Фрактальное сжатие

Фрактальное сжатие изображений

- В декабре 1992 года, перед самым Рождеством, компания Microsoft выпустила свой новый компакт-диск Microsoft Encarta. С тех пор эта мультимедиа-энциклопедия, содержащая информацию о животных, цветах, деревьях и живописных местах, не покидает списки наиболее популярных энциклопедий на компакт-дисках. В недавнем опросе Microsoft Encarta опять заняла первое место, опередив ближайшего конкурента - Комптоновскую мультимедиа-энциклопедию. Причина подобной популярности кроется в удобстве использования, высоком качестве статей и, главное, в большом количестве материалов. На диск записано 7 часов звука, 100 анимационных роликов, примерно 800 масштабируемых карт, а также 7000 качественных фотографий. И все это - на одном диске! Напомним, что обычный компакт-диск в 650 Мбайт без использования компрессии может содержать либо 56 минут качественного звука, либо 1 час видео разрешения с разрешением 320x200 в формате MPEG-1, либо 700 полноцветных изображений размером 640x480.

- Чтобы разместить больше информации, необходимы достаточно эффективные алгоритмы архивации. Мы не будем останавливаться на методах архивации для видео и звука. Речь пойдет о новом перспективном алгоритме - фрактальном сжатии графической информации.
- Когда в 1991 году впервые была опубликована информация о возможностях фрактального сжатия, она наделала много шуму. Майкл Барнсли, один из разработчиков алгоритма, утверждал, что разработан способ нахождения коэффициентов фрактала, сколь угодно близкого к исходной картинке.
- Фракталы, эти красивые образы динамических систем, ранее использовались в машинной графике в основном для построения изображений неба, листьев, гор, травы. Красивое и, что важнее, достоверно имитирующее природный объект изображение могло быть задано всего несколькими коэффициентами. Неудивительно, что идея использовать фракталы при сжатии возникала и раньше, но считалось практически невозможным построить соответствующий алгоритм, который подбирал бы коэффициенты за приемлемое время.

- Итак, в 1991 году такой алгоритм был найден. Кроме того, в дальнейших его статьях декларировался ряд уникальных возможностей новой технологии. Так, фрактальный архиватор позволяет, например, при распаковке произвольно менять разрешение (размеры) изображения без появления эффекта зернистости. Более того, он распаковывает гораздо быстрее, чем ближайший конкурент JPEG, и не только статическую графику, но и видео. В качестве примера приводилась программа, показывающая на машине с процессором i386/33 МГц цветной видеофильм с частотой 20 кадров в секунду без всякой аппаратной поддержки. В отличие от JPEG, в алгоритм изначально заложена возможность управлять степенью потерь на участках с максимальными потерями качества. Коэффициент сжатия изображений в целом примерно как у JPEG, но на некоторых реальных картинках достигалось сжатие в 10000 (!) раз.
- Звучит это более чем внушительно, поэтому необходимо спокойно разобраться с преимуществами, которые обещает фрактальная компрессия, а также с возможными неприятными сторонами этого алгоритма.

История фрактального сжатия

- Рождение фрактальной геометрии обычно связывают с выходом в 1977 году книги Б. Мандельброта "Фрактальная геометрия природы". Одна из основных идей книги заключалась в том, что средствами традиционной геометрии (то есть используя линии и поверхности), чрезвычайно сложно представить природные объекты. Фрактальная геометрия задает их очень просто.
- В 1981 году Джон Хатчинсон опубликовал статью "Фракталы и самоподобие", в которой была представлена теория построения фракталов с помощью системы итерируемых функций (IFS, Iterated Function System).
- Четыре года спустя появилась статья Майкла Барнсли и Стефана Демко, в которой приводилась уже достаточно стройная теория IFS. В 1987 году Барнсли основал Iterated Systems, компанию, основной деятельностью которой является создание новых алгоритмов и ПО с использованием фракталов.

