Image Progressive Acquisition for Hardware Systems

Jianxiong Liu, Christos Bouganis, Peter Y.K. Cheung
Department of Electrical and Electronic Engineering
Imperial College London
Email: {jianxiong.liu09, christos-savvas.bouganis, p.cheung}@imperial.ac.uk

***Abstract*—As the resolution of digital images increases, accessing raw image data from memory has become a major consideration during the design of image/video processing systems. This is due to the fact that the bandwidth requirement and energy consumption of such image accessing process has increased. Inspired by the successful application of progressive image sampling techniques in many image processing tasks, this work proposes to apply similar concept within hardware systems to efficiently trade image quality for reduced memory bandwidth requirement and lower energy consumption. Based on this idea, a hardware system is proposed that is placed between the memory subsystem and the processing core of the design. The proposed system alters the conventional memory access pattern to progressively and adaptively access pixels from a target memory external to the system. The sampled pixels are used to reconstruct an approximation to the ground truth, which is stored in an internal image buffer for further processing. The system is prototyped on FPGA and its performance evaluation shows that a saving of up to 85% of memory accessing time and 33%/45% of image acquisition time/energy is achieved on the benchmark image "lena" while maintaining a PSNR of about 30 dB.**

I. Introduction

Nowadays, image capturing devices enjoy image sensors with increased pixel resolution. Consequently, image processing systems have to deal with images with ever growing size which leads to higher memory bandwidth requirements and higher energy consumption when such images are accessed. Moreover, this trend becomes even more dominant in the design of hardware systems as the performance gap between the processing cores and memory elements tends to become wider as the technology scales [1][2].

Many works try to address this problem from different aspects. The hierarchical memory systems are extensively used where the locality of the data is often exploited for lower data accessing cost [3]. Specific to image processing systems, fast but expensive local buffers are widely used to temporarily store image patches read from external slow memory, for processing unit to access in a convenient manner [2]. The use of image buffer is also due to the requirement by the nature of many image processing algorithms, that image data is often random-accessed repeatedly within a small local region.

Most of these studies target DRAM as it is the most common type of memory in modern hardware systems due to its large capacity and well developed structure [2]. The special structure of DRAMs is often exploited to reduce not only the image acquisition time but also the image access energy. Common strategies of image accessing/storing aim to reduce the number of row activations of the target DRAM as it is the most time consuming and energy hungry among DRAM operations. On the other hand, the spatial locality of image data transforms into temporal correlation of signals on data bus. Therefore data bus energy consumption during image accessing is also studied on top of general purpose computing systems [4]. There are also dedicated image accessing systems built for specific image processing tasks such as video decoding process, where the energy consumption of the memory accessing process is even more dominant than that of the decoding process in certain situations [5].

This work approaches the problem from a different perspective. Inspired by the successful application of image progressive acquisition techniques in many fields [6] [7], this work applies the concept of progressive sampling to the image accessing process in hardware systems. The key idea is to only access pixels of statistical importance, that is pixels that cannot be easily predicted and have a high variance potentially, and interpolate the rest of the pixels internally in order to obtain an approximation of the original image. In this way the system can trade off between image quality and the time/energy cost of image acquisition process. A hardware block is proposed which alters the conventional memory access pattern. It is compatible with common memory access interface and can be adopted by such interface as a support module with ease. With DRAM as an example target memory, the evaluation shows the capability of the proposed system to efficiently trade image quality for reduced bandwidth requirement and lower time/energy consumption of the image acquisition process.

II. Background

A. Image access in hardware systems

Natural images are two dimensional data matrices that usually have significant spatial correlation. Such image data is often accessed by image processing algorithms in a "block-type" pattern [2], such as in DCT, feature extraction, and video decoding systems. This pattern refers to the processing of image data in nested loops. The outer most loops can be seen as a moving analysis window that captures a local region (macro block) of the image, while the inner loops are operations centred on each pixel. These operations often see each pixel used multiple times across consecutive loops and therefore if all pixels within a local area are readily buffered into the internal fast memory, the total cost (time and energy) of the algorithms is reduced. The conventional access pattern sees all pixels of an image macro block read from the memory row by row and stored in the local buffer to be access by the client unit. According to the structure of memory systems, optimizations have been made to accelerate the process and/or lower its energy consumption by reorganising the access patterns [2][4].

