

Research Article

VLSI Implementation of a High-Performance Nonlinear Image Scaling Algorithm

Osamah Ibrahim Khalaf ¹, Carlos Andrés Tavera Romero,²
A. Azhagu Jaisudhan Pazhani ³ and G. Vinuja⁴

¹Al-Nahrain University, Baghdad, Iraq

²Universidad Santiago de Cali, Cali, Colombia

³Ramco Institute of Technology, Rajapalayam, Tamilnadu, India

⁴Vins Womens Christian College of Engineering, Kanyakumari, Tamilnadu, India

Correspondence should be addressed to A. Azhagu Jaisudhan Pazhani; alagujaisudhan@gmail.com

Received 12 May 2021; Revised 6 June 2021; Accepted 21 June 2021; Published 22 July 2021

Academic Editor: Ayush Dogra

Copyright © 2021 Osamah Ibrahim Khalaf et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study implements the VLSI architecture for nonlinear-based picture scaling that is minimal in complexity and memory efficient. Image scaling is used to increase or decrease the size of an image in order to map the resolution of different devices, particularly cameras and printers. Larger memory and greater power are also necessary to produce high-resolution photographs. As a result, the goal of this project is to create a memory-efficient low-power image scaling methodology based on the effective weighted median interpolation methodology. Prefiltering is employed in linear interpolation scaling methods to improve the visual quality of the scaled image in noisy environments. By decreasing the blurring effect, the prefilter performs smoothing and sharpening processes to produce high-quality scaled images. Despite the fact that prefiltering requires more processing resources, the suggested solution scales via effective weighted median interpolation, which reduces noise intrinsically. As a result, a low-cost VLSI architecture can be created. The results of simulations reveal that the effective weighted median interpolation outperforms other existing approaches.

1. Introduction

Digital image interpolation or scaling is an issue that has recently received great attention. Image scaling is a process of resizing a digital image, and it is a nontrivial process that involves a tradeoff between efficiency, smoothness, and sharpness. Nowadays, the image scalar is widely adopted in portable healthcare devices, digital electronic equipment, digital camera, digital photo frame, mobile phone, touch panel computers, etc. It has become a significant trend to design a low-cost, high-quality, and high-performance image scalar by the VLSI technique for multimedia products. As the graphic and video applications of mobile handset devices grow up, the demand and significance of image scaling are more and more outstanding. The image scaling algorithms based on

interpolation are basically of two types: linear and nonlinear interpolation methods. The simplest linear interpolation method is a nearest neighbour algorithm which is a low-complexity algorithm, but it results in scaled images with blocking and aliasing artifacts. The most widely used scaling method is bilinear interpolation algorithm by which the target pixel can be obtained by using the linear interpolation model in both horizontal and vertical directions. Another popular polynomial-based method is bicubic interpolation algorithm, which uses an extended cubic model to acquire the target pixel by a 2D regular grid. The nonlinear interpolation methods such as weighted median interpolation, curvature interpolation, bilateral filter, and autoregressive model greatly improve image quality by reducing blocking, aliasing, and blurring effects compared to linear methods.

Many image scaling algorithms have been developed mostly based on interpolation and are edge-oriented. In this section, major scaling algorithms are explained. An edge-oriented area-pixel scaling processor was implemented with low-complexity VLSI architecture [1]. A simple edge-catching technique is adopted to preserve the image edge features. A JPEG edge-oriented area-pixel scaling processor performs scale-up/scale-down transformation by using the area-pixel model instead of the common point model with a simple edge-catching technique to preserve edge features effectively so as to achieve better image quality [2]. The direct implementation of area-pixel scaling requires some extensive floating-point computations so that a suitable approximate low-cost VLSI implementation technique has been used. A novel image zooming algorithm using curvature interpolation was developed. It results in clear images of sharp edges which are already denoised and superior to those obtained from linear methods and PDE-based methods [3]. A real-time FPGA architecture of the extended linear convolution for the image scaling method [4] provides simple hardware architecture design with low computation cost. Compared to the latest bicubic hardware design [5], the architecture saves about 60% of hardware cost.

A low-complexity memory-efficient image scaling processor uses bilinear interpolation with combined sharpening and clamp filters as the prefilter to reduce the blurring and aliasing artifacts [6]. The prefilter is implemented efficiently using T and inverse T models which need two line buffers for processing. Recently, a high-boost filtering-based image scaling algorithm has been developed [7, 8]. In the research environment, improved image processing based on both digital and remote sensing [9] images is analyzed by machine learning techniques [10–13] such as deep network [14], neural network [15, 16], and Markov analysis [17]. Other authors are well-versed in researching profound information to some level and their researches are cited as [18–20]. Wireless sensor network (WSN) [21–28], mobile ad hoc network (MANET) [29–31], web application [32, 33], cryptography security [34, 35], and cloud computing [36] play a significant role in the Internet of Things (IoT) [37, 38]. This efficient IoT network handles a huge volume of data named as big data (BD). This unstructured way of big data information holds huge irrelevancy along with redundant image details which are usually difficult to handle and access. So, the researchers proposed several new BD approaches [39, 40] to acquire the relevant details from the web. By these data, an effective image scaling approach is designed to insist its performance in the VLSI architecture.

In the proposed work, instead of using linear interpolation, a nonlinear method is adopted to enhance the performance of image scaling with reduced hardware complexity. For analyzing the performance of the proposed work, two recently developed image scaling techniques [7, 41] are explained in the following section.

2. Existing Techniques

A low-cost high-quality image scaling processor has been recently proposed [41]. It consists of a sharpening spatial filter, clamp filter, and bilinear interpolation. Figure 1 shows

the block diagram of the bilinear interpolation-based image scaling processor.

