Увеличение степени компрессии фрактального сжатия путем указания качества участков изображения

Ватолин Д.С. (dmitriy@graphics.cs.msu.su)
Московский Государственный Университет

АННОТАЦИЯ

В данной статье предлагается изменить фрактальный алгоритм сжатия изображений таким образом, чтобы появилась возможность задавать степень качества отдельных участков изображения. Сохранение семантически наиболее важных участков изображения с лучшим качеством, позволяет добиться большей окончательной степени сжатия при том же субъективном (визуальном) качестве всего изображения.

Рассматривается, как влияет предложенный подход на скорость сжатия. Описывается класс реальных приложений, которые могут использовать данный подход, и сложность построения универсальной меры качества изображений.

КЛЮЧЕВЫЕ СЛОВА

Фрактальное сжатие, оптимизация алгоритма, PSNR, IFS, image compression, fractals, time complexity.

0. ВВЕДЕНИЕ И ПРЕДЫДУЩИЕ РАБОТЫ

ИДЕЯ ДАННОГО ПОДХОДА

Основная задача, которую мы решаем — повышение степени сжатия изображений. Когда практически достигнут предел сжатия изображения в целом и различные методы дают очень небольшой выигрыш, мы можем существенно (до 2 раз) увеличить степень сжатия за счет изменения качества разных участков изображения.

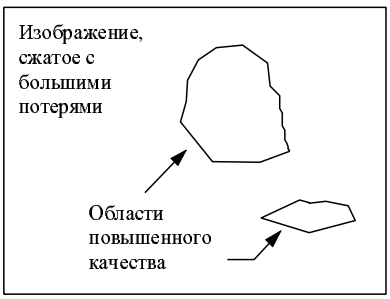

Рисунок 1. Локальное улучшение качества областей изображения

Проблемой этого подхода является то, что необходимо каким-то образом получать расположение наиболее важных для человека участков изображения. Например, таким участком на фотографии человека является лицо, а на лице — глаза. Если при сжатии портрета с потерями с большим коэффициентом будут размыты предметы, находящиеся на заднем плане — это будет не существенно. Однако, если будет размыто лицо или глаза — экспертная оценка степени потерь будет низкой.

Работы по автоматическому выделению таких областей сейчас ведутся. В частности, созданы алгоритмы автоматического выделения лиц на изображениях. Идут попытки выделения наиболее значимых контуров и т.д. Однако очевидно, что универсальный алгоритм в ближайшее время создан не будет, поскольку для этого требуется построить схему восприятия изображения мозгом человека.

На сегодня вполне реально применение полуавтоматических систем, в которых качество областей изображения будет задаваться интерактивно. Данный подход уменьшает количество возможных областей применения модифицированного алгоритма, но *позволяет достичь большей степени сжатия*.

КЛАССЫ ПРИЛОЖЕНИЙ

Рассмотрим ограничения на требования приложений к алгоритму:

1) Для приложения должна быть критична (максимальна) степень компрессии, причем настолько, что возможен индивидуальный подход к каждому изображению.
2) Изображение должно сжиматься один раз, а распаковываться множество раз.
3) Изображение должно распаковываться быстро.
4) Скорость сжатия не должна играть роли.

В качестве примеров приложений, удовлетворяющим этим ограничениям, можно привести практически все мультимедийные продукты на CD-ROM. И для CD-ROM энциклопедий, и для игр важно записать на диск как можно больше информации, а графика, как правило, занимает до 70% всего объема диска. При этом технология производства дисков позволяет сжимать каждое изображение индивидуально, максимально повышая степень сжатия.

Интересным примером являются WWW сервера. Для них тоже, как правило, выполняются изложенные выше ограничения. Технологии plug-ins и использование Java позволяют хранить изображения на сервере в любом формате. При этом совершенно не обязательно индивидуально

подходить **к каждому** изображению, поскольку по статистике 10% изображений будут запрашиваться 90% раз. Т.е. для крупных справочных или игровых серверов появляется уменьшать время загрузки изображений и степень загруженности каналов связи адаптивно.

