JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

FPGA Implementation of Non-Adaptive Image Interpolation Algorithms – A Survey

Dr.Ashwani Sethi [1] Er.Vishal Kumar[2]

[1,2]University College of Engg & Technology

[1,2]Guru Kashi University, Talwandi Sabo

ABSTRACT

Image interpolation is a method of generating a high resolution image from a low resolution image by introducing new pixels from existing pixels. High resolution images are necessary in a variety of image processing applications like bio-medical imaging, printing, satellite image processing, video processing and surveillance. Image interpolation is mainly classified into two types as non-adaptive and adaptive image interpolation. Non-adaptive image interpolation is preferred for reducing the chip as the non-adaptive image interpolation requires less number of computations than adaptive image interpolation. This paper analyses a different non-adaptive image interpolation algorithms and their implementations on field programmable gate array (FPGA). The analysis is performed in terms of interpolated image quality based on its peak signal to noise ratio (PSNR), number of look up table (LUTs) and configurable blocks (CLBs) required for realizing the interpolation algorithm in FPGA. The survey shows that the first order polynomial interpolation method outperforms the other interpolation techniques with moderate computational complexity.

Keywords— bilinear, bi-cubic, convolution interpolation, FPGA, image scaling

I. INTRODUCTION

Image interpolation is extensively used for matching digital images of different image sensors with different image resolution by enhancing the resolution of given input image [1]. Image interpolation is also termed as image up-scaling and image up-sampling [2]. Image up-sampling or up-scaling, furthermore called as single-image high resolution and which is a significant process for different image processing applications and machine vision like high definition television, digital photographs, object recognition and image editing. Image interpolation is moreover necessary for different system of geometric transformation such as image resizing, deformation of images and image rotation. Image interpolation is also used to decrease the artifacts like blurring and aliasing effect of any geometric transformation [3]. Furthermore, image interpolation is vital for eliminating impulse noise by assuming noisy pixels as missing pixels [4]. Additionally, image interpolation is utilized to replicate sharp textures and sharp edges by decreasing pixel blocking and blurring [5]. Thus, recently, digital image interpolation has been frequently utilized in multimedia devices like graphics renderers, high definition television, and media players [6]. Therefore, image interpolation is an essential and important part of modern electronic systems [7].

The image interpolation process generates empty spaces in the input image and then filling in the empty spaces with the suitable pixel values. This makes the interpolation process yielding different solutions based on the concept utilized to find those values. For instance, in the nearest neighbour algorithm, the empty space is filled by using the nearest neighbouring pixel [8]. Image interpolation is a process by which a small input image is made larger output image by improving the number of pixel values comprising the small image [9] and it is referred as upscaling. It is used when the image is changed from one pixel level to another pixel level [5]. Image interpolation works on the concept of image re-sampling. Re-sampling is a process of converting a discrete image that is represented at one set of coordinate point to a new set of coordinate points. The interpolation process is used to finding the information for missed pixels or undefined pixels in a source image based on the information presented by given pixels. The given pixel information normally includes information related to density, coordinates, colour or gay level [10]. Interpolation is usually performed by convolving an input image with a small kernel of the weighting function [11].

Image interpolation is categorized as non-adaptive image interpolation and adaptive image interpolation. Non-adaptive image interpolation algorithm trains the image interpolation model utilizing the whole image sample set. But the adaptive image interpolation algorithm trains the model by utilizing only useful local information.

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

These are also called as globally non-adaptive (linear) and locally adaptive (non-linear) interpolations. The non-adaptive image interpolation techniques are conventional nearest neighbour, bilinear, bicubic and cubic-spline. These methods have low computational complexity but not able to adapt to changing pixel structures, any they produce artifacts like blurred edges and blocking effects in the interpolated image. The adaptive image interpolation produces better results than non-adaptive image interpolation since it is easier to get some functions that are able to producing predictions on some specified areas of the images [12]. The non-adaptive image interpolation treats pixels as a series of raw intensity values and it does not use run-time decisions on the technique to be used. But the adaptive image interpolation classifies pixels as either the regions of edge and non-edge. Based on is pre-processing, different techniques are applied on pixels of different regions [13].

