UDK 621.397

FRACTAL CODING OF VIDEO SEQUENCES BY GENETIC ALGORITHM

Osharovska O.V., Patlaenko M.O.

O.S. Popov Odessa National Academy of Telecommunications, 1 Kuznechna St., Odessa, 65029, Ukraine.

osharovskaya@gmail.com, nick_msa@ukr.net

Abstract. The results of application of the modified genetic algorithm of fractal coding to still images and video sequences are presented. The dependence of the compression coefficient on the size of the rank block is obtained. A comparison of fractal compression with standard MPEG-4 compression algorithms is performed and it is shown that it is possible to achieve double the compression coefficients at the same signal-to-noise ratio values. In the video sequence, both I-frames and predicted P, B frames were subjected to fractal compression. The simulation results showed what time in seconds is spent when encoding one frame. When the size of the rank block from 20 to 4 image elements was reduced, the encoding time increased by more than ten times for a still image, and for a video sequence less than twice, which indicates the perspectives of fractal compression of television images.

Key words: image, fractal coding, similarity, rank and domain blocks, compression ratio, signal to noise ratio.

Abstract. Представлены результаты применения модифицированного генетического алгоритма фрактального кодирования для неподвижных изображений и видеопоследовательностей. Получена зависимость коэффициента сжатия от размера ранга. Проведено сравнение фрактального сжатия со стандартными алгоритмами сжатия MPEG-4, и показано, что можно добиться удвоения коэффициентов сжатия при одинаковых значениях отношения сигнал/шум. В видеопоследова-

Osharovska O.V., Patlaenko M.O.

131
однако как I-кадры, так и предсказанные кадры P и B подвергались фрактальному сжатию. Результаты моделирования показали, какое время в секундах затрачивается при кодировании одного кадра. При уменьшении размера рангового блока с 20 до 4 элементов изображения время кодирования увеличилось более чем в десять раз для неподвижного изображения, а для видеопоследовательности менее чем в два раза, что говорит о перспективности фрактального сжатия телевизионных изображений.

**Ключевые слова:** изображение, фрактальное кодирование, самоподобие, ранговые и доменные блоки, коэффициент сжатия, отношение сигнал/шум.

В фрактальном сжатии, как и в других алгоритмах сжатия, механизмы, зависящие от степени сжатия и степени потери, являются очень важными. По настояющему, достаточно большая серия таких методов была разработана [1, 2, 3]. Целью настоящей статьи является сравнительный анализ качества восстановленных изображений после фрактального сжатия в параметрическом генеалогическом алгоритме как статичных изображений, так и видеостримов.

Первым, можно ограничиться числом аффинных преобразований, естественно, обеспечивая сжатие не меньше заданного значения. Вторым, можно требовать, чтобы в случае, когда разница между обработанным фрагментом и его лучшим приближением превышала некоторый порог, этот фрагмент обязательно сжимался. Третьим, можно исключать фрагменты маленького размера, например, четыре точки. Изменяя пороговые значения и приоритеты этих условий, мы получим гибкость в контроле отношения сжатия в диапазоне от точного соответствия до любом сжатия. Обратите внимание, что гибкость этого метода будет гораздо выше, чем у ближайшего "конкурента" - алгоритма MPEG.

**The Fractal Encoding of Still Images.** Фрактальный алгоритм преобразования изображений работает с прямоугольными областями размера, одинакового размера области – размер областей фиксировался с самого начала алгоритма до конца. Преимущество этого алгоритма состоит в том, что с правильным выбором размеров обрабатываемых областей, качество всего изображения задано к концу кодирования. В случае недостаточной степени сжатия в каждом фрагменте изображения изображение разбивается на четыре части, обрабатываемые в том же способе, как и все остальные.

Пусть яркостный компонент изображения разбит на N групп прямоугольных областей Rᵢ, для каждой из которых мы находим соответствующий образ Dᵢ и трансформацию Wᵢ, определяемую коэффициентами (c₁, c₂, ..., cₖ) так, что для каждого r из Rᵢ существует d из Dᵢ, такое, что r = Wᵢ(d). Дальнейшее, трансформации Wᵢ должны быть компрессивными, т.е. такими, что для всех dᵢ, dᵢ из Dᵢ это утверждение верно [3].

Демеративы базового алгоритма фрактального сжатия состоит в том, что для каждого фрагмента изображения R алгоритм проходит через все области D и все варианты их ориентаций, выполняя пер-пиксельные операции поиска новых трансформаций.

