Arab Gulf Journal of Scientific Research, 27 (1&2): 26- 32 (2009)

Research Paper Ref. No 2429

Segmentation and Approximation Coding Based Algorithm for Compression of Solid Color Images

استخدام التجزئة وخوارزمية التشفير التقريبية لضغط الصور ذات الألوان المتراصة

Abdul Quaiyum Ansari

عبدالقيوم أنصاري

Department of Electrical Engineering, Jamia Millia IslamiaUniversity

New Delhi -110025, India, E-mail: aqansari@ieee.org

Abstract: This paper presents an approach to image compression for solid color pictures using segmentation and solid color image. An approch is presented that targets the solid color regions in the image as the source of redundancy and attempts to minimize this redundancy. The approch works by identifying regions in the image having pixels with similar color characteristics with respect to the specified tolerance level. To identify the solid color regions in the image, breadth-first traversal algorithm is used. The abbreviated form of these regions with necessary information is stored in the encoded image using recursive boundary splitting and the image is decoded using polynomial interpolation. In order to measure the quality of the image we have incorporated stretch factor. The results procured have been very satisfying and encouraging as it clearly has a sizeable lead over other conventionally used compression techniques.
Keywords: *Tolerance Level, Recursive Boundary Splitting, Piecewise Polynomial Interpolation, Solid Color Image/ Picture, Free Hanging End Points, Intersection Point.*

المستخلص: هذا البحث يقدم طريقة لضغط الصور ذات الألوان المتراصة (solid color) باستخدام التجزئة (segmentation). تعتمد الطريقة على أن المناطق ذات الألوان المتراصة تحتوي علي معلومات فائضة (redundancy)، ويتم في هذا البحث محاولة تقليص هذه المعلومات الفائضة وذلك من خلال تمييز المناطق ذات النقاط (pixels) المتشابهة بالخصائص اللونية إعتماداً على مدى اللون المحدد. لتمييز مناطق الألوان ذات الألوان المتراصة تم استعمال خوارزمية breadth-first traversal. مختصرات هذه المناطق مع المعلومات الضرورية يتم تخزينها في الصور المشفرة باستخدام فصل حدود المناطق المتكرر وذلك بتطبيق استيفاء متعدد الحدود (polynomial). لقياس جودة الصور المعالجة تم تطبيق عامل إمتداد (stretch factor). النتائج التي تم الحصول عليها مقنعة ومشجعة من حيث معدل ضغط الصور مقارنة مع الطرق المتعارف عليها.
كلمات مدخلية: *مستوى الاحتمال، إنشقاق الحد التكراري، استيفاء متعدد الحدود، الصور ذات الألوان المتراصة، النقاط الأخيرة المعلقة الحرة، نقطة تقاطع.*

INTRODUCTION

Image compression has impact on the storage as well on the transmission time. Image compression is the application of data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. The image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rate, introduce compression artifacts. Lossless

compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually losses.

There are various image compression techniques run-lent encoding, PCX, BMP, TGA, TIFF, DPCM, Predictive coding, entropy coding, LZW, etc. However, we have taken a different approach based on identification of solid color region in the image. Any image compression algorithm must ensure that the compressed image is exactly similar to the original image.

In this work, we have exploited the segmentation and approximation in order to achieve best results. The segmentation and approximation coding method is the process of representing the image as a mosaic of regions, where every pixel in a particular region has sufficient degree of uniformity with respect to a certain feature (e.g., grey level, texture). Each region is then assigned certain parameters related to the characterizing feature associated with it (Duda and Hart, 1973; Pavlidis, 1980).

In this paper we propose an algorithm for segmentation and approximation approach. The segmentation part of the algorithm uses recursive boundary splitting method to extract the control points of the curves bounding each region. For finding the control points of the curves during the segmentation phase of the algorithm, recursive boundary splitting (Duda and Hart, 1973; Pavlidis, 1980) approach has been used. In this approach segments or curves are recursively divided into smaller segments with respect to the specified limiting condition (Sonka, *et al*. 2001), as shown in Figure (1).

