УДК 621.396.62

АНАЛИЗ ВЛИЯНИЯ ПОМЕХ НА СЖАТИЕ ИЗОБРАЖЕНИЯ ФРАКТАЛЬНЫМИ АЛГОРИТМАМИ

А.И. Стрелков[1], В.И. Барсов[2], Р.Ф. Пшеничный[2]
([1]Харьковский университет Воздушных Сил,
[2]Украинская инженерно-педагогическая академия, Харьков)

Рассмотрены математические модели фрактальных алгоритмов сжатия на предмет устойчивости к воздействию помех случайного характера, предложены пути разрешения данной проблемы. Рассмотрена возможность создания оптимизированного алгоритма, устойчивого к воздействию помех.

помехи, сжатие изображения, фрактальные алгоритмы

Постановка проблемы. В [1] были приведены результаты эксперимента, согласно которым применение фрактального алгоритма сжатия целесообразно только при высоком отношении сигнал/шум. В случае, когда уровень помехи высокий, потери возрастают, а степень архивации понижается, что делает применение фрактального алгоритма неэффективным. Следовательно, при высоком уровне помех, когда аппаратными средствами понизить его невозможно, становится актуальной разработка методов повышения помехоустойчивости алгоритма сжатия.

Для дальнейшей разработки помехозащищенных алгоритмов необходимо рассмотреть, чем же помехи так влияют на алгоритм. Фрактальные алгоритмы далеко не самые распостраненные, но с точки зрения рациональности было бы не уместным разрабатывать новый алгоритм, основываясь, например, на алгоритме JPEG. В настоящее время этот алгоритм устарел, и на его основе разработан новый стандарт сжатия JPEG2000 – так называемый «многослойный JPEG». К тому же, фрактальные алгоритмы сжатия позволяют сжимать изображение с гораздо большим коэффициентом сжатия, без видимого ухудшения качества изображения. Таким образом, рационально было бы рассмотреть именно фрактальные алгоритмы на предмет влияния на них помех ввиду того, что по предложенным критериям оценки фрактальный алгоритм дает наилучшие результаты.

Анализ литературы. Различные методы сжатия изображений основываются на устранении тех или иных форм избыточности, в частности фрактальные методы рассматривают самоподобие как источник избыточности [2, 3]. Считается, что самоподобие является свойством почти всех природных объектов и их изображений [4], и, следовательно, устранение этой формы избыточности может значительно уменьшить объем данных, необходимых для описания природного объекта или его изображения. Примером может служить «папоротник Барнсли» (рис. 1).

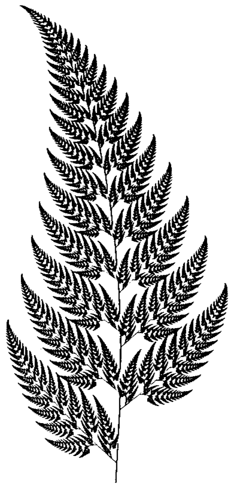

Рис. 1. Папоротник Барнсли

Фрактальное кодирование (сжатие) полутоновых изображений основано на гипотезе, согласно которой в любом изображении можно обнаружить локальное самоподобие различных его частей. Существующие алгоритмы фрактального сжатия, как правило, придерживаются следующей схемы кодирования [5]. Кодируемое изображение разбивается на множество не перекрывающихся блоков (ранговых областей), для каждого из которых, в пределах этого же изображения, ищется блок большего размера (домен), пиксели которого путем некоторого преобразования, задаваемого несколькими коэффициентами, переводились бы в пиксели ранговой области. При этом для поиска оптимального соответствия ранговых областей и доменов необходим полный перебор вариантов, что влечет за собой значительные вычислительные затраты. Из преобразований, переводящих домены в ранговые области, формируется отображение, переводящее изображение в изображение. При этом кодом изображения будут являться местоположение и размеры ранговых областей, а также коэффициенты преобразований, описывающих самоподобие внутри изображения. Количество бит, необходимых для описания кода, будет существенно меньше количества бит, необходимых для описания исходного изображения. В известных фрактальных методах сжатия изображений значение коэффициента сжатия может достигать 100 при приемлемом качестве восстановления.

