International Journal of Innovative and Emerging Research in Engineering
Volume 3, Special Issue 1, ICSTSD 2016

Design of Vector Quantization Codebooks using A Genetic Algorithm

Mr. Pravin Wararkar
Dept. of EXTC,
SVKM's NMIMS MPSTME, Shirpur
Dhule, India
pwararkar@gmail.com

Ankit Dixit
Dept. of EXTC,
SVKM's NMIMS MPSTME, Shirpur
Dhule, India
ankitdxt@gmail.com

Girish Patil
Dept. of EXTC,
PRMIT&R, Badnera
Amravati, India
patilgirish213@gmail.com

Pratiksha Meshram
Dept. of IT,
SVKM's NMIMS MPSTME, Shirpur
Dhule, India
pratiksha.kaya@gmail.com

***Abstract*— Data compression techniques recode data into more compact forms. One such technique is vector quantization, which maps groups of input symbols, called vectors, onto a small set of vectors, called the codebook. Each vector in the codebook is a codeword. The indexes of the codewords represent the original vectors, and writing the codewords that the indexes indicate restores a facsimile of the original data.**

The similarity of the restored data to the original under vector quantization depends on the codebook, and several algorithms have been proposed for designing it from a training set of typical vectors. This paper describes a genetical algorithm for the problem of codebook design. The genetic algorithm's chromosomes represent partitions of the training set; 'each vector maps to the codeword that is the centroid of its set in the partition. To speed up its operation, the genetic algorithm uses fitness inheritance to assign fitness values to most new chromosomes, rather than evaluating them. Tests using five standard digitized images compare the genetic algorithm to a popular non-genetic algorithm for codebook design. The genetic algorithm is found to be effective, but slow.

Keywords—PIC microcontroller, RF Modem, wireless communication, transciever

I. INTRODUCTION

Data compression techniques recode data into more compact forms; such techniques are particularly useful when large amounts of data are to be stored or transmitted. Among the many techniques for compressing data is vector quantization, which breaks a sequence of symbols into blocks of uniform size, called vectors, and maps these vectors onto a small set of similar vectors. The set is called the codebook, and each vector in it is called a codeword. The codewords are ordered, and the indexes of the original vectors' codewords then represent the original sequence; writing the indexed codewords constructs a facsimile of it. This technique is lossy—the restored sequence is not identical to the original—but it can achieve high levels of compression.

Clearly the codebook is central to vector quantization's effectiveness, and several algorithms have been proposed to design a codebook from a training set of typical vectors. The search for a good codebook, however, is in general difficult.

The genetic algorithm has been proposed as a general purpose search technique suitable for searching large and ill-formed spaces. This paper describes a genetic algorithm that designs vector quantization codebooks for data compression and compares the algorithm's performance with that of the LBG algorithm, a popular algorithm for code-book design, on an image compression task. We find that, though the genetic algorithm is slow, it is competitive with the LBG algorithm and it holds promise of being an effective mechanism for designing vector quantization code-books, at least for training sets of moderate size.

The following sections of this paper describe the problems of data compression and, specifically, of image compression; vector quantization and the centrality of the code-book in it; the LBG algorithm for codebook design; a genetic algorithm for codebook design; and tests that compress and reconstruct five standard images to examine the performance of the genetic algorithm and compare it to the LBG algorithm.

II. COMPRESSION TECHNIQUES

This section presents the details of various compression techniques.

A. **Data Compression**

Wireless In computing, data are represented by sequences of symbols that are themselves sequences of bits. Data compression takes advantage of structural properties of these sequences in order to represent them in more compact forms; that is, data compression seeks to re-represent data using fewer bits. A simple example is **Huffman coding**, which encodes more frequently occurring symbols with short sequences of bits, and less common symbols with longer codes. Morse code, in which a single dot represents the letter 'e' is a Huffman code.

Having been compressed for storage or transmission, data must be restored to something like their original form. Thus

a data compression technique consists of two steps: the first maps the original sequence of symbols x into its compressed form x_c; the second reconstructs a sequence x' from x_c. When $x' = x$ that is, when the restored data are identical to the original, the technique is called lossless. Lossless techniques are generally used to compress text, which we want to be able to restore verbatim.

