Interlinear Image Interpolation Scheme for Real Time Application

Abhinash Kumar Jha, Vishwajeet Narwal, Yashi Gupta
The LNM Institute of Information Technology,
Jaipur,India

Ayush Kumar
Dept. of Computer Science
The State University of New York
USA

***Abstract*—Interpolation is a technique for obtaining new unknown data points within the range of discrete known data points and is often used to recover an image from its downsampled version, or to simply perform image expansion. Recently a lot of interpolation algorithms are proposed, but these interpolation algorithms are highly computationally expensive. Hence these algorithms cannot be implemented and used in real time applications. In view of real time applications we have proposed a computationally simple interpolation algorithm. In our proposed algorithm the unknown pixels are categorized into three types depending upon their spatial position. For each type of pixels, distinct methods are used for prediction. The experimental results show that our proposed algorithm excels the conventional interpolation methods in visual effect, and has a lower complexity. Therefore, the algorithm adapts to real-time image resizing.**

***Keywords*—Interpolation, Low Computational, Interlinear, Edge Preservation, Zooming, Real time Application**

I. INTRODUCTION

Image Interpolation has a wider set of applications in the field of image processing. It enables the user to generate an image of high resolution(HR) from its given low resolution(LR) image interactively. Besides its applications in remote sensing, image interpolation is applied in diverse areas ranging from computer graphics, rendering, editing and medical image construction to outline image viewing.

The main objective of an interpolation technique is to increase pixel density per unit area of an image to obtain an interpolated image version from a low resolution one [1]. If, as illustrated in Fig. 1, we have a discrete sequence $f(x_k)$ of length N, and this sequence is filtered and downsampled by a factor of 2, we get another sequence $g(x_n)$ of length $N/2$. The interpolation process aims at estimating a sequence $l(x_k)$ of length N, which is as close as possible to the original sequence $f(x_k)$.

Upon surveying the literature it can be found that several good interpolation schemes like Nearest Neighbour [2], Bilinear [3], Bicubic [4] , Splines [5] are well known and are used in many real time applications. However, these methods are either computationally expensive or are not able to yield good results.

Bilinear and Bicubic Interpolation are among the classical low complex interpolation algorithm. In Bilinear the predicted pixel value is a weighted average of pixels in the nearest 2-by-2 neighborhood, while in Bicubic the output pixel value is a weighted average of pixels in the nearest 4-by-4 neighborhood.Both of these algorithms despite of being fast and low complex, suffers from problem of error propagation as both of them are using predicted value of pixels to predict values of further unknown pixels which leads to degradation of result.

978-1-4799-5364-6/14/$31.00

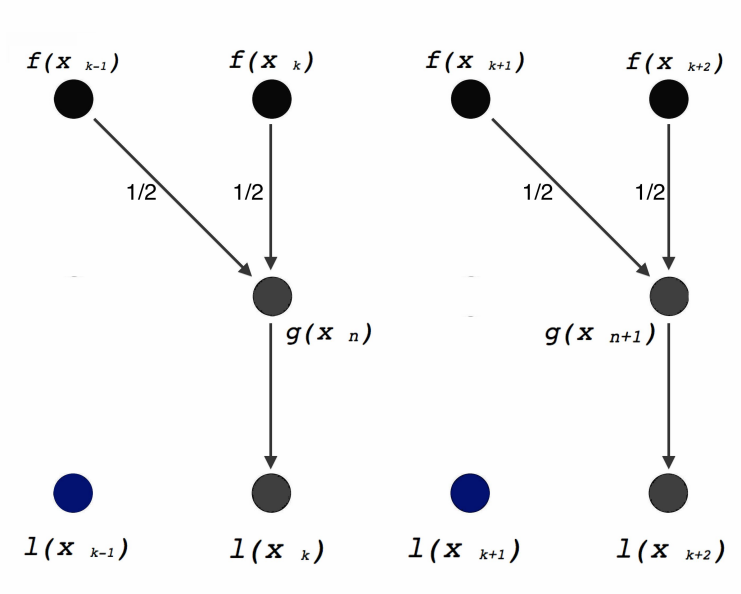

Fig. 1. Signal downsampling and interpolation.