978-3-9815370-2-4/DATE14/©2014 EDAA

As a well studied field of research, image compression is useful in reducing the size of image data and therefore saving bandwidth needed for accessing the image. But in most image processing systems, the cost of performing such compression/decompression on intermediate data is prohibited and outweighs any benefits of the approach. In addition, image processing algorithms usually require random accessibility of pixels and this makes it even more impractical to compress large chunks of an intermediate image [2]. To preserve the random accessibility and control the cost of compression/decompression, people have developed the frame recompression technique [8][9], which is essentially a light weight local compression algorithm.

The proposed sampling method approaches the problem from a different perspective, which is to reduce the cost of image acquisition by reducing the number times of memory accessing. Therefore it does not conflict with existing data management strategies and image recompression techniques in implementation and can be used as a supplement to existing systems. In this paper however, the discussion is on raw pixels and only includes linear/block mapping the image as the data organizing strategy as it is a widely used strategy and allows a better characterisation of the benefits of the proposed system.

B. Image progressive sampling

The family of progressive image transmission (PIT) algorithms aims to organize and transmit image data in a way that most "significant" bits of the data are transmitted first. As such they are designed to make efficient use of otherwise limited bandwidth to transmit image data. Most techniques in this family, such as those used in image compression standards, rely on pre-processing of the ground truth image to better organize the image data and therefore achieve a better quality to bandwidth ratio. The techniques of image progressive sampling however, specifically targets the scenario where such pre-processing is not available and blind point sampling is the only option. In these situations, image progressive sampling relies on stochastic method that iteratively samples pixels while refining the underlying model. As established early by [10], based on already sampled pixels, the next sampling locations are determined by candidates' priority scores that are computed from the local estimated statistics in the neighbourhood:

$$f(\mathbf{x}_i) = \|\mathbf{x}_i - \mathbf{x}_{s_1}\|_2 * \max_{k \neq l}(B_{min}(\mathbf{x}_{s_k}, \mathbf{x}_{s_l})) \qquad (1)$$

where B_{min} is an estimation of local minimum bandwidth and $\mathbf{x}_i$ is the coordinate vector of pixel i. The pixels s_k are neighbouring pixels to i. In [10] the neighbourhood is defined to be the vertices of the Delaunay triangle that contains i. Pixels sampled are used by interpolation algorithms to reconstruct an approximation to the ground truth image.

Pixels of high priority score are considered to be statistically significant as they are estimated to be of most potential variance in their values. Sampling pixels of significance is likely to reduce the uncertainty of the reconstruction process. In this way, the system is able to progressively and adaptively acquire pixels that can bring the most potential improvement to the quality of the reconstructed image, resulting in an efficient use of available bandwidth. This work exploits the above concept within the remits of designing a hardware system in order to minimise the cost (time and energy) of the image acquisition process.

III. Proposed progressive sampling based memory interface

A. Scenario setup

Without loss of generality, a scenario is set to add constraints to the discussion, which is shown in Fig.1. Due to its popularity in hardware systems, DRAM is chosen as the example target memory that holds the image to be accessed. The proposed system works within existing DRAM interface, replacing the conventional address generator to issue access locations in an interactive manner. The system progressively identifies and samples pixels of high statistical importance from the external memory, in order to achieve a required image quality level with as few accesses as possible and therefore to reduce the time and energy consumption of the image sampling process. At the end of the sampling process, it is also responsible of filling in missing pixels by reconstructing the image macro block using already sampled pixels. The reconstructed image block is stored in the local buffer from where a client processing unit can access. Compared with conventional DRAM interface, the modified interface only alters the sampling pattern and communicates with the target DRAM in an interactive way.