Real-time image/video processing applications attain a set of image data from different source, process the image data and provide as output images that later are investigated depending on the application. Throughout the history different options for real-time image data processing have been used. One of these options is the usage of supercomputers or any other dedicated workstations. A different option is the usage field programmable gate arrays (FPGAs), which are digital programmable/reconfigurable devices; they are programmed with the description of a structure based on basic hardware components. Currently, FPGAs are available with the capacity of millions of gates. Reconfigurable computing is a popular method for an end solution of any image processing system. By comparing with other options like application specific integrated circuits (ASICs) and microprocessors, the ASIC is very efficient but it cannot be modified by user, and microprocessor is more flexible but it is not enhanced for some applications. FPGAs are used in image processing applications like computer vision due to their capacity of integrating parallel architecture to perform real-time operations. FPGA based image processing systems reduce complexity and increase the speed of operation by using N number of stages for any complex operation [14]. FPGAs also provide the best platform for testing as well as prototyping of any system [15]. FPGAs provide cost-effective and an easy way to evaluate digital signal processing (DSP) algorithms from an implementation perception. The DSP algorithms are developed and tested by using high level languages like MATLAB or C. After testing design, the high level algorithmic description is translated onto an FPGA platform by using intellectual property (IP) cores. The IP core based system design reduces the time and effort of hardware development [16].

The remaining part of this paper is organized as follows. In section 2, the basic hardware architecture of image interpolation algorithm is briefed. In section 3, different FPGA implementation of non-adaptive image interpolation algorithms are briefly explained. Section 4 provides a comparative analysis and finally section 5 provides conclusions.

II.HARDWARE ARCHITECTURE OF IMAGE INTERPOLATION

The hardware architecture of image interpolation algorithms like bi-cubic, linear convolution, winscale and polynomial convolution interpolation includes the coordinate calculation unit, memory bank, weighting coefficients generator, vertical interpolator, virtual pixel buffer and horizontal interpolator. Each of the units performs specified operations for the interpolation. The hardware architecture with its different units is shown in Figure 1 [17].

