Abstract: An advanced algorithm of online image memory size compression has been introduced in this paper. There are 3 parts into which whole process of online image memory size compression is divided. These are 1) Image capture 2) Pixel level Image compression 3) Image mapping and storage. A block based compression algorithm is being proposed in this. The reference pixel that is the brightest pixel value is chosen first of all and then we compare all the pixel values of the block with the pixel value of reference pixel. After that a comparator circuit is being employed so that in consonance with the proposed mapping the differential values of the successive pixels are further compared through comparator circuit. And then by using the intermediate precision bits selected number of differential values are quantized and calculated. Thus the bits required for on-pixel memory storage are reduced. This enables reduced silicon area for System on chip. Mentor Graphics’ Design Architect tool is being used to synthesize the algorithm.

Keywords: Compressive sensing, Digital pixel sensors (DPSs), Block – Based compression algorithm.

I. INTRODUCTION

The history of image processing technology in electronics dates from the invention of television system in 1927 [5]. From this time, the image processing technology has been studied and developed mainly in terms of television broadcasting technology, such as NTSC or PAL. On an image capturing devices, the solid-state imagers have been invented in 1970s, which employ the CCD (charge coupled device) as a signal transformer [6]. The recent development of VLSI technologies enables us to fabricate not only photo-detectors and signal transfer circuits, but also a simple image processing circuits in one image sensor chip, so called computational sensors [7]. Image sensors are basically distinguished by their pixel sensors i.e. whether it is active pixel sensor or passive pixel sensor [8]. Passive imagers require less amount of MOS devices or lesser silicon area but correspondingly active pixels with a trade-off provides better picture quality. Hence, most of image sensors are active pixel image sensor. However the history of CMOS active pixel sensors is not as long as it is for CCD active pixel sensors. First research paper on CMOS image sensor in IEEE Xplorer was published on ‘A CMOS facsimile video signal processor’ [9] in 1985. From the introduction of CMOS in image sensors, it is used in image processing as analog pixel part or as a processor. In image sensors introduction of image sensor reduces power dissipation, improve switching capability and helpful in analog to digital conversion due to its switching capability. In 1994, CMOS image sensors were reported to be described in the concept of image compression [10]. A CMOS photodiode was used to reduce the dynamic range [11], as reduced dynamic range require lesser number of bits for digital data storage. So from that on the concept of compression or rather on-pixel compression was directly linked with dynamic range of frame. However, various algorithms and methodologies were applied to reduce the dynamic range but one commonly used methodology was to capture >> store >> compress technique. There are various hardware friendly algorithms [12] designed for such methodology. In modern era a new methodology of image acquisition and online image compression is used and being implemented, it is capture >> compress >> store [13]. As this approach of image compression is newer one, hence there is not much work reported in this area. Due to its enhanced results over old methodology, this technique has large potential of research.
II. DESIGN ALGORITHM

A. Block-Based Compression

In various image compression algorithms whole image captured by optical device is divided into blocks instead of using whole image. The concept of block-based compression enables a reduced dynamic range of differential pixel values for adaptive quantization.

Here are some block-level distributions of image.

(1) In this block representation all pixels are provided with a full precision of 8-bits to each pixel. Such kind of distribution is treated as original image acquisition.

(2) In this block representation we choose the brightest pixel value in the block and provide it full precision of 8-bits. While all other pixels are captured with an inter-medium precision of 3 bits. As the brightest pixel value represents the highest intensive pixel, so comparing the brightest pixel with neighbouring pixel will provide a reduced dynamic range. A reduced dynamic range is so as captured with reduced number of bits.

(3) This is the proposed image compression algorithm, in which the brightest pixel is captured with full precision of 8-bits. And then after the block is sub-divided into smaller blocks (e.g. 2×2). The pixel of each sub block is captured as differential pixel value with an inter-medium precision of 3-bits. While next pixel is stored with 1-bit. The differential values of previous pixel and current pixel are compared using a comparator [29]-[30], and 1-bit output of comparator is stored.

B. Mapping of pixels

In the mapping of pixels it is compulsory to trace the flow of data at pixel and block level. Here is an introductory data flow block representation.

