DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-276

FPGA Implementation of Gamma Correction using a Piecewise Linear Approach for a Small Size Endoscopic Camera

Sheikh Shanawaz Mostafa[a], L. Natércia Sousa[a], Nuno Fábio Ferreira[a,b], Ricardo M. Sousa[c], Joao Santos[c], F. Morgado-Dias[a,b], Martin Wäny[c]

[a]Madeira Interactive Technologies Institute, Madeira Tecnopolo 9020-105, Funchal, Portugal.

[b]University of Madeira, Rua dos Ferreiros 9000-082, Funchal, Portugal.

[c]Awaiba Lda, Madeira Tecnopolo 9020-105, Funchal, Portugal.

Abstract

A gamma function is essential for adjusting the response of any display device. In the case of endoscopy, it is even more important because for endoscopy it is not only the satisfaction of the user but the diagnosis of patient's problems that could lead to life and death decision sometimes. In this paper, a technique of approximating the gamma function is applied using a piecewise linear method. It has 10 bits input and output pixels of color (RBG) channels. VHDL is used to describe the function and implemented in a Spartan 6 FPGA to achieve high computation and parallel processing. It was tested on a small endoscopy camera called NanEye. The system has 31 reconfigurable gamma function values from 1 to 4 with 0.1 intervals. It has a small footprint in terms of memory and no specialized DSP processor. The average mean absolute error of the implemented solution is 2.1747. The system can process up to 750 million pixel components per second in a Spartan 6 FPGA.

Introduction

Nowadays endoscopic cameras are used for detecting many diseases including the rectal and colon cancers. In 2009 only in USA 18,328,000 endoscopies were performed which have an estimated cost of $32.4 billion [1]. The advantage over other imagining technique like x-ray is that it gives more visual information like color, texture, etc. However, to show the video or picture from the endoscopy it has to be gamma corrected. Although the modern display systems have almost linear characteristic unlike the old CRT monitors. Still the gamma correction is done due to coding [2], [3] and matching the nonlinearity of the human vision system described by Weber–Fechner law [4]. Real time gamma correction closes the gap between the human vision and the digital display so that the physician can take advantage of the technology.

Eq. 1 represents the gamma function and the output pixels of the system (Output pixel value) have to be the inverse gamma power of the input pixels (Input pixel value).

$$\text{Output pixel value} = (\text{Input pixel value})^{(1/\text{gamma})} \tag{1}$$

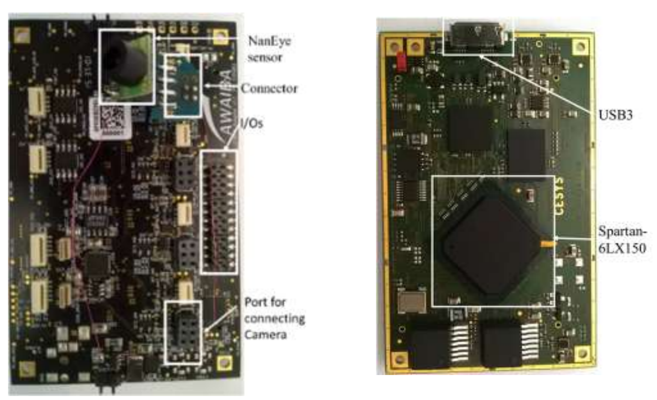
The goal of this research is to design a gamma function for a small endoscopic camera. The NanEye sensors from AWAIBA, which have 42Fps – 55 Fps [5], fit in this criteria because of their low cost, low power, minimal footprint camera which makes it ideal test case. If the user want to use single one or multiple sensors (for example, 3D view) it needs high number of computations which is quite impossible in traditional hardware. That's why the field-programmable gate array (FPGA) is chosen for this problem. However, the implementation of the non-linear gamma function is quite complex in FPGA and it consumes a lot of FPGA resources. So at the end the extended goal of this research is to find a less resource hungry solution which can be implemented in FPGA for the gamma function describe in Eq.1.

The solution includes piece-wise linear polynomial approximations methods. The system can apply the gamma function on three color channels simultaneously.

This paper is organized as follows: materials and methods section discusses about the FPGA, camera, board and the methods used to design the piece-wise linear system. In the next section, the proposed design, presents the actual representation of the design realized in the FPGA. In the result and discussion section the piece-wise linear gamma corrected pixels are analyzed. At the end, the conclusion gives the summary of the design and its advantages and disadvantages.

