# A GENERALIZED INTERPOLATIVE VQ METHOD FOR JOINTLY OPTIMAL QUANTIZATION AND INTERPOLATION OF IMAGES

F. Fekri, R. M. Mersereau, R. W. Schafer

Center for Signal & Image Processing Georgia Institute of Technology Atlanta, GA 30332

## ABSTRACT

In this paper we discuss the problem of reconstruction of a high resolution image from a lower resolution image by a jointly optimum interpolative vector quantization method. The interpolative vector quantizer maps quantized low dimensional 2x2 image blocks to higher dimensional 4x4 blocks by a table lookup method. As a special case of generalized vector quantization (GVQ), a jointly optimal quantizer and interpolator (GIVQ) is introduced to find the corresponding code books for the low and high resolution image. In order to incorporate the nearest neighborhood constraint on the quantizer and also to obtain the desired distortion in the interpolated image, a deterministic annealing based optimization technique has been applied. With a small interpolation example, we will demonstrate the superior performance of this method over nonlinear interpolative vector quantization (NLIVQ), in which the interpolator is optimized for a given input quantizer.

## 1. INTRODUCTION

Increasing image resolution is of great interest for many imaging applications. The application that motivated this work is the problem of scanning a text image at low resolution (e.g., 300 dpi) and reproducing it by printing at a higher resolution (e.g., 600 dpi). There are many other possible applications including image enlargement, and enhancement of coded images.

Standard approaches of interpolation, such as the bilinear method and spline techniques, while giving better quality in the smooth portions of images, tend to smooth edges and other discontinuities. The notion of generalized interpolative VQ (GIVQ), has evolved from our earlier implementation of nonlinear interpolative VQ (NLIVQ), introduced by Gersho [2]. The quality of the interpolated image by NLIVQ is not promising. Only small improvement in image quality can be achieved by substantially increasing the codebook size, but this increases the encoder complexity and also the susceptibility of the interpolator to data outside the training set. As an alternative to the highly suboptimal NLIVQ method, we propose GIVQ as a nonlinear interpolation technique based on a generalized vector quantization (GVQ). The basic idea of GVQ was proposed in [1] where the mapping of an observable random vector

X to a finite set of estimated values of a random vector Y is treated as a single operation. Successful application of GVQ to the coding of noisy sources [4] encourages us to apply it to the interpolation problem. In generalized interpolative VQ (GIVQ), a low dimensional feature vector X is mapped into a increased dimension signal vector Y producing a high resolution image. While a quantized interpolator is mandated by this method to limit the complexity of interpolation, it may also be usefully employed to transmit fewer data samples through a channel. Then the receiver interpolates the signal vector from the quantized features.

GIVQ is a joint optimization method designed by a deterministic annealing approach [5], [6]. During the design phase of GIVQ, the training vectors are assigned to clusters in a probabilistic way with the probability distributions chosen to be Gibbs distributions. Consequently, the joint optimization of the quantizer and interpolator is formulated within a probabilistic structure in which Shannon's entropy is maximized subject to a constraint on the expected overall distortion in the signal vectors Y. In this paper, we will develop the GIVQ algorithm and demonstrate the comparison between NLIVQ and GIVQ.

#### 2. INTERPOLATIVE VQ

#### 2.1. Nonlinear Interpolative VQ, (NLIVQ)

Suppose that training pairs  $(X_t, Y_t)$  correspond, respectively, to the low and high resolution image vector pairs. NLIVQ has a sequential design procedure. A conventional nearest-neighbor VQ (optimum quantizer) is first designed using a training set of low dimensional input (feature) vectors  $X_t$  without regard to the statistical correlation between  $X_t$  and  $Y_t$ . For this step, to avoid local minimum solutions we used a deterministic annealing based codebook design algorithm [5], [6] instead of the more usual LBG method [3]. Then, the interpolative decoder is designed to be optimal for the given encoder (quantizer). For the squared-error distortion measure, it suffices to let the i'th interpolative code word be the average of all training vectors  $Y_t$  whose corresponding low dimensional vectors have been encoded with index *i*. Consequently, NLIVQ is a suboptimal procedure because the interpolation and quantization are not designed jointly. Besides, the NLIVQ design objective of minimizing the quantization error in X is not matched to the objective of the interpolation problem which is concerned with minimizing the distortion in the output signal vector Y.

This work was supported by Hewlett Packard Corporation.