- Всего через год, в 1988 году, он выпустил фундаментальный труд "Фракталы повсюду". Помимо описания IFS, в ней был получен результат, известный сейчас как Collage Theorem, который лежит в основе математического обоснования идеи фрактальной компрессии.
- Если построение изображений с помощью фрактальной математики можно назвать прямой задачей, то построение по изображению IFS - это обратная задача. Довольно долго она считалась неразрешимой, однако Барнсли, используя Collage Theorem, построил соответствующий алгоритм. (В 1990 и 1991 годах эта идея была защищена патентами.) Если коэффициенты занимают меньше места, чем исходное изображение, то алгоритм является алгоритмом архивации.
- Первая статья об успехах Барнсли в области компрессии появилась в журнале BYTE в январе 1988 года. В ней не описывалось решение обратной задачи, но приводилось несколько изображений, сжатых с коэффициентом 1:10000, что было совершенно ошеломительно.

- Но практически сразу было отмечено, что несмотря на броские названия ("Темный лес", "Побережье Монтере", "Поле подсолнухов") изображения в действительности имели искусственную природу. Это, вызвало массу скептических замечаний, подогреваемых еще и заявлением Барнсли о том, что "среднее изображение требует для сжатия порядка 100 часов работы на мощной двухпроцессорной рабочей станции, причем с участием человека".
- Отношение к новому методу изменилось в 1992 году, когда Арнауд Джеквин, один из сотрудников Барнсли, при защите диссертации описал практический алгоритм и опубликовал его. Этот алгоритм был крайне медленным и не претендовал на компрессию в 10000 раз (полноцветное 24-разрядное изображение с его помощью могло быть сжато без существенных потерь с коэффициентом 1:8 - 1:50); но его несомненным достоинством было то, что вмешательство человека удалось полностью исключить. Сегодня все известные программы фрактальной компрессии базируются на алгоритме Джеквина. В 1993 году вышел первый коммерческий продукт компании Iterated Systems.

- Ему было посвящено достаточно много публикаций, но о коммерческом успехе речь не шла, продукт был достаточно "сырой", компания не предпринимала никаких рекламных шагов, и приобрести программу было тяжело.
- В 1994 году Ювал Фишер предоставил во всеобщее пользование исходные тексты исследовательской программы, в которой использовалось разложение изображения в квадродерево и были реализованы алгоритмы оптимизации поиска. Позднее появилось еще несколько исследовательских проектов, которые в качестве начального варианта программы использовали программу Фишера.
- В июле 1995 года в Тронхейме (Швеция) состоялась первая школа-конференция, посвященная фрактальной компрессии. Таким образом, многие важные события в области фрактальной компрессии произошли за последние три года: алгоритм только-только начинает развиваться.

Идея

- Фрактальная архивация основана на том, что с помощью коэффициентов системы итерируемых функций изображение представляется в более компактной форме. Прежде чем рассматривать процесс архивации, разберем, как IFS строит изображение.
- Строго говоря, IFS - это набор трехмерных аффинных преобразований, переводящих одно изображение в другое. Преобразованию подвергаются точки в трехмерном пространстве (x координата, y координата, яркость).
- Наиболее наглядно этот процесс продемонстрировал сам Барнсли в своей книге "Фрактальное сжатие изображения". В ней введено понятие Фотокопировальной Машины, состоящей из экрана, на котором изображена исходная картинка, и системы линз, проецирующих изображение на другой экран. Каждая линза проецирует часть исходного изображения. Расставляя линзы и меняя их характеристики, можно управлять получаемым изображением. На линзы накладывается требование они должны уменьшать в размерах проектируемую часть изображения. Кроме того, они могут менять яркость фрагмента и проецируют не круги, а области с произвольной границей.

- Одна шаг Машины состоит в построении с помощью проецирования по исходному изображению нового. Утверждается, что на некотором шаге изображение перестанет изменяться. Оно будет зависеть только от расположения и характеристик линз и не будет зависеть от исходной картинки. Это изображение называется неподвижной точкой или аттрактором данной IFS. Collage Theorem гарантирует наличие ровно одной неподвижной точки для каждой IFS. Поскольку отображение линз является сжимающим, каждая линза в явном виде задает самоподобные области в нашем изображении. Благодаря самоподобию мы получаем сложную структуру изображения при любом увеличении.
- Наиболее известны два изображения, полученных с помощью IFS треугольник Серпинского и папоротник Барнсли Первое задается тремя, а второе - пятью аффинными преобразованиями (или, в нашей терминологии, линзами). Каждое преобразование задается буквально считанными байтами, в то время, как изображение, построенное с их помощью, может занимать и несколько мегабайт.