The combined sharpening and clamp filters serve as the prefilter to reduce blurring and aliasing artifacts in the scaled image. Hence, the computing resources and memory buffers are reduced by using this technique [41]. The clamp filter, a low-pass filter, is combined with the sharpening spatial filter as the prefilter to reduce the blurring effect. For efficient hardware implementation, a 3×3 clamp filter and 3×3 sharpening filter are combined together into a 5×5 filter as

$$\text{kernel [prefilter]} = \frac{\begin{bmatrix} 1 & 1 & 1 \\ 1 & C & 1 \\ 1 & 1 & 1 \end{bmatrix}}{(C+8)} * \frac{\begin{bmatrix} 1 & 1 & 1 \\ 1 & S & 1 \\ 1 & 1 & 1 \end{bmatrix}}{(S+8)}, \quad (1)$$

where C is the clamp parameter used to enhance the differences along the direction of edges to reduce the unwanted discontinuous edges and aliasing effects and S is the sharp parameter which is used to vary the degree of sharpening. These parameters are set according to the characteristics of the image. The kernel of the combined filter is given as

$$\begin{bmatrix} -1 & -2 & -3 & -2 & -1 \\ -2 & -2-C+S & -4-C+S & -2-C+S & -2 \\ -3 & 4-C+S & -8+SC & 4-C+S & -3 \\ -2 & -2-C+S & 4-C+S & -2-C+S & -2 \\ -1 & -2 & -3 & -2 & -1 \end{bmatrix}. \quad (2)$$

Figure 2 shows the low-cost VLSI implementation of the prefilter for the local window of size 5×5 [6, 41]. It consists of 25 shift registers to store 25 pixels of the 5×5 local window which is convolved with the coefficients of the prefilter. The convolution operation needs 8 shifters (SH), 5 shifter-adders (SA), 8 calculation units (CU), 1 multiplier-adder (MA), and 24 adders. The calculating unit is designed with a reconfigurable feature for computing clamp and sharp parameters.

The main limitation of this technique is high complexity due to combined filter design, logic for implementing hardware sharing, and reconfigurable techniques.

Another recent technique, a nonadaptive image scaling algorithm, using high-boost filtering has been proposed [7]. The image scaling is performed by linear interpolation, and then enhancement is done using high-boost filtering.

This technique results in high-quality image scaling, but the VLSI implementation needs complex hardware. Though many efficient image scaling algorithms have been developed, additional processing is required to enhance the scaling performance. Hence, the main focus of this work is to develop an efficient image scaling algorithm with less hardware complexity. Figure 3 shows the block diagram of nonadaptive linear interpolation-based image scaling.

For real-time applications, another VLSI architecture is implemented by using the anisotropic probabilistic neural network (APNN) [42]. APNN is one of the interpolation

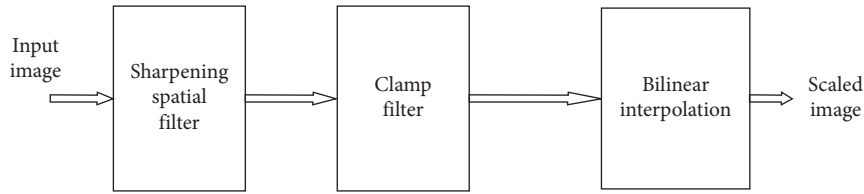


FIGURE 1: Block diagram of the bilinear interpolation-based image scaling processor.