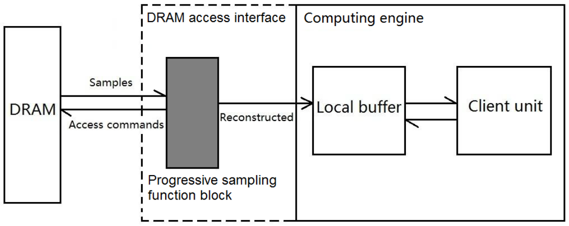

Fig. 1. System working scenario

Different from the conventional memory access method, the proposed system only accesses the target memory during the sampling process and the added interpolation process does not require the memory to be accessed. In the remainder of this paper, the "image sampling process" refers only to the time period when the proposed system accesses the memory, while the "image acquisition process" specifically refers to the whole process of sampling and reconstructing the target image[1].

B. Design of the sampling procedure

A straightforward implementation of image sampling is to uniformly refine the image. Starting from a relatively high sampling rate, during each iteration the sampling process reduces the sampling rate by a factor of 2 and samples all missing pixels belonging to this sampling rate. The process stops when available bandwidth is depleted and a reconstruction of image can be retrieved from the sampled pixels. Although this scheme is straightforward and cheap to implement, it does not have any data adaptability, therefore the pixels sampled are not always statistically significant. This leads to the inability of the system to make efficient trade between bandwidth and image quality.

[1] The bandwidth requirement is therefore determined by the image sampling process, independent of the interpolation process that follows.

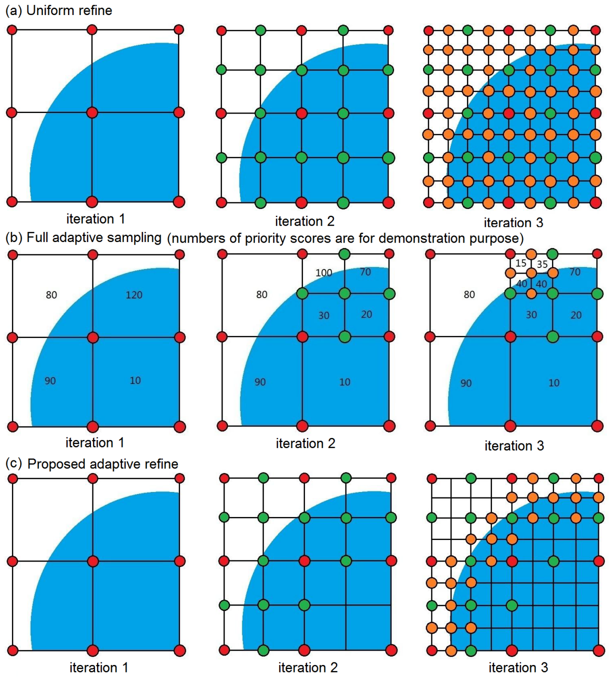

Fig. 2. Progressive sampling methods: uniform sampling (top); full adaptive sampling (middle); proposed adaptive sampling (bottom).

With hardware implementation in mind, this work takes a different approach to uniform sampling and adopts the estimation of priority scores of candidate pixels. The use of variants of priority scores holds key to many progressive point sampling techniques. Most sampling procedures start with a coarse sampling pattern of the target image, and iteratively identify and sample the pixels of most significance to the improvement of the reconstruction quality. Although defined differently, variants of priority scores share a similar base concept. The priority score from Eq.1 is extended in this work to a more general form that describes such concept:

$$f(\mathbf{x}_i) = d_{\mathbf{x}_i,P} * v_i \qquad P: sampled\ pixel\ locations \tag{2}$$

where $\mathbf{x}_i$ is the coordinate vector of a candidate unsampled pixel, and $d_{\mathbf{x}_i,P}$ measures the likelihood of determining pixel $\mathbf{x}_i$ with existing samples. This distance therefore includes, but is not limited to Euclidean distance. The term v_i is the estimated variance of pixel $\mathbf{x}_i$. An instance of progressive sampling using priority scores (denoted as full adaptive sampling in this article) is shown in Fig.2. This adaptive sampling procedure works on regular grid and the priority of pixels is determined by the priority score of its containing image block:

$$f(b_i) = area(b_i) * [\max_i(p(\mathbf{v}_i)) - \min_i(p(\mathbf{v}_i))] \tag{3}$$

where $area(b_i)$ is the area of the block b_i and $p(\mathbf{v}_i)$ is the pixel value of one of the four vertices of b_i at location $\mathbf{v}_i = (x_i, y_i)$. In every iteration, the block of the highest score is refined to a finer resolution and the process keeps running until a user defined quality requirement is met or there is no more pixel to sample from. This sampling procedure is able to provide an optimised sampling pattern at any time so that interpolation using these samples can achieve a high image quality.

