

A Hardware Architecture for Better Portable Graphics (BPG) Compression Encoder

Umar Albalawi
Computer Science and Engineering
University of North Texas, USA.
Email: UmarAlbalawi@my.unt.edu

Saraju P. Mohanty
Computer Science and Engineering
University of North Texas, USA.
Email: saraju.mohanty@unt.edu

Elias Kougianos
Engineering Technology
University of North Texas, USA.
Email: elias.kougianos@unt.edu

Abstract—This paper proposes a hardware architecture for the newly introduced Better Portable Graphics (BPG) compression algorithm. Since its introduction in 1987, the Joint Photographic Experts Group (JPEG) graphics format has been the *de facto* choice for image compression. However, the new compression technique BPG outperforms JPEG in terms of compression quality and size of the compressed file. The objective of this paper is to present a hardware architecture for enhanced real time compression of the image. The complexity of the BPG encoder library is reduced by using hardware compression wherever possible over software compression because of the real time requirements, possibly in embedded systems with low latency requirements. BPG compression is based on the High Efficiency Video Coding (HEVC), which is considered a major advance in compression techniques. In this paper, only image compression is considered. The proposed architecture is prototyped in MATLAB[®]/Simulink[®]. The experimental results prove that the visual quality of BPG compression is higher than that of JPEG with equal or reduced file size. To the best of the authors' knowledge, this is the first ever proposed hardware architecture for BPG compression.

Keywords—Image Compression, VLSI Hardware Architecture, Better Portable Graphics (BPG), JPEG

I. INTRODUCTION AND MOTIVATION

As multimedia usage continues to expand with an enormous number of applications, the demand for high quality images with acceptable size has been dramatically increased. BPG [1] is a novel step in the field of image compression that aims to supersede the decades-old *de facto* JPEG format [2] with its distinct attributes, meeting modern display requirements (high quality and lower size) of developers, programmers and graphic businesses. HEVC (High Efficiency Video Coding) [3] and compatibility considerations are accommodated in the form of a small JavaScript (56 KB) decoder which is one of the key composing elements of the new format. Unlike JPEG, BPG does not require supplementary browser plug-ins to display the compressed image. Other attributes that differentiate BPG from JPEG and make it an excellent choice include the following:

- The open source and royalty-free and patent-free nature of the BPG justifies it as a more appropriate choice for users because they do not need to be concerned with legal issues.
- BPG is close in spirit to JPEG and can offer lossless compression.

- With advanced quality features, BPG offers different chroma formats making it compatible with multiple video encoding schemes such as analog, digital, and JPEG encoding schemes.
- Different chroma formats are supported including grayscale, RGB, YCgCo, YCbCr, Non-premultiplied alpha, and Premultiplied alpha.
- BPG uses a range of metadata for efficient conversion including EXIF, ICC profile, and XMP.

From the above discussion, it is evident that BPG compression is an obvious choice to meet modern technology requirements: high quality and lower size. The organization of the paper is as follows: Section II describes related studies and the novel contributions of this paper; Section III illustrates the BPG encoder compression algorithm. In Section IV, the proposed hardware architecture for the BPG encoder is discussed. Section V illustrates the functionality of the proposed BPG encoder architecture with Simulink[®] based simulations. Conclusions are presented in Section VI.

II. RELATED EXISTING RESEARCH AND NOVEL CONTRIBUTION OF THIS PAPER

A. Related Research

The JPEG standard's successor, JPEG-2000, intends to overcome several of the existing shortcomings such as better compression ratios, compression scalability, and resolution accuracy. Ghodhbani *et al.* [4] suggested that hardware implemented JPEG-2000 encoding is more efficient and optimized than current software implementations and demonstrated an optimized EBCOT algorithm architecture implemented on an FPGA platform. Improved operational efficiency was observed for a pipelined BPC encoder implemented in the VHDL Hardware Description Language (HDL).

Liu *et al.* [5] particularly studied the HEVC which implements compression methods based on 64×64 blocks and minimum recursive block partitions of 4×4 . Prediction modes and tree-structures improve HEVC coding efficiency. A fully pipelined parallel HEVC implementation with negligible Peak Signal-to-Noise-Ratio (PSNR) was demonstrated that allows real-time encoding, such as 1080p at 30fp with minimal hardware at 600 MHz.

