

Технологии обработки графических объектов

Компьютерная графика представляет собой одну из современных технологий создания различных изображений с помощью аппаратных и программных средств компьютера, отображения их на экране монитора и затем сохранения в файле или печати на принтере..

Существует два способа представления графических изображений: растровый и векторный. Соответственно различают растровый и векторный форматы графических файлов, содержащих информацию графического изображения.

Растровые форматы хорошо подходят для изображений со сложными гаммами цветов, оттенков и форм. Это такие изображения, как фотографии, рисунки, отсканированные данные.

Векторные форматы хорошо применимы для чертежей и изображений с простыми формами, тенями и окраской.

Растровая графика. Наиболее просто реализовать растровое представление изображения. Растр, или растровый массив (bitmap), представляет совокупность битов, расположенных на сетчатом поле-канве. Бит может быть включен (единичное состояние) или выключен (нулевое состояние). Состояния битов можно использовать для представления черного или белого цветов, так что, соединив на канве несколько битов, можно создать изображение из черных и белых точек.

Растровое изображение напоминает лист клетчатой бумаги, на котором каждая клеточка закрашена черным или белым цветом, в совокупности формируя рисунок.

Основным элементом растрового изображения является пиксел (pixel). Под этим термином часто понимают несколько различных понятий: отдельный элемент растрового изображения, отдельная точка на экране монитора,

отдельная точка на изображении, напечатанном принтером. Поэтому на практике эти понятия часто обозначают так:

пиксел — отдельный элемент растрового изображения;

видеопиксел — элемент изображения на экране монитора;

точка — отдельная точка, создаваемая принтером или фотонаборным автоматом. Цвет каждого пиксела растрового изображения — черный, белый, серый или любой из спектра — запоминается с помощью комбинации битов. Чем больше битов используется для этого, тем большее количество оттенков цветов для каждого пиксела можно получить. Число битов, используемых компьютером для хранения информации о каждом пикселе, называется битовой глубиной или глубиной цвета.

Наиболее простой тип растрового изображения состоит из пикселов, имеющих два возможных цвета — черный и белый. Для хранения такого типа пикселов требуется один бит в памяти компьютера, поэтому изображения, состоящие из пикселов такого вида, называются 1-битовыми изображениями. Для отображения большего количества цветов используется больше битов информации. Число возможных и доступных цветов или градаций серого цвета каждого пиксела равно двум в степени, равной количеству битов, отводимых для каждого пиксела. 24 бита обеспечивают более 16 миллионов цветов. О 24-битовых изображениях часто говорят как об изображениях с естественными цветами, так как такого количества цветов более чем достаточно, чтобы отобразить всевозможные цвета, которые способен различать человеческий глаз.

Основной недостаток растровой графики состоит в том, что каждое изображение для своего хранения требует большое количество памяти. Простые растровые картинки, такие как копии экрана компьютера или черно-белые изображения, занимают до нескольких сотен килобайтов памяти. Детализированные высококачественные рисунки, например, сделанные с

помощью сканеров с высокой разрешающей способностью, занимают уже десятки мегабайтов.

Для разрешения проблемы обработки объемных (в смысле затрат памяти) изображений используются два основных способа:

1. увеличение памяти компьютера;
2. сжатие изображений.

Другим недостатком растрового представления изображений является снижение качества изображений при масштабировании.

Векторная графика. Векторное представление, в отличие от растровой графики, определяет описание изображения в виде линий и фигур, возможно, с закрашенными областями, заполняемыми сплошным или градиентным цветом. Хотя это может показаться более сложным, чем использование растровых массивов, но для многих видов изображений использование математических описаний является более простым способом.

В векторной графике для описания объектов используются комбинации компьютерных команд и математических формул для описания объектов. Это позволяет различным устройствам компьютера, таким как монитор и принтер, при рисовании этих объектов вычислять, где необходимо помещать реальные точки.