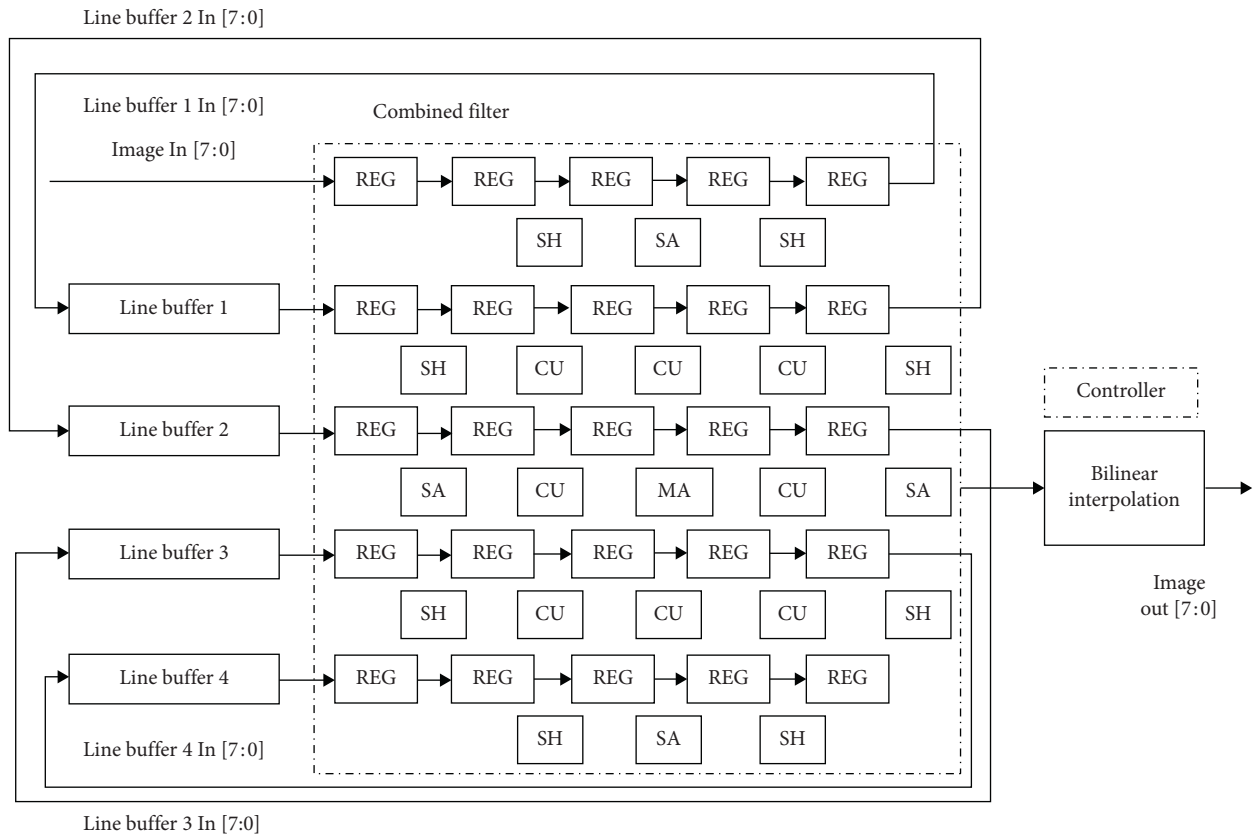


FIGURE 2: VLSI implementation of the prefilter.

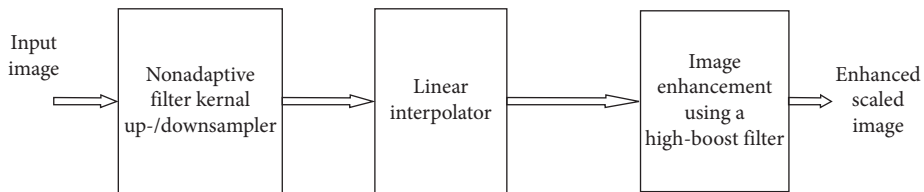


FIGURE 3: Block diagram of nonadaptive linear interpolation-based image scaling.

techniques employed within the VLSI architecture that sharply improves the edge region and greatly reduces the blurred effect of an image. In this implementation, the processing speed is usually four times faster than that of the personal computer at 3.4 GHz. However, a huge amount of resource utilization is one of the shortcomings made in the hardware APNN. To save hardware resource utilization, a

new approach of the VLSI system is proposed based on unified textual and dynamic compressive features (UTDCF) [43]. It performs several paradigms of memory-centric levels, multiple pipelines, and processing circuits to attain a high frame rate of object tracking capability. This approach not only consumes fewer resources and high speed but also attains reasonable memory consumption. Obviously, these

massive parallel circuits attain greater advantages in terms of real-time performance. However, it cannot compete with the majority of embedded applications.

3. Proposed Work

3.1. Motivation. Image scaling operation enlarges or reduces the size of the image (spatial resolution) in terms of pixels. Image resolution refers to the amount of information an image can hold and is controlled by the number of pixels or bit depth/pixel. As the resolution of an image changes from the capturing device to display or to print device, image scaling is normally required, especially when to match low-resolution display devices to high-resolution devices, and vice versa. When the resolution is larger, the scaling (enlarging) can be possible without any loss of sharpness and image details. Figure 4 shows the enlarged version of the images captured with different resolutions and squaring effects found in edges of the low-resolution image. Also, the scaled image needs larger memory and longer processing time. Hence, the main focus of this work is to develop a memory-efficient, high-performance VLSI architecture for image scaling algorithm.

3.2. Effective Weighted Median Interpolation-Based Image Scaling. In this work, an effective weighted median interpolation- (EWMI-) based image scaling algorithm is developed. It is a nonlinear method which performs interpolation as well as denoising. Hence, a low-cost hardware architecture is implemented, and it results in scaled images of high visual quality without using any prefiltering compared to linear interpolation methods. Figure 5 shows the block diagram of the proposed EWMI image scaling architecture. The major blocks are the register set to hold four neighbours, sorting block, and impulse noise detector and remover blocks. For computing the effective weighted median value for interpolation and denoising, a 3×3 local window is considered. An efficient sorter architecture is designed with two features such as precomputation logic and low computation complexity. Precomputation logic is added for power saving, and the sorted array size is made to odd for selecting the median value without addition and division operations.

For image scaling, first, an empty array of size $2N \times 2N$, where $N \times N$ is the input image size, is constructed and stored in memory. The array elements are labelled as shown in Figure 6. The elements with label $a_{i,j}^{00}$ are replaced by the original pixel values.

The remaining elements are interpolated using the proposed EWMI image scaling algorithm. Next, the elements with label $x_{i,j}^{11}$ are taken for interpolation. Its four diagonal neighbours (original pixel values) in the local window are multiplied with a weight value of 1 and given to the sorter unit. The median is computed from the sorted array, and then the effective median value is computed to replace $x_{i,j}^{11}$. Next, elements with labels $x_{i,j}^{10}$ and $x_{i,j}^{01}$ are interpolated by considering two horizontal neighbours (original pixels) and two vertical neighbours (previously

interpolated pixels). The neighbours with interpolated values are assigned with a suitable weight value in the range of 0.2 to 0.9. Pixels in the scaled array are interpolated by taking diagonal or vertical and horizontal neighbours in a local window of size 3×3 as shown in Figure 7. In order to speed up this processing, interpolation using diagonal neighbours can be overlapped with interpolation using vertical and horizontal neighbours using the pipelining technique.

The sorter unit contains binary comparator and swap units. Bitwise comparisons are performed, and the pre-computation logic is used to avoid unnecessary switching. Significant power reduction is achieved with negligible area overhead. Sorted array X is with four elements, and two centre values are to be added and divided by 2 for computing the median value. To reduce the hardware complexity, the maximum value or minimum value of the sorted array is duplicated to make the array size as 5. $X(3)$ is now chosen as the median value for X_{med} extra computations. Next, X_{med} is tested whether it is a noisy pixel (equal to 0 or 255) or not. If not, $X_{eff} = X_{med}$; otherwise, X_{eff} suitable representative value is computed as per the proposed algorithm.

