AREA EFFICIENT IMAGE COMPRESSION USING WAVE DECODING at CHIP LEVEL

K.Ambika
Assistant Professor
Department of Electronics and Comm. & Engg.
Mohammad Sathak A.J. College of Engineering

B.Thilagavathi
Assistant Professor
Department of Electronics and Comm. & Engg.

ABSTRACT

The recent emergence of new applications in area of wireless network based on low bandwidth protocols such as Zig Bee (up to 250Kbps) and Bluetooth (up to 1Mbps) and some biomedical applications (wireless camera pill) have created some challenges in the field of image processing. This can be overcome by the technique called "Image Compression". This paper proposes a new colour image compression technique based on quad tree decomposition. To reduce the computational complexity, the quad tree decomposition is applied on to each block of the image instead of applying it to the entire image. The experimental results have indicated that the performance of the suggested method is much better when compared to most popular existing encoders.

Keywords

Image Compression, Modulation, Differential Pulse Code Modulation, Quadrant Tree Decomposition, Adaptive Pulse Code Modulation, Wave decoding.

1. INTRODUCTION

Humans have always seen the world in colour but only recently have been able to generate vast quantities of colour images with such ease. In the last three decades, we have seen a rapid and enormous transition from gray scale images to colour ones. Today, colour and multi spectral properties of images are becoming increasingly crucial to the field of image processing often extending or replacing previously known gray scale techniques. In a digital true-colour image, each colour component is quantized with 8 bits and so a colour is specified with 24 bits. As a result, there are 224 possible colours for the image. Furthermore, a colour image usually contains a lot of data redundancy and requires a lot of storage space.

In order to lower the transmission and storage cost image compression is desired. JPEG-2000 suffers from ringing and blurring artifacts. In this paper an algorithm is proposed which improves the performance of colour image coding. This is possible by using efficient colour demonstration based on quad tree decomposition scheme. The input image is partitioned into small blocks. For each block, the number of colours is examined. If the block has one or two colours, no further processing is needed and the code for this block will be output; otherwise, the block is subdivided until each sub-block has 2 colours or less. After subdivision, the code of each sub-block will be output. The rest of this paper is organized as follows. Section 2 describes the proposed system. Section 3 describes about the Adaptive Pulse Code Modulation. Section 4 describes about Quadrant Tree Decomposition algorithm. Section5 describes about Wave decoding method. Section 6 concludes this paper.

2. PROPOSED SYSTEM

The above shown diagram is the overall architecture of the proposed system. The input image can be directly taken from the CMOS camera or a stored image is used. The image is first compressed using predictive method called Differential Pulse Code Modulation (DPCM) with Adaptive Quantization. This method produces output by comparing the result of the current pixel value with the previous pixel values. The difference of these will be compressed instead of compressing the original value. Following this Quadrant Tree Decomposition takes place. It is tree like structure which has a root node and child nodes. Here the entire image will be divided into quadrants.

If the difference between the pixel values in a quadrant is high then that particular quadrant will be sub-divided. Thus, this process will be repeated until a unique value of pixel is reached. After compressing the image Wave decoding is applied to this as this work is mainly focussed on reducing the area at chip level.

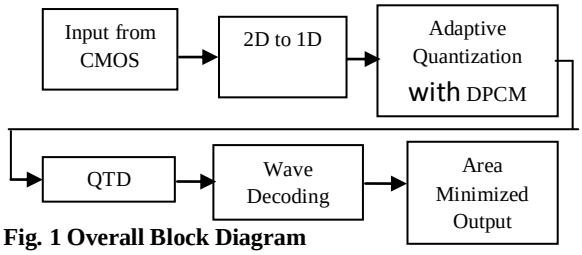

Fig. 1 Overall Block Diagram

3. ADAPTIVE DIFFERENTIAL PULSE CODE MODULATION (ADPCM)

DPCM scheme can be made adaptive in terms of the predictor or the quantizer or both. Adaptive Prediction usually reduces the prediction error prior to the quantization, and thus, for the same bit rate, the reduced dynamic range of the quantizer input signal results in less quantization error and better reconstructed image quality. On the other hand, adaptive quantization aims at reducing the quantization error directly by varying the decision and reconstruction levels according to the local image statistics. Non-adaptive predictors generally perform poorly at the edges where abrupt changes in the pixel values occur. An adaptive scheme improves the prediction in such edges.