ВВЕДЕНИЕ ВО ФРАКТАЛЬНУЮ КОМПРЕССИЮ

Фрактальный алгоритм, впервые описанный в [1], теперь достаточно хорошо известен [2]. Основным недостатком алгоритма является то, что при осуществлении сжатия изображений требуются значительные вычислительные ресурсы, в то время как *разархивация происходит очень быстро*.

Как известно, фрактальный алгоритм позволяет достичь самых высоких степеней сжатия (до 1000 раз) для сложных по своей структуре изображений. Применение фрактального сжатия для небольших изображений и небольших степеней сжатия не дает существенного выигрыша, поскольку мы получаем аналогичные результаты для JPEG и Wavelet алгоритмов. Однако при больших степенях сжатия применение фрактального алгоритма позволяет получить заметный выигрыш. Последнее положение и определяет *актуальность излагаемого подхода для фрактального алгоритма* сжатия изображений.

Основная сложность в применении нашего подхода к фрактальному алгоритму заключается в том, что он использует в качестве «словаря» блоков само изображение при восстановлении. Т.е. ухудшение качества в какой-то области влечет за собой ухудшение качества всего изображения. Для того, чтобы избавиться от этого эффекта, требуется существенно изменить схему алгоритма компрессии. Подчеркнем, что *алгоритм декомпрессии при применении данного подхода не изменяется* совершенно.

Кроме того, при использовании данного подхода полностью пропадает возможность использования любых мер определения степени потерь. Только экспертная оценка дает адекватный результат.

ПРЕДЫДУЩИЕ РАБОТЫ

Излагаемый подход основан на изменении разбиения изображения при сжатии. К настоящему моменту исследовано довольно много различных схем разбиения. Их можно классифицировать как:

- Схемы *иерархического* разбиения (квадродеревом [3], полигональное [4] и HV разбиение [5]) при которых происходит только дробление элементов в процессе итераций;
- Схемы *итеративного* разбиения (триангуляция Делоне [6], прямоугольниками [7], эвристический поиск [8], эволюционное [9] и адаптивное [10] разбиения) при которых происходит дробление и объединение элементов в процессе итераций.

Ближе всего к теме данной статьи вопросы построения оптимального иерархического разбиения, которые также можно использовать для квадродеревьев, рассматриваемые с [11].

1. ОПИСАНИЕ АЛГОРИТМА

ОСНОВНАЯ ИДЕЯ

Основная идея реализации предложенной идеи достаточно проста: изменяется критерий выбора блока. Нам необходимо брать блоки для приближения заданного участка не из участка, наилучшим образом приближающего данный, а из участка, который достаточно хорошо приближает данный и не сильно страдает при компрессии. Это позволяет повышать качество выбранного участка не только за счет уменьшения дробления разбиения квадродерева и минимизирует потери от неравномерного приближения разных участков изображения. Поскольку при прочих равных для приближения «качественного» участка используются блоки из других «качественно» приближенных участков. В качестве критерия оптимального блока берется то же среднеквадратичное расстояние с учетом поправки. Если в качестве исходного расстояния между блоками берется:

$$[\min]\ R = \sum_{i=1,j=1}^{n,n} (s \cdot a_{i,j} + o - b_{i,j})^2$$

где $s \in [0,1]$ и o могут быть явно получены по методу наименьших квадратов, то новый критерий выглядит так:

$$R' = R + s \sum_{i=1,j=1}^{n,n} r_{i,j}$$

где $r_{i,j} = \dfrac{R_{i,j}}{n^2}$ — удельное приближение точки блока A, который переводится в блок B в данном преобразовании.

Сложность этого подхода заключается в том, что критерий оптимальности (маску качества $r_{i,j}$ приближения изображения) мы получаем только после сжатия. Т.е. при ускорении алгоритма и отказе от полного перебора, мы не только не достигаем оптимума, но и вынуждены пользоваться смещенным критерием.

ПОСТРОЕНИЕ АЛГОРИТМА

Введем понятия доменных и ранговых областей. Области, D_i *из которых* производится выбор подобных областей называются *ранговыми* (A); области R_i, *для которых* подбираются подобные области, называются *доменными* (B).

