Entropy- and Complexity-constrained Classified Quantizer Design for Distributed Image Classification

Hua Xie and Antonio Ortega
Integrated Media Systems Center, Department of Electrical Engineering-Systems
University of Southern California,
Los Angeles, CA 90089
Email: huaxie@sipi.usc.edu, ortega@sipi.usc.edu

***Abstract*—**
In this paper, we address the issue of feature encoding for distributed image classification systems. Such systems often extract a set of features such as color, texture and shape from the raw multimedia data automatically and store them as content descriptors. This content-based metadata supports a wider variety of queries than text-based metadata and thus provides a promising approach for efficient database access and management. When the size of the database becomes large and the number of clients connected to the server increases, the feature data requires a significant amount of storage space and transmission bandwidth. Thus it is useful to devise techniques to compress the features. In this paper, we propose an optimal design of a classified quantizer in a rate-distortion-complexity optimization framework. A Decision Tree Classifier (DTC) is applied to classify the compressed data. We employ the Generalized Breiman, Freidman, Olshen, and Stone (G-BFOS) algorithm to design the optimal pre-classifier, which is a pruned subtree of the decision tree, and to perform the optimal bit allocation among classes. The optimization is carried out based not only on a rate budget, but also on a coding complexity constraint. We illustrate this framework by showing a texture classification example. Our results show that by using a classified quantizer to encode the features, we are able to improve the percentage of correct classification by 11%, as compared to using a quantizer without preclassification at the same rate. This improvement in classification also leads to a reduction of the number of images transmitted between server and client.

I. Introduction

In multimedia classification applications, such as content-based retrieval of image/video data [1], multiple features(color, texture, shape, etc.) are extracted from the query signal and compared with the database to find the best match. There are several options available to design the system. One option is to keep a full-fledged classification engine locally at each client and then contact the server only once the best match images have been identified. Obviously this method is too complex to be applicable to clients which have limited storage space and processing power. Another option is to have the image/video compressed and sent to the remote server, where it will be classified. This second method may be problematic if bandwidth is limited and the server response may be slow when a large number of clients are connecting and sending requests to the server at the same time. We propose an intermediate solution where features are extracted and compressed at the client, sent through the link and classified remotely at the centralized server. We call this method distributed classification.

This research has been funded in part by NASA under grant AIST-0122-0005 and by Integrated Media Systems Center, a National Science Foundation Engineering Research Center.

There are three factors that should be taken into consideration in designing distributed classification systems: bandwidth requirement for transmitting the feature data, classification accuracy, and the computational complexity imposed on the clients. In real applications, various feature sets (color, texture, shape, motion, etc) are extracted to represent the content information of images/videos. Besides the global feature information, local feature extraction is often performed to constrain the feature extraction within homogeneous regions [2] to enable localized queries that search for given features in specific parts of an image. Bandwidth requirements for transmitting these features are not trivial and can become a bottleneck when a large number of clients are querying the database server at the same time. Using compressed features in this context will lead to significant bandwidth and disk I/O savings since the feature data can be represented with far fewer bits. Now with the classification performed on compressed feature data instead of uncompressed data, classification error will be introduced due to compression. Features may need to be transmitted again for the mislabeled images due to the classification error that is introduced. Thus, efficient compression algorithms are needed to reduce the bit rate of the feature data and at the same time their impact on classification performance is minimized. All compression schemes involve trade-offs between coding complexity and coding performance. Obviously, if more memory and more computation power are available at the clients, better coding performance will be achievable.

In real applications a Decision Tree Classifier(DTC) is often used to enable fast query response for large image databases. Images which are similar in the feature space get assigned to the same tree node as traversing down this classification tree. Thus data tends to be clustered within each node. This property can be exploited to enable efficient quantization and entropy coding. In this paper we propose to perform a coarse version of the classification as a pre-processing before the data is com-

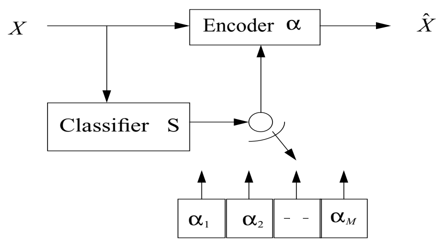

Fig. 1. The block diagram of a classified quantization system. Separate encoders $\{\alpha_i\}$ are designed for the classes $i = 1, ..., M$. The input vector X is first classified and then encoded with encoder specifically designed for the class.