В этой статье, мы рассмотрели параметрический алгоритм фрактального сжатия, в котором статистические параметры были предварительно рассчитаны для блоков размерности. Фактически, следующие статистические параметры были предложены:

1. **Стандартное отклонение в ранговом блоке** (1)
   \[ \delta = \sqrt{\frac{\sum_{i}(P_{x,y} - \mu)^2}{N_i}}. \]  
   \[ (1) \]

2. **Асимметрия в ранговом блоке** (2)
   \[ a = \frac{\sum_{i}(P_{x,y} - \mu)^3}{N_i \cdot \delta^3}. \]  
   \[ (2) \]

3. **Перепалка контраста между пикселями в ранговом блоке** (3)

**Osharoska O.V., Patlaenko M.O.**

Fractal coding of video sequences by genetic algorithm
The standard deviation $\beta$ of the brightness of the pixel within rank block (4)

$$
\beta = \frac{\sum_{x=1}^{bw} \sum_{y=1}^{bh} \left( \frac{I_w}{2} \right)^2 + \left( \frac{I_v}{2} \right)^2}{\sum_{x=1}^{bw} \sum_{y=1}^{bh} \left( \frac{I_w}{2} \right)^2 + \left( \frac{I_v}{2} \right)^2},
$$

where the parameter $w$ is calculated by formula (5)

$$
\omega = \frac{\sum_{x=1}^{bw} \sum_{y=1}^{bh} \left( \frac{I_w}{2} \right)^2 + \left( \frac{I_v}{2} \right)^2}{N_I}.
$$

The maximum of the horizontal (6) or vertical segment gradient (7)

$$
h = \frac{\sum_{x=1}^{bw} \sum_{y=1}^{bh} \left( x - \frac{I_w}{2} \right) \cdot (P_{X,Y} - \mu)}{\sum_{x=1}^{bw} \sum_{y=1}^{bh} \left( x - \frac{I_w}{2} \right)^2},
$$

$$
v = \frac{\sum_{x=1}^{bw} \sum_{y=1}^{bh} \left( y - \frac{I_v}{2} \right) \cdot (P_{X,Y} - \mu)}{\sum_{x=1}^{bw} \sum_{y=1}^{bh} \left( y - \frac{I_v}{2} \right)^2},
$$

The following conventions are used in the formulas: $I$ – segment of the image;

$N_I$ – number of pixels in the segment $I$;

$P_{X,Y}$ – pixel brightness value at the point $(x, y)$;

$\mu$ – average pixel value in the segment $I$;

$I_x, I_y$ – horizontal and vertical coordinates of a rank block in a domain block.

Initially, the comparison of domains and ranks is performed on these characteristics, which significantly reduces (in dozens of times) the subsequent amount of computation, in comparison with the basic algorithm of fractal compression. Next, the values of the vector of characteristics for each domain block are calculated and stored, and when processing the ranking block, its characteristic vector is first computed, then the distance between the vector of characteristics of a given rank and the vector of characteristics of each domain block is calculated. For the subsequent comparison, we select only the given $q$-percent of domains with the minimum distance $d$ to this rank. Further, the search is performed with the only difference being that $q\%$ of domains having the characteristics are selected.

From transformations that transfer domains to rank areas, a mapping is created that takes the image to an image. In this case, the image code will be the location and dimensions of rank areas, as well as the coefficients of transformations describing the self-similarity inside the image. The number of bits required to describe the code will be substantially less than the number of bits required to describe the original image. The compression ratio is the ratio of the bit representation of the image to the bit representation of the code.

Five types of still images were subjected to fractal coding. The following test images were selected: a group of trees, a lone tree, a portrait of a girl, a mother with a child, a vase on the table.
The number of rank blocks, corresponding to their size, varied from twenty to two image elements. We used a parametric algorithm with an allowable incidence of the average brightness of rank-block descendant from the average brightness of the rank block-genome of no more than 5%. The procedure for selecting domains with closest distances from the vector of characteristics to rank provides selectivity, thereby limiting the number of domain blocks for enumeration, thereby reducing the number of operations and calculating the conversion coefficients. Parameters characterizing the time of encoding and decoding, as well as the average pixel error for a given permissible error for one type of image “a lone tree” (Fig.1) are given in Table 1.

Table 1 – Parameters for still image

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>Rank 20</th>
<th>Rank 12</th>
<th>Rank 8</th>
<th>Rank 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain blocks</td>
<td>225</td>
<td>841</td>
<td>2977</td>
<td>4562</td>
</tr>
<tr>
<td>Rank blocks</td>
<td>1651</td>
<td>1504</td>
<td>1264</td>
<td>1228</td>
</tr>
<tr>
<td>Average pixel error, %</td>
<td>3,79</td>
<td>3,92</td>
<td>3,73</td>
<td>3,69</td>
</tr>
<tr>
<td>Compression ratio</td>
<td>4,35</td>
<td>4,68</td>
<td>5,38</td>
<td>5,51</td>
</tr>
<tr>
<td>Coding time, s</td>
<td>4,58</td>
<td>12,86</td>
<td>30,25</td>
<td>49,39</td>
</tr>
<tr>
<td>Decoding time, s</td>
<td>1,91</td>
<td>1,62</td>
<td>1,41</td>
<td>1,25</td>
</tr>
</tbody>
</table>

The image of a single tree, divided into rank blocks, 12 by 12 in size, is shown in Figure 1. The blockiness of the image is clearly visible.

The number of rank blocks varied from 1200 to 1700. The number of domain blocks varied from 225 to 4500. The time spent on coding ranged from 4 seconds to 50 seconds and grew with the number of domain blocks. The per-pixel error ranged from 3,7 to 3,9%, which indicates a practical independence from the number of blocks, and, primarily, from the chosen algorithm is fractal.

