Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

An improvement of efficient nonlinear color image interpolator: Theory and FPGA implementation

Anis Ridha Boudabbous *, Marwa Jomaa Graja

Department of Computer Engineering and Networks, College of Computer and Information Sciences, Jouf University, Sakakah, Saudi Arabia

ARTICLE INFO

Article history:
Received 6 January 2022
Received in revised form
17 April 2022
Accepted 3 November 2022

Keywords:
Nonlinear interpolator
Color image
Edge preserving
Vector rational function
FPGA

ABSTRACT

In this paper, a new proposal for a nonlinear edge-preserving interpolator and hardware implementation is presented. As a new idea for a color image interpolator, our proposal is focusing on the interpolated pixel and we tried to adjust it in order to have better quality. To evaluate the effectiveness of the proposed idea for preserving images, we implemented it using different color images and we evaluated using different evaluation measurements. Then, we compared our new proposal with the traditional nonlinear Edge preserving interpolator. The obtained results confirm that our proposed Edge preserving is better than the old interpolator. It also demonstrates consistent image quality performance among a variety of images. The hardware implementation based on FPGA shows that we are able to gain image quality without increasing the size of the circuit once implemented in hardware. We show that our proposed interpolator for Edge preserving improves considerably the image quality and represents a fast solution when implemented in hardware. Despite a small increase in FPGA resources, we obtain an average improvement of the image quality of about 35.75% using the NCD metric.

1. Introduction

Interpolation is a value estimation process between two pre-existing pieces of information. It is used in scanners, digital devices, and image editing software to produce an image with a resolution (the number of vertical and horizontal pixels) greater than that captured by the sensor. Interpolation "invents" pixels where there were none and inserts them between two originals to increase the image size. Generally, the major problem with interpolation is that it can greatly increase an image defect by enlarging it (Han, 2013; Getreuer, 2011).

The simplest type of interpolation is linear interpolation, which consists of joining the given points. It can be used to estimate the values between the table data. This method is quick and easy but it lacks precision. The estimation error shows that linear interpolation is not very accurate. Therefore, it is important to focus on the nonlinear interpolator which gives better performances.

However, in the literature, Bicubic interpolation is usually considered the best method, yet it may not produce the best results (Han, 2013). In general, interpolation works better when applied to a large amount of information. An image with a small number of pixels will show more degradation traces than an initially richer image. In image processing, interpolation consists of calculating a pixel according to its neighboring pixels (Fig. 1). This technique is used during an image transformation loss.

The transformations at the image level are described by a 3x3 homogeneous matrix. For each pixel (i_n, j_m), of the destination image, given (n,m) the dimensions of the image, a position (x,y) is calculated in the source image. Then the new destination pixel value is calculated according to the vicinity of the source position. Then, we determine a value by interpolation with the neighboring pixels (Han, 2013; Getreuer, 2011). To reduce the crenellation (aliasing), p(i_2, j_2) is calculated by interpolation. Interpolation produces a grayscale (or color) image.

In our research, we will focus exclusively on the function of interpolation in the context of color image resolution. The development of image

* Corresponding Author.
Email Address: aboudabbous@ju.edu.sa (A. R. Boudabbous)
https://doi.org/10.21833/ijaas.2023.02.014
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-5488-2382

interpolators for multidimensional data processing has great importance for various applications.

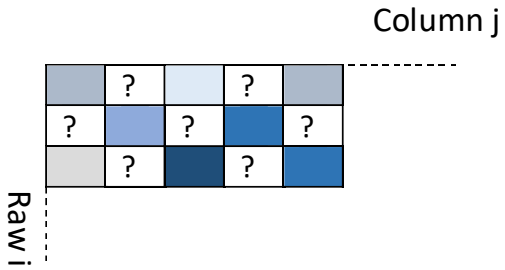

Fig. 1: Pixels loss during image transformation

This paper is organized as follows. Section 2 presents an overview of the different adaptive interpolation algorithms. Section 3 presents an outline of the edge-preserving interpolator. Section 4 presents our proposal as a new idea to improve the existing Edge-Preserving Interpolator presented by Cheikh et al. (1998). In section 5, we present the validation results of the Edge-Preserving Interpolator tested using some standard images and a comparison with a similar algorithm offered by Cheikh et al. (1998). Section 6 presents the hardware implementation results using Intel/Altera platform by means of Quartus II and a comparison with the implementation of the old one (Cheikh et al., 1998). Finally, conclusions are drawn in section 7.

