ADAPTIVE INTERPOLATION BASED LOSSY IMAGE COMPRESSION AND DECOMPRESSION

Tejashree Tukaram Patil[1], Prof. Gajanan Kadam[2]

[1]Post Graduate Student, Department of ECE,, KLS Gogte Institute of Technology College, Belagavi, India
[2]Professor, Department of ECE, KLS Gogte Institute of Technology College, Belagavi, India

Abstract : A simple algorithm based on adaptive interpolation for compression and decompression is proposed. The purpose of this project will be to discuss an algorithm for compressing a digital image and then acquire back the original image from the compressed one. The objective of this project is to provide better compression ratio and validate the results using MATLAB programming. As Compared to JPEG it provides better compression ratio.The suggested method can be applied to any type of image. The method offers compression ratio depending on dimensions of image. The suggested technique is utilized on any still gray-scale image.

Keywords: Lossy Image Compression, Newton Interpolation, Compression Ratio.

1. INTRODUCTION

Image compression is important for efficient and effective transmitting and storing of images. Compression may be done either using Lossy compression method or Lossless compression method. Lossless compression reduces the no of bits by determining and dropping statistical redundancy such that information is not lost in lossless compression. In Lossless compression type, the image reconstructed is same after compression as the original image but with Low compression ratios, whereas in Lossy compression type, the reconstructed image is not as same as original image but provides High compression ratios. Lossy compression strategy lessens the bits by expelling unneeded or lesser critical data. Hence, high compression ratios can be achieved for the images where, some loss does not matter. Objective of project is to reduce irrelevance i.e, omit pixels which are not easily noticeable by human eyes and redundancy from image data so that for storing and transmitting the data without causing degradation of quality of the images. We make use of non-linear interpolation to reconstruct the original image back from compressed image. The proposed decompression algorithm can also be used universallly to enlarge any given image. A few quality estimation procedures such as, "Compression Ratio", "Signal to Noise Ratio", "Peak signal to Noise Ratio" and "Mean Square Error" are calculated of original and compressed image.

2. LITERATURE SURVEY

Image compression has turned out to be important for storages and transmission [1]. The second-order differences of the contiguous pixels gray values demonstrates the relation among the pixels. The second-order differences of the adjacent pixels gray values shows the relation among the pixels [2].Image compression is a technique/strategy for reproducing a original image without influencing the picture quality. Interpolation is an estimation of a value within two known values in a series of values. Image interpolation is used for transfering an image form a resolution to other resolution with same visual information. Image interpolation exists in every digital images at some specific stage point such as in photo enlargement. Image interpolation can be used on different regions, for example, PC Designs, rendering of images, editing, medical image construction and outline image

viewing. The process of converting an image starting with one scale then onto another is known as image scaling .The image interpolation can be done in two sorts, such as adaptive interpolation and non-adaptive interpolation. Problems with non-adaptive technique are blurred-edges or artifacts [3]. Lossy compression provides a good compression ratio and reduce the size of files, when the images are used for storing or transmitting through the network [4]. Lossless compression doesn't lessen the document file as Lossy [5]. When an image compressed is decompressed, there may be loss of information provided no compromise of image quality [6]. When considering Neural networks for compressing images they provide low compression rates so, this technique can't be used instead of some standard compression codec such as JPEG [7].

3. PROCEDURE

The basic idea is to implement for gray-scale image. Consider gray-scale image of size (m×n) which has to be compressed. Gray-values of image ranges from 0 to $2^b - 1$. Zero is black and $2^b - 1$ is white and in-between values are shades of black to white. The method offers varying compression ratios depending on the dimensions of the original images and the acquired decompressed images are similar to the original ones.