Figure 1 – The size of the rank block 12

Fig. 2 shows the dependence of the compression ratio on the size of the rank blocks. Fig. 3 shows the dependence of the signal-to-noise ratio on the size of the ranks. Fig. 4 shows the dependence of the average error of coding time.
The Fractal Encoding of Movie Images. The video sequence can be compressed by two variants of fractal algorithms. More common is an algorithm in which each frame is treated as a separate two-dimensional image and encoded independently. Prediction on previous frames can speed up the encoding process. Moreover, you can use the already available set of rank and domain blocks. An interesting solution is to create a library of three-dimensional cubes - rank and domain blocks. Of course, storing all possible rank and domain blocks is perhaps desirable, makes it difficult to find the correct block. Already there is a solution that is close to the optimal solution for combining frames that are close in content to groups. This solution is used in MPEG-2, MPEG-4 compression standards. Frame groups contain different types of frames, known as I-frames, P-frames, B-frames. In standard MPEG encoder the main compression is provided by a discrete cosine transform and the selection of thresholds for the spectral coefficients.

However, not everything is as smooth as it may seem. If the image is uniform then the magnification results in an excellent outcome. However, if you compress the still life image, then it is very difficult to predict which new fractal structures will arise. You can increase almost any image two or three times, archiving each time with a small degree of loss.

Scaling is a unique feature inherent in fractal compression. Over time, it will probably be actively used in both special scaling algorithms and in many applications. Indeed, this requires the
concept of "application in the window". It would be nice if the image shown in the window 100x100 had the same clarity and detail when zoomed to the full screen – 1024 × 768.

To study the efficiency of the modified genetic algorithm of fractal coding, a sequence of five difference frames was chosen. Table 2 shows the performance of the algorithm. Fig. 4 shows the sequence of five different frames from the video sequence.

Table 2 – Indicators of genetic compression algorithm

<table>
<thead>
<tr>
<th>Indicators</th>
<th>Frame number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Domain blocks</td>
<td>571</td>
</tr>
<tr>
<td>Rank blocks</td>
<td>1838</td>
</tr>
<tr>
<td>Acceptable error, %</td>
<td>0,05</td>
</tr>
<tr>
<td>Average pixel error, %</td>
<td>2,61</td>
</tr>
<tr>
<td>Compression ratio</td>
<td>100,35</td>
</tr>
<tr>
<td>Coding time, s</td>
<td>1,73</td>
</tr>
<tr>
<td>Signal to noise ratio, dB</td>
<td>35,2</td>
</tr>
</tbody>
</table>

Attention is drawn to the increase in the compression ratio for the video sequence compared to the compression of a single image. The number of rank and domain blocks in this example is not an indicator for comparison because of different semantic content.

The size of the rank block is one of the main decisive factors in determining the compression ratio. Obviously, the larger the size of the rank block, the higher the compression ratio, but the processing time increases and the mid-pixel error grows (see Fig. 6).
With the use of the modified algorithm for four different semantics, the coding time varied from 4.8 to 49.39 p. For the same video sequences, the basic algorithm showed the coding time from 154.61 to 1068 p.

A comparison is also made with standard MPEG encoders (Fig.7).

First, note that both algorithms operate with 8-bit (in grayscale) and 24-bit full-color images. Both are lossy compression algorithms and provide similar archiving factors. Both the fractal algorithm and MPEG have the ability to increase the compression ratio by increasing the loss. In addition, both algorithms are very well parallelized. Differences begin when we consider the time required for algorithms to archive/unzip. So, the fractal algorithm does compression hundreds and even thousands times longer than MPEG. Unpacking the image, on the contrary, will happen five to ten times faster. Therefore, if the image is compressed only once, but transmitted over the network and decompressed many times, then it is more profitable to use a fractal algorithm. MPEG uses image decomposition by cosine functions, so the loss in it (even at specified minimum losses) is
manifested in waves and halos on the border of sharp color transitions. It is for this effect that they do not like to use it when compressing images that are prepared for high-quality printing: this effect can become very noticeable [4].

Fractal algorithm compression is spared this disadvantage. Moreover, when printing an image, each time, you must perform a zoom operation, since the raster (or a lineage) of the printing device does not coincide with the image raster. When converting, there can also be several unpleasant effects that can be combated or scaled by software (for low-cost printing devices such as conventional laser and inkjet printers) or by supplying a printing device with its processor, hard drive and a set of image processing programs (for expensive phototypesetting automata). As you might guess, when using a fractal algorithm such problems do not exist.

In this article, the comparative analysis of fractal coding of still and moving images is carried out. By time spent on encoding, the encoding of video sequences turns out to be faster. The final testing of the high-speed circuit showed that with photorealistic images, where the character of compression losses by the MPEG algorithm is less noticeable, the fractal compression is slightly behind. With raster images of geometric figures, fractal compression noticeably benefits MPEG, which is due to a higher degree of self-similarity of the images. The MPEG algorithm copes well with rectangles and lines arranged vertically or horizontally. On compression of text images, the algorithm works poorly (image text).

REFERENCES:


ЛИТЕРАТУРА: