International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7843-7855
© Research India Publications
http://www.ripublication.com

Interpreting Low Resolution Images Using Edge Interpolation Techniques: A Comparative Analysis

[1]Tarun Gulati, [2]Kapil Gupta
Dept. of Electronics & Communication
M.M. University, Mullana
Ambala, India
[1]gulati_tarun@rediffmail.com, [2]kapil_mbm@yahool.com
Dushyant Gupta
Dept. of Electronics & Communication
G. J. University, Hisar
Hisar, India
gupty2tank@yahoo.co.uk

Abstract

Images with high resolution and fine details are always desired and required in many visual tasks. For this purpose, the interpolation functions and/or methods are used in post-processing of images. The capability to digitally interpolate images to higher resolution with good image quality depends upon the selected interpolation function/method. The interpolated image quality is usually assessed in terms of sharpness of edges and freedom from artefacts. In addition, the ease of computation is also an important factor. This paper focuses on edge-directed interpolation approaches. A large number of edge-directed interpolation (EDI) methods have been developed to date; however, a review of key EDI methods is presented below.

Index Terms: NEDI, INEDI, ICBI, COSO, EDI

Introduction

A method for digitally interpolating images to higher resolution using edge direction consists of two phases: rendering and correction. The rendering phase is edge-directed. In this approach, from the low resolution image data, a high resolution edge map is generated by first filtering with a rectangular center-on-surround off (COSO) filter and then piecewise linear interpolation between the zero crossings is performed in the filter output. The rendering phase is based on bilinear interpolation modified to prevent interpolation across edges, as determined from the estimated high resolution edge map. During the correction phase, the mesh values (on which the rendering is based) is modified and predicted by a sensor model operating on the high resolution output of the rendering phase. The overall process is repeated iteratively. Figure 1.1 shows the structure of the edge-directed interpolation algorithm.

The procedure depicted in Fig. 1.1 can be described by the following equations:

$$y_k[n] = \mathcal{R}(\tilde{x}_k[m]) \tag{1.1}$$

$$\hat{x}_k[n] = S(y_k[n]) \tag{1.2}$$

$$\tilde{x}_{k+1}[m] = \tilde{x}_k[m] + \lambda(\hat{x}_k[m] - x[m]) \tag{1.3}$$

Where the points on the low and high resolution lattices are denoted by m and n respectively, the true sensor data by $x[m]$, the corrected sensor data by $\tilde{x}[m]$, the edge-directed rendering step by the operator $\mathcal{R}$**,** the interpolated image by $y[n]$, the sensor model by the operator S, and the estimated sensor data by $\hat{x}[m]$. It is to be noted that the sensor model S is a simple block average of the high resolution pixels in the unit cell for each pixel in the low resolution lattice. In addition, it is assumed that the term k denote the iteration index and, λ is a constant that controls the gain of the correction process. The iteration starts with the initial condition $\tilde{x}_0[m] = x[m]$.

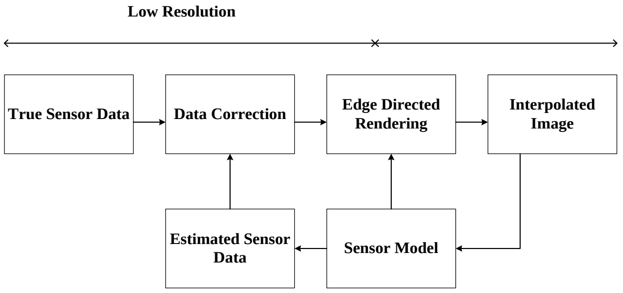


Figure 1.1: Structure of edge-directed interpolation algorithm

The equations 1.1 to 1.3 can also be represented as classical successive approximation procedure. The value of λ depends on computing the probability of edges in a region, and is computed as follows.

Firstly, the COSO filter (mentioned above and as shown in Fig. 1.2) is used to estimate the sub-pixel edge map. This filter has a constant positive center region embedded within a constant negative surround region. This filter mimics the point spread function for the Laplacian-of-Gaussian (LOG) and requires five additions/subtractions and one multiplication per output point when recursively implemented with row and column buffers.