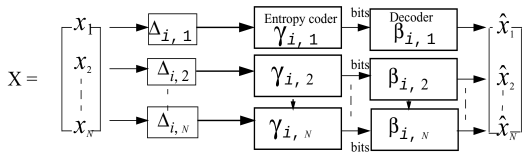

Fig. 2. Structure of the coding system. A bank of stepsizes $\{\Delta_{i,1}, \Delta_{i,2}, ..., \Delta_{i,N}\}$ are first applied to the scalar components $\{x_1, x_2, ..., x_N\}$ of X. Then independent entropy coding $\{\gamma_{i,j}, j = 1, ..., N\}$ were used to code the quantization indexes. The decoder $\beta_{i,j}$ consists of entropy decoding followed by inverse quantization. $\hat{X} \in R^N$ is the reconstructed vector for input $X \in R^N$.

pressed, and design separate encoders for each of the resulting classes. In previous proposed Classified Vector Quantization schemes [3][4], the classifier and the codebook for each class are designed in a sequential manner and the complexity of the resulting system plays no role in the design process. Instead, in this paper, we present a framework where the pre-classifier and the quantization parameter for each of the classes are searched in a joint manner under rate and complexity constraints using Generalized BFOS algorithm. Although we present an example where a simple uniform scalar quantization is employed and K-means tree is used as the classifier, the idea can easily be generalized to any classified quantization system where a decision tree is involved.

This paper is organized as follows: Section II provides a brief description of classified quantization techniques and introduces the novelty of our work. Section III states the problem that we try to solve and Section IV describes our proposed algorithm to solve the problem. The performance of proposed method is shown in section V using a simple texture classification example. Note that although we show the result specifically for texture classification, the idea can be used in other scenarios as well where we have multiple features to perform the classification. Finally, section VI concludes this work and discusses prospective future work.

II. Classified quantization systems

In Classified Quantization systems, inputs are first classified and then quantized with quantizers specifically designed for each of the classes. Figure 1 shows a generic classified quantization system. There are two components involved in a classified quantization system design: Source modeling, where the classifier is designed to best model the source distributions; Optimal quantizer design for each of the classes. In this work, we address the issue of optimal design of a classified quantizer in a rate-distortion-complexity framework. The rate and complexity constraints play the role of deciding how much preclassification we can afford at the transmitter and what parameters to use for each of the quantizers.

Classified Vector Quantization (CVQ) was proposed by Ramamurthi and Gersho [3] for image coding. An edge-oriented classifier was first designed to classify image blocks into either edge blocks or non-edge blocks. Then codebooks were designed specifically for the resulting classes using LBG algorithm. CVQ achieved better perceptual quality with significantly lower complexity compared to ordinary VQ. Riskin [4] proposed an algorithm to optimally allocate bits among classes using the Generalized BFOS algorithm for a M-class CVQ. The bit allocation was modeled as a $(M, 1)$ tree where the root had M children and each child is an unary subtree representing a linked list of codebook sizes for the corresponding class in increasing size order. The process of optimal bit allocation among the M classes involves assigning a maximum number of bits to each source and then reducing the number bits in order of increasing magnitude slope of increase in distortion to decrease in rate. Note that the main difference between [4] and our work is that we jointly search for the optimal pre-classifier and bit allocation among classes under both rate budget and complexity constraints. In [4] and [3], a M-class classifier was assumed to be fixed during the design of quantizers, and no complexity constraint was imposed in the system design. In this paper, we focus on the optimal design of the pre-classifier, as a pruned version of the original classifier, and the quantizers for each of the classes. We assume an original decision tree classifier was designed beforehand.

III. Problem formulation

Some reasonable structural restrictions are imposed on the quantizers to simplify our formulation. Uniform scalar quantizer is employed as the baseline quantizer due to its simplicity, and entropy coding such as Huffman coding is applied to losslessly encode the quantization indexes. The stepsizes are chosen from a pre-defined discrete set. Figure 2 shows the structure of such a coding system. Each classified encoder α_i consists of a bank of uniform quantizers $\{\Delta_{i,1}, \Delta_{i,2}, ..., \Delta_{i,N}\}$ followed by entropy coding $\{\gamma_{i,j}, j = 1, ..., N\}$ of the quantization indexes. The decoder $\beta_{i,j}$ consists of entropy decoding followed by inverse quantization. $\hat{X} \in R^N$ is the reconstructed vector for input $X \in R^N$.