- Становится понятно, как работает архиватор, и почему ему требуется так много времени. Фактически, фрактальная компрессия - это поиск самоподобных областей в изображении и определение для них параметров аффинных преобразований.
- В худшем случае, если не будет применяться оптимизирующий алгоритм, потребуется перебор и сравнение всех возможных фрагментов изображения разного размера. Даже для небольших изображений при учете дискретности мы получим астрономическое число перебираемых вариантов. Даже резкое сужение классов преобразований, например, за счет масштабирования только в определенное число раз, не позволит добиться приемлемого времени. Кроме того, при этом теряется качество изображения. Подавляющее большинство исследований в области фрактальной компрессии сейчас направлены на уменьшение времени архивации, необходимого для получения качественного изображения.

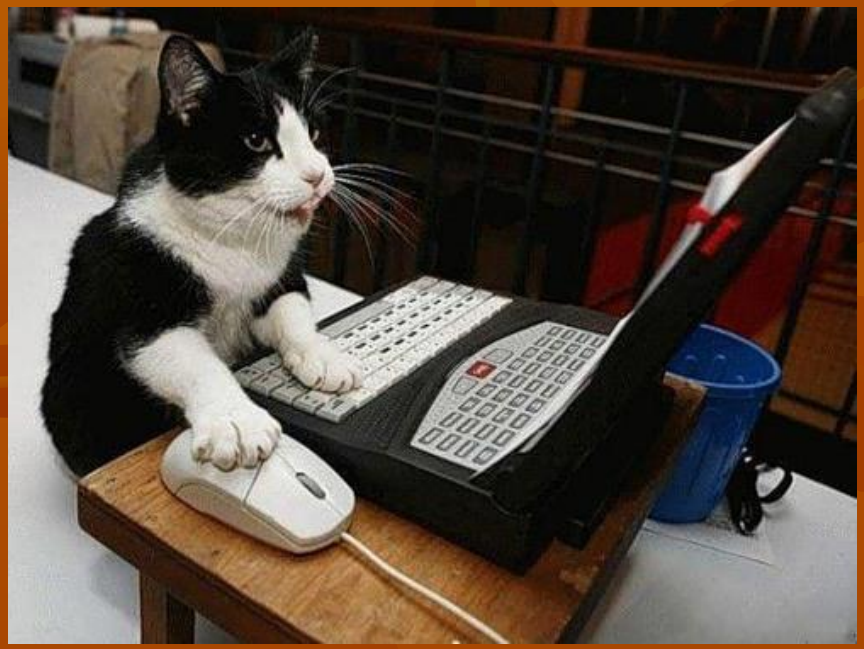
Оценка потерь и способы их регулирования

- До сих пор мы не затронули несколько важных вопросов. Например, что делать, если алгоритм не может подобрать для какого-либо фрагмента изображения подобный ему? Достаточно очевидное решение - разбить этот фрагмент на более мелкие и попытаться поискать для них. Однако понятно, что процедуру эту нельзя повторять до бесконечности, иначе количество необходимых преобразований станет так велико, что алгоритм перестанет быть алгоритмом компрессии. Следовательно, допускаются потери в какой-то части изображения.

- Для фрактального алгоритма компрессии, как и для других алгоритмов сжатия с потерями, очень важны механизмы, с помощью которых можно будет регулировать степень сжатия и степень потерь. К настоящему времени разработан достаточно большой набор таких методов. Во-первых, можно ограничить количество 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 (для него обязательно заводится несколько линз). В-третьих, можно запретить дробить фрагменты размером меньше, допустим, четырех точек. Изменяя пороговые значения и приоритет этих условий, можно очень гибко управлять коэффициентом компрессии изображения: от побитного соответствия, до любой степени сжатия.

Возможности масштабирования

- Итак, мы выяснили, что IFS задает фрактальную структуру, сколь угодно близкую к нашему изображению. При внимательном рассмотрении процесса построения изображения с ее помощью становится понятно, что восстанавливаемое изображение может иметь любое (!) разрешение. В самом деле, возвращаясь к аналогии с Фотокопировальной Машиной, можно сказать, что нам не важно до какой сетки растра будет огрубляться установившееся неподвижное изображение. Ведь Машина работает вообще с непрерывными экранами.
- На этапе архивации проводится распознавание изображения, и в виде коэффициентов хранится уже не растровая информация, а информация о структуре самого изображения. Именно это и позволяет при развертывании увеличивать его в несколько раз. Особенно впечатляют примеры, в которых при увеличении изображений природных объектов проявляются новые детали, действительно этим объектам присущие (например, когда при увеличении фотографии скалы она приобретает новые, более мелкие неровности).