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

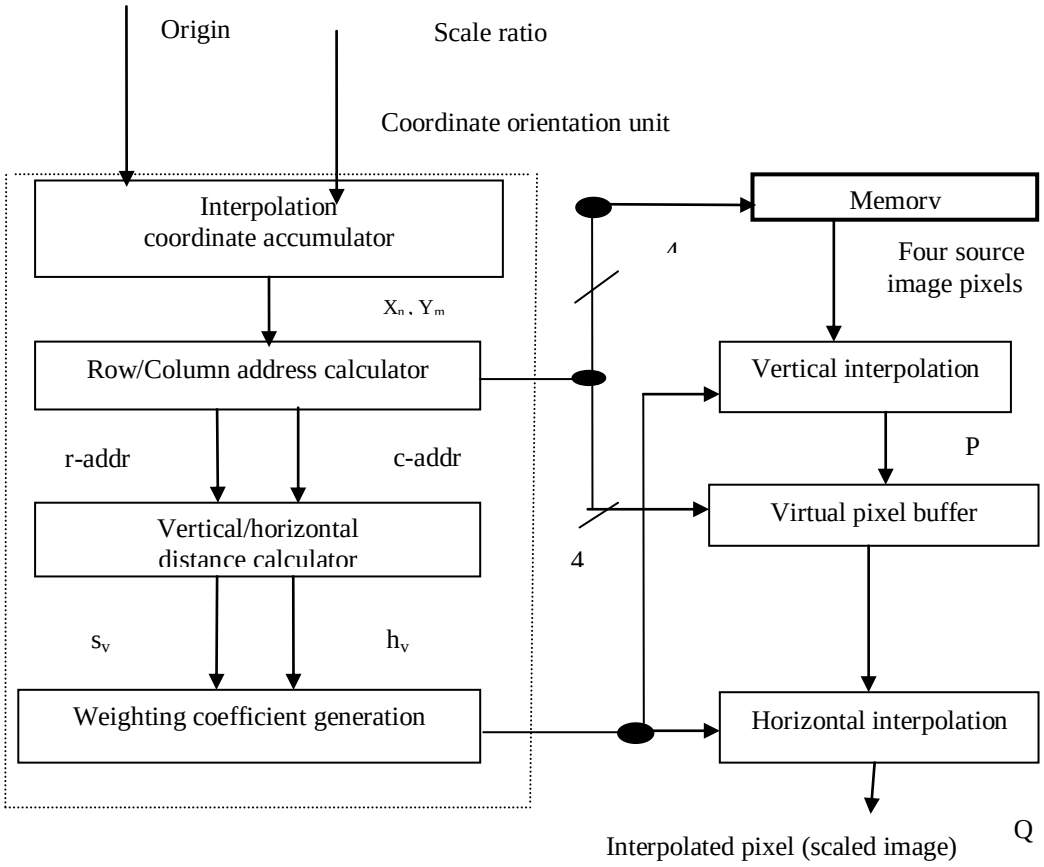

Figure 1: Block diagram of the hardware architecture of image Interpolation

The main blocks of coordinate calculation unit are interpolation coordinate accumulator, row / column address calculator and vertical and horizontal distance calculator. For image scaling, the coordinate system of interpolated pixels is different from that of a source image. The coordinates of an interpolated pixel depend on scale ratio and thus, they require precise calculations of integer and fraction.

The relevant co-ordinate of the current target pixel to be interpolated is calculated by the co-ordinate accumulator. It takes two scaling factors as inputs: one for the horizontal direction and the other for the vertical direction. The scaling ratio of each direction is determined by these parameters.

In the circuit of row / column address calculator, the operation of vertical or horizontal address orientation is controlled by using a control signal (vertical/horizontal). This signal determines the vertical (y_m) and the horizontal (x_n) coordinates. When the control signal (vertical/horizontal) is vertical, then the row addresses (r-addr) is obtained and when the control signal (vertical/horizontal) is horizontal the column addresses (c-addr) is obtained.

In interpolation, vertical and horizontal weighting coefficients are generated. The method of generating horizontal weighting coefficient is similar to the way of calculating vertical weighting coefficient. In the weighting coefficient generator, the distance in vertical direction (s_v) and horizontal direction (h_v)are calculated by using vertical/horizontal distance calculator. The vertical distance calculator calculates distance between the source pixel $A_{i+1,j}$ and the virtual interpolated pixel P_j. Similarly, horizontal distance calculator calculates distance between the virtual interpolated pixel P_j and the final interpolated pixel Q.

The source image which is used for the interpolation operation is stored in memory bank. The image pixels are retrieved from the memory bank based upon the generated row and column address. For every interpolation operation, source pixels that around an interpolated point of a source image are accessed from this memory bank. The virtual pixels created from vertical interpolation are stored in the virtual pixel buffer, and from the virtual buffer the horizontal interpolation unit accesses the virtual pixels for horizontal interpolation.

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

III.REALIZATION OF NON-ADAPTIVE IMAGE INTERPOLATIONS IN FPGA

The hardware architecture of various non-adaptive image interpolation algorithms have been implemented by using FPGAs during the last decade.

Bi-cubic interpolation for picture scaling was proposed by Maganda & Estrada [14]. Three main blocks make up the hardware architecture for bi-cubic interpolation (HABI): the first generates coefficients, which implements the bi-cubic function to be used in HABI; the second performs the interpolation process; and the third is a control unit that synchronises the processing and pipeline stages. The architecture is designed to operate with monochrome photos, but it may be modified to work with RGB colour images as well. This design is implemented on a Virtex-II pro FPGA from Xilinx. The system runs 10 times faster than an Intel Pentium 4-based PC at 2.4 GHz. The hardware architecture requires about 890 CLBs, 28 Block RAMs at 100 MHz operating frequency and 3.50 ms processing time. The architecture is a small, efficient module that may be integrated with other high-performance architecture to construct more robust applications such as tracking systems, robotics, and mobile apps. This architecture may be upgraded to increase parallelism without affecting control or memory.

An efficient bi-cubic convolution interpolation is presented by Lin et al. [18] for digital image processing algorithm. The architecture of reducing the computational complexity of generating coefficients and decreasing the number of memory access times is proposed. The number of memory access times required to interpolate a row in this approach is constant and does not change with the scale ratio. The number of multipliers and adders used to generate coefficients are less than the original bi cubic algorithm. The hardware architecture of this method has low computational cost and it is easy to implement. The hardware of this algorithm is analysed by using FPGA. The FPGA implementation demonstrates that the algorithm requires only about 437 logic blocks but the bi-cubic algorithm [14] requires about 890 logic blocks.