Materials and Methods

The materials and methods used to design the system are described on the following subsections.



(a) AWAIBA NanEye USB3.0 v1.0 *(b) CESYS EFM-02 board*

Figure 1: Hardware system used for design and verification of gamma correction.

Materials

The endoscopic camera used here is a NanEye-RGB which has a footprint of 1.0x1.0x1.65 mm. In the picture (Fig. 1(a)), the camera is in a protective casing so its looks bigger. It is a true system on chip camera, with 10 bit serial data transmission over LVDS. It has 250 x 250 pixels (0.0625MP). The frame rate of camera is 42Fps – 55 Fps and it is adjustable over power supply to match any display. NanEye has electronic rolling shutter [5], [6].

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-276

For connecting the NanEye to the FPGA, AWAIBA's NanEye USB3.0 v1.0 board (Fig. 1 (a)) is used. A connector is needed to connect the sensor to the board. This custom designed board from AWAIBA can support up to four NanEye cameras. It has numerous I/Os for debug purpose.

The FPGA is soldered to CESYS EFM-02 board (Fig. 1 (b)) [7]. This board has a Spartan 6 XC6LX150 FPGA which has 147,443 logic cells, 1,335kb distributed RAM, 4,824 kb block RAM. The CESYS EFM-02 board uses USB 3 to power the FPGA. It also supply power to AWAIBA NanEye USB3.0 v1.0 board and the NanEye sensors. The programing of the FPGA is done through the USB3.

Methods

The simplest solution to implement gamma function would be the use of the Look Up Table (LUTs) or Read-only memory (ROMs) [8]. For this work the endoscopy camera NanEye is used, which has 10 bit pixels. The goal of the work is to make a gamma function adjustable from 1 to 4 with 0.1 intervals. Even a systems with a good memory it is costly to make such design which has 31 gamma functions. So the alternative solution is the piece-wise linear polynomial approximation [9]. Nevertheless, one linear approximation is not enough for faithful representation of a non-linear function like the gamma function. So implementing the piece-wise linear approximation creates more questions like how many segmentation is needed and how the segmentation could be done. Due to the nature of the gamma function: in the lower (at the beginning) part of the function has more curve than the upper portion of the curve that's why linear segmentation is not a good solution. So several different works [9], [10], [2] suggested that a nonlinear segmentation would be better solution.

For this work the gamma function is divided into ten linear segments, as Eq. 2 presents, where C_{1x} the slope or gradient is, C_{2x} is the intercept, and x indicates the segmentation number which represents a section. These sections are incremented with two's power to take advantage of non-linearity of the gamma function. The coefficients of each segment sections, i.e., slope and intercept are calculated with 95% confidence bounds. The same process is done for 1 to 4 gamma values with 0.1 increments.

$$\text{Output pixel value} = \text{Input pixel value} \times \mathrm{C}_{1\mathrm{x}} + C_{2x} \qquad (2)$$

Proposed Design

The proposed design can be divided into three sections: ROM, multiplexer and the mathematical operators. The input of the system is the coded gamma values to select proper gamma corrections. R, G, B are the red, green and blue color channel. Each one is 10 bits wide for both inputs and outputs which need the gamma corrections. Fval, and Lval are the vertical synchronization and horizontal synchronization signals. These two are used to synchronize the design blocks and pixel data. The Reset and Clock inputs are driven by the global reset and main clock used on the whole FPGA design.

ROM

The slope and intercept values indicated as coefficients for 10 segments of each 31 gamma values are calculated using the method described in the methods section. These 16 bits long coefficients are saved in the FPGA as a ROM (see Fig.2) by implementing a LUT based memory. According to user gamma selections this ROM delivers 20 coefficients (2 for each segments) to the Multiplexer.

Multiplexer

After the gamma value is chosen by the user. According to Eq. 2, to calculate valid output pixels respective coefficients among 20 coefficients is needed to be chosen for three different color channels. It is done by using a multiplexer (Fig.2).

Arithmetic Operations

Eq. 2 is implemented to get the accurate results for the color channels. It was implemented by using a multiplier [11] and an adder [12] using LogiCORE IP from Xilinx (Fig.2). To reduce the process time the system uses three parallel arithmetic operations for three different color channels. This way no buffer is needed to hold the pixel values when the system is performing operation on other color channel.