- Но не все так гладко, как может показаться. Если изображение однородно (на фотографии только скала), то при увеличении получаются отличные результаты, однако, если сжимать изображение натюрморта, то предсказать, какие новые фрактальные структуры возникнут, очень сложно. Впрочем, вдвое-втрое можно увеличить практически любое изображение, при архивации которого задавалась небольшая степень потерь.
- Масштабирование - уникальная особенность, присущая фрактальной компрессии. Со временем ее, видимо, будут активно использовать как в специальных алгоритмах масштабирования, так и во многих приложениях. Действительно, этого требует концепция "приложение в окне". Было бы неплохо, если бы изображение, показываемое в окне 100x100, хорошо смотрелось при увеличении на полный экран - 1024x768.

Сравнение с JPEG

- Сегодня наиболее распространенным алгоритмом архивации графики является JPEG. Сравним его с фрактальной компрессией.
- Во-первых, заметим, что и тот, и другой алгоритм оперируют 8-битными (в градациях серого) и 24-битными полноцветными изображениями. Оба являются алгоритмами сжатия с потерями и обеспечивают близкие коэффициенты архивации. И у фрактального алгоритма, и у JPEG существует возможность увеличить степень сжатия за счет увеличения потерь. Кроме того, оба алгоритма очень хорошо распараллеливаются.
- Различия начинаются, если мы рассмотрим время, необходимое алгоритмам для архивации/разархивации. Так, фрактальный алгоритм сжимает в сотни и даже в тысячи раз дольше, чем JPEG. Распаковка изображения, наоборот, произойдет в 5-10 раз быстрее. Поэтому, если изображение будет сжато только один раз, а передано по сети и распаковано множество раз, то выгодней использовать фрактальный алгоритм.

- JPEG использует разложение изображения по косинусоидальным функциям, поэтому потери в нем (даже при заданных минимальных потерях) проявляются в волнах и ореолах на границе резких переходов цветов. Именно за этот эффект его не любят использовать при сжатии изображений, которые готовят для качественной печати: там этот эффект может стать очень заметен.
- Фрактальный алгоритм избавлен от этого недостатка. Более того, при печати изображения каждый раз приходится выполнять операцию масштабирования, поскольку растр (или линиатура) печатающего устройства не совпадает с растром изображения. При преобразовании также может возникнуть несколько неприятных эффектов, с которыми можно бороться либо масштабируя изображение программно (для дешевых устройств печати типа обычных лазерных и струйных принтеров), либо снабжая устройство печати своим процессором, винчестером и набором программ обработки изображений (для дорогих фотонаборных автоматов). Как можно догадаться, при использовании фрактального алгоритма таких проблем практически не возникает. Вытеснение JPEG фрактальным алгоритмом в повсеместном использовании произойдет еще не скоро (хотя бы в силу низкой скорости архивации последнего), однако в области приложений мультимедиа, в компьютерных играх его использование вполне оправдано.

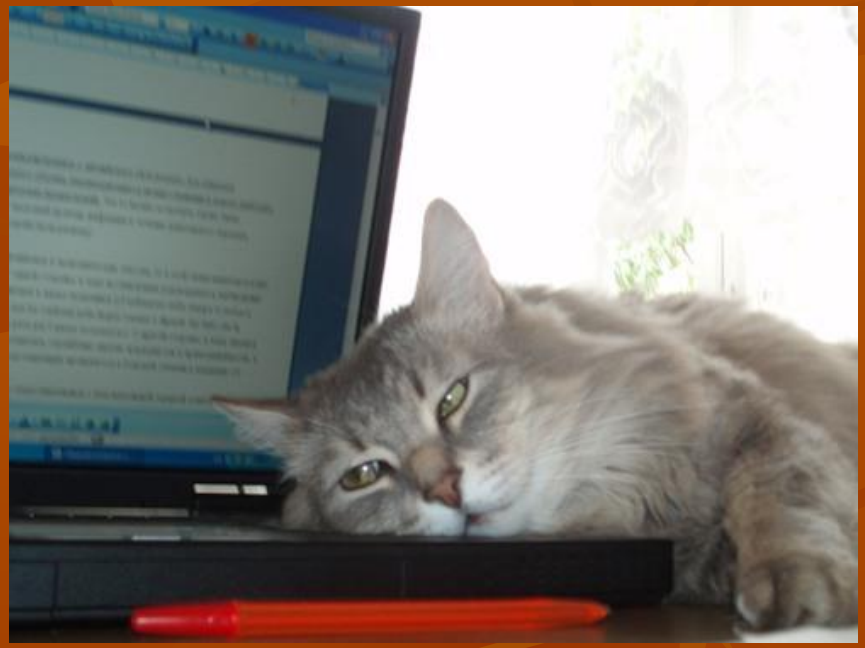
Берегись патентов!