Our goal is to find the optimal subtree $S^* \preceq \mathcal{T}$ as the pre-classifier and the set of stepsizes $\{\Delta^*_{i,j}, j = 1, ..., N\}$ for quantization of each class i, such that the overall distortion is minimized subject to a rate budget R_b and complexity constraint

C_b.

$$D^*=\min_{S^*,\{\Delta^*_{i,j}\}}\sum_{i=1}^{|\tilde{S}|} P_i\times D_i(\Delta_{i,1},\Delta_{i,2},\ldots,\Delta_{i,N}) \quad (1)$$
$$\text{s.t. } R(S,\{\Delta_{i,j}\})\le R_b \text{ and } C(S)\le C_b$$

Let us introduce the notations that will be used in this paper. $D()$, $C()$ and $R()$ are tree functionals defined as:

$$\begin{aligned} D(S,\{\Delta_{i,j}\}) &= \sum_{t\in\tilde{S}} P(t)\times d(t) \\ R(S,\{\Delta_{i,j}\}) &= \sum_{t\in\tilde{S}} P(t)\times r(t) \\ C(S,\{\Delta_{i,j}\}) &= \sum_{t\in\tilde{S}} P(t)\times l(t)+w\times|\tilde{S}| \end{aligned} \quad (2)$$

where $\tilde{S}$ is the set of leaf nodes of S and its size equals the number of encoders that need to be stored; $P(t)$ is the probability that a given input vector traverses node t; $l(t)$ is the length of the path from root to node t, and reflects the cost of traversing the classification tree S from root to node t; w is an positive weighting factor; Note that $C()$ is a weighted sum of computational complexity and memory requirement of the classified encoder. $r(t)$ and $d(t)$ are the operating entropy rate and distortion, respectively, of the quantized output for the sample space at node t, with the set of quantization stepsizes $\{\Delta_{i,j}\}$ applied to quantize the data at node t:

$$\begin{aligned} r(t_i,\{\Delta_{i,j}\}) &= \sum_{X\in t_i}\sum_{k=1}^{N} H(\hat{x}_k) \\ d(t_i,\{\Delta_{i,j}\}) &= \sum_{X\in t_i}\sum_{k=1}^{N} d(x_k,\hat{x}_k) \end{aligned} \quad (3)$$

Instead of solving the constrained problem (2), we use Lagrange Multipliers and solve the dual problem:

$$\min_{S^*}[\min_{\{\Delta^*_{i,j}\}}\{D(S,\{\Delta_{i,j}\})+\lambda\times R(S,\{\Delta_{i,j}\})+\mu\times C(S)\}] \quad (4)$$

The trade-offs between the data rate $R()$, distortion $D()$ and complexity $C()$ are played through adjusting the two multipliers λ and μ. Now we need to find the optimal multipliers λ and μ such that the rate and complexity constraints are satisfied with equality. It is not strait forward since we have two Lagrangian multipliers instead of one. We propose a nested optimization algorithm next to solve this problem.

IV. The Algorithm

The BFOS algorithm proposed by Friedman et al. [5] is a Lagrangian based method. It works by minimizing the functional $J(S) = \delta(S) + \lambda \times l(S)$ over all pruned subtrees, with $\delta(S)$ and $l(S)$ being the average distortion and rate of the tree structured vector quantizer defined on $\mathcal{T}$ and pruned to S. Chou et al. [6] extended the BFOS algorithm by generalizing the two components of the cost functional to any tree functional $U_1(S)$ and $U_2(S)$. It prunes off branches T_t of tree $\mathcal{T}$ in order of increasing slope $\mu = \frac{-\Delta(U_1,t)}{\Delta(U_2,t)}$ to find its optimal pruned subtrees. $\Delta(U_i,t)$ is the change of the tree functional U_i(increase for U_1 and decrease for U_2) if the branch rooted at node t is pruned off. It was proved that the generalized BFOS(G-BFOS) algorithm is capable of tracking out the extreme points which lie on the convex hull of the operating (U_1, U_2) pairs over all possible pruned subtrees $S \preceq \mathcal{T}$.