2. Interpolation algorithms overview

In the literature, many interpolation algorithms have been proposed. We can cite, the special case of Nearest-neighbor interpolation (Han, 2013). This algorithm is the simplest one. It is fast and is the only one that does not add any new color by interpolating the image. To calculate the color of an interpolated pixel, it simply takes the color of the nearest pixel in the original image. This technique does not give very satisfactory results and produces a lot of block effects since, for example, we double the size of an image, and each pixel is in fact recopy 4 times (Han, 2013).

In the same context, most interpolation algorithms use a convolution kernel like the interpolation with splines. It takes a certain number of neighbors of the pixel to be calculated and makes an average weighted according to their distance to the interpolated pixel to obtain the result. Since computer images are often represented by matrices, interpolation can easily be applied by representing its convolution core by a matrix. Thus, to apply the interpolation, it is necessary to multiply for each pixel, the matrix of the pixel and its neighbors by the matrix of the convolution kernel to obtain the required pixel (Getreuer, 2011). This interpolator has been improved in the Bilinear interpolation (B-Spline Order 2) version. The bilinear interpolation takes into account the 4 neighbors of the pixel (called Von Neumann neighborhood) that is calculated, it applies a bilinear function (so order 2) on the 4 neighbors and then applies the result on the interpolated pixel. This technique gives good results than the Nearest-neighbor technique. It allows to obtain of smooth contours but also produces a small pixilation effect. So, the Bicubic interpolation (B-Spline Order 4) partially solves this problem. The Bicubic interpolation works on the same principle as the bilinear interpolation, but takes into account the 8 neighbors of the pixel (called Moore's neighborhood) and applies a Bicubic function (so order 4). Like bilinear interpolation, it produces smooth contours, but the effect of pixilation is less compared to lower-order interpolations (Han, 2013; Getreuer, 2011).

Other techniques used for interpolation like Bell interpolation (Abdullah et al., 2018), and Hermite interpolation (Seta et al., 2009) were mainly focused on the speed constraint. Indeed, Bell interpolation provides a smooth image with continuous contours. It is moreover relatively fast. And Hermit interpolation allows for keeping a sharpness for the image, even if it depends a lot on the processed image. There is however a slight pixilation effect. This algorithm is one of the fastest. Like the Hermite interpolation, the Mitchell interpolation (Parsania and Virparia, 2016) is a special case of the Bicubic interpolation. It is also fast but produces a smoother image, with continuous contours. The Lanczos interpolation algorithm is one which provides the best results. It gets very sharp images, but it is very slow to run. In addition, it can sometimes produce artifacts in the resulting image. There are many others algorithms often used for the re-sampling of an image. It must therefore choose the algorithm that best corresponds to what you want to obtain, according to the operation carried out on the considered image, the speed at which you want to process the image, and the desired quality for the output image (Fadnavis, 2014).

Completely different methods from the previous ones performing adaptive interpolations, in order to preserve the objects, present in the scene and their contours in particular. These techniques often control interpolation by contour detection or image segmentation. An example of such techniques is given by Khan et al. (2020). Other methods rely on fractal decompositions that allow the fine properties interpolated by the original sampling to be found (Kok and Tam, 2019). Another very interesting color image interpolator has appeared (Jha et al., 2014) to minimize all non-preferred effects cited in the preview paragraphs called edge-preserving interpolator. This interpolator is based on a vector rational function to perform the interpolated pixel.