The effective median value is used for interpolation; also, the relative distance between the pixels in the local window is computed, and the elements which are deviating much from the distance are also identified and replaced by X_{eff} . Hence, the proposed EWM computation not only performs scaling but also is used for detecting and removing the impulse noise. Figure 5 represents the modified image scaling algorithm.

3.3. EWMI Algorithm

Step 1: given input image of size $N \times N$, construct an array of size $2N \times 2N$

Step 2: label array elements in pixel positions as $a_{i,j}^{00}$, $x_{i,j}^{01}$, $x_{i,j}^{10}$, $x_{i,j}^{11}$, where $i, j = 1$ to 3, as shown in Figure 7

Step 3: replace elements labelled $a_{i,j}^{00}$ by the respective original pixel values

Step 4: define a 3×3 local window for each array element $x_{i,j}^{11}$, and perform the following:

- Read and sort the four diagonal neighbours in an ascending order
- Duplicate the max value into sorted array X to make the array size as X [5]
- Select the median value as the centre value $X = X_{med}$ [3]
- Define impulse noise values $I1 = 0$ and $I2 = 255$
- If $X_{med} \neq I1$ or $X_{med} \neq I2$, then $X_{eff} = X_{med}$, and go to "(g)"
- Else, compute $X_{eff} = (I1 + I2)/4$
- Find the relative difference between adjacent pixels ($d_i = X_i + 2 - X_{i+1}$) and choose $D_{max} = \max(d_i)$
- Replace $x_{i,j}^{11}$ by X_{eff} and its diagonal neighbours whose $D_i > D_{max}$

Step 5: define a 3×3 local window for each array element $x_{i,j}^{10}$ and

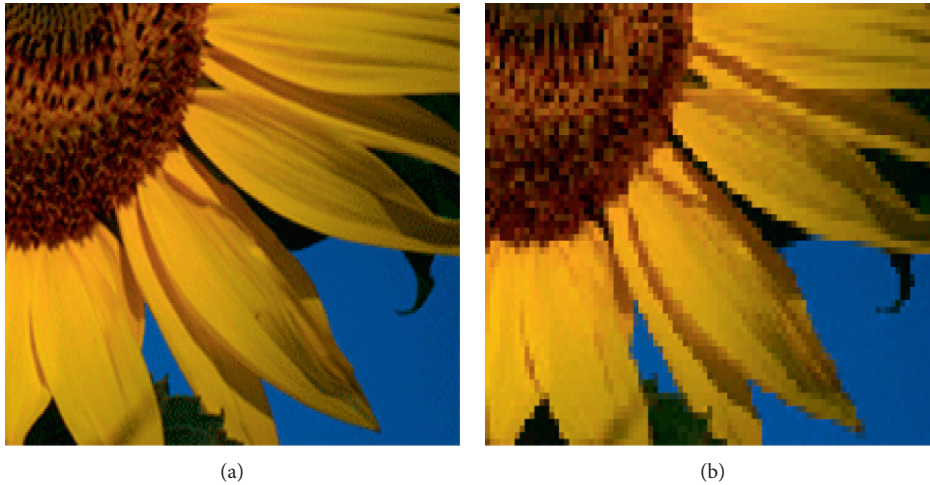


FIGURE 4: Scaled images in low and high resolutions. (a) 3 megapixels. (b) 1 megapixel.

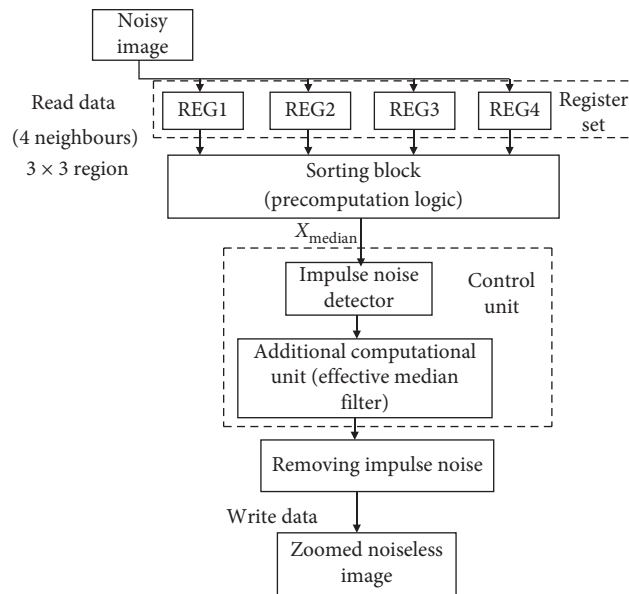


FIGURE 5: Block diagram of the modified image scaling algorithm.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}
 \begin{bmatrix} a_{1,1}^{00} & x_{1,1}^{01} & a_{1,2}^{00} & x_{1,2}^{01} & a_{1,3}^{00} & x_{1,3}^{01} \\ x_{1,1}^{10} & x_{1,1}^{11} & x_{1,2}^{10} & x_{1,2}^{11} & x_{1,3}^{10} & x_{1,3}^{11} \\ a_{2,1}^{00} & x_{2,1}^{01} & a_{2,2}^{00} & x_{2,2}^{01} & a_{2,3}^{00} & x_{2,3}^{01} \\ x_{2,1}^{10} & x_{2,1}^{11} & x_{2,2}^{10} & x_{2,2}^{11} & x_{2,3}^{10} & x_{2,3}^{11} \\ a_{3,1}^{00} & x_{3,1}^{01} & a_{3,2}^{00} & x_{3,2}^{01} & a_{3,3}^{00} & x_{3,3}^{01} \\ x_{3,1}^{10} & x_{3,1}^{11} & x_{3,2}^{10} & x_{3,2}^{11} & x_{3,3}^{10} & x_{3,3}^{11} \end{bmatrix}$$

FIGURE 6: Scaled array with labelled elements.