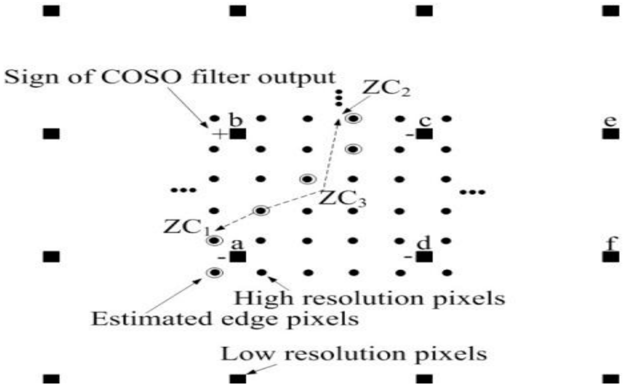


Figure 1.2: Lattices for edge-directed interpolation

To determine the high resolution edge map, the COSO filter output between points on the low resolution lattice is linearly interpolated to estimate zero-crossing positions on the high resolution lattice. This is done as follows. For each group of 4 low resolution pixels (e.g. points *a, b, c* and *d* in Fig. 1.2), the sign of COSO filter output at each point is examined. For example, for the case shown in Fig. 1.2**,** the interpolation is done along the lines *a - b, b - c,* and *b - d* to obtain the estimated zero-crossing points zc_1, zc_2, and zc_3. The edge is then approximated within the low resolution cell bounded by the points *a*, *b*, *c* & *d* by the two line segments zc_1 - zc_3 and zc_3 - zc_2. The obtained curve is then quantised to the high resolution lattice, yielding the estimated edge pixels, as shown in Fig. 1.3. The remaining sign geometries are treated similarly. In each case, the edge contour is approximated by two adjoining line segments, and then quantized to the high resolution lattice.

In the next stage, the low resolution image is first pre-processed, as shown in Fig. 1.3. This is necessary to mitigate the effect of errors in the estimated high resolution edge map.

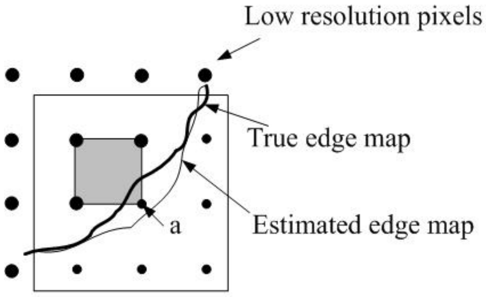


Figure 1.3: Preprocessing to mitigate effect of errors in estimated high resolution edge map. Pixel '*a*' will be replaced by the average of its neighboring pixels on the same side of the estimated edge

In Figure 1.3, the true edge map crosses the square; but the estimated one does not. Thus pixel '*a*', which has a much smaller value than the other three low resolution corner pixels, is used to interpolate the high resolution pixels within the square, leading to undesirably small interpolated values. To solve this problem, value of *a* is compared with the mean μ_a of the subset of its 8 nearest neighbour low resolution pixels which are not separated from *a* by an estimated edge. If the value of *a* differ from μ_a by more than one standard deviation of the pixels whose average comprises μ_a, then '*a*' is replaced by μ_a. After pre-processing of the low resolution image, the rendering operation is performed. In nutshell, the EDI method is based on estimating the edge orientation and accordingly tunes the interpolation coefficients. However, this approach quantizes the edge orientation into a finite number of choices (e.g., horizontal, vertical or diagonal) which affects the accuracy of the imposed edge model. This paper is divided into four section. In section I, basic idea about edge directed interpolation is explained. Section II described the related work done in this field. Methodology of different edge interpolation techniques are presented in section III and results and discussions are given in section IV.

Key Literature Review

Many researchers and engineers have proposed and implemented algorithms for improving the performance of the above mentioned techniques. This section presents a review of related research done over the last decade.