Gribbon & Bailey [19] presented bilinear interpolation for real-time image processing. This work is useful in different applications, such as lens distortion correction where the input coordinates follow a curved path that spans multiple rows. The aim of this work is to improve the quality of interpolated image with high speed. A specific caching scheme is proposed to get the needed pixels in a single clock cycle, which would enhance the speed of operation. The hardware architecture of this interpolation is implemented on Spartan-II FPGA. The FPGA implementation utilizes about 329 CLBs out of 1172 CLBs and three numbers of Block RAM out of 14 Block RAMs. The use of three-point interpolation rather than the standard four-point interpolation introduces extra flaws into the recorded pixel values, which is the system's fundamental flaw. The fact that input coordinates do not emerge on straight lines complicates bilinear interpolation, which can be used to rectify lens distortion.

Fahmy [15] presented a generalised parallel bilinear interpolation architecture for vision systems. Bilinear image interpolation is extensively used in computer vision for generating pixel values for locations that lie off the pixel grid in an image. For every sub-pixel, the values of four neighbours are utilized to calculate the interpolated value. This work improves memory access by parallelism in embedded memories and increases the speed of interpolation. This method attains performance of 250M samples per second in a modern device. The memory requirements of this method differ linearly with image size. For larger images that would need external memory, the resulting speed would depend on the greatest access speeds for the external memory. The software used in this method is Xilinx Virtex 4 XC4VSX55 FPGA. This architecture utilizes 362 Slices, 256 numbers of 18 K bit Block memories, three numbers of DSP48 blocks. This system can operate at 250 MHz and can process 512 x 512 pixel images. This also implies that the Virtex – 5 device can process pictures with a resolution of 1024 by 1024 pixels. Computer vision can be done using this technology. The main disadvantage of this system is that it needs a big amount of memory.

Andreadis & Amanatiadis [20] presented a digital image scaling algorithm for both grey-scale and colour images of any resolution in any scaling factor. This technique takes a mask of up to four pixels and utilises two parameters to compute the final brightness of each pixel: the percentage of area the mask covers from each source pixel and the difference in luminosity between the source pixels. This interpolation operates on linear

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

area domain and uses continuous area filtering. For quick real-time deployment, it can execute both upscale and downscale procedures. This approach is run on a Quartus II FPGA with a 55MHz operating frequency. This interpolation's hardware design has 20 additions and 13 multiplications. This approach has a lower root mean square error (RMSE) than other interpolation schemes such as closest neighbour, bilinear, and winscale [21](Kim et al. 2003)21. The winscale19, on the other hand, involves less calculations, such as 11 adds and 10 multiplications.

Amanatiadis et al. [22] presented architecture for fuzzy area based image interpolation algorithm. This algorithm uses a mask joined with neighbourhood fuzzy. The functions that pay to the interpolated image are the areas of the source pixels, overlapped with a mask, and the intensity difference between the source pixels. Fuzzy if-then policy is used for those functions to perform the interpolation. This algorithm also utilizes a continuous domain filtering. This algorithm is used for both colour and grey-scale images with any scaling factor. This method achieves a lowest RMSE as 0.0139 for the test image pepper. The hardware architecture of this algorithm is simulated and implemented by using Altera Quartus II FPGA. This system interpolates gray-scale images with 10 bit resolution for a maximum image size of 1024 x 1024. The operating frequency of this system is 65 MHz.

Chen [23] presented an image scaling processor by using bilinear interpolation, clamp filter and spatial sharpening filter. The filters are used as pre-filters to avoid the blurring and aliasing artifacts of bilinear interpolation. The sharpening filter enhances the edges by removing associated noise and the clamp filter is used to smoothening unwanted discontinuous edges in the boundary regions. The filtered pixels are the either up-scaled or down-scaled by bilinear image interpolation technique. This work achieves a highest PSNR as 27.87 dB by using sharpening filter and bilinear interpolation, 28.80 dB by using clamp filter and bilinear interpolation and 28.78 dB by using combined filter and bilinear interpolation. The hardware architecture of this interpolation algorithm is evaluated by using Altera FPGA EP2C70F896C6 core. This architecture contains only 6.08 K gate counts and only one line buffer memory.

