КОМПРЕССИЯ ИЗОБРАЖЕНИЯ ФРАКТАЛЬНЫМ ПРЕОБРАЗОВАНИЕМ

Йорданова С. М., Тянев Д. С.
Технический университет – г. Варна, Болгария
sl@newmail.com; dstyanev@mbox.digsys.bg

Abstract

Yordanova Sl. M. , Tyanev D. S., Image compression with fractal transformation. *The use of fractal for image compression is needed for finding such fractals that can present the image with the needed punctuality. For this purpose is possible to be used deterministic approach or stochastic approach. When the deterministic approach is used, for each point from the image is used IFS transformation. The new points are drawn and the process is repeated as long as is needed till the final picture is ready.*

1. Введение

Обработка изображения включает два этапа – кодирование и декодирование. Основная идея компрессии состоит в том, что изображение может быть восстановлено с помощью сохраненного набора преобразований, занимающего меньший объем памяти, чем оригинал. Целью настоящей работы является выполнение кодирования в виде итеративной системы $w: F \to F$, где F есть функционально полное метрическое пространство, а w - оператор свертки. Для компрессии здесь применяется фрактальное преобразование [1, 2, 3]. Предложен алгоритм компрессии и показан результат компрессии неподвижного изображения.

2. Фракталы и итеративная функциональная система

Понятие фрактал введено Манделбротом и определяется как кривая, размерность которой (по Хаусдорфу-Безиковичу) больше размерности Евклидового пространства. Метод итеративных функциональных систем (ИФС), который предложил Барнсли, применяет m последовательных афинных преобразований - $w_1, w_2, \dots, w_m$, с заданными вероятностями [1, 2]. Аффинные преобразования - это комбинации ротации, траляции и масштабирования координатных осей n-мерного пространства. Преобразование точки (x,y) с поверхности изображения в точку (x_n, y_n) описывается [2] системой:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = w.\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ab \\ cd \end{bmatrix}.\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} ax+by+e \\ cx+dy+f \end{bmatrix} \qquad (1)$$

где параметры *a,b,c,d* определяют ротацию и масштабирование. Стабильность итеративной системы требует такие масштабы, которые обеспечивают сжатие расстояния между точками. Трансляция определяется параметрами *e* и *f*. Выполнение этого преобразования над геометрической фигурой приводит к ее кручению, сжатию и перемещению.

Использованные фракталы для компрессии изображения должны обеспечить требуемую точность. Для этой цели можно применить детерминистический или стохастический подходы. Имея в виду большие требований к объему оперативной памяти этот подход применяется редко.

При стохастическим подходе выбирается случайным образом одна из четырех аффинных операций. Отдельные части изображения рисуются вероятностным образом "летающим пятном". Общий вид кодированного ИФС изображения сформируется в начальном стадии процесса. Считается, что численная реализация этого алгоритма более проста [1, 2, 3, 6].

3. Алгоритм компрессии изображения фрактальными преобразованиями

Создания трехмерных изображений осуществляется аффинными преобразованиями вида (2) [2, 3]:

$$\begin{bmatrix} x_n \\ y_n \\ x_n \end{bmatrix} = w.\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} abc \\ def \\ ghm \end{bmatrix}.\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} n \\ q \\ r \end{bmatrix} \qquad (2)$$

Когда сводимые трансформации расширяются в трехмерные для создания фрактальных поверхностей, предлагается использовать следующий подход [6, 8]:

- Фрактальная поверхность рассматривается, как степень яркости двумерного изображения, которое может быть создано итеративно преобразованием (2), где *Z* есть ось яркости.
- Система (2), т.е. пара (*x,y*) определяет одно значение яркости *Z*.

Рассмотрим изображение типа "плоскость под наклоном" (фигура 1). Верхний правый угол изображения белого цвета (Z=0,5), нижний левой угол - черного цвета (Z=-0,5), а цвет промежуточных уровней изменяется плавно от нуля (Z=0).