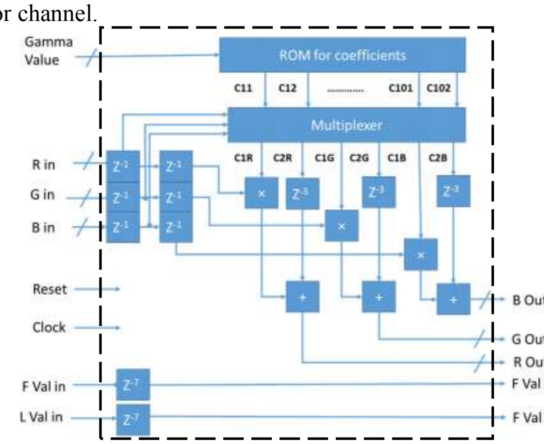


Figure 2: Block diagram of the implemented System in the FPGA.

Results and Discussion

The designed gamma correction system is analyzed in three different ways. First the error between the mathematical function and the hardware output is discussed and histogram analysis for the both with look-up table interpolation is analyzed. Then the quality of image is compared with the piecewise linear method implemented in hardware system and mathematical implementation of the gamma function. At the end the resources used by the designed hardware and Intellectual Property (IP) is also compared.

Error Analysis

Since the system has a 10 bits input and output for pixels, the error is calculated in 10 bits. Since the endoscopy system has general setting of gamma 1.2, the difference between mathematical calculation and hardware is shown in Fig. 3. The pixel values is represented in Digital Numbers (DN) which is 0 to 1023 for 10 bits picture it is also true for the camera used to verify the system.

The error is also calculated using Mean Absolute Error (MAE) using Eq. 3, in which M indicates mathematically gamma corrected pixels, H indicates the hardware corrected pixels and i is the input pixels.

$$\mathrm{MAE} = \frac{1}{1024} \sum_{i=0}^{1023} \left| \begin{array}{l} \text{corrected pixel}_{\mathrm{i,M}} \\ -\text{corrected pixel}_{\mathrm{i,H}} \end{array} \right| \qquad (3)$$

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-276

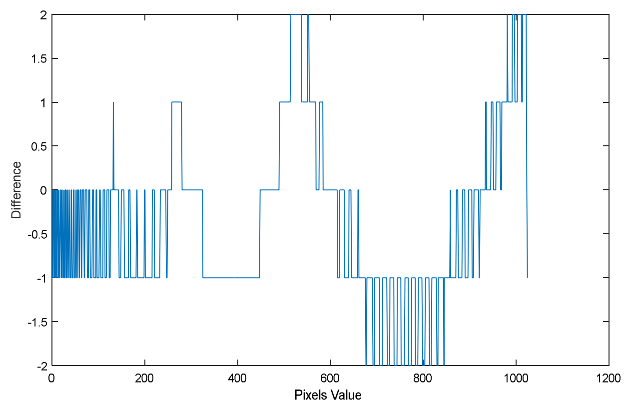


Figure 3: The difference between mathematical gamma and piecewise linear gamma in FPGA for the gamma value of 1.2

The Table 1 presents the MAE of the system with an average error of 2.1747. For the endoscopic camera the default value chosen is 1.2. From Table 1, it is clear that at the default value the error is less than the average. In addition to that, it is understandable that the difference is less than 3 DN.

Table 1: Mean Absolute Error for gamma range 1 to 4 for 10 bits

Gamma Value	MAE
1	0
1.1	0.775390625000000
1.2	1.13378906250000
1.3	1.47167968750000
1.4	1.74218750000000
1.5	1.94628906250000
1.6	2.10839843750000
1.7	2.15820312500000
1.8	2.30468750000000
1.9	2.36523437500000
2	2.46972656250000
2.1	2.45312500000000
2.2	2.46972656250000
2.3	2.54492187500000
2.4	2.52441406250000
2.5	2.52050781250000
2.6	2.49316406250000
2.7	2.49902343750000
2.8	2.52734375000000
2.9	2.47949218750000
3	2.46386718750000
3.1	2.45605468750000
3.2	2.45605468750000
3.3	2.45605468750000
3.4	2.41210937500000
3.5	2.42773437500000
3.6	2.38183593750000
3.7	2.38183593750000
3.8	2.34570312500000
3.9	2.32324218750000
4	2.32519531250000
Average MAE	**2.1747**

As it is described in the methods section, this design divided the gamma function over ten segments. The box whiskers representation is done in Fig. 4 to understand the contributions of errors by each of the segments. From this figure, an interesting phenomenon is understandable: the Least Significant Bits (LSB) (pixel value 0 and 1) are responsible for most of the errors in the system which has less effects on human eyes.