The Ultra High Definition Television (UHDTV) format is expected to support 3840×2160 and 7680×4320 resolutions at

120 fps. This implies a data throughput 100 times higher than current 1080p HDTV. Zhou *et al.* [6] proposed optimizations such as pre-normalization, hybrid path coverage, binarization components, context modeling and lookahead rLPS to reduce the path delay of the BAE. These optimizations are possible by exploiting the incompleteness of data dependencies in rLPS updating, which yields a Context-Adaptive Binary Arithmetic Coding (CABAC) encoder at 4.37 bins/s i.e. a 45.3% optimization costing 4.8% BPCC performance degradation and 62.5% better performance than current architectures.

Optimized VLSI architecture techniques allow high performance SAO encoding in HEVC. Mody *et al.* [7] demonstrated 4 K resolution at 60 fps at 200 MHz using 0.15 mm² of the silicon area in a 28 nm CMOS process with artifact avoidance algorithms, which provide 4.3% savings in SAO encoding.

B. Novel Contribution of this Paper

A schematic overview of the proposed BPG compression encoder is shown in Figure 1. As the initial step, an input file is read. Then, the details of the image are extracted. In this paper, only an image (not video) encoder is considered and the modified flow is shown in Fig. 2. The main objective of this paper is to describe a hardware architecture of the BPG compression encoder. To the best of the authors' knowledge, this is the first ever proposed hardware architecture of BPG compression encoder. In this paper the complexity of the BPG encoder library is reduced by using hardware compression on a subset of the complete BPG specification. The novel contributions of this research include the following:

- The first-ever architecture for hardware BPG compression.
- A Simulink[®]-based prototype of the algorithm implementation.
- Experimental analysis and comparison of the proposed architecture versus JPEG.

The advantages of a hardware versus software implementation include the following:

- Real-time image encoding with minimal hardware.
- Significant reduction in power usage as opposed to a general-purpose processor.
- Dedicated circuitry that does not slow down the host.
- Hardware is less susceptible to malicious software such as viruses, trojans etc.

III. THE NEW IMAGE COMPRESSION ALGORITHM BPG

BPG is a new image format offering several advantages over the JPEG format. It achieves higher compression ratio with smaller size than JPEG for similar quality. In the BPG format, lossless compression, animation, various color spaces (grayscale, YCbCr, RGB, YCqCo), and chroma formats are supported [1]. The reference BPG image library and utilities (libbpg) can be divided into four functions: BPG encoder, BPG decoder, Javascript decoder, and BPG decoding. The BPG encoder takes JPEG or PNG images as input, performs BPG compression and provides the corresponding BPG image. The

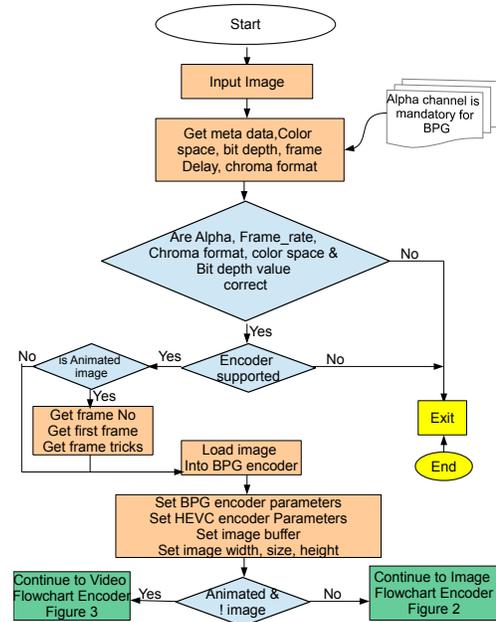


Fig. 1: BPG Encoder Algorithm.

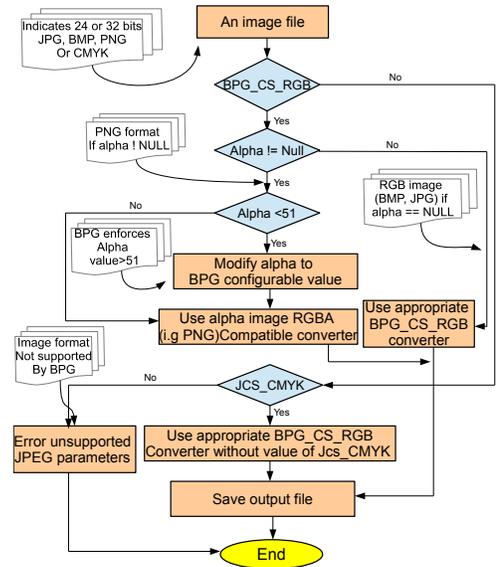


Fig. 2: BPG Image Encoder Algorithm.

