Fast and Efficient quantization for image compression using reconfigurable platform

Dr.S.VIJAYARAGHAVAN,

Assistant Professor, SCSVMV Deemed University,Kancheepuram

Abstract

This paper presents an image compression scheme based on a block based compression algorithm referred to as Vector Quantization combined with the Differential Pulse Code Modulation technique. The proposed image compression technique called the Novel Vector Quantization scheme results in enhanced image quality. The generated codebooks for the NVQ scheme are more robust for image coding. The proposed system was validated through Field Programmable Gate Array. The resulting implementation achieved good compression and image quality with moderate hardware system complexity at low computational cost.

Index Terms- *Image Compression, Quantization, JPEG 2000, Bit Stream, Code Books, FPGA*

INTRODUCTION

Image compression:

Compression is a process that creates a compact data representation for storage and transmission purposes. Media compression usually involves the utilization of special compression tools because media is different from the generic data. Generic data file, such as a computer executable program, a Word document, must be compressed with lossless technique. Even a single bit error may render a data file useless. On the other hand, distortion is tolerable in the media compression process; as it is the content of the media that is of great importance, rather than the exact bit. Since the size of the original media, whether it is an image, a sound clip, or a movie clip, is mostly very large, it is essential to compress the media at a considerably high compression ratio. Such high ratio compression is usually achieved through two mechanisms. First is to ignore the media components that are less perceptible, and second is to use entropy coding to explore information redundancies which exist in the source data.

Conventional media compression solution focuses on one function i.e, to turn a media into a compact bit stream representation, whose compression ratio is determined at the time the compressed bit stream is formed. Yet, different applications may have different requirements of the compression ratio and tolerance of the compression distortion. A publish application may require a compression scheme with very little distortion, while a web application may tolerate relatively large distortion in exchange of a smaller compressed media. Recently, a category of media compression algorithms termed scalable compression emerges to offer the ability to trade between the compression ratio and distortion after the compressed bit stream has been generated. In scalable compression, a media is first compressed into a master bit stream, where a subset of the master bit stream may be extracted to form an application bit stream with a higher compression ratio. With scalable compression, a compressed media can be effortlessly tailored for applications with vastly different compression ratio and quality requirement, which is particularly valuable in media storage and transmission. Image compression is an application of data compression that encodes the original image with few bits. The objective of image compression is to reduce the redundancy of the image and to store or transmit data in an efficient form. The main goal of such system is to reduce the storage quantity as much as possible, and the decoded image displayed in the monitor can be similar to the original image.

Digital images are used every day. A digital image is essentially a 2D data array x (i,j), where i and j index the row and column of the data array, and each of the data point x (i,j) is referred as a pixel. For the gray image, each pixel is of an intensity value G. For color image, each pixel consists of a color vector (R, G, B), which represent the intensity of the red, green and blue components, respectively. Because it is the content of the digital image that matters the most, the underlying 2D data array may undergo big changes, and still convey the content to the user. An example is shown in

Figure , where the original image Lena is shown at left as a 2D array of 512x512. Operations may be performed on the original image so that it is suited for a particular application. For example, when the display space is tight, we may subsample the original image to a smaller image of size 128x128, as shown in the upper-right of figure . Another possible operation is to extract a rectangular region of the image starting from coordinate (256,256) to (384,384), as shown at the middle-right. The entire image may also be compressed into a compact bit stream representation, e.g. by JPEG, as shown in the bottom-right. In each case, the underlying 2D data array is changed tremendously. Nevertheless, the primary content of the image, the face of the girl remains legible to the user.

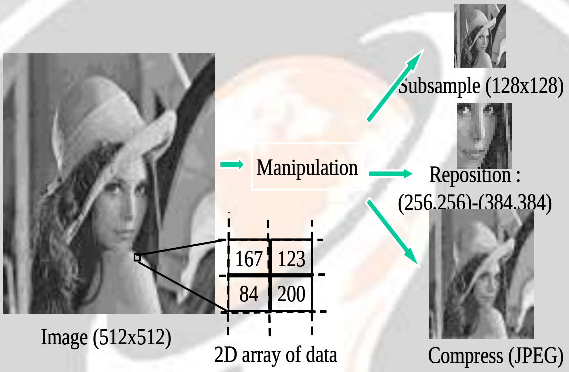

Figure 1 Digital image and image manipulation.

