

Color Image Vector Quantization Using Binary Tree Structured Self-Organizing Feature Maps

Jyh-Shan Chang, Jenn-Huei Jerry Lin, and Tzi-Dar Chiueh

Department of Electrical Engineering, National Taiwan University,
Taipei, Taiwan 10617, R.O.C.

Abstract— With the continuing growth of the World Wide Web (WWW) services over the Internet, the demands for rapid image transmission over a network link of limited bandwidth and economical image storage of a large image database is increasing rapidly. In this paper, a binary tree structured Self-Organizing Feature Map neural network is proposed to design the image vector codebook for quantizing color images. Simulations show that the algorithm not only produces codebooks with lower distortion than the well-known GLA-T algorithm but also performs better in differential index entropy which means more compression can be achieved with this algorithm. It should also be noticed that the obtained codebook is particularly well suited for progressive image transmission because it forms a binary tree in the input space.

Keywords— Vector quantization, SOFM, Binary tree, Progressive image transmission.

I. INTRODUCTION

Since the advent of the Internet, there have been great changes in information exchange. Among the services provided over the Internet, WWW service and video conferencing are the most important ones. Through these services, we can exchange information and communicate directly with the others with the help of colorful visual information. The key point for the success is the transmission of the color image. Full-color digital images typically use 24 bits to specify the color of each pixel of the images with 8 bits for each of the primary components, red, green, and blue. Because of the large number of bits required to represent a color image, it is obvious that transmitting the image over the Internet in uncompressed form is completely out of question. To overcome this difficulty, an enormous wealth of compression techniques are studied over the past few decades which make the transmission of images feasible. In this paper, vector quantization (VQ) which takes advantage of the spatial redundancy of image to compress image is used. With this tech-

nique, the transmitted image is first divided up into fixed-size rectangles. The proposed algorithm called binary-tree-structured SOFM (BTSOFM) is used to design a table of rectangles of the same size as the image rectangles. This table is called the codebook. Each rectangle is transmitted by looking it up in the codebook and just sending the index instead of the rectangle itself. If the codebook is properly designed, an image can be transmitted with very few bits which means a saving in bandwidth without loss of visual quality. We begin by giving a brief introduction to VQ in Section II. The algorithm for BTSOFM is presented in Section III. The results and comparisons with tree structured LBG algorithm are presented in Section IV.

II. VECTOR QUANTIZATION

Mathematically, a vector quantization system can be defined by two mappings: an encoder and a decoder. An encoder γ is defined by

$$\gamma : R^k \rightarrow I \subset \{0, 1\}^* . \quad (1)$$

A decoder β is defined by

$$\beta : I \rightarrow C = \{\beta(i); i \in I\} . \quad (2)$$

The VQ encoder reads an input pattern X and generates the index of the codeword $\gamma(X)$, while the VQ decoder uses this index to produce the corresponding codeword. Usually, the criterion used for choosing a VQ system is the squared Euclidean distance between the input pattern X and its encoded codeword $\beta(\gamma(X))$. The goal of VQ design is to find the mappings that minimize the overall distortion. However, direct use of full search VQ suffers from a serious complexity barrier. An alternative is tree-structured VQ (TSVQ)[1] which has the advantage the encoding complexity grows linearly with bit rate.

