, a “bit allocation algorithm” is used to specify the rate at which each component of the transformed data block will be coded. This rate description is typically embodied in a quantization matrix, which describes the coarseness with which each component of the transformed data block will be encoded. The optimal bit allocation strategy is data dependent. Given a transform that is effective at decorrelating and compacting the signal and a bit allocation strategy that is well matched to the statistics of the transformed data, a transform code can achieve an effective vector dimension approximately equal to the transform block size. Further, if the transformation is not computationally complex, the resulting transform code will have a much lower complexity than a vector quantizer with a comparable vector dimension.

The Karhunen-Loève Transform (KLT) is a data-dependent transform that achieves optimal decorrelation (all off-diagonal terms of the transformed data’s covariance matrix are identically equal to zero) and optimal energy compaction [4]. To date, the KLT has not been popular in data compression algorithms due to its data-dependence. Relying solely on traditional techniques, the transform would have to either be chosen in advance (which means that the statistics of the

data have to be known in advance) or computed during the encoding process and communicated to the decoder, which is expensive both in terms of computational complexity and in terms of rate. Given these difficulties with using the optimal transform, source code designers to date have relied primarily on non-optimal, data-independent transform codes. For example, the JPEG image coding standard uses the Discrete Cosine Transform (DCT), which does a good job of approximating the KLT for many natural or smooth images with high correlation [5] but achieves less success with low correlation, high contrast images such as images of text.

We here use universal source coding techniques to design a universal transform code, a transform code whose performance on every source in some broad class of sources approaches the performance that would be achieved if the optimal transform and the optimal bit allocation were used for every source encountered. The algorithm employs a collection of transform / bit allocation pairs. In Section 2 we describe the weighted universal transform coding algorithm. Section 3 contains experimental results.

## 2. THE WEIGHTED UNIVERSAL TRANSFORM CODING ALGORITHM (WUTC)

Let  $x^l = (x_1, \dots, x_l) \in \mathcal{X}^l$  represent an  $l$ -dimensional data vector. If we multiply  $x^l$  by an  $l \times l$  invertible matrix  $T$ , then  $T$  represents a linear transform of the data. If we then scalar quantize the  $i$ th component of the resulting transform vector at rate  $b_i$ ,  $i = 1, \dots, l$ , then  $b^l$  represents a bit allocation strategy. Suppose that we are given some generic scheme for encoding a transformed vector  $Tx^l$  with bit allocation  $b^l$ . Then associated with any transform / bit allocation pair  $(T, b^l)$  is a quantizer  $C = \beta \circ \alpha$  with encoder  $\alpha : \mathcal{X}^l \rightarrow \mathcal{S}$  and decoder  $\beta : \mathcal{S} \rightarrow \hat{\mathcal{X}}^l$  that together map the input space  $\mathcal{X}^l$  of possible data vectors to the output space  $\hat{\mathcal{X}}^l$  of possible reproductions by way of a binary prefix code  $\mathcal{S}$ . Let  $d(x^l, (T, b^l)) = d(x^l, \beta(\alpha(Tx^l)))$  be the total distortion achieved by transforming  $x^l$  with transform matrix  $T$  and then quantizing the resulting transform domain vector with bit allocation  $b^l$ . Similarly, let  $r(x^l, (T, b^l)) = |\alpha(Tx^l)|$  denote the associated rate. (While  $r(x^l, (T, b^l))$  equals  $\sum_i b_i$  on average,  $r(x^l, (T, b^l))$  will vary with  $x^l$  in a variable-rate system.)

We next consider a collection  $(T_1, b_1^l), (T_2, b_2^l), \dots, (T_M, b_M^l)$  of transform / bit allocation pairs. Using the quantization interpretation of a two-stage weighted universal code [3], we consider this collection to be a

*codebook* of transform / bit allocation pairs. Thus we define a "first-stage quantizer"  $\tilde{\beta} \circ \tilde{\alpha}$  with encoder  $\tilde{\alpha} : \mathcal{X}^N \rightarrow \tilde{\mathcal{S}}$  and decoder  $\tilde{\beta} : \tilde{\mathcal{S}} \rightarrow \mathcal{C}$  that maps the input space of possible data blocks  $x^N$  to the output space  $\mathcal{C}$  of possible transform / bit allocation pairs  $(T, b^l)$ . We here assume that  $N$  is a multiple of  $l$ . The first-stage encoder chooses for each  $N$ -block a single transform / bit allocation pair. We then use the chosen pair to encode each of the  $l$ -vectors in  $x^N$ . In transform codes like JPEG,  $N$  equals the size of a single image. For many applications, we may want smaller  $N$  to allow the transform / bit allocation pair to change within a single image. The example of Section 3 uses  $N = l$ , which means that we describe a new transform / bit-allocation pair for each data block.

