Scaling Transform Methods For Compressing a 2D Graphical image

Ms. A. J. Rajeswari Joe

Research Scholar, Bharathiyar University
Assistant Professor, Department of MCA,
GSS Jain College for Women, Chennai

rajoes@yahoo.com

Dr. N. Rama

Research Supervisor, Bharathiyar University
Assistant Professor, Department of Computer Science,
Presidency College, Chennai

ABSTRACT

*Transformation is a process of converting the original picture coordinates into a different picture coordinates either by adding some values with original coordinates(Translations) or Multiplying some values with original coordinates(called Linear transformations like rotation, reflection, scaling, and shearing**).** In this paper, we compress a two dimensional picture using 2D scaling transformation. In the several scenarios, the utilization of this technique for image compression resulted in comparable or better performance, when compared to the Different modes of image transformations. In this paper We tried a new code for compressing an 2d gray scale image using scaling transform methods. Matlab concepts are applied to compress the image. We have plan to apply The techniques and develop a code for compressing a 3d image.*

KEYWORDS

Scaling factors (compression factors), two dimensional scaling transformation, decompression, experimental result,.

1. Introduction

Scaling is the process of expanding or compressing the dimensions (i.e., size) of an object. An important application of scaling is in the development of viewing transformation, which is a mapping from a window used to clip the scene to a view port for displaying the clipped scene on the screen. In Euclidean geometry, changing the size of an object is called a scale. We scale an object by scaling the x and y coordinates of each vertex in the object uniform scaling (or isotropic scaling,) is a linear transformation that enlarges (increases) or shrinks (diminishes) objects by a scale factor that is the same in all directions. The result of uniform scaling is similar (in the

geometric sense) to the original. A scale factor of 1 is normally allowed, so that congruent shapes are also classed as similar. More general is scaling with a separate scale factor for each axis direction. Non-uniform scaling (anisotropic scaling) is obtained when at least one of the scaling factors is different from the others; a special case is directional scaling or stretching (in one direction). Non-uniform scaling changes the shape of the object; e.g. a square may change into a rectangle, or into a parallelogram if the sides of the square are not parallel to the scaling axes (the angles between lines parallel to the axes are preserved, but not all angles). In this paper we have compressed lena's image which is a grayscale 2d image using Matlab coding. 2-D Dimensional Cosine Transform is applied on Lena's image (352*352) as a test image. DCT is applied to lena's image through scaling factor 2, scaling factor 4 and scaling factor 8, and the result is analysed. The results are then compared with various compression methods. We used Peak Signal-to Noise Ratio (PSNR) and Mean Square Error (MSE) to observe the quality of the compressed image and original image.

2.Scaling factors(compression factors)

A scaling transformation alters the size of an object. The operation is accomplished by multiplying each coordinate by scaling factors S_x, S_y . When (S_x S_y,) < 1 the object or image is compressed. Where S_x **,** S_y are the scaling factors along each axis with respect to the local coordinate system of the model. The scaling transformation allows a transformation matrix to change the dimensions of an object by shrinking or stretching along the major axes centered on the origin. It should be noted that the scaling is always about the origin along each dimension with the respective scaling factors. This means that if the object being scaled does not overlap the origin, it will move farther away if it is scaled up, and closer if it is scaled down. Scaling with respect to a selected fixed position (S_x S_y) can be represented with the following transformation sequence:

1. Translate the fixed point to the origin
2. Scale the object relative to the coordinate Origin
3. Translate the fixed point back to its original Position.

3. Two Dimensional Scaling

Let S_x , S_y be the scale in the positive x and y directions respectively. Then the scaled vertex is given by

$$x' = x \cdot S_x \qquad (1)$$

$$y' = y \cdot S_y \qquad (2)$$

If, S_x = S_y=S then it is said to be *homogenous* or *uniform* scaling[4] transformation that maintains relative proportions of the scaled objects. The Magnification factor is |s|. All points move **s** times away from the origin. If |s| < **1**, all the points move towards the origin, or demagnetized.

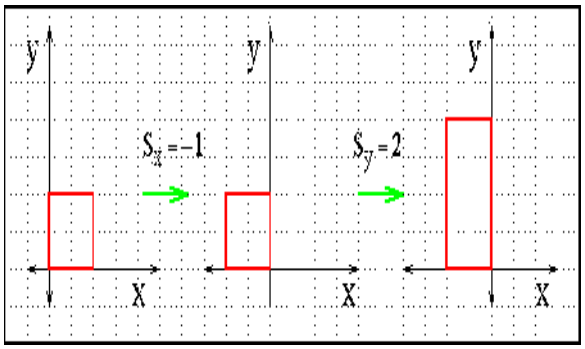

Fig1. Scaling the original picture with scale factors S_x=-1, S_y= 2

If $S_x \neq S_y$ then the scaling is called differential scaling. When either S_x or S_y equals to one, simplest differential scaling. 2-Dimensional scaling in the matrix form is given by

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

In the case where $S_x = S_y = k$, scaling increases the area of any surface by a factor of k^2 and the volume of any solid object by a factor of k^3. Such a scaling changes the diameter of an object by a factor between the scale factors, the area by a factor between the smallest and the largest product of two scale factors, and the volume by the product of all two.