Among the operations, the compression creates a compact representation of the image data. It is an essential operation for image storage and transmission. In this paper, we focus our attention on JPEG 2000, which is a next generation image compression standard. JPEG 2000 distinguishes itself from older generation of compression standard, such as JPEG, not only by higher compression ratio, but also by an array of new functionalities. The most noticeable one among them is the scalable functionality. From a compressed JPEG 2000 bit stream, it is possible to extract a subset of the bit stream that decodes to an image of lower quality (with higher compression ratio), lower-resolution, and smaller spatial region. In other words, instead of manipulating the image in the space domain as shown in figure , we may manipulate the image directly on the compressed domain, and form a new bit stream that suits the application better.

QUANTIZATION

The quantization process of JPEG 2000 is very similar to that of a conventional coder, such as JPEG. However, the functionality is very different. In a conventional coder, the quantization process determines the allowable distortion of the transform coefficients, as the quantization result is lossless. In JPEG 2000, the quantized coefficients are embedded coded, thus additional distortion can be introduced in the following entropy coding steps. The main functionality of the quantization module is thus to map the coefficients from floating representation into integer so that they can be more efficiently processed by the entropy coding module. The image coding quality is not determined by the quantization step size δ, but by the subsequent bit stream assembler. The default quantization step size in JPEG 2000 is thus rather fine, e.g., $\delta=1/128$.

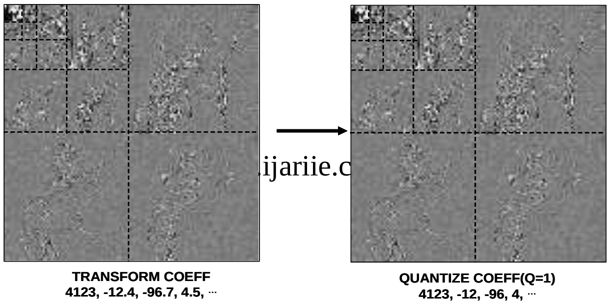

Figure 2 Quantization module

The quantized coefficients are partitioned into packets. Each sub band is divided into non-overlapping rectangles of equal size. Three rectangles corresponding to the same space location at the directional sub bands HL, LH, HH of each resolution level comprise a packet. The packet partition provides spatial locality as it contains information needed for decoding image of a certain spatial region at a certain resolution.

The packets are further divided into non-overlapping rectangular code-blocks, which are the fundamental entities in the entropy coding operation. By operating the entropy coder on relatively small code-blocks, the original and working data of the entire code-blocks can reside in the cache of the CPU during the entropy coding operation, thus greatly improves the encoding and decoding speed. In JPEG 2000, the default size of a code-block is 64x64.

VECTOR QUANTIZATION

A natural extension of ordinary scalar quantization schemes is to quantize multiple components simultaneously, also known as Vector Quantization. Instead of performing quantization on a single source input signal, a vector quantizer codes clusters of signals from multiple input sources by their centers. In an image coding context, a block of k pixels will form a vector. Each image pixel is considered to be an element in this k- dimensional vector. The block size is usually equal to $\sqrt{k} \times \sqrt{k}$ for image coding, which is a square block. This square block is also called a tile. Pictorial illustration can be found in Figure 1(a), where the image shown is in a size of 8×8 and a block is in the size of 4×4. Raster or Morton Z scanning pattern can be used within each block. The image is Vector Quantized using a block-based scanning strategy. Typical sizes for the block can be 2×2, 4×4, or 8×8, etc.

The pixel values in the $\sqrt{k} \times \sqrt{k}$ matrix block are then rearranged into a $1 \times k$ vector following a certain scanning sequence. A vector of the input signal as a whole forms the input to the vector quantizer. At both the encoder and the decoder sides of the vector quantizer, there is a set of k- dimensional vectors, which is called codebook of the vector quantizer. The vectors in the codebook are referred to as code- word or code vector. Each codeword is selected to quantize the input vector and is assigned a binary index in order to represent it. This vector is fed as the input to the Vector Quantizer, where the stored code words in the codebook have the same dimension as this vector. The graphical representation of the input image sequence vector and code words are illustrated in Figure 1(b). The parameter k is the Block size, for typical sized block, for instance a 4×4 block, $k = 4 \times 4 = 16$. The value N is the Codebook size, which indicates the number of code words that are stored in the codebook, the typical values are 16, 32, 64, 128, 256, 1024, etc.