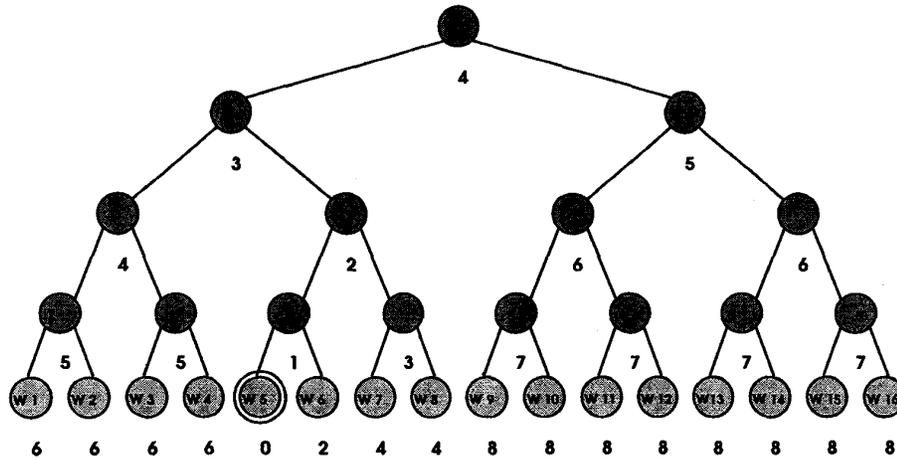


Fig. 1. Cell structure of the BT-SOFM with 16 terminal nodes. Numbers under the nodes are the distances (hops) to the winner node W5.

VQ has been an efficient method for lossy data compression, especially for speech and images [2] [3]. Linde, Buzo, and Gray [4] extended Lloyd's [5] basic design of scalar quantization to the general case of vector quantization. This algorithm has become one of the most famous algorithms for designing a codebook and is best known as the LBG algorithm or generalized Lloyd algorithm (GLA). In the LBG algorithm the training vectors are first partitioned into a set of subspaces according to their distances to the centroids of the subspaces. And for each subspace the centroid vector and the average distortion is recalculated. The process is iterated until the average distortion goes below a predefined threshold.

III. BINARY-TREE-STRUCTURED SOFM

Self-Organizing Feature Maps (SOFM) proposed by Kohonen have been known to be capable of learning structured clusters without supervision [6]. A comparison of coded images by the GLA and SOFM clustering algorithm is presented in [7]. In the original SOFM proposed by Kohonen, the neurons are organized in 1-D linear array, 2-D rectangular, or hexagonal grid. As the dimensionality of the training patterns increases and the training-pattern distribution becomes more complex, SOFM with 1-D or 2-D structure have more difficulty in representing the statistical nature of the training patterns with reasonable number of neurons. The proposed algorithm is a modification of SOFM. Instead of 1-D or 2-D structure, the neurons in the SOFM network are organized in a complete binary tree structure.

The learning algorithm of BT-SOFM is similar to that of the original SOFM. In BT-SOFM, tree search

is used to locate the nearest neuron (the winner) to a training pattern. The distance between two nodes is modified as the number of hops between them along the binary tree instead of the Euclidean distance between them. A four-level, sixteen-terminal-node BT-SOFM is given in Fig. 1 as an illustration. Initial codewords in BT-SOFM form a full binary tree in the input space and as the training goes on, this high-dimensional tree is gradually stretched so that the terminal-node codewords reflect training-pattern statistics. Also interlevel codewords are updated in such a way that a binary tree is retained during the training process. Therefore, the final codebook also has a full binary tree structure, making it suitable for tree search VQ (TSVQ).

IV. EXPERIMENTS AND RESULTS

We use the BT-SOFM to train TSVQ codebooks for color images. The network is an eight-level binary tree, containing 254 interlevel nodes and 256 terminal nodes. The training starts with a neighborhood of radius 16, covering all nodes in the tree. All training patterns are presented to the BT-SOFM sequentially, and the network is updated as described in Sec. III. "Equilibrium" is assumed after the decrease in MSE goes below a predefined threshold, then the radius is decreased by one. When the radius reaches zero, i.e., when the neighborhood contains only the winning node itself, and percentage improvement in MSE is less than a threshold, the training stops. Fig. 2 shows a TSVQ codebook generated by the BT-SOFM model from the color "lena" image. In Fig. 2(a), the first two blocks in the first row are the two first-level codewords; the next four blocks in the second row are