Цель работы: выявление последствий влияния зашумленности изображения на степень сжатия изображений фрактальными алгоритмами.

Основная часть. Рассмотрим математическое описание метода фрактального сжатия. Пусть $x = (x_1, x_2, \dots, x_n)^T \in R^n$ – исходное изображение (сигнал). Рассмотрим аффинное преобразование

$$W : x \to W(x) = Ax + b\,, \tag{1}$$

состоящее из матрицы A и вектора смещения b. Будем рассматривать только W, обладающие свойством сжимаемости:

$$\exists s \in [0,1) : d(W(x), W(y)) < sd(x,y) \forall x, y \in R^n\,, \tag{2}$$

тогда существует единственная неподвижная точка отображения W [4]:

$$\exists x_f = W(x_f)\,. \tag{3}$$

Процесс кодирования состоит в определении A и b таких, что расстояние $d(x_0, x_f)$ между исходным сигналом x_0 и неподвижной точкой x_f отображения W минимально, а сами A и b имеют сравнительно простое представление. По полученному ИФС-коду можно восстановить его неподвижную точку (а, следовательно, и исходный сигнал), решая итеративным образом уравнение $x_f = W(x_f)$. Начиная с произвольного начального приближения $x_0 \in R^n$, последовательность итераций $x_k = W^k(x_0)$ сходится к искомой неподвижной точке $x_f = \lim\limits_{k\to\infty} x_k$. Таким образом, если выполнены все указанные выше требования, то с помощью ИФС можно итерационным путем получить сколь угодно хорошую аппроксимацию исходного изображения. Но на практике решение обратной задачи построения ИФС-кода требует проведения слишком большого объема вычислений. Теорема о коллаже, доказанная М. Барнсли, позволяет значительно упростить процедуру построения фрактального кода. Согласно этой теореме, вместо минимизации расстояния $d(x_0, x_f)$ между исходным образом и неподвижной точкой ИФС, можно минимизировать расстояние $d(x_0, W(x_0))$ между исходным образом и его «коллажем», полученным с помощью данной ИФС.

Таким образом, построенный фрактальный код является субоптимальным, и является единственным применяемым на практике. Это значит, что имеют место потери в качестве изображения.

С учётом вышесказанного, схема компрессии выглядит так: изображение x разбивают на кусочки x_i, называемые ранговыми областями. Далее для каждой области x_i находят область d_i и преобразование w_i такие, что выполняются следующие условия: 1) d_i по размерам больше x_i; 2) $w_i(x_i)$ имеет ту же форму, размеры и положение, что и x_i; 3) коэффициент u преобразования w_i должен быть меньше единицы; 4) значение должно быть как можно меньше. Первые три условия означают, что отображение w_i будет сжимающим. А в силу четвёртого условия кодируемое изображение x и его образ w(x) будут похожи друг на друга. В идеале x = w(x). А это означает, что наше изображение x и будет являться неподвижной точкой w. Именно здесь используется подобие различных частей изображения (отсюда и название – «фрактальная компрессия»). Как оказалось, практически все реальные изображения содержат такие похожие друг на друга, с точностью до аффинного преобразования, части.

Таким образом, для компрессии изображения w нужно: 1) разбить изображение на ранговые области x_i (непересекающиеся области, покрывающие все изображение); 2) для каждой ранговой области x_i найти область d_i (называемую доменной), и отображение w_i, с указанными выше свойствами; 3) запомнить коэффициенты аффинных преобразований w, положения доменных областей d_i, а также разбиение изображения на домены.

Соответственно, для декомпрессии изображения нужно: 1) создать какое-то (любое) начальное изображение x_0; 2) многократно применить к нему отображение w (объединение w_i).

Так как отображение w сжимающее, то в результате, после достаточного количества итераций, изображение придёт к аттрактору и перестанет меняться. Аттрактор и является нашим исходным изображением. Декомпрессия завершена.

Пусть дано изображение $m \times n$ точек (m и n кратны 8), 256 градаций серого. Ранговые и доменные области будем брать квадратными. Исходное изображение разобьём на ранговые области размером 8×8 точек. Доменные области будем искать размером 16×16 точек путём перебора всех возможных положений. Существует всего 8 аффинных преобразований, переводящих квадрат в квадрат (повороты на 0, 90, 180 и 270°, зеркальные отражения относительно центральной горизонтали, центральной вертикали, от главной и побочной диагоналей). Осталось найти только коэффициенты для преобразования цвета. Но значения u и v (контрастности и яркости) можно легко найти аналитически.