- (a) Apply the scale value of 0.6 to two horizontal neighbours
- (b) Read and sort the two horizontal and two vertical neighbours in an ascending order
- (c) Repeat steps from 4 (b) to 4 (h)

Step 6: define a 3 × 3 window for each array element $x_{i,j}^{10}, x_{i,j}^{01}$ and

- (a) Apply the scale value of 0.6 to two horizontal and two vertical neighbours
- (b) Read and sort the eight neighbours in an ascending order
- (c) Repeat steps from 4 (b) to 4 (h).

Using the proposed EWMI image scaling algorithm, interpolation and denoising can be done simultaneously to produce high-quality scaled images. For low-resolution images and images with blurring, defocusing and noise can be scaled with high visual quality using the proposed algorithm.

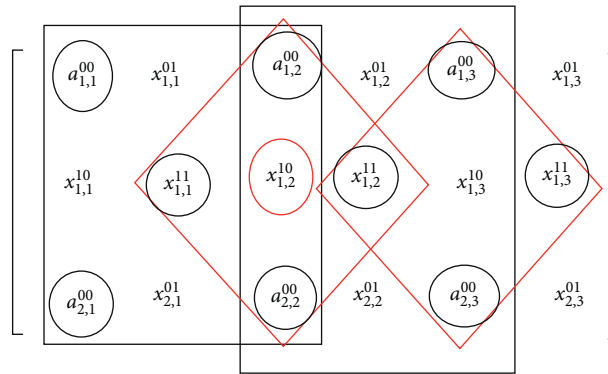


FIGURE 7: Pipelined processing of successive interpolations.

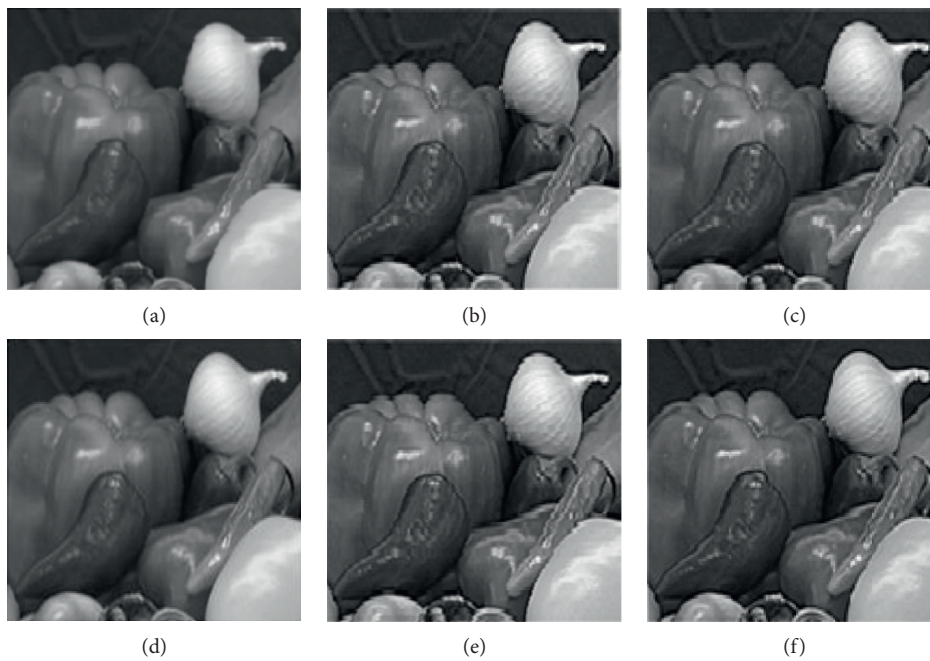


FIGURE 8: Comparison of existing and proposed techniques with degree of scaling 2 using digital natural images. (a) Bilinear interpolation [44]. (b) BI with the prefilter [41]. (c) High-boost filter [7]. (d) APNN [42]. (e) UTDCF [43]. (f) Proposed EWMI.

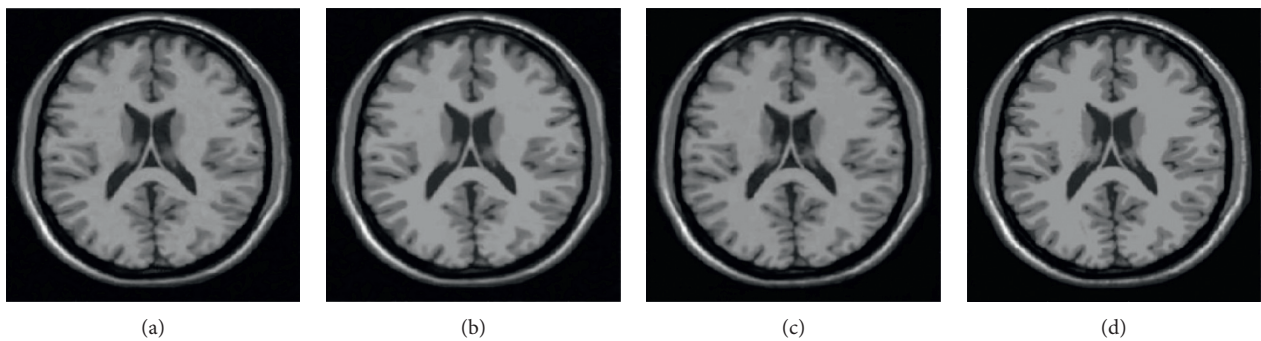


FIGURE 9: Continued.