Если исходный алгоритм кратко записать как:

Шаг 1. Подготовка всех структур данных.
Шаг 2. Поиск наилучшего блока:

```
for (по всем подготовленным ранговым блокам) {
  for (по всем доменным блокам) {
     current= сохраним текущие координаты;
     D=image->скопируем из изображения блок с
        текущими коорд.;
     current_dist=Среднеквадр._расст.(D Rij);
     if(current_dist < distij) {
        distij = запомним расстояние;
        bstij = запомним лучшие координаты;
     }
  }
}
Сохраним_Лучшие_Коэффициенты(все best);
```

В приведенном коде `image` — сжимаемое изображение, $dist_{ij}$ — массив в котором хранятся наименьшие расстояния, а bst_{ij} — массив с коэффициентами преобразований. Данная схема упрощена, поскольку на практике, при построении разбиения в виде квадродерева, используются более сложные структуры.

Модифицированный алгоритм отличается тем, что мы реально запоминаем не одно преобразование для каждого приближаемого блока, а создаем сортированный список $\mathbf{bestl_{ij}}$ таких преобразований (т.е. используется уже не массив структур коэффициентов, а массив сортированных списков этих структур), упорядоченных по среднеквадратичному расстоянию **current_dist**. На псевдокоде изменения будут выглядеть так:

```
for (по всем подготовленным ранговым блокам) {
  for (по всем доменным блокам) {
     current= сохраним текущие координаты;
     D=image->скопируем из изображения блок с
        текущими коорд.;
     current_dist=Среднеквадр._расст.(D Rij);
     Вставить_в(bestij, координаты, current_dist);
  }
}
//Подготовим оценочное изображение.
for (по всем bestij) {
  Закрасить в оц. изобр. квадрат с
  яркостью, соответствующей точности
  приближения лучшего преобразования
}
for (по всем bestij) {
  for (по всем преобразованиям в bestij)
     Выбрать в списке лучшее преобразование
}
Сохраним_Лучшие_Коэффициенты(все best);
```

Рисунок 2-1. Тестовое изображение Barbara 512x512 Grayscale, сжатое только с учетом минимизации среднеквадратичного расстояния

На рисунке 2-1 представлено тестовое изображение Barbara, сжатое с использованием исходного алгоритма. Данная фотография является стандартной при сравнении алгоритмов сжатия, поскольку содержит много неприятных «полосатых» текстур (скатерть, кресло, одежда), на которых многие алгоритмы сжатия с потерями либо создают муаровые узоры, либо существенно теряют качество. Использованный фрактальный алгоритм на этом изображении, достаточно хорошо воспроизводит эти области за счет использования гибридной схемы с алгоритмом квантования векторов [12]. Однако существенная для восприятия часть изображения — лицо достаточно существенно потеряло в качестве.

Рисунок 2-2. Откорректированное тестовое изображение Barbara.

На откорректированном изображении (рис. 2-2.) видно, как возросло качество лица и правой руки. В увеличенном виде изменения на лице хорошо видны на рисунках 2-3.

Рисунок 2-3. Участок локального улучшения качества в увеличенном виде.

Как уже было сказано выше, реально при построении системы итерируемых функций, приближающих данное изображение, используется разбиение в виде квадродерева. В нашем алгоритме критерием дальнейшего разбиения узла квадродерева служит пороговое значение среднеквадратичного расстояния приближения соответствующего узла. Не останавливаясь в данной статье на подробностях реализации взвешенного вычисления пороговых величин [2, 3, 11] опишем, как они вычисляются в модифицированном алгоритме.

Обозначим как B старое взвешенное предельное значение среднеквадратичного расстояния. Тогда новое вычисляется как:

$$B' = B \cdot \left(1 - \frac{\sum_{i=x}^{x+size} \sum_{j=y}^{y+size} q_{i,j}}{255 \cdot size^2} \right)$$

Где q это яркость точек в изображении, используемом как маска (рис. 3). Фактически по этому изображению строятся весовые коэффициенты, влияющие на разбиение квадродерева. Как *size* обозначен размер соответствующего блока в квадродереве.

Рисунок 3. Вид изображения-маски

Оценочные изображения, получаемые на последнем шаге алгоритма, задают, насколько точно приближен соответствующий блок квадродерева.