In order to get better quality of interpolated image, various other new low complexity interpolation algorithms have been developed so far. These algorithms involves steps like classifying edges and smooth areas, direction of edges etc. These steps while not effecting the time complexity , increases the run time. Jaiswal et al.[6] suggested the low complex adaptive Interpolation algorithm, in which the unknown pixels are divided into several bins depending upon the characteristics of the neighbouring pixels and a fixed set of prediction coefficient are defined for prediction of unknown pixels. Jha et. al. [11] , an edge preserving technique was proposed based on inverse gradient weights as well as pixel locations for interpolation. Battiato et. al. [7] proposed a new edge adaptive zooming algorithm. The basic idea of their algorithm is to perform a gradient-controlled, weighted interpolation. The method, however, does not require a preliminary gradient computation because the relevant information is collected during the zooming process. The detection of edges may not increases complexity but is time consuming.

The motivation behind this work is to develop a very low complex interpolation algorithm which can be implemented and used in real time applications with better objective and

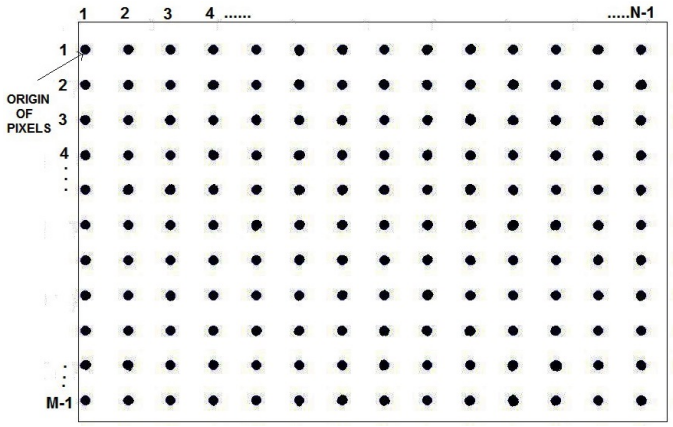

Fig. 3. Sign convention used for determining the current spatial location of pixels

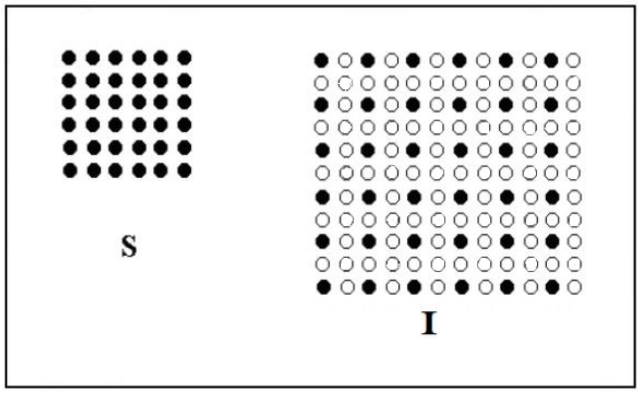

Fig. 4. Transtion of Low Resolution Image to Odd-Odd location of High resolution image

subjective quality as compared to other low complex algorithm in literature. Unlike other low complex algorithm , we are classifying unknown pixels in only three types based on spatial location similar to bilinear and each type of pixels are computed in different manner. In order to get better quality , we need to mitigate the propagation of error which occurs due to use of predicted pixels in prediction of unknown pixels.For this reason, we are using known values of pixels to predict all three types of missing pixels.
In the proposed algorithm, we solved the problem of error propoagation by using only known pixels which are originally derived from low resolution image while expansion.

Our experiments show that the proposed low complexity algorithm takes less time as compared to other algorithms in literature and beats in quality pixel replication, bilinear interpolation and bicubic interpolation. Remaining part of the paper is organised as follows. Section II discusses proposed algorithm which is summarised in flowchart also.Reason for the splendid performance of algorithm is discussed in section III Simulation results and concluding remarks are made in section IV and V respectively.

II. Proposed Algorithm

In this Section we give a detailed description of the proposed algorithm. Classification of Pixels are done into four cateogaries viz., odd-odd, even-even, odd-even and even-odd based on the current spatial location. The Sign convention used for determining the current spatial location of pixels in image is shown in Fig. 3 . Initially the LR image is mapped to HR version and the remaining pixels are predicted using three prediction scheme as per their current spatial location. All unknown pixels are predicted by using only original pixel values of HR image which is derived from the mapping from LR version. This technique serves our objective of mitigating the error propagation that occurs in most of the classic methods of interpolation like bilinear interpolation . The procedure is summarized in Data flow Diagram as shown in Fig. 2 .