However it can be seen that such adaptive sampling is to some extents against the structure of existing hardware systems and DRAMs in concept. While the design of modern hardware systems emphasises the use of data locality to reduce the cost of accessing, full adaptive sampling utilizes the data locality in a different way. It decouples data transmitted in a stream in an attempt to achieve maximum entropy gain with a limited bandwidth. The design of such sampling procedure is based on the assumption that switching sampling location has no significant cost, which is not true for DRAM memory systems. Therefore this work proposes the adaptive refine process which adopts a modified process based on full adaptive sampling but is more suitable for DRAM accessing (Fig.2 (c)).

The proposed adaptive refine follows the same steps as the full adaptive but adaptively refines every block belonging to the current sampling rate, that has a priority score higher than a given threshold, instead of refining only the block that has the highest priority score at the moment. In practice, the threshold can increase gradually as well to adapt to currently available bandwidth. Although this adaptive refine process cannot guarantee a best sampling pattern in between different threshold levels, it still produces the same sampling pattern as full adaptive sampling does when each threshold is met. By allowing flexibility during the sampling process, adaptive refine allows the system to better determine the order of accessing the DRAM.

IV. Structure of the Proposed System

The proposed image sampling system is intended to be used as a system block that replaces conventional address generator within dedicated memory accessing interface. In this work, the system is prototyped on an FPGA device in order to be verified, tested and evaluated.

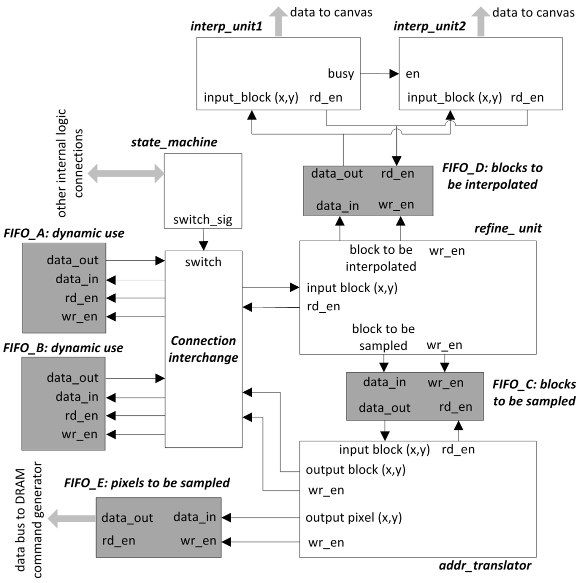

Fig. 3. The structure of proposed system

Fig.3 depicts a high-level view of the system. In general the proposed design generates pixel addresses for the DRAM interface to access a macro block from the original image. The system operates on a local buffer that is prepared for buffering the macro block of the target image. Sampled pixels are filled into this buffer and based on these samples, the system decides where to sample next. During the process, the target macro block is divided into a number of smaller blocks depending on the sampling statistics. Starting from a coarse uniform sampling pattern, the system checks priority scores of existing blocks and refines their resolution accordingly. After all blocks of the current sampling rate are processed, the system moves onto next resolution level. When the sampling process achieves a given quality threshold it stops and the remaining missing pixels in the buffer are filled by Bilinear interpolation.The proposed design characterises each block by its resolution level and anchor, which is the coordinates of its upper left pixel.