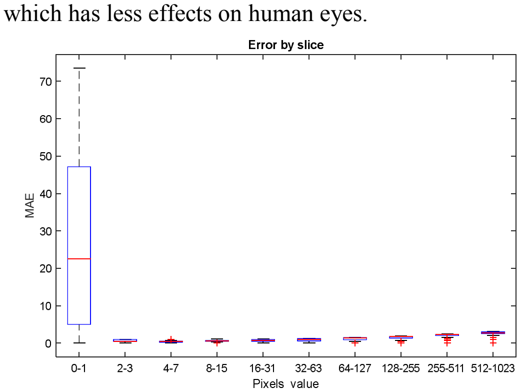


Figure 4. Box whiskers representation of Mean Absolute Error (MAE) corresponding slices for all 31 gamma choices. Red is median box is standard deviation whiskers are max and min errors.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-276

An alternative method described in [13] is the look-up table interpolation. Though the solution does not exists for 10 bits inputs, the solution stores every 4th sample, which can reduce the number of block RAMs used by 75.

The MSE of the look-up table interpolation and the piecewise linear method for all 31 gamma values are shown in Fig. 5. It shows that below gamma 1.4 the piecewise linear method has less error. The maximum error for piecewise linear method is 2.545, and it occurs at gamma 2.3. In the case of look-up table interpolation, the maximum MSE is 1.779 and it occurs at 3.9 gamma value. So the difference between the look-up table interpolation and piecewise linear method is less than one DN.

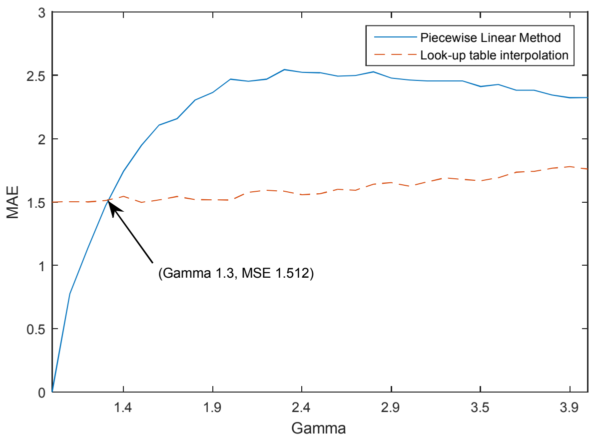


Figure 5. Comparison between look-up table interpolation (which stores every 4th sample) and piecewise linear method in 10bits.

A RGB picture of 250×250 10bits is generated using uniformly distributed pseudorandom integers from 0 to 1023. An almost flat histogram generated by random number ensure that the picture does not have any inefficient or too much lighting conditions. And the 1.2 gamma corrected picture's histogram shown in Fig. 6. The figure clearly indicate that both design: look-up table interpolation and piecewise linear method are following the mathematically corrected pixels curve. However, piecewise linear method has less difference with mathematical one compared to look-up table interpolation.

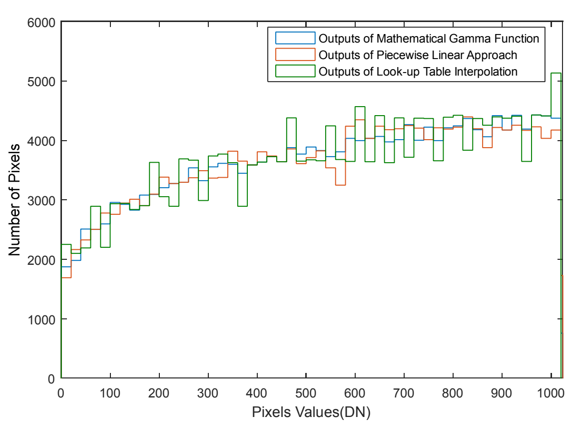


Figure 6. Comparison of image in terms of histogram mathematical gamma function, look-up table interpolation and piecewise linear gamma function.

Picture quality

The piecewise linear gamma is implemented in a Spartan 6 XC6SLX150T. It was connected via USB3 to the computer. As the data outputted by the designed block has 10 bit, the 2 LSB are right shifted, so that an 8-bit picture can be captured by Awaiba's viewer. This truncating 2 LSB bits is done due to the picture and video representation in true color which is 24-bit for RGB colors.