Initial step of the scheme focuses on the expansion of source LR image (S) into a HR image grid (I) ($M \times N$ onto a regular grid of size $2M \times 2N$) . The transition from source LR image to interpolated HR image follows the mapping :

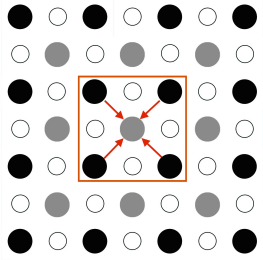

Fig. 5. Boxed region shows the Pixels used for the prediction of pixels at even-even spatial location.

$$M : S \mapsto I$$

governed by the equation :

$$M(S(i,j)) = I(2i-1, 2j-1) \quad (1)$$

The effect of the initial transition can be seen in Fig. 4 . Remaining $(\frac{3}{4})$ pixels of I, are filled differently depending on the cateogary of current pixel's spatial location. The following steps is used for the prediction of the remaining pixels of the I

A. Diagonal Prediction Scheme

Diagonal prediction scheme is used to predict the missing pixels lying at the even-even position. For a 3 X 3 mask, the value assigned to the unknown pixels at even-even position is the average of the Close-4 neighbours i.e, the pixels we obtain by rejecting Nearest 4-neighbours in Nearest 8-neighbours as shown in Fig. 5 . Simply the four neighbouring pixels lying diagonally are used for the prediction of the pixels at even-even positions. This step is very similar to that of the bilinear interpolation and in general is given by :

$$I(2i,2j) = \frac{1}{2(n+1)} \sum_{y=1}^{n<4} \sum_{z=1}^{n<4} I(2(i-y)\pm 1, 2(j-z)\pm 1) \quad (2)$$

The larger would be the value of n the more computationally inefficient algorithm would be. As the current pixel is inspired only by neighbouring pixels for n $>$ 4 error would propagate into the prediction.

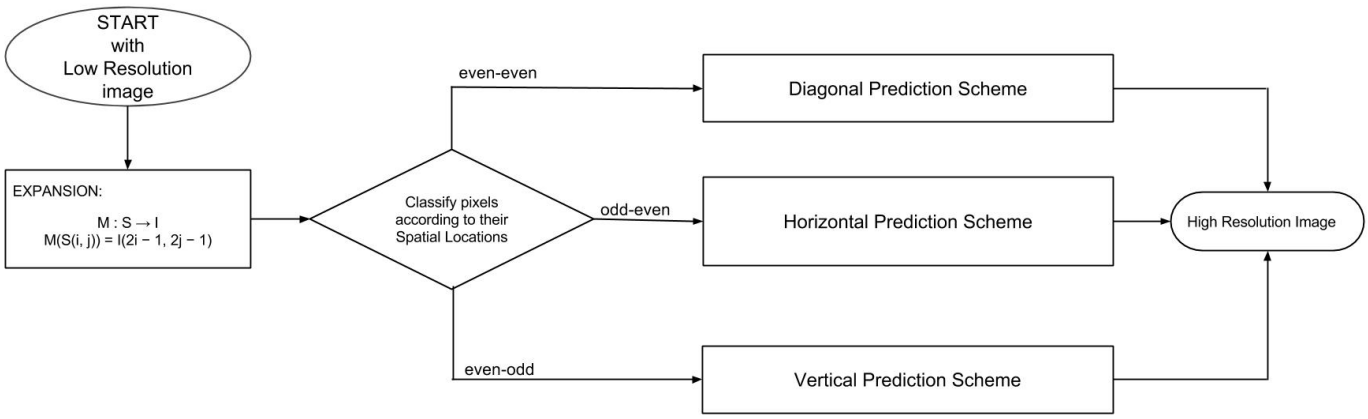

Fig. 2. Data Flow Diagram decribing the procedure for the Proposed Interpolation Scheme

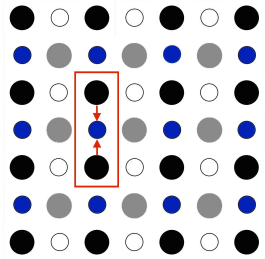

Fig. 6. Pixel selection for 3 X 3 mask as per Vertical Prediction scheme for fillling Even-Odd pixels