In many applications, however, it is sufficient that x' approximate x; that is, that the restored data be a reasonable facsimile of the original. Such techniques are called lossy. Audio signals, for example, can suffer a loss of fidelity and still be intelligible. In general, and not surprisingly, lossy methods offer greater compression than do lossless methods. A measure of the degree of compression is the **compression ratio**:

$$\text{Compression ratio} = \frac{\text{length of orignal data string}}{\text{length of compresseddata string}}$$

B. Image Compression

Projecting a three-dimensional scene onto a two-dimensional surface forms an image. An image can be represented by a two-dimensional array of values, each of which represents the average brightness or color intensity over a small region of the image. The values associated with each small region of the image comprise **pixels**, and the array is a **digitized image**. Digitized images are often very large, so the utility of data compression applied to such data, called image compression, is clear.

Both lossless and lossy compression techniques can be applied to digitized images. For example, when an image contains large regions of uniform values, **run-length coding** is suitable. This technique represents an image with a sequence of pairs of integers. Each pair (v,l) represents a run of the value v of length I. Fax transmissions, in which the value zero occurs nearly all the time, use run-length coding.

Another lossless compression technique, called **predictive coding**, records not the sequence of pixel values themselves but instead the difference of each value from its pre-decessor. This technique assumes that such differences are in general small and can be represented by short codes. Images usually satisfy this assumption.

In many applications, however, the reconstructed image need not be a verbatim reproduction of the original, and we can employ lossy compression techniques. The simplest such technique is **scalar quantization**, which maps the original values to a smaller set of coding values, often via, a linear mapping. For example, gray-scale values from 0 to 255 might be mapped (by truncating their least significant four bits; that is, by dividing by 16) to the values 0 to 15. In exchange for a loss of gray-scale detail, this coding reduces the size of an image by half.

As noted above, predictive coding is efficient when neighboring pixels are similar. If the prediction error is quantized, predictive coding becomes a lossy compression method.

One of the most popular methods of image compression is **vector quantization** (VQ), which extends the idea of scalar quantization to groups of pixels.

Having been compressed for storage or transmission, data must be restored to something like their original form. Thus a data compression technique consists of two steps: the first maps the original sequence of symbols x into its compressed form x_c; the second reconstructs a sequence x' from x_c. When $x' = x$ that is, when the restored data are identical to the original, the technique is called lossless. Lossless techniques are generally used to compress text, which we want to be able to restore verbatim.

C. Vector Quantization

Vector quantization recodes fixed-length sequences of in-put symbols—here, pixels—by mapping each sequence onto a sequence in a small finite set. Each sequence is called a vector; the vectors onto which the original vectors are mapped are **codewords**; the finite set of codewords is the **codebook**. Within the codebook, each codeword has an index. VQ can be used to compress data of any kind; when it compresses images, each vector typically represents a small region of an image, say a 4 ×4 rectangle of pixels.

Given a codebook, compressing an image with VQ consists of identifying the codeword for each vector in the image. The indexes of the codewords then represent the image, and restoring the image requires only writing the codewords corresponding to the indexes; these codewords comprise the (lossy) restored image.

The length of the codewords and the size of the code-book determine VQ's compression ratio. If each pixel requires p bits and each vector consists of k pixels, then each codeword requires pk bits. If the codebook contains AT codewords, then an index requires log2 N bits. An image whose dimensions are m x n requires mnp bits in its uncompressed form, and (mn/k)log2 N bits compressed, so the compression ratio is

$$Copmression\ Ratio = \frac{length\ of\ orignal\ data\ string}{length\ of\ compressed\ data\ string}$$

The fidelity of the restored image depends oh the code-book. If the vectors in the image are in general close to the codewords, the restored image will be similar to the original. Codebooks are typically generated by examining a training set of vectors sampled from images the VQ sys-tem will compress. Two conditions describe optimal VQ encoding and decoding, respectively.