Battiato S. et. al. [1] presented a method to take into account information about discontinuities or sharp luminance variations while doubling the input picture. This is realized by a nonlinear iterative procedure and authors used the cross-correlation and the PSNR between the original picture and the reconstructed picture to assess the quality of reconstruction. Raghupathy A. et. al. [2] proposed a scaling algorithm based on the oriented polynomial image model. They developed a simple classification scheme that classifies the region around a pixel as an oriented or non-oriented block. Based on this classification, a nonlinear oriented interpolation is performed to obtain high quality video scaling. Also, they developed an efficient architecture for scaling a Quarter Common Intermediate Format (QCIF) image to 4CIF format. Frakes D. H. et. al. [3] mentioned that the problem of interslice magnetic resonance (MR) image reconstruction arises in a broad range of medical applications. Therefore, authors have developed a method of vascular morphology reconstruction based on adaptive control grid interpolation (ACGI) function as a precursor to visualization and computational analysis. Muresan D.D. et. al. [4] presented a novel interpolation method based on optimal recovery and adaptively determining the quadratic signal class from the local image behavior. The algorithm first determines the local quadratic signal class from local image patches and then applies optimal recovery to estimate the missing samples. Chen M. J. et. al. [5] partition digital images into homogeneous and edge areas based on the analysis of the local structure on the images. Luong H. et. al. [6] exploited the repetitive character of the image. A great advantage of proposed approach is that more information is available at disposal, which leads to a better reconstruction of the interpolated image. Wang Q. et. al. [7] proposed an isophote orientation-adaptive interpolation method that reduces zigzagging in edges as well as in ridges. Tam W. S. et. al. [8] presented an improvement of the NEDI method, namely the Modified Edge-Directed Interpolation (MEDI). Authors proposed a different training window to mitigate the interpolation error propagation problem. Later on, they found the similar training window had been proposed in the Improved Edge-Directed Interpolation (IEDI). Zhang X. et. al. [9] proposed a new edge-directed image interpolation algorithm which can preserve the edge features and natural appearance of images efficiently. In the proposed scheme, authors first get a close-form solution of the optimal interpolation coefficients under the sense of minimal mean square error by exploiting autoregressive model (AR) and the geometric duality between the low-resolution and high-resolution images. Guangming S. et. al. [10] proposed a context-based image resolution up-conversion technique. It performs image interpolation and de-convolution jointly in a single estimation framework and uses two 1-D context-based interpolators to estimate a missing HR pixel in two orthogonal directions. Dung T. V. et. al. [11] proposed a selective data pruning based compression scheme to improve the rate-distortion relation of compressed images and video sequences. The original frames are pruned to a smaller size before compression. After decoding, they are interpolated back to their original size by an edge-directed interpolation method. Mishiba K. et. al. [12] presented a novel edge-adaptive image interpolation method using an edge directed smoothness filter. Their approach estimated the enlarged image from the original image based on an observation model. Ramadevi G. et. al. [13] reduced computation time by new ICBI (Iterative Curvature Based Interpolation) technique. Iterative Curvature Based Interpolation (ICBI) is based on a two-step grid filling and an iterative correction of the interpolated pixels obtained by minimizing an objective function depending on the second order directional derivatives of the image intensity.

Methodology

Edges are very important features in natural images; so exploiting the geometric regularity of edges becomes of paramount importance in many image processing tasks. The new edge directed interpolation (NEDI) extends the covariance-based adaptation method into a multi-resolution framework. This is done by recognizing the geometric duality between the low-resolution covariance and the high-resolution covariance which couple the pair of pixels along the same orientation. In NEDI, the estimated high-resolution covariance is used to derive the optimal minimum mean squared error (MMSE) interpolation by modelling the image as a locally stationary Gaussian process. The computational complexity of covariance-based adaptive interpolation is about two orders of magnitude higher than that of linear interpolation. The analytical model of NEDI is presented below.

In NEDI, it is assumed that the low-resolution image $X_{i,j}$ of size $H \times W$ directly comes from of size of $2H \times 2W$ image, i.e., $Y_{2i,2j} = X_{i,j}$. And, the interlacing lattice $Y_{2i+1,2j+1}$ is interpolated from the lattice $Y_{2i,2j} = X_{i,j}$, using the fourth-order linear interpolation (Fig. 1.4) as:

$$\hat{Y}_{2i+1,2j+1} = \sum_{k=0}^{1} \sum_{l=0}^{1} \alpha_{2k+l} Y_{2(i+k),2(j+l)} \tag{1.4}$$

Where, the interpolation includes the four nearest neighbors along the diagonal directions. In NEDI, it is assumed that the natural image source can be modeled as a locally stationary Gaussian process and it uses the classical Wiener filtering theory [22] to find the optimal MMSE linear interpolation coefficients, as:

$$\vec{\alpha} = R^{-1}\vec{r} \tag{1.5}$$

Where, $R = [R_{kl}], (0 \leq k, l \leq 3)$and $\vec{r} = [r_k], (0 \leq k \leq 3)$ are the local covariances at the high resolution. The low-resolution covariance $\hat{R}_{kl}, \hat{r}_k$ can be easily estimated from a local window of the low-resolution image using the covariance method as:

$$\hat{R} = \frac{1}{M^2} C^T C, \hat{\vec{r}} = \frac{1}{M^2} C^T \vec{y} \tag{1.6}$$