This prediction is based on certain direction of the edges and on the previous pixel values. In adaptive prediction technique the prediction was scaled based on the previous reconstructed level. Using the same idea, an adaptive quantization scheme can also be developed

3.1. TYPES OF ADAPTIVE PREDICTION

There are two types of prediction namely,
Forward Adaptive Prediction
Backward Adaptive Prediction

3.2. FORWARD ADAPTIVE PREDICTION

3.2.1. It's based on the input of DPCM
3.2.2. More sensitive to variation
3.2.3. Depends on side information

3.3. BACKWARD ADAPTIVE PREDICTION

3.3.1 It's based on the output of the DPCM
3.3.2 Less sensitive to the variation
3.3.3 Independent of side information

3.4. BLOCK DIAGRAM OF DPCM

The proposed boundary adaptation scheme can be best described using an ordered set of boundary points. The quantization process is a mapping from a scalar-valued signal into one of the reconstruction intervals. Obviously, this Quantization process introduces quantization error when the number of quantization intervals is less than the number of bits needed to represent any element in a whole set of data. For a 1-bit quantizer, there will be just one adaptive boundary point delimiting two quantization intervals, with 0 and 255.

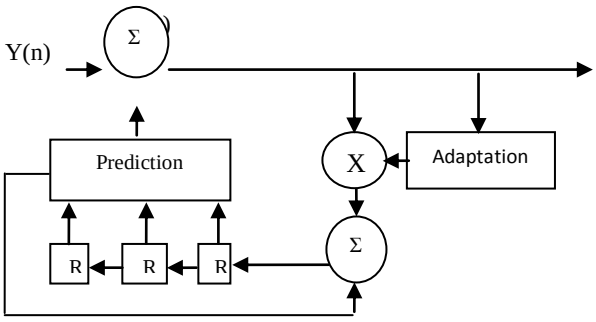

Fig. 2 Adaptive DPCM With Quantizer

In the proposed system, when a new pixel X (n) is read-out, its value is first estimated as BPp (n) through a backward predictor. Three registers denoted as, R1, R2, R3 are used to store the history values of the previously reconstructed pixels. The output value is then found by comparison method and the resulting value is taken as a codeword based on which the register value is updated and moreover it's also used to adjust the adaptation step size parameter. The step size is preferred to be large so as to track rapid fluctuations in the pixel value. On the other hand its preferred to be small so as to avoid large amplitude oscillations. To avoid the above said conditions we have to follow the rules given below. First is, if the quantization interval does not change between two consecutive pixel readings, we consider that current quantization parameter is far from the optimum value and so it's multiplied by the value greater than 1. Second is, if the quantization interval changes between two consecutive pixel readings, we consider that the current quantization parameter is near to the optimum value and so it's reset to the initial value.