Lin et al. [24] presented a linear interpolation, which is low-cost hardware architecture with digital image scaling for real time requirements. This scheme has the advantage of low operation complexity which reduces the coefficient generating effort and hardware cost with the interpolation quality, compatible to that of bi-cubic convolution interpolation [14]. The number of adders and subtractors used to generate weighting coefficients in this method are much less than the bi-cubic algorithm[14]. The hardware architecture of this algorithm is implemented on Virtex –II FPGA to process digital image scaling for HDTV in real-time. As a result, this architecture has solved the problem of interpolation's computational complexity while also simplifying the hardware circuit. The high performance architecture of extended linear interpolation can be simulated at 104MHz with 379 LBs (Logic Blocks) which is able to process digital image scaling. This architecture requires about 26200 gates but the winscale architecture [21] requires about 29000 gates. In addition, this method achieves that higher PSNR (33.33dB). But the winscale [21] and bi-cubic achieve lower PSNR as 29.74 dB and 33.06 dB respectively for the 3/2 upscaling image after 2/3 downscaling of the test image. Further, this method achieves a highest PSNR 35.29 dB for the 2/3 downscaling image after 3/2 upscaling of the test image (tank). This image interpolation is a low cost architecture with the interpolation quality compatible to that of bi-cubic convolution interpolation.

Lin et al. [25] presented an efficient extended linear image interpolation. Extended linear interpolation necessitates the creation of 16 weighting factors from 16 neighbouring pixels in the original picture. The computational cost of creating weighting coefficients is reduced using this approach. The principle of generating coefficients is similar to the previous methods [18] and [22]. However, this method simplifies the technique of generating weighting coefficients. The number of arithmetic elements, such as adders and subtractors for generating weighting coefficients, are less than the bi-cubic [14], [21] and [22]. The FPGA implementation tells that this architecture utilizes only about 376 configurable logic blocks (CLBs) but the bi-cubic architecture [14] requires that about 890 CLBs, 379 CLBs respectively. Furthermore, this method provides higher PSNR than winscale [21] and bi-cubic [14] algorithms. The achieved PSNR of this method is 35.29 dB for the test image (tank) for 3/2 upscaling after 2/3 downscaling, which is similar to the previous method [23].

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

Kim et al. [21] proposed a winscale image interpolation algorithm for image scaling. Using an area pixel model rather than a point pixel model, this approach can perform both scale-up and scale-down transformations. To keep things simple, it calculates one pixel of a scaled image using no more than four pixels from the original image. It may also deliver greater quality due to features like fine-edge and variable smoothness. Winscale offers a nice scale property and is simple to use. It preserves the edge characteristic of an image well, and handles streaming data directly and requires only small amount of memory. Despite the up/down ratio, picture interpolation may be done with four line buffers. Furthermore, the winscale approach has a lower RMSE and higher picture quality than the bilinear technique, according to this research. At a 65 MHz operational frequency, this solution uses 29000 gates. Winscale may be used in a wide range of digital display devices that require picture scaling, particularly in applications that demand high image quality at a cheap cost of hardware. This algorithm achieves PSNR as 38.63 for the test image airplane.

Lin et al. [17] proposed a fast first-order polynomial convolution interpolation (FFOPCI) for real-time digital image reconstruction with low computational complexity. This work presents high-performance architecture of a novel first-order polynomial convolution interpolation for digital image scaling. In general, a higher order model produces better image interpolation, but also necessitates more sophisticated computations. Using first order polynomial convolution with acceptable quality, this study attempts to decrease the computing cost. The suggested method's kernel is made up of first-order polynomials and approximates the ideal sinc-function in the range [-2, 2]. This approach decreases the computational complexity of creating weighting coefficients, resulting in a simple hardware architecture, minimal computing cost, and easy compliance with real-time requirements. The Virtex-II FPGA is used to implement the architecture. The number of adders and subtractors used to generate weighting coefficients in the architecture is much less than that of bi-cubic interpolation[18]. Consequently, this architecture has solved the problem of computational complexity of interpolation and furthermore, simplified the circuit and reduced chip area. The Virtex – II FPGA utilizes 414 CLBs for this algorithm but the Virtex – II utilizes 437 CLBs for bi-cubic [18] algorithm. This method achieves highest PSNR as 49.53 for the test image (airplane) by ¾ downscaled image after 4/3 upscaling.