3. Edge-preserving interpolator concept

The interpolation technique presented in this paper is an Edge-Preserving Interpolator for a color image. The interpolation of the color image acts in two ways and attempts to get a better estimation

concerning the color and intensity of the pixel, built on the values of surrounding pixels.

The interpolator works with four reduced multivariate data samples a, b, c, and d (3×3 mask), in order to recreate the absent sample x in the central location. In this interpolation technique, the masks presented in Fig. 2 are applied to perform the new pixels (Jha et al., 2014; Lu et al., 2012).

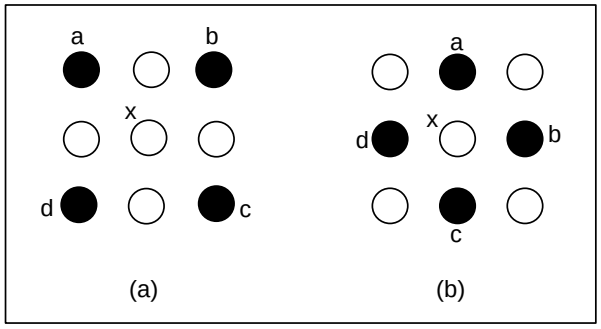

Fig. 2: A 3x3 mask applied in the interpolator

The standardized value of x satisfies two conditions. The first one, alone weight is a positive quantity. The second one is in a smooth region (where adopting that all four neighbors of the central pixel have the same value). The addition of the weights is equivalent to 1, which certifies that the output is fair. By means of Eq. 1 the value of the new pixel (output of the interpolation) can be performed:

$$x = \frac{\sum_{\{u\neq v;u,v\in(a,b,c,d)\}} W_{uv}.(u+v)}{2\sum_{\{u\neq v;u,v\in(a,b,c,d)\}} W_{uv}} = \frac{1}{W}\left(\frac{w_{ab}+w_{ac}+w_{da}}{2}a + \frac{w_{ab}+w_{bc}+w_{bd}}{2}b + \frac{w_{ac}+w_{bc}+w_{cd}}{2}c + \frac{w_{bd}+w_{cd}+w_{da}}{2}d\right) \quad (1)$$

The overhead equation defines how to perform pixel 'x' value by means of identifying a, b, c, and d pixels. Where $W = w_{ab} + w_{bc} + w_{cd} + w_{da} + w_{ac} + w_{bd}$ based on vector-rational function. The weights are calculated as follows:

$$W_{uv} = \frac{1}{12+k\|u-v\|} \quad (2)$$

The rational function given by Eq. 2, is used to perform the weights W_{uv}, which defines the l_1 norm. The k factor is a user-defined parameter subject to the request or application. Referring to Cheikh et al. (1998), the value 0.025 is selected. This interpolation process applies a weighted average of recognized pixels. This fact is due to the use of norm, which requires the computation of the rational function in Eq. 3.

$$N_{uv} = \|u-v\| = (R_u - R_v)^2 + (G_u - G_v)^2 + (B_u - B_v)^2 \quad (3)$$

This procedure of this interpolation validates the conclusion that if a pixel differs from another pixel only in one component, it will not be used in the pixel x computation, consequently, they do not attack an additional part of the image. The focal goal evoked is to increase the quality of the output image. Therefore, this interpolation technique involves many calculations as long as the costly calculation of the rational function (Eq. 3).

4. Improved edge-preserving interpolator

Our highest goal is to get an effective interpolation and compare it with the old known Edge Preservation Interpolator (Cheikh et al., 1998; Lu et al., 2012). Our goal is to clearly show an improvement of the new idea used in the Edge Preservation Interpolator solution compared to the old version of this interpolator. As a new idea for this color image interpolator, it acted on the interpolated pixel and tried to adjust it in order to have better quality. The adjustment of the new interpolated pixels is done by the following Eq. 4.

$$X_{int\_new} = (X_{zommed} + X_{int})/2 \quad (4)$$

where, in our case, X_{zoomed} is selected as the modified pixel from the window after zooming the image. X_{int} presents the pixel interpolated with Eq. 1. This new idea can be described in Fig. 3.