4. QUADRANT TREE DECOMPOSITION

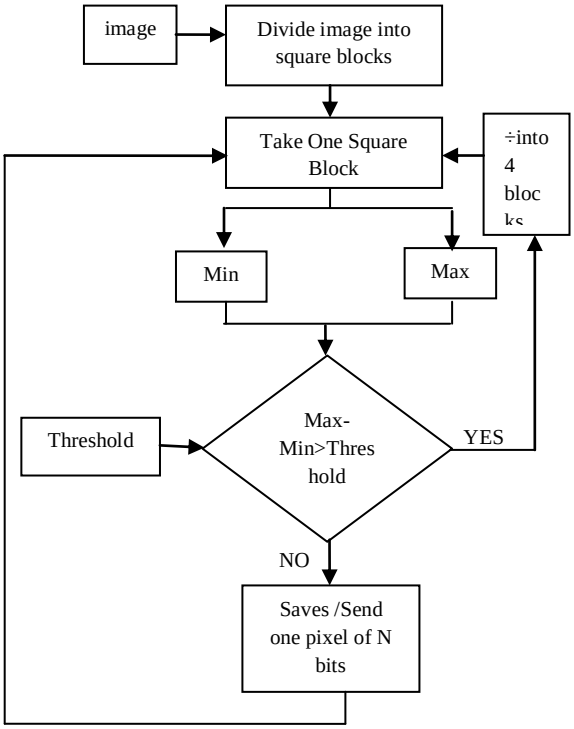

Fig. 3 Flowchart of QTD

It has a simple structure and can be implemented in analog/digital (mixed signal) hardware. The algorithm works as a tree. An image is partitioned into equal blocks. The size of the maximal block can vary from application to application. If a pixel with a minimal value and a pixel with a maximal value, belonging to the block, do not lie within a pre-specified interval (threshold), the block is further divided into four sub blocks. This process continues until all blocks meet the min/max criterion or reach the minimum size block, a 1*1 pixel. The criterion for block division can be different from the min/max criterion. This criterion was used here because of its simplicity of implementation for parallel hardware. After the image is partitioned into the blocks, it is enough to send/store only one pixel value from each block with additional information about the size of the block.

An example of the quad-tree representation is illustrated in Figure above [3], where the illustration is a block of size 8x8. There are a few ways to encode the quad-tree representation as shown in figure. For example, we may assign 1 to the internal node that needs to be further divided and 0 to a leaf node where no more subdivision is required. In the case of a leaf node, the colour of the corresponding array should be encoded immediately after 0. Let 0 and 1 represent colour green and blue, respectively.

Then, if we encode the quad- tree from the root to leaves in a depth-first order, the final binary code of the quad-tree reads as "1101101000000101010100001 1000" given numbers can be read in the format shown below. The first pixel of the figure will act as a parent node and the consecutive pixels will be the corresponding child nodes of the respective parent nodes. The nodes will be divided until a single pixel value is reached.

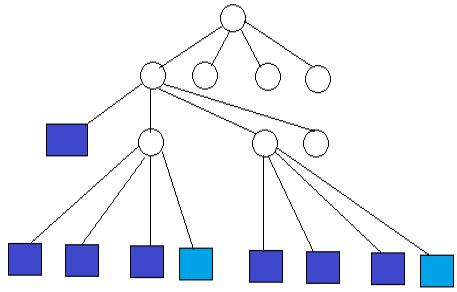

Fig. 4 Tree Structure

The below shown number format is an example for the working of Quadrant Tree Decomposition compression division.

"0000000" – undivided
"0000001" – one level of subdivision
"0000010" – two levels of subdivision
……
"1000000" – seven levels of subdivision.

The above given diagram is an example for the implementation of the quadrant tree decomposition method. The quadrants are divided until a single pixel value is found out. If there is any difference in the colour means then it will be divided, else it will be left as it is.

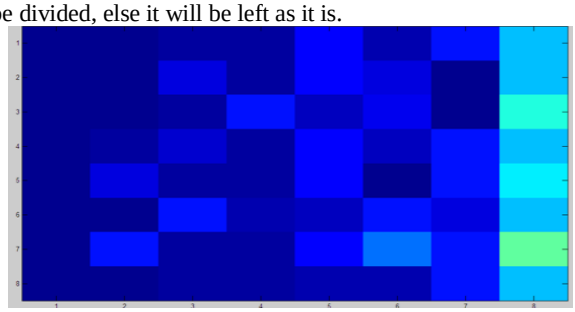

Fig. 5 8*8 Block Image

The compression procedure based on the Quadrant Tree Decomposition algorithm is performed by building a tree, which contains spatially redundant data in the quantized image. Multiple hierarchical layers of the tree, corresponding each to a square block within the pixel array, are constructed hierarchically in one iteration while the pixels are being scanned and quantized. Therefore, the proposed VLSI architecture enables one iteration adaptive quantization as well as the construction of the entire hierarchical structure of the quadrant tree during the scanning phase of the imager. Fig. 3 describes the addressing strategy of an array, which permits to systematically construct the corresponding QTD structure.