The polynomial approximation regions are reproduced by means of polynomial functions in (x, y); the task of the encoder is to find the optimum coefficients. The input to approximation method is the control points of the curves or segments obtained during segmentation. We have used B-spline's (De Boor, 1994; Paglieroni and Jain, 1988) piecewise polynomial interpolation technique (Sonka, *et al.* 2001) to obtain the approximated curves as shown in Figure (2), where (a), (b), and (c) represent convex $(n+1)$ polygon for B-spline of the n^{th} order, and (d) represents a 3 rd order spline.

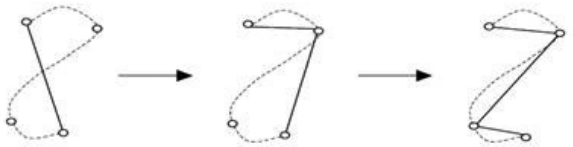


Fig. 1. Recursive boundary splitting (Sonka, *et al*. 2001, 243).

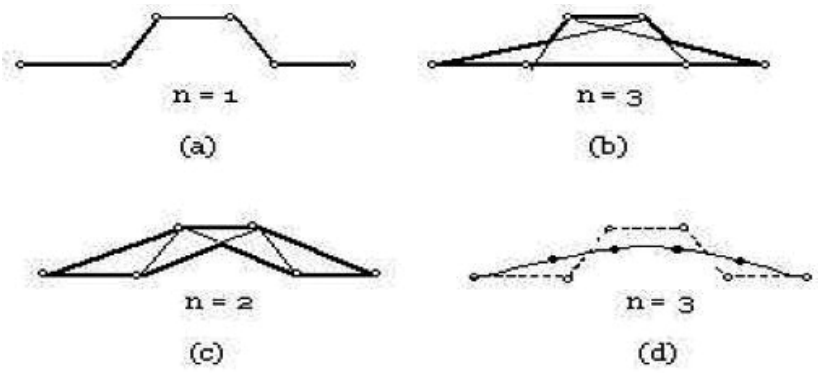


Fig. 2. Splines of order n (Sonka, *et al*. 2001, 243).

METHODOLOGY

We have proposed an algorithm which can perform lossy and losses compressions as discussed below.

Loss-less Version

For a compression technique to be called loss-less, it must ensure that the compressed image is exactly similar to the original image. Although if we are applying the tolerance level criteria for comparing color intensities of pixels while deciding the regions, it would not be possible to get a decoded image that is purely loss-less. However, the image will still be able to retain the exact shape of the edges as in the original image. The advantage of using the tolerance level criteria of comparing grey level of pixels is the lesser number of regions leading to a higher compression ratio.

Encoding

Step I

If the stretch factor value is found acceptable, the starting pixel is stored as a seed pixel with its coordinates and color. Then we traverse the image using breadth-first algorithm (Silvela and Portillo, 2001; Commen, *et al*. 1995) looking for sharp contrasts (with respect to the specified tolerance level for achieving an image that is partially loss-less) in only horizontally or vertically adjacent pixels. Whenever a sharp contrast is encountered, the current pixel is marked as a boundary point, and the first sharp contrasting pixel encountered is stored as a seed point for the next region. We continue traversing the image and marking boundary points and seed pixels until the entire image is covered. Thus, the entire image is transformed into a set of boundary points and seed pixels (Figure 3b).

Step II

This step involves finding the curves from the boundary points, obtained in step 1. These curves are not intersected by any other curve. In other words, all the curves that lie between two directly connected intersection points (boundary points having more than two boundary points as its direct neighbours) are to be identified. These intersection points represent the end points of the curve, whereas in some cases the end points may not be intersection point. In such cases, the free endpoints of the curve are considered as end points. This step first identifies each of these curves with its end points and its color, and then groups these curves on the basis of their color, thus reducing the amount of information by storing just one color for a group of curves that have the same color. The final encoded image for the loss-less version contains every point of all the boundary curves, the color of the curve (in groups), and the seed points for every region in the image (Figure 3c).

