Dirichlet Tessellation's technique to compress a true color image using a Lossy compression

Amani Y. Noori, Suhad F. Majeed
University of Mustansiriyah, College of Basic Education, Computer science department
E-mail: snowhite105@gmail.com
E-mail: soledad.ahmed@yahoo.com

Abstract: In this paper, we show how to use the concept of Dirichlet Tessellations to compress, store and reconstruct an image without affecting on its dimensions and represent it with an acceptable quality, where a true color image has compressed by 60.05% with mean square error (MSE) = 9.6081 which represents the error between the restored image and the original image, and peak signal-to-noise ratio (PSNR) =38.3044 dB which represents the similarity between the restored image and the original image, using MATLAB R2017a. Dirichlet Tessellation has simply defined as dividing the space into geometric shapes by generating finite set of distinct points, each shape contains one of the distinct points and comprising that part of the space nearer to that distinct point than to any of the other points. We have used two algorithms for image compression, First algorithm selects set of distinct points distributed uniformly in an image and store their locations along with pixel values. In the second algorithm random selection of distinct points distributed uniformly in an image and store their locations along with pixel values. In order to reconstruct the image, Saved distinct points placed at their corresponding locations in a new image that is formed, where two algorithms used, the first algorithm based on the concept of a growing region. It's Region-Based image segmentation method, by checking the pixels adjacent to the saved distinct points and delimiting whether the pixels should add to the regions of saved distinct points depending on the region's membership criteria such as pixel intensity. The second algorithm uses one of the Dirichlet Tessellations characteristics, Which divides an image into polygonal regions based on the distinct points that saved, Each pixel in the confined plane of saved distinct points will have the same characteristics of this point, This is done by taking each pixel in an image and calculating the minimum distance between pixel location and saved sites using the distance equation, This process repeated until each pixel assigned its value and specifying all color regions in the image.

Keyword: Dirichelt Tessellations concept, edge detection, lossly compression
1. Introduction

Image compression is an effective technique for reducing the size of a graphical file without degrading the quality of the image and reduces the storage requirement area. It additionally reduces the time required for images to send over the Internet or downloaded from web pages [1]. There are two types of compression lossless compression and lossy Compression. The image compression method used relies upon the quality wanted for the restored image [2]. In case the image compression technique should provide a high-quality result with no losing data, A Lossless Compression method is used. This method used where a high rung of accuracy is an absolute necessity for the restored image, this is often used in the compression of texts and medical images [3]. When the restored image loses some of its data, A Lossy Compression method is used. This method based on reducing duplicate data while preserving the basic details of the image [4], Where the image is restored without noticing of any change in its important structural properties, The restored image somewhat identical to the original image with an acceptable quality [5], This is usually used for compressing images and videos. In this paper, we propose a method of Lossy Compression based on the concept of Dirichlet Tessellations.

2. Related Work

Lossy compression is a technique for reducing excess data in an image and recreate it in an efficient form, which means to represent the image with less number of bits per pixel in order to store or transfer while preserving the edges and accurate information in the image [6] [7]. There are several different techniques of lossy compression. In 1987, Fractal compression invented by Barnsley and Alan Sloan. It is a very helpful technique in the field of image compression. It is an asymmetrical compression method, which means it takes more time/effort to compress an image rather than decompressing it. This technique depends on the existence of self-symmetry in an image. It has the benefit of faster decompression speed while giving equal or better compression ratio. However, the lengthy compression stepped remains the main drawback of this technique, which will preclude it from being used in applications that need to send compressed images with minimal delay [8].In 1992, JPEG standard created by Joint Photographic Experts Group based on the discrete cosine transform (DCT). It is a symmetrical compression method, That means the compression and the decompression processes take approximately the same amount of time/effort. The JPEG compression algorithm utilized for the web, it works better on images with smooth variations of color. It divides an image into 8 × 8 blocks, then it converts the intensity data of the pixels into their frequency space equivalents. The result is a set of 64 cosine functions with various amplitudes. It is compressed by discarding higher frequency cosine terms [9]. The major drawback of DCT, while inputs from reprocessed 8 × 8 blocks are integer-valued, the output values are usually real-valued. So, we need a quantization step to make some decisions about the values in each DCT block and produce output that is an integer-valued. Furthermore, it is not appropriate for line drawings and other textual or iconic graphics, where the sharp contrasts between adjacent pixels can cause noticeable artifacts [10]. In this paper, we use another new technique for lossy compression. This technique depends on the concept of Dirichlet Tessellation. It provides some advantages in image fidelity on both of fractal compression and JPEG standard. It also has a higher compression ratio.
Applying Sobel Operator to eliminate useless information and capturing important events of image