Li and Yu [26] presented a block region of interest method for FPGA implementation of image interpolation algorithm. In this method, a block region of interest is used for large image reconstruction using 2D sinc interpolation. This method utilizes a partitioning scheme for subdividing the target image into blocks on the region of interest in the source image. This scheme improves the memory access and it uses a special memory pattern to issue 16 pixels for each and every interpolation. The hardware architecture of this method is realized by pipelining on Virtex 7 FPGA, This method utilizes 187680 LUTs, 288272 registers. The number of LUTs used by this method is larger than the number of LUTs utilized by the first order polynomial convolution interpolation [17].

IV.COMPARATIVE ANALYSIS

The comparison on different non-adaptive image interpolation techniques is performed on the basis of interpolated image quality and the computational effort required performing interpolation. PSNR, LUTs, and CLBs on various FPGA devices are among the metrics evaluated for this study. Table 1 gives the comparison of different interpolation architecture of non-adaptive image interpolation algorithms such as bi-cubic (BC), efficient bi-cubic (EBC), bilinear (BL), parallel BL, digital image scaling (DIS), fuzzy area-based (FAB), linear convolution (LC), extended linear (EL), winscale, FFOPCI and sinc interpolation (SI) in terms of PSNR (dB), gate count (K), and number of configurable logic blocks.

Based on Table 1, the different types of FPGA used for designing these interpolation schemes are Virtex II, Virtex IV, Quartus II, and Spartan II devices. By comparing Virtex-II based interpolation methods, the bicubic interpolation [14] system requires a large number of CLBs as 890. But the extended linear interpolation [23] requires a less number of CLBs as 376 and the gate count of this method is only 25.98 K. But the bilinear interpolation [21] requires less gate count as 6.08 K. However, based on quantitative measure (PSNR), the FFOPCI [16] achieves a highest PSNR as 49.53 dB with moderate number of logic elements as 414 CLBs. Therefore, a high quality image interpolator with less computational complexity is implemented by the principle

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

of FFOPCI. The recent sinc interpolation based architecture [24] utilizes a very large number of LUTs on Virtex 7 FPGA.
Figure 2 and Figure 3 show the comparative analysis on PSNR and computational complexity respectively.

Table 1: Comparison of FPGA based non-adaptive image interpolation algorithms

Interpolation method	FPGA	PSNR (dB)	Gate count (K)	CLBs/LUTs(Nos)
BC [14]	Virtex-II	N/A	N/A	890
EBC[18]	N/A	N/A	N/A	437
BL [19]	Spartan-II	N/A	N/A	329
PBL [15]	Virtex- IV	N/A	N/A	362
DIS[20]	Quartus-II	N/A	N/A	N/A
FAB[22]	Quartus-II	N/A	N/A	N/A
BL [23]	Altera	28.80	6.08	N/A
LC [24]	Virtex - II	33.33	26.2	379
EL [25]	Virtex - II	35.29	25.98	376
Winscale [21]	N/A	38.63	29	N/A
FFOPCI [17]	Virtex-II	49.53	28.9	414
SI [26]	Virtex 7	N/A	N/A	187680

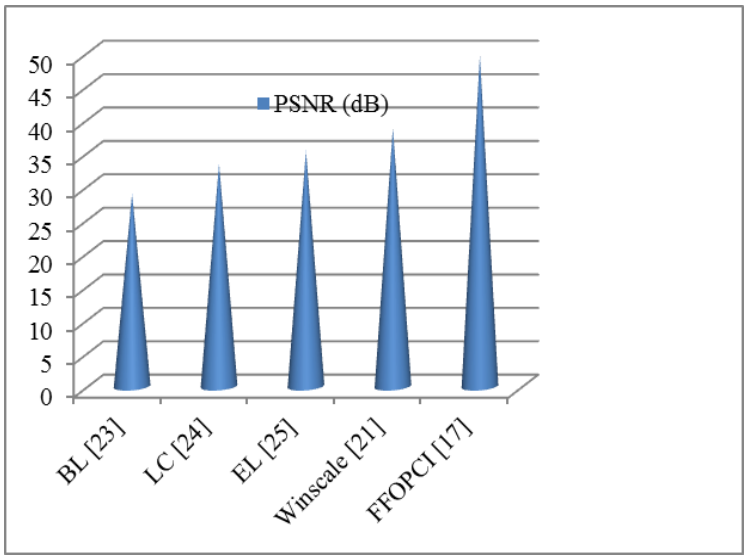

Figure 2: Comparison of PSNR of FPGA based non-adaptiveinterpolation algorithms

As shown in Figure 2 the FFOPCI [16] provides the highest PSNR among the various non-adaptive interpolation algorithms.