Where, $\vec{y} = [y_1 \cdots y_k \cdots y_{M^2}]^T$ is the data vector containing the $M \times M$ pixels inside the local window and C is a $4 \times M^2$ data matrix whose kth column vector is the four nearest neighbors of y_k along the diagonal direction. Combining equations 3.5 and 3.6 to have:

$$\hat{\alpha} = (C^T C)^{-1}(C^T \vec{y}) \tag{1.7}$$

Therefore, the interpolated value of $Y_{2i+1,2j+1}$can be obtained by substituting equation 1.7 into 1.4.

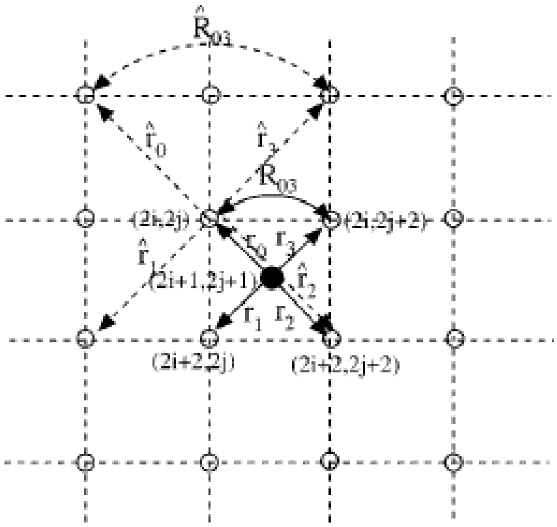


Figure 1.4: Geometric duality when interpolating $Y_{2i+1,2j+1}$from $Y_{2i,2j}$.

The edge-directed property of covariance-based adaptation comes from its ability to tune the interpolation coefficients to match an arbitrarily-oriented step edge. The principal drawback with covariance-based adaptive interpolation is its prohibitive computational complexity. In order to manage the computational complexity in NEDI, the covariance-based adaptive interpolation is only applied to edge pixels

(pixels near an edge); for non-edge pixels (pixels in smooth regions), the bilinear interpolation is used. This hybrid approach is based on the observation that only edge pixels benefit from the covariance-based adaptation and edge pixels often consist of a small fraction of the whole image.

NEDI technique, as mentioned above, has several problems. So, it is modified to increase the interpolation accuracy, and the new technique is called improved New Edge Directed Interpolation (iNEDI). A region growing method is defined in the iNEDI method in which the four valued neighboring pixels of the central point is taken and the neighbours (in the original grid) of these pixels is added iteratively with the following properties:

- The gray level between the maximum and minimum value of the four neighbours is not less than threshold (as in the central point).
- The gray level of each pixel is not larger than the maximum value of the gray level of the four neighbors of the central incremented by a threshold MARGIN and not lower than the minimum of the 4 neighbours of the central point decremented by the same MARGIN.
- The Euclidean distance between the pixel and central point is less than *r*.

The "edge" region is enlarged with the same rules by increasing *r* up to a maximum value R. With this selective procedure and the control on the residual, the probability of obtaining a good interpolation is increased. But, there is still the possibility of having unwanted high frequencies (that are not excluded by the constant covariance condition and may occur in case of a small number of samples in the fit). For this reason, a further constraint is placed by replacing any interpolated value outside the intensity range of the four neighbours with closest of the values delimiting that range (i.e. maximum or minimum). Also in the NEDI method, the interpolated pixel values change with the global brightness, i.e., they do not depend only on differences between neighboring values; but on the absolute value. This effect is removed by changing the NEDI constraint by subtracting the average of the four neighbours intensities from the values inserted in C and $\vec{y}$, i.e. replacing C with

$$C' = \begin{pmatrix} I_{h_1-1,k_1-1} - \bar{I}_{h_1,k_1} & I_{h_1-1,k_1-1} - \bar{I}_{h_1,k_1} & \cdots & \cdots \\ I_{h_2-1,k_2-1} - \bar{I}_{h_2,k_2} & I_{h_2-1,k_2-1} - \bar{I}_{h_2,k_2} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ I_{h_N-1,k_N-1} - \bar{I}_{h_N,k_N} & I_{h_N-1,k_N-1} - \bar{I}_{h_N,k_N} & \cdots & \cdots \end{pmatrix} \tag{1.8}$$