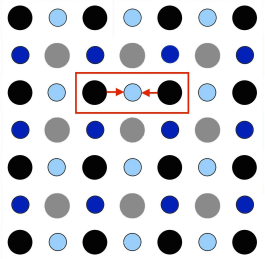

Fig. 7. Pixel selection for 3 X 3 mask as per Horizontal Prediction scheme for fillling Odd-Even pixels

B. Vertical Prediction Scheme

Using Vertical Prediction scheme, the missing pixels lying at even-odd position are predicted. For a 3 X3 mask the values of the pixels lying at the even-odd position are predicted as the average of the pixels lying at odd-odd position in nearest-4 neighbours as shown in Fig. 6. In general, this scheme is governed by equation :

$$I(2i, 2j-1) = \frac{1}{n+1} \sum_{z=1}^{n<4} I(2i \pm z, 2j-1) \qquad (3)$$

C. Horizontal Prediction Scheme

The remaining pixels lying at the odd-even position are predicted by Horizontal Prediction Scheme. This prediction scheme is very similar to the Vertical prediction scheme . For a 3 X 3 mask, the pixels lying at the odd-even position are predicted by taking the average of the odd-odd pixels of the nearest-4 one as in Fig. 7 .In other words, the neighbouring pixels lying at the odd-odd position and are placed just right and left of the current spatial location of the pixel which is to be predicted. The general equation of the scheme is given as:

$$I(2i, 2j-1) = \frac{1}{n+1} \sum_{z=1}^{n<4} I(2i-1, 2j \pm z) \qquad (4)$$

III. REASON FOR PERFORMANCE OF PROPOSED ALGORITHM

The emphasis of this algorithm is not to predict new edges that are absent in LR image, but is to interpolate existing edges in LR image to High Resolution. Consider an horizontal edge in LR image (can be seen in figure). Now we can observe that the pixels that can represent the horizontal edge in HR image are either the original pixels (odd-odd) or the pixels located at odd-even position as can be seen in Fig. 8 .

Similarly, Consider an vertical edge in LR image (can be seen in figure). Now we can observe that the pixels that can represent the vertical edge in HR image are either the original pixels (odd-odd) or the pixels located at even-odd position as can be seen in Fig. 9 .

Similarly, Consider an diagonal edge in LR image (can be seen in figure). Now we can observe that the pixels that can represent the diagonal edge in HR image are either the original

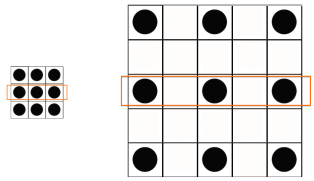

Fig. 8. Interpolating Horizontal Edge

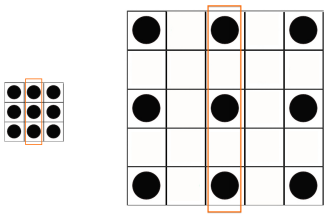

Fig. 9. Interpolation of Vertical edge

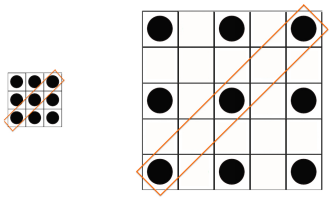

Fig. 10. Interpolation of Diagonal edge

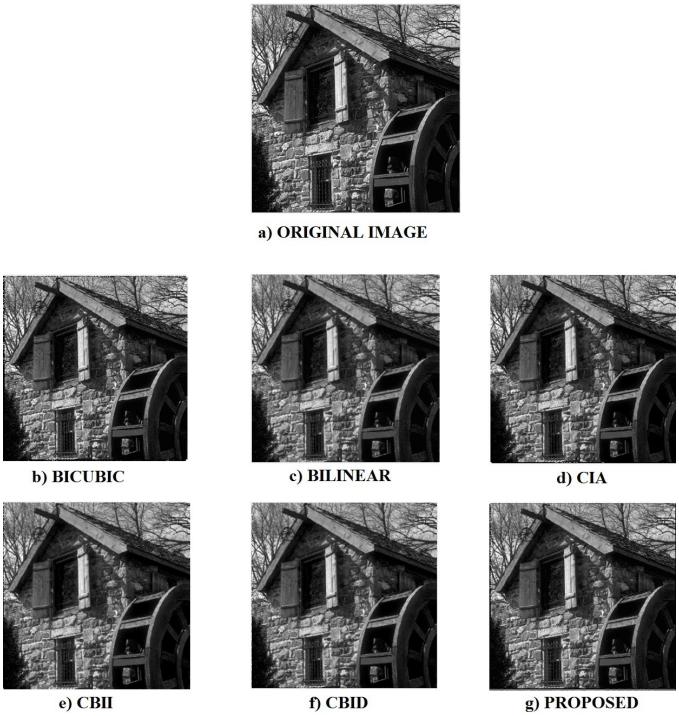

Fig. 11. Comparison demostrating Visual Quality of sample image (a) interpolated by the different methods.

pixels (odd-odd) or the pixels located at even-even position as can be seen in Fig. 10 .