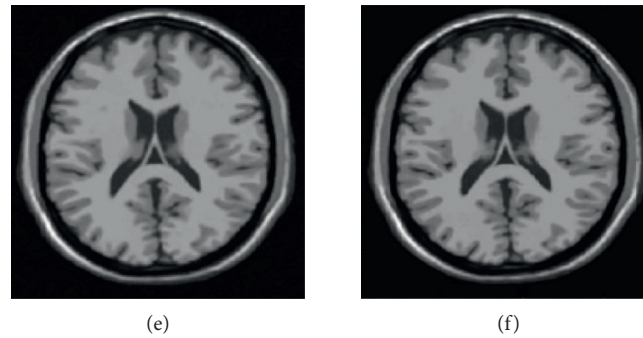


FIGURE 9: Various techniques are comparatively analyzed by using the digital MR image. (a) Bilinear interpolation [44]. (b) BI with the prefilter [41]. (c) High-boost filter [7]. (d) APNN [42]. (e) UTDCF [43]. (f) Proposed EWMI.

TABLE 1: Comparison of PSNR of existing and proposed image scaling techniques with scaling factor 2.

Image details	BI [44]	BI with the prefilter [41]	High-boost filter [7]	APNN [42]	UTDCF [43]	Proposed EWMI
High-resolution images (4 megapixels)						
Parrots	28.70	39.86	45.94	46.56	47.32	48.39
Flower	27.66	39.45	46.57	46.86	47.92	49.13
Vegetables	29.24	39.58	46.46	47.21	47.56	48.47
Low-resolution images (1 megapixel)						
Pepper	25.78	31.98	41.23	42.65	43.12	45.44
Camera man	25.93	32.34	40.74	43.23	44.19	45.67
Parrots	26.14	33.56	41.32	42.87	45.54	46.46
Images with impulse noise (0.4–0.7 variance)						
Parrots	22.45	28.34	39.11	40.04	41.38	42.33
Missile	22.11	28.45	38.76	39.56	42.95	43.67
Lena	21.67	27.99	39.10	40.17	41.93	43.56
Images with JPEG artifacts and blur						
Parrots	19.75	25.61	36.23	38.54	39.80	41.23
Flower	20.44	25.45	36.79	37.76	38.79	40.74
Vegetables	20.69	26.77	37.21	39.98	40.85	41.32

TABLE 2: Test image samples used for performance analysis.









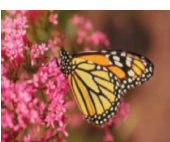
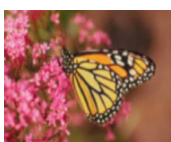
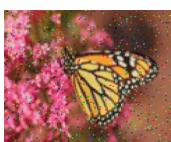
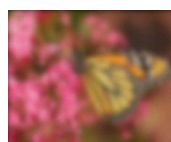
High resolution (4 megapixels)	Low resolution (1 megapixel)	Image with impulse noise	Image with JPEG artifacts and blur
			
			
			

TABLE 3: Comparison of computing resources and memory requirement.

Computing resources	Existing (BI with the prefilter) [41]	Proposed effective WMI
Registers	25	4
Multipliers	5	—
Adder and subtractor	35	1
Shifters	17	1
Comparator and swap units	—	5
Multiplexer	2	—
Line buffers	4	—

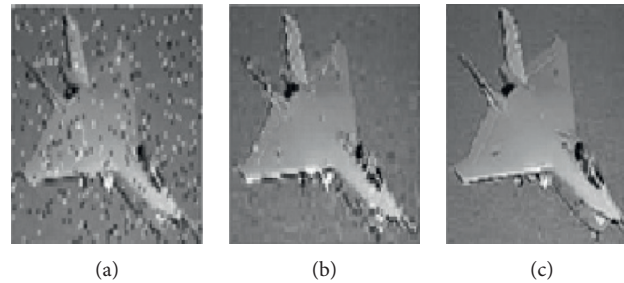


FIGURE 10: Comparison of existing and proposed techniques with degree of scaling 2. (a) Noisy image. (b) BI with the prefilter. (c) Proposed EWMI.

4. Simulation Results and Performance Analysis

Xilinx ISE design suite 13.2 tool has been used for implementing the VLSI architecture of the proposed effective WMI image scaling algorithm using Verilog HDL, and the MATLAB R2010b image processing tool box is used to verify the visual quality of the scaled images. The performance of scaled images of the proposed and existing techniques [7, 41–44] is analyzed in terms of PSNR. Image samples from the LIVE image quality database and real-time blur image database [45, 46] are used for the performance analysis. The real-time blur image database [46] contains 585 images with resolutions ranging from 1280×960 to 2272×1704 pixels. In this work, based on the resolution, two types of image samples with high- and low-resolution ranges of different sizes are taken, and scaling is performed.

A high-resolution image of size 256×256 is scaled into 512×512 . From Figures 8 and 9, it is confirmed that the edge details are well preserved and better in the proposed than the existing algorithms for both digital natural images and magnetic resonance (MR) image (medical data).

Table 1 represents the comparison of the PSNR value of existing and proposed image scaling techniques with the scale factor as 2. Table 2 highlights the test sample images. Table 3 describes the comparison of computing resources and memory requirement.

Table 1 gives PSNR values of scaled images. Some of the sample images from the LIVE image quality database [45] and real-time blur database [46] are given in Table 2. From Table 1, the proposed technique results have better quality (PSNR) than than the existing algorithms for various

resolutions, and the artifacts are effectively removed by the proposed algorithm. The performance is even better for noisy images. Figure 10 shows the image of size 128×128 with 0.4 impulse noise and the scaled images by existing and proposed scaling algorithms.

Table 3 gives details of computing resources used for the implementation of existing [41] and proposed architectures for image scaling. It is found that the proposed EWMI image scaling architecture is of low cost compared to the existing one [41]. It inherently removes the noise and is used for scaling operations, whereas in the existing techniques, separate filters are needed to preserve edge details. Hence, the proposed architecture is a low-cost, memory-efficient, and high-quality image scaling algorithm.