Where, *h*, *k* ∈ *w*(i,j) and changing $\vec{y}$ with

$$\vec{y}\,' = \left(I_{h_1,k_1} - \bar{I}_{h_1,k_1},\ I_{h_2,k_2} - \bar{I}_{h_2,k_2}, \dots, I_{h_N,k_N} - \bar{I}_{h_N,k_N}\right)^T \tag{1.9}$$

Where

$$\bar{I}_{h_1,k_1} = \frac{I_{h_1-1,k_1-1}+I_{h_1-1,k_1+1}+I_{h_1+1,k_1-1}+I_{h_1+1,k_1+1}}{4} \tag{1.10}$$

I(i,j) is then clearly obtained as:

$$I(i,j) = \alpha'.\left(I_{i-1,j-1}, I_{i-1,j+1}, I_{i+1,j-1}, I_{i+1,j+1}\right) + \bar{I}_{i,j} \qquad (1.11)$$

This change clearly makes the matrix C' rank deficient. The fact that C is rank deficient, means that the solution to the least squares problem is not unique. A method that is often used to find an unique solution is to select the minimum norm solution, that is obtained through the computation of the Moore-Penrose pseudo inverse. If it is assumed that the local four pixel configuration is the sum of a term exactly modelled by the constant convariance model plus an error term (i.e., for an odd point in the first step: $\vec{I}_4 = \left(I_{2i,2j}, I_{2i,2j+2}, I_{2i+2,2j}, I_{2i+2,2j+2}\right) = \vec{I}_0 + \vec{I}_{err}$, the squared error on the interpolated value $I_{2i+1,2j+1} = \vec{\alpha}^*.\vec{I}_4$ is $\left(\vec{\alpha}^*.\vec{I}_{err}\right)^2$ and it is in general lowered by choosing the minimum norm solution for α^*. Therefore, the overconstrained system is solved using this method.

Iterative Curvature Based Interpolation (ICBI), like above discussed edge-directed methods, approximately doubles the image size every time it is applied. It also follows the same concept of firstly putting original pixels in an enlarged grid then filling the holes. The hole filling is done in two steps. Firstly, the linear interpolation is performed on the closest points in the direction along which the second-order derivative of the image brightness is lower. After each hole filling step, an iterative refinement is then performed to update the values of the newly inserted pixels by minimizing the local variations of the second-order derivatives of the image intensity while trying to preserve strong discontinuities.

For the first step, the interpolated value is usually computed as:

$$\boldsymbol{I_{2i+1,2j+1} = \vec{\alpha}.\left(I_{2i,2j}, I_{2i,2j+2}, I_{2i+2,2j}, I_{2i+2,2j+2}\right)} \qquad (1.12)$$

The coefficients vector $\vec{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ is estimated from the neighbouring pixels in the grid. The way it is calculated depends upon the algorithm. In ICBI, a better constraint is mentioned by assuming that coefficients in multiplying opposite neighbors are equal; which will give:

$$\boldsymbol{I_{2i+1,2j+1} = \vec{\beta}.\left(I_{2i,2j} + I_{2i+2,2j+2}, I_{2i,2j+2} + I_{2i+2,2j}\right)} \qquad (1.13)$$

Using this approach, the algorithm iteratively refines the interpolated pixels by locally minimizing a function that should be zero when the constraint is valid. Similar to the NEDI algorithm, an overconstrained system is obtained in ICBI and is solved to find β_1 and β_2. However in this case, the inverted matrix is full-ranked. Thus, the above equation can be written as:

$$\boldsymbol{\beta_1(I_{2i,2j} - 2I_{2i+1,2j+1} + I_{2i+2,2j+2}) + \beta_2(I_{2i,2j+2} - 2I_{2i+1,2j+1} + I_{2i+2,2j}) = (1 - 2(\beta_1 + \beta_2))I_{2i+1,2j+1}} \qquad (1.14)$$

