**Методы компрессии и сжатия изображений**

Графические данные, особенно данные растровых файлов, занимают колоссальное количество дискового пространства. Например, растровое изображение формата А4 в цветовой модели CMYK при разрешении 300 точек на дюйм занимает порядка 30 мегабайт дискового пространства. Хорошо, если файл один, и ваша публикация не использует других изображений (что крайне маловероятно). Ситуация в корне изменяется тогда, когда вы создаете некий шедевр, например, галерею репродукций картин А2 формата, при этом она с трудом умещается на 100 листах, запечатанных с двух сторон. При самых скромных подсчетах (120 мегабайт х 100 листов х 2 стороны у каждого листа), растровые изображения в этом формате при таком количестве листов будут занимать порядка 24 гигабайт дискового пространства. На чем вы собираетесь хранить такую публикацию ? А теперь представьте, что у вас несколько заказчиков, и работы каждого из них хранятся в нескольких вариантах оформления, кроме того, для большинства заказов вы сохранили выполненный проект на разных стадиях его готовности, чтобы в случае желания заказчика все в корне и кардинально изменить, вы могли быстро это выполнить.... Естественно, все эти данные сохранить будет очень и очень сложно. Именно поэтому, а также потому, что дисковое пространство обычно достаточно дорого обходится (не смотря на то, что устройства для хранения цифровой информации постоянно дешевеют, их все время требуется больше и больше, что требует немалых капиталовложений), были изобретены множество методов сжатия данных самого различного типа, в том числе и графических. О наиболее распространенных и широко использующихся мы сейчас поговорим.

В самом начале разделим существующие методы сжатия изображений на две условные категории - на архивацию (сжатие), и на компрессию (конвертирование). Разница между этими способами в том, что второй не подразумевает полного восстановления исходного сохраненного изображения в полном качестве. Но каким бы не был алгоритм компрессии данных, для работы с ним файл нужно проанализировать и распаковать, т. е. вернуть данные в исходный незапакованный вид для их быстрой обработки (обычно это происходит прозрачно для пользователя). Ниже мы рассмотрим способы сжатия подробнее.

Архивация, или сжатие графических данных, возможно как для растровой, так и для векторной графики. При этом способе уменьшения данных, программа анализирует наличие в сжимаемых данных некоторых одинаковых последовательностей данных, и исключает их, записывая вместо повторяющегося фрагмента ссылку на предыдущий такой же (для последующего восстановления). Такими одинаковыми последовательностями могут быть пикселы одного цвета, повторяющиеся текстовые данные, или некая избыточная информация, которая в рамках данного массива данных повторяется несколько раз. Например, растровый файл, состоящий из подложки строго одного цвета (например, серого), имеет в своей структуре очень много повторяющихся фрагментов.

Компрессия (конвертирование) данных - это способ сохранения данных таким образом, при использовании которого не гарантируется (хотя иногда возможно) полное восстановление исходных графических данных. При таком способе хранения данных обычно графическая информация немного "портится" по сравнению с оригинальной, но этими искажениями можно управлять, и при их небольшом значении ими вполне можно пренебречь. Обычно файлы, сохраненные с использованием этого способа хранения, занимают значительно меньше дискового пространства, чем файлы, сохраненные с использованием простой аривации (сжатия). Сильная степень компрессии при использовании второго способа сжатия и дает этому способу хранения данных право на существование (в противном случае, все поголовно использовали бы сжатие без потерь). Как правило, при сохранении данных с использованием компрессии, имеется возможность компромисса между размером выходного файла и его качеством. Понятно, что возможна оптимизация только по одному параметру (чем меньше качество, тем меньше объем выходного файла, и наоборот).

Ниже будут рассмотрены некоторые алгоритмы сжатия данных, их достоинства и недостатки, а также оптимальная их сфера применения (т. е. те типы изображений, к которым данный метод сжатия лучше применять).

Вначале рассмотрим несколько алгоритмов сжатия данных, которые не вносят изменения в исходные файлы и гарантируют полное восстановление данных.