The mapping of pixels in image acquisition helps to optimize the image compression. Here are some examples of different kind of mapping usually found in image compression algorithms.
Linear scanning
In linear scanning, image pixels are scanned row wise i.e. first of all first row of pixels will be scanned and then after it moves to next row within a block as shown in Fig. 2.4 and blocks are also scanned in similar fashion [23].

<table>
<thead>
<tr>
<th>I(i,0)</th>
<th>I(i,1)</th>
<th>I(i,2)</th>
<th>I(i,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I(i,4)</td>
<td>I(i,5)</td>
<td>I(i,6)</td>
<td>I(i,7)</td>
</tr>
<tr>
<td>I(i,8)</td>
<td>I(i,9)</td>
<td>I(i,10)</td>
<td>I(i,11)</td>
</tr>
<tr>
<td>I(i,12)</td>
<td>I(i,13)</td>
<td>I(i,14)</td>
<td>I(i,15)</td>
</tr>
</tbody>
</table>

Fig3.1. Linear scanning of pixels within a block

Linear sub-block scanning
In linear sub-block scanning, whole block is firstly sub divided in 2×2 blocks and then performs linear scanning sub-block wise instead of block wise as shown in Fig. 2.5 [14].

<table>
<thead>
<tr>
<th>I(i,0)</th>
<th>I(i,1)</th>
<th>I(i,2)</th>
<th>I(i,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I(i,4)</td>
<td>I(i,5)</td>
<td>I(i,6)</td>
<td>I(i,7)</td>
</tr>
<tr>
<td>I(i,8)</td>
<td>I(i,9)</td>
<td>I(i,10)</td>
<td>I(i,11)</td>
</tr>
<tr>
<td>I(i,12)</td>
<td>I(i,13)</td>
<td>I(i,14)</td>
<td>I(i,15)</td>
</tr>
</tbody>
</table>

Fig3.2. Linear sub-block scanning

<table>
<thead>
<tr>
<th>I(i,0)</th>
<th>I(i,1)</th>
<th>I(i,2)</th>
<th>I(i,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I(i,4)</td>
<td>I(i,5)</td>
<td>I(i,6)</td>
<td>I(i,7)</td>
</tr>
<tr>
<td>I(i,8)</td>
<td>I(i,9)</td>
<td>I(i,10)</td>
<td>I(i,11)</td>
</tr>
<tr>
<td>I(i,12)</td>
<td>I(i,13)</td>
<td>I(i,14)</td>
<td>I(i,15)</td>
</tr>
</tbody>
</table>

Fig3.3. Hilbert mapping of pixels.

In Hilbert mapping, image pixels are addressed as per Hilbert curve. However Hilbert mapping help improve in reducing the temporal and spatial redundancies. And also provides a good speed of sampling. But Hilbert mapping changes with number of pixels per block and also with number of blocks per frame. And also this mapping scheme is not regular one for each sub block [15].

<table>
<thead>
<tr>
<th>I(i,0)</th>
<th>I(i,1)</th>
<th>I(i,2)</th>
<th>I(i,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I(i,4)</td>
<td>I(i,5)</td>
<td>I(i,6)</td>
<td>I(i,7)</td>
</tr>
<tr>
<td>I(i,8)</td>
<td>I(i,9)</td>
<td>I(i,10)</td>
<td>I(i,11)</td>
</tr>
<tr>
<td>I(i,12)</td>
<td>I(i,13)</td>
<td>I(i,14)</td>
<td>I(i,15)</td>
</tr>
</tbody>
</table>

Fig3.4. Proposed pixel mapping

In our proposed pixel mapping we will select the brightest pixel value from the block and provide it full precision of 8-bits. Then after we will calculate differential pixel values by comparing reference pixel with other pixel and store the differential values using a 3 bit precision. The remaining pixels are compared differential pixel values previous differential value and current differential value. These pixels are stored using 1-bit precision.
III. SYNTHESIS RESULTS