In ICBI, it is assumed that the local approximations of the second-order derivatives along the two perpendicular directions $(I_{2i,2j} - 2I_{2i+1,2j+1} + I_{2i+2,2j+2})$ and $(I_{2i,2j+2} - 2I_{2i+1,2j+1} + I_{2i+2,2j})$ divided by the local intensity $I_{2i+1,2j+1}$are constant. Another assumption of the ICBI is that the local gain is null, i.e., $\beta_1 + \beta_2 = 1/2$. Using these assumptions, the ICBI technique after the computation of the new pixel values by taking the average of the two neighbors in the direction of the lowest second-order derivative; then defines an energy component at each new pixel location that is locally minimized when the second-order derivatives are constant. The interpolated pixel values are then modified in an iterative greedy procedure trying to minimize the global energy. The intensity value corresponding to the lower energy is then assigned to the pixel. This procedure is iteratively repeated until the sum of the modified pixels at the current iteration is lower than a fixed threshold, or the maximum number of iterations has been reached. The number of iterations can be also fixed in order to adapt the computational complexity to timing constraints. After the second hole-filling step (assigning values to all the remaining empty pixels), the iterative procedure is repeated in a similar way, just replacing the diagonal derivatives in the energy terms with horizontal and vertical ones and iteratively modifying only the values of the newly added pixels.

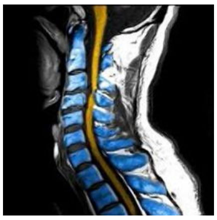
Results and Discussion

In order to investigate the effects of various edge directed interpolation methods viz. NEDI, INEDI, ICBI, EGII, the same are implemented in MATLab. To investigate the effect of above mentioned functions, a CT scan image of spine is taken. The image is resized to 50%. This reduced size image is then zoomed to its original size using various edge interpolation methods as shown in Fig. 1.5 . The parameters used for the evaluation of various techniques are processing time, PSNR, MSE, SSIM, Mutual information and SNR. These parameters are evaluated for the above mentioned techniques and tabulated in TABLE I. It is found that processing time of EGII technique is small compared to all other techniques but PSNR value is highest in case of ICBI technique which shows that this technique has lesser mean square error (MSE) and higher SNR value as tabulated in TABLE I. Also structural similarity index in case of ICBI is higher as compared to other techniques. The graphs are plotted between various parameters for different edge interpolation methods as shown in Fig. 1.6.

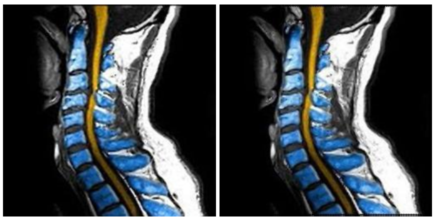
Original Image

Reduced Image

NEDI



INEDI



ICBI **EGII**

Figure 1.5: Results of various Edge interpolation techniques

Table 1:

PARAMETERS	**ALGORITHM**			
	NEDI	**INEDI**	**ICBI**	**EGII**
Time Taken	20.6480	420.7805	159.3083	18.1021
PSNR	27.4180	27.9197	28.1129	28.0233
MSE	117.8371	104.9804	100.4119	102.5062
Maximum MSE	175	141	158	161
SSIM	0.9265	0.9335	0.9394	0.9347
Mutual Information	3.0584	2.9234	3.0262	3.0664
SNR	6.8063	6.9430	7.5785	7.3051

Conclusion

From the above said discussion, it is concluded that the processing time of EGII technique is lesser than all other techniques. The processing time of NEDI is comparable to EGII but this technique has accuracy disadvantage. The drawbacks of NEDI are overcomed in INEDI which is further improved in ICBI technique. PSNR, MSE, SNR and SSIM of ICBI technique is better than all other techniques except processing time. So, it is concluded that ICBI is the best edge interpolation technique if processing time is not a major issue.

References

[1] S. Battiato, G. Gallo, F. Stanco, "A locally adaptive zooming algorithm for digital images," Image and Vision Computing 20, pp. 805–812, 2002.

[2] A. Raghupathy, N. Chandrachoodan, K. J. R. Liu, "Algorithm and VLSI Architecture for High Performance Adaptive Video Scaling," IEEE Transactions On Multimedia, vol. 5, No. 4, Dec 2003

[3] D. H. Frakes, C. P. Conrad, T. M. Healy, J. W. Monaco, M. Fogel, S. Sharma, M. J. T. Smith, A. P. Yoganathan, "Application of an Adaptive

Control Grid Interpolation Technique to Morphological Vascular Reconstruction," IEEE Transactions On Biomedical Engineering, vol. 50, No. 2, Feb 2003.

[4] D. D. Muresan, T. W. Parks, "Adaptively Quadratic (AQua) Image Interpolation," IEEE Transactions On Image Processing, vol. 13, No. 5, May 2004.