RLE (Run - length encoding) - метод сжатия данных, при котором одинаковые последовательности одних и тех же байт заменяются однократным упоминанием повторяющегося байта (или целой цепочки байтов), и числа его повторений в исходных данных. Например, строка типа 0100 0100 0100 0100 0100 0100 0100 0100, описывающая некую группу пикселов будет заменена на запись типа 0100 х 8, и т.д. Применяется этот тип сжатия в тех случаях, когда изображение имеет большие участки одинакового цвета, цифровое представление которых идентично. В основном, этот тип сжатия применим для монохромных изображний, сохраненных в цветовой модели Bitmap, где при сжатии данных с его использованием можно добиться наилучших результатов. Для сжатия других типов данных (в том числе, и не графических) алгоритм применим, но малоэффективен, так как сжимаемые данные должны иметь простую повторяющуюся структуру). Этот алгоритм имеет еще одно важное преимущество, заключающееся в его относительной простоте, что позволяет быстро производить распаковку из этого формата и упаковку в этот формат (как вы помните, все графические данные для их обработки должны быть предварительно распакованы, а любая компрессия или архивация применяется, в основном, для временного или постоянного хранения файла). В принципе, на основе этого несложного алгоритма, работают более совершенные и более сложные (а также менее быстрые) методы сжатия графических данных, которые мы рассмотрим ниже. Этот метод сжатия графических фанных испольуется для файлов формата PSD, BMP и других.

CCITT Group 3, CCITT Group 4 - Два похожих метода сжатия графических данных, работающие с однобитными изображениями, сохраненными в цветовой модели Bitmap. Основаны на поиске и исключении из исходного изображения дублирующихся последовательностей данных (как в предыдущем типе сжатия, RLE). Различием является лишь то, что эти алгоритмы ориентированы на упаковку именно растровой графической информации, так как работают с отдельными рядами пикселов в изображении. Изначально алгоритм был разработан для сжатия данных, передаваемых через факсимильные системы связи (CCITT Group 3), а более совершенная разновидность этого метода архивации данных (CCITT Group 4) подходит для записи монохромных изображений с более высокой степенью сжатия. Как и предыдущий алгоритм, он, в основном, подходит для сжатия изображений с большими одноцветными областями. Его достоинством является скорость выполнения, а недостатком - ограниченность применения для компрессии графических данных (не все данные удается таким образом эффективно сжать). Этот метод сжатия графических фанных испольуется в файлах формата PDF, PostScript (в инкапсулированных объектах) и других.

LZW (Lemple-Zif-Welch) - алгоритм сжатия данных, основанный на поиске и замене в исходном файле одинаковых последовательностей данных, для их исключения, и уменьшения размера "архива". В отличие от предыдущих рассмотреных методов сжатия, в данном случае производится более "интеллектуальный" просмотр сжимаемого cодержимого, для достижения большей степени сжатия данных. Данный тип сжатия не вносит искажений в исходный графический файл, и подходит для обработки растровых данных любого типа - монохромных, черно - белых, или полноцветных. Наилучшие результаты получаются при компрессии изображений с большими областями одинакового цвета или изображений с повторяющимися одинаковыми структурами. Этот метод позволяет достичь одну из самых наилучших степеней сжатия среди других существующих методов сжатия графических данных, при одновременном полном отсутствии потерь или искажений в исходных файлах. Этот метод сжатия графических фанных испольуется в файлах формата TIFF, PDF, GIF, PostScript (в инкапсулированных объектах) и других.

ZIP - метод сжатия данных, аналогичный методу, использованному в популярном алгоритме архивации PKZip. В основу метода сжатия положен метод, аналогичный LZW. Как и предыдущий метод сжатия данных, этот способ не вносит искажений в исходный файл, и лучше всего подходит для обработки графических данных с одинаковыми одноцветными или повторяющимися областями. Этот метод сжатия графических фанных испольуется в файлах формата PDF, TIFF и некоторых других.