The system consists of three major units: $refine\_unit$, $addr\_translator$, and $interp\_unit$. In the first iteration the system starts at an initial high sampling rate. The $refine\_unit$ checks the priority score of existing blocks stored in $FIFO\_A$, and determines if they need further sampling. Blocks with priority score higher than the given threshold are recorded in $FIFO\_C$ and others are recorded in $FIFO\_D$. Blocks stored in $FIFO\_D$ are processed by a set of $interp_unit$s to be interpolated as they are considered to have met the quality requirement.The $addr\_translator$ fetches blocks from $FIFO\_C$ and generates sampling addresses, which are passed to the rest of the interface logic to generate appropriate access commands. Each block refined to a finer sampling level in this way is subsequently divided into four smaller blocks, which are recorded in $FIFO\_B$. When all blocks in $FIFO\_A$ are handled by $refine\_unit$, the system moves onto the next iteration and halves the current sampling rate. During the transition to the next iteration, $FIFO\_A$ and $FIFO\_B$ switch place and the newly generated blocks are checked by $refine\_unit$ in the next iteration.

V. Results

The proposed system was synthesised and placed and routed on a Stratix IV FPGA. Since the system is designed to replace only the conventional address generator, the rest of the DRAM interface and the DRAM response are both simulated with Modelsim testbench instead of being actual implementation. The generated DRAM accessing addresses are passed to DRAM power models designed by Rambus and HP, which in turn report the corresponding DRAM energy consumption of the input access pattern. Various 1Gb DDR3s are simulated by the power model from Rambus[11] as the target DRAM, and the test is also carried out on two smaller sized SDRAM memories modelled by the CACTI tool designed by HP[12]. The DRAMs simulated by Rambus model and CACTI model are of 55 nm and 45 nm technology respectively, while both are of 8 bit I/O and have a burst length of 8.

The synthesised system performs sampling and reconstruction of several benchmark images which are: lena, barbara, and boat. All of them are of size 527x527 and transformed to grayscale image with each pixel represented by a 8-bit value. The system works on non-overlapping macro blocks of size 17x17 on the target image. For each test, the system aims to achieve a target priority score threshold ranging from 150 (best quality) to 1800 (worst quality).

In the following sections, the proposed adaptive refine method is firstly evaluated against various reference sampling methods. Then a detailed evaluation of the proposed system is given together with an evaluation of the impact of the proposed system to an image compression application. Finally the mapping from FPGA prototype to ASIC system is discussed.

A. Evaluation of the proposed adaptive refine procedure

The proposed adaptive refine is first evaluated against three reference methods for its ability to trade PSNR for reduced bandwidth requirement and DRAM access energy. The references are as described in section III: conventional accessing pattern that reads every pixel from DRAM, uniform refine, and full adaptive sampling on a regular grid. This test aims to analyse the upper limit of the performance of the proposed system, temporarily ignoring the time and energy cost overhead introduced by the proposed system itself. For this test both linear and block mapping strategies are used. For linear mapping each row of the image is stored in a single page within the DRAM, whereas for block mapping each macro block is stored in a single page such that the row switching activities are reduced while reading a macro block. Fig.4(a) shows the target image that needs to be acquired, where Fig.4(b) shows an instance of the reconstructed image from the proposed adaptive refine sampling method where the threshold was set to 600. Fig.4(c) shows the achieved PSNR of the reconstructed image in the local buffer as a function of the percentage of the pixels sampled. It shows that the proposed adaptive refine, as original full adaptive sampling does, has a significant improvement in image PSNR vs. number of pixels sampled and it fills the large gaps between data points from uniform refine method. The graph in Fig.4(d)(e) are the corresponding DRAM access energy and time at each achieved PSNR level. It shows that the introduced cost overhead (DRAM access