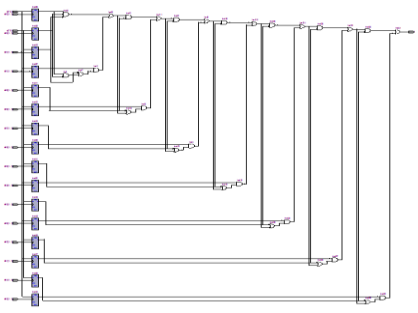

Fig. 6 Functional Diagram of QTD

For the sake of clarity, only a 4*4 pixel array is illustrated; however, the same concept is extended to any pixel array size. First, the array is divided into four quadrants. Each quadrant is further divided into four sub- blocks and two more bits are used in order to encode the new sub-tree. The overall structure now presents a root and 16 leaves, each of them are encoded using a 4-bit address. The procedure is repeated in a hierarchical way until the image is segmented down to the pixel level. The leaves of the tree correspond to the pixels of the array image, and each hierarchical level within the tree corresponds to a quadrant grouping of the image array. An important and interesting feature in this addressing methodology is the mapping relationship between the pixels' address in the tree and that in the pixel array. It can be easily observed that the row and column addresses are the even and the odd bits of the pixel's tree address. The QTD circuit for a 1-bit Q. Note that higher number of bits are obtained using a digital comparator instead of single XNOR (equality) gate. The circuit operates in two modes, namely: 1) tree construction and 2) tree trimming modes. In the first mode, the goal is to obtain the flag bits for all layers of the tree while scanning the quantized pixel values. Once all of the tree is constructed, the trimming mode starts and is performed in parallel for the entire tree structure using a single clock cycle. In this mode, the Const/Trim signal provided by the control circuit is set to "0," allowing to reset the flag bits of a given layer if and only if its parents present a flag bit value equal to "1." For example, if the root flag is equal to "1," all flip-flops belonging to lower layers are reset in one clock cycle using the flag bit of the root. Once the tree is trimmed, first the flag bits of the tree are transmitted to the receiver end. The flag bits are used to control the read-out sequence such that compressed pixels are skipped while transmitting the compressed image data. This procedure does not require buffering the image data as the pixel array is accessed in order to retrieve non-compressed pixel values. The adaptive Q is enabled again during this phase and is performed only on non-compressed quadrants.

5 .WAVE DECODING

Wave Decoding is a method to reduce area. Flip-flops can be replaced by simple buffers. This will reduce the area to a greater extent and also the clock period can be reduced at such times while using buffer instead of combinational circuits. Reduction of the clock period will increase the speed of the device to a certain extent. Wave decoding is a undergoing research process for low cast VLSI design to reduce the area. Since it is mainly concentrated to reduce the unwanted delays used in the verilog design, it slightly speed-up the processing time. It is concentrated to gate level logic design compared to some sequential modelling because of its high processing delays. Normally the decoding process is one of the main procedure for controlling and commanding the hardware unit to do desired application. The single pipeline decoder and double pipeline decoders are normal in the VLSI design. They have the ability to control the process but uses more delay unit to operate without any collapse. But Wave decoding is mainly used for reducing these delay elements while there is a input which has no possibilities of collapse. Traditional combinational circuits uses delay fault model. There are two types of faulty model normally plays major role in the combination logic namely lump delay at faulty gate and also

the path delay. Wave Decoding avoids overheads, there is no internal synchronizers as buffers are used. Main advantage is that it consumes low power and less area. This method is proposed and the area can be reduced by using the same.