[5] M. J. Chen, C. H. Huang, W. L. Lee, "A fast edge-oriented algorithm for image interpolation," Image and Vision Computing 23, pp. 791–798, 2005.

[6] H. Luong, A. Ledda, W. Philips, "An Image Interpolation Scheme for Repetitive Structures," ICIAR 2006, LNCS 4141, pp. 104–115, 2006.

[7] Q. Wang, R. K. Ward, "A New Orientation-Adaptive Interpolation Method," IEEE Transactions On Image Processing, vol. 16, No. 4, April 2007.

[8] T. Acharya, P. S. Tsai, "Computational Foundations of Image Interpolation Algorithms," ACM Ubiquity vol. 8, 2007.

[9] W. S. Tam, C. W. Kok, W. C. Siu, "A Modified Edge Directed Interpolation For Images," 17th European Signal Processing Conference (EUSIPCO 2009) Glasgow, Scotland, Aug 24-28, 2009

[10] X. Zhang, S. Ma, Y. Zhang, L. Zhang, W. Gao, "Nonlocal Edge-directed Interpolation," Advances in Multimedia Information Processing-PCM, pp. 1197-1207, 2009.

[11] G. Shi, W. Dong, X. W. L. Zhang, "Context-based adaptive image resolution upconversion," Journal of Electronic Imaging 19(1), 013008, Jan–Mar 2010.

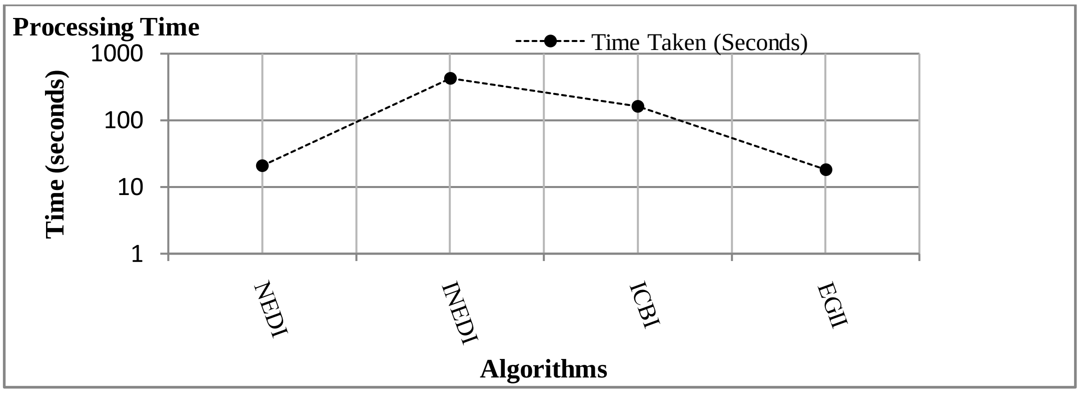
[12] D. T. Vo, J. Sole, P. Yin, C. Gomila, T. Q. Nguyen, "Selective Data Pruning-Based Compression Using High-Order Edge-Directed Interpolation," IEEE Transactions On Image Processing, vol. 19, No. 2, Feb 2010.

[13] G.Ramadevi, T.Mallikarjuna, "Real Time Artifact-Free Image Upscaling," IOSR Journal of Electronics and Communication Engineering, vol. 3, Issue 1, pp. 12-19, Sep-Oct 2012.

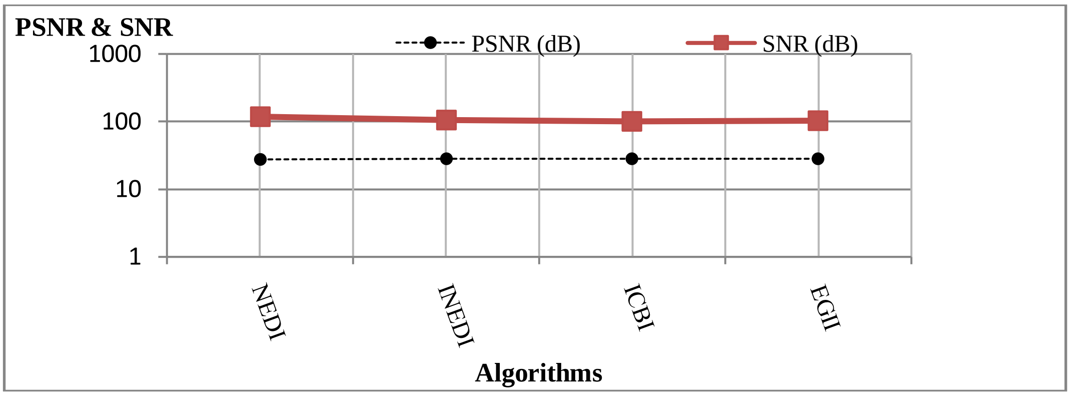
[14] S. Yu, Q. Zhu, S. Wu, Y. Xie, "Performance Evaluation of Edge-Directed Interpolation Methods for Images," Computer Vision and Pattern Recognition, Mar 2013

