JOINTLY OPTIMAL INTERPOLATION AND HALFTONING OF TEXT IMAGES USING A DETERMINISTIC ANNEALING VECTOR QUANTIZATION METHOD

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

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

ABSTRACT

This paper presents an approach for the effective combination of interpolation with a halftoning process to reconstruct a high resolution binary image from a lower resolution gray level one. We study a nonlinear interpolative method that maps quantized low dimensional 2×2 image blocks to higher dimensional 4×4 binary blocks using a table lookup operation. In the generalized interpolative VQ (GIVQ) approach [2], we jointly optimize the quantizer and interpolator to find matched codebooks for the low and high resolution images. Then, to obtain a binary interpolative codebook to incorporate digital halftoning with interpolation, we present a binary constrained optimization method using GIVQ. In order to incorporate the nearest neighbor constraint on the quantizer while minimizing the distortion in the interpolated binary image, a deterministic-annealing-based optimization technique is applied. With a few interpolation examples, we demonstrate the superior performance of this method over the NLIVQ method (especially for binary outputs) and other standard techniques e.g., bilinear interpolation and pixel replication.

1. INTRODUCTION

Increasing image resolution is of great interest for many imaging applications such as image enlargement (in medical imaging and digital photography), and enhancement of coded images (in multimedia applications). The application that motivated this work is the problem of gray level scanning of text images at low resolution (e.g., 300 dpi) followed by reproduction at a higher resolution (e.g., 600 dpi) by printing on binary devices.

Standard approaches to interpolation rely on unrealistic assumptions about images. Bilinear interpolation, for example, assumes continuity of the image. Bicubic interpolation imposes continuity on both the image and its first derivatives. Spline techniques may make assumptions about the continuity of even higher order derivatives. In effect, all these standard methods make a lowpass assumption about images. As an alternative to these standard techniques, other interpolation methods exploit information relevant to edge preservation to enhance the quality of the resulting images. Median filtering [4] preserves approximately the sharpness of isolated image transitions. However, the edge preserving property of the median filter does not apply at corners or certain other two dimensional structures. Another class of edge preserving interpolation methods are the directional interpolation techniques [5] which perform one-dimensional interpolation along the minimum variation direction. One problem with directional interpolation techniques is that downsampling in edgy regions introduces uncertainty in the estimate of the orientation and location. This can create large interpolation errors at sharp edges, such as on the contours of the letters in text images. In multiresolution methods [6] the available bandwidth is extended to half of the new sampling rate. These methods exploit the regularity of edges across resolution scales to estimate the high frequency information that is required for the interpolation.

In all of these interpolation techniques, better performance comes at the expense of higher complexity than required by the standard linear methods. The focus of our work is to produce an interpolated halftoned image that is particularly appropriate for binary devices such as inkjet printers. Halftoning is the process of rendering an original continuous tone image (for example 256 gray levels) into a binary image containing only two intensity levels (black and white). Overviews of halftone algorithms without image interpolation have been given in [7]. Interpolation and halftoning, when performed in two separate steps, are suboptimal and computationally expensive. Their separate treatments suboptimal because the interpolation design objective does not match the distortion measure used in the halftoning process. This paper addresses the following question: Can interpolation be combined effectively with halftoning so that the overall computational cost is reduced while maintaining a competitive quality? If the answer to this question is yes, then what is the best method to use for this problem? Vector quantization (VQ) is certainly a reasonable solution. It has the characteristic that it can be used to map a set of observable low resolution image blocks into a collection of binary higher resolution reproduction blocks.

In this paper, we propose a generalized interpolative VQ (GIVQ) method [2] in which the interpolation and halftoning steps are performed jointly under a common distortion measure. The notion of generalized interpolative VQ (GIVQ), has evolved from our earlier implementation of nonlinear interpolative VQ (NLIVQ) [1], introduced by Gersho [8]. As an alternative to the highly suboptimal NLIVQ method, we propose GIVQ as a nonlinear interpolation technique based on deterministic annealing [11]. During the design phase of GIVQ, the training vectors are assigned to clusters in a probabilistic fashion with probability distributions chosen to be Gibbs distributions. Consequently, the joint optimization of the quantization, interpolation, and halftoning can be formulated within a probabilistic structure.

