VLSI Implementations of Compressive Image Acquisition using Block Based Compression Algorithm

Muthukumaran Narayanaperumal[1] and Ravi Ramraj[2]
[1]Department of Electronics and Communication Engineering, Anna University, India
[2]Department of Computer Science and Engineering, Anna University, India

Abstract: *In this research paper, consists of compressing the images within each pixel before the storage processes, hence the size of the memory gets reduced. This can be done by the proposed method namely block based compression algorithm which uses the differential coding scheme. Here, differential values are captured and then quantized. The differential coding scheme uses the concept of selecting the brightest pixel as the reference pixel. The difference between brightest pixel and subsequent pixel is calculated and quantized. Hence, their range is compressed and the spatial redundancy can be removed using block based compression algorithm. Thus, the proposed scheme reduces the accumulation of error and also, reduces the requirement of memory. Thus the Peak Signal to Noise Ratio (PSNR) value can be improved and Bits Per Pixel (BPP) value can be reduced. The future scope of the project is that the quality of the image can be further improved with high peak signal to noise ratio value using some other compression techniques.*

Keywords: *Image capture, image store, image compression, joint photographic experts group, PSNR, compression ratio, CMOS image sensor, VLSI implementation.*

Received May 28, 2013; accepted September 26, 2013; published online August 17, 2014

1. Introduction

The concept of an image has expanded to include three dimensional data sets and even four dimensional volume-time data sets [1]. Digital images are widely available from the internet, CD-ROMs and inexpensive charge coupled device cameras and scanners and frame grabbers [3, 7]. Software for manipulating images is also, widely available. Image refers to two dimensional light intensity function f(x, y) where, x and y denotes the spatial co-ordinates and the value of at any point (x, y) is proportional to the brightness of that point. Images can have either digital or analogue representation. The image is digitized first. A digital image f(x, y) has been discredited both in spatial co-ordinate and brightness [1, 8]. In digital representation of an image, it is represented as an array of numbers. The digitized image can be characterized by its intensity levels or scales of gray which range from 0 (black) representing the darkest intensity level to 255 (white) representing the brightest intensity level.

Each element in the image matrix is called image element or pixel. A typical size is a 512×512 array with 128 gray levels [9]. In a color image, the representation is similar except at each location of the matrix, the number represents three primary colors namely Red, Green and Blue (RGB) [9]. For a 24 bit color representation per pixel, the number is divided into three 8 bit segments. Each segment represents the intensity of one of the color primaries. The number of bits used to represent each pixel determines how many colors or shades of gray can be displayed [11, 12].

The objective of image compression is to decrease the number of bits required to store and transmit images without any measurable loss of information. Lossless compression techniques allow exact reconstruction of the original, but the achievable compression ratios are only of the order approximately 2:1 [10]. In general, the representation of digital image requires a large memory. The greater the size of a particular image, the greater the memory it needs. On the other hand, most images contain duplicate data. There are two duplicated parts of data in the image. The first is the existence of a pixel that has the same intensity as its neighboring pixels [1, 8]. These duplicated pixels waste more storage space. The second is that the image contains many repeated sections. These identical sections do not need to be encoded many times to avoid redundancies and an image compression is used to minimize the memory requirement in representing a digital image [6, 10]. The performance of the proposed image compression approach is analyzed with the terms such as gate, clock cycles required, power, processing rate and processing time [4].

2. Problem Formulations

The existing methods namely Wavelet Based Coding (WBC), Joint Photographic Experts Group (JPEG), Discrete Cosine Transform (DCT) based coding are the ancient methods which has many disadvantages such as: Complex hardware implementation, high cost and consuming very high power [14]. In order to overcome this, proposed method is taken into account.

2.1. JPEG

The JPEG compression method is a capable of compressing image data with a pixel value of reasonable speed and efficiency [8]. It is not a single algorithm and it may be thought a toolkit of image compression methods of the user. In this JPEG produce a very small and then the compressed images but poor quality so block based compression algorithm is the best methods for many applications.