This can be summarized by saying that by predicting odd-even, even-odd and even-even pixels, we can interpolate horizontal, vertical and diagonal edges respectively.

This analogy assumes that there is edge at every pixel, but even in opposite case, i.e : smooth area, the pixel is predicted as an average of two nieghbouring pixels that are known to us with 0 % prediction error.

There may be some cases in which this algorithm may fail for example : - if a image contain most of the edges in even-even,odd-even or even-odd pixels, then its downsampled version may not contain any edge. Hence interpolated image will have less edges.

IV. EXPERIMENTAL ANALYSIS

A visual comparison of the quality of the different methods is presented in Fig. 11 which shows an original image together with interpolated versions obtained by all tested methods, and conrms that our presented approach is capable of providing superior image quality. The interpolated images are resized in order to fit to the size of paper. An extensive set of experiments were carried out to measure the efficiency and efficacy of the proposed algorithm. A dataset [10] comprising of 50 grayscale images were used as shown in Fig. 12 . All the experiments were carried on Intel Core i5 Processor with clock cycle of 2.4Ghz having two cores, L2 cache of 256KB per core and L3 cache of 3MB on 64-bit Matlab version(7.13.0.564) as a simulation tool.
Objective quality measurement of the algorithm was done using peak-signal to noise ratio(PSNR) defined as per the equation :

$$\text{PSNR}(O, I) = 10 \log_{10} \frac{255^2}{\text{MSE}(O, I)}, \tag{5}$$

Five other interpolated algorithms were implemented for putting our proposed one into context namely : bilinear interpolation, bicubic interpolation, content adaptive interpolation (CAI) [8], context-based image dependent interpolation (CBID)[9] , and context-based image independent interpolation [9].Table I describes the PSNR results and Computational runtime is show in Table II . It can be seen that our proposed Algorithm works better than others not only in terms of objective quality but also in terms of computational efficiency. Our algorithm is not only computationally efficient but also posses Edge Preservation property. A comparison can be seen in Fig. 13 .

V. CONCLUSION

In this paper, we have presented a computationally efficient interpolation algorithm.Our proposed algorithm uses three prediction schemes viz, diagonal, vertical and horizontal based on the current spatial location .Through extensive experiments it can be found that our proposed algorithm not only works better in terms of computation but also enhances visual quality. Another major advantage is effective edge preservation property of the scheme. In future work, we will aim at weighted calculation in all the prediction scheme in order to improve the visual quality. The present algorithm can also be extended for color images.

Fig. 12. Dataset of 50 Images used for the Evaluation of the Proposed Interpolation Scheme.

Fig. 13. Visual comparison showing Edge preservationproperty by various algorithm.

REFERENCES

[1] Wittman, Todd. "Mathematical techniques for image interpolation." *Report Submitted for Completion of Mathematics Department Oral Exam, Department of Mathematics, University of Minnesota, USA (2005).*

[2] Franke, Richard. "Scattered data interpolation: Tests of some methods." *Mathematics of computation 38.157 (1982): 181-200.*

[3] Jensen, Kris, Dimitris Anastassiou "Subpixel edge localization and the interpolation of still images." *Image Processing, IEEE Transactions on 4.3 (1995): 285-295.*

[4] Fritsch, Frederick N., Ralph E. Carlson. "Monotone piecewise cubic interpolation." , *SIAM Journal on Numerical Analysis 17.2 (1980): 238-246.*

[5] Hou, Hsieh S., H. Andrews. "Cubic splines for image interpolation and digital filtering." *Acoustics, Speech and Signal Processing, IEEE Transactions on 26.6 (1978): 508-517.*

[6] Prasad Jaiswal, Sunil, Vinit Jakhetiya, Ayush Kumar, Anil Kumar Tiwari. "A low complex context adaptive image interpolation algorithm for real-time applications.", *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International. IEEE, 2012.*