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

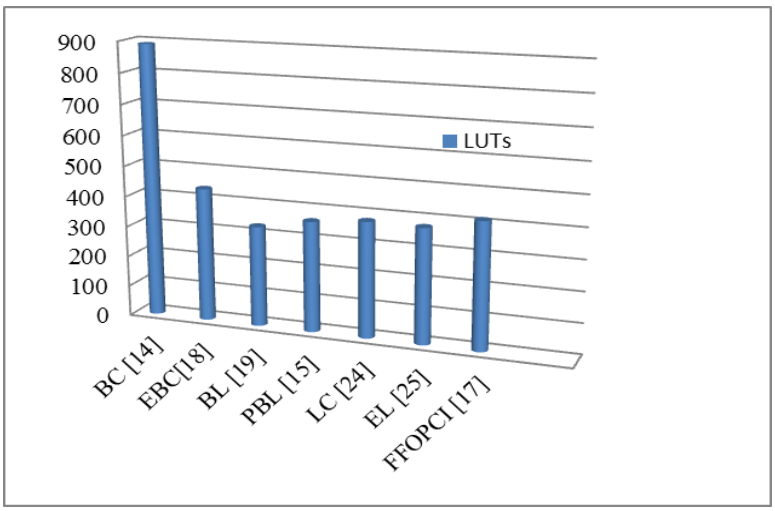

Figure 3: Number of LUTs in FPGA based non-adaptive interpolation architectures

Further, as shown in Figure 3 the FFOPCI [16] utilizes a moderate number of LUTs on FPGA. But the very recent work of sinc interpolation [24] utilizes a large number of CLBs on FPGA. Thus the FFOPCI [16] is suitable for further study and implementation of low complexity image up-scaling processor.

V.CONCLUSION

In this work, an extensive analysis is performed on the characteristics of non-adaptive image interpolation algorithms and their implementations on FPGA. Based on the exhaustive study, it is understood that the interpolated image quality is improved by using polynomial convolution interpolation and the computational complexity of image interpolation method is reduced by decreasing number of arithmetic operations on generating weighting coefficients of image interpolation. The survey shows that the first order polynomial convolution interpolation provides higher PSNR than other interpolation methods and it works on moderate number of LUTs for providing low computational complexity. This work also suggests that interpolation quality is improved by higher order convolution method and the complexity is reduced by decreasing number of arithmetic operation.

REFERENCES

[1] Arcelli, C, Brancati, N, Frucci, M, Ramella, G & Baja, GSD, 'A fully automatic one-scan adaptive zooming algorithm for color images', Journal of Signal Processing, Elsevier, 2014, vol. 99, pp. 61-71.

[2] Zhao, Y, Wang, R, Wang, W & Gao, W, High resolution local structure-constrained image up-sampling', IEEE Transaction on Image processing , 2015, vol. 24, no. 11, pp. 4394-4407.

[3] Delibasis, KK & Kechriniotis A, 'A new formula for bivariate hermite interpolation on variable step grids and its application to image interpolation', IEEE Transactions on Image Processing, 2014, vol. 23, no. 7, pp. 2892-2904.

[4] Kalyoncu, C, Yoygar, O & Demirel, H, 'Interpolation based impulse noise removal', IET Image Processing, 2013, vol. 7, no.8, pp. 777-785.

[5] Liu, X, Zhao, RX, Ma, S, Gao, W & Sun, H, 'Image interpolation via regularized local linear regression', IEEE Transactions on Image Processing, 2011, vol. 20 no. 12, pp. 3455-3469.

[6] Giachetti, A & Asuni, N, 'Real-time artifact-free image upscaling', IEEE Transactions on Image Processing, 2011, vol. 20, no.10, pp. 2760-2768.

[7] Chen, CL & Lai, CH, 'Iterative linear interpolation based on fuzzy gradient model for low-cost VLSI implementation', IEEE Transactions on VLSI Systems, 2014, vol. 22, no. 7, pp. 1526-38.

ISSN- 2394-5125 VOL 6, ISSUE 04, 2019

[8] Sharma, S & Walia, R, 'Zooming digital images using modal interpolation', International Journal of Application or Innovation in Engineering and Management, 2013, vol. 2, no. 5, pp.305-310.