We can represent a triangle, shown in matrix form, [5] using homogeneous coordinates of the vertices as :

A 0 0 1
B 1 1 1
C 5 2 1

By choosing the scaling factor s=1/2

The matrix of scaling for compressing an image is:

$$(S_{(1/2,1/2)}) \qquad \begin{matrix} ½ & 0 & 0 \\ 0 & ½ & 0 \\ 0 & 0 & 1 \end{matrix}$$

Multiply original matrix coordinates with matrix of scaling results the coordinates of either compressed image or decompressed image based on the scaling factors selected.

So the new coordinates A'B'C' of the scaled triangle ABC can be found as:

A 0 0 1	½ 0 0	A 0 0 1
B 1 1 1	0 ½ 0	B ½ ½ 1
C 5 2 1	0 0 1	C 5/2 1 1

Thus, the new coordinates are A'=(0,0), B'=(1/2, 1/2), C'= (5/2, 1)

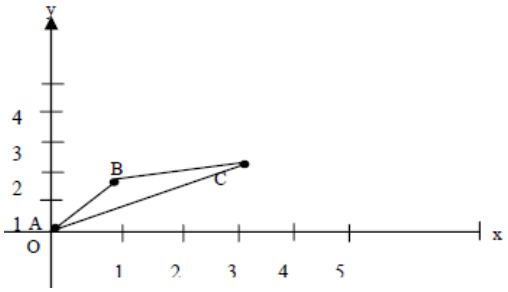

Figure2: Object after scaling with $S_x = S_y = 1/2$ (Compressed picture)

4. Decompression

The inverse scaling matrix[4] is obtained by replacing *sx* and *sy* with 1/*sx* and 1/*sy*.

$$[x^{-1} \quad y^{-1} \quad 1] = [x \quad y \quad 1] \begin{bmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

The scaling discussed so far has the origin (0,0) as the fixed point and scaling is about the origin. When an object is scaled, it is moved *sx* and *sy* times away from the origin. It is also possible to have any arbitrary point as a fixed point, and scale about that point.

We can represent a triangle, shown in matrix form, using homogeneous coordinates of the vertices as :

A 0 0 1
B 1 1 1
C 5 2 1

choosing scaling factor s=2

The matrix of scaling for decompression is:

$$(S_{(2,2)}) \qquad \begin{matrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{matrix}$$

So the new coordinates A'B'C' of the scaled triangle ABC can be found as:

A 0 0 1 2 0 0 A 0 0 1
B 1 1 1 0 2 0 B 2 2 1
C 5 2 1 0 0 1 C 10 4 1

Thus, A'=(0,0), B'=(2,2), C'= (10, 4)

The following figure (3) shows the effect of scaling with s_x=s_y =2.

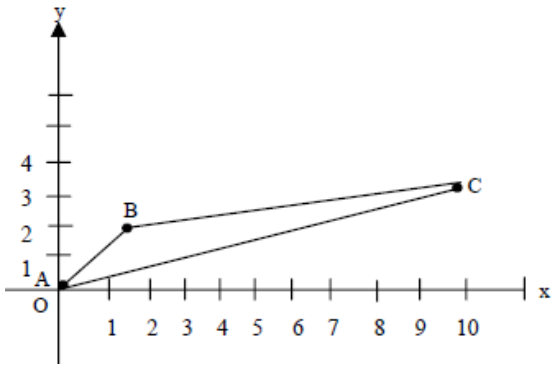

Figure2: Object after scaling with $1/s_x$ and $1/s_y$. (i.e) $S_x = S_y = 2$
(DeCompressed picture)

5. Related Work

5.1 The Discrete Cosine Transform

DCT Attempts to decorrelate the image data after decorrelation each transform coefficient can be encoded without dropping off compression efficiency[7]. DCT separates images into parts of different frequencies where less important frequencies are discarded through quantization and important frequencies are used to retrieve the image during decompression. Compared to other input dependent transforms, DCT has many advantages: (1) It has been implemented in single integrated circuit; (2) It has the ability to pack most information in fewest coefficients; (3) It minimizes the block like appearance called blocking artifact that results when boundaries between sub-images become visible