3.1 COMPRESSION

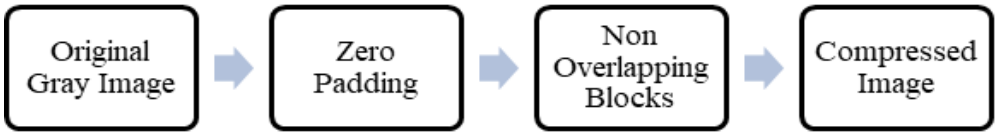

Figure 1: Block Diagram of Compression

Consider ‘b’ bit gray-scale image. Begin from first pixel in the image and select 8 adjoining pixels with the end goal that a square is framed. The zoomed pixels of square are shown as in Figure 2.

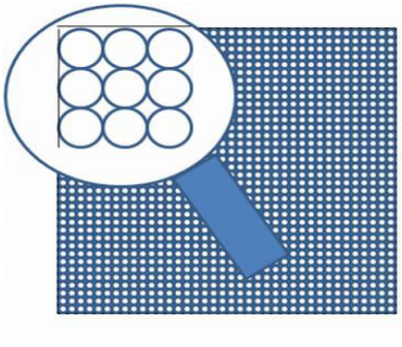

Figure 2: Magnified pixels

	1	2	3	4	5	6	7	8	9	10
1	194	173	188	172	179	178	195	191	199	183
2	178	177	177	177	178	178	182	185	184	182
3	180	183	185	181	175	172	175	180	182	182
4	190	188	185	190	174	165	167	176	180	181
5	188	186	183	180	174	165	167	176	180	181
6	180	180	182	182	181	181	182	181	175	169
7	194	173	168	172	176	178	185	191	199	193
8	178	177	177	177	178	178	182	185	184	182
9	180	183	186	181	175	172	176	180	182	182
10	174	177	180	182	183	186	186	183	183	177

Figure 3: Original Image matrix

Divide ‘m’ and ‘n’ by three.If ‘m’ and ‘n’ are not divided by three, do zero padding to make m and n divisible by 3.This padding may tends to cause some error which is removed when original image is acquired back after decompression procedure. The original image is then divided into non-overlapping squares by starting with first pixel and selecting 8 adjacent pixels such that square is formed and so that, no pixel is missed.Consider an original Image in matrix form which is of the size (256, 256). The starting pixel of the image is at position (1, 1). Figure 3 represents a part of this matrix. To compress it by the above discussed method first check whether 256 is divisible by 3. If not then pad the original image by 2 rows and 2 columns. The non-overlapping squares (sets) are formed starting from the first pixel. The

center pixels boundaries are highlighted as shown in Figure 4 and are selected and stored in a new matrix, say *N* shown in Figure 5.

	1	2	3	4	5	6	7	8	9	10
1	194	173	188	172	179	178	195	191	199	183
2	178	177	177	177	178	178	182	185	184	182
3	180	183	185	181	175	172	175	180	182	182
4	190	188	185	190	174	165	167	176	180	181
5	188	186	183	180	174	165	167	176	180	181
6	180	180	182	182	181	181	182	181	175	169
7	194	173	168	172	176	178	185	191	199	193
8	178	177	177	177	178	178	182	185	184	182
9	180	183	186	181	175	172	176	180	182	182
10	174	177	180	182	183	186	186	183	183	177

Figure 4: Original Image in matrix form.

	1	2	3	4	5	6	7	8	9	10
1	177	178	185	184	181	180	167	178	193	179
2	188	177	178	171	181	181	173	173	187	182
3	176	176	193	184	183	178	183	178	184	199
4	167	171	175	188	175	185	164	191	176	178
5	182	160	170	176	181	174	178	170	169	176
6	173	188	167	169	174	185	171	189	170	177
7	176	181	188	180	167	177	182	178	184	177
8	172	175	160	174	168	187	182	180	179	178
9	172	174	192	176	175	175	184	176	180	183
10	188	168	179	178	176	175	184	178	181	173

Figure 5: Compressed Image Matrix, N

As it can be seen in Figure 4 pixel values at position (2,2), (2,5), (2,8) becomes pixel values at position (1,1), (1 ,2), (1,3) respectively in Figure 5. The whole image is compressed in the similar way.