Generate image map by using Dirichlet Tessellation technique

Using a generated image map to reconstruct the basic structure of the image

Tiling of the new image that reconstructed by using Distance Equation and characteristics of Dirichlet Tessellation

Block Diagram of Compression and Decompression

3. Compression The Image

To compress the image we need to eliminate some information, Especially unnecessary information without affecting the general structure of the image.

The first step reduces an image by using Dirichlet Tessellations technique, this technique used to generate image map by choosing distinct points in an image and store their locations along with pixel values (RGB) of these points, using two algorithms of Dirichlet Tessellations.

- The first algorithm chooses pixel uniformly.
- The second algorithm chooses pixel randomly depending on edges detection algorithm.

Input: A true-color image (24 bits per pixel). The dimensions of the image is 341 x 512 x 3 uint8, As shown in the figure (1).

```matlab
img = imread ('cat.png');
s1 = first algorithm (img);
s2 = second algorithm (img);
```

Output: s1, s2 Set of certain pixels saved as distinct points (represents Image Map).
3.1. First Algorithm For Compression

1. Choose distinct points uniformly throughout the image by moving a specific distance from one pixel to another.
2. Perform image compression by storing the locations of pixels along with their values (RGB).

The following codes perform first algorithm for compression.

```matlab
Function vs = first algorithm ( img )
    r = 1 ;
    for i = 4 : 7 : size(img ,1)
        for j = 4 : 7 : size(img ,2)
            vs (r , :) = [ i   j   double (img ( i , j , 1)) / 255    double ( img (i , j , 2)) / 255
double ( img ( i , j , 3)) / 255] ;
            r = r + 1 ;
        end
    end
end
```

The problem with this algorithm loss a lot of image details when we decompressed, as shown in figure (2) because it choose pixels uniformly without focus on regions containing more details, for this reason we use the second algorithm to solve this problem.
3.2. Second Algorithm For Compression

1. Convert RGB image to a grayscale image, as shown in the figure (3).
2. Finding edge detection of the image by using edge detector algorithm, as shown in the figure (4).
3. Choose certain pixels in the image by using this condition.

   "edgeimg(i,j) >= 70"

Perform image compression by saving them as distinct points. The following codes perform second algorithm for compression. In this algorithm, we get large numbers of generating sites, but it is better than the first algorithm because it preserves the important structural properties of an image.

```matlab
Function vs = second algorithm ( img )
grayimg = rgb2gray ( img ) ;
f = edgedetector ( grayimg , thrsh ) ;
edgeimg = edgedetector ( f , 0 ) ;
r = 1 ;
for i = 1 : size ( img , 1 )
    for j = 1 : size ( img , 2 )
        if   edgeimg (i , j) >= 70
            vs (r , :) = [ i    j    double( img (i,j,1)) /255
                          double (img (i , j , 2)) /255
                          double ( img ( i , j ,3 )) /255 ] ;
        end
    end
    r = r + 1 ;
end
```

Figure (2) Image Compressed by using First Algorithm

Figure (3) Grayscale Image
4. **Edge Detection**

It is just a process of locating the edges of objects in an image. It is a very important step to comprehend image features [11].

Edges usually indicate the items in the image where the gray value changes considerably from one pixel to another. It symbolizes regions in the image with sharp intensity variation representing object boundaries. The reason behind detecting sharp edges in the image is for capturing important events [12].

An edge detector reduces the amount of data to be processed and remove useless information, while conserving the important structural properties of an image. It is a good way to solve the problem of the large space occupied by the image in computer memory and transmission over the Internet [13]. Different methods used to detect edges in image processing among of these we use Sobel operator.

The Sobel operator applies a 2-D spatial gradient metering on the image. It uses a pair of horizontal and vertical gradient masks, Which are 3x3 for edges detection function [14]. The Sobel detector is incredibly delicate to noise in an image, It effectively highlights them as edges, Smoothing the image.

**Sobel mask for X - direction vertical Gx**

\[
G_x = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{bmatrix}
\]

**Sobel mask for Y- direction horizontal Gy**

\[
G_y = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{bmatrix}
\]

The magnitude of the vector \( \Delta \varepsilon \)

\[
\Delta \varepsilon = \text{mag} = \sqrt{G_x^2 + G_y^2}
\]

4.1. **Edge Detector Algorithm For Image**

1. Convert grayimg to double.
2. The filter mask is $3 \times 3$ to process the last position (row) of the image. We have to do this ($(grayimg, 1) - 2$). And we will do the same for the last position (column) of the image ($(grayimg, 2) - 2$). Thus the image's borders are left.

3. Using the mask $G_x$ for $X$ direction, $G_y$ for $Y$ direction to obtain the gradient of the image by finding the magnitude of the vector $\Delta \epsilon$ [14].

4. Threshold the image, display the logical image, if the Sobel gradient values are lesser than the threshold value, then replace it with threshold value [15], As shown in the figure (4). The following codes perform edge detector algorithm for image.

```matlab
Function edgeimg = edgedetector ( grayimg , thrsh )

d = double ( grayimg ) ;
For i = 1 : size (grayimg , 1) - 2
    For j = 1 : size (grayimg , 2) - 2
        Gx = ((2 × d (i+2 , j+1) + d (i+2 , j) + d (i+2 , j+2)) - (2 × d (i , j+1) + d (i , j) + d (i , j+2))) ;
        Gy = ((2 × d (i+1 , j+2) + d (i , j+2) + d (i+2 , j+2)) - (2 × d (i+1 , j) + d (i , j) + d (i+2 , j))) ;
        grayimg (i , j) = sqrt (Gx .^ 2 + Gy .^ 2) ;
    end
end
edgeimg = max ( grayimg , thrsh ) ;
edgeimg ( edgeimg == round ( thrsh )) = 0 ;
```

Figure (4) Edge Detection of an Image by using Sobel Gradient
5. Decompression The Image

The concept of decompression the image is simple, we only need to define some features that are distinguishing regions in the image, usually depends on the original image to know the important information for segmentation and the result that we want. In order to reconstruct the image, Saved distinct points placed at their corresponding locations in a new image that formed, where two algorithms had used for decompression.

- The first algorithm based on the concept of growing regions.
- The second algorithm uses one of the characteristics of the Dirichlet Tessellations.

Input: s1, s2 Distinct points have saved as an image map.

\[
\text{[row col ~]} = \text{size (img)};
\]

\[
r1 = \text{growing region (s2, row, col)};
\]

\[
r2 = \text{dirichlet (s2, row, col)};
\]

Output: r1, r2 Image after decompression (Original Image).

5.1. Growing Regions

Growing regions are a simple way to fragment image based on regions. It categorized as an Image fragmentation method based on pixels because it includes choosing of initial pixels (seed points). This technique based on similarity regions or homogeneous regions of the image by comparing each pixel adjacent to the seed points and delimiting whether pixels should add to the region of seed point [16].

The concept is simple, we just need seed points to represent the property we need from the original image. First step place the saved distinct points as initial regions, these regions are growing from these saved distinct points to adjacent points based on the area's membership criteria such as pixel intensity, If adjacent pixels have the same intensity value with the saved points we add them to the regions of saved distinct points. These regions continue growing until they encounter other regions.

5.1.1. Growing Regions Algorithm For Decompression

1. Create an image, the same dimensions of the original image.
2. Place saved distinct points as initial regions in the new image that formed, as shown in the figure (5).
3. Regions are growing from these saved distinct points to adjacent points based on the region's membership criteria, as shown in the figure (6).

The following codes perform growing regions algorithm for decompression.
Function r1 = growing regions (s2, row, col)

```matlab
r1 = zeros (row, col, 3) - 1;
L = size (s2, 1);
for t = 1 : L
    r1 (s2 (t, 1), s2 (t, 2), :) = s2 (t, 3 : 5);
end
n = 1;
while (n == 1)
    n = 0;
    q = r1;
    for i = 1 : row
        for j = 1 : col
            if r1 (i, j, 1) == -1
                n = 1;
            else
                for k = 1 : 3
                    for m = 1 : 3
                        z = k - 2 + i;
                        g = m - 2 + j;
                        if (z <= row) & (z >= 1) & (g <= col) & (g >= 1) & (q (z, g, :) == -1)
                            q (z, g, :) = r1 (i, j, :);
                        end
                    end
                end
            end
        end
    end
    pause (0.01);
    r1 = q;
end
```

Figure (5) Place Saved Sites as Initial Regions
The reconstructed image has some distortions, this algorithm causes excessive blurring at the high contrast edges in the image. As a result, we using Dirichlet algorithm to solve this problem.

5.2. Dirichlet Tessellation

A set of distinct points that divides the space into polygonal regions or geometric shapes, which we call tiles. The resultant space subdivision is known as Dirichlet Tessellation. It is one of the most useful constructs associated with such a point configuration [17] [18].

The concept is tiling of the new image that is formed using geometric shapes, without any interpenetration and no lacunae [19]. Polygons or geometric shapes are generating from saved distinct points. Each polygon defines an effect area around a saved distinct point so that any pixel inside the polygon will have the same characteristics of the saved distinct point of that polygon. This is done by taking each pixel in an image and calculating the minimum distance between a pixel’s location and saved distinct points using the distance equation.

\[ d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \]

This process repeated until each pixel assigns its value and specifying all color regions in the image.

5.2.1. Dirichlet Algorithm For Decompression

1. Create an image, the same dimensions of the original image.

2. Calculate the minimum distance between
   Pixel location in an image and saved distinct points using the distance equation.

3. Each pixel inside the polygon will have the same characteristics of the saved distinct point of that polygon, as shown in the figure (7). The following codes perform Dirichlet algorithm for decompression.
Function \( r2 = \text{Dirichlet} (s2, \text{row}, \text{col}) \)

\[
\begin{align*}
    r2 &= \text{zeros} (\text{row}, \text{col}, 3) ; \\
    L &= \text{size} (s2, 1) ; \\
    \text{For } i &= 1 : \text{row} \\
    &\quad \text{For } j = 2 : \text{col} \\
    &\quad \quad \text{For } k = 1 : L \\
    &\quad \quad \quad d (k) = \sqrt{(i - s2(k, 1))^2 + (j - s2(k, 2))^2} ; \\
    &\quad \quad \text{End} \\
    &\quad [mn, pos] = \text{min} (d) ; \\
    &\quad r2(i, j, :) = s2(pos, 3 : 5) ; \\
    &\text{end} \\
    &\text{pause} (0.001) ; \\
    &\text{end}
\end{align*}
\]

Create an image, the same dimensions of the original image.

Tiling of the new image that has formed using one or more geometric shapes, by using distance equation

\[
d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}
\]

to calculate the minimum distance between pixel locations in an image and saved distinct points.

Each pixel in the confined plane of saved distinct points will have the same characteristics of that points.

Block Diagram of Dirichlet Algorithm for Decompression

We calculate the compression ratio for knowing the quantity of data that produced by the compression algorithm to represent the restored image, where size of a graphical file of the original image was 253579, after the compression 209249, which mean the image has compressed by 60.05% with an acceptable quality and this is a good result.
6. Results And Discussion
For simulating our algorithms, the decompression algorithms have applied on four RGB images. Where dirichlet algorithm provides better results than growing regions algorithm. The results are shown in Figure : 8.1, 8.2, 8.3, in Figure : 9.1, 9.2, 9.3, in Figure : 10.1, 10.2, 10.3 and Table-1. Where compression ratio(CR) represents the ratio between the uncompressed size of the image and compressed size, mean square error (MSE) represents the error between the restored image and the original image, and peak signal-to-noise ratio (PSNR) represents the similarity between the restored image and the original image.

Table-1 shows the comparison between the two algorithms depending on compression ratio (CR), mean square (MSE), peak signal-to-noise ratio (PSNR), for the test images.

| Test Images | dirichlet algorithm | growing regions algorithm |
|-------------|---------------------|---------------------------|
| Cat.png     | CR 60.05% MSE 9.6081 PSNR 38.3044 dB | CR 57.27% MSE 13.9431 PSNR 36.6872 dB |
| Peppers.png | CR 60.24% MSE 23.6356 PSNR 34.3951 dB | CR 56% MSE 33.9850 PSNR 32.817 dB |
| Bird.png    | CR 61.08% MSE 14.8151 PSNR 36.4237 dB | CR 56.73% MSE 19.1415 PSNR 35.311 dB |
| Lena.png    | CR 56.03% MSE 50.8857 PSNR 31.0648 dB | CR 50.68% MSE 77.5581 PSNR 29.2345 dB |
7. Conclusion And Future Work

This research focuses on how to use the concept of Dirichlet Tessellation to compress an image a Lossy Compression method. Where two algorithms had used to compress the image, both of them generated an image map by choosing distinct points from the original image and store their locations along with pixel values, the second algorithm is the best because it preserves the basic details of an image by using edges detector algorithm.

Two algorithms had used for decompression, both of them were using the saved image map to restore the image. Where growing regions algorithm is based on the area's membership criteria such as pixel intensity to reconstruct the colored areas in the image, This algorithm compressed the image by 57.27% with some distortion, where mean square error (MSE)=13.9431 which represents the error between the restored image and the original image, and peak signal-to-noise ratio (PSNR) =36.6872 dB which represents the similarity between the restored image and the original, So we are using a Dirichlet algorithm for decompression, which depends on the characteristics of saved distinct points and distance equation for tiling of the new image that is formed using geometric shapes, Without any interpenetration and no lacunae. The quality of the restored image is acceptable. It compressed by 60.05%, where mean square error (MSE) = 9.6081 and peak signal-to-noise ratio (PSNR) = 38.3044 dB.

In the future work, we will provide an overview of video compression techniques with a focus on techniques that use motion estimation, such as MPEG, True motions estimation technique, and Object-based estimation technique.
References

[1] Vartika Singh A Brief Introduction on Image Compression Techniques and Standards International Journal of Technology and Research Advances Volume of 2013 issue II.
[2] Mohammad Kabir Hossain, Shams MImam, Khondker Shajadul Hasan, and William Perrizo,” A Lossless Image Compression Technique Using Generic Peano Pattern Mask Tree”, IEEE, pp. 317-322, 2008.
[3] Ming Yang and Nikolaos Bourbakis,” An Overview of Lossless Digital Image Compression Techniques”, IEEE, pp. 1099-1102, 2005.
[4] Anilkumar Katharotiya, Swati Patel1 Mahesh Goyani, “ Comparative Analysis between DCT & DWT Techniques of Image Compression ”, Journal of Information Engineering and Applications Vol 1, No.2, 2011.
[5] V.K Padmaja and Dr. B. Chandrasekhar,” Literature Review of Image Compression Algorithm “, IJSER, Volume 3, pp. 1-6, 2012.
[6] P. aggarwal, B. Rani, “ Performance Comparison of Image Compression Using Wavelets ”, IJISC, IEEE, pp: 97-100,2010.
[7] Jau-Ji Shen and Hsiu-Chuan Huang, " An Adaptive Image Compression Method Based on Vector Quantization ", IEEE, pp. 377-381, 2010.
[8] Anson, L.F.," Fractal Image Compression ", BYTE, Oct. 1993, p. 195-202.
[9] ken cabeen and Peter Gent," Image Compression and the Descrete Cosine Transform " Math 45,College of the Redwoods.
[10] N. Saroya and P. Kaur, “ Analysis of image compression algorithm using DCT and DWT transforms ”, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 2, 2014.
[11] A. El-Zaart, " A Novel Method for Edge Detection Using 2 Dimensional Gamma Distribution ", Journal of Computer Science 6 (2), 2010, pp. 199-204.
[12] Melin P, Claudia IG, Juan RC, Mendoza O and Oscar (2013) Edge detection method for image processing based on generalized type-2 fuzzy logic. IEEE Transactions on Fuzzy Systems.
[13] A Novel Digital Algorithm for Sobel Edge Detection.
[14] A Descriptive Algorithm for Sobel Image Edge Detection, O. R.Vincent, O. Folorunso Proceedings of Informing Science & IT Education Conference (InSITE) 2009.
[15] Duda, R. O. Vehart, P. E., Pattern Classification and Scene Analysis, John Wiley and Sons, NY, 271-273, 1973.
[16] X. Yu, J. YlJaaski, " A new algorithm for image segmentation based on region growing and edge detection ", Proc. Int. Symp. Circuits and Systems, vol. 1, pp. 516-519, 1991.
[17] Green, P. J. and Sibson, R., " Computing Dirichlet Tessellations in the Plane ", Comput. J. 21(1978), 168–173.
[18] Miles, R. E., " The Random Division of Space ", Suppl. Adv. Appl. Prob. (1972), 243–266.
[19] Grunbaum, B. and Shephard, G. C., " Tilings with Congruent Tiles ”, Bull. (New Series) Amer. Math. Soc. 3 (1980), 951–973.