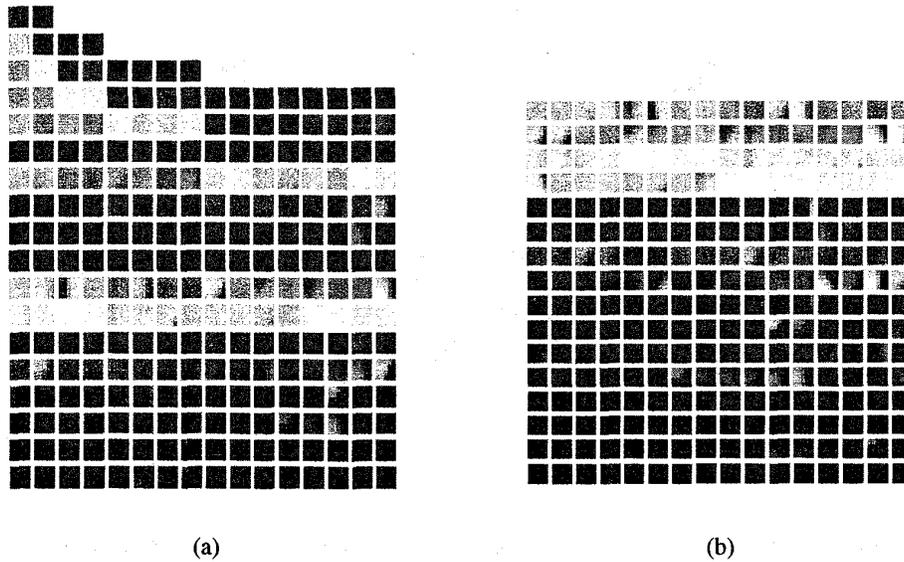


Fig. 2. Final TSVQ codebooks generated by the BTSOFM algorithm. (a) Interlevel codewords. (b) Terminal (leaf) codewords.

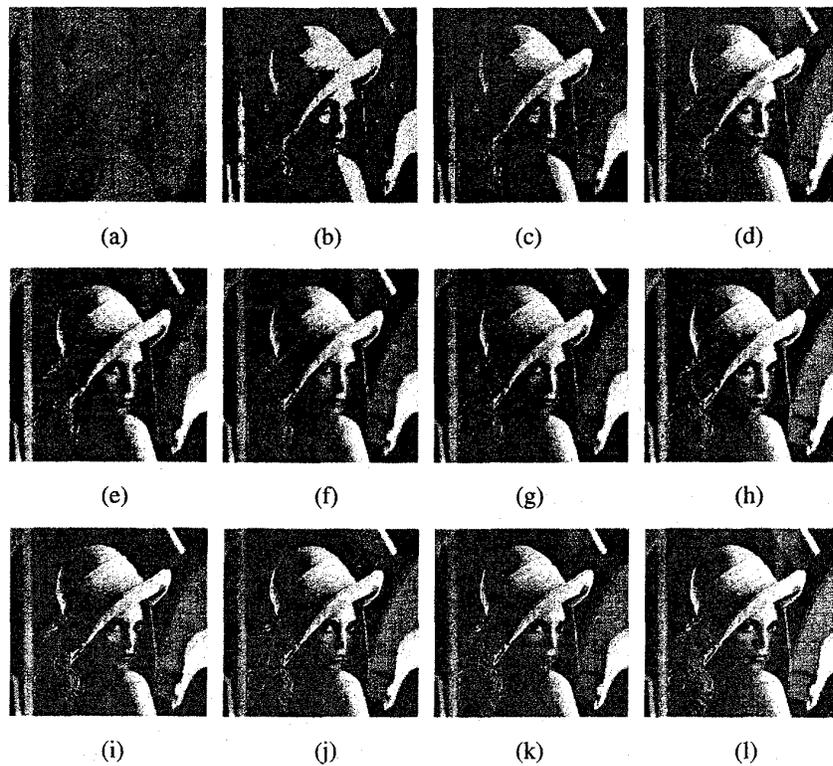


Fig. 3. Progressively-coded color images using interlevel codewords in a BTSOFM TSVQ codebook. (a) to (l) 1 to 12 bits/block, i.e., 0.0625 bpp to 0.75 bpp.

the second-level codewords, and so on. The terminal (leaf) codewords are shown in Fig. 2(b).