Векторную графику часто называют объектно-ориентированной или чертежной графикой. Имеется ряд простейших объектов, или примитивов, например: эллипс, прямоугольник, линия. Эти примитивы и их комбинации используются для создания более сложных изображений. Если посмотреть содержание файла векторной графики, обнаруживается сходство с программой. Он может содержать команды, похожие на слова, и данные в коде АЗСИ, поэтому векторный файл можно отредактировать с помощью текстового редактора. Приведем в условном упрощенном виде команды, описывающие окружность:

объект — окружность;

центр — 50, 70; радиус — 40;

линия: цвет — черный, толщина — 0.50;

заливка — нет.

Данный пример показывает основное достоинство векторной графики — описание объекта является простым и занимает мало памяти. Для описания этой же окружности средствами растровой графики потребовалось бы запомнить каждую отдельную точку изображения, что заняло бы гораздо больше памяти.

Кроме того, векторная графика в сравнении с растровой имеет следующие преимущества:

- простота масштабирования изображения без ухудшения его качества;
- независимость объема памяти, требуемой для хранения изображения, от выбранной цветовой модели.

Недостатком векторных изображений является их некоторая искусственность, заключающаяся в том, что любое изображение необходимо разбить на конечное множество составляющих его примитивов.

Растровая и векторная графика существуют не обособлено друг от друга. Так, векторные рисунки могут включать в себя и растровые изображения. Кроме того, векторные и растровые изображения могут быть преобразованы друг в друга — в этом случае говорят о конвертации графических файлов в другие форматы. Достаточно просто выполняется преобразование векторных изображений в растровые. Не всегда осуществимо преобразование растровой графики в векторную, так как для этого растровая картинка должна содержать линии, которые могут быть идентифицированы программой конвертации (типа CorelTrace в составе пакета CorelDraw) как векторные примитивы. Это

касается, например, высококачественных фотографий, когда каждый пиксел отличается от соседних.

Разрешающая способность — это количество элементов в заданной области. Этот термин применим ко многим понятиям, например, таким как:

- разрешающая способность графического изображения;
- разрешающая способность принтера как устройства вывода;
- разрешающая способность мыши как устройства ввода.

Например, разрешающая способность лазерного принтера может быть задана 300 dpi (dot per inch — точек на дюйм}, что означает способность принтера напечатать на отрезке в один дюйм 300 отдельных точек. В этом случае элементами изображения являются лазерные точки, а размер изображения измеряется в дюймах.

Разрешающая способность графического изображения измеряется в пикселах на дюйм. Отметим, что пиксел в компьютерном файле не имеет определенного размера, так как хранит лишь информацию о своем цвете. Физический размер пиксел приобретает при отображении на конкретном устройстве вывода, например, на мониторе или принтере.

Разрешающая способность технических устройств по-разному влияет на вывод векторной и растровой графики.

Так, при выводе векторного рисунка используется максимальное разрешение устройства вывода. При этом команды, описывающие изображение, сообщают устройству вывода положение и размеры какого-либо объекта, а устройство для его прорисовки использует максимально возможное количество точек. Таким образом, векторный объект, например, окружность, распечатанная на принтерах разного качества, имеет на листе бумаги одинаковые положение и размеры. Однако более гладко окружность выглядит

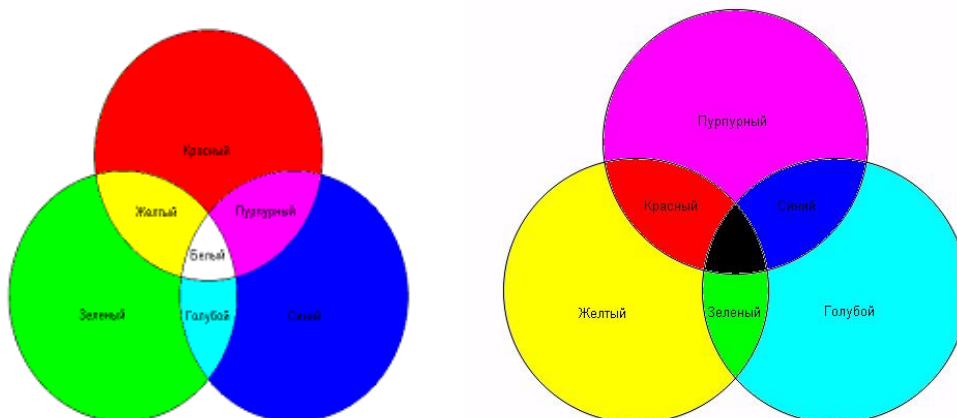
при печати на принтере с большей разрешающей способностью, так как состоит из большего количества точек принтера.