А теперь рассмотрим алгоритмы и методы конвертирования данных, которые вносят изменения в исходные файлы, показывая при этом более высокую степень упаковки графических изображений.

JPEG (Joint Photographic Experts Group) - метод, используемый для хранения полутоновых и полноцветных изображений, позволяющий добиться наивысшей степени сжатия и минимальный размер выходного файла. Основан алгоритм на особенностях восприятия человеческим глазом различных цветов, и достаточно громоздок с вычислительной точки зрения, так как занимает много процессорного времени. Происходит кодирование файла в несколько этапов. Во-первых, изображение условно разбивается на несколько цветовых каналов, для дальнейшего анализа. Затем, изображение разбивается на группы, по 64 пиксела в каждой группе, которые представляют из себя квадратные участки изображения размером 8х8 пикселов, для последующей обработки. Затем, цвет пикселов специальным образом кодируется, исключается дублирующая и избыточная информация, причем при описании цвета большее внимание уделяется скорее яркостной, чем цветовой составляющей, так как человеческий глаз воспринимает больше изменения яркости, чем конкретного цветового тона. Полученные данные сжимаются по RLE или LZW - алгоритму, для получения еще большей компрессии. В результате, на выходе мы получаем файл, иногда в десятки раз меньший, чем его неконвертированный аналог. Однако, чем меньше размер выходного файла, тем меньше степень "аккуратности" при работе программы - конвертора, и, соответственно, ниже качество выходного изображения. Обычно, в программах, позволяющих сохранять растровые данные, возможно задание некоего компромисса между объемом выходного файла и качеством изображения. При наивысшем качестве, обхем выходного файла в 3-5 раз меньше исходного незапакованного. При наименьшем - меньше исходника в десятки раз, но, как правило, при этом качество изображения не позволяет его где-либо использовать. Как правило, для сохранения достойного уровня качества, используют наивысшую из доступных степень качества. Данный формат предназначен для хранения, в основном, фотографических изображений с большим количеством оттенков и цветовых переходов, и практически не подходит для хранения однотонных изображений типа кадров из мультфильмов, скриншотов и пр.(сжатие будет слишком низким, или качество картинки окажется просто недопустимым). Этот метод сжатия графических фанных испольуется в файлах формата PDF, PostScript (в инкапсулированных объектах), собственно, в JPEG и других.

Главным недостатком компрессии с частичной потерей качества, является то, что эти потери, выражающиеся в искажении цветового тона или появлении характерной кубической структуры в контрастных участках изображения (проявляются так называемые артефакты), возникают каждый раз при сохранении изображения, и "накладываются" друг на друга при многократном сохранении файла в этом формате. Поэтому специалисты рекомендуют использовать форматы с частичной потерей качества только для хранения окончательных результатов работы, а не промежуточных рабочих файлов.

Кроме того, в качестве вспомогательных средств, которые могут использоваться для понижения объемов файлов можно рассмотреть изменение цветовой модели графического файла, изменение разрешения растрового файла и ресемплирование (изменение глубины цвета пикселов).

Изменение цветовой модели файла. Например, файлы в цветовом пространстве CMYK больше аналогичных файлов в пространстве RGB на 33% (так как в CMYK имеется дополнительный четвертый черный канал). Если вы не планируете печать ваших файлов, или уверенны в том, что вы сможете корректно провести цветоделение (переход в субтрактивную модель CMYK) позже, вы можете хранить рабочие файлы в RGB.

Изменение разрешения растрового файла. Файл с разрешением 600 точек на дюйм больше своего аналога разрешением в 300 точек в четыре раза, а качество печати при повышенном разрешении не всегда будет выше, нем при номинальном его значении. Так что если разрешение избыточно, можете его понизить. Понижение разрешения файла - необратимый процесс, и никакя последующая интерполяция потом потерянные пикселы не восстановит. Так что при задании необходимого разрешения также будьте внимательны и рациональны. Следует учесть, что параметр разрешения контуров применительно к векторной графике не имеет отношения к объему выходного файла (это уже несколько другое разрешение и несколько другое понятие), а влияет на аккуратность "прорисовки" вектора при его растеризации в устройстве, где производится печать. Так что уменьшение этого параметра для векторной графики не уменьшит объем ваших файлов, а только ухудшит качество печати.