#### 2.2. Generalized Interpolative VQ, (GIVQ)

GIVQ is based on GVQ [1]. In GVQ the estimate of a random vector Y is formed from a random vector X using an estimator  $h(\cdot)$  that is constrained to take on a finite set of N values. The mapping h(x) is viewed as a generalized vector quantizer (GVQ) that optimally generates a quantized approximation to Y from an observation of X.

GVQ defines a partition of the k dimensional input space  $R^k$  into N regions where N is the codebook size. The partition regions  $R_i$  are defined as:

$$R_i = \{ x \in R^k : h(x) = y_i \} \quad i = 1, 2, \dots, N$$
 (1)

The GVQ design objective is to minimize the distortion of estimating Y by h(x) defined as  $d_f(Y, h(x))$ . As depicted in Fig. 1, the GIVQ consists of an encoder  $\varepsilon$  followed by an interpolative decoder (ID). The encoder maps the low dimensional input vector X to an index  $i \in \{1, 2, \ldots, N\}$ by applying a nearest neighbor rule over a low dimensional codebook C with size N. Then the interpolative decoder looks up an increased dimension signal vector Y in codebook  $C^*$ .

The main difference between GIVQ and NLIVQ is that NLIVQ tries to minimize the distortion in the input space (optimum quantizer), while GIVQ's objective is to minimize the distortion in the output space. Consequently, unlike NLIVQ, the code words of GIVQ in codebook C are not necessarily the 'centroids' of the input space vectors assigned to the same partition. Similar to GVQ, for a mean squared error distortion measure, the optimum GIVQ satisfies the necessary conditions given in [1].

To formulate the GIVQ problem, let the sets  $\{x_i\}_{i=1}^N$ and  $\{y_i\}_{i=1}^N$  be the code words in the low and high dimensional vector space, respectively. Also consider  $d_f(\cdot, \cdot)$  and  $d_i(\cdot, \cdot)$  to be the distortion measure in the feature vector space X and increased dimension vector space Y, respectively. Then, for a given set of training pairs  $T = \{(x_i, y_t)\}$ , we want to optimize the code words  $\{x_i\}_{i=1}^N$  and  $\{y_i\}_{i=1}^N$  so that the total distortion in the signal space Y is minimized:

$$\min_{\{x_i\},\{y_i\}} \{D\} = \min_{\{x_i\},\{y_i\}} \left\{ \sum_{(x_t,y_t)} d_i(y_t, h(x_t)) \right\}$$
(2)

Here  $h(\cdot)$  is GIVQ mapping function consistent with the VQ nearest neighbor encoding rule:

$$\begin{cases} if \quad i = \arg\min_{j} \{ d_f(x_t, h(x_j)) \} \\ then \quad let \quad y_i = h(x_t) \end{cases}$$
(3)

From (2) and (3), it is obvious that this joint optimization problem is not a trivial one. Deterministic annealing DA has been shown by [4] to be a successful method to solve the optimization problem while imposing the nearest neighbor constraint given by (3). Therefore, we have chosen the (DA) approach for the joint optimization of the quantizer and interpolator.

### 3. GIVQ BY DETERMINISTIC ANNEALING

Deterministic annealing (DA) [5] is a probabilistic framework to solve optimization problems. Although convergence



Figure 1: Block diagram of a Generalized Interpolative VQ.

to the global minimum is not assured, it has been shown to be a successful method for avoiding local minima. Besides, DA provides a probabilistic encoding rule for GIVQ that can be exploited to enforce the nearest neighbor constraint on the encoder while minimizing the distortion in the signal space Y. Randomization of the partition subject to a constraint on the encoder entropy results in a Gibbs distribution. This data clustering becomes a fuzzy membership operation in which each vector X is assigned to every cluster by associative probabilities given by the Gibbs' distribution:

$$p(x \in R_j) = \frac{\exp(-\gamma d_f(x, x_j))}{\sum_{k=1}^N \exp(-\gamma d_f(x, x_k))}$$
(4)

where  $\gamma$  is a positive scalar parameter controlling the degree of randomness. By this fuzzy membership, each input vector X belongs to all of the clusters with a probability that depends on its distance from the code words representing those clusters. In this formulation it is assumed that the data assignment is an independent operation ignoring the correlation between adjacent image blocks. However in the continuation of this work , we will incorporate this correlation into the structure of GIVQ. The associated Shannon entropy for this random partitioning is defined by:

$$H = -\sum_{x} p(x \in R_j) \log[p(x \in R_j)]$$
(5)