2. IMAGE INTERPOLATION USING VQ

2.1. Interpolative VQ System

Interpolative VQ is a mapping of an observable random vector X to a finite set of estimated values of a random vector Y by table lookup. In our interpolative VQ, the encoder takes 2×2 image blocks of a 300 dpi image as its input and quantizes them based on the partitioning specified by the codebook C. These quantized 2×2 image blocks are mapped to 4×4 image blocks by the interpolative decoder (600 dpi VQ decoder) from codebook C^*. In other words, the interpolative VQ takes low dimensional feature vectors X and maps them into increased dimensional signal vectors Y producing a high resolution image. An interpolative VQ system is fully specified by the two codebooks, one (the encoder codebook of size N labeled C in Fig. 1 containing 2×2 codewords) for the low resolution image blocks and the other (the decoder codebook of size N labeled C^* in Fig. 1 containing 4×4 codewords) for the corresponding high resolution image blocks, and a rule for mapping the feature vectors to the signal vectors.

To train the interpolative VQ system, we require a set of 300 dpi images and their 600 dpi counterparts. Inasmuch as we did not have actual images at the two resolutions that were carefully co-registered, we generated the 300 dpi low resolution training images by lowpass filtering the 600 dpi training images using a simple separable filter with a row and column impulse response given by

$$h(n) = .25\delta(n) + .5\delta(n-1) + .25\delta(n-2) \qquad (1)$$

and downsampling by a factor of two horizontally and vertically. With this approach, 2×2 blocks of the low resolution training images correspond exactly to 4×4 blocks of the originally scanned high resolution training images. These pairs of blocks (x_t, y_t) in the low and high resolution spaces were used to train the system.

2.2. Formulation of Generalized Interpolative VQ Algorithm

Generalized interpolative VQ (GIVQ) is based on GVQ [9]. 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. In the GIVQ a low dimensional feature vector X is mapped into a increased dimension signal vector Y producing a high resolution image. The GIVQ 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

$$R_i = \{x \in R^k \;\; : \;\; h(x) = y_i\} \qquad i = 1, 2, \ldots, N. \qquad (2)$$

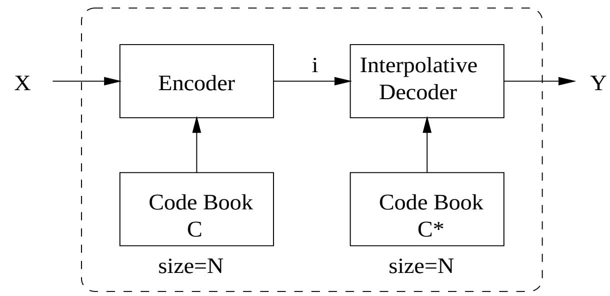

Figure 1: Block diagram of the Interpolative Vector Quantization method for text image interpolation.

The GIVQ design objective is to minimize the error of estimating Y by $h(x)$ defined as $d_i(Y, h(x))$. GIVQ is described by the same block diagram as NLIVQ; i.e. as depicted in Fig. 1, the GIVQ consists of an encoder followed by an interpolative decoder. 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 the corresponding increased dimension signal vector Y in codebook C^*.

The main difference between GIVQ and NLIVQ is that NLIVQ minimizes 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 codewords of the GIVQ in codebook C are not necessarily the 'centroids' of the input (feature) space vectors assigned to the same partition. Like GVQ, for a mean squared error distortion measure the optimum GIVQ satisfies the necessary conditions given in [9].

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

$$\min_{\{x_i\},\{y_i\}} \{D\} = \min_{\{x_i\},\{y_i\}} \left\{ \sum_{(x_t,y_t)} d_i(y_t, h(x_t)) \right\}. \qquad (3)$$

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

$$\begin{cases} if & i = \; \arg\min_j \{d_f(x_t, h(x_j))\} \\ then & let \;\; y_i = h(x_t) \end{cases} \qquad (4)$$

From (3) and (4), it is obvious that the joint optimization problem is not a trivial one. Deterministic annealing DA has been shown by [11] to be a successful method for solving the optimization problem while imposing the nearest neighbor constraint given by (4). Therefore, we have chosen the (DA) approach for the joint optimization of the quantizer and interpolator.