The **nearest neighbor condition** defines an optimal VQ encoding, given a codebook. It requires simply that every vector map to the nearest codeword in the codebook. Euclidean distance often measures the distance between vectors. The **centroid condition** defines an optimal VQ decoding.. The centroid of a set of vectors is their average; the centroid condition requires that each codeword be equal to the centroid of the vectors that map to it. These two conditions drive codebook design, which we now consider.

The most commonly used algorithm for VQ codebook design is due to Linde, Buzo, and Gray and is called the LBG algorithm,

It is an iterative algorithm based on the two conditions just described that improves an initially random codebook using a training set. The following sketch summarizes the LBG algorithm, given a set of training vectors and a codebook size N.

1. Randomly partition the. training vectors into IV disjoint sets; the N centroids of the sets of training vectors are the initial codewords.
2. Map each training vector to the nearest codeword.
3. If there is a codeword to which no vectors map, divide the vectors that map to the most popular codeword into two groups, map one group to the codeword whose group was empty.
4. Replace each codeword with the centroid of the vectors that map to it.
5. If the termination condition is met, terminate; otherwise, go to step 2.

The termination condition may be that the average distance between the training vectors and their codewords is less than some threshold or that the most recent iteration did not reduce the average distance. The LBG algorithm converges to a local minimum of the average distance, de-pending on the initial codebook.

Fig 1 From the left: the original 256×256 Lena image; the image as reconstructed from the LBG codebook; and the image as reconstructed from the GA-generated codebook.

III. CONCLUSIONS

In this work, a cost-effective microcontroller and RF based wireless announcement system is proposed with an effective approach. This system will provide facilities like high quality of reproduction of original sound and greater selectivity of particular receiver.

Also, security of voice signal is possible since the code sent for selection of particular receiver is known to a user only. So, only authorized user can access the system. In case, if an unauthorized person wants to access the system through his RF modem and pair of wireless microphone and receiver, the

system will not respond because it will be accessible only through a specific code feed to the microcontroller.

In this project, adding extra number of receivers to the installed system is very simple process. Just tuning the receiver section to the same frequency as of transmitter and assignment of unique code to the receiver is possible with little bit modification in a programming will make it possible to add more number of receivers.

In addition, because the communicating entities can freely move, one can place the announcement system wherever it is required without the cost incurred with cabling unlike the wired communication approach. It is simply like a plug and play device.

References

[1] Ankit Rawat, Sushila Chahar, Himanshu Bhojwani, "Moderate Quality Of Voice Transmission Using 8-bit Micro-controller Through Zigbee", *International Journal of Research in Engineering and Technology(IJRET)*, 2014, pp- 855-862.

[2] Ms. Reshma Sultana, Ms. Seema Sultana, Mr. Mohammad Osman, "Design and Implementation of a Teacher-Student Interaction System Based On ZIGBEE and RFID", *International Journal of Engineering Research and Applications, (IJERA)*,2014, pp.307-311.

[3] Marcel Meli, Martin Gysel, Marc Sommerhalder, "Using IEEE 802.15.4 / ZigBee in audio applications", Embedded world conference, 2006, pp.57-68.

[4] L. Luo, W. Cao, C. Huang, T. Abdelzaher, J. A. Stankovic, and M. Ward, "Enviromic: Towards cooperative storage and retrieval in audio sensor networks," *27th International Conference on Distributed Computing Systems*, 2007, pp. 34-34.

[5] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over Sensor Networks,"*27th IEEE International of Real-Time Systems Symposium*, 2006, pp. 291-302.

[6] H. Y. Song, and S. H. Cho, "Performances of IEEE 802.15.4 Unslotted CSMA-CA for Voice Communications," *The 17th Asia-Pacific Conference on Communications*, 2011.

[7] H.Y. Song, H.C.Yoon, and S.H.Cho, "Implementation and analysis of IEEE 802.15.4 MAC for Voice Communications," *The 4th Joint Workshop between HYU and BUPT*, 2009.

[8] M. Ruta, F. Scioscia, T. D. Noia, and E. D. Sciascio, "A hybrid ZigBee/Bluetooth approach to mobile semantic grids," *International Journal of Computer Systems Science & Engineering*, 2010, pp. 49-63.

[9] http://www.enginnersgarage.com

[10] http:// www.microchip.com