In many image communication/retrieval environments, it is preferable to provide successively better approximations as the coded information arrives rather than wait until all information is specified. The simplest and most intuitive way to realize progressive transmission is the bit-plane method. In this method, the index of n bits for each image block to be transmitted is organized into n bit planes, where the i th most significant bits of all codeword indices make up the i th bit plane. In each pass of transmission, data corresponding to one bit plane are sent. Since the obtained codebook forms a binary tree in the input space, it is particularly well suited for progressive image transmission because the codewords are progressively more detailed representations of the image blocks to be coded. This property would enable us to transmit a coarse rendition of the image over a congested network so that an early impression of the image can be observed. And successively better approximations of the image is provided as more information about the image arrives. As an illustration, a twelve-level binary-tree-structured BTSOFM codebook is generated. At first, the two first-level codewords are used to vector quantize the blocks in the image and the data for the first bit plane are therefore obtained. At the receiver side, a coarse image can be reconstructed after the first bit plane is received. A series of images, shown in Fig. 3, demonstrates the progressive transmission of images achieved by the BTSOFM model.

To investigate the performance of this algorithm, a modified version of the GLA algorithm called GLA-TSVQ (GLA-T) algorithm [8] is used for comparison. As the training progresses down the tree in the GLA-T algorithm, the training sets are partitioned such that no training pattern affects a node unless that training pattern affects its parent node as well. This would result a tree structure codewords too. Both of these two TSVQ algorithms are applied to a digital color image "lena" of 512×512 pixels with 8-bit resolution. The comparison is done when each of the 12 bit-planes is received by the decoder at each intermediate step. The PSNR's of the two algorithms are shown in Fig. 4. It is clearly shown in this figure that the BTSOFM algorithm is superior to the GLA-T algorithm.

The codewords generated by the BTSOFM model are topologically more correlated, in other words, adjacent codewords are more similar than those farther away. Since spatially neighboring blocks in images usually are similar, the corresponding codewords should be similar, if not identical. Therefore, the codeword indices of adjacent blocks should be more correlated in the BTSOFM case. One of the popular

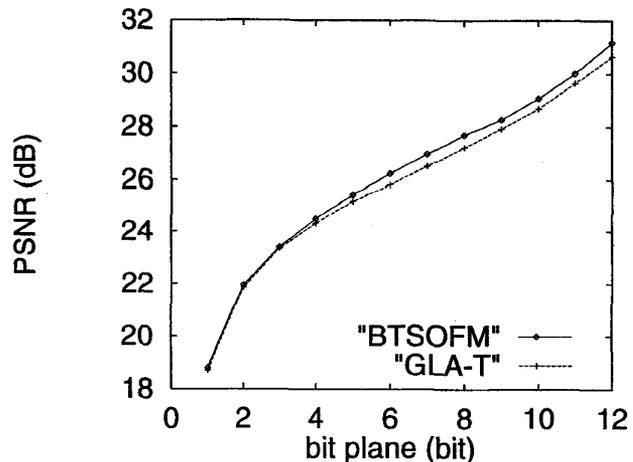


Fig. 4. Comparison of the PSNR's after each bit plane is received.

source coding techniques is predictive vector quantization. Here the codeword index of the previous block is used as the predicted index of the current block. Since adjacent blocks in images are often similar and the indices of two similar (usually neighboring) BTSOFM-generated TSVQ codewords are also similar. Therefore, the difference between two consecutive indices should often be small. To investigate this phenomenon, the GLA-T algorithm is used again for comparison. Because of its training algorithm, the codewords within this structure are less correlated and are expected to perform poorer in codeword index prediction. For comparison, the differences between two consecutive codeword indices of an image after the data of each bit plane have been transmitted are computed. The information entropy of these differential indices is calculated. The resulting cumulative differential index entropy versus PSNR of image coded by the BTSOFM-generated TSVQ codebook and the GLA-T TSVQ codebook is shown in Fig. 5. From Fig. 5 we notice that the BTSOFM algorithm always perform better than the GLA-T algorithm as each bit plane is successively transmitted. This superiority in the cumulative differential index entropy implies that further compression can be achieved with this algorithm when source coding is applied after VQ. That is, the indices to be transmitted for each bit plane can be further compressed with source coding techniques such as LZ coding or entropy coding. Then, instead of sending the indices for each bit plane directly, the compressed indices are transmitted thus achieve further compression. The comparison of the cumulative index size transmitted with the BTSOFM algorithm, the GLA-T algorithm, and the original uncompressed