5. Conclusion

In this work, VLSI implementation of an effective WMI image scaling algorithm is proposed. The main contribution of this work is developing an effective weighted median technique capable of performing interpolation as well as denoising. As the degree of scaling increases, the proposed technique removes the blurring and preserving edge details compared to other existing techniques. The proposed work yields better performance with reasonable hardware complexity. In future, techniques to minimize the hardware complexity of effective WMI image scaling will be performed. The major limitation of the proposed image scaling algorithm is that it is used only for zooming and for scale down, and proper modifications will be done.

Data Availability

The data that support the findings of this study are available within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was funded by Dirección General de Investigaciones of Universidad Santiago de Cali (call no. 01-2021).

References

- [1] C. H. Chun-Ho Kim, S. M. Si-Mun Seong, J. A. Jin-Aeon Lee, and L. S. Lee-Sup Kim, "Winscale: an image-scaling algorithm using an area pixel model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 549–553, 2003.
- [2] S. Jain and D. Suresh, "A new Jpeg image scaling algorithm based on the area pixel model," *International Journal of Engineering Science*, vol. 2, pp. 46–56, 2013.
- [3] H. Hakran Kim, Y. Youngjoon Cha, and S. Seongjai Kim, "Curvature interpolation method for image zooming," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1895–1903, 2011.
- [4] C. C. Lin, M. H. Sheu, H. K. Chiang, W. K. Tsai, and Z. C. Wu, "Real-time FPGA architecture of extended linear convolution for digital image scaling," in *Proceedings of the IEEE International Conference on Field-Programmable Technology*, pp. 381–384, Taipei, Taiwan, December 2008.
- [5] C. C. Lin, M. H. Sheu, H. K. Chiang, C. Liaw, and Z. C. Wu, "The efficient VLSI design of BI-CUBIC convolution interpolation for digital image processing," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 480–483, Seattle, WA, USA, May 2008.
- [6] S. L. Shih-Lun Chen, H. Y. Hong-Yi Huang, and C. H. Ching-Hsing Luo, "A low-cost high-quality adaptive scalar for real-time multimedia applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1600–1611, 2011.
- [7] S. Safinaz, "An efficient algorithm for image scaling with high boost filtering," *International Journal of Scientific and Research Publications*, vol. 4, no. 5, pp. 1–9, 2014.
- [8] O. Wisesa, A. Adriansyah, and O. I. Khalaf, "Prediction analysis sales for corporate services Telecommunications company using gradient boost algorithm," in *Proceedings of the 2nd International Conference on Broadband Communications, Wireless Sensors and Powering*, pp. 101–106, Yogyakarta, Indonesia, September 2020.
- [9] X. Wang, J. Liu, O. Ibrahim Khalaf, and Z. Liu, "Remote sensing monitoring method based on BDS-based maritime joint positioning model," *Computer Modeling in Engineering and Sciences*, vol. 127, no. 2, pp. 801–818, 2021.
- [10] O. I. Khalaf, "Preface: smart solutions in mathematical engineering and sciences theory," *Mathematics in Engineering, Science and Aerospace (MESA)*, vol. 12, no. 1, pp. 1–4, 2021.
- [11] O. Wisesa, A. Adriansyah, and O. Ibrahim Khalaf, "Prediction analysis for business to business (B2B) sales of telecommunication services using machine learning techniques," *Majlesi Journal of Electrical Engineering*, vol. 14, no. 4, pp. 145–153, 2020.
- [12] S. Sengan, P. Vidya Sagar, O. Ibrahim Khalaf, and R. Dhanapal, "The optimization of reconfigured real-time datasets for improving classification performance of machine learning algorithms," *Mathematics in Engineering, Science and Aerospace (MESA)*, vol. 12, no. 1, 2021.
- [13] X. Xiang, Q. Li, S. Khan, and O. I. Khalaf, "Urban water resource management for sustainable environment planning using artificial intelligence techniques," *Environmental Impact Assessment Review*, vol. 86, Article ID 106515, 2021.
- [14] G. Suryanarayana, K. Chandran, O. I. Khalaf, Y. Alotaibi, A. Alsufyani, and S. A. Alghamdi, "Accurate magnetic resonance image super-resolution using deep networks and Gaussian filtering in the stationary wavelet domain," *IEEE Access*, vol. 9, Article ID 71406, 2021.
- [15] G. Li, F. Liu, A. Sharma et al., "Research on the natural language recognition method based on cluster Analysis using neural network," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9982305, 13 pages, 2021.
- [16] S. Dalal and O. I. Khalaf, "Prediction of occupation stress by implementing convolutional neural network techniques," *Journal of Cases on Information Technology*, vol. 23, no. 3, pp. 27–42, 2021.
- [17] S. Sengan, G. R. Koteswara Rao, O. Ibrahim Khalaf, and M. Rajesh Babu, "Markov mathematical analysis for comprehensive real-time data-driven in healthcare," *Mathematics in Engineering, Science and Aerospace (MESA)*, vol. 12, no. 1, 2021.
- [18] A. Tuan Hoang, X. Phuong Nguyen, O. Ibrahim Khalaf et al., "Thermodynamic simulation on the change in phase for carburizing process," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 1129–1145, 2021.
- [19] C. A. Tavera Romero, D. F. Castro, J. H. Ortiz, O. I. Khalaf, and M. A. Vargas, "Synergy between circular economy and industry 4.0: a literature review," *Sustainability*, vol. 13, no. 8, p. 4331, 2021.
- [20] T. Xuan Tran, X. Phuong Nguyen, D. Nam Nguyen et al., "Effect of poly-alkylene-glycol quenchant on the distortion, hardness, and microstructure of 65Mn steel," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3249–3264, 2021.
- [21] S. Rajasoundaran, A. V. Prabu, J. B. V. Subrahmanyam et al., "Secure watchdog selection using intelligent key management in wireless sensor networks," *Materials Today: Proceedings*, vol. 47, 2021.
- [22] C. A. Tavera, J. H. Ortiz, O. I. Khalaf, D. F. Saavedra, and T. H. H. Aldhyani, "Wearable wireless body area networks for medical applications," *Computational and Mathematical Methods in Medicine*, vol. 9, 2021.
- [23] G. M. Abdulsahib and O. I. Khalaf, "An improved cross-layer proactive congestion in wireless networks," *International Journal of Advances in Soft Computing and Its Applications*, vol. 13, no. 1, pp. 178–192, 2021.
- [24] G. M. Abdulsahib and O. I. Khalaf, "Accurate and effective data collection with minimum energy path selection in wireless sensor networks using mobile sinks," *Journal of Information Technology Management*, vol. 13, no. 2, pp. 139–153, 2021.
- [25] O. I. Khalaf and G. M. Abdulsahib, "Optimized dynamic storage of data (ODSD) in IoT based on blockchain for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 14, 2021.
- [26] A. F. Subahi, Y. Alotaibi, O. I. Khalaf, and F. Ajesh, "Packet drop battling mechanism for energy aware detection in wireless networks," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 2077–2086, 2020.