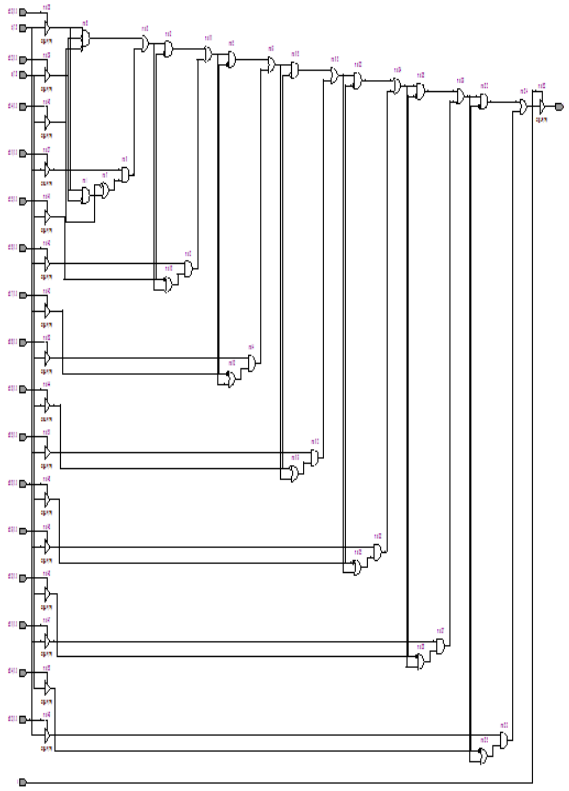

Fig. 7 QTD Using Wave Decoding

6. SUMMARY

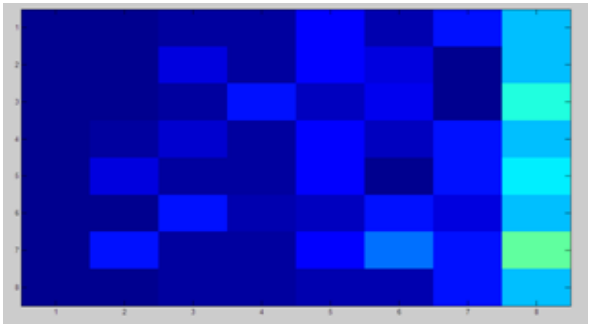

Fig. 8 Matlab input

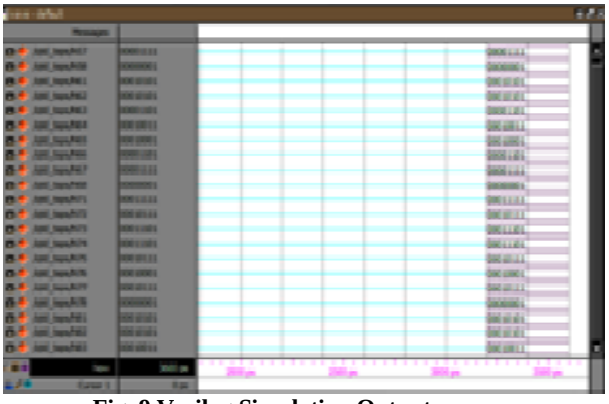

Fig. 9 Verilog Simulation Output

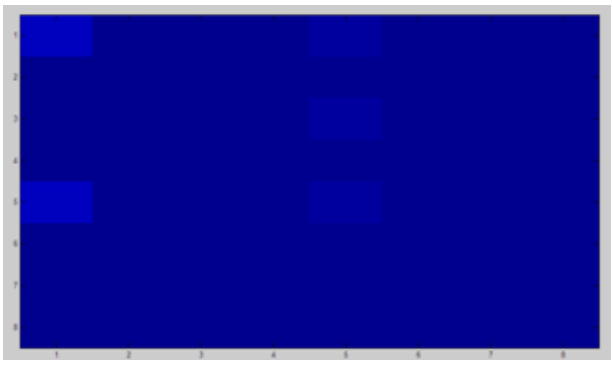

Fig. 10 Matlab output

ise Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,175	1,536	76%	
Number of 4 input LUTs	19,182	1,536	1248%	OVERMAPPED
Logic Distribution				
Number of occupied Slices	9,878	768	1286%	
Number of Slices containing only related logic	9,418	9,878	95%	
Number of Slices containing unrelated logic	460	9,878	4%	
Total Number of 4 input LUTs	19,327	1,536	1258%	OVERMAPPED
Number used as logic	19,182			
Number used as a route-thru	145			
Number of bonded IOBs	2	94	2%	
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	