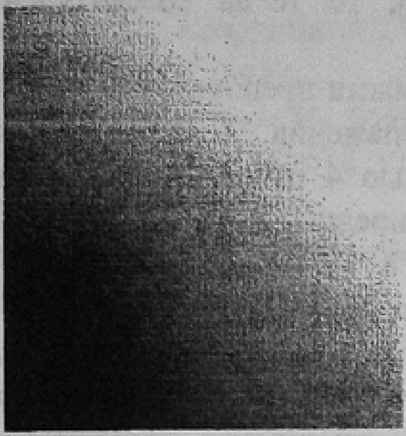

Рис.1 - Изображение типа *"плоскость под наклоном"*

В таблице 1 показаны значения ИФС параметров, необходимые для генерирования показанного изображения. Представлены изменения в значениях после первых трех итераций.

Таблица 1

ИФС-код изображения

	A	B	C	D	G	H	E	F	I
1	0,5	0,0	0,0	0,5	0,25	0,25	0,25	0,25	0,25
2	0,5	0,0	0,0	0,5	0,25	0,25	-0,25	0,25	0,00
3	0,5	0,0	0,0	0,5	0,25	0,25	0,25	-0,25	0,00

Этот пример иллюстрирует возможность создания яркостной картины при использованием трехмерных сводимых преобразований.

В случае представления двумерной картины преобразованием (1) получится следующее описание [1, 4, 5]:

$$w_1 \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

$$w_2 \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}; \tag{3}$$

$$w_3 \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}.$$

Каждое w_i задается 6 реальными числами (элементы матриц – 1/2 и 0). Таким образом, будут использованы 18 чисел плавающей запетой единичной точностью и для них будут необходимы 72 байта памяти. Если примем, что для изображения A∞ (после последней итерации) требуются

256x256x1=8192 [B], то тогда коэффициент компрессии будет равен приблизительно 113.

Такими сводимыми преобразованиями могут быть найдены подобия в разных частях изображения. Может быть определено соответствие между блоками размерностью 4 отчетами либо больше, например 8 отчетами. Процедура работы предполагает покрытие изображения блоками “4” - граничными. Блоки “8” называют доменными и могут быть сформированы из любой части изображения [7, 8].

Для каждого граничного блока нужно найти соответствующий его форме доменный блок. Увеличивая количество различных шаблонов, мы увеличиваем шанс отыскания хорошего соответствия. Для этого имеются следующие подходы:

- Каждый отчет можно начинать с доменного блока. Доменные блоки могут перекрывать друг друга.
- Данные об амплитуде в блоке можно масштабировать и смещать. Значение масштабного коэффициента ограничивается упомянутым выше критерием скручивания.
- Данные в доменном блоке могут быть переупорядоченными – например зеркальной трансформацией.

Доменный блок в одном изображении будет иметь кубическою форму – по оси Z изображается степень “серого” цвета. Основа его 8x8 пикселов, а высота - 0÷255. Трансформация преобразует этот куб в меньший граничный блок размерами 4x4 в другом масштабе, о котором в силе следующее ограничение – объем граничного блока должен быть меньшим, чем объем доменного блока. Таким образом, возможная ошибка обходится (возможно обойти), а на следующих итерациях уменьшается. В одномерном пространстве данные можно обрабатывать геометрически, например их зеркальном преобразованием. В двумерном пространстве могут быть совершены 8 простых геометрических действий типа ротаций и зеркальных преобразований [3, 4, 5].

После сказанного предлагается следующий алгоритм для фрактальной компресии неподвижного изображения:

1. Для каждого граничного блока определяется соответствующий доменный блок, чье местоположение неизвестно, но будет определено во время процесса кодирования.
2. Ищется доменный блок данного класса и преобразование, которое минимизирует расстояние от него до граничного блока.
3. При отсутствии хорошего соответствия, увеличивается класс допустимых преобразований доменных блоков с целью уменьшения ошибки.
4. После нахождения доменного блока необходимо выполнить кодирование коэффициентов преобразований. В данном примере

преобразования имеют 4 параметра: координаты x и y доменного блока, смещение и масштаб степени "серого" цвета.