BPG decoder does the reverse function. With a small Javascript decoder, the BPG format is supported by most web browsers. The BPG decoding allows any BPG image to be decoded in any program. In the proposed architecture, the focus is in the BPG image encoder compression. The BPG encoder is based on HEVC encoding [3]. HEVC is considered the prime candidate to replace H.264 encoders due to its compression efficiency [8]. The HEVC project aims at reducing the bitrate compared to H.264/AVC because it is more parallel-friendly [9][10]. Figure 1 shows the initial steps of the BPG encoder algorithm. It can be seen from the figure that at some point the encoder must check whether an input is a video (dynamic image) or static image. If the input is video the algorithm

proceeds to the video encoder, shown in Fig. 3; otherwise, the algorithm continues to the image encoder, illustrated in Fig. 2.

After reading the image, the encoder does initialization processes to read meta data, color space, bit depth, etc. There is an essential step in which the algorithm must check two conditions: bit depth and color space. Bit (color) depth refers to the amount of data that can be used to indicate the color of each pixel [11]. It can be represented by different numbers: 8, 10, 12, \dots . It describes the number of bits used to represent colors per pixel. The concern with images that have high bit depths are data storage, and required transmission bandwidth. Also, some displays are not capable of reproducing all of these colors. Undoubtedly, there must be a trade off between quality and bit depth. The BPG compression encoder strictly considers images with bit depth of 8.

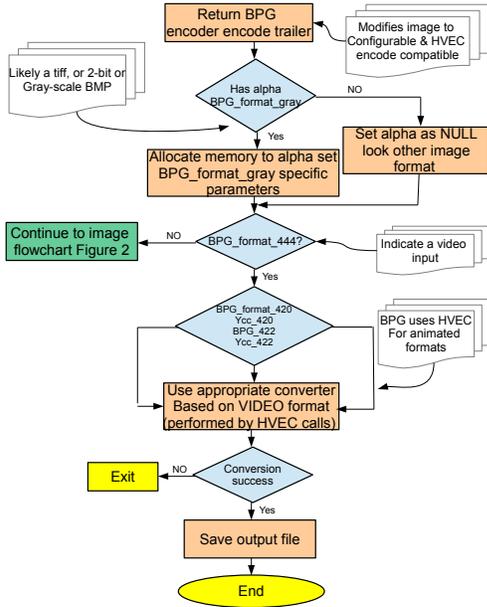


Fig. 3: BPG Video Encoder Algorithm.

IV. PROPOSED HARDWARE ARCHITECTURE FOR THE BPG ENCODER COMPRESSION ALGORITHM

The hardware architecture of BPG encoder compression is presented in this section. From the discussion in Section III, BPG compression encoding can be divided into two phases: the pre-encoding (initialization) phase and HEVC encoding, which are shown in Fig. 4. In general, compression can be classified into two categories, lossless and lossy. When the exact original data is recovered, this is called lossless compression, while in the lossy case, a close approximation of the original data is obtained. BPG is capable of both lossy and lossless compression.

A. Initialization Phase

Images can have different pixel depth, color spaces, and alpha channel. There are initialization procedures that have to be completed before doing the compression encoding. The first procedure is to obtain the image details: meta data, color space, pixel depth, and alpha. The BPG compression encoder

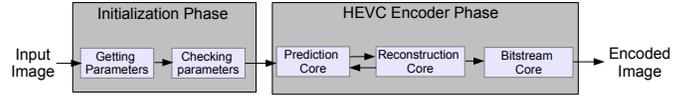


Fig. 4: BPG Encoder Block Diagram.

algorithm requires images with bit depth of 8 and true color or grayscale color spaces. These are essential requirements, otherwise the encoder provides an error message indicating that bit depth or color space are not supported. Algorithm 1 illustrates the steps of the initialization phase.