3.2 DECOMPRESSION

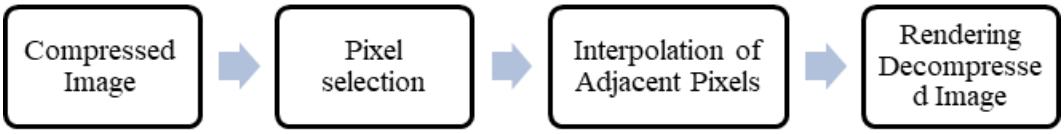

Figure 6: Block Diagram of Decompression

The above given method involves application of second order interpolation over the compressed image. Any pixel in the image has two values. First its position (x, y) in the image and second is its gray value G at that position. Now let us assume that compressed image is of size (m′, n′) and starts from (1, 1). Acquire the first pixel, P_1, which is at position (1, 1) in the compressed image. Consider we are moving in x direction. The next pixels chosen are P_2 and P_3 which are at position (1, 2) and (1, 3) in the compressed image. Obtain the gray values of P_1, P_2 and P_3 which are G_1, G_2 and G_3 respectively. Figure 4 shows us that originally two pixels were present between any two given adjacent pixels of compressed image. Interpolate two pixels between P_1 and P_2 and two between P_2 and P_3 as follows,

1	2	3	4	5	6	7	8	9
P_1	IP_1	IP_2	P_2	IP_3	IP_4	P_3	IP_5	IP_6

Figure 7: Two pixels interpolated

Applying Newton's divided difference formula on table below we get our interpolation formula to find adjacent unknown pixels.

ith (positions)	f_i	Δf_i	$\Delta^2 f_i$
1	G_1	$\Delta G_1 = G_2 - G_1 /(4-1)$	$\Delta^2 G_1 = \Delta G_2 - \Delta G_1/(7-1)$
4	G_2	$\Delta G_2 = G_3 - G_2 /(7-4)$	
7	G_3		

Table 1: Newton Forward Divided Difference Table

The second order curve passing through these points is given by the following equation (1),

G = ((((a-(2*b)+c)*(o*o))+(((16*b)-(11*a)-(5*c))*o)+((28*a)-(14*b)+(4*c)))/18) (1)

where, a=P_1, b=P_2 and c=P_3

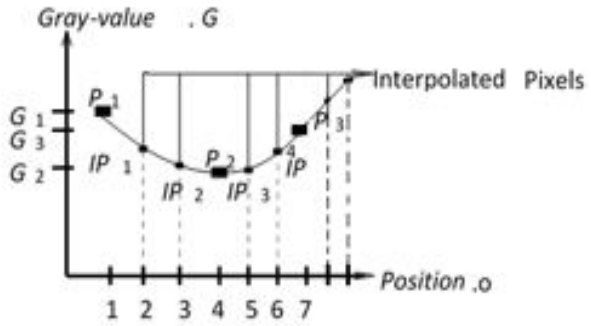

Figure 8: Derieved Curve

We use Equation (1) to Calculate the values of the six interpolated pixels, say IP_1, IP_2, IP_3,IP_4, IP_5 and IP_6 which are at position 2, 3, 5, 6, 8 and 9 using pixel values of P_1, P_2 and P_3 respectively.IG_1, IG_2, IG_3, IG_4, IG_5 and IG_6 are the calculated gray values of IP_1, IP_2, IP_3, IP_4, IP_5 and IP_6 respectively. Then Store G_1, IG_1, IG_2, G_2, IG_3, IG_4, G_4, IG_5 and IG_6 in a new matrix at positions corresponding to the pixel's position in the decompressed image, i.e., at (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), and (1, 9) as shown in figure 9.