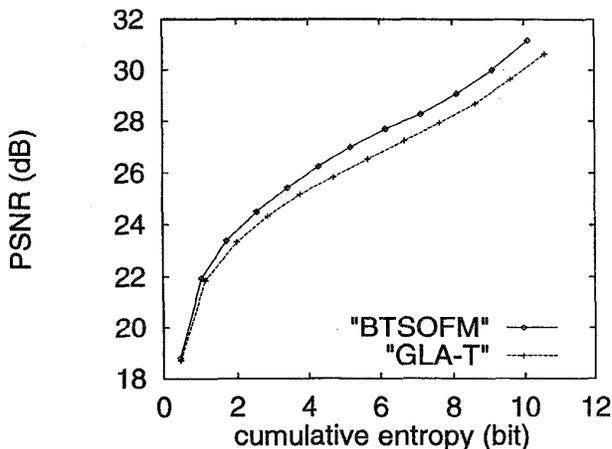


Fig. 5. Comparison of the cumulative differential index entropy.

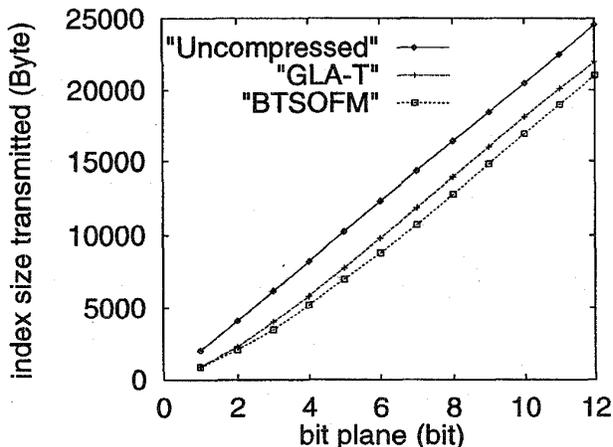


Fig. 6. Comparison of the cumulative index size transmitted.

indices is shown in Fig. 6. A 17% less index size can be obtained with the BTSOFM algorithm compared to the uncompressed indices.

V. CONCLUSIONS

In this paper, a modified SOFM algorithm that has a binary-tree cell structure, called binary-tree-structured SOFM (BTSOFM) is used to vector quantize color images. We have demonstrated that the obtained codebook is particularly well suited for progressive color image transmission because of its binary-tree structure. We have also demonstrated that the BTSOFM algorithm not only produces codebooks with lower distortion than GLA-T algorithm but also performs better in differential index entropy which means more compression than GLA-T algorithm can be achieved by the BTSOFM algorithm.

REFERENCES

- [1] T. D. Chiueh, T. T. Tang, and L. G. Chen, "Vector Quantization Using Tree-Structured Self-Organizing Feature Maps," *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 9, pp. 1594-1599, Dec. 1994.
- [2] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, pp. 4-29, 1984.
- [3] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.
- [4] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun.*, Vol. COM-28, pp. 84-95, January 1980.
- [5] S. P. Lloyd, "Least-Squares Quantization in PCM," *IEEE Trans. Inform. Theory*, Vol. IT-28, pp. 129-137, March 1982.
- [6] T. Kohonen, *Self-Organization and Associative Memory*. New York: Springer-Verlag, 1988.
- [7] N. M. Nasrabadi and Y. King, "Vector quantization of images based upon the Kohonen self-organizing feature maps," in *Proc. IEEE Int. Conf. Neural Networks*, 1988, pp. I-101 to I-108, 1988.
- [8] A. Buzo, A. H. Gary, Jr., R. M. Gray, and J. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, pp. 562-574, 1980.