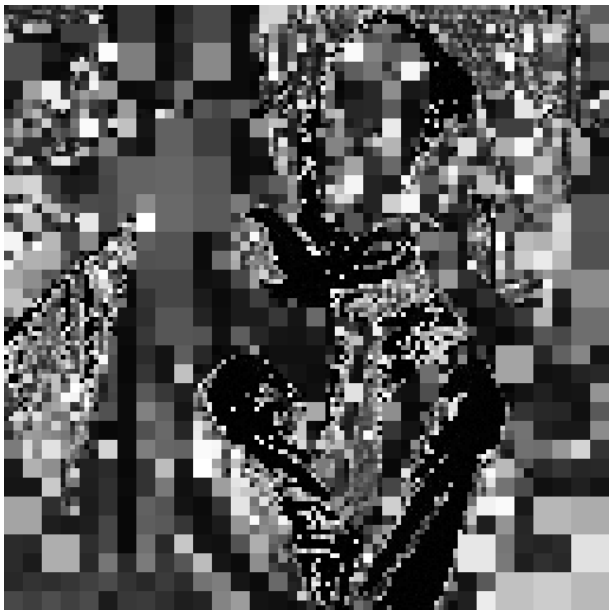

Рисунок 4-1. Степень приближения блоков квадродерева для оригинального алгоритма.

На рисунках 4-1 и 4-2 более яркие квадраты изображения соответствуют худшему приближению, а более светлые — лучшему. Хорошо видно, как более сложные участки изображения представлены более мелкой сеткой. Использование маски дает заметное на глаз снижение яркости на соответствующих участках изображения.

Рисунок 4-2. Степень приближения блоков квадродерева для модифицированного алгоритма — маска, использованная для отбора блоков.

Если рассмотреть модуль разности между элементами изображения (рис. 5, контрастность увеличена в 32 раза), то видно, насколько данный алгоритм улучшает приближение выбранных областей изображения. При этом качество остального изображения и степень компрессии практически не изменяются.

Рисунок 5. Разница между исходным изображением и изображениями, полученными после декомпрессии, для оригинального и модифицированного алгоритмов.

Построенная модификация фрактального алгоритма практически не изменяет время компрессии и совершенно не влияет на процедуру декомпрессии. Последнее весьма существенно для реальных приложений.

2. РЕЗУЛЬТАТЫ

ВРЕМЯ И СТЕПЕНЬ СЖАТИЯ

Основным результатом работы является то, что построен реально функционирующий алгоритм способный адаптивно изменять качество областей изображения. Степень сжатия уменьшается пропорционально размеру светлых областей в изображении-маске. Время компрессии увеличивается крайне незначительно. Даже на Pentium 200 это меньше секунды на изображение 512x512. Т.е. даже при максимальной скорости компрессии замедление не превышает 2%.

ПОВЫШЕНИЕ КАЧЕСТВА ИЗОБРАЖЕНИЯ

Модификация алгоритма такова, что даже если не производится локальное улучшение качества изображения, общее качество изображений увеличивается. На рис. 6 приведено изменение PSNR тестовых изображений для разных скоростей работы алгоритма (используется разное количество классов разбиения множества блоков [3]). Улучшение качества незначительно, но стабильно.

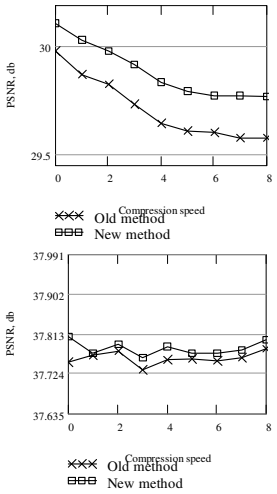

Рисунок 6. Улучшение качества изображения для изображений Barbara и Dmc

При большей степени компрессии, как и ожидалось, наблюдаемый выигрыш больше, чем для невысоких степеней. Это повышает актуальность фрактального алгоритма, поскольку при невысоких степенях компрессии выгоднее применять JPEG.

