Multiresolution Tree Structured Vector Quantization

Jia Li Navin Chaddha Robert M. Gray

Information Systems Lab, Electrical Engineering Department
Stanford University, Stanford, CA 94305
E-mail: jiali@isl.stanford.edu, navin@sleep.stanford.edu, gray@isl.stanford.edu

Abstract

In some applications of progressive image transmission, the images are viewed at several resolutions with fewer bits at lower resolutions. A multiresolution tree structured vector quantizer is developed to produce an embedded code, so that the quality of the image is optimized for the corresponding resolution at any number of bits. The resolution at which the image is viewed given a particular number of bits is determined by the specific decoder. The multiresolution tree structured vector quantizer presented in the paper generates the codebook by greedy tree growing, which is an extension of the generalized BFOS algorithm. The tree is grown one step further by splitting the node which will yield the best ratio of the change in distortion at the corresponding resolution of current bit rate to the change in rate. The decoder has codewords of all resolutions obtained by optimal centroiding for a given resolution and a given encoder partition. The encoding of an image is essentially the same as BFOS algorithm and the difference is that instead of having a fixed distortion measure, the distortion measure is defined for the corresponding resolution at a particular bit rate. This algorithm is compared with the generalized BFOS algorithm for image quality at different resolutions.

I Introduction

Multiresolution image transmission and storage is becoming more popular with the rapid increase of graphical applications on computer networks. Many art museums now have their collection of masterpieces on line. The huge number of images puts a heavy load on transmission and storage. To speed up access to the images, they are often transmitted and stored in an embedded way at several resolutions. For browsing, for example, the images are usually stored as icons, which are spatially scaled small versions of the originals. Users interested in a particular image can retrieve the larger and more detailed version of the image. In this paper, we refer to the different sizes to which an image is scaled as resolutions. Together scaled versions of the image are called the multiresolution representation of the image. An image of lower resolution is viewed in a smaller size. Our goal is to design an embedded code so that the bit stream first optimizes the quality of the image in low resolution and switches to optimization of the quality in higher resolutions as more bits arrive. This code will allow progressive transmission of the images. The bits used to represent the small size images can help recover the larger and detailed form of the images because of the embedding property of the code.

The successive approximation nature of the tree structured vector quantizer [2, 3] makes it suitable for progressive code design. In previously developed TSVQ algorithms, the tree is grown to minimize the distortion at one resolution, which means the measure of distortion is fixed. In the multiresolution TSVQ algorithm developed here, the tree is grown to minimize distortion defined at the resolution corresponding to the current bit rate. The bit rates at which to switch the goal of minimization is determined by the user. Within one resolution, we grow the tree by an extension of the BFOS algorithm [4], which is referred to as greedy growing algorithm in [5, 1].

We begin with an introduction to the greedy growing TSVQ algorithm. Section III describes multiresolution TSVQ algorithm. Section IV shows that the tree designed by the multiresolution algorithm can be used to progressively transmit images at a fixed resolution with minor degradation compared with the tree designed at that one resolution. The simulation results are presented in section V.

II Greedy Growing TSVQ Algorithm

In the greedy growing algorithm, the tree is designed one node at a time. An unbalanced tree is grown from scratch and the path map of the unbalanced tree is used as the code. To optimally trade off rate and distortion, an "impurity function" is defined to measure the quality or penalty of a particular node in a tree. The "goodness" of a split is defined as the decrease in node impurity and is given by

$$\triangle i(s,t) = i(t) - p_L i_L(t) - p_R i_R(t)$$

where $i(t)$ is the impurity of the node t, $i_L(t)$ and $i_R(t)$ are the impurities of the left child and right child of node t respectively, p_L is the proportion of the samples in node t that go to the left child and p_R is the proportion that go to the right child. The node with the highest goodness of split is split. When the impurity is taken as distortion, which is the case in our algorithm, $\triangle i(s,t)$ is the ratio of decrease in distortion to change in rate.

Readers are referred to [5] for more detailed description of the algorithm and the simulation results.