NOVEL VECTOR QUANTIZATION

The natural extension of the concept of Differential Pulse Code Modulation [4] can also be applied in the ordinary Vector Quantization algorithm, which is referred to as the Novel Vector Quantization (NVQ).

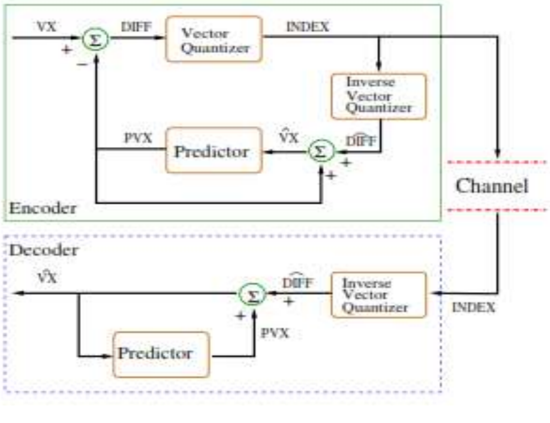

Figure 3 Novel Vector Quantization

The difference vector will then be compared with all the code words in the codebook that are stored in the vector quantizer. The codeword which is closest to the difference vector will be picked-up and added to the prediction vector to reconstruct the current block. The reconstructed vector is then used to calculate the prediction vector of the next iteration. Each codeword has its own binary label or index. The index of the codeword which has the lowest distortion will be sent to the decoder through a channel. On the decoder side, the received index value is used to retrieve the codeword that can best represent the difference vector. By adding the difference vector with the prediction vector, the reconstructed image vector can be obtained. Both the encoder and the decoder have the same decoding, prediction and reconstruction mechanisms. The prediction vector is obtained based on the reconstructed vector. This is because the decoder has no other side information about the original input vector, and therefore should not be used for performing the prediction in order to eliminate cumulative errors.

FIELD PROGRAMMABLE GATE ARRAY

The field-programmable gate array (FPGA) is a type of programmable device. Programmable devices are a class of general-purpose chips that can be configured for a wide variety of applications, having capability of implementing the logic of hundreds or thousands of discrete devices. Programmable read-only memory (PROM), erasable programmable read-only memory (EPROM) and electrically erasable PROM (EEPROM) are the oldest members of that class, while programmable logic device (PLD) and programmable logic array (PLA) represent more recent attempts to provide the end-user with an on-site customization using programming hardware.

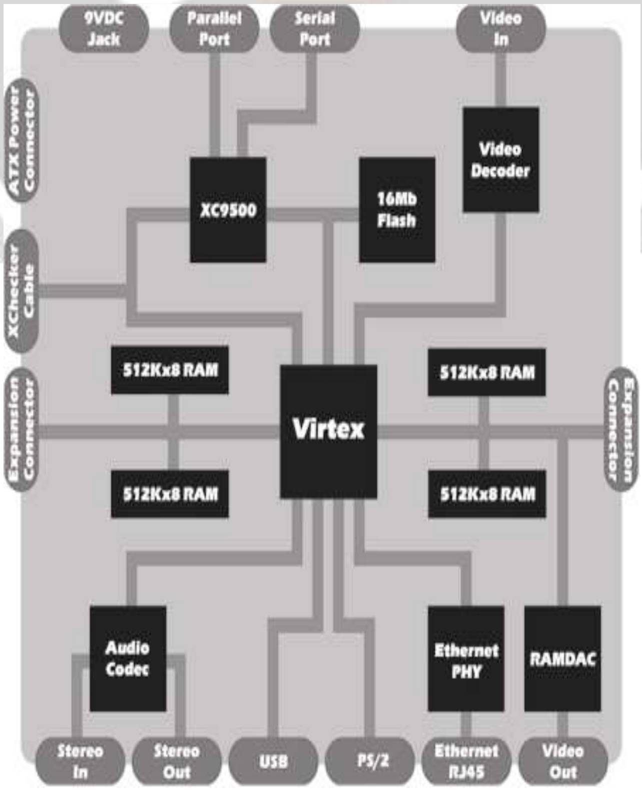

Figure 4 Virtex XSV800 Board Layout

This technology, however, is limited by the power consumption and time delay typical of these devices. In order to address these problems and achieve a greater gate density, metal programmed gate arrays (MPGA) were introduced to programmable logic industry market. An MPGA consists of a base of pre-designed transistors with customized wiring for each design. The wiring is built during the manufacturing process, so each design requires expensive custom masks and long turnaround time. Since FPGAs are application-specific ICs it is often argued that FPGA is an application specific integrated circuit (ASIC) technology.