5. Данные ставят в последовательность, кодированная, например методом Хафмана. Для изображения 512x512 пикселов существуют 128x128=16304 преобразующих значений.

6. Со стороной декодера из этих двоичных данных получаются коэффициенты преобразований.

7. Начиная с некого первоначального изображения (в общем случае с "пустого" екрана) после некоторого количества итераций получается реальное изображение.

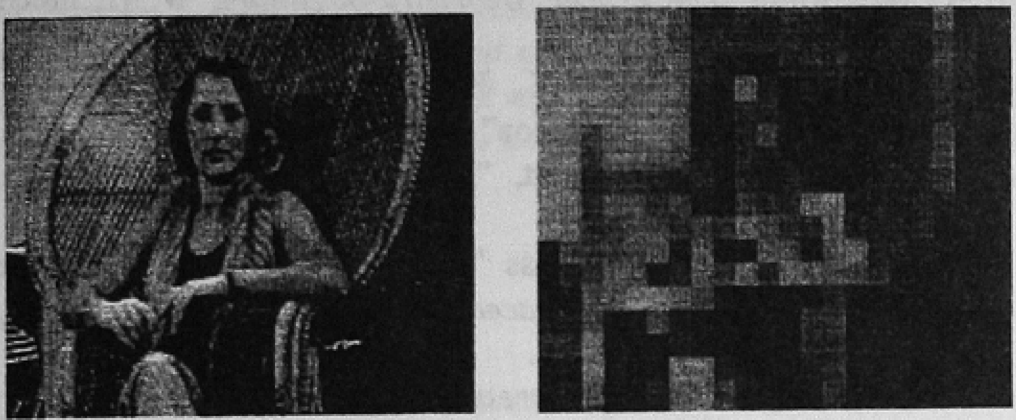

Рис. 2 - Результат компрессии стандартного изображения *"Lena"*

Представленный алгоритм может быть расширен и для кодирования динамических изображений, где граничные блоки будут размерами 4x4x4, а доменные – 8x8x8. В идеальном случае извлечение доменных блоков можно ожидать получить из произвольной части последовательности кадров, которое, однако, предполагает запоминание всех кадров в памяти. На практике последовательность кадров делят на "пачки", которые сохраняют в памяти и над которым применяют 3D преобразований. Впоследствии временную информацию используют для кодирования пространственой.

Выводы

- Метод фрактальных преобразований не имеет ограничения относительно коэффициента компрессии, однако очень большие его значения приводят к затуманенным изображениям.
- Разрешение дисплея не имеет значения, так как фрактальное изображение представляет бесконечную серию преобразований. Независимо от степени увеличения, детали изображения не меняются.
- Методы, использующие фракталы для представления и компрессии сигналов и изображений, перспективные. Они являются

основой для аппаратно-программных средств недорогих мультимедийных систем. Они найдут применение для реализации телевидения высокой разделительной точностью (HDTV).

Коротко было изложены результаты исследования практического применения фрактального преобразования для компрессии изображений и основные полученные выводы.

Литература

1. M. Barnsley, L. Hurd, *Fractal image compression*, AK Peters, 1993;
2. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freedman and Company, New York, 1977.
3. Йорданова С. М., Рачев Б., Наумов В., *Методи за компресия на информация*, изд.: "Г. Бакалов" – София, 1999 г.
4. W. J. Chen and W. K. Pratt, *"Scenes adaptiv coder"*, Trans. Comput. IEEE, vol 32, 1999.
5. R. F. Musmann and D. Preuss *"Compression variable-length coding for efficient compression of spacecraft television data"*, Trans. Comput. Technol. IEEE, vol 119, 1996.
6. J. Storer, *Image and text compression*, Kluwer Acad. Publ. 1999.
7. Farrelle P. M., *Recursive block coding for image data compression*, New York, 1999.
8. De Natale F. B., Desoli G. S., *"Adaptive least-squares bilinear intrerpolation"*, Electron. Lett., 2000, 29, pp 1638-1640.

Дата надходження до редколегії: 23.12.2003 р.