Значительно большее влияние разрешающая способность устройства вывода оказывает на вывод растрового рисунка. Если в файле растрового изображения не определено, сколько пикселей на дюйм должно создавать устройство вывода, то по умолчанию для каждого пиксела используется минимальный размер. В случае лазерного принтера минимальным элементом служит лазерная точка, в мониторе — видеопиксел. Так как устройства вывода отличаются размерами минимального элемента, который может быть ими создан, то размер растрового изображения при выводе на различных устройствах также будет неодинаков.

Цвета Некоторые предметы видимы потому, что излучают свет, а другие — потому, что его отражают. Когда предметы излучают свет, они приобретают в нашем восприятии от цвет, который видит глаз человека. Когда предметы отражают свет, то их цвет определяется цветом падающего на них света и цветом, который эти объекты отражают. Излучаемый свет выходит из активного источника, например, экрана монитора. Отраженный свет отражается от поверхности объекта, например, листа бумаги.

Существуют два метода описания цвета: система аддитивных и субтрактивных цветов.

Система аддитивных цветов работает с излучаемым светом. Аддитивный цвет получается при объединении разноцветных лучей света. В системе используются три основных цвета: красный, зеленый и синий (Red, Green, Blue — RGB). При смешивании их в разных пропорциях получается соответствующий цвет. Отсутствие этих цветов представляет в системе черный цвет. Схематично смешение цветов показано на рисунке 1.



а) Аддитивный цвет б) Субтрактивный цвет

Рисунок 7 - Система смешения цветов

В системе субтрактивных цветов происходит обратный процесс: какой-либо цвет получается вычитанием других цветов из общего луча света. При этом белый цвет получается в результате отсутствия всех цветов, а присутствие всех цветов дает черный цвет. Система субтрактивных цветов работает с отраженным цветом, например, от листа бумаги. Белая бумага отражает все цвета, -окрашенная — некоторые поглощает, остальные отражает.

В системе субтрактивных цветов основными являются голубой, пурпурный и желтый цвета (Cyan, Magenta, Yellow — CMY) — дополнительные красному, зеленому и синему. Когда эти цвета смешивают на бумаге в равной пропорции, получается черный цвет. Этот процесс проиллюстрирован на рисунке б. В связи с тем, что типографские краски не полностью поглощают свет, комбинация трех основных цветов выглядит темно-коричневой. Поэтому для корректировки тонов и получения истинно черного цвета в принтеры добавляют немного черной краски. Системы цветов, основанные на таком принципе четырехцветной печати, обозначают аббревиатурой CMYK (Cyan, Magenta, Yellow, black).

Существуют и другие системы кодирования цветов, например, представление его в виде тона, насыщенности и яркости (Hue, Saturation, Brightness — HSB).

Тон представляет собой конкретный оттенок цвета, отличный от других: красный, голубой, зеленый и т. п. Насыщенность характеризует относительную интенсивность цвета.

При уменьшении, например, насыщенности красного цвета, он делается более пастельным или блеклым. Яркость (или освещенность) цвета показывает величину черного оттенка, добавляемого к цвету, что делает его более темным. Система HSB хорошо согласуется с моделью восприятия цвета человеком. Тон является эквивалентом длины волны света, насыщенность — интенсивности волны, а яркость — общего количества света. Недостатком этой системы является необходимость преобразования ее в другие системы: RGB — при выводе изображения на монитор; CMYK — при выводе на четырехцветный принтер.

Другое обозначение системы HSB — HSL- (Hue, Saturation, Light — тон, насыщенность и освещенность).

Рассмотренные системы работают со всем спектром цветов — миллионами возможных оттенков. Однако пользователю часто достаточно не более нескольких сотен цветов. В этом случае удобно использовать индексированные палитры — наборы цветов, содержащие фиксированное количество цветов, например, 16 или 256, из которых можно выбрать необходимый цвет. Преимуществом таких палитр является то, что они занимают гораздо меньше памяти, чем полные системы RGB и CMYK.