5.2 The One-Dimensional DCT

The DCT of a list of n real numbers s(x), [8]
where x=0, 1, ….., n-1, is the list of length n given by:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)u}{2N}\right]$$

For u= 0, 1, 2, … N-1.
Similarly, the inverse transform is [5]defined as

$$f(x) = \sum_{x=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right]$$

Thus, the first transform coefficient is the coefficient is the average value of the sample sequence

5.3 The Two-Dimensional DCT

The Discrete Cosine Transform (DCT) is one of many transforms that takes its input and transforms it into a linear combination of weighted basis functions. These basis functions [11]are commonly the frequency. The 2-D Discrete Cosine Transform is just a one dimensional DCT applied twice, once in the x direction, and again in the y direction. One can imagine the computational complexity of doing so for a large image. Thus, many algorithms, such as the Fast Fourier Transform (FFT), have been created to speed the computation.

The DCT computes the i, j^{th} entry of the DCT of an image.

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases}$$

N is the size of the block that the DCT is applied on. The equation [7]calculates one entry (i, j^{th}) of the transformed image from the pixel values of the original image matrix. For the standard 8*8 block that JPEG compression uses, N equals * and x and y range from 0 to 7. Therefore D (I,j) would be as in equation:

$$D(i,j) = \frac{1}{4} C(i) C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right]$$

Because the DCT uses cosine functions, the resulting matrix depends on the horizontal and vertical frequencies. Therefore an image block with a lot of change in has a very random looking resulting matrix of a large value for the first element and zeroes for the other element.

6. Experimental Result

- Original image is divided into blocks of 8 x 8.
- Pixel values of a black and white image range from 0-255 but DCT is designed to work on pixel values ranging from -128 to 127 .Therefore each block is modified to work in the range.

- DCT is applied to each block by multiplying the modified block with DCT matrix on the left and transpose of DCT matrix on its right.
- Each block is then compressed through quantization.
- Quantized matrix is then entropy encoded.
- Compressed image is reconstructed through reverse process.
- Inverse DCT is used for decompression.

6.1 Quantization

Quantization is achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible. A quantization matrix is used in combination with a DCT coefficient matrix to carry out transformation. Quantization is the step where most of the compression takes place.DCT really does not compress the image because it is almost lossless. Quantization makes use of the fact that higher frequency components are less important than low frequency components. It allows varying levels of image compression and quality through selection of specific quantization matrices. Thus quality levels ranging from 1 to 100 can be selected, where 1 gives the poorest image quality and highest compression, while 100 gives the best quality and lowest compression. As a result quality to compression ratio can be selected to meet different needs. JPEG committee suggests matrix with quality level 50 as standard matrix. For obtaining quantization matrices with other quality levels, scalar multiplications of standard quantization matrix are used. Quantization is achieved by dividing transformed image matrix by the quantization matrix used. Values of the resultant matrix are then rounded off. In the resultant matrix coefficients situated near the upper left corner have lower frequencies .Human eye is more sensitive to lower frequencies .Higher frequencies are discarded. Lower frequencies are used to reconstruct the image[18].

6.2. Entropy Encoding

After quantization, most of the high frequency coefficients are zeros. To exploit the number of zeros, a zig-zag scan of the matrix is used yielding to long string of zeros.[18].

To evaluate the performance of the proposed scheme,[13] 2-D DCT is applied on Lena's image (352*352) as a test image. DCT is applied to lena's image through scaling factor 2, scaling factor 4 and scaling factor 8, and observe the result. The results are then compared with various compression methods. We used Peak Signal-to Noise Ratio (PSNR) and Mean Square Error (MSE) for a compressed image. This ratio is often used as a quality measurement between the original and compressed image.

To compute the PSNR, first calculate the mean-squared error using the following equation:

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

Where x (m, n) and y (m, n) are the two images of the size m*n. In this case x is the original image and y is the compressed image