JPEG is the lossy compression techniques but here with we have discussed about the block based compression algorithm for lossless compression. Using in this techniques we can get high compression ratio and low mean square error.

2.2. DCT

DCT is the technique for representing waveforms as a weighted sum of cosine DCT which is commonly used for data compression. It expresses a sequence of finitely many data points in terms of sum of cosine functions oscillating at different frequencies. It is a fourier related transform similar to Discrete Fourier Transform (DFT), but using only real transform. DCTs are equivalent to DFTs of roughly twice the length operating at real data with even symmetry, where in some variants, the input or output data are shifted by half a sample [2].

Each pixel value in the 2D matrix is quantized using 8 bits which produces a value in the range of 0 to 255 for the intensity/luminance values and the range of -128 to +127 for the chrominance values. All values are shifted to the range of -128 to +127 before computing DCT. All 64 values in the input matrix contribute to each entry in the transformed matrix. The value in the location F [0, 0] of the transformed matrix is called the DC coefficient and is the average of all 64 values in the matrix [2]. The other 63 values are called the AC coefficients and have a frequency coefficient associated with them. Spatial frequency coefficients increase as we move from left to right that is horizontal or from top to bottom that is vertical.

2.3. WBC

Wavelet series is the representation of real or complex valued function by the certain orthonormal series generated by the wavelet. It is the form of data compression which is well suited for image compression, audio and video compression [4, 8]. The goal of wavelet compression is to store data as much as possible in a file. Using the wavelet transform, the wavelet compression methods are adequate for representing transients such as percussion sound in audio or high frequency components in 2D images. This means the transient elements of the data signal can be represented by a smaller amount of information. Wavelet is a mathematical function that divides the data into different frequency components and then fits each component with a resolution suitable for its scale [8, 9]. Wavelet is a waveform that effectively has a duration limit of zero mean value. Some applications that have been successfully realized by utilizing such wavelet are image data compression, watermarking, edge detection, radar systems and encoding fingerprints. In fact, there are many coefficients in the representation of wavelet with very small or zero value [10, 12]. This characteristic gives the opportunity to perform image data compression.

3. Problem Solution

The procedure of the proposed method consists of taking the sample image which can be in any file format, compressing it and retrieving the original image. Thus, the size of the required memory would be reduced. The proposed compression algorithm uses the concept of differential coding scheme. In this scheme, there is an accumulation of error in sharp edges in the sample image. This is a serious drawback in differential coding scheme. In order to, avoid the accumulation of error, brightest pixel in the block is taken as reference pixel, which is used to encode the absolute values. The number of reference pixels is a tradeoff between quality of image and compression ratio. In the proposed compression algorithm, the sample image is decomposed into blocks. Within each block, the reference pixel is chosen. The reference pixel is taken as the brightest pixel in which differential values are always positive. The differential values between pixel within the block and reference pixel is calculated, then the differential values are quantized. As a result, the memory is compressed to the dynamic range. In addition to that, the differential values are always obtained with respect to reference pixel, since the accumulation of error is completely avoided. The proposed method is used to increase the value of Peak Signal to Noise Ratio (PSNR) and reduces the value of Bits Per Pixel (BPP) and compression ratio.

3.1. Sample Image

CMOS image sensor usage is neglected because of high cost and complexity of CMOS image sensor [1, 13]. Hence, the sample image is taken. The image taken will be in any of the following file format such as: JPEG, Tagged Image File Format (TIFF), GIFF and PNG. JPEG is designed for compressing either full color or gray-scale images of natural scenes. Photographic images are better stored in a lossless JPEG format. The JPEG format also is used as the image compression algorithm in many PDF files. The

TIFF remains widely accepted as a photograph file standard in the printing business. The TIFF is a flexible format that normally saves 8 bits or 16 bits per color namely RGB for 24 bit and 48 bit.