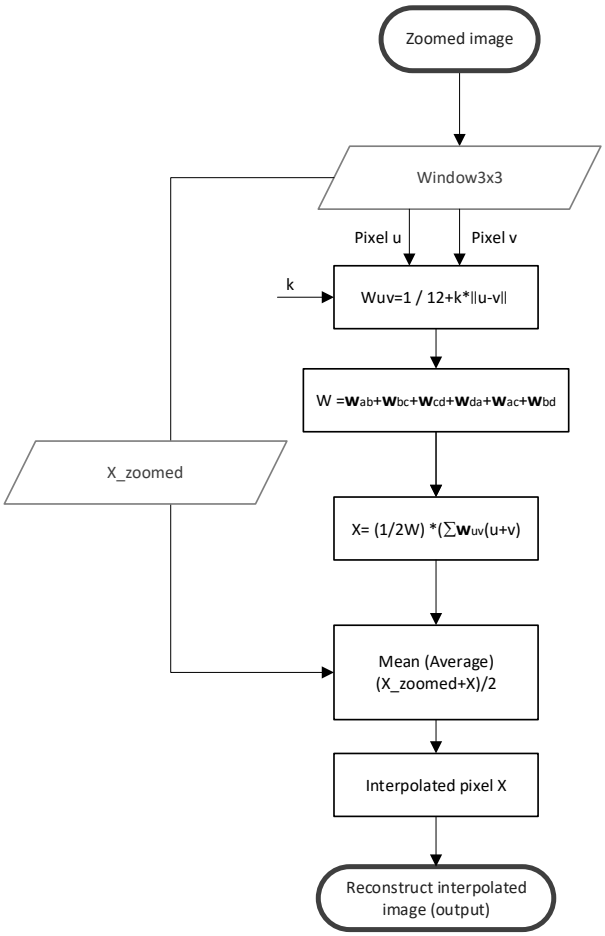

Fig. 3: Flowchart of the new proposed interpolator

5. Results and discussion

In this section, the old version of the interpolator and the improved version, are compared. Our main objective is to obtain an efficient improvement in the color image using our new idea concerning the edge-

preserving interpolator. Also, our goal is to demonstrate an improvement in the quality of the interpolated images and compare the obtained results with the old interpolator in the same conditions. To evaluate the effectiveness of the proposed idea for preserving image details, a comparison is done between the old interpolator and the improved version, using some test images. To compare the images, we use three different measurements MAE, MSE, and NCD (Lukac et al., 2005). Let i and j be the position indices of the sample, the Mean Absolute Error (MAE) is definite as:

$$MAE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|Z_{i,j} - I_{i,j}| \quad (5)$$

where, $Z_{i,j} - I_{i,j} = (R_{i,j}^{Z} - R_{i,j}^{I}) + (G_{i,j}^{Z} - G_{i,j}^{I}) + (B_{i,j}^{Z} - B_{i,j}^{I})$ (6)

The MSE means squared error, defined as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\|Z_{i,j} - I_{i,j}\|_{2} \quad (7)$$

For both cases in the MAE and MSE measures, $(R_{i,j}^{Z}, R_{i,j}^{I}), (G_{i,j}^{Z}, G_{i,j}^{I})$, and $(B_{i,j}^{Z}, B_{i,j}^{I})$ represent the RGB color components respectively of the zoomed frame and interpolated frame. The NCD means normalized color difference measurements for color images. These metrics are defined in the RGB color space. The NCD (defined in the *L*a*b** color space) is commonly used to measure color conservation. To determine the NCD measure, the image must be transformed to the *L*a*b** color space. The error among color vectors:

$$\triangle E = \left[(L_{i,j}^{Z} - L_{i,j}^{I})^2 + (a_{i,j}^{Z} - a_{i,j}^{I})^2 + (b_{i,j}^{Z} - b_{i,j}^{I})^2\right]^{1/2} \quad (8)$$