- [27] O. I. Khalaf and G. M. Abdulsahib, "Energy efficient routing and reliable data transmission protocol in WSN," *International Journal of Advances in Soft Computing and Its Applications*, vol. 12, no. 3, pp. 45–53, 2020.
- [28] O. I. Khalaf, G. M. Abdulsahib, and B. M. Sabbar, "Optimization of wireless sensor network coverage using the bee algorithm," *Journal of Information Science and Engineering*, vol. 36, no. 2, pp. 377–386, 2020.
- [29] O. I. Khalaf, K. A. Ogudo, and M. A. Singh, "Fuzzy-based optimization technique for the energy and spectrum efficiencies trade-off in cognitive radio-enabled 5G network," *Symmetry*, vol. 13, no. 1, p. 47, 2021.
- [30] O. I. Khalaf, F. Ajesh, A. A. Hamad, G. N. Nguyen, and D.-N. Le, "Efficient dual-cooperative bait detection scheme for collaborative attackers on mobile ad-hoc networks," *IEEE Access*, vol. 8, Article ID 227962, 2020.
- [31] A. Abdullah Hamad, A. S. Al-Obeidi, E. H. Al-Taiy, O. Ibrahim Khalaf, and D. Le, "Synchronization phenomena investigation of A new nonlinear dynamical system 4-D by gardano's and lyapunov's methods," *Computers, Materials & Continua*, vol. 66, no. 3, pp. 3311–3327, 2021.
- [32] O. Ibrahim Khalaf, M. Sokiyna, Y. Alotaibi, A. Alsufyani, and S. Alghamdi, "Web attack detection using the input validation method: dpda theory," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 3167–3184, 2021.
- [33] C. A. T. Romero, J. H. Ortiz, O. I. Khalaf, and A. R. Prado, "Web application commercial design for financial entities based on business intelligence," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3177–3188, 2021.
- [34] A. A.R. Alkhafaji, N. Nur Amir Sjarif, M. A. Shahidan et al., "Payload capacity scheme for quran text watermarking based on vowels with kashida," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3865–3885, 2021.
- [35] S. K. Prasad, J. Rachna, O. I. Khalaf, and D.-N. Le, "Map matching algorithm: real time location tracking for smart security application," *Telecommunications and Radio Engineering*, vol. 79, no. 13, pp. 1189–1203, 2020.
- [36] E. Nabil Al-Khanak, S. Peck Lee, S. Ur Rehman Khan et al., "A heuristics-based cost model for scientific workflow scheduling in cloud," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3265–3282, 2021.
- [37] H. Zhao, P. L. Chen, S. Khan, and O. I. Khalafe, "Research on the optimization of the management process on internet of things (IoT) for electronic market," *The Electronic Library*, vol. 38, 2020.
- [38] M. Krichen, S. Mechti, R. Alroobaea et al., "A formal testing model for operating room control system using internet of things," *Computers, Materials & Continua*, vol. 66, no. 3, pp. 2997–3011, 2021.
- [39] M. Javed Awan, M. Shafry Mohd Rahim, H. Nobanee, A. Yasin, O. Ibrahim Khalaf, and U. Ishfaq, "A big data approach to black Friday sales," *Intelligent Automation & Soft Computing*, vol. 27, no. 3, pp. 785–797, 2021.
- [40] X. Zheng, F. Ping, Y. Pu, Y. Wang, C. E. Montenegro-Marin, and O. I. Khalaf, "Recognize and regulate the importance of work-place emotions based on organizational adaptive emotion control," *Aggression and Violent Behavior*, vol. 14, Article ID 101557, 2021.
- [41] S.-L. Chen, "VLSI implementation of a low-cost high-quality image scaling processor," *IEEE Transactions on circuits and systems, express briefs*, vol. 60, no. 1, 2013.
- [42] C.-H. Chen, H.-W. Chang, and C.-M. Kuo, "VLSI implementation of anisotropic probabilistic neural network for real-time image scaling," *Journal of Real-Time Image Processing*, vol. 16, no. 1, pp. 71–80, 2019.
- [43] W. He, J. Zhang, Y. Lin et al., "A low-cost high-speed object tracking VLSI system based on unified textural and dynamic compressive features," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, 2020.
- [44] P. A. Dilip, K. Rameshbabu, K. P. Ashok, and S. A. Shivdas, "Bilinear interpolation image scaling processor for VLSI architecture," *International Journal of Reconfigurable and Embedded Systems*, vol. 3, no. 3, 2014.
- [45] H. R. Sheikh, Z. Wang, L. K. Cormack, and A. C. Bovik, "LIVE image quality assessment database release 2," <http://live.ece.utexas.edu/research/quality/subjective>.
- [46] "BID-blurred image database," <http://www.lps.ufrj.br/profs/eduardo/>.