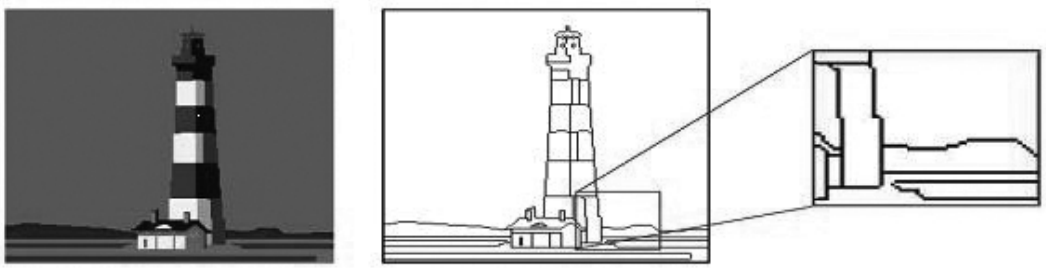
Decoding

Step I

We start reproducing/decoding the image by drawing all the boundary curves using the points and color of the curve stored in the encoded image (Figure 4a).

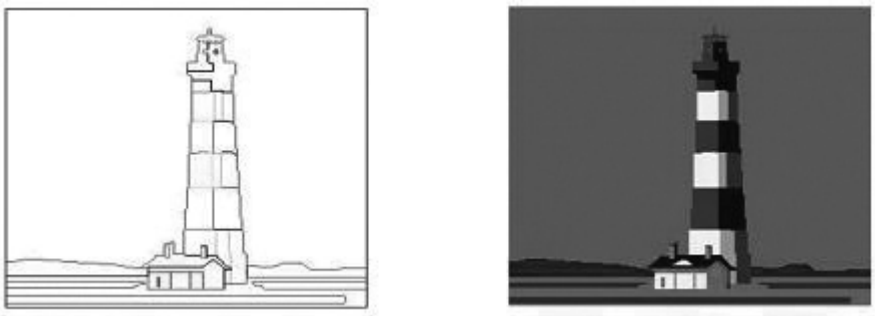
Step II

After drawing the boundaries we have to fill colors in the image using the information from the seed points, we stored in the encoding phase. We use the location and color of the seed pixel and start coloring each pixel of that region with the color of the seed pixel, while traversing in breadth-first manner. In a similar fashion, every region in the image is filled to produce an image which is visibly similar to the original image (Figure 4b).



a) Original b) step-I c) Step-II

Fig. 3. Loss-less Encoding.



a) Step I b) Step II (Decoded Image)

Fig. 4. Loss less Decoding.

Calculation of the Compression Ratio

Compression is the process of transforming raw image data (in binary form) to compressed (or encoded) format, the resulting image being somewhat smaller than the original. The ratio between the two sizes is called the compression ratio; e.g., if the original image size is 100 kilobytes, and the compressed size is 25 kilobytes, then a compression ratio of 4:1 has been achieved.

To calculate the compression ratio, the following methodology is used: If the image size is N for an image of width X and height Y and *b* = number of bits required to store color intensity of one pixel in the image; *n* = total number of regions identified in the image; B_i = number of boundary points of flat color region *i*; *C* = number of curves identified in the image; and *L* = number of bits required to store the location of a pixel. Then N is given by:

$$N = X * Y = \sum_{i=0}^{n} (\text{total number of points in region i})$$

We have introduced a term, Stretch Factor (*r)*, which is a measure of the elongated-ness of the flat color regions in the image. i.e.,

$$\text{Stretch Factor} = \frac{\sum_{i=0}^{n} B_i}{N} \quad 0 < \text{Stetch Factor} < 1$$

The lesser the Stretch Factor value, the higher will be the compression ratio. The compression ratio is given by:

$$\text{Compressio n ratio} = \frac{L \times \sum_{i=0}^{n} B_i + n(L+b) + C \times b}{N \times (b/8)}$$

This algorithm is faster than the lossy compression algorithm but gives a lesser compression ratio.