Ресемплирование (изменение глубины цвета растрового изображения) - это изменение начальной глубины цвета файла. Некоторые оцифровывающие устройства выдают растровую информацию с глубиной цвета, превышающую достаточное для печати значение 8 бит на канал. Это иногда оправдано, так как большее значение бит на канал позволяет задавать большее число градаций цвета, что требуется, например, при сильной, "кардинальной" цветокоррекции - сильном осветлении или затенении отдельных участков. Однако, в большинстве случаев для хранения растровых данных в различных цветовых моделях с лихвой достаточно глубины цвета 8 бит на канал. Кроме того, один из стандартов сжатия для RGB изображений подразумевает использование разного количества бит для разных цветовых составляющих (обычно наибольшее количество бит используется для зеленого канала). Также, большинство фильтров Adobe Photoshop рассчитано на работу с изображениями с глубиной цвета в 8 бит (с изображением, использующим нестандартную глубину цвета, становится практически невозможно работать, так как большинство фильтров рассчитаны на значение глубины цвета в 8 бит).

Хорошим примером настройки опций сжатия графики является диалоговое окно Job Options - Compression из программы Adobe Acrobat Distiller, опции которого рассмотрены ниже.

Рис. 1. Диалоговое окно Adobe Acrobat Distiller

Для различных типов изображений, которые могут быть составляющими файла PostScript (про такие объекты говорят, что они инкапсулированы) - для - полноцветных (color bitmap), полутоновых черно-белых (grayscale) и для штриховых объектов (bitmap, 1 bit per pixel) указаны различные установки параметров сжатия, являющиеся оптимальными для создания PDF - документа, оптимизированного для печати и сжатого с минимальными потерями качества. В качестве параметра сжатия изображения выбрана альтернатива JPEG с максимально возможным качеством. Выходное разрешение растровых полноцветных изображений выбрано не более 300 точек на дюйм, и в случае превышения указанного предельного разрешения к изображению будет применен алгоритм бикубической интерполяции с понижением разрешения (bicubic downsampling) - это наиболее медленный, но и наиболее качественный алгоритм интерполяции. Кроме этого алгоритма, допустимо также указать алгоритмы Average downsampling (усреднение значения цвета пикселов) и Subsampling (полное отсутствие интерполяции, берется один из пикселов, и его цвет устанавливается как цвет всего участка изображения). Естественно, последний вариант работает быстрее всех, но и с наименьшим качеством выходного изображения. Для черно - белых полутоновых изображений (grayscale) установки интерполояции, предельного разрешения и метода сжатия аналогичны прерыдущему примеру. Для монохромных изображений, заданных в цветовой модеди Bitmap с глубиной цвета 1 бит на пиксел параметры несколько другие. Во-первых, значение предельного разрешения установлено на уровне 1200 точек на дюйм, что является корректным для изображений в этой цветовой модели (напомню, что спектр разрешений для изображений в этой цветовой модели обычно задается в пределах от 800 до 2540 точек на дюйм). Метод интерполяции в случае превышения разрешения также бикубический, как и в предыдущих примерах, а вот метод сжатия выбран именно ZIP (возможен также выбор CCITT Group 3, CCITT Group 4, или RLE). При этом не происходит никаких потерь качества в выходном изображении. Кроме того, с учетом специфики цветовой модели, невозможно задание алгоритмов сжатия типа JPEG.

Последняя опция - Compress text and line art - подразумевает сжатие текстовых и векторных данных по одному из алгоритмов, подобных ZIP или LZW. Естественно, подразумевается полное восстановление данных при распаковке файла.