

Фрактальное преобразование в компрессии изображения

Йорданова С. М., Тянев Д. С.

1. УВОД

Обработка изображения включает два этапа – кодирование и декодирование. Основная идея компрессии состоит в том, что изображение может быть восстановлено с помощью сохраненным набором преобразований, занимающие меньшим объемом памяти, чем оригинал. Целью настоящей работы является выполнение кодирования в виде итеративной системой $w: F \rightarrow F$, 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, чья размерность (по Хаусдорфу-Безиковичу) больше размерности Евклидова пространства. Метод итеративных функциональных систем (ИФС), который предложил Барнсли, применяет m последовательных аффинных 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:

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

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 крутению, сжатию и перемещению.

Используемые фракталы для компрессии изображения должны обеспечить требуемую точность. Для этой цели можно применить детерминистический или стохастический подходы. Имея ввиду больших 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 пятношком”. Общий вид кодированого ИФС изображения сформируется в начальном стадии процесса. Считается, что численная реализация этого алгоритма более простой [1, 2, 3, 6].

3. АЛГОРИТМ КОМПРЕССИИ ИЗОБРАЖЕНИЯ ФРАКТАЛЬНЫМИ ПРЕОБРАЗОВАНИЯМИ

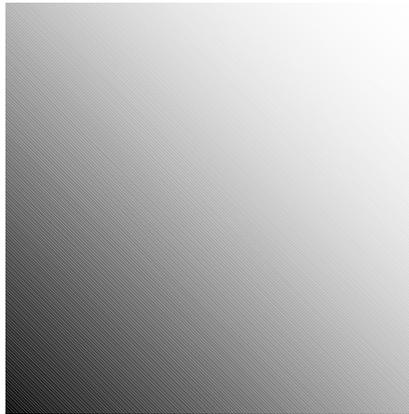
Создания тримерных изображений осуществляется аффинными преобразованиями вида (2) [2, 3]:

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

Когда сходимые трансформации расширяются в три-мерные 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 левый угол - черного цвета ($Z=-0,5$), а цвет междинных уровней изменяется плавно от нуля ($Z=0$).



Фиг.1 Изображение тип “плоскость под наклоном”

В таблице 1 показаны значения ИФС параметров, необходимые для генерирования показанного изображения. Представлены изменения значений 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 использовании тримерных сходимых преобразований.

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

$$(3) \quad \begin{aligned} w_1 \cdot \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \\ w_2 \cdot \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} \\ w_3 \cdot \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} \end{aligned}$$

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

последней итерации) требуются $256 \times 256 \times 1 = 8192$ [В], то тогда коэффициент компрессии будет равен приблизительно 113.

Такими сходимыми 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” - названы граничными. Блоки “8” называют доменными и могут быть сформированы из любой части изображения [7, 8].

Для каждого граничного блока нужно найти соответствующий его форму доменный блок. Увеличивая количество различных шаблонов, которых можно извлечь с изображения, увеличивается 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 переупорядочены – например зеркальной трансформацией.

Доменный блок в одном изображении будет иметь кубическую форму – по оси Z изображается степень “серого” цвета. Основа его 8×8 пикселей, а высота - $0 \div 255$. Трансформация преобразует этот куб в меньший граничный блок размерами 4×4 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 действиях типа ротаций и зеркальные преобразования [3, 4, 5].

После сказанного предлагается следующий алгоритм для фрактальной компрессии 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, закодированная например методом Хафмана. Для изображения 512×512 пикселей существуют $128 \times 128 = 16384$ преобразующих значений.
6. Со стороны декодера из этих двоичных данных получают коэффициенты преобразований.
7. Начиная с некоего первоначального изображения (в общем случае с “пустого” экрана) после некоторого количества итераций получается реальное изображение.

Каждое w_i задается 6 реальными числами (элементы матриц – $1/2$ и 0). Таким образом будут использованы 18 чисел плавающей запятой единичной точностью и для

них будут необходимы 72 байта памяти. Если примем, что для изображения А (после последней итерации) требуются $256 \times 256 \times 1 = 8192$ [В], то тогда коэффициент компрессии будет равен приблизительно 113.