III Multiresolution TSVQ Algorithm

First, we use an example to describe the multiresolution representation of an image. Suppose a 512×512 image, denoted by $x(i,j), i,j = 0,...,511$, is shown at two resolutions. In the first resolution, it is shown as a 256×256 image, denoted by $y(i,j), i,j = 0,...,255$, where $y(i,j) = \frac{1}{4}(x(2i,2j)+x(2i+1,2j)+x(2i,2j+1)+x(2i+1,2j+1))$. The second resolution is the full image. If we form 2×2 subblocks of the image as the vectors to quantize, only a feature value of the original vector $\mathbf{v}$ is shown at the first resolution. In the example, the feature value is the average of all the components of $\mathbf{v}$. To build a mathematical model, we suppose that the image at full resolution is represented by sequence of vectors $(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)$. At lower resolutions, it is represented by $(\mathbf{f}(\mathbf{v}_1), \mathbf{f}(\mathbf{v}_2), ..., \mathbf{f}(\mathbf{v}_n))$, where $\mathbf{f}(\mathbf{v})$ is a vector valued feature function of $\mathbf{v}$ and the dimension of $\mathbf{f}(\mathbf{v})$ is lower than that of $\mathbf{v}$. In the more general case, the image at low resolution is given by sequence $(\mathbf{v}'_1, \mathbf{v}'_2, ..., \mathbf{v}'_n)$, where $\mathbf{v}'_i$ with lower dimension has strong correlation with $\mathbf{v}_i$, but is not necessarily a function of $\mathbf{v}_i$. An example of this is the wavelet decomposition of an image. The images at several resolutions are the base band images after different levels of wavelet decomposition. In the algorithm presented below and the simulation, we used the former narrow sense of multiresolution, but the algorithm can also be applied to the more general case just by changing $\mathbf{f}(\mathbf{v}_i)$ to $\mathbf{v}'_i$.

The switching rate r from resolution 1 to resolution 2, which is also referred as the target rate for resolution 1, is determined by the user. In the process of growing the tree, when the average rate per vector is below r, the distortion for every vector $\mathbf{v}$ is defined as $\|\mathbf{w} - \mathbf{f}(\mathbf{v})\|^2$, where $\mathbf{w}$ is the codeword to which $\mathbf{f}(\mathbf{v})$ is quantized. When the tree is grown to some depth, the average rate per vector surpasses r and at this point, the distortion for every vector $\mathbf{v}$ is redefined as $\|\hat{\mathbf{v}} - \mathbf{v}\|^2$, where $\hat{\mathbf{v}}$ is the codeword to which $\mathbf{v}$ is quantized.

Suppose the number of resolutions is N and the training vector sequence is $(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)$.At resolution j, the sequence $(\mathbf{f}_j(\mathbf{v}_1), \mathbf{f}_j(\mathbf{v}_2), ..., \mathbf{f}_j(\mathbf{v}_n))$ is viewed. The switching rate (bit per vector) from resolution j to $j+1$ is r_j. The maximum rate to grow the tree is denoted as TargetRate. The algorithm is stated as below.

1. $j = 1, \quad \text{RATE} = 0.0$
2. Design root node (For the MSE distortion measure, just take the mean of the entire sequence $(\mathbf{f}_1(\mathbf{v}_1), \mathbf{f}_1(\mathbf{v}_2), ..., \mathbf{f}_1(\mathbf{v}_n))$). Set the designing resolution of root node to 1.
3. Define impurity function at resolution j for every node t as
$$i(t) = (\sum_{l=1}^{n(t)} \|\mathbf{f}_j(\mathbf{v}_{(l)}) - \mathbf{w}_j(t)\|^2) p(t)$$
where $\mathbf{v}_{(l)}$'s are the samples that go to node t, $p(t)$ is the proportion of $\mathbf{v}_{(l)}$'s to the total number of vectors and $\mathbf{w}_j(t)$ is the codeword at node t, i.e., the centroid of $\mathbf{f}_j(\mathbf{v}_{(l)}), \quad l = 1, ..., n(t)$.
4. Split the node with the maximum $\triangle i(s,t)$. Denote the node by t_{max}. Set the designing resolutions of the left and right child of t_{max} to j. Calculate $\mathbf{w}_m(t_{max,L})$ and $\mathbf{w}_m(t_{max,R}), \quad m = 1, 2, ..., N$.
5. RATE $+ p(t_{max}) \to$ RATE, where $p(t_{max})$ is the proportion of vector that go to node t_{max}.
6. if RATE $>$ TargetRate, end.
7. if RATE $\geq r_j$,
$j+1 \to j$, set the designing resolution of all the leaves of the tree to j, go back to 3; else, go to 4.

The progressive encoding process is as follows.

1. $j = 1$, RATE $= 0$, BITS $= 0$.
2. Set the node $t(\mathbf{v}_i)$ of all the samples $\mathbf{v}_i, i = 1, ..., n$ to root node.
3. For $i = 1 : n$
If $t(\mathbf{v}_i)$ is not leaf node
   (a) Suppose the designing resolution of node $t(\mathbf{v}_i)$ is j
   if $\|\mathbf{f}_j(\mathbf{v}_i) - \mathbf{w}_j(t_L(\mathbf{v}_i))\| < \|\mathbf{f}_j(\mathbf{v}_i) - \mathbf{w}_j(t_R(\mathbf{v}_i))\|$, $t_L(\mathbf{v}_i) \to t(\mathbf{v}_i)$, output bit 0. Otherwise, $t_R(\mathbf{v}_i) \to t(\mathbf{v}_i)$, output bit 1.
   (b) BITS $+ 1 \to$ BITS, RATE $=$ BITS$/n$
   (c) If RATE $>$ TargetRate, break
4. If RATE $<$ TargetRate, go back to 3, otherwise, end.