This error is hired to estimate the NCD quantity:

$$NCD = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\triangle E^{1/2}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[(L_{i,j}^{Z})^2 + (a_{i,j}^{Z})^2 + (b_{i,j}^{Z})^2\right]^{1/2}} \quad (9)$$

where, $\left[(L_{i,j}^{Z})^2 + (a_{i,j}^{Z})^2 + (b_{i,j}^{Z})^2\right]^{1/2}$ is the magnitude of the uncorrupted image (original) pixel vector in the *L*a*b** color space and M and N are the image sizes. The *NCD* metric expresses the measure of color distortion. Table 1 resumes the obtained results of the old and the new rational interpolator tested on five different images of size 151×78.

From Table 1, we obtain an average improvement of 49.35% in MAE and 74.80% in MSE, and 35.75% in the NCD metric. The closest and most important metric to the human eye is the NCD. Then we draw in the following Fig. 4, the NCD comparison between the old interpolator and the new one. NCD comparison between old and new interpolator. As a result of the simulations, we obtained a clear improvement and encouraged results using the proposed interpolator. In Fig. 5, we can clearly view the amelioration of the proposed interpolator compared to the old one.

Table 1: Comparison of obtained results using test image of size 151×78

Image	Old Interpolator (Cheikh et al., 1998)			New Interpolator (proposed)		
	MAE	MSE	NCD	MAE	MSE	NCD
Tree	26.92	658.16	0.0356	13.56	165.30	0.0219
Bird	31.80	849.60	0.0357	16.00	213.26	0.0222
Boat	17.72	266.19	0.0286	08.95	67.33	0.0180
House	09.25	84.49	0.0199	04.85	21.88	0.0139
Flower	11.50	119.50	0.0217	05.86	30.63	0.0149

6. Hardware implementation using FPGA

A hardware implementation using FPGA for the old and the new interpolator is performed. The synthesis targeted the Altera FPGA was made using the Altera Quartus II tool. Table 2, illustrates the hardware spent in terms of ALUTs (Adaptive Look-Up Tables), and DSP block and Input-output pins in Stratix EP2S60F672C3 FPGA.