The total distortion associated with encoding data block  $x^N$  with transform / bit allocation pair  $\tilde{\beta}(\tilde{\alpha}(x^N))$  is

$$d(x^N, \tilde{\beta}(\tilde{\alpha}(x^N))) = \sum_{i=1}^{N/l} d(x_i^l, \tilde{\beta}(\tilde{\alpha}(x^N))).$$

The total rate associated with encoding  $x^N$  includes both the rate associated with describing the transform / bit allocation pair  $\tilde{\beta}(\tilde{\alpha}(x^N))$  and the rate associated with using the chosen pair to describe the data. Thus

$$r(x^N, \tilde{\beta}(\tilde{\alpha}(x^N))) = |\tilde{\alpha}(x^N)| + \sum_{i=1}^{N/l} r(x_i^l, \tilde{\beta}(\tilde{\alpha}(x^N))).$$

Then, using a Lagrangian in order to minimize the distortion subject to a constraint on the rate, the optimal first-stage encoder  $\tilde{\alpha}^*$  for a given collection of transform / bit allocation pairs  $\tilde{\beta}$  is

$$\tilde{\alpha}^*(x^N) = \arg \min_{s \in \tilde{\mathcal{S}}} [d(x^N, \tilde{\beta}(s)) + \lambda r(x^N, \tilde{\beta}(s))]$$

for every  $x^N$ . We call the optimal first-stage encoder a *nearest neighbor* encoder.

Likewise, the optimal first-stage decoder  $\tilde{\beta}^*$  for a given first-stage encoder  $\tilde{\alpha}$  satisfies

$$\begin{aligned} \tilde{\beta}^*(s) = \arg \min_{(T, b^l) \in \mathcal{C}} & E [d(X^N, (T, b^l)) \\ & + \lambda r(X^N, (T, b^l)) | \tilde{\alpha}(X^N) = s] \end{aligned}$$

for every  $s \in \mathcal{S}$ . We call the process of designing the optimal first-stage decoder *decoding to the centroid*.

Given that the KLT maximizes the coding gain over all orthogonal transform codes (e.g., [6, Appendix C]), we here set the transform in the optimal transform / bit allocation pair to the KLT matched to the statistics of the data to be coded. Using this choice we accomplish the optimal decorrelation and energy compaction for the source in operation. The KLT is calculated as follows. For a given index  $s$ , let  $m_s^l$  be the

mean of the  $l$  vectors that mapped to  $s$ , i.e.,  $m_s^l = (l/N) \sum_{i=1}^l N/LE[X_i^l | \hat{\alpha}(X^N) = s]$ . Likewise, let  $V_s$  be the correlation matrix associated with these vectors, i.e.,  $V_s = (l/N) \sum_{i=1}^l N/LE[(X_i^l)(X_i^l)' | \hat{\alpha}(X^N) = s]$ . Then the transform  $T_s^*$  has, in the first row, the eigenvector corresponding to the largest eigenvalue of  $V_s$ , in the second row, the eigenvector corresponding to the second largest eigenvalue, and so on.

Given a transform, the optimal bit allocation may be accomplished by an optimal bit-allocation design algorithm. For example, if the individual components of the transform data block are scalar quantized and then described using independent entropy codes, as in the example of [1], then each term in the quantization matrix may be chosen independently to minimize the Lagrangian performance associated with that component.

The WUTC design algorithm employs an iterative descent technique to minimize the expected Lagrangian performance. We initialize the algorithm with an arbitrary prefix code  $\tilde{\mathcal{S}}$  and collection  $\{\tilde{\beta}(\tilde{s}) : \tilde{s} \in \tilde{\mathcal{S}}\}$  of transform / bit allocation pairs. Each iteration proceeds through three steps which we enumerate below.

- 1 *Nearest Neighbor Encoding.* Optimize the first-stage encoder  $\hat{\alpha}$  for the given first-stage decoder  $\tilde{\beta}$  and prefix code  $\tilde{\mathcal{S}}$ .
- 2 *Decoding to the Centroid.* Optimize the first-stage decoder  $\tilde{\beta}$  for the newly redesigned first-stage encoder and the given first-stage prefix code  $\tilde{\mathcal{S}}$ .
- 3 *Optimizing the Prefix Code.* Optimize the first-stage prefix code  $\tilde{\mathcal{S}}$  for the newly redesigned first-stage encoder  $\hat{\alpha}$  and decoder  $\tilde{\beta}$ . The optimal prefix code  $s^*$  for a given first-stage encoder  $\hat{\alpha}$  and decoder  $\tilde{\beta}$  is the entropy code matched to the probabilities  $P\{\hat{\alpha}(X^N) = s\}$ , for which the ideal codelengths are

$$|s^*| = -\log P\{\hat{\alpha}(X^N) = s\}.$$

Each step of the algorithm decreases the expected Lagrangian performance. Since the Lagrangian performance cannot be negative, the algorithm is guaranteed to converge. The proposed algorithm therefore guarantees a locally optimal solution.