Given the designed tree structure, the decoder can refine the image at the required resolution bit by bit. After reading in every bit, the decoder knows which node the current vector $\mathbf{v}_i$ goes to. Suppose the node is t and the resolution at which to show the image corresponding to current bit rate per vector is m. The decoder will replace $\mathbf{f}_m(\mathbf{v}_i)$ by $\mathbf{w}_m(t)$. After every bit is read in, the decoder will decide whether to switch to higher resolution according to user determined parameters.

IV Compatibility with One Resolution TSVQ

In the multiresolution TSVQ algorithm, the tree design and the encoding of the image are targeted to multiresolution viewing of the image. However, the decoder can still recover the highest resolution image starting from the first bit in the stream instead of waiting until a certain number of bits have been read in. The reason is that for every node t, the centroids at all the resolutions $\mathbf{w}_m(t), m = 1, 2, ..., N$ are calculated. The decoder can replace $\mathbf{v}_i$ by $\mathbf{w}_N(t)$ when $\mathbf{v}_i$ goes to node t. The next section will compare the performance of multiresolution TSVQ and single resolution TSVQ designed for the highest resolution and will show that the PSNR at the highest resolution by multiresolution TSVQ is better when the bit stream is partly read in.

V Experiment and Results

In the experiment, we used 512×512 CT chest scan images as training image and test image. We assume that the images are viewed at three resolutions. The highest resolution is the full image. The second resolution shows the image at $\frac{1}{4}$th of the original size. Every pixel in the second resolution is the average of a 2×2 block in the original image. The lowest resolution shows the image at $\frac{1}{16}$th of the original size and every pixel in this resolution is the average of a 4×4 block in the original image. The vector $\mathbf{v}_i$ to be quantized has dimension 64 and is formed by the pixel intensities in 8×8 blocks.

In Figure 1, the PSNRs of the compressed test image at all three resolutions by multiresolution TSVQ and one resolution TSVQ are compared. In the multiresolution TSVQ tree design, 5.0 bits per vector (0.078bpp) is the switching rate from resolution 1 to 2, 9.0 bits per vector (0.141bpp) is the switching rate from resolution 2 to 3 and 15.0 bits per vector (0.23bpp) is the final rate of design at resolution 3. In the one resolution TSVQ tree design, the distortion at resolution 3 (full view of the image) is the minimization goal. Four test images are used and the results are similar. Hence in the graphs, we only show the result from one test image.

For comparison, Figure 2 shows the original test image at the three resolutions and the compressed image at the target rates of the corresponding resolutions.