After truncating the bits the results are compared with 8 bits gamma function, and the MSE is 0.9227. It is clear from Fig. 7 that the error reduces compared to the 10 bits output in every gamma values. So the system's true color error is less than the 10 bits error. To match the number of the point between the 8bits gamma and designed gamma function down sampling is done.

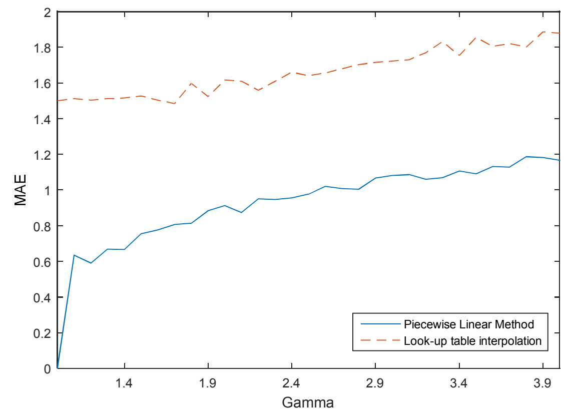


Figure 7. Comparison between look-up table interpolation which stores every 4th sample and Piecewise Linear Method in 8bits.

Fig. 8 is showing mathematically corrected gamma value of 1.2 in computer. And the Fig. 9 is the piecewise linear method. Both of pictures are in 8 bit color representation. There is almost no visible difference between the FPGA generated picture and the actual gamma equation in the computer.

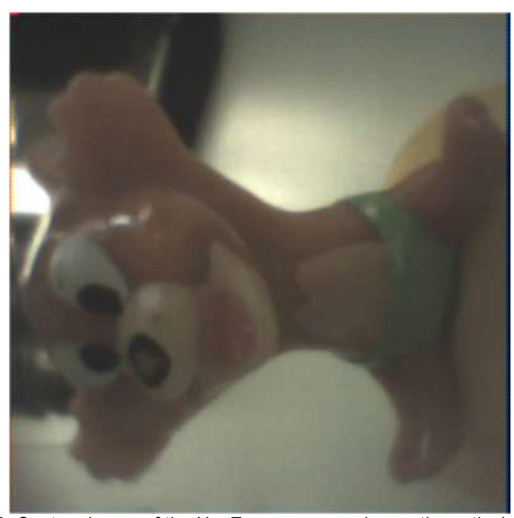


Figure 8. Capture image of the NanEye camersa using mathematical gamma function of 1.2

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-276

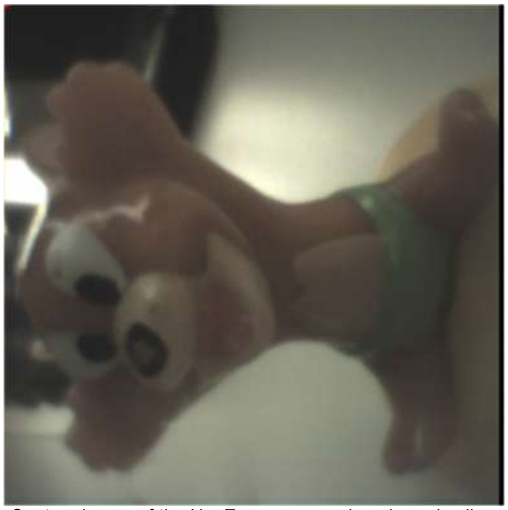


Figure 9. Capture image of the NanEye camera using piecewise linear gamma of 1.2 implemented in FPGA.

The histograms in Fig. 10 is generated using the pictures in Fig. 8 and Fig. 9. It is done to understand the practical characteristics of the piecewise linear system. From the figure it is seen that the hardware implementation with the camera is following the histogram of mathematical gamma function as it is predicted with generated pattern in Fig. 6 and Fig. 5.

However, a difference are noticeable at the beginning of the histogram which is due to the error in lower pixel values (see Fig. 4). In addition to that, slight left shift of the histogram is visible for piecewise linear method at highest histogram which may result slightest darker picture compare to mathematical gamma function in some cases.