Algorithm 1 Initialization phase algorithm

```

1:  $Parameters \leftarrow \{PixelDpth, ColorSpace, AlphaChannel\}$ 
2:  $Resolution \leftarrow \{pixels/inch\}$ 
3:  $Bitdepth \leftarrow \{MetaData/ImageSize\}$ 
4: while  $Lenght > 2$  do
5:   if  $Bitdepth = 8$  then
6:      $AlphaChannel \leftarrow \emptyset$ 
7:     PRINT "ERROR: BitDepth is not supported"
8:   if  $MetaDatacolor < 1$  then
9:     PRINT "ERROR: ColorSpace is not supported"
10: PRINT "Bit Depth is 8 and correct color type"
11: PRINT "Image accepted for BPG compression"
12: end

```

B. HEVC Encoder Phase

BPG encoding is based on the HEVC encoder, which is considered a major advance in compression techniques. HEVC offers high coding efficiency because of the intelligent approach that is used to reduce the area (pixels) that is encoded [12]. HEVC uses an 8×8 block as the basic coding unit, and the Discrete Cosine Transform (DCT) or the Discrete Sine Transform (DST) as the transformation mechanism to the frequency domain. In HEVC, the amount of information content (entropy) is considered context-adaptive binary arithmetic coding (CABAC) only. The HEVC encoder encodes the pictures into a bitstream, which contains a sequence of data known as a Network Abstraction Layer (NAL). The encoder stores pictures in the Decoder Picture Buffer (DPB) as illustrated in Fig. 5. A picture in HEVC is divided into one or multiple slices, which contain one or multiple slice segments.

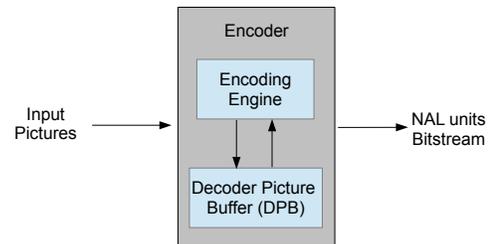


Fig. 5: HEVC Encoding Block Diagram.

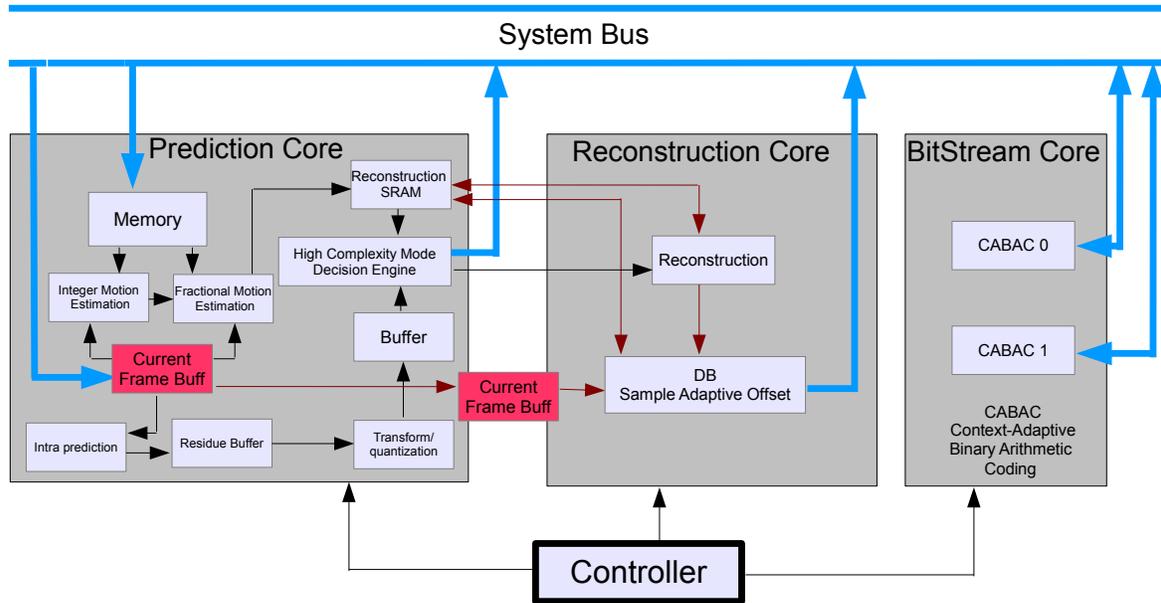


Fig. 6: HEVC Encoder System.