1	2	3	4	5	6	7	8	9
G_1	IG_1	IG_2	G_2	IG_3	IG_4	G_3	IG_5	IG_6

Figure 9: Gray values of interpolated pixels

Now select next three pixels in x-direction as P_1, P_2 and P_3 and interpolate the adjacent pixels by the same procedure as discussed above. Repeat the process in x-direction until the whole image is decompressed in this direction.Similarly, acquire the First pixel, P_1, which is at position (1, 1) in the compressed image.Consider we are moving in y direction. The next pixels chosen are P_2 and P_3which are at position (2, 1) and (3, 1) in the compressed image. Obtain the gray values of P_1, P_2 *and* P_3which are G_1, $G_2$and G_3respectively. Interpolate two pixels between P_1 and $P_2$and two between $P_2$and P_3.Similarly, Repeat the process in y-direction until the whole image is decompressed in this direction.For example let us consider the compressed matrix of Figure 5. Take first three pixels at positions (1, 1), (1, 2) and (1, 3). As we are move in horizontal direction we mark 1, 4, 7 on o− axis and 176, 176 and 193 on G−axis respectively for the three points. Now by Using the second order equation we find six new points on this curve corresponding to values $o_2 = 2$, $o_3 = 3$, $o_5 = 5$, $o_6 = 6$, $o_7 = 7$ and $o_8 =$

8 on o − axis. The values come out to be 174, 174, 180, 185, 184 and 184. These six values are put in a new matrix as shown by dashed arrows in Figure 10. This procedure is continued in x- direction so that, whole image is decompressed in horizontal direction. Then the same method is repeated vertically on compressed matrix as shown by solid arrows in Figure 10 and then we obtain the final matrix.

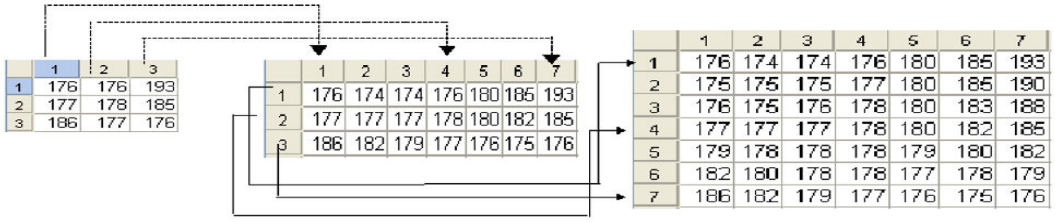

Figure 10: Compression to Decompression.

4. PERFORMANCE EVALUATION OF IMAGE COMPRESSION

4.1 Compression ratio= original image size divided by compressed image size. CR=n1/n2 …….(2)

4.2 Mean Square Error= MSE is the squared error between the compressed image and original image

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

I (i, j) is original image, K (i, j) is decompressed image and m, n are the image dimensions

MSE=(1/(m*n))*sum(sum((original-decompressed).^2))……..(3) where, (‘m’ x ‘n’) are rows and columns of image and original is the input image and decompressed is the final degraded image. The mean squared error is comparing our original image pixel values to degraded image pixel values.

4.3 PSNR is defined as the, ratio of most extreme conceivable energy of image signal and power with noise of same image. In logarithmic decibel scale, PSNR is usually expressed.

PSNR=10*log 10(255^2/ MSE) (4)

4.4 SNR is defined as the, ratio of the signal power of the image to the noise power of the image.

4.5 Computational time- Compression and decompression is the amount of time required for compressing and decompressing an image.

Time=Compression Time+ Decompression Time (5)

5. RESULTS

The proposed method is evaluated using some different images.