Now the optimization of GIVQ can be formulated as a minimization of an objective distortion D defined by (2) subject to encoder entropy constraint (5):

$$\min_{\{x_i\},\{y_i\},\gamma} F = \min_{\{x_i\},\{y_i\},\gamma} \{D - \theta \cdot H\}$$
(6)

in which  $\theta$  is a temperature in the annealing process. By this definition, we have an effective objective distortion. The free energy F is minimized to obtain the minimal distortion D in the signal vector Y, while imposing the nearest neighbor encoding rule by gradually reducing the randomness H through a gradually decreasing temperature  $\theta$ . Using (2),(5), and (6) gives:

$$F = \sum_{(x_t, y_t)} \sum_{j=1}^{N} p(x_t \in R_j) \{ d_i(y_t, y_j) + \theta \log[p(x_t \in R_j)] \}$$
(7)

To obtain the necessary optimality conditions for minimizing F, we set to zero the derivatives of F with respect to  $\{x_i\}_{i=1}^N, \{y_i\}_{i=1}^N, \text{ and } \gamma$ . By solving the resulting equations for the case of the squared-error measure in  $d_f(\cdot, \cdot)$  and  $d_i(\cdot, \cdot)$ , each representative in the signal vector Y is defined as the center of mass of the fuzzy cluster:

$$y_{j} = \frac{\sum_{(x_{t}, y_{t})} p(x_{t} \in R_{j}) y_{t}}{\sum_{x_{t}} p(x_{t} \in R_{j})} \quad j = 1, 2, \dots, N$$
(8)

and the corresponding representative in the feature vector space can be derived as:

$$x_{j} = \frac{\sum_{(x_{t}, y_{t})} (F_{x_{j}} - F_{x}) p(x_{t} \in R_{j}) x_{t}}{\sum_{x_{t}} (F_{x_{j}} - F_{x}) p(x_{t} \in R_{j})} \quad j = 1, 2, \dots, N \quad (9)$$

Here  $F_{x_i}$ , and  $F_x$  are defined for any  $(x_t, y_t)$  as:

$$F_x = \sum_{j=1}^{N} p(x_t \in R_j) F_{x_j}$$
(10)

$$F_{x_j} = d_i(y_t, y_j) + \theta \log[p(x_t \in R_j)]$$
(11)

The scalar parameter  $\gamma$  should satisfy the following optimality equation:

$$\frac{\partial F}{\partial \gamma} = \sum_{(x_t, y_t)} \sum_{j=1}^N F_{x_j} p(x_t \in R_j) \cdot \left\{ \sum_{k=1}^N p(x_t \in R_k) d_f(x_t, x_k) - d_f(x_t, x_k) \right\} = 0 \quad (12)$$

Since an analytic form for  $\gamma$  cannot be derived from (12), we use a gradient descent method to optimize F with respect to  $\{x_i\}_{i=1}^N$ ,  $\{y_i\}_{i=1}^N$  and  $\gamma$  at any given temperature  $\theta$ . The algorithm starts at a very high temperature and small value of  $\gamma$  with only one initial representative in both the feature and signal spaces. As the temperature decreases the balance between distortion and entropy changes in (7) toward less randomness. For each gradually decreasing temperature, the representatives and scalar parameter are optimized. This procedure continues until  $\theta$  reaches a critical value given by:

$$\theta_c = 2\lambda_{(C_{xy}^T C_{xy} C_{yy}^{-1})} \tag{13}$$

Here  $\lambda$  is an eigenvalue of its matrix argument,  $C_{xy}$  and  $C_{yy}$  are cross-covariance and covariance matrices defined for each cluster by taking in account the probabilistic data assignment. At the critical temperature, the critical cluster's code word in the higher dimension space Y is split in the direction of the eigenvector corresponding to  $\lambda$ . Also the split in the corresponding feature code word X is initiated along the direction of the projection of that eigenvector into

the feature space. This procedure of decreasing the temperature and optimizing (7) continues as described, and every time the temperature hits the critical temperature of any cluster, the corresponding code word is split. By proceeding in this fashion, the number of code words increases to the desired value. At that point the splitting is stopped and the temperature is driven to zero while the parameters in (7) are optimized. At the limit of  $\theta \to 0$  (and a large value of  $\gamma$ ) the randomness is highly limited while the distortion in signal space is minimized through (7). It is important to note that due to the memory and computation cost, the training of GIVQ over text image data does not using the splitting procedure described above. Instead we randomly initialized both feature space code words  $\{x_i\}_{i=1}^{N}$  and signal space code words  $\{y_i\}_{i=1}^{N}$ , then we optimized (7) with respect to them for each given temperature.