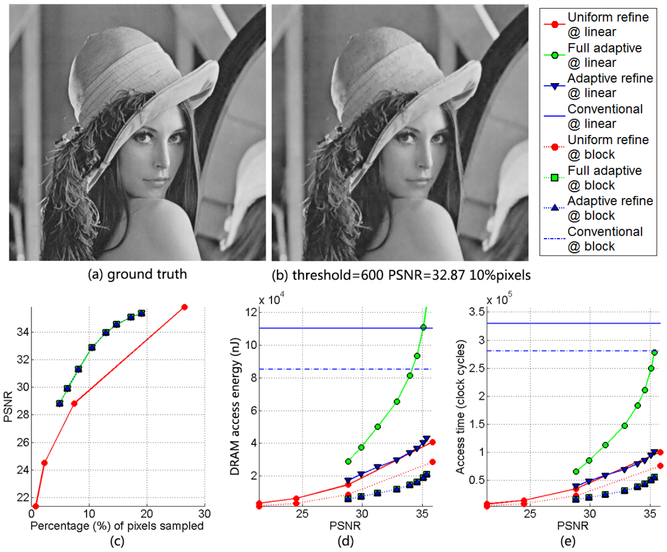

Fig. 4. (a)(b) Comparison between ground truth image and the reconstruction using pixels sampled at a threshold of 600. (c) Reconstruction PSNR vs. percentage of pixels sampled; (d) DRAM access energy vs. reconstruction PSNR; (e) DRAM access time vs. reconstruction PSNR; From left to right, the data points of adaptive refine and full adaptive sampling algorithms in these graphs are results from threshold of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively; the data points of uniform refine algorithm are results from sampling rate of 16, 8, 4, and 2 respectively.

energy and time) from progressive sampling methods is more obvious on linear mapped memory content, but the proposed adaptive refine allows the system to more flexibly organise the accessing pattern, resulting in a much lowered cost overhead than full adaptive sampling. In the case of block mapped memory content such overhead is minimized, and therefore a greater reduction of access energy and access time can be seen. Nevertheless, for both mapping strategies a reduction in DRAM occupation time and access energy can be seen with PSNR up to 34 dB.

Apart from the cost overhead of DRAM access, the implementation and execution of the proposed system also inevitably introduces overhead. Any bandwidth or energy saved from the DRAM side has to be compared with the cost of implementing the method. This is discussed in the following subsection based on the evaluation data of the implemented system.

B. Evaluation of the proposed system on FPGA

A detailed evaluation of the system is discussed in this section taking into account the overheads imposed by the proposed system. The proposed system is compared against conventional address generator in DRAM access interface. Block mapped image content is used in the following test as it is the most popular storing strategy used in hardware systems.

1) Hardware resource usage: The conventional address generator in DRAM interface often acts as a simple counter and therefore the hardware resource required to implement it is omitted in this evaluation. Table.I reports the added cost of hardware resources for implementing the proposed system.

TABLE I. HARDWARE RESOURCE USAGE OF THE PROPOSED SYSTEM

	Combinational ALUTs	Logic registers	Block RAM bits	DSP block 18-bit elements
refine_unit	156	109	0	0
interp_unit (x2)	600	412	0	16
addr_translator	104	82	0	0
FIFOs	662	428	2160	0
total	1522	1031	2160	16

2) Acquisition time: The reported max frequencies of the design at "slow" and "fast" corner case[2] are 200 MHz and 357 MHz respectively. The DRAM access time (in clock cycles) as well as the total image acquisition time spent including interpolation are reported in Fig.5. The reference lines show the image acquisition time required by conventional image accessing process, transformed to equivalent clock cycles at different clock frequencies (X times of the system frequency).

It can be seen that the achievable PSNR differs with test subject. The image "barbara" has more complex local structures than the other two images and therefore the PSNR of its reconstruction is lower, even when a same threshold is met. In general, due to the reduced number of sampled pixels, the proposed system has a much lower DRAM occupation time (sampling process) than that of the conventional access method even if the DRAM data bus runs at 2x the frequency of the proposed system. This results in a much reduced bandwidth requirement of DRAM and when needed, it frees the DRAM early on to be accessed by other processing units in a large system. However a significant amount of time is spent on interpolating the image. Nevertheless, the total image acquisition time is reduced in most cases in this test if the system runs at the same clock frequency as the DRAM data bus. In this particular test two $interp_unit$s are used, but more of this module can be added to accelerate the process in the expense of more hardware resources.

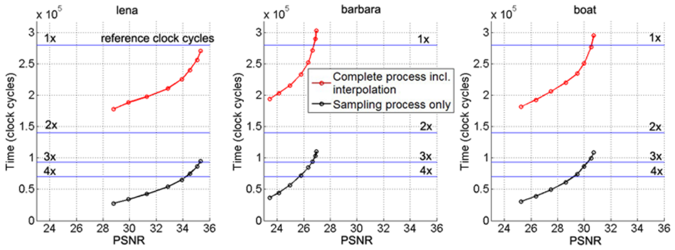

Fig. 5. Time requirement for sampling process, and complete acquisition process including interpolation. The X axis shows the achieved PSNR given different levels of thr. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively.