We define two tree functionals as follows:

$$\begin{aligned} U_1(S,\{\Delta_{i,j}\}) &= D(S,\{\Delta_{i,j}\})+\lambda\times R(S,\{\Delta_{i,j}\}) \\ U_2(S,\{\Delta_{i,j}\}) &= C(S,\{\Delta_{i,j}\}) \end{aligned} \quad (5)$$

As the sample space is hierarchically partitioned by the Nearest-Neighbor classification tree $\mathcal{T}$ defined in Euclidean space, within each node t data tends to be clustered. Due to this clustering property of the classifier, for a fixed multiplier λ, the tree functional U_1 tends to be monotonically decreasing as the tree grows. On the other hand, the complexity of the system is monotonically increasing since both the depth of the tree and the number of encoders become larger. This ensures that the optimal point will be found by pruning the tree. We propose a nested optimization algorithm to jointly search for the optimal subtree S^* and the set of quantization stepsizes $\{\Delta_{i,j}\}$ for a given rate budget and complexity constraint.

The basic idea is: First initialize the multiplier λ; Then for this fixed multiplier, we choose the set of stepsizes which minimize the functional $u_1(t)$ $\forall t \in \mathcal{T}$. We call this process "populate"; Then the G-BFOS algorithm is used to prune the original tree $\mathcal{T}$ until the complexity constraint C_b is satisfied. Now we have found the optimal multiplier μ to meet the complexity constraint given our choice of λ; Compute the resulting operating rate R with these two multipliers, adjust the multiplier λ to λ_{new} using bisection method [7] and populate nodes with λ_{new}. Then we prune the original tree $\mathcal{T}$ with the functional $u_1^{new}(t)$ updated with multiplier λ_{new} until target complexity budget is satisfied, then repeat the process until convergence. Note that every time we prune, we always start from the original tree $\mathcal{T}$. The detailed algorithm is as follows:

Proposed Algorithm : *Optimal design of classified quantizer under entropy and complexity constraints*

[Step 0]: *Initialize $\lambda_l \le \lambda_u$.*

[Step 1]:*Populate nodes with λ_l, prune the tree using G-BFOS until $C(S) \le C_b$. If $R_{\lambda_l}(S) \ge R_b$ does not hold, let $\lambda_l^{new} = a \times \lambda_l, a < 1$ and repeat step 1; Otherwise goto step 2.*

[Step 2]:*Populate nodes with λ_u, prune the tree using G-BFOS until $C(S) \le C_b$. If $R_{\lambda_u}(S) \le R_b$ does not hold, let $\lambda_u^{new} = \lambda_u \times \frac{1}{a}, a < 1$ and repeat step 2; Otherwise goto step 3.*

[Step 3]:*$\lambda_{next} = |\frac{D_{\lambda_l}-D_{\lambda_u}}{R_{\lambda_l}-R_{\lambda_u}}|$, populate nodes with λ_{next} and prune until the complexity constraint is satisfied. If $R_{next} = R_{\lambda_u}$, stop; Else if $R_{next} > R_b$, let $\lambda_l = \lambda_{next}$, repeat step 3; Else let $\lambda_u = \lambda_{next}$, repeat step 3.*

V. Experimental results

We show the performance of the proposed method using a simple texture classification example. The idea can be used in general scenarios where the classification is based on multiple image features. We perform the optimization on a binary decision tree $\mathcal{T}$. $\mathcal{T}$ is built in a top-down manner using K-means algorithm until only one class is left at each leaf node. The training set $\mathcal{L} = \{X, Y\}$ is a labeled vector sequence obtained by

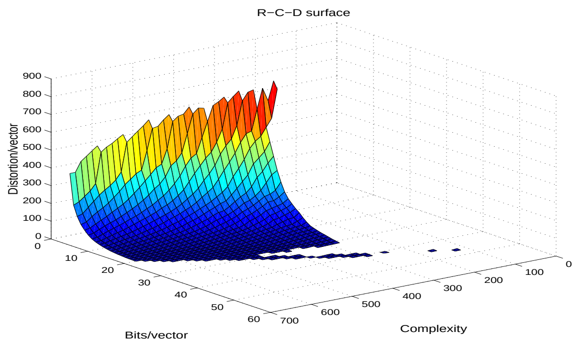

Fig. 3. The Rate-Distortion-Complexity surface obtained by proposed optimization framework. We employed traditional mean square error in this example. The complexity is evaluated as the cost of traversing the tree plus a weighted storage cost for storing the encoders, with the weighting factor $w = 1$ for this example. Rate is computed as the entropy rate of the quantization outputs.