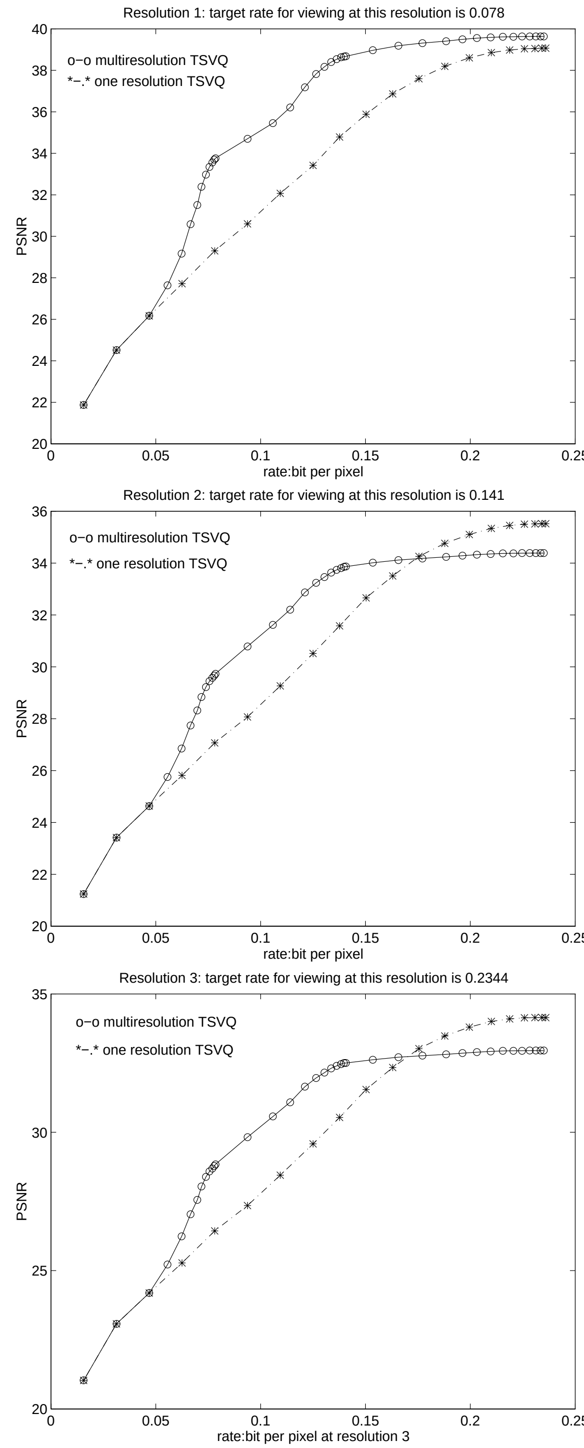
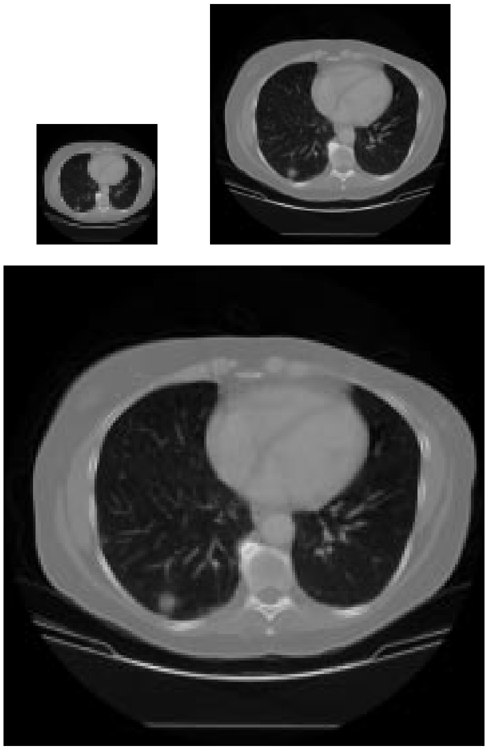


Figure 1. Comparison of multiresolution TSVQ and one resolution TSVQ



Original Test Image at the Three Resolutions

0.078bpp 0.141bpp

0.23bpp

Figure 2. Compressed image versus original image

From Figure 1, we can see that below the target viewing rates of resolution 1 and 2, the PSNRs by multiresolution TSVQ are considerably better than those by normal one resolution TSVQ. At 0.078bpp, the target rate for resolution 1, multiresolution TSVQ is about $5dB$ better. At 0.141bpp, the target rate for resolution 2, multiresolution TSVQ is around $2dB$ better. Even at the full resolution, which is the resolution optimized by the normal TSVQ, the performance of normal TSVQ is not always better than that of the multiresolution TSVQ. When the rate is below 0.16bpp, the multiresolution TSVQ outperforms normal TSVQ. At the final rate 0.23bpp, the multiresoltion TSVQ only loses about $1dB$ to normal TSVQ. We tried different vector dimensions and different number of resolutions in the simulation. The common trend is that multiresolution TSVQ is considerably better at the low resolutions at the corresponding target rates and its performance at the full resolution is surpassed by normal TSVQ when the rate is high. But the PSNR loss for the multiresolution TSVQ is usually within $3dB$.

References

[1] E. A. Riskin and R. M. Gray. A Greedy Tree Growing Algorithm for the Design of Variable Rate Vector Quantizers. In *IEEE Trans. on Signal Processing*, volume 39, no. 11, November 1991.

[2] E. A. Riskin, T. Lookabaugh, P. A. Chou, and R. M. Gray. Variable Rate Vector Quantization for Medical Image Compression with Applications to Progressive Transmission. In *Proceedings of Medical Imaging III: Image Capture and Display*, pages 110–120, 1989.

[3] E. A. Riskin, T. Lookabaugh, P. A. Chou, and R. M. Gray. Variable Rate Vector Quantization for Medical Image Compression. In *IEEE Trans. on Medical Imaging*, Sep 1990.

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees. The Wadsworth Statistics/Probability Series*. Belmont, California: Wadsworth, 1984.

[5] E. A. Riskin. *Variable Rate Vector Quantization of Images*. PhD thesis, Stanford University, Stanford, CA 94305, May 1990. Also available at Stanford University, Terman Engineering Library.