При работе с изображением компьютер создает палитру и присваивает каждому цвету номер, затем при указании цвета отдельного пиксела или объекта просто запоминается номер, который имеет данный цвет в палитре. Для запоминания числа от 1 до 16 необходимо 4 бита памяти, а от 1 до 256 —

8 битов, поэтому изображения, имеющие 16 цветов называют 4-битовыми, а 256 цветов — 8-битовыми. При сравнении с 24 битами, необходимыми для хранения полного цвета в системе RGB, или с 32 битами — в системе CMYK, экономия памяти очевидна.

При работе с палитрой можно применять любые цвета, например, системы RGB, но ограниченное их количество. Так, при использовании 256-цветовой палитры в процессе ее создания и нумерации каждый цвет в палитре описывается как обычный 24-битовый цвет системы RGB. А при ссылке на какой-либо цвет уже указывается его номер, а не конкретные данные системы BOB, описывающие этот цвет.

Масштабирование изображений. Масштабирование заключается в изменении вертикального и горизонтального размеров изображения. Масштабирование может быть пропорциональным — в этом случае соотношение между высотой и шириной рисунка не изменяется, а меняется общий размер, и непропорциональным — в этом случае оба измерения изменяются по-разному.

Масштабирование векторных рисунков выполняется просто и без потери качества. Так как объекты векторной графики создаются по их описаниям, то для изменения масштаба векторного объекта, достаточно изменить его описание. Например, чтобы увеличить в два раза векторный объект, следует удвоить значение, описывающее его размер.

Масштабирование растровых рисунков является намного более сложным процессом, чем для векторной графики, и часто сопровождается потерей качества. При изменении размеров растрового изображения выполняется одно из следующих действий:

- одновременное изменение размеров всех пикселей (в большую или меньшую сторону);

- добавление или убавление пикселей из рисунка для отражения производимых в нем изменений, называемое выборкой пикселей в изображении.

Простейший способ изменения масштаба растрового рисунка состоит в изменении размера всех его пикселей. Так как внутри самого рисунка пиксели не имеют размера и приобретают его уже при выводе на внешнее устройство, то изменение размера пикселей раstra в сильной степени похоже на масштабирование векторных объектов — необходимо сменить только описание пиксела, а остальное выполнит устройство вывода.

Устройство вывода для создания пиксела определенного физического размера использует столько своих минимальных элементов (лазерных точек — для лазерного принтера, видеопикселей — для монитора), сколько сможет. При масштабировании изображения количество входящих в него пикселей не меняется, а изменяется количество создаваемых устройством вывода элементов, идущих на построение отдельного пиксела изображения. На рисунке 8 показан пример масштабирования растрового изображения — увеличения его в два раза по каждому измерению.

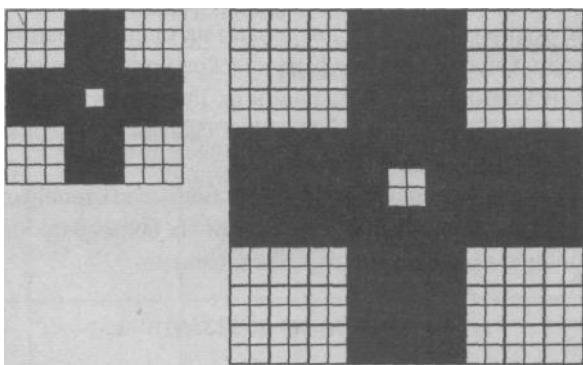


Рисунок 8 - Масштабирование растрового изображения

Выборка растрового рисунка может быть сделана двумя различными способами.

1) По первому способу просто дублируется или удаляется необходимое количество пикселей. При этом в результате масштабирования, как правило,

ухудшается качество изображения. Например, при увеличении размера рисунка возрастают его зернистость и дискретность. При уменьшении размера рисунка потери в качестве не столь заметны, однако при последующем восстановлении уменьшенного рисунка до прежнего размера опять возрастают зернистость и дискретность. Это связано с тем, что при уменьшении размера рисунка часть пикселей была удалена из исходного изображения и потеряна безвозвратно, а при последующем восстановлении размеров рисунка недостающие пиксели дублировались из соседних.