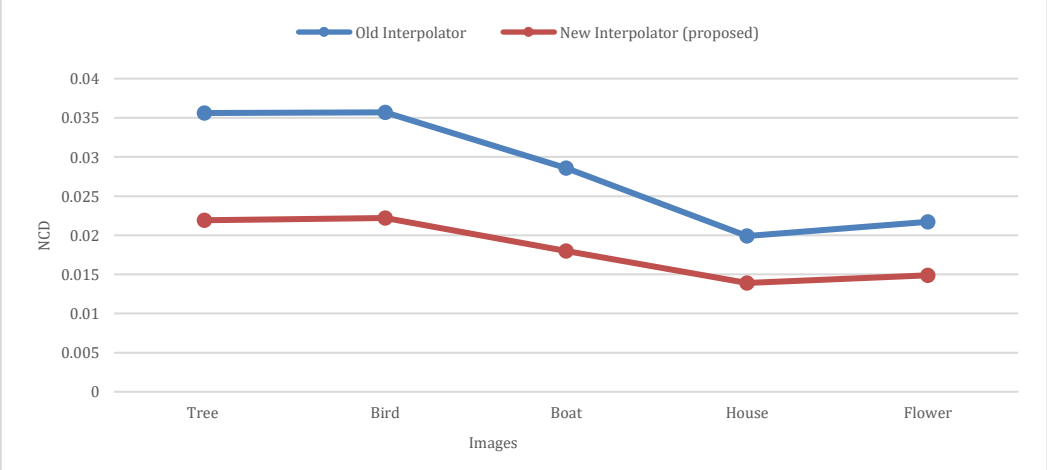

Fig. 4: NCD comparison between old and new interpolator

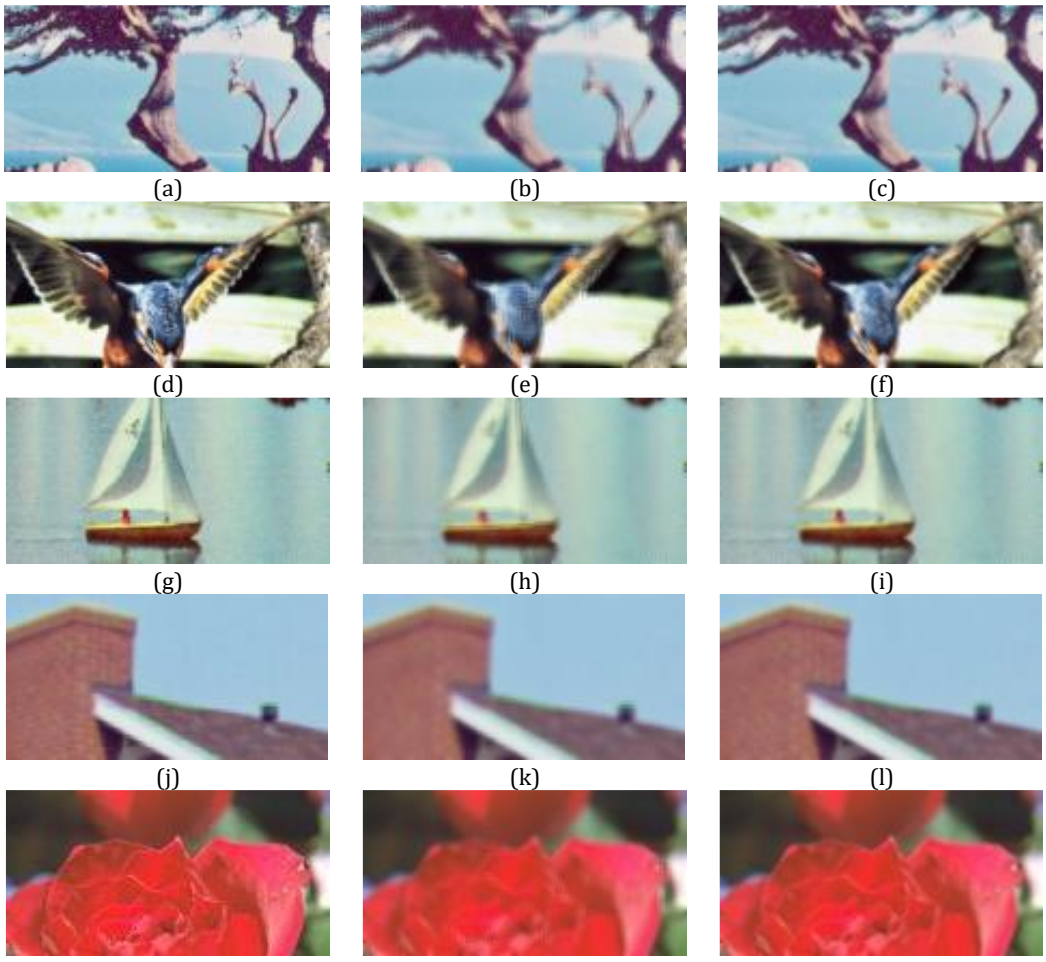

Fig. 5: (a, d, g, j, m) Original zoomed images, (b, e, h, k, n) Interpolated image using old Edge-Preserving Interpolator (c, f, i, l, o) Interpolated image using New Edge-Preserving Interpolator

Table 2: The implementation results in Stratix II FPGA

Resources	Old interpolator	New interpolator (Proposed)
ALUTs	967 / 48,352 (≈2 %)	985 / 48,352 (2 %)
Pins	148 / 493 (30 %)	148 / 493 (30 %)
DSP block	129 / 288 (45 %)	136 / 288 (47 %)

The HW architecture, described in VHDL, of the new interpolator, achieves 47% of DSP and 30% of Pins. Only 2% of ALUTs are used. Compared to the old version of the interpolator, a few LUTs and DSPs are consumed. But this difference is very negligible. So it is able to gain image quality without increasing the size of the circuit once implemented in hardware.