Fig. 11 Xilinx output without WD

Project File:	ise.ise	Current State:	Placed and Routed
Module Name:	quadtree	• Errors:	
Target Device:	xcv600e-8bg432	• Warnings:	
Product Version:	ISE 10.1 - WebPACK	• Routing Results:	
Design Goal:	Balanced	• Timing Constraints:	
Design Strategy:	Xilinx Default (unlocked)	• Final Timing Score:	

ise Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	7,429	13,824	53%	
Number of 4 input LUTs	11,613	13,824	84%	
Logic Distribution				
Number of occupied Slices	6,910	6,912	99%	
Number of Slices containing only related logic	4,784	6,910	69%	
Number of Slices containing unrelated logic	2,126	6,910	30%	

Fig. 12 Xilinx output with WD

A Simple but efficient colour image compression technique based on Quadrant Tree Decomposition is presented here. Computation of the proposed method makes, it to appear very simple when compared with the other algorithms. An 8*8 colour image is compressed efficiently using this algorithm. The compression is done using HDL codings which can be implemented in FPGA kit. This work is mainly focussed on area minimization at chip level using Wave Decoding method. By using this wave decoding method the number of gates can be reduced. In this paper the gates are reduced to the maximum using this method and it is shown in the Xilinx report.

7. REFERENCES

[1] Antonio Ortega, and Martin Vetterli, (1997,May) "Adaptive Scalar Quantization Without Side Information", *IEEE Transactions On Image Processing*, vol. 6, No. 5, pp.665 676.

[2] E. Artyomov, G. Levi, Y. Rivenson, O. Yadid-Pecht, (2005,July) "Morton (Z)scan based real-time variable resolution CMOS image sensor," *IEEE Trans. Circuits System I*, Reg. Papers, vol. 15, no. 7, pp. 947-952.

[3] Artyomov and O. Yadid-Pecht, (2006,Oct)"Adaptive multiple-resolution CMOS active pixel sensor," *IEEE Trans.Circuits System I*, Reg. Papers, vol. 53, no. 10, pp. 2178-2186.

[4] S.Chen, A. Bermak, D. Martinez and W. Yan (2007,Jan)"Adaptive-quantization digital image sensor for low-power imagecompression,"*IEETran Circuit System I*, Reg. Papers, vol. 54, no. 1, pp. 13 .

[5] Hanan Samet, (1985,Jan) "A Top Down Quad Tree Traversal Algorithm",*IEEE Trans. on Pattern Analysis And Machine Intelligence*, vol. PAMI-7, no.1, pp. 94-98.

[6] A. Kitchen, A. Bermak and A. Bouzerdoum (2005,Dec) "A digital pixel sensor array with programmable dynamic range," *IEEE Trans. Electron Devices*, vol. 52, no. 12, pp. 2591-2601.

[7] M. W.Hoffman, W. D. Len-Salas, K.Sayood and N.Schemm (2007,November)"A CMOS imager with focal plane compression using predictive coding," *IEEE J. Solid-State Circuits*, vol. 42, no. 11, pp. 2555-2572.

[8] Z. Lin, M. W. Hoffman, W. D. Leon-Salas, N. Schemm and S. Balkir (2008,Octber)"A CMOS image sensor for multi-level focal plane image decomposition", *IEEE Trans. Circuits System I*, Reg. Papers, vol. 55, no. 9, pp. 2561-2572.

[9] Milin Zhang, and Amine Bermak, (2010,March)"Compressive Acquisition CMOS Image Sensor: From the Algorithm to Hardware Implementation", *IEEE Transactions on Very Large Scale Integration (VLSI)*,vol. 18, no. 3, pp. 490-500.

[10] Shoushun Chen, Amine Bermak, and Yan Wang, (2011,April)"A CMOS Image Sensor With On-Chip Image Compression Based on Predictive Boundary Adaptation and Memoryless QTD Algorithm", *IEEE Transactions on VLSI*, vol. 19, no. 4, pp. 538-547.