2) По второму способу с помощью определенных вычислений можно создать пиксели другого цвета, определяемого цветами первоначального пикселя и его окружения. Этот метод называется интерполяцией и является более сложным, чем простое дублирование. При интерполяции кроме дублируемых пикселей, отбираются и соседние с ними, с помощью которых вновь создаваемые пиксели получают от существующих усредненный цвет или оттенок серого. В результате переходы между пикселями становятся более плавными, что позволяет убрать или уменьшить эффект «пилообразности» изображения.

Сжатие изображений. Как и многая информация, графика может быть сжата. Это выгодно с точки зрения экономии памяти компьютера, так как, например, высококачественные изображения, как уже говорилось, имеют размеры до нескольких десятков мегабайтов. Для файлов графических изображений разработаны множество схем и алгоритмов сжатия, основными из которых являются следующие:

1. групповое сжатие;
2. кодирование методом Хаффмана;
3. сжатие по схеме LZW;
4. арифметическое сжатие;
5. сжатие с потерями,

6. преобразование цветов RGB в цвета YUV.

В основе большинства схем сжатия лежит использование одного из следующих свойств графических данных: избыточность, предсказуемость и необязательность. В частности, групповое кодирование (RLE) основано на использовании первого свойства. Кодирование по методу Хаффмана и арифметическое кодирование, основанные на статистической модели, используют предсказуемость, предлагая более короткие коды для более часто встречающихся пикселей. Алгоритмы сжатия с потерями основаны на избыточности данных.

Следует учесть, что алгоритм, обеспечивающий большую степень сжатия, обычно более сложный и поэтому требует для распаковки данных больше процессорного времени.

Рассмотрим подробнее несколько алгоритмов сжатия.

Групповое сжатие. Групповое сжатие представляет собой одну из самых простых схем сжатия файлов. Суть его заключается в том, что серия повторяющихся величин заменяется единственной величиной и ее количеством. На примере можно заметить выгоду в длине между «aabbccccdddeeeaaa» и «2a7b1c3d4e3a». Данный алгоритм прост в реализации и хорошо сжимает графические файлы с большими однотонными областями. Групповое кодирование используется во многих форматах растровых файлов, таких как TIFF, PCX и т. д.

Кодирование методом Хаффмана. Смысл метода Хаффмана заключается в замене данных более эффективными кодами. Более короткие коды используются для замены более часто появляющихся величин. Например в выражении aabbccccdeeeeeeeef есть шесть уникальных величин, с частотами появления: a:1, b:3, c:3, d:2, e:9, f:1. Для образования минимального кода используется двоичное дерево. Алгоритм объединяет в пары элементы, появляющиеся наименее часто, затем пара объединяется в один элемент, а их

частоты объединяются. Это действие повторяется до тех пор, пока элементы не объединятся в пары. В данном примере надо объединить a и f — это первая пара, a присваивается нулевая ветвь, а f — 1-я. Это означает, что 0 и 1 будут младшими битами кодов для a и f соответственно. Более старшие биты будут получены из дерева по мере его построения.

Суммирование частот дает в итоге 2. Теперь самая низкая частота — 2, поэтому пара a и f объединяется с d (которая тоже имеет частоту 2). Исходной паре присваивается нулевая ветвь, а d — ветвь 1. Таким образом, код для a заканчивается на 00; для f на 01, d заканчивается на 1 и будет на один бит короче по сравнению с кодами для a и f.

Дерево продолжает строиться подобным образом так, что наименее распространенные величины описываются более длинными кодами. Данное кодирование нуждается в точной статистике, выражающейся в том, как часто каждая величина появляется в файле. Следовательно, для работы по схеме Хаффмана необходимо два этапа: на первом этапе создается статистическая модель, на втором кодируются данные. Следует отметить, что компрессия и декомпрессия, по Хаффману, — достаточно медленный процесс.