3.2. Compression

Compression is becoming important for video and still imaging devices. Image compression is the application of the data compression on digital images. The objective is to reduce the spatial and temporal redundancy of the image data in order to store or transmit data in an efficient form. Various image compression algorithms were proposed and established as standards, such as the joint photographic expert group standards, DCT based coding and the wavelet based coding. Compressive acquisition is to compress the image during acquisition which is performed by block based compression algorithm.

3.3. Quantization

Quantization is the process of mapping a large set of input values to a smaller set such as rounding values to some unit of precision. A device or algorithmic function that performs quantization is called quantizer, which reduces the inaccuracy of the output. The error introduced by quantization is referred to as quantization error or round off error. Quantization is involved to some degree in nearly all digital images processing, as the process of representing an image in digital form ordinarily involves rounding. Quantization forms the core of essentially all lossy compression algorithms. Because quantization is a many to few mapping, it is an inherently nonlinear and irreversible process, because the same output value is shared by multiple input values, it is impossible in general to recover the exact input value when given only the output value. The set of possible output values may be finite or infinite. The input and output sets involved in quantization can be defined in rather general way.

3.4. Spatial Redundancy

An element that is duplicated within a structure, such as pixels in a still image and bit patterns in a file is known as spatial redundancy. The analog signal is then quantized and digitized before being stored in a pixel-level memory. Then, the data will undergo the compression procedure to reduce the amount of data to be transmitted. A typical signal flow is therefore described by three basic steps namely: Capture→ store →compress. This consumes high memory space. In the paper, the issue of high storage requirement is proposed by compressing the data during image capturing procedure prior to storage phase. Storing the compressed data will leads to reduced memory requirement at the pixel level, hence smaller pixel size and improved fill factor is achieved.

3.5. Explanation of Block Based Compression Algorithm

The block size of an image is assigned. Clearly, the choice of block size must be informed, so as to achieve the best balance between image quality and compression. Initially the gray scale image is taken as input and initialization of image size is done. As the size of the input image (gray scale image) is not the multiples of block size, the image is resized and new dimensions of input image are obtained. The number of non-overlapping blocks is calculated by dividing the total size of an input image by the block size which is assigned. The logical matrix can be obtained from the condition of mean value. If the calculated mean is greater than the current block value, logical ‘1’ replaces the original value of the current block and if the calculated mean is less than the current block value, logical ‘0’ replaces the original value of the current block. From that, the logical 1’s are taken as K.

Secondly, the input is taken as color image which consists of three components namely RGB. In this, the red component is denoted as 1, green as 2 and red as 3. The steps are same as followed in the gray scale image. Hence, PSNR value can be obtained in which the green component has high PSNR value when compared to red and blue component. Finally, compression ratio and BPP value can be calculated for both gray scale image and color image.

3.6. BPP

BPP are defined as the number of bits of information stored per pixel in an image. If an image is 24 BPP, it is also, called a 24 bit image, a true color image, or a 16 M color image. Sixteen million is roughly the number of different colors that can be represented by 24 bits, where there are 8 bits for each of the RGB values. A 32 bit image is a specialized true-color format used in image files, where the extra byte carries information that is either converted or ignored when the file is loaded. If an image is 16 BPP, it is also, called a 16 bit image, a high color image or a 32 K color image. Thirty two thousand is roughly the number of different colors that can be represented by 16 bits, where there are 5 bits for each of the RGB values.

If an image is 8 BPP, it is also, called an 8 bit image or a 256 color image. Two hundred fifty-six is the number of different colors that can be achieved by using the image data as 8 bit indexes to an array of colors called a palette.

4. Result and Discussion

4.1. Input Image

Sample image is taken as an input. Input image may be in any several file formats such as: GIFF, TIFF, JPEG and PNG. In our project, the original image format is

JPEG. Various sizes of images can be taken as input. The given image size is chosen as 1024×1024.

Figure 1 shows the input image. The input image may be of grayscale or RGB image. The sample image is shown using the command 'imshow'. Then, the image is stored as pixel of matrix array format.

Figure 1. Input image.