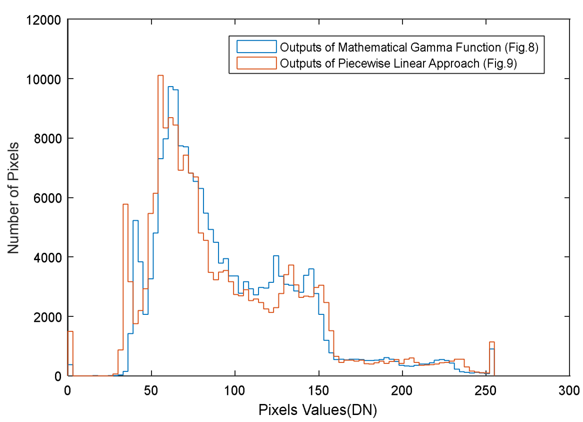


Figure 10. Comparison of images in terms of histogram between actual gamma function and piecewise linear gamma function with NanEye camera.

Performance and Resource Utilization

The comparison of resources used by the designed system and the ones used by LogiCORE IP Gamma Correction v3.0 [13] is done for Spartan 6 in Table 2. It is worth mention that the common resources used by both methods are not implemented in same way. The proposed system resources shown here were verified after generating the bin file. For LogiCORE IP data shown here is from the document [13]. N/A in table indicates that there is no available information in the document [13] to compare that particular resource.

From the comparison in the Table 2 it is seen that piecewise linear gamma do not use any BRAM16 and LUT6-FF pairs. On the other hand it uses DSP48 and BUFG which is not used by LogiCORE IP. FFs is used by both systems however piecewise linear gamma used a lot of FFs compared to LogiCORE IP. The design system has a maximum clock frequency of 250 due to the arithmetic ip cores. However, it's using 100 MHz clocking frequency when using the NanEye camera to capture the images or videos.

Table 2: Comparison of Resource Utilization and Target Speed for Spartan-6 - xc6slx150, between the LogiCORE IP Gamma Correction v3.0 [13] and piecewise linear gamma function.

Resource	LogiCORE IP	piecewise linear gamma
FFs	72	952(fully used 449)
LUT6-FF Pairs	59	0
BRAM16	3	0
BUFGs	N/A	1
DSP48A1s	N/A	3
Clock Frequency (MHz)	256	100 (max 250)

Conclusion

The proposed design uses the piecewise linear method to solve nonlinearity which has 2.1747 MAE (in 10 bits) actual hardware error. However, when the system is showing the picture in true color (8bits RGB) it is reduced to 0.9227. Moreover, the error can be reduced by adding more segments especially in lower portion of the function.

To save a large amount of memory space compared with traditional RAM based method piecewise linear method is clearly an alternative. It is not only saves the block RAMs but also has only ten linear equations to achieve the goal which reduces the number of coefficients as well as LUTs uses.

Furthermore, no external memory is used, which has two fold advantages: normally LUTs are faster and consume less power than the external memory.

The maximum speed of the designed system is restricted by the IP core used for arithmetic operations. However, this system can run up to 250 MHz which is almost same as the IP core (6 MHz less).

The different protocol like AXI4 or Wishbone can be added to the system to control the system externally.

Same design method can be used for different bit size of input output systems. In ROM based method it can be seen that from 10

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-276

bits inputs outputs to 12 bits inputs outputs the number of BRAM16 changes from 3 to 9 which is 200 percentage increment of memory[13]. In theory if the bit size is increased current piecewise linear methods have to increase two 16 bits coefficients for each bit increments other resources stays almost the same which is a significant achievements over the ROM based method.

References

[1] A. F. Peery, E. S. Dellon, J. Lund, S. D. Crockett, C. E. McGowan, W. J. Bulsiewicz, N. J. Shaheen, "Burden of Gastrointestinal Disease in the United States: 2012 Update." Gastroenterology, vol. 143, Issue. 5, pp. 1179–1187.e3, 2012.

[2] D.-U. Lee, R.C.C Cheung, J.D. Villasenor, "A Flexible Architecture for Precise Gamma Correction", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol: 15, Issue: 4, Pages: 474 - 478, 2007.

[3] S. Kang, H. Do, B. Cho, S. Chien, and H. Tae, "Improvement of low gray-level linearity using perceived luminance of human visual system in PDP-TV," IEEE Trans. Consum. Electron., vol. 51, no. 1,pp. 204–209, Feb. 2005.

[4] S. Hecht, "A theory of visual intensity discrimination," J. General Physiol., vol. 18, no. 5, pp. 767–789, 1935.