Таблица 8 - Типы графических файлов

Название	Тип	Использование	Фирма	Расширение
BMP (Windows BitMap)	Растровый	Хранение и отображение информации в среде Windows	Microsoft	bmp

GIF	Растровый	Передача данных в сети CompuServe	CompuServe Inc	gif
Kodak Photo CD	Растровый	Для фотографической информации	Eastman Kodak	pcd
PCX	Растровый	В графических редакторах	Zsoft Corp	pcx
JPEG	Растровый	Для фотографической информации	Joint Photographic Experts Group	jpg
TIFF	Растровый	Обмен данными между настольными издательскими системами	Aldus Corp	tif
DXF	Векторный	Обмен чертежами и данными САПР	Autodesk Inc	dxf
CDR	Векторный	Чертежная издательская и другие виды графики	Corel	cdr

WMF	Векторный	Хранение и отображение информации в среде Windows	Microsoft	wmf
-----	-----------	---	-----------	-----

Рассмотрим структуру файлов изображений типа BMP и TIFF, получивших наиболее широкое распространение на практике.

Формат BMP: Windows Device Independent Bitmap. BMP (аппаратно-независимое побитовое изображение Windows) поддерживается любыми Windows -совместимыми программами. Структура файла BMP используется Windows для хранения растровых изображений. В этом формате хранятся рисунки фона, пиктограммы и другие растровые изображения Windows. Этот формат сводит к минимуму вероятность ошибок или неправильной интерпретации растровых данных.

Формат BMP правильно отображает данные независимо от аппаратных и программных устройств (монитора компьютера, видеокарты и драйвера дисплея). Такая независимость от устройств обеспечивается применением системных палитр. Однако данный формат имеет и свои недостатки. Только версии формата с 4- и 8-битовым цветом поддаются сжатию (к ним можно применить метод сжатия RLE), следовательно, 24-битовые файлы BMP будут очень большими. Кроме того, применение файлов BMP ограничено платформами Windows и OS/2.

Каждый файл BMP содержит заголовок файла, заголовок изображения, растровые данные и карту цветов (кроме изображения 24-битным цветом).

Заголовок файла BMP содержит информацию о типе и размере файла, а также о расположении в нем данных. После заголовка файла следует структура, задающая информацию о размере, цвете и сжатии изображения.

Сжатие. Поле biCompression в BITMAPINFO определяет, какой тип сжатия используется в файле. Если значение поля равно BI_RGB, то рисунок

не сжимается. Если же значение поля равно BI_RLE4 или BI_RLE8, то рисунок использует метод сжатия групповым кодированием для изображений, соответственно, с 4 бит/пиксел или 8 бит/пиксел.

Карта цветов. Значение поля biClrUsed определяет, будут ли использоваться растровым рисунком индексированные цвета. Если рисунок состоит из 1-, 4-, или 8-битовых пикселов, то должна использоваться карта цветов. Если значение поля biClrUsed равно нулю, то применяется либо полная карта, либо 24-битовый цвет. Если поле не равно нулю, то его значение равно действительному количеству цветов, которые будут доступны программе-генератору графики или драйверам устройств. Поле biClrImportant определяет, сколько цветов в индексированной карте считаются важными для изображения рисунка. Если его значение равно нулю, то важными будут все цвета.

Растровые данные. За картой цветов следуют растровые данные. Они хранятся в виде 1-, 4-, или 8-битовых индексов карты цветов или в виде буквенных 24-битовых данных системы RGB. Пикселы в растре записываются, начиная с нижнего левого угла изображения, и читаются слева направо и снизу вверх.

TIFF: Tagged Image File Format. Формат TIFF сохраняет множество данных изображения в помеченных полях, что и определило его название («Формат файла помеченного изображения»). Каждое помеченное поле хранит информацию о растровом рисунке или ссылку на другие поля. Программа, читающая файл, может пропускать неизвестные или ненужные ей поля. Эта многогранность позволяет формату находить применение в различных компьютерных системах. Кроме того, формат TIFF может сохранять разнообразные дополнительные данные о растровом рисунке, включая: кривую коррекции для изображения с оттенками серого; поля детальной информации об изображении (название программы, автора, дату создания и комментарии); размер изображения и разрешающую способность; детальную информацию о цвете оригинала.

Большинство программ, читающих файл TIFF, способны без проблем прочесть файлы этого формата, созданные на других системах. Структура формата