4.2. Logical Matrix

Logical matrix is a matrix whose entries are either 1's or 0's. The sample image values which are in numerical format are converted into logical (binary) format i.e., logical 1's and 0's.

Figure 2 shows the logical matrix of the input image. The logical matrix has the values of 0's and 1's. Since, the image is not suitable for compression process, the image is converted into logical data and is suitable for that process.

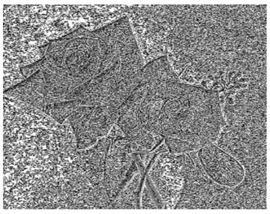


Figure 2. Logical Matrix.

4.3. Compressed Image

Original image is compressed using block based compression algorithm. The example block of size 4×4 is taken randomly from the logical matrix. A single block which is taken as follows:

$$\begin{pmatrix} 248 & 248 & 249 & 250 \\ 248 & 247 & 248 & 249 \\ 248 & 247 & 247 & 284 \\ 248 & 247 & 247 & 247 \end{pmatrix}$$

From the example block, reference pixel which has the greatest value is chosen as brightest pixel. The brightest pixel is given below. Brightest pixel=250. The difference between the brightest pixel value and neighboring pixel value is calculated. Differential values are calculated as below.

$$\begin{pmatrix} 2 & 2 & 1 & 0 \\ 2 & 3 & 2 & 1 \\ 2 & 3 & 3 & 2 \\ 2 & 3 & 3 & 3 \end{pmatrix}$$

For the example block, the mean value is calculated by adding the individual pixels and dividing the sum by total number of pixel values. Hence, the mean value is calculated as below:

$$Mean=247.8750$$

From the calculation of mean value, the example block is converted into binary value such as the value greater than or equal to mean is denoted as true (logical 1's) and the value less than mean is false (logical 0's). The resultant matrix is given below:

true	true	true	true
true	false	false	true
true	false	false	true
true	false	false	false

The sum of the true's (logical 1's) are calculated as *K* value which is equal to 10.

4.3.1. Compressed Output

The compressed output is obtained below with the size of 256×256 from the original image size of 1024×1024. The compressed image is then converted into the logical matrix to retrieve the original image.

The memory of the original image is found as 2097152 and the memory of the compressed image is obtained as 524288. From the memory, compression ratio and BPP values are calculated. CR=0.0208 and BPP=0.5000

Figure 3. Compressed image.

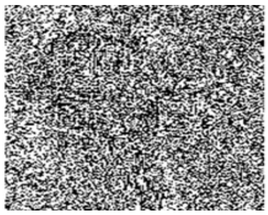


Figure 4. Logical Matrix.

4.4. Decompressed Image

The compressed image obtained is not suitable for general process. Thus, the decompression of that image is done to retrieve the original image. Thus, the decompressed image is the retrieved form of original image and is obtained in the size of 1024×1024.

Figure 5 shows the decompressed image which has the same quality as that of the original image without any information loss. For the good quality image, PSNR value will be high and BPP value is low.

Figure 5. Decompressed image.

4.4.1. Output Values of PSNR

The PSNR value for decompressed image is obtained as 48.6083 db for red component, 48.3268 db for green component, 47.7463 db for blue component.

Figure 6. Output values of PSNR.

4.4.2. Output Value of Compressed Ratio

It is the ratio of memory of compressed image (compressed size) to the memory of original image (uncompressed size).

Figure 7. Output value of CR.

Figure 8. Output value of BPP.

4.4.3. Output Value of BPP

The number of bits of information stored per pixel in an image is known as BPP.

BPP value is obtained as 1 for the block size of 4×4. Table 1 shows the PSNR values for RGB components, Compression ratio and BPP values for various block sizes such as 2×2, 4×4 and 8×8. For the block size 2×2, PSNR value is high but the image quality will be low. For the block size 8×8 the image quality will be good, but has low Peak signal to noise ratio value. For the block size 4×4 the quality of the image will be better with high PSNR value. Hence, 4×4 block size is chosen for better output.