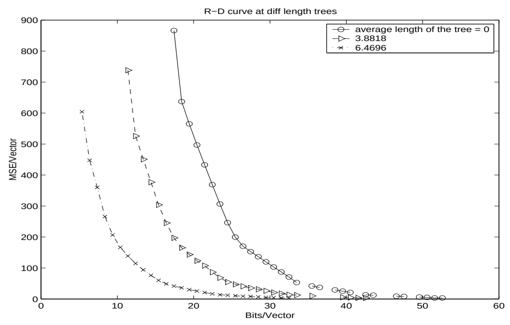

Fig. 4. Comparison of R-D performance between systems with pre-classification (using tree lengths 3.8818 and 6.4696) and without pre-classification (tree length 0).

computing the wavelet feature of the Brodatz texture album. X is a ten-dimensional feature vector and Y is the associated texture label. Details of obtaining the features are available in [8]. The stepsizes $\{\Delta_{i,j}\}$ are chosen from a predefined discrete set $\{32, 16, 8, 4, 2\}$. Figure 3 shows the obtained Rate-Distortion-Complexity surface. We can see the tradeoff between rate, distortion and complexity of the system and we can verify the convexity of the surface, as predicted by Goyal and Vetterli [9]. In Figure 4, we show the rate-distortion performance for this coding system with and without pre-classification. Substantial coding gain is obtained by employing a classified quantizer instead of a single quantizer, or by using a higher-complexity classification tree. In real applications, depending on how much complexity is allowed for the encoder, we are able to choose the best subtree as the pre-classifier which gives as much coding gain as possible.

We also perform texture classification with the compressed data using the decision tree classifier $\mathcal{T}$ and the result is shown in Figure 5. As is clearly shown in the figure, a lower classification error rate was achieved by using a classified encoder than a single encoder. At transmission rate of 20 bits/vector, a reduction of 11% in the classification error rate is achieved by using the classified encoder, with the pre-classification tree

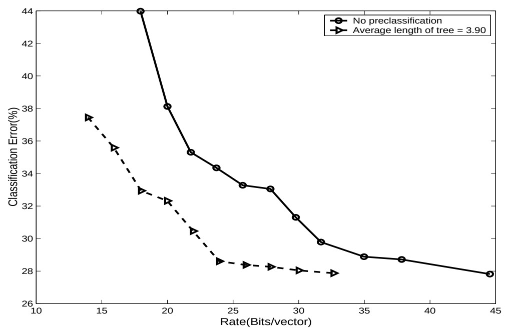

Fig. 5. Classification performance for systems with pre-classification(use tree length 3.90) and without pre-classification.

length equals to 3.9. Although we assume the traditional mean squared error distortion in this experiment, it can be extended to other distortion measures. Further study is needed on design of the distortion metric other than Mean Square Error to use for classification tasks.

VI. Conclusion

In this paper, we have proposed an optimal framework for the design of an entropy- and complexity-constrained classified quantization system. The Generalized BFOS(G-BFOS) was applied to jointly search for the optimal pre-classifier and the quantization parameters for each of the classes. The coding gain of employing such a system was illustrated by a texture classification example. The framework could be easily be generalized to classified quantization systems where decision tree is involved. Our future work would be to optimally design the quantizers for the set of classes in the sense of best classification performance, with the fidelity criterion extended from traditional mean square error to more meaningful distortion measure for classification. The work of finding the optimal transform coding is also under progress.

References

[1] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos, "The QBIC project: Querying images by content using color,texture, and shape," *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, vol. 1908, pp. 173–187, 1993.
[2] W.Y.Ma, *Netra : A Toolbox for Navigating Large Image Databases*, Ph.D. thesis, University of California, Santa Barbara, 1997.
[3] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. on Communications*, vol. 34, pp. 1105–1115, Nov. 1986.
[4] E. A. Riskin, "Optimal bit allocation via the generalized BFOS algorithm," *IEEE Trans. on Information Theory*, vol. 37, pp. 400–402, March. 1991.
[5] L. Breiman, J. H. Freidman, R. A. Olsehn, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.
[6] P. A. Chou, T. Lookabough, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. on Info. Theory*, vol. IT-35, pp. 299–315, March. 1989.
[7] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. on Image processing*, vol. 2, pp. 160–175, April. 1993.
[8] H. Xie and A. Ortega, "Feature representation and compression for content-based image retrieval," *VCIP*, vol. 4310, pp. 111–122, Jan. 2001.
[9] V. Goyal and M. Vetterli, "Computation distortion characteristics of block transform coding," in *Proc. of ICASSP*, Munich, Germany, April 1997.