Если есть две последовательности значений цвета пикселов a_1, a_2, … , a_n (доменной области) и b_1, b_2, … , b_n (ранговой области), то можно минимизировать среднеквадратичное отклонение цвета пикселов, представляющее собой вариант метрики различия изображений

$$x = \sum_{i=1}^{n} (u \cdot a_i + v - b_i)^2 \,. \tag{4}$$

Для этого достаточно приравнять частные производные x по u и по v к нулю, и решить уравнение относительно u и v. Тогда:

$$u = \left(n \cdot \sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} b_i \sum_{i=1}^{n} a_i \right) \Bigg/ n \cdot \sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i \right)^2 ; \; v = \left(\sum_{i=1}^{n} b_i - u \cdot \sum_{i=1}^{n} a_i \right) \Bigg/ n \,, \tag{5}$$

при этом, если $n \cdot \sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i \right)^2 = 0$, то $u = 0$; $v = \sum_{i=1}^{n} b_i \Big/ n$.

Итак, какие же данные необходимо хранить в результате. Сетка разбиения на ранговые области постоянная для всех изображений, её хранить не надо. Остаётся положение ранговых областей (верхнего левого угла), номер преобразования и коэффициенты яркости и контрастности. Как же оценить степень потерь и регулировать их в этом случае? Во-первых, можно ограничить степень аффинных преобразований, заведомо обеспечив степень сжатия не ниже фиксированной величины. Во-вторых, можно потребовать, чтобы в ситуации, когда разница между обрабатываемым фрагментом и наилучшим его приближением будет выше определенного порогового значения, этот фрагмент дробился обязательно. В-третьих, можно запретить дробить фрагменты размером, допустим меньше четырех пикселей. При этом, при повышении степени сжатия соответственно и качество результирующего изображения становится ниже. А при сжатии сильно зашумленного изображения падение эффективности сжатия значительно, ввиду того, что в сжимаемом изображении подобные области гораздо мельче пи сильном зашумлении.

В данном случае имеет смысл разработка методов оптимизации алгоритма, учитывающих его субоптимальность.

Выводы. Предложен вариант оптимизации алгоритма фрактального сжатия изображений, исходя из которого можно:

– оценить влияние случайной помехи на качество изображения после сжатия по алгоритму;

– выявить дальнейшие этапы по сведению к минимуму или устранению субоптимальности алгоритма;

– затрачивать для компрессии и декомпрессии приемлемые вычислительные мощности.

Перспективы дальнейших исследований. Работа над оптимизацией фрактального сжатия позволит свести к минимуму влияние помех на степень сжатия и качество результирующего изображения, что очень актуально для следящих систем, работающих в условиях высоких помех (атомные электростанции, системы видеонаблюдения в условиях плохой видимости).

ЛИТЕРАТУРА

1. *Стрелков А.И., Барсов В.И., Пшеничный Р.Ф. Исследование возможности применения фрактального алгоритма сжатия для обработки зашумленного изображения // Системи обробки інформації. – Х.: ХУ ПС, 2005. – Вып. 2 (42). – С. 127 – 231.*
2. *Franti P., Nevalainen O. A Two-Stage Modeling Method for Compressing Binary Images by Arithmetic Coding // The Computer Journal. – 1993. – 36. – P. 615 – 622.*
3. *Memon N., Wu X. Recent Developments in Context-Based Predictive Techniques for Lossless Image Compression // The Computer Journal. – 1997. – Vol. 40, No. 2/3.*
4. *Peitgen H.O., Jurgens H., Saupe D. Chaos and Fractals. – Berlin: Springer-Verlag, 1992.– 542 p.*
5. *Fisher Y. Fractal image compression // SIGGRAPH'92 Course Notes. – 1992. Vol.12. – P. 7.1 – 7.19.*

Поступила 11.10.2005

Рецензент: доктор технических наук, профессор Ю.В. Стасев,
Харьковский университет Воздушных Сил.