[7] Battiato, Sebastiano, Giovanni Gallo, Filippo Stanco "A locally adaptive zooming algorithm for digital images.", *Image and vision computing 20.11 (2002): 805-812.*

[8] Chan, Tai-Wai, Oscar C. Au, Tak-Song Chong, Wing-San Chau. "A novel content-adaptive interpolation.", *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on. IEEE, 2005.*

[9] Jaiswal, Sunil Prasad, Vinit Jakhetiya, Anil Kumar Tiwari. "An efficient image interpolation algorithm based upon the switching and self learned characteristics for natural images.", *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on. IEEE, 2011.*

[10] "Decsai.ugr.es." "Dataset of standard 512x512 grayscale test images", *http://decsai.ugr.es/cvg/CG/base.htm*

[11] Abhinash Kumar Jha, Ayush Kumar, Schaefer, G. ; Ahad, M.A.R. "An efficient edge preserving image interpolation algorithm", *Informatics, Electronics & Vision (ICIEV), 2014 International Conference on. IEEE, 2014.*

images	Bilinear	bicubic	oscar	cbii	cbid	interlinear
1	23.83	23.33	24.37	24.99	24.81	**25.11**
2	27.70	28.62	30.54	30.49	30.50	**30.68**
3	28.09	27.38	29.13	29.42	29.49	**29.77**
4	28.77	29.34	31.08	30.97	31.04	**31.38**
5	27.69	31.33	33.92	33.74	33.79	**34.07**
6	26.25	26.02	27.26	27.67	27.83	**28.13**
7	19.46	19.71	20.74	21.11	21.05	**21.17**
8	28.49	29.22	30.88	30.89	30.93	**31.18**
9	24.98	25.87	27.01	27.13	27.15	**27.30**
10	24.69	27.74	29.99	29.96	**32.41**	32.39
11	18.95	18.93	20.03	20.46	20.42	**20.51**
12	25.16	26.25	27.56	28.11	28.17	**28.33**
13	26.93	26.15	27.94	28.08	**30.18**	29.89
14	21.14	20.64	21.29	21.88	21.95	**22.11**
15	24.50	23.58	24.36	25.04	25.07	**25.35**
16	22.40	21.54	22.28	22.86	**22.92**	22.80
17	23.76	22.50	24.12	24.38	24.50	**24.91**
18	20.65	19.52	20.41	21.06	21.11	**21.27**
19	26.74	25.81	27.20	27.85	27.96	**28.20**
20	25.69	25.39	26.41	26.89	26.91	**27.23**
21	27.40	27.60	28.69	28.92	28.95	**29.21**
22	23.86	23.14	24.24	24.65	**24.70**	24.68
23	29.87	31.98	34.83	34.48	34.50	**34.86**
24	28.22	29.45	31.13	31.33	31.36	**31.57**
25	29.98	32.43	33.82	33.89	33.90	**34.16**
26	28.72	30.81	32.69	32.83	32.84	**32.96**
27	31.07	30.82	32.70	32.90	32.89	**33.04**
28	27.98	28.58	29.74	29.97	29.97	**30.03**
29	24.25	23.13	24.14	24.63	**24.69**	24.57
30	26.37	26.03	27.50	27.64	27.84	**28.07**
31	20.46	19.32	19.79	20.58	**20.59**	20.54
32	25.91	24.47	25.95	26.51	26.61	**26.73**
33	26.04	26.42	28.14	28.04	28.13	**28.35**
34	26.94	26.00	27.64	27.84	27.96	**28.02**
35	26.63	27.87	29.83	29.97	30.05	**30.13**
36	23.38	24.45	25.80	26.05	26.11	**26.30**
37	27.62	27.24	28.65	28.78	28.82	**29.00**
38	26.98	29.40	31.14	30.98	31.07	**31.47**
39	26.47	30.82	32.48	32.69	**32.75**	32.73
40	27.67	33.40	36.04	35.65	35.72	**36.13**
41	24.45	26.76	28.67	28.78	28.80	**29.07**
42	25.34	25.60	26.63	26.85	27.03	**27.16**
43	28.75	28.50	31.63	31.33	31.54	**32.10**
44	17.91	17.09	18.12	18.43	**18.51**	18.45
45	25.72	26.77	28.72	29.27	**29.68**	29.51
46	24.84	23.99	25.41	25.65	25.82	**25.83**
47	22.39	21.35	22.25	22.62	**23.15**	23.14
48	26.03	27.73	30.29	30.18	30.24	**30.57**
49	17.72	**22.90**	17.37	18.25	18.23	18.33
50	25.82	27.59	29.13	29.14	29.25	**29.28**
average	25.41	26.01	27.39	27.64	27.80	**27.96**