[9] Oliver R & Hanqiang, C, 'Nearest neighbor value interpolation', Proceedings of international Journal of Advanced Computer Science and Applications', 2012, vol. 3, no. 4, pp. 1-6.

[10] Roy, R, Pal, M & Gulati, T, 'Zooming digital images using interpolation techniques', International Journal of Application of Innovation in Engineering and Management, 2013, vol. 2, no. 4, pp. 34-45.

[11] Shi, J & Reichenbach, SE, 'Image interpolation by two-dimensional parametric cubic convolution', IEEE Transactions on Image Processing, 2006, vol. 15, no. 7, pp. 1857-1870.

[12] Moses, CJ & Selvathi, D, 'A survey on adaptive image interpolation based on quantitative measure', Asian Journal of Research in Social Sciences and Humanities, 2016, vol. 6, no. 4, pp. 341-360

[13] Szydzik, T, Callico, GM & Nunez, A, 'Efficient FPGA implementation of a high-quality super-resolution algorithm with real-time Performance, IEEE Transactions on Consumer Electronics, 2011, vol. 57, no. 2, pp. 664-673.

[14] Maganda, MAN & Estrada, MOA, 'Real-time FPGA based architecture for bi-cubic interpolation: An application for digital image scaling', Proceedings of the 2005 international conference on reconfigurable computing and FPGA, 2005, pp. 1-8.

[15] Fahmy, SA, 'Generalized parallel bilinear interpolation architecture for vision systems', Proceedings of international conference on reconfigurable computing and FPGAs, 2008, pp. 331-336.

[16] Deng, L, Sobti, K, Zhang, Y & Chakrabarti, C, 'Accurate area, time and power models for FPGA-based implementations', Journal of Signal Processing Systems, Springer, 2011, vol. 63, pp. 39-50.

[17] Lin, CC, Shen, MH, Liaw, C & Chiang, HK, 'Fast first order polynomial convolution interpolation for real-time digital image reconstruction', IEEE Transactions on Circuits and Systems for Video Technology, Sept. 2010, vol. 20, no. 9, pp. 1260-1264.

[18] Lin, CC, Shen, MH, Chiang, HK, Liaw, C & Wu, ZC, 'The efficient VLSI design of bi-cubic convolution interpolation for digital image processing', Proceedings of IEEE international symposium on circuits and Systems, June 2008, pp. 480-483.

[19] Gribbon, KT & Bailey, DG, 'A novel approach to real-time bilinear interpolation', Proceedings of IEEE international workshop on electronic design, test and applications (DELTA), 2004, pp. 126-131.

[20] Kim, CH, Seong, SM, Lee, JA & Kim, LS, 'Winscale: An image-scaling algorithm using an area pixel model', IEEE Transactions on Circuits and Systems for Video Technology, 2003, vol. 13, no. 6, pp. 549-553.

[21] Andreadis, I & Amanatiadis A, 'Digital image scaling', Proceedings of IEEE international conference on instrumentation and measurement technology, 2005, pp. 2028-2032.

[22] Amanatiadis, A, Andreadis, I & Konstantinidis, K, Deswign and implementation of a fuzzy are-based image-scaling technique', IEEE Transaction on Instrumentation and Measurement, 2008, vol. 57, no. 8, pp. 1504-1513.

[23] Chen, SS, 'VLSI implementation of a low-cost high-quality image scaling processor', IEEE Transactions on Circuits and Systems – II, Express Briefs, 2013, 60 (1), pp. 31-5.

[24] Lin, CC, Shen, MH, Chiang, HK, Tsai, WK & Wu, ZC, 'Real-time FPGA architecture of extended linear convolution for digital image scaling', Proceedings of IEEE international conference on ICECE technology, Dec 2008, pp. 381-384.

[25] Lin, CC, Shen, MH, Chiang, HK, Liaw, C, Wu, ZC & Tsai, WK, 'An efficient architecture of extended linear interpolation for image processing', Journal of Information Science and Engineering, Jan 2010, vol. 26, pp. 631-48.

[26] Li, L & Yu, F, 'Block region of interest method for real-time implementation of large and scalable image reconstruction', IEEE Signal Processing Letters, 2015, vol. 22, no. 11, pp. 1908-1912.