2. VECTOR QUANTIZATION AND VECTOR QUANTIZER: A REVIEW

The idea of 'Quantization' has a long history in the literature starting from 1898. Today, we use this method in many different areas, in many different forms. The most popular application of quantization is Analog to Digital conversion. The historical story of the quantization is summarized in a review of Robert M. GRAY and David L. NEUHOFF titled 'QUANTIZATION' [1].

In the information theory literature quantization is generally considered a form of data compression. The goal of quantization is to encode the data from a source that is characterized by its probability density function with as minimum rate (number of bits of reproduction code) as possible in such a way that reproducing may be recovered from the bits with as high quality as possible. We can easily remark the trade-off between average distortion (quality) and rate that are primary performance criteria of quantization [1].

Quantizers or more generally data compression are widely used in the communications systems and signal processing. Some of its application areas include data compression (image, audio and video), image restoration and classification, pattern and speech recognition, texture classification and detection, coding, teleconferencing, remote sensing, radar, sonar, computer communication, facsimile transmission, image-data base management etc…

We can classify quantizers into two different sub-categories with respect to their input vector size. If the input vector is one dimensional, the quantizer is classified as 'Scalar quantizer'; otherwise it is classified as 'Vector Quantizer'.

In recent years, there has been much interest focused on Vector Quantization. Driving forces behind this trend can be summarized as follow [2]:

1. According to Shannon's rate-distortion theory, a better performance is always achievable in theory by coding a block of signal (vector) instead of coding each signal individually (scalar).

2. Designing more sophisticated coding/decoding systems becomes possible with technology improvements.

3. As technology enhancement advances, the requirements placed on communication sub-systems become more and more demanding. Several key driving technologies strongly require data compression techniques (for example : HDTV)

4. The nature of the signal to be presented in computer moves from artificial signals (like texts…) to signal closer to those in physical world (sound, images…) with tend to be more unpredictable and hard to characterize analytically.

Let us now define the vector quantization. Vector quantization is a mapping Q from m dimensional vector space R^m into a finite subset T of R^m ($T \subset R^m$). It is denoted by

$$Q : R^m \rightarrow T \qquad 2.1$$

T is the set of reproduction vectors and it is called template-book (code-book) for the vector quantizer.

$$T = \{t_i | i = 1...N\} \qquad 2.2$$

N is the number of template-vectors in template-book. For every input vector **X**, a template-word $\mathbf{t_i}$ is selected as the representation for **X**. This process is called quantization phase (or the template-book search phase) of the Vector Quantizer. It is denoted by

$$Q(\mathbf{X}) = t_i \qquad 2.3$$

Then the code word $\mathbf{t_j}$ is represented by some symbols (normally the address of the template-vector in the template-book). This is called the encoding phase of the Vector Quantizer.

Rate (code rate or resolution) measures the number of bits per vector component used to represent the input vector and gives an indication of the accuracy or precision that is achievable with a vector quantizer if the template-book is well designed. N and k being the number of template-vector in template-book and the vector size, rate can be expressed as follow

$$r = \frac{\log_2 N}{k} \qquad 2.4$$

The average quantization error between input source and their reproduction template-vector is called the distortion of the Vector Quantizer.

Clearly, increasing the number of template-vectors in the template-book can decrease the distortion of a Vector Quantizer and normally will increase the rate also. The major concern for a Vector Quantizer template-book design is the trade-off between distortion and rate. The quality of the template-book is judged by reproduced vector fidelity, usually qualified by the Signal to Noise Ratio (SNR):

$$SNR = -10\log_{10}\left(\sum_{i=1}^{m}\left[\frac{(x_i - t_i)}{x_i}\right]^2\right) \qquad 2.5$$

In (2.5), $\mathbf{x_i}$ is the input signal; $\mathbf{t_i}$ is the reproduced output and $\mathbf{m}$ is the length of the template-vector.

We can illustrate mapping as a partition process of the input vector space to hyper-planes represented by the template-vectors in the template-book. We will see later 3 simple partition examples in Chapter 4.

The vector quantizer can be considered a Kohonen's Self-Organizing Feature Map (SOM). SOM are based on competitive learning. The output neurons of the network compete among themselves to be activated or fired, with the result that only one output is on at any one time. An output neuron that wins the competition is called 'Winner-Takes-All' neuron or simply '*winner'* neuron.

SOM is characterized by the formation of a topographic map of the input pattern in which the spatial locations of the neurons are indicative of intrinsic statistical features contained in the input pattern.

Kohonen's model belongs to the class of vector coding algorithm. The model provides a topological mapping that optimally places a higher dimensional input space into a fixed number of vectors. The Kohonen's model may be derived in two different ways. One may use basic ideas of Self-Organization, motivated by neurobiological considerations to derive the model. Alternatively, one may use a vector quantization approach that uses a model involving an encoder and a decoder, which is motivated by information theoretic considerations.

The principal goal of the SOM is to transform an incoming signal pattern of arbitrary dimension into a one or two-dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion.

The algorithm responsible for the formation of the SOM starts by initializing the synaptic weight randomly in the network in order to prevent imposing prior order on the feature map. After the initialization, there are 3 essential process to perform for the formation of SOM.