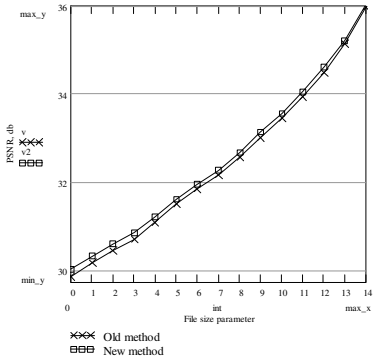

Рисунок 7. Улучшение качества изображения для изображений Barbara для разного параметра размера файла

СТАТИСТИКА ИЗМЕНЕНИЯ ПРЕОБРАЗОВАНИЙ

Как уже говорилось, блоки в новом алгоритме помещаются в сортированный список, из которого потом производится выбор оптимального. Выбор нулевого элемента списка — выбор элемента с наименьшим найденным среднеквадратичным расстоянием — соответствует оригинальному алгоритму. Посмотрим, как распределяется выбор первых пяти элементов списка на первых четырех уровнях квадродерева (рис. 8).

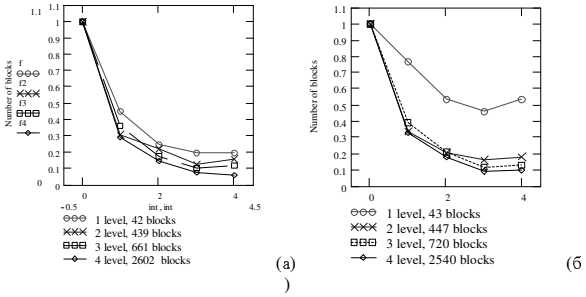

Рисунок 8. Изображение Barbara распределение выбора координат преобразований для малого (а) и большого (б) времени компрессии для сжатия в 7 раз

Для удобства сравнения количество преобразований нормировано относительно преобразований, оставленных без изменений. Хорошо видно, что при увеличении времени компрессии до полного перебора (б) увеличивает количество выбранных ненулевых преобразований. Однако на рис. 9-1 хорошо видно, что в массе своей увеличение времени компрессии не изменяет распределения номеров выбранных преобразований. На изображении Dmc можно репрезентативно продемонстрировать изменение распределения в большем диапазоне степеней компрессии.

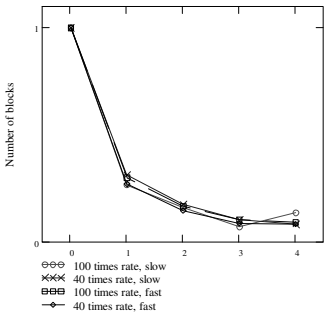

Рисунок 9-1. Распределение выбора координат всех блоков для изображения Dmc малого (slow) и большого (fast) времени компрессии

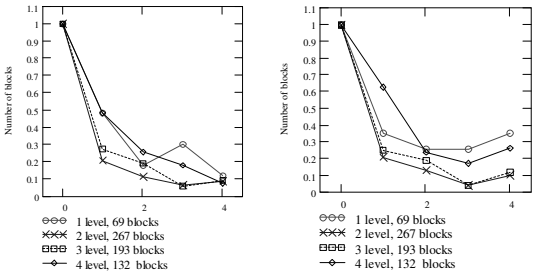

Рисунок 9-2. Изображение Dmc распределение для малого (а) и большого (б) времени компрессии для сжатия в 100 раз

Ломаный характер кривых объясняется малым количеством блоков на этих уровнях квадродерева. На рисунке 10 приведено качество изображения для сжатия в 100 раз. При сжатии в 40 раз, параметры которого иллюстрируются на рис. 9-1, качество изображения уже не позволяет сказать, что изображение подвергалось сжатию.