HEVC encoding is performed in three stages: prediction, reconstruction, and bitstream core. The prediction core is the essential stage because it handles intra and inter prediction in parallel, where the reconstruction code constructs reference frames at each time of the encoded frame [12]. The bitstream core performs CABAC. The three core stages of the HEVC encoder are the following:

- **Inter Prediction:** in this block the essential task is to reduce the temporal redundancy by comparing a current prediction unit with neighboring prediction units, which can be done by motion estimation.
- **Intra Prediction:** to reduce spatial redundancy.
- **Transform and Quantization:** transform is the next step, which is performed after reducing the temporal and spatial redundancy. The size of transform can be 4×4 , 8×8 , 16×16 , or 32×32 , and DCT is used. After the process of transform, the sample is quantized and transformed to entropy coding.
- **Entropy Coding:** the main objective is to eliminate redundancy which has not been removed by the prediction stage.

The proposed hardware architecture of the HEVC encoder is shown in Fig. 6.

V. PROPOSED BPG ENCODER ARCHITECTURE AND SIMULINK[®] BASED SIMULATIONS

The system-level architecture of the proposed BPG compression encoder is illustrated in Figure 7. The blocks shown in the dotted lines explain the initialization phase and HEVC encoder. The initialization phase preprocesses the input image and obtains image details: bit depth, alpha, chroma format, and a code for color space. Postprocessing verifies bit depth and color space. The HEVC encoder receives the verified image

and then starts performing the splitting, intra frame prediction, DCT, IDCT, and quantization processes.

A. MATLAB[®]/Simulink[®] Based Modeling

The proposed algorithm is prototyped in MATLAB[®]/Simulink[®] Version 8.3 (R2014a), with the Computer Vision System Toolbox Version 9.7 [13]. The HEVC encoder model is shown in Fig. 8. The methodology that is used to represent the high level system modeling is bottom-up. The first step is focused on building function units; the next step is to integrate these units into sub-systems; and finally, verifying and testing overall system functionality. MATLAB[®]/Simulink[®] offers image processing functions and modules that facilitate fast prototyping. Another advantage of using MATLAB[®]/Simulink[®] is the availability of function units such as DCT/IDCT and block processing. In addition, the system-level modeling can be accomplished using different modules: Color Conversion and DCT domain compression.

B. Experimental Results

Four standard test images were selected: Bear, IceClimb, Lena, and Wallpaper, with different spatial and frequency characteristics. The test images are encoded using the proposed BPG compression encoder. Describing the type and amount of degradation in reconstructed compressed images is considered a major concern in evaluating picture quality in image compression systems. It has been proven [14] that some measures of image quality correlate well for a given compression algorithm but they are not reliable for an evaluation across different algorithms. Thus, the most common measures of image quality were used in this work: Root Mean Squared Error (RMSE) [15] given in Eqn. 1 and Peak Signal to Noise Ratio (PSNR) [16]

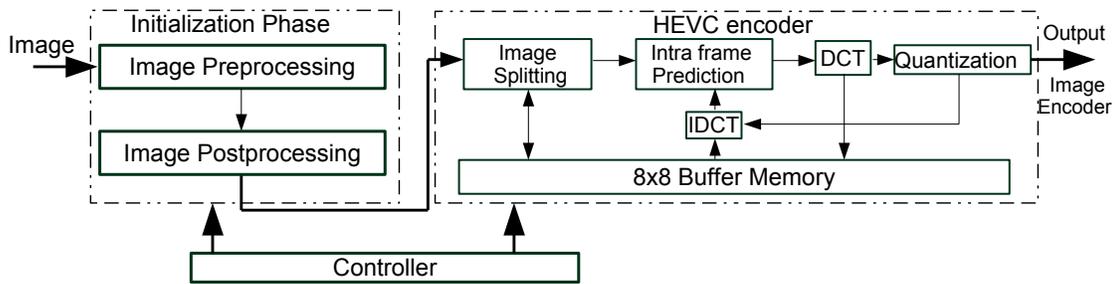


Fig. 7: System Level Architecture of the Proposed Algorithm.