3. OPTIMIZATION BY DETERMINISTIC ANNEALING

3.1. GIVQ DESIGN METHODOLOGY

Deterministic annealing (DA) [11] is a probabilistic framework to solve optimization problems. Although convergence to the global minimum is not assured, it has been shown to be a successful method for avoiding many local minima. 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 [11]

$$p(x \in R_j) = \frac{\exp(-\gamma d_f(x, x_j))}{\sum_{k=1}^{N} \exp(-\gamma d_f(x, x_k))} \quad j = 1, 2, \ldots, N, \quad (5)$$

where γ 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 codewords representing those clusters. In this formulation it is assumed that the data assignment is an independent operation that ignores the correlation between adjacent image blocks. The associated Shannon entropy for this random partitioning is defined by

$$H = -\sum_{x} \sum_{j=1}^{N} p(x \in R_j) \log[p(x \in R_j)]. \quad (6)$$

Now the optimization of GIVQ can be formulated as the minimization of an objective distortion D defined by (3) subject to an encoder entropy constraint (6)

$$\min_{\{x_i\},\{y_i\},\gamma} F = \min_{\{x_i\},\{y_i\},\gamma} \{D - \theta \cdot H\} \quad (7)$$

in which θ 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 θ. Using (3), (6), and (7) gives

$$F = \sum_{(x_t, y_t)} \sum_{j=1}^{N} p(x_t \in R_j)\{d_i(y_t, y_j) + \theta \log[p(x_t \in R_j)]\}. \quad (8)$$

To obtain the necessary optimality conditions for minimizing F, we set to zero the derivatives of F with respect to $\{x_i\}_{i=1}^N$, $\{y_i\}_{i=1}^N$, and γ. 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

$$y_j = \frac{\sum_{(x_t, y_t)} p(x_t \in R_j) y_t}{\sum_{x_t} p(x_t \in R_j)} \qquad j = 1, 2, \ldots, N \quad (9)$$

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

$$x_j = \frac{\sum_{(x_t, y_t)} (F_{x_j} - F_x) p(x_t \in R_j) x_t}{\sum_{x_t} (F_{x_j} - F_x) p(x_t \in R_j)} \qquad j = 1, 2, \ldots, N. \quad (10)$$

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

$$F_x = \sum_{j=1}^{N} p(x_t \in R_j) F_{x_j} \quad (11)$$

$$F_{x_j} = d_i(y_t, y_j) + \theta \log[p(x_t \in R_j)]. \quad (12)$$

The scalar parameter γ should satisfy the following optimality equation

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

To optimize F with respect to $\{x_i\}_{i=1}^N$, $\{y_i\}_{i=1}^N$ and γ at any given temperature θ, we use (9) for the decoder parameters and a gradient descent method for both the scalar and encoder parameters.

3.2. Algorithm Description

The algorithm starts at a very high temperature and small value of γ with only one initial representative in both the feature and signal spaces ($N = 1$). As the temperature decreases, the balance between distortion and entropy in (8) changes toward less randomness. For each gradually decreasing temperature, the representatives and scalar parameter are optimized. This procedure continues until θ reaches a critical value given by [1]

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

Here λ is an eigenvalue of its matrix argument, and C_{xy} and C_{yy} are cross-covariance and covariance matrices defined for each cluster by taking into account the probabilistic data assignment

[1] This formula is derived in [11] for the case in which the feature space and signal space are the same.

$$C_{xy}^j = \frac{\sum_{(x_t,y_t)} p(x_t \in R_j)(x_t - \hat{x_t})(y_t - \hat{y_t})^T}{\sum_{x_t} p(x_t \in R_j)} \quad j = 1, 2, \ldots, N \tag{14}$$

$$C_{yy}^j = \frac{\sum_{(x_t,y_t)} p(x_t \in R_j)(y_t - \hat{y_t})(y_t - \hat{y_t})^T}{\sum_{x_t} p(x_t \in R_j)} \quad j = 1, 2, \ldots, N \tag{15}$$

where $\hat{x_t}$ and $\hat{y_t}$ are the mean values of x_t and y_t, respectively. At the critical temperature, the codeword of the critical cluster in the higher dimension space Y is split in the direction of the eigenvector corresponding to λ. Also the split in the corresponding feature (low dimensional) codeword X is initiated along the direction of the projection of that eigenvector into the feature space. This procedure of decreasing the temperature and optimizing (8) continues as described, and every time the temperature hits the critical temperature of any cluster, the corresponding codeword is split. By proceeding in this fashion, the number of codewords increases to the desired value. At that point the splitting is stopped and the temperature is driven to zero while the parameters in (8) are optimized. In the limit as $\theta \to 0$ (and a large value of γ) the randomness is highly limited while the distortion in the signal space is minimized through (8). It is worth noting that to save memory and computation cost, the training of the GIVQ algorithm does not have to use the splitting procedure (starting from $N = 1$) described above. Instead, it can be trained by using a suboptimal method that randomly initializes both feature space codewords $\{x_i\}_{i=1}^N$ and signal space codewords $\{y_i\}_{i=1}^N$, then optimizes (8) with respect to the parameters for each given temperature.