7. Conclusion

The methodology for developing an improvement of the Edge-Preserving interpolator is described. In this way, our work is essential to study the theory and the software implementation approach of the Edge-Preserving Interpolator. We tested on several images already used in global test banks. As a result of the simulations, we obtained an outstanding improvement in the interpolation quality. we obtain an average improvement of 35.75% using the NCD metric. The work can be improved by inserting this interpolator into a complete image processing chain. As a limitation, the Edge preserving interpolator requires much processing time computation. This is due to the expensive computation of the vector-rational function given by the W_{uv} equation.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Abdullah D, Fajriana F, Maryana M, Rosnita L, Siahaan APU, Rahim, R, and Hadikurniawati W (2018). Application of interpolation image by using bi-cubic algorithm. Journal of Physics: Conference Series, 1114: 012066. https://doi.org/10.1088/1742-6596/1114/1/012066

Cheikh FA, Khriji L, Gabbouj M, and Ramponi G (1998). Color image interpolation using vector rational filters. In the SPIE Conference, Nonlinear Image processing IX, International Society for Optics and Photonics, San Jose, USA, 3304: 242-249. https://doi.org/10.1117/12.304604

Fadnavis S (2014). Image interpolation techniques in digital image processing: An overview. International Journal of Engineering Research and Applications, 4(10): 70-73.

Getreuer P (2011). Linear methods for image interpolation. Image Processing On Line, 1: 238-259. https://doi.org/10.5201/ipol.2011.g_lmii

Han D (2013). Comparison of commonly used image interpolation methods. In the 2nd International Conference on Computer Science and Electronics Engineering, Atlantis Press, Hangzhou, China, 10: 1556-1559. https://doi.org/10.2991/iccsee.2013.391

Jha AK, Kumar A, Schaefer G, and Ahad MAR (2014). An efficient edge preserving image interpolation algorithm. In the 2014 International Conference on Informatics, Electronics and Vision, IEEE, Dhaka, Bangladesh: 1-4. https://doi.org/10.1109/ICIEV.2014.6850820 **PMCid:PMC3952416**

Khan S, Lee DH, Khan MA, Siddiqui MF, Zafar RF, Memon KH, and Mujtaba G (2020). Image interpolation via gradient correlation-based edge direction estimation. Scientific Programming, 2020: 5763837. https://doi.org/10.1155/2020/5763837

Kok CW and Tam WS (2019). Fractal image interpolation: A tutorial and new result. Fractal and Fractional, 3(1): 7. https://doi.org/10.3390/fractalfract3010007

Lu Y, Cai X, Zhai Z, and Qin X (2012). An image interpolation method with edge-preserving. In: Qian Z, Cao L, Su W, Wangn T, and Yang H (Eds.), Recent advances in computer science and information engineering: 237-242. Springer, Berlin, Germany. https://doi.org/10.1007/978-3-642-25792-6_36

Lukac R, Plataniotis KN, Venetsanopoulos AN, and Smolka B (2005). A statistically-switched adaptive vector median filter. Journal of Intelligent and Robotic Systems, 42(4): 361-391. https://doi.org/10.1007/s10846-005-1730-2

Parsania PS and Virparia PV (2016). A comparative analysis of image interpolation algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 5(1): 29-34. https://doi.org/10.17148/IJARCCE.2016.5107

Seta R, Okubo K, and Tagawa N (2009). Digital image interpolation method using higher-order Hermite interpolating polynomials with compact finite-difference. In the APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, Sapporo, Japan: 406-409.