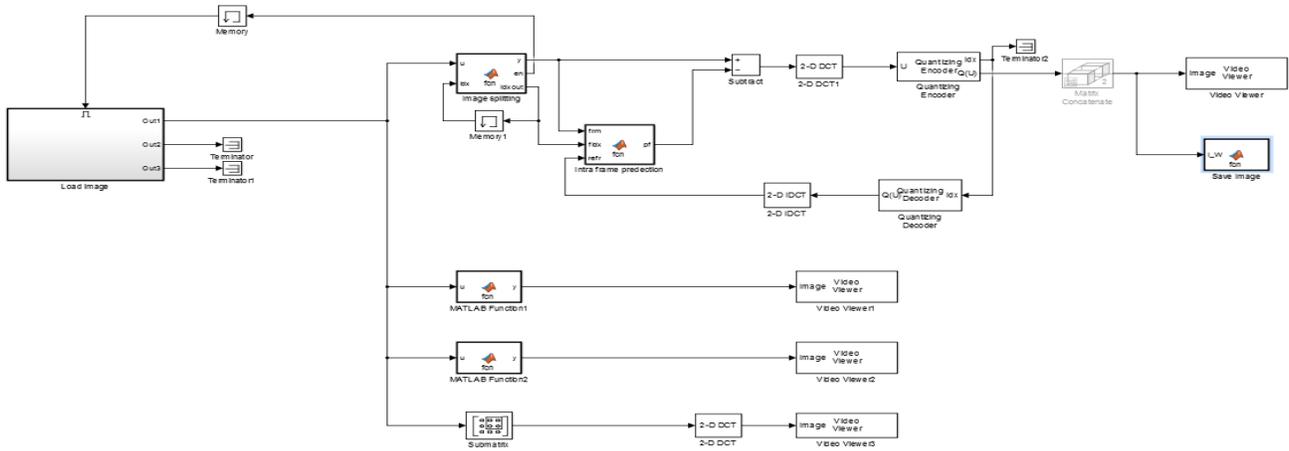


Fig. 8: BPG Compression Encoder in Simulink®.

given in Eqn. 2:

$$RMSE = \frac{1}{\sqrt{mn}} \sum_{j=1}^{m-1} \sum_{n=1}^N \|(O(i, j) - O'(i, j))\|^2 \quad (1)$$

$$PSNR = 10 \log \left(\frac{(2^n - 1)^2}{MSE} \right) = 10 \log \left(\frac{255^2}{MSE} \right) \quad (2)$$

The test images and the corresponding BPG images format are shown in Fig. 9, Fig. 10, Fig. 11, and Fig. 12. Table I illustrates the related metrics for each compression technique and test image. It is observed that for essentially the same PSNR, the size of the BPG image is substantially reduced.

VI. CONCLUSIONS AND FUTURE DIRECTIONS OF RESEARCH

In this paper, a hardware architecture to perform BPG compression encoder in images is presented. The encoding scheme can be divided into two phases. First is the initialization phase, which reads an image and extracts its details then verifies specific parameters such as bit depth, alpha, and color space. The second phase is HEVC encoding, which is considered a major advance in compression techniques. The proposed architecture is prototyped in Simulink®. The

experimental results are compared with existing JPEG techniques in terms of quality and size and indicate the superior compression characteristics of BPG. Further work could include proposed hardware architecture as prototype using a hardware description language like Verilog and also making hardware in actual silicon. Also, since this paper only considers image compression, further work can be done on BPG video compression; the algorithm is clarified in Section III. The BPG architecture will be soon be integrated with encryption and or digital watermarking capabilities [13], [17]. Future research directions also include developing energy-efficient as well as high-performance architectures which can be used in image or video communications in Internet of Things (IoT) frameworks.

REFERENCES

- [1] F. Bellard, "The BPG Image Format," <http://bellard.org/bpg/>, last Accessed on 09/20/2015.
- [2] G. K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii-xxxiv, Feb 1992.
- [3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1649 – 1668, 2012.
- [4] K. Sarawadekar and S. Banerjee, "Area Efficient, High-speed VLSI Design for Ebcot Block Coder in JPEG 2000," in *Proceedings of*



(a) JPEG Image.



(b) BPG Image

Fig. 9: Compression of Bear Image (256×256).