Lossy Version

The lossy version compromises the quality of the resulting image and takes a longer processing time to give a better compression ratio.

Encoding

Step I

Step I of encoding in lossy version of the algorithm is the same as step I of the loss-less version. i.e., the entire image is traversed in breadth first fashion, and the boundary pixels of flat/solid color regions are identified with seed point for every region in the image (Figure 5b).

Step II

This step also involves finding the curves from the boundary points (as in step II of the algorithm for loss-less version), obtained in step I. These curves are not intersected by any other curve. In other words, all the curves that lie between two directly connected intersection points are to be identified. This step first identifies each of these curves with its end points and its color, and then groups these curves on the basis of their color, thereby reducing the amount of information by storing just one color for a group of curves that have the same color (Figure 5c).

Step III

The final step of the encoding process of these curves (obtained in step II), is to find the control points of these curves. Control points are points on the curve that can represent the entire curve without actually storing every point on the curve. A control point finding algorithm is discussed in the Introduction section. Another parameter to the algorithm is the accuracy factor. The more the accuracy required, the more would be the number of control points for a curve/ boundary. These control points can be used to reproduce the original curve with the specified end points. The final encoded form of the image contains control points of all the boundary curves in the image, the color of each boundary curve, and the seed points for every region (Figure 5d).

Decoding

Decoding part of the algorithm mainly involves the following two steps.

Step I

We start reproducing the image by drawing the curves using the control points and color of the curve stored in the encoded image. These control points will actually be used to draw the boundaries of the flat color regions using B-splines as discussed before (Figure 6a).

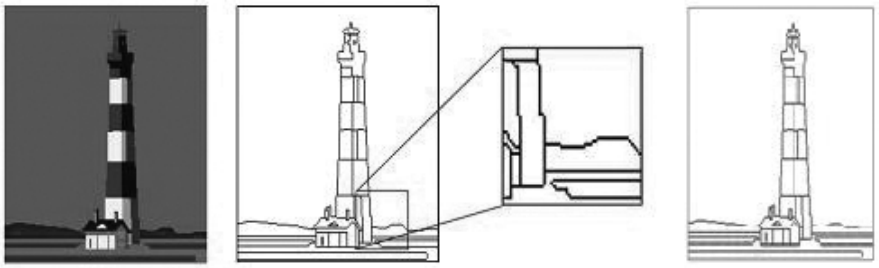
Step II

This step is the same as step II of decoding in previous version of the algorithm. Every region is colored using the information from the seed points stored in the encoded image (Figure 6b).

Calculation of the Compression Ratio

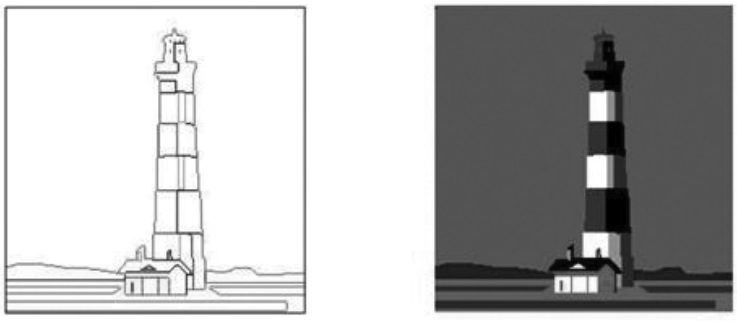
To calculate the compression ratio in the lossy version, the following Methodology is used: If the image size is N for an image of width X and height Y and b = number of bits required to store color intensity of one pixel in the image; n = total number of regions identified in the image; P_i = number of control points of flat color region i; C = number of curves identified in the image; and L = number of bits required to store the location of a pixel, then

$$N = X * Y = \sum_{i=0}^{n} (\text{total number of points in region i})$$



a) Original Image b) Step I c) Step II d) Step III

Fig. 5. Lossy Encoding.



a) Step I b) step II (decoded image)

Fig. 6. Lossy Decoding.