- Специфическая проблема, с которой приходится сталкиваться при реализации алгоритма фрактальной компрессии, - проблема лицензирования. На алгоритм в 1990 году Майкл Барнсли и Алан Слоан получили патент № 4.941.193, а в 1991 году Барнсли получил второй патент N 5.065447 В них рассматривается модифицированная схема представления изображения, названная Разделенная Система Итерируемых Функций (Partitioned IFS), и алгоритм автоматически переводит изображение в PIFS.
- Сложно сказать, как повлияет патентование на дальнейшую судьбу алгоритма. Например, принадлежность девяти патентов на различные модификации арифметического кодирования компании IBM помешало использованию его в алгоритме JPEG. В конце концов оно было заменено кодированием по Хаффману.
- В целом ситуация с патентованием алгоритмов компрессии весьма специфична: эти алгоритмы сложны в разработке, достаточно компактны, они требуются в коммерческих программах. Поскольку многие хотят их использовать, а механизм определения уникальности алгоритмов несовершенен, возникает масса казусов.

- Известный судебный процесс по поводу MS DOS 6.0 был спровоцирован незаконным использованием корпорацией Microsoft алгоритма LZRW1, правами на который обладает компания Stac. Интересно, что правами на такой же алгоритм обладает и фирма Gibson&Graybill, поскольку патентное бюро США не смогло в свое время установить, что выдает патент на очень похожий алгоритм. Отметим также, что Рос Вильямс, именем которого назван алгоритм, не имеет к этим двум патентам никакого отношения...
- На простейшее RLE сжатие, идея которого не сложнее идеи пузырьковой сортировки, выдано три (!) патента. Их текст рассматривается как анекдот (авторам пришлось усложнять и обобщать), но, тем не менее, если вам в голову пришло нечто схожее, при реализации алгоритма в коммерческой программе надо быть очень осторожным.

Возможности видеокомпрессии

- Итак, одной из основных проблем, с которой пришлось столкнуться при построении алгоритма фрактальной компрессии, является поиск самоподобных участков в изображении. Это основная идея, благодаря которой осуществляется сжатие. Подобный метод можно применить и при архивации видео. Как правило, соседние кадры отличаются не сильно, и изменения между ними в основном, состоят в сдвиге, повороте или растяжении какой-либо части изображения. Таким образом, изменения между двумя кадрами можно задать небольшим количеством аффинных преобразований.
- До недавнего времени такой подход рассматривался как утопический, ввиду чрезвычайно большого объема вычислений, требуемых при поиске соответствующих преобразований. Но достижения в области фрактальной архивации статической графики позволяют пересмотреть взгляды. Всего четыре года назад для архивации изображения с помощью фракталов требовалась мощная рабочая станция и многие часы работы.



- Сегодня с той же задачей справляется средний ПК всего за несколько минут. Найдены алгоритмы, существенно оптимизирующие процесс поиска аффинных преобразований. При этом, по-прежнему рано говорить о повсеместном использовании фрактальной архивации для видео, поскольку даже если будет тратиться на один кадр 5 минут машинного времени (что совсем немного), то для архивации десятиминутного ролика (то есть 15 тысяч кадров) потребуется почти два месяца непрерывной работы.
- Время компрессии можно уменьшить, используя информацию о предыдущем кадре. Как правило, движение, начатое на одном из кадров, продолжается достаточно долго. Поэтому, пользуясь информацией о том, какие объекты и как сдвинулись на экране в предыдущем кадре, можно аппроксимировать их движение и на следующий кадр. Можно воспользоваться и тем, что фрактальный алгоритм легко распараллеливается.

Сжатие рисунка

Спасибо за внимание