TABLE I
SIMULATION RESULTS FOR A CROPPED 20×32 PIXEL CAMERAMAN IMAGE OF A PRECISION OF 8-BITS PER PIXEL

<table>
<thead>
<tr>
<th>Testing samples</th>
<th>Block size</th>
<th>Quantizer bits</th>
<th>PSNR (dB)</th>
<th>Bits per block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman cropped image</td>
<td>4</td>
<td>3</td>
<td>20.52</td>
<td>32</td>
</tr>
<tr>
<td>Cameraman cropped image</td>
<td>4</td>
<td>4</td>
<td>21.21</td>
<td>36</td>
</tr>
</tbody>
</table>

A. Synthesis results

In the synthesis of our algorithm firstly we design a pixel level CMOS Active Pixel circuit. As per the requirement of our algorithm we design this circuit. After designing the pixel level circuit we form a block level arrangement of pixels for testing.

The dynamic range of differential pixel output can be then arbitrated from the synthesis of whole block. Figure 4 shows the output for the Pixel 2, where Vfire and Vdiff2 are inputs to comparator of pixel 2 and Vout2 is output of pixel 2.

By the analysis of an ordinary photodiode following data comes in existence.

TABLE II
SHORT CIRCUITED CURRENT, OPEN CIRCUITED VOLTAGE AND INPUT INTENSITY RELATIONSHIP FOR AN ORDINARY PHOTODIODE

<table>
<thead>
<tr>
<th>Light</th>
<th>Intensity (I&lt;sub&gt;x&lt;/sub&gt;)</th>
<th>Current (I&lt;sub&gt;sc&lt;/sub&gt;)</th>
<th>Voltage (V&lt;sub&gt;oc&lt;/sub&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct sunlight</td>
<td>1000</td>
<td>30 uA</td>
<td>500 mV</td>
</tr>
<tr>
<td>Overcast day</td>
<td>100</td>
<td>3 uA</td>
<td>440 mV</td>
</tr>
<tr>
<td>Twilight</td>
<td>1</td>
<td>0.03 uA</td>
<td>310 mV</td>
</tr>
<tr>
<td>Full moon night</td>
<td>0.1</td>
<td>3000 pA</td>
<td>280 mV</td>
</tr>
<tr>
<td>Clear night / no moon</td>
<td>0.001</td>
<td>30 pA</td>
<td>250 mV</td>
</tr>
</tbody>
</table>
IV. ALGORITHM IMPLEMENTATION

A. Overview

Fig. 5 illustrates the architecture of the overall system. The sensor array is divided into 4×4 blocks. A single pixel consists of a MOS switch, a photodiode, a differential amplifier stage, and a comparator. As shown in Fig. 5, the whole image is divided into small blocks, and for smooth mapping of pixels and blocks controller bits are required. There are two blocks select controllers i.e. row blocks select controller and column blocks select controller. Memory size of these controllers varies with respect to size of block. This is the block based representation of the image. However, in block-based architecture, it is also quite possible to compress the image and store it with reduced number of bits. Hence, various decomposition theorems are here to compress the image at block level prior to storage like QTD i.e. quadrant tree decomposition. A MOS switch is used to level up Vn voltage to Vdd. And also, a MOS switch distinguishes one frame with another and hence reduces power dissipation. A photodiode is used as a sensor device which converts light intensity into equivalent voltage as described in TABLE II. Hence, voltage at node Vn will be proportional to light intensity.

In our design, we use 3-4 bits counter to count the differential pixel values and correspondingly store them in on-chip memory.
(B) 4×4 Bloc
(A) Sensor Array
V. CONCLUSION

In this paper, online image compression concept is adduced and is synthesized using Mentor Graphics Design Architect tool. The paper illustrates the advantages of proposed algorithm 1) reduced silicon area required for DPS; 2) improved fill-factor; 3) online compression processing, which enables the concept of parallel processing. Results illustrate that the proposed algorithm results in more than 50% on chip memory.

VI. FUTURE WORK

This work was synthesized using Design tools, however capturing images using camera and processing images on hardware implemented design may provide enhanced results.

REFERENCES


(C) Pixel

Fig5. Overview of the overall system (a) Architecture of the sensor array. (b) Architecture of one block. (c) Circuit diagram of one pixel.