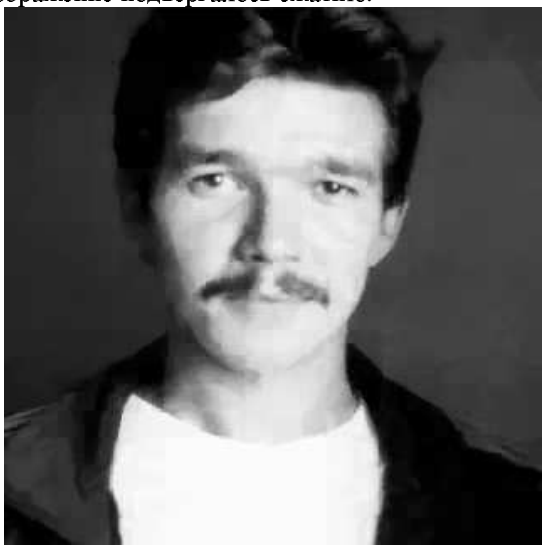

Рисунок 10. Изображение 330x320 TrueColor Dmc сжатое в 100 раз

3. ЗАКЛЮЧЕНИЕ И ПЕРСПЕКТИВЫ РАБОТЫ

Дальнейшая работа в этом направлении будет связана с дальнейшим совершенствованием выбора блока. В частности после проведения компрессии будет проведена декомпрессия и замерено расстояние у преобразований в начале сортированного списка преобразований с получившимся изображением. Это существенно уточнит выбор блока, поскольку будет учтено не только то, что в этой области несколько блоков приблизили ранговый блок с такой-то точностью, но и то, в какой части блока и какая погрешность была допущена.

Более того, поскольку практически половина преобразований меняется после проведения выбора, имеет смысл провести процедуру выбора не один раз.

Такой подход позволит еще более увеличить качество изображения при высоких степенях компрессии.

ЛИТЕРАТУРА

[1] *A.E. Jacquin*. Image coding based on a fractal theory of iterated contractive image transformations. // IEEE Trans. Image Processing 1 18-30, 1992;

[2] Fractal Image Compression — Theory and Application to Digital Images. Y. Fisher (ed.). Springer-Verlag, New York 1994;

[3] *Y. Fisher*. Fractal image compression with quadtrees. In: [2];

[4] E. Reusens, Partitioning complexity issue for Iterated Function Systems based image coding, in Proc. of VII European Signal Processing Conference, Vol. 1, Edinburg, U.K., September 1994, pp. 171-174

[5] *Y. Fisher, S. Menlove*, Fractal encoding with HV partitions, in [2];

[6] *F. Davoine, M. Antonini, J.-M. Chassery, M. Barlaud,* Fractal image compression based on Delaunay triangulation and vector quantization, in Proc. of IEEE Transaction on Image Procession, Vol. 5, No. 2, February 1996, pp. 338-346;

[7] *F. Davoine, J. Svensson, J.-M. Chassery,* A mixed triangular and quadrilateral partition for fractal image coding, in Proc. of IEEE International Conference on Image Processing, Washington, D.C, 1995;

[8] *L. Thomas, F. Deravi,* Region-based fractal image compression using heuristic search, in Proc. of IEEE Transaction on Image Procession, Vol. 4, No. 6, June 1995, pp. 832-838;

[9] *D. Saupe, H. Hartenstein,* Evolutionary fractal image compression, in Proc. of IEEE International Conference on Image Processing, Laussane, September 1996, pp. 129-132;

[10] *M. Ruhl, H. Hartenstein, D. Saupe,* Adaptive partitioning for fractal image compression, in Proc. of IEEE International Conference on Image Processing, Santa Barbara, October 1997;

[11] *D. Saupe, M. Ruhl, R. Hamzaoui, L. Grandi, D. Marini,* Optimal hierarchical partitions for fractal image compression, in Proc. of IEEE International Conference on Image Processing, Chicago, October 1998;

[12] *Д.С. Ватолин*. Гибридная схема фрактальной компрессии и квантования векторов для малых блоков. // Материалы Graphicon-98, 1998;

[13] *Д.С. Ватолин*. Тенденции развития алгоритмов архивации графики. // Открытые системы, N-4. Зима 1995;

[14] *Д.С. Ватолин.* Использование ДКП для ускорения фрактального сжатия изображений. // Программирование, N 4 1999.

Visual Quality Increasing using Image Parts Improvement for Fractal Image Compression

Dmitriy S. Vatolin (dmitriy@amc.ru)
Moscow State University

In the given paper it is offered to change fractal image compression. Most important image areas better quality saving, allows to achieve the greater final compression rate at the same visual image quality.

Influence of the offered approach to compression speed is considered. The set of real applications which can use the given approach is described.