[5] *Specification NanEye 2D Web v4.4*, AWAIBA, Lda Madeira, October 19, 2015. http://www.cmosis.com/products/product_detail/naneye

[6] *Product Sheet NanEye Area Scan Sensors*, AWAIBA, Lda Madeira, http://www.cmosis.com/products/product_detail/naneye

[7] *EFM-02 Hardware reference (Hardware Revision 1.2)*, CESYS GmbH, Herzogenaurach, Germany, UG102 (v1.7) April 07, 2014 www.cesys.com

[8] B. Lucas, "Method and apparatus for converting floating-point pixelvalues to byte pixel values by table lookup," U.S. Patent 5 528 741, Jun.18, 1996.

[9] T. Lin, H. Cheng, and C. Kung, "Adaptive piece-wise approximation method for gamma correction," U.S. Patent 6 292 165, Sep. 18, 2001.

[10] E. Kim, S. Jang, S. Lee, T. Jung, and K. Sohng, "Optimal Piece Linear Segments of Gamma Correction for CMOS Image Sensors." IEICE Transactions vol. 88-C, no.11, pp. 2090-2093, 2005.

[11] *LogiCORE IP Multiplier v11.2*, Xilinx, Inc. San Jose, California, (DS255), March 1, 2011.

[12] *LogiCORE IP Adder/Subtracter v11.0*, Xilinx, Inc. San Jose, California, (DS214), March 1, 2011.

[13] *LogiCORE IP Gamma Correction v3.0*, Xilinx, Inc. San Jose, California, (DS719), September 21, 2010

Authors Biography

Sheikh Shanawaz Mostafa received the B.Sc. Engg. degree in Electronics and Communication Engineering from Khulna University (KU), Khulna, Bangladesh, in 2010 and the M.Sc. degree in Biomedical Engineering from Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh, in 2012. He is currently pursuing the Ph.D. degree in Networked Interactive Cyber Physical Systems at Instituto Superior Técnico, Portugal. His research interest includes the development of hardware for biological/medical treatment/diagnosis techniques.

L. Natércia Sousa received her Master degree in Telecommunications and Energy Networks Engineering from Universidade da Madeira in 2015.She is currently research assistant on Madeira Interactive Technologies Institute (MITI). The projects she is involved are developed in association with Awaiba, and they are focusing on FPGA design.

Nuno Ferreira received the diploma degree in Networks and Telecommunications Engineering from the Universidade da Madeira, Portugal, in 2005. He is currently finishing his doctoral degree in Electrotechnics Engineering at the Universidade de Aveiro, and teaching at Universidade da Madeira. He has integrated a team for VHDL development in an Image Processing R&D project, in collaboration with Awaiba company. He is currently with a research fellowship in numerical simulation of gas discharges with the Physics group at Universidade da Madeira.

Ricardo M. Sousa received his Master's in Telecommunications and Energy Networks Engineering from the University of Madeira, 2014. In 2015 he won a IS&T/SPIE Electronic Imaging 2015 Best Student Paper Award for a paper presented at: Image Sensors and Imaging Systems conference. He currently works at Awaiba Lda, where he is an Electronic Engineer specialized in FPGA firmware development and image sensor test and characterization. He is a member of the SPIE society.

João Santos received his Degree in Software Engineering in 2009 and his Masters in Software Engineering in 2010 from University of Madeira. Since then he has worked as a Computer Programmer in Awaiba, Funchal, Portugal. His work is focused in drivers development, design GUI and image processing algorithms.

F. Morgado-Dias, received the B.S. degree (5-years) from the University of Aveiro in 1994 and the M.S. degree from the University Joseph Fourier in Grenoble, France in 1995 and a PhD from the University of Aveiro, Portugal, in 2005.He was Lecturer at the Technical University of Setúbal, Portugal and is currently Assistant Professor at the University of Madeira and Researcher at Madeira Interactive Technologies Institute. His research interests include Renewable Energy, Artificial Neural Networks and FPGA implementations.

Martin Wäny graduated in microelectronics IMT Neuch,tel, in 1997. In 1998 he worked on CMOS image sensor at IMEC. In 1999 he joined the CSEM, as PHD student in the field of digital CMOS image sensors and High dynamic range pixels. He won the Vision price for the invention of the LINLOG Technology (2000) and the Photonics circle of excellence award of SPIE (2001). He is founder and CEO of AWAIBA Lda and co-founded Photonfocus AG.