The stretch Factor here takes into account the control points of the curves instead of all boundary points as in the case of the loss-less version, as follows:

$$\text{Stretch Factor} = \frac{\sum_{i=0}^{n} P_i}{N} \quad 0< \text{Stetch Factor}<1$$

The compression ratio is given by

$$\text{Compressio n ratio} = \frac{L \times \sum_{i=0}^{n} P_i + n(L+b) + C \times b}{N \times (b/8)}$$

The compression ratio in this version is more because here only the control points of the curves are stored in the encoded image, whereas in the first version all boundary points are stored as it is in the encoded image. This is obvious since the number of boundary points will be much larger than the number of control points.

RESULTS

As depicted in Figure (7), the compression ratio is inversely proportional to the Stretch Factor. This means that the compression ratio will be more for the images in which flat areas are more compact, hence requiring lesser number of boundary or control points.

The above results, especially for lossy version, were very satisfying and encouraging, in addition to that it clearly has a sizeable lead over other conventionally used compression techniques (Table 1). This still can be improved through some finer refinements in the method. The results for loss-less method are also competing with the established methods to a considerable extent.

CONCLUSION

The proposed algorithm is suitable for graphics that have large areas of solid colors and few areas of fine detail if any. This algorithm is very well suited to images such as logos, banners advertisements, and text rendered on solid backgrounds. To identify the solid color regions in the image, breadth-first traversal algorithm is used.

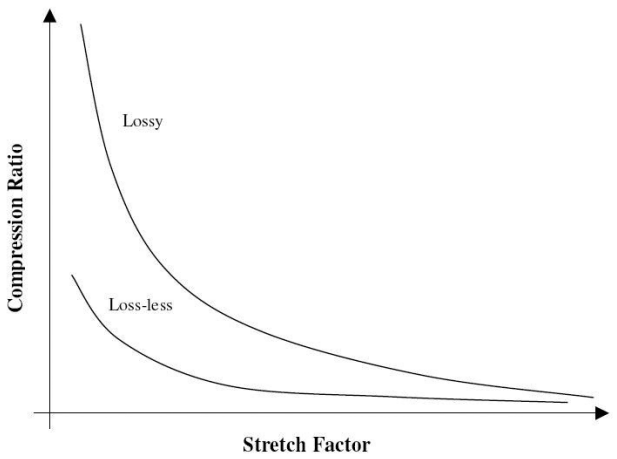


Fig. 7. Relation between Stretch Factor and Compression Ratio.

Table 1. Comparison of Compression Ratios.

	GIF	JPEG	Proposed Loss-less	Proposed Lossy
Average Compression Ratio	50-60:1	20-30:1	20-25:1	85-95:1

In order to measure the quality of the image we have incorporated Stretch Factor. The results procured have been very satisfying and encouraging as it clearly has a sizeable lead over other conventionally used compression techniques. However, the proposed algorithm is not efficient for graphics that have lots of texture and variation with few areas of solid colors.

REFERENCES

Cormen, T, Leiserson, C and **Rivest, R** (1995) *Introduction to Algorithms*. MA MIT Press, Cambridge.

De Boor, C (1994) *A Practical Guide to Spline*. Springer Verlag Press, NewYork.

Duda, **RO** and **Hart**, **DE** (1973) *Pattern Classification and Scene Analysis*. John Wiley Press, NewYork.

Paglieroni, DW and **Jain**, **AK** (1988) Control point transforms for shape representation and measurement. *Journal of Computer Vision, Graphics and Image Processing* **42** (1): 87-111.

Pavlidis,T (1980) *Structural Pattern Recognition*. Springer Verlag, New York.

Silvela, J, Portillo, J (2001) Breadth-first search and its application to image processing problems. *IEEE Transactions on Image Processing* **10** (8): 1194 – 1199.

Sonka, M, Hlavac, V and **Boyle, R** (2001) Image Processing, Analysis and Machine Vision. *In Chapman & Hall Computing*. 2nd ed.

Ref. No. (2429)
Rec. 22/04/2007
In-revised form: 27/04/2009