$$PSNR(db) = 10 * \log\left(\frac{255^2}{MSE}\right)$$

Figure :3 Original image

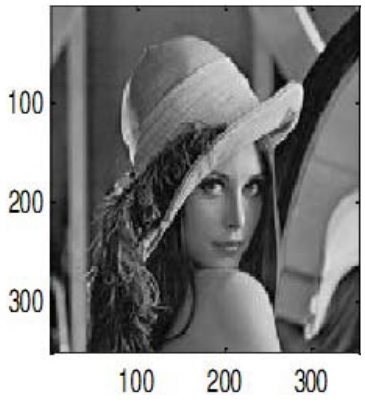

Figure :4 scaling factor=2

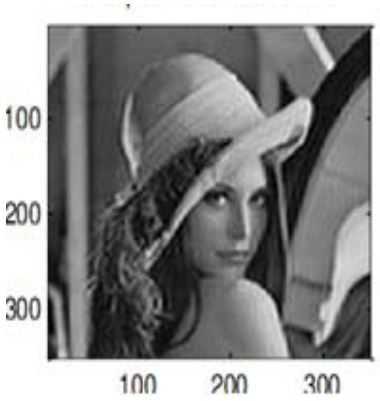

Figure : 5 scaling factor=4

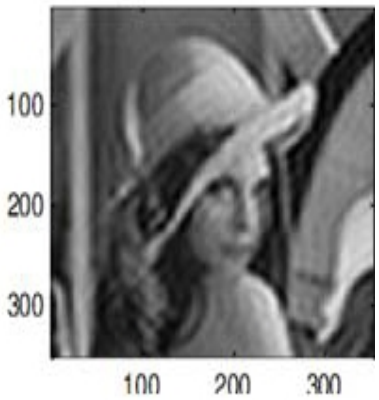

Figure : 6 scaling factor=8

Table 1: Ratio of MSE & PSNR

	Scaling factor=2	**Scaling factor=4**	**Scaling factor=8**
MSE	48.6	43.3	39.0
PSNR	0.89	3.03	8.2

7. Conclusion

From the above experiments, high compression ratio and better image quality accomplished which is better than existing methods. This paper has concentrated on development of efficient and effective algorithm for still image compression. Fast and lossless compression algorithm using 2-D DCT is developed. From the Results it is observed that the encoding time is reduced with little degradation in image quality compare to anticipated method. Compression ratio is also increased, while comparing the wished-for method with other methods. Our future work involves improving image quality by increasing PSNR value and lowering MSE value.

References

[1] Intoduction to Data compression by Khalid sayood, Third edition, (Morgan Kaufmann series in multimedia information and systems)

[2] W. Yan, P. Hao, C. Xu, ”Matrix factorization for fast DCT algorithms”,IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, 2006.

[3] http://vedyadhara.ignou.ac.in/wiki/images/5/58/B2U1mcs-053.pdf

[4] Mathematical elements for computer graphics By David. F. Rogers, J.Alan Adams, McGraw-Hill Science/Engineering/Math; 2 edition (August 1, 1989)

[5] Computer graphics-Donald Hearn, M. Pauline Baker, Prentice-Hall, 1986, University of Minnesota, 20 Jan 2010
[6] http://en.wikipedia.org/wiki/Data_compression
[7] Andrew B. Watson, NASA Ames Research, "Image Compression Using the Discrete Cosine Transform", Mathematica Journal, 4(1),1994, p. 81-88.
[8] Nageswara Rao Thota, and Srinivasa Kumar Devireddy, "Image Compression Using Discrete Cosine Transform", GeorgianElectronic Scientific Journal: Computer Science and Telecommunications 2008|No.3(17).
[9] Swastik Das and Rashmi Ranjan Sethy, "A Thesis on Image Compression using Discrete Cosine Transform and Discrete Wavelet Transform", Guided By: Prof. R. Baliarsingh, dept of Computer Science & Engineering, National Institute of Rourkela.
[10] F. Feing, S. Winograd, "Fast algorithms for the discrete cosine transform", IEEE Transactions on Signal Processing, vol. 40, no. 9,September, 1992.
[11] N. Ahmed, T. Natarajan, and K.R. Rao,"Discrete Cosine Transform", IEEE Trans. Computers, 90-93, Jan 1974.
[12] Wallace, G. 1991. The JPEG still picture compression standard communications of the ACM 34(4): 30-44.
[13] Chih-chang chen, Oscal T-C. Chen "A Low complexity computation scheme of discrete cosine transform and quantization with adaptation to block content", Dept of Electrical Engineering, 2000 IEEE.
[14] Chih-chang chen, Oscal T-C. Chen "signal and Media Laboratories", Dept of Electrical Engineering, 2000 IEEE.
[15] Yung-Gi Wu, "Medical Image compression by sampling DCT coefficient", 2002 IEEE.
[16] V. Kober, "Fast algorithm for the computation of sliding discrete Hartley transform", IEEE Transactions on Signal Processing, vol.55, Issue 6, pp.2937-2944, June, 2007.
[17] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 2nd Edition, Pearson Education, ISBN-81-7808-629-8, 2002.
[18] Ken cabeen and Peter Gent,"Image Compression and the Descrete Cosine Transform"Math 45,College of the Redwoods.