1. Result and Comparison

Original Image1(83kb) Compressed Image (5.52kb) Decompressed Image (30.9kb)

Figure 11: Compression and Decompression of Image 1

Original Image2 (92kb) Compressed Image (6.72kb) Decompressed Image (40.6kb)

Figure 12: Compression and Decompression of Image 2

IMAGE	TECHNIQUE	COMPRESSION RATIO
IMAGE 1 83KB	JPEG	11.84
	JPEG 2000	13.26
	INTERPOLATION	15
IMAGE 2 92.9KB	JPEG	8.14
	JPEG 2000	9.12
	INTERPOLATION	14
IMAGE 3 185KB	JPEG	6.55
	JPEG 2000	7.34
	INTERPOLATION	27

Figure 13: Comparison with JPEG and JPEG 2000

IMAGE	BIT DEPTH	ORIGINAL SIZE	COMPRESSED SIZE	DE-COMPRESSED SIZE	CR	MSE	PSNR (dB)	SNR (dB)	TIME (s)
IMAGE 1	8	83KB	5.52KB	30.9KB	15	440	2.16e+01	11.90	0.5422
IMAGE 2	8	92.9KB	6.72KB	40.6KB	14	444	2.165e+01	14.05	0.57738
IMAGE 3	8	185KB	6.75KB	43.58KB	27	368	2.24e+01	16.86	0.4800

Figure 14: Final Results

We can directly say, Compression ratio of Interpolation method is greater than JPEG and JPEG 2000.From the results we can see that when the image size is large then our algorithm provides better compression then Lossy jpeg. But for small images, to maintain the quality of decompressed image and so that the details are not lost our compression method compresses less.

2. Universal Decompression Algorithm

The proposed decompression algorithm can also be used to enlarge any given grayscale image. Figure shows cropped part of Lena image. When our reconstruction algorithm is applied on it we obtain the image displayed below.

Figure (a): Original Image Figure (b): Enlarged Image

Figure 15: Image Enlarged

As our method is Lossy, it leads to some degradation of the reconstructed image.

6. CONCLUSION

This project proposed a compression algorithm to reduce the image size and which is inexpensive. The compression algorithm is implemented using the MATLAB 2015a software on different test images. The decompression algorithm, based on interpolation, is then implemented on the compressed image. The decompression algorithm enlarges any given grayscale image. We can see from the results that when compared to JPEG, it provides us better compression ratio. From the results it can be seen that Compression ratio of Interpolation method is higher than JPEG and JPEG 2000. The suggested method is suitable for applications which cannot handle too many complex calculations, like mobile phones without compromising on their quality, video downloading such as in YouTube and Whatsapp for sending messages.

7. FUTURE SCOPE

The present work mainly focuses on the effective image compression and decompression using adaptive interpolation on images. In proposed compression technique there is good scope for future improvement. Quality of Image and Compression ratio further can be improved. In the near future, more compression Techniques will be considered which can compress with higher compression ratio and maintain the quality of image. The future enhancement of this work would be to apply it for videos such as on YouTube application and text data compression.

REFERENCES

[1] Barbhuiya, AHM Jaffar Iqbal, Tahera Akhtar Laskar, and K. Hemachandran. "An approach for color image compression of JPEG and PNG images using DCT and DWT." Computational Intelligence and Communication Networks (CICN), 2014 International Conference on. IEEE, 2014.

[2] Xiao, Jianping, et al. "Adaptive interpolation algorithm for real-time image resizing." Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on. Vol. 2. IEEE, 2006.

[3] Mahajan, Shruti H., and Varsha K. Harpale. "Adaptive and Non-adaptive Image Interpolation Techniques."Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE, 2015.

[4] Hemalatha, M., and S. Nithya. "A Thorough Survey on Lossy Image Compression Techniques." International Journal of Applied Engineering Research 11.5 (2016): 3326-3329.
[5] Arora, Sunny, and Gaurav Kumar. "Review of Image Compression Techniques." (2018).
[6] Jain, Anil K. "Image data compression: A review." *Proceedings of the IEEE* 69.3 (1981): 349-389.
[7] Vilovic, Ivan. "An experience in image compression using neural networks." *Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006 focused on*. IEEE, 2006.
[8] R.C.Gonzalez, R. E. Woods,Digital Image Processing,2nd Edition.