RESULTS

The Vector Quantization and Novel Vector Quantization schemes are designed based on the window-based scan patterns. VQ and NVQ schemes with Raster and Morton Z scan are built. For the NVQ system, the predictor described earlier is used to estimate the current block that are being encoded.

FPGA DESIGN SUMMARY REPORT

VQ

LOGIC UTILIZATION	**USED**	**AVAILABLE**	**UTILIZED**
Number of Slice Flip Flops	1552	28672	5%
Number of 4 input LUTs	8822	28672	30%
Frequency (MHz)	75		

NVQ

LOGIC UTILIZATION	**USED**	**AVAILABLE**	**UTILIZED**
Number of Slice Flip Flops	4374	28672	15%
Number of 4 input LUTs	15209	28672	53%
Frequency (MHz)			

The tested images are not used for the generation of codebooks. This indicates that by coding the errors between blocks, the codebook is less dependent on images, which potentially implies that a fixed or universal codebook can be used to code any image. This is mainly due to the fact that the codebook generated from residual blocks is more compact than that of the ordinary blocks so that although the simple predictor is far from optimal, the overall performance of the NVQ system is quite satisfactory and robust.

The VQ scheme with Morton Z scan and the NVQ system with Raster scan is implemented in the FPGA platform based on XILINX Virtex-II. Images to be tested are in the size of 128×128. Raster Scan method is used for NVQ in order to reduce the required memory size for storing adjacent blocks. In the NVQ scheme, there are 7 pixel values that should be stored for each block. In total, $7\times (128/4) \times 8 = 1792$ registers should be allocated for storing neighboring blocks. The codebook size is fixed to 64, which translates into 0.375bpp. This size is chosen as it provides a good compromise between image quality and system complexity.

CONCLUSION

This paper presents a vector-based image compression algorithm, which consists of Vector Quantization scheme followed by novel compression technique. Even though the added DPCM block requires extra hardware resources, it was demonstrated that the codebook generated by coding the residual blocks makes the system more robust. The universal residual code- book can be versatile and applicable for different images. The proposed compression scheme was implemented on a reconfigurable platform. The moderate complexity of the algorithm makes it a very suitable candidate for on-chip compact and low power image compression algorithm that can be suitable for real time multimedia applications.

REFERENCES

[1] G. Iddan, G. Meron, A. Glukhousky and P.Swain "Wireless capsule endoscopy", *Nature*, pp.405 - 407, 2000.

[2] A. Olyaei, R. Genov, "Mixed-Signal Harr Wavelet Compression Image Architecture", *Midwest Symposium on Circuits and Systems(MWSCAS'05)*, Cincinnati, Ohio, 2005.

[3] Kawahito *et Al.*, "CMOS Image Sensor with Analog 2-D DCT-Based Compression Circuits", *IEEE Journal of Solid-State Circuits*, Vol. 32, No.12, pp.2029 - 2039, December 1997.

[4] Yan Wang *et Al.*, "*FPGA Implementation of a Predictive Vector Quantization image compression algorithm for image sensor applications", IEEE International Symposium on Electronic Design, Test & Applications, Vol.no.4, 2008.*

[5] Xilinx Corporation., "*VirtexTM 2.5 Volt Field Programmable Gate Arrays", DS003-2 (v2.8.1), 2002.*

[6] K. Andra, T. Acharya and C. Chakrabarti "*Efficient VLSI implementation of bit plane coder of JPEG2000", Appears in proceedings of SPIE Applications of digital image processing XXIV, vol. 4472.*

[7] D. Taubman, "*High performance scalable image compression with EBCOT", IEEE Transactions on Image Processing, vol.9, July 2000.*

[8] J. E. Fowler Jr., M. R. Carbonara, S. C. Ahalt, *"Image Coding Using Differential Vector Quantization", IEEE Transactions on Circuits andSystems for Video Technology, vol. 3, No. 5, Oct 1993.*

[9] S. A. Rizvi, N. M. Nasrabadi, "*Predictive Residual Vector Quantization", IEEE Transactions on Image Processing, vol. 4, No. 11, Nov 1995.*