3) Energy consumption: To evaluate the energy consumption, the core dynamic energy consumed by the proposed system is analysed by Quartus PowerPlay analyser, as the cost of implementing the sampling procedure on top of the conventional address generator in DRAM access interface. Fig.6(a) shows the breakdown of energy consumption of the sampling and interpolation process spent purely by the proposed system. The total energy consumption that includes DRAM access energy is reported in Fig.6(b), normalized by the DRAM energy consumption of conventional accessing method.

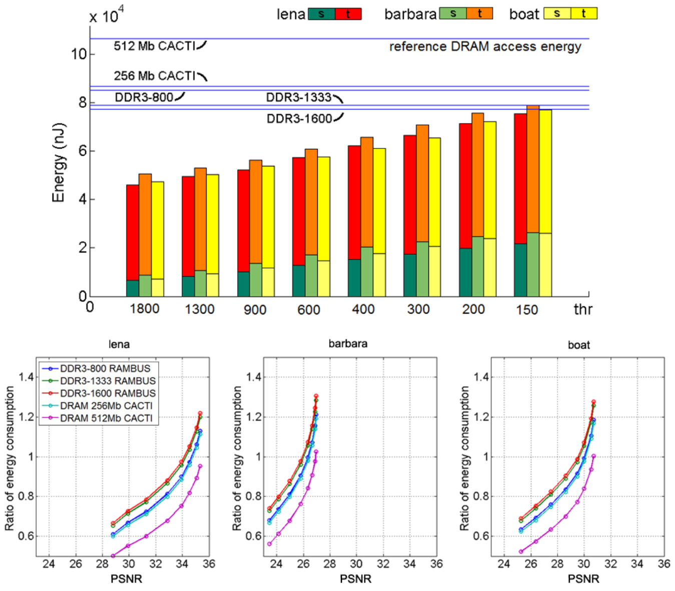

Fig. 6. (a) Breakdown of energy consumption by the proposed system, for sampling (s) process, and complete (t) process including interpolation. (b) The ratio of total energy consumption of the proposed system (including corresponding energy spent on sampling from DRAM) to that of the memory access by conventional access method. Different DRAM models are used as target memory. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively.

[2] Quartus default settings for Stratix IV.

The obtained results show that the energy consumed by the address generating process is much lower than that of the interpolation process. A reduction of total energy consumption (including DRAM access energy) can be seen when the threshold is above about 400 if DDR3s are targeted. For general purpose SDRAMs simulated by CACTI, a reduction of energy consumption can be seen across most threshold levels. In the case of "lena", a reduction of up to 45% can be seen while maintaining a PSNR above 30 dB.

C. *Case study on JPEG2000*

The proposed system is evaluated under a real application scenario in order to assess its impact under a real-life problem. The selected application is the JPEG2000 standard and it was chosen due to its widely usage. The compression unit access image blocks read either in the conventional access pattern, or by the proposed system. The image quality of the compression output using both image acquisition methods are compared with each other. Fig.7 shows the differences of MSE between compression output using ground truth image and that using the image acquired by the proposed system. The black curve is the MSE difference when no compression is used and it is in fact the same as in Fig.6 but presented in MSE. When the acquired image is processed by the compression unit, it can be seen that the image quality difference keeps decreasing as the compression rate increases. This shows that some loss of image quality due to progressive sampling is absorbed by the process of compression. In general, if the client tends to remove image redundancy as the proposed system does then the impact of quality loss due to applying the proposed system is reduced, and therefore the system can achieve an even larger gain in bandwidth and image acquisition time/energy.