[15] T. Gulati, M. Pal, "Interpreting Low Resolution CT Scan Images Using Interpolation Functions," International Journal of Computer Applications (0975 – 8887), vol. 74– No.3, July 2013

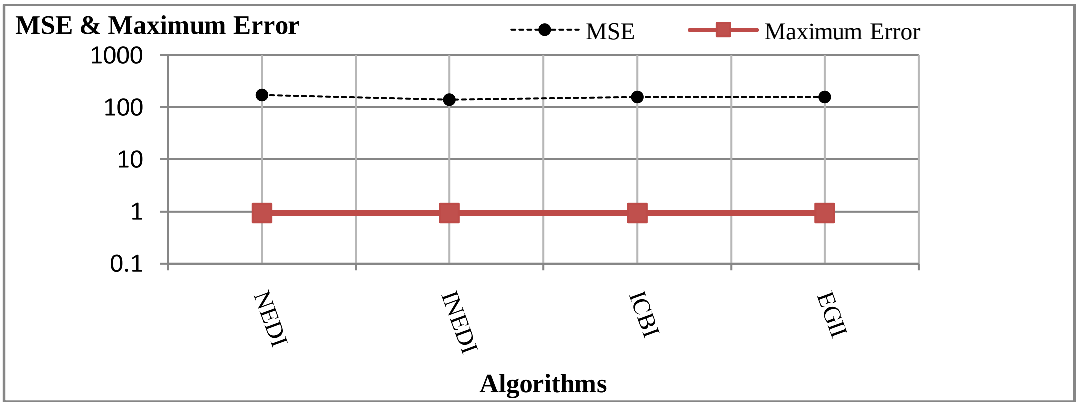
[16] T. Gulati, H.P. Sinha, "Interpreting Low Resolution MRI Images Using Polynomial Based Interpolation," International Journal of Engineering Trends and Technology, vol. 10, No. 13, Apr 2014


Processing Time
Time Taken (Seconds)
Time (seconds)
1000
100
10
1
NEDI
INEDI
ICBI
EGII
Algorithms

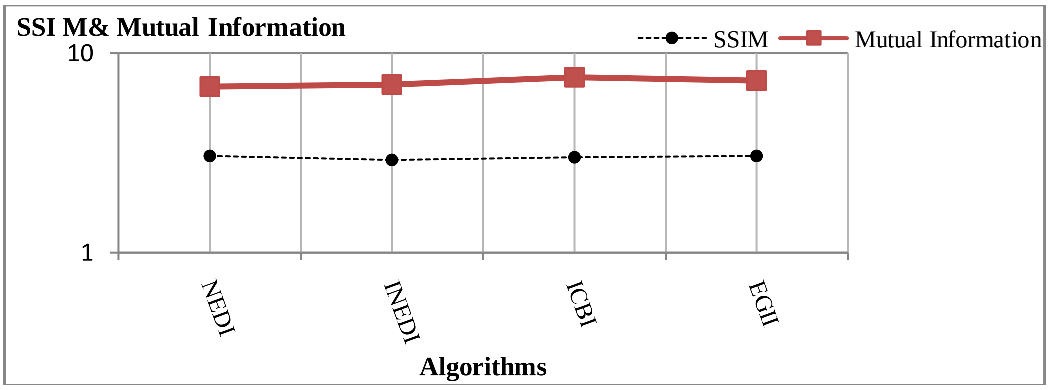
(a)


PSNR & SNR
PSNR (dB)
SNR (dB)
1000
100
10
1
NEDI
INEDI
ICBI
EGII
Algorithms

(b)


MSE & Maximum Error
MSE
Maximum Error
1000
100
10
1
0.1
NEDI
INEDI
ICBI
EGII
Algorithms

(c)



(d)

Figure 1.6: Graphs showing the result of various parameters for different edge interpolation techniques.