Такими сходимыми 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” - названы граничными. Блоки “8” называют доменными и могут быть сформированы из любой части изображения [7, 8].

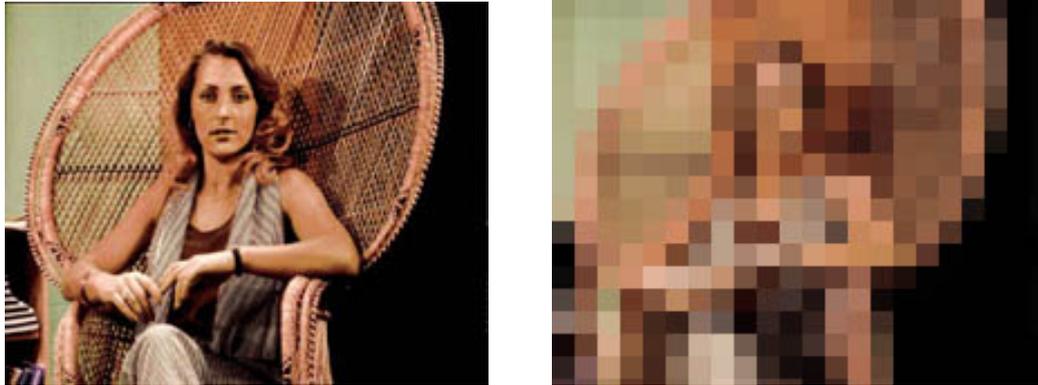
Для каждого граничного блока нужно найти соответствующий его форму доменный блок. Увеличивая количество различных шаблонов, которых можно извлечь с изображения, увеличивается шанс отыскивания хорошего соответствия. Для этого имеются следующие подходы:

- 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 переупорядочены – например зеркальной трансформацией.

Доменный блок в одном изображении будет иметь кубическую форму – по оси Z изображается степень “серого” цвета. Основа его 8×8 пикселей, а высота - $0 \div 255$. Трансформация преобразует этот куб в меньший граничный блок размерами 4×4 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 зеркальным преобразованием. В двумерном пространстве могут быть совершены 8 простых геометрических действиях типа ротаций и зеркальные преобразования [3, 4, 5].

После сказанного предлагается следующий алгоритм для фрактальной компрессии неподвижного изображения:

8. Для каждого граничного блока определяется соответствующий доменный блок, чье местоположение неизвестно, но будет определено во время процесса кодирования.
9. Ищется доменный блок данного класса и преобразование, которое минимизирует расстояние от него до граничного блока.
10. При отсутствии хорошего соответствия, увеличивается класс допустимых преобразований доменных блоков с целью уменьшения ошибки.
11. 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.
12. Данные ставят в последовательность, закодированная например методом Хаффмана. Для изображения 512×512 пикселей существуют $128 \times 128 = 16384$ преобразующих значений.
13. Со стороны декодера из этих двоичных данных получают коэффициенты преобразований.
14. Начиная с некоего первоначального изображения (в общем случае с “пустого” экрана) после некоторого количества итераций получается реальное изображение.



Фиг. 2 Результат компресии стандартного изображения “Lena”

Представленный алгоритм может быть расширен и для кодирования динамических изображений, где граничные блоки будут размерами $4 \times 4 \times 4$, а доменные – $8 \times 8 \times 8$. 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. В последствии временную информацию используется для кодирования пространственной.

4. ВЫВОДЫ:

- Метод фрактальных преобразований не имеет ограничения относительно коэффициента компресии, однако очень большие его значения приводят к затуманенным изображениям.
- Разделительность 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 компресии сигналов и изображений, перспективны. Они являются основой для аппаратно-программных средств недорогих мультимедийных систем. Они найдут применение для реализации телевидения высокой разделительной точностью (HDTV).

Коротко было выложено исследование о практическом применении фрактального преобразования для компресии изображений и основные выводы, получены глядя на него критично.

5. 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.