(a) JPEG Image.



(b) BPG Image

Fig. 10: Compression of IceClimb Image (512×512).



(a) JPEG Image.



(b) BPG Image

Fig. 11: Compression of Lena Image (512×512).



(a) JPEG Image.



(b) BPG Image

Fig. 12: Compression of Wallpaper Image (128×128).

TABLE I: Quality Metrics for the Compression Technique and Test Image

Test Image	Code	Size (KB)	RMSE	PSNR
Bear Image	JPEG (input image)	19.4	0.015	84.02
	BPG image	15.8	0.012	84.82
IceClimb Image	JPEG (input image)	85.3	0.012	86.035
	BPG image	78.4	0.010	86.11
Lena Image	JPEG (input image)	29.3	0.023	80.6
	BPG image	26.4	0.20	80.75
Wallpaper Image	JPEG (input image)	6.2	0.022	81.1
	BPG image	4.41	0.19	81.22

International Conference on Electronics and Information Engineering (ICEIE), vol. 2, 2010, pp. V2–110 – V2–113.

- [5] C. Liu, W. Shen, T. Ma, Y. Fan, and X. Zeng, “A Highly Pipelined VLSI Architecture for All Modes and Block Sizes Intra Prediction in HEVC Encoder,” in *Proceedings of IEEE 10th International Conference on ASIC (ASICON)*, 2013.
- [6] D. Zhou, J. Zhou, W. Fei, and G. S., “Ultra-High-Throughput VLSI Architecture of H.265/HEVC CABAC Encoder for UHD TV Applications,” in *Proceedings of IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, 2014, pp. 497 – 507.
- [7] M. Mody, H. Garud, S. Nagori, and D. K. Mandal, “High Throughput VLSI Architecture for HEVC SAO Encoding for Ultra HDTV,” in *Proceedings of IEEE International Symposium on Circuits and Systems*, 2014.
- [8] M. U. K. Khan, M. Shafique, and J. Henkel, “Software Architecture of High Efficiency Video Coding for Many-Core System with Power-Efficient Workload Balancing,” in *Proceedings Automation and Test Design in Europe Conference and Exhibition (DATE)*, 2014, pp. 1–6.
- [9] C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, “Parallel Scalability and Efficiency of HEVC Parallelization Approaches,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1827 – 1838, 2012.
- [10] M. Shafique, M. U. K. Khan, and J. Henkel, “Power Efficient and Workload Balanced Tiling For Parallelized High Efficiency Video Coding,” in *Proceedings IEEE International Conference on Image Processing*, 2014, pp. 1253 – 1257.
- [11] S. P. Mohanty, E. Kougianos, W. Cai, and M. Ratnani, “VLSI Architectures of Perceptual Based Video Watermarking for Real-time Copyright Protection,” in *Proceedings of the 10th International Symposium on Quality Electronic Design (ISQED)*, 2009, pp. 527 – 534.
- [12] V. Sze, M. Budagavi, and G. J. Sullivan, Eds., *High Efficiency Video Coding (HEVC)*. Springer International Publishing, 2014.
- [13] S. P. Mohanty and E. Kougianos, “Real-Time Perceptual Watermarking Architectures For Video Broadcasting,” *Elsevier Journal of Systems and Software (JSS)*, vol. 19, no. 12, pp. 724 – 738, 2011.
- [14] M. Mrak, S. Grgic, and M. Grgic, “Picture Quality Measures in Image Compression System,” in *Proceedings The IEEE Region 8 Computer as a Tool EUROCON*, 2003, pp. 233–236.
- [15] C. J. Willmott and K. Matsuura, “Advantages of the Mean Absolute Error (MAE) over the Root Mean Square error (RMSE) in assessing average model performance,” in *Proceedings Climate Research*, vol. 30, 2005, p. 79–82.
- [16] Q. Huynh-Thu and M. Ghanbari, “Scope of Validity of PSNR in Image/Video Quality Assessment,” in *Electronics Letters*, vol. 44, 2008, pp. 800 – 801.
- [17] N. M. Kosaraju, M. Varanasi, and S. P. Mohanty, “A High-Performance VLSI Architecture For Advanced Encryption Standard (AES) Algorithm,” in *Proceedings of the 19th International Conference on VLSI Design*, 2006, pp. 481–484.