#### 4. SIMULATION RESULTS

In order to demonstrate the advantage of GIVQ over NLIVQ for the interpolation problem, we have performed some experiments. As an example, we consider five-cluster data with uniform distributions. The signal space vectors Y are four dimensional random vectors with 'centroids' of clusters at:

- 1	.1	.1	.6	.7	.4	
	.05	.05	.05	.05	.95	
	.1	.7	.7	.1	.4	
	.95	.95	.95	.95	.05	/

Here each column of the matrix is the exact coordinate of the 'centroid' of one of the clusters. Fig. 2a shows the distributions of the two dimensional feature space X obtained by down-sampling (odd samples) the signal vectors Y. We used these 4500 pairs of (X,Y) to train GIVQ and NLIVQ with codebook size of five. The resulting clustering by NLIVQ for the two-dimensional feature space is shown in Fig. 2b, and the corresponding code words are plotted as bold discs. Since this method partitions the feature space to minimize the distortion in this space, those vectors whose location are in the corners of the center cluster (this cluster is shown by the symbol) are assigned to the neighboring clusters. Thus the signal space code words (shown by the columns of the following matrix) obtained by the Euclidean means of the vectors of the resulting partitions are different from the true 'centroids' given by the columns of the previous matrix.

$$\left(\begin{array}{cccccc} .11 & .12 & .58 & .68 & .39 \\ .16 & .2 & .21 & .17 & .95 \\ .12 & .67 & .67 & .12 & .39 \\ .84 & .8 & .78 & .83 & .05 \end{array}\right)$$

In Fig. 2c, the resulted partitioning of the feature space by GIVQ has been plotted. In this method the feature space is partitioned such that the signal space distortion is minimized. Thus the feature space code words (shown by bold discs in Fig. 2.c) are different from the 'centroids' of the feature space. This produces very accurate (the same as the true center points) signal space code words. We also obtained the resulting total rms distortion of the signal space Y over training set. Comparing the distortion of NLIVQ

(1292.7) with that of GIVQ (584.9) shows remarkably better performance of GIVQ.



(a) Distribution in 2 - D projection of feature space X



(b) Resulting clustering by NLIVQ (2 - D projection).



(c) Resulting clustering by GIVQ (2 - D projection).

Figure 2: (a) A five-cluster uniformly distributed feature space and the partitions produced by: (b) NLIVQ. (c) GIVQ.

#### 5. CONCLUSION

We presented an interpolation method that makes use of a vector quantizer that is jointly optimized for encoder and interpolative decoder. The example illustrates the superior performance of GIVQ over NLIVQ. Although the training procedure is complicated, the interpolation is only a table look up task. For our application we want to interpolate a 300 dpi image to 600 dpi. For training we can scan the training images at 600 dpi and downsample them by two in each dimension to obtain the corresponding low resolution images. With this approach, 2x2 blocks of the low resolution training images correspond exactly to 4x4 blocks of the high resolution training images. Our results for NLIVQ have shown little improvement in image quality by increasing the codebook size from 128 to 256. Now, we are in the process of training GIVQ over text image data using codebook sizes of 128 and 256. In the future, we will consider the problem of the joint optimization of interpolation and halftoning under a common distortion measure. This will enable us to render a gray level interpolated text image to an image suitable for binary devices.

#### 6. REFERENCES

- A. Gersho, "Optimal Vector Quantized Nonlinear Estimation", in IEEE Intl. Symp. On Inform. Theory, page 170, San Antonio, TX, 1993.
- [2] A. Gersho, "Optimal Nonlinear Interpolative Vector Quantization", IEEE Trans. On Comm., COM-38:1285-1287, 1990.
- [3] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. On Comm., COM-28: 84-95, 1985.
- [4] A. Rao, D. Miller, K. Rose, and A. Gersho, "A Generalized VQ Method for Combined Compression and Estimation", In Proc. IEEE Intern. Conf. on Acoustics Speech and Sig. Proc., pp. 2032-2035, Atlanta, May 1996.
- [5] K. Rose, E. Gurewitz, and G. C. Fox, "Vector Quantization by Deterministic Annealing", IEEE Trans. On Inform. Theory, Vol. 38, No. 4, July 1992.
- [6] K. Rose, E. Gurewitz, and G. C. Fox, "A Deterministic Annealing Approach to Clustering", Pattern Recognition Lett., vol. 11, pp. 580-594, 1990.