TABLE I. OBJECTIVE EVALUATION OF DIFFERENT ALGORITHMS BY PSNR

images	Bilinear	bicubic	oscar	cbii	cbid	interlinear
1	21.33	89.78	441.76	467.68	854.88	**17.03**
2	22.43	87.51	366.33	435.91	885.91	**15.55**
3	23.97	91.52	382.47	476.44	975.10	**15.81**
4	20.56	85.68	358.88	419.14	837.61	**15.71**
5	19.66	83.72	354.06	433.01	829.92	**16.18**
6	19.50	84.55	369.93	419.89	833.47	**16.73**
7	19.35	83.90	340.35	413.44	836.03	**17.82**
8	**20.47**	84.26	368.49	474.26	864.72	21.71
9	**19.52**	94.56	366.98	484.42	990.09	20.19
10	20.64	90.01	377.02	446.45	894.05	**15.46**
11	28.84	85.88	361.63	413.73	827.13	**16.21**
12	19.32	84.81	359.42	417.60	842.37	**15.01**
13	19.19	84.17	339.30	419.54	824.17	**16.86**
14	19.02	87.20	355.42	411.74	833.57	**15.81**
15	**21.18**	83.79	366.11	420.68	827.51	24.46
16	20.66	83.28	366.50	410.52	839.74	**20.52**
17	19.42	90.77	383.58	420.06	823.29	**16.07**
18	30.26	116.40	440.72	428.32	872.42	**18.86**
19	31.22	85.73	375.99	465.78	871.26	**17.27**
20	22.87	90.01	367.82	423.02	822.18	**15.53**
21	25.27	84.35	355.18	420.14	821.64	**15.29**
22	20.56	84.53	347.48	412.69	818.85	**15.78**
23	19.82	86.16	356.37	422.97	838.79	**15.43**
24	20.00	84.34	357.15	411.21	829.32	**16.54**
25	20.32	83.28	365.72	417.14	827.55	**15.71**
26	20.09	88.78	356.91	416.55	815.22	**15.57**
27	20.71	83.70	357.00	410.86	835.26	**17.16**
28	20.11	83.70	349.42	429.08	814.53	**15.07**
29	19.47	84.52	350.34	411.10	827.89	**18.08**
30	20.55	88.90	360.59	418.44	827.10	**15.84**
31	26.98	84.32	357.26	421.10	851.60	**15.86**
32	19.53	83.06	352.67	428.15	826.71	**15.80**
33	21.35	83.55	356.46	417.06	834.40	**18.86**
34	21.92	83.35	360.30	417.60	839.35	**15.60**
35	19.32	85.49	347.49	415.59	824.07	**15.29**
36	**19.76**	83.64	353.52	413.14	819.58	21.02
37	19.61	83.38	358.94	424.04	841.00	**15.40**
38	**19.70**	84.12	350.20	426.37	850.09	20.99
39	21.34	83.43	358.40	412.58	827.15	**15.52**
40	21.16	93.15	346.27	412.95	839.51	**15.74**
41	20.19	85.31	351.87	426.76	826.33	**16.62**
42	21.72	84.18	350.95	419.51	827.06	**16.02**
43	20.31	83.72	362.26	418.32	818.43	**15.82**
44	21.10	84.73	338.68	427.17	847.82	**15.80**
45	20.84	90.20	355.50	421.67	851.89	**16.58**
46	22.91	84.48	354.07	416.99	821.43	**15.28**
47	20.46	87.85	357.01	419.95	828.42	**18.68**
48	20.90	83.97	350.53	424.70	814.35	**15.52**
49	22.24	93.76	335.26	407.40	821.29	**16.45**
50	19.17	84.07	344.75	418.54	847.67	**17.47**
average	21.34	86.59	360.83	425.23	841.99	**16.87**

TABLE II. COMPUTATIONAL TIME(IN MILLISECONDS) COMPARISON OF VARIOUS ALGORITHM