1. Competition :

   For each input pattern, the neurons in the network compute their respective values of a discriminant function (i.e. Euclidean distance between input pattern and synaptic weights). This discriminant function provides the basis for competition among the neurons. The particular neuron with the largest value of discriminant function is declared *winner* of the competition.

2. Cooperation :

   The winning neuron determines the spatial location of a topological neighborhood of excited neurons with respect to neighborhood function, thereby providing the basis cooperation among such neighboring neurons.

3. Synaptic Adaptation :

   This last mechanism enables the excited neurons to increase their individual values of the discriminant function in relation to the input pattern through suitable adjustments applied to their synaptic weights with respect to a learning rate parameter. The adjustments are made such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.

In the long run, when all the vectors X of the input space have been presented sufficient times, the responses of the neurons are spatially ordered. Thus a projection of the input subspace onto the networks occurs. Furthermore, this mapping conserves the topology associated with the input space.

The basic aim of SOM algorithm is to store a large set of input vectors $X \in R^m$ by finding smaller set of prototypes $w_j \in T \subset R^m$ so as to provide a good approximation to the original input space R^m. Thus the SOM algorithm is a vector quantization algorithm.

The problem of designing a VQ system is constituted by 2 part.

First part of the problem concern to design an ‘optimal’ template-book for a given probability density function of input vector space in order to minimize average distortion in replacing any input vector by the closest template-vector. This part of the problem is the subject of Information Theory. The most famous algorithm to find out the ‘optimal’ template-book (code-book) is LBG algorithm [3]. There are also different algorithms and methods in literature. [4-7]

The second part of the problem, which concerns the subject of this thesis, is to implement Vector Quantizer System for a given template-book.

According to the definition, implemented VQ system consists of two main functional parts:

The first part measures the distances between input vectors and each of the template-vectors, and then selects the smallest distance among them. Selecting process is a classification operation for a given distance metric. This part realizes 'The Template-book Searching phase' of the Vector Quantizer system.

The second part consists of a demultiplexing output stage encoding the index of the winner template-vector. This part realizes 'The encoding phase' of the Vector Quantizer system.

The schematic view of implemented VQ system can be seen in Figure 2.1. In the schematic, The first part of the system consists of 'Distance block' and 'MINNET block' (Loser-Takes-All). The second part of the system consists of 'The Encoder block'.

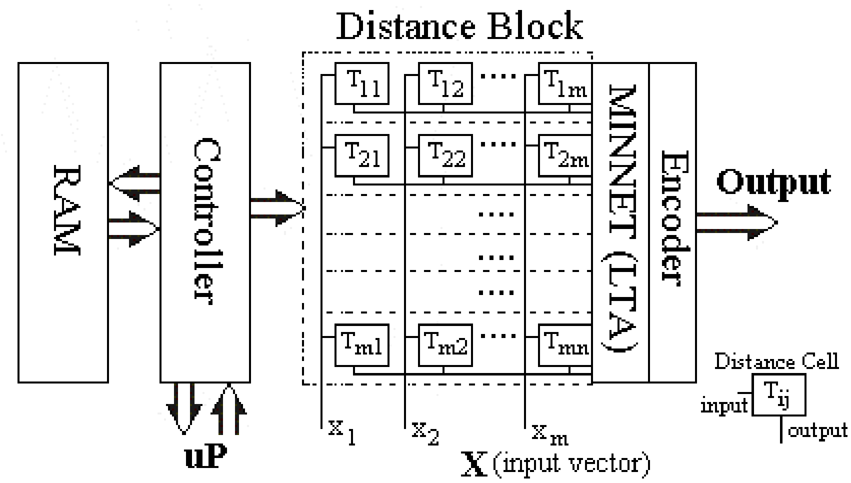

Figure 2.1: Schematic view of Vector Quantizer system

The functions of the building blocks of VQ system in Figure 2.1 can be summarized as follows:

1. Distance Block

   This block is responsible of the measurement of the distances between input and template-vectors. It consists of rows of Distance Cells (DCELL). Each Distance Cell computes the distance between the input vector element and the template-vector element that is stored in the cell, for a specific distance metric. Output of each DCELL is summed to constitute row output that is equal to the distance between input vector and the template-vector of that row. Thus, for parallel computing of distances (for parallel searching of template-vector), the number of DCELL rows in Distance block must be equal to the number of template-vectors in the template-book. The speed of the Distance block is determined by the computation speed of one DCELL block. It is obvious that for decreasing silicon area consumption of this block, unit DCELL block must be as small as possible. Or, although it consumes more operation time, we can prefer the sequential computation of distances, which is also possible.

2. MINNET (Loser-Takes-All) Block

This block is responsible of selecting the smallest distance among the distances computed in the Distance block rows' output. This is well-know Loser-Takes-All network. Clearly, increasing the input number (number of row or number of template-vector in template-book) slows the network and increases its silicon area consumption for a given resolution.

3. Encoder Block

This is the functional block that is previously referred as demultiplexing output stage. The Encoder block is responsible of coding of the selected template-vector's index with respect to given algorithm.

4. Controller and RAM blocks

Digital 'Controller Block' is responsible of writing and reading operations of the template-book. Communications with the microprocessor that want to read (or write) the template-vectors are organized also by this block. RAM block stores the template-book. Control signals of RAM block is produced by Controller block.

In the next chapter, I will analyze all the building blocks of VQ system, which are used to implement the system on silicon in detail.