3.3. Binary Constrained GIVQ Algorithm for Halftoning

Our approach for combined interpolation and halftoning of text images is to design an interpolative VQ whose encoder codewords $\{x_i\}_{i=1}^N$ are gray level and decoder codewords $\{y_i\}_{i=1}^N$ are binary. This would be a special case of GIVQ in which a gray level low dimensional feature vector X is mapped into a binary increased dimensional signal vector Y producing a high resolution binary image. In the following, we present a simple extension of the GIVQ algorithm for the design of jointly optimal encoder and interpolative decoder codebooks.

Here we assume that the range of image intensities is $[0, 1]$. We use the previous training set (x_t, y_t) in the low and high resolution space except that y_t is a binary version of the high dimensional signal vectors. One method of imposing binary constrained codewords $\{y_i\}_{i=1}^N$ into the optimization of free energy F in (8) is to use a simple thresholding operation, which gives the nearest binary codeword according to the mean-squared-error distortion measure. Consequently, we optimize F with respect to $\{x_i\}_{i=1}^N$, $\{y_i\}_{i=1}^N$ and γ at any given temperature θ as it is described in previous section, then we threshold the codewords $\{y_i\}_{i=1}^N$ and continue with the optimization of F at the next temperature. By this iterative technique the initial binary codewords are improved through the deterministic annealing process. It is important to note that both the decoder codewords $\{y_i\}_{i=1}^N$ and the data distribution of the signal vectors Y in this case lie on a vertices of a hypercube. Consequently, the free energy F takes on only discrete values as the annealing process proceeds. Thus, the optimization by a gradient descent method is easily trapped in a local minimum and is sensitive to the initial location of the encoder and decoder codewords. Hence, it is helpful to start with only one codeword and use the splitting procedure described in the previous section to avoid the local minima.

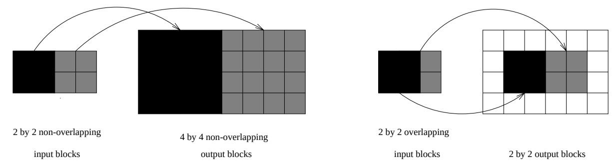

Figure 2: Illustration of context and non-context block mapping.

4. EXPERIMENTAL RESULTS

To demonstrate the successful interpolation of the text images using interpolative VQ, we conducted some experiments. As mentioned earlier, we proposed GIVQ mainly for simultaneously interpolating and halftoning of text images. We trained a binary constrained GIVQ and an NLIVQ system over a training set consisting of a binary high resolution (600 dpi) and its corresponding gray level low resolution (300 dpi) image. Both the NLIVQ and GIVQ systems were trained using 8 point Times New Roman font text then tested using the Courier font at the same size. In this experiment, only a codebook of size 64 was designed.

Vector quantization by itself exploits the inter-pixel correlation that exists within the 2×2 low resolution image block, but it does not take advantage of the correlation between adjacent image blocks. One approach to improve the quality of the interpolated images is to use context instead of encoding the input blocks independently. One possible solution is to increase the size of the image blocks, but this requires a very large codebook size to improve the performance, and this solution is not attractive because of the computational cost required for its implementation. As shown in Fig. 2, a more feasible solution is to map overlapping 2×2 input blocks into 4×4 output blocks and then to extract the four non-overlapping center pixels from the 4×4 output image blocks in the codebook C^*. In this method, each 2×2 input block overlaps three other input blocks. Consequently each pixel in the low resolution image is used four times in the interpolation process.