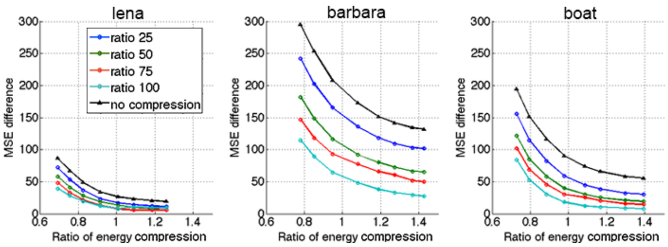

Fig. 7. The quality difference of compressed image, using both conventional accessing method and the proposed system. DDR3-800 is used as target memory. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively.

D. *Targeting an ASIC implemenetation*

The evaluation of the proposed system on FPGA suggests its potential in saving both bandwidth and time/energy of the image acquisition process. It is also desirable to be adopted in existing ASIC memory interfaces as a subsystem that provides an alternative image accessing methods when bandwidth or energy is of a major concern. According to [13], the ASIC implementation of a same system has an average of 4.6x decrease in path delay, and an average of 14x decrease in dynamic power consumption running the same test vector. A projection of the proposed FPGA implementation to ASIC by these factors sees the proposed system able to reduce both image acquisition time and energy. It will catch up with DDR3-800 but for faster models it still requires more $interp\_unit$ to accelerate the interpolation process, in order to be capable of reducing total image acquisition time. On the other hand, the energy consumption of the system will be reduced greatly, making the system promising in saving energy by large margin.

VI. Conclusion

In this work a hardware oriented image progressive acquisition procedure is proposed as well as a hardware design of the proposed system. The proposed system is able to act as an alternative address generator in conventional memory access interface, to provide the ability to efficiently trade image quality for lowered image acquisition time and/or energy. The evaluation of the implemented system shows a reduction of up to 85% of memory accessing time and 33%/45% of image acquisition time/energy is achieved on benchmark image "lena" while maintaining a PSNR of about 30 dB. When an image compression application is targeted, some of the information loss injected by the proposed system can be hidden by the application itself, leading to even larger gains in image acquisition time and energy consumption.

References

[1] C. Carvalho, "The gap between processor and memory speeds," in *Proc. of IEEE International Conference on Control and Automation*, 2002.

[2] H. Kim and I.-C. Park, "High-performance and low-power memory-interface architecture for video processing applications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 11, pp. 1160–1170, 2001.

[3] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach.* Elsevier, 2012.

[4] Y. Li and T. Zhang, "Reducing dram image data access energy consumption in video processing," *Multimedia, IEEE Transactions on*, vol. 14, no. 2, pp. 303–313, 2012.

[5] D. Zhou, J. Zhou, X. He, J. Zhu, J. Kong, P. Liu, and S. Goto, "A 530 mpixels/s 4096x2160@ 60fps h. 264/avc high profile video decoder chip," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 4, pp. 777–788, 2011.

[6] Z. Devir and M. Lindenbaum, "Adaptive range sampling using a stochastic model," *Journal of computing and information science in engineering*, vol. 7, no. 1, pp. 20–25, 2007.

[7] A. Adamson, M. Alexa, and A. Nealen, "Adaptive sampling of intersectable models exploiting image and object-space coherence," in *Proceedings of the 2005 symposium on Interactive 3D graphics and games.* ACM, 2005, pp. 171–178.

[8] T. Y. Lee, "A new frame-recompression algorithm and its hardware design for mpeg-2 video decoders," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 6, pp. 529–534, 2003.

[9] T. Yng, B.-G. Lee, and H. Yoo, "A low complexity and lossless frame memory compression for display devices," *Consumer Electronics, IEEE Transactions on*, vol. 54, no. 3, pp. 1453–1458, 2008.

[10] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Zeevi, "The farthest point strategy for progressive image sampling," *Image Processing, IEEE Transactions on*, vol. 6, no. 9, pp. 1305–1315, 1997.

[11] T. Vogelsang, "Understanding the energy consumption of dynamic random access memories," in *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture.* IEEE Computer Society, 2010, pp. 363–374.

[12] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "Cacti: An integrated cache and memory access time, cycle time, area, leakage, and dynamic power model," Technical Report HPL-2008-20, HP Laboratories, Tech. Rep., 2008.

[13] I. Kuon and J. Rose, "Measuring the gap between fpgas and asics," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 2, pp. 203–215, 2007.