### 3. EXPERIMENTAL RESULTS

In Figure 1, we compare the performance of the WUTC and WUBA algorithms to the performance of single KLT / bit allocation pair and the performance of a DCT-based code analogous to the JPEG algorithm. All transform code implementations discussed here use independent entropy codes to losslessly encode each quantized coefficient. WUTC and WUBA contain 64

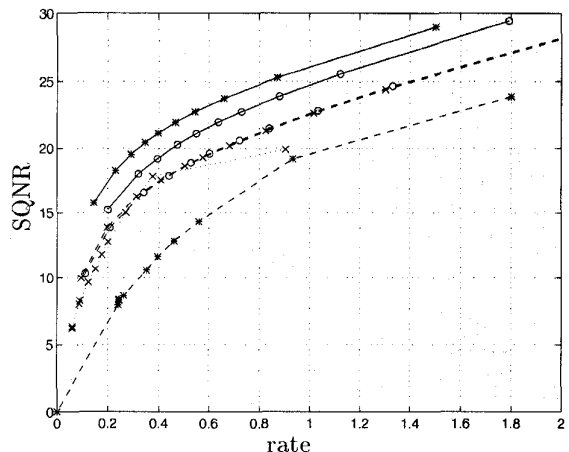
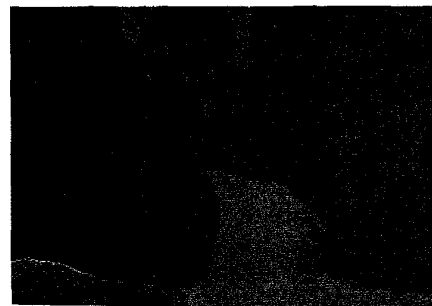


Figure 1: Comparison of SQNR results on a collection of combined text and gray scale images. The systems tested include WUTC (stars with solid line), WUBA (circles with solid line), a single bit allocator system with a single KLT ('x's with dashed line) and with the DCT (circles with dashed line), entropy constrained WUVQ with a maximum of 256 codebooks and 4 code-words per codebook and first- and second-stage vector dimensions equal to 16 and 4 respectively ('x's with dotted line), and ECVQ with vector dimension equal to 4 (stars with dashed line).

transform / bit allocation pairs and bit allocations respectively, and  $N = l = 64$  for all experiments. Each system was trained on a single 2048 pixel by 2048 pixel image scanned from a page of *IEEE Spectrum Magazine* and tested on another page from the same issue. Each page had roughly equal amounts of text and gray scale material. All rates are reported in terms of entropy. WUTC achieves up to 2 dB improvement over WUBA, up to 3 dB improvement over a single bit allocation system, up to 6 dB improvement over WUVQ, and up to 10 dB improvement over ECVQ. The performance curves for all transform coding systems can be expected to shift slightly to the left if they use a lossless code more efficient than independent entropy coding, such as the JPEG code (run-length followed by Huffman coding) or a zerotree code. In Figure 2, we compare the performance of the WUTC to the performance of an optimal transform code with a single transform / bit allocation pair. The image used is independent of the test set.

#### 4. REFERENCES

- [1] M. Effros and P. A. Chou. Weighted universal bit allocation. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, 1995. To appear.
- [2] P. A. Chou. Code clustering for weighted universal VQ and other applications. In *Proc. Int'l Symp. Information Theory*, page 253, Budapest, Hungary, June 1991. IEEE.
- [3] P. A. Chou, M. Effros, and R.M. Gray. A vector quantization approach to universal noiseless coding and quantization. *IEEE Trans. Information Theory*, 1994. Submitted.
- [4] A. Koschman. On the filtering of nonstationary time series. In *Proceedings 1954 Electron. Conf.*, page 126, 1954.
- [5] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Trans. Computers*, 23:90-93, January 1974.
- [6] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, 1993.



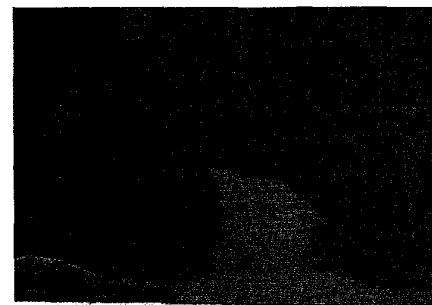
raw the lines of demarcation in  
work?

—Matt Miller,



raw the lines of demarcation in  
work?

—Matt Miller,



raw the lines of demarcation in  
work?

—Matt Miller,

Figure 2: Results of optimal single transform coding and optimal universal transform coding on a single mixed text and image file. The top image is the original. The middle image results from optimal transform coding using a single transform and bit allocation and rate of 0.20 bits per pixel. The bottom image results from universal transform coding using 64 transform / bit allocation pairs and rate 0.23 bits per pixel.