Fig. 3(a) and (f)[2] show the 300 dpi gray level image and the desired 600 dpi binary image, respectively. The results

[2]The font is magnified five times by sample pixel replication so that fine differences can be readily observed

of interpolation from 300 dpi to a binary 600 dpi image using the pixel replication method, the bilinear interpolation technique, and the NLIVQ algorithm are presented in Fig. 3(b), (c) and (d), respectively. The binary images are produced by thresholding the gray level high resolution images generated by these methods. Furthermore, Fig. 3(e) shows the result of interpolation from 300 dpi to a binary 600 dpi image using the binary constrained GIVQ system. By inspection of these figures, we notice that the image generated by pixel replication is very jagged, and suffers from blocking effects. Bilinear interpolation results in fat characters. Moreover, the serifs on the letters are filled in by this method. It is also noticeable that thresholding NLIVQ produces burst errors. Comparison of the images shows that the GIVQ produces the binary image that is closest to the ideal binary image.

5. CONCLUDING REMARKS

In this paper, we presented two algorithms for text image resolution enhancement. Although the training procedure is computationally intensive, the interpolation requires only a table lookup. We proposed the generalized interpolative vector quantization (GIVQ) method to design jointly optimal codebooks for the encoder and decoder. Finally, we solved the problem of the joint optimization of interpolation and halftoning under a common distortion measure by using a binary constrained deterministic annealing algorithm. Consequently, by this method, the interpolation and halftoning of images are carried out in a single step by a simple table lookup operation. Our preliminary results showed better performance of GIVQ over NLIVQ for binary images.

6. REFERENCES

[1] F. Fekri, R. M. Mersereau, and R. W. Schafer, “Enhancement of Text Images Using a Context Based Nonlinear Interpolative Vector Quantization Method”, Proc. IEEE Intern. Conf. on Image Proc., 1998.

[2] F. Fekri, R. M. Mersereau, and R. W. Schafer, “A Generalized Interpolative VQ Method for Jointly Optimal Quantization and Interpolation of Images”, Proc. IEEE Intern. Conf. on Acoustics, Speech, and Sig. Proc., vol.5, pp. 2657–2661, 1998.

[3] A. Aldroubi, M. Unser, and M. Eden, “Cardinal Spline Filters: Stability and Convergence to the Ideal Sinc Interpolator”, Signal Proc., vol. 28, pp. 127–138, 1992.

[4] B. Zeng and A. N. Venetsanopoulos, “A Comparative Study of Several Nonlinear Image Interpolation Schemes”, Proc. of SPIE, vol. 1818 , 1992, pp. 21–29.

[5] S. D. Bayrakeri and R. M. Mersereau, “A New Method for Directional Image Interpolation”, Proc. IEEE Intern. Conf. on Acoustics, Speech, and Sig. Proc., pp. 2383–2386, 1995.

[6] S. G. Chang, Z. Cvetkovic, and M. Vetterli, “Resolution Enhancement of Images Using Wavelet Transform Extrema Extrapolation”, Proc. IEEE Intern. Conf. on Acoustics, Speech, and Sig. Proc., pp. 2379–2382, 1995.

[7] R. A. Ulichney, Digital Halftoning, Cambridge, MA: MIT press, 1987.

[8] A. Gersho, “Optimal Nonlinear Interpolative Vector Quantization”, IEEE Trans. on Comm., 38:1285–1287, 1990.

[9] A. Gersho, “Optimal Vector Quantized Nonlinear Estimation”, in IEEE Intl. Symp. on Inform. Theory, p. 170, 1993.

[10] Y. Linde, A. Buzo, and R. M. Gray, “An Algorithm for Vector Quantizer Design”, IEEE Trans. On Comm., 28:84–95, 1980.

[11] K. Rose, E. Gurewitz, and G. C. Fox, “Vector Quantization by Deterministic Annealing”, IEEE Trans. On Inform. Theory , Vol. 38, pp. 1249–1258, No. 4, July 1992.

(a)Blurred 300 dpi 8 point Courier font.

(b) High resolution binary image obtained by using pixel replication and thresholding.

(c) High resolution binary image obtained by using bilinear interpolation and thresholding.

(d) High resolution binary image obtained by the NLIVQ method and thresholding.

(e) High resolution binary image obtained by binary constrained GIVQ.

(f) Ideal 600 dpi 8 point Courier font.

Figure 3: Comparison of the results of the standard interpolation and halftoning techniques with those of the GIVQ and NLIVQ systems with codebook size of 64 trained on the 8 point Times New Roman font.