Table 1. Comparison for various block size.

Block size	PSNR values			CR	BPP
	Red	Green	Blue		
2×2	54.6059	53.2721	52.7109	4	0.166670
4×4	48.6083	48.3268	47.7463	1	0.041667
8×8	45.8122	46.0519	41.4115	0.25002	0.010417

Table 2 shows the comparison of different image sizes such as 1024×1024, 512×512, 216×216 and their decompressed image.

Table 2. Comparison of different size of images.

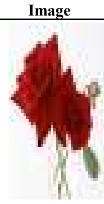
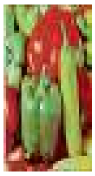
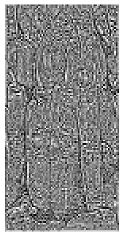
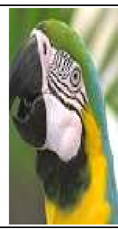
Image Size	Input Image	Compressed Image	Logical Data	Decompressed Image
1024×1024				
512×512				
216×216				

Table 3 shows the comparison of PSNR, CR and BPP values for various image sizes such as 1024×1024, 512×512, 216×216. For image size 1024×1024, PSNR value is high. This results in less memory requirement. For image size 512×512, PSNR value becomes low with increased BPP. For image size 216×216, PSNR value becomes very low with increased BPP value, when compared with other two image sizes.

Table 3. Comparison of PSNR, CR and BPP.

Image Size	PSNR Values			CR	BPP
	Red	Green	Blue		
1024×1024	48.6083	48.3268	47.7463	1	0.041667
512×512	43.3912	42.5526	42.8343	0.041669	1.0001
216×216	40.68	40.9171	40.6213	0.041681	1.0003

Table 4 shows the storage memory is reduced to a large extent. Here, the PSNR and BPP values were obtained as 30 db and 1.5 BPP in existing method. But, in our block based compression method, PSNR value is improved as 48 db and BPP value is reduced to 1 BPP. This enables more than 75% memory saving.

Table 4. Comparisons of existing method and proposed method.

Methods	PSNR	BPP
Existing method	30 dB	1.5
Proposed method	48 dB	1.0

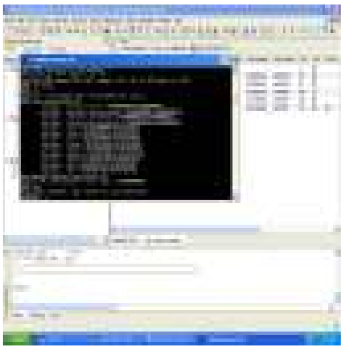


Figure 9. Xilinx platform studio.

Figure 10. Spartan 3EDK.

Figure 11. Spartan 3EDK with connected ports.

5. Conclusions

This differential coding scheme uses the reduced number of bits as the dynamic range is compressed. Block based compression algorithm is used to remove spatial redundancy. Quality of the image is further increased and requires less memory space. This improves the peak signal to noise ratio value. The BPP value gets reduced. Thus, the increased peak signal to noise ratio and decreased BPP values will result in better quality of the image. The future scope of the project is that the quality of the image can be further improved with high peak signal to noise ratio value using some other compression techniques.

Acknowledgement

This work was supported in part by Anna University Recognized Research Center Lab at Francis Xavier Engineering College, Tirunelveli, India.

References

[1] Bermak A. and Zhang M., "Compressive Acquisition CMOS Image Sensor: From the Algorithm to Hardware Implementation," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 3, pp. 490-500, 2010.

[2] Bermak A., "A CMOS Image with PFM/PWM based Analog-to-Digital Converter," *in Proceedings of IEEE International Symposium on Circuits System*, Arizona, USA, pp. 53-56, 2002.

[3] Carlson B., "Comparison of Modern CCD and CMOS Image Sensor Technologies and Systems for Low Resolution Imaging," *in Proceedings of IEEE Sensors*, Florida, USA, pp.171-176, 2002.

[4] Devangkumar S. and Chandresh V., "VLSI-Oriented Lossy Image Compression Approach using DA-Based 2D-Discrete Wavelet," *the International Arab Journal of Information Technology*, vol. 11, no. 1, pp. 59-68, 2014.

[5] El-Gamal A. and Eltoukhy H., "CMOS Image Sensors," *IEEE Circuits Devices Magazine*, vol. 21, no. 3, pp. 6-20, 2005

[6] El-Gamal A., "Trends in CMOS Image Sensor Technology and Design," *in Proceedings of International Electron Devices Meeting*, CA, USA, pp. 805-808, 2002.

[7] Fossum E., "Active Pixel Sensors: Are CCD's Dinosaurs?," *in Proceedings of the 3rd Charge-Coupled Devices and Solid State Optical Sensors*, *SPIE*, California, USA, pp. 2-14, 1993.

[8] Gupta A., Taubman D., and Nooshabadi S., "Near-Optimal Low Cost Distortion Estimation Technique For JPEG 2000 Encoder," *in Proceedings of IEEE International Conference Acoustics, Speech Signal*, Toulouse, France, pp. 14-19, 2006.

[9] Noble P., "Self-Scanned Silicon Image Detector Arrays," *IEEE Transactions Electron Devices*, vol. 15, no. 4, pp. 202-209, 1968.

[10] Palanisamy G. and Samukutti A., "Medical Image Compression using a Novel Embedded Set Partitioning Significant and Zero Block Coding," *the International Arab Journal of Information Technology*, vol. 5, no. 2, pp. 132-139, 2008.

[11] Renshaw D., Denyer P., Wang G., and Lu M., "ASIC Image Sensors," *in Proceedings of IEEE International Symposium on Circuits and Systems*, LA, USA, pp. 3038-3041, 1990.

[12] Theuwissen A., "CMOS Image Sensors: State-of-the-Art and Future Perspectives," *in Proceedings the 33th European Solid State Circuits Conference*, Munich, Germany, pp. 21-27, 2007.

[13] Tubaishat M. and Madria S., "Sensor Networks: An Overview," available at: http://robertdick.org/aeos/reading/tubaishat-sensor-nets.pdf, last visited 2003.

[14] Zhang M. and Bermak A., "A low Power CMOS Image Sensor Design for Wireless Endoscopy Capsule," *in Proceedings of IEEE Biomedical Circuits and Systems Conference*, Maryland, USA, pp. 2355-2358, 2007.

Muthukumaran Narayanaperumal received the BE degree in ECE from Anna University, India, in 2007 and the ME degree in Applied Electronics from Anna University, India, in 2010. Currently, he is working toward the PhD Information and Communication Engineering Degree, Anna University, India. Currently, he is working as an Associate Professor and Research Center lab in Francis Xavier Engineering College, Tirunelveli. His major research interests are image processing/compression, digital and analog very large-scale integration circuit design. He conducted several projects in the area of image processing, image compression, very large-scale integration circuit. Since 2008 he has published more than 3 journals in International and 18 National/International conferences papers. He is an Editor in Journal of Electrical Engineering and Technology (South Korea).

Ravi Ramraj is an Editor in International Journal of Security and its Applications (South Korea). He is presently working as a Professor and Head of the Department of CSE, Francis Xavier Engineering College, Tirunelveli. He completed his BE degree in Computer Science and Engineering from Thiagarajar College of engineering, Madurai in 1994 and ME degree in CSE from Jadavpur Government research University, Kolkatta. He has completed his PhD degree in Networks from Anna University Chennai. He has 18 years of experience. He is the first person in India from a private institution who was the judge (Chair person) for an International conference in IIT, Chennai. He published 12 International/National Journals and 4 international Journals are under process. He actively participated in 15 international Conference, 80 National Conferences and bagged shields in many. He is also a full time recognized guide for various Universities. Currently he is guiding 9 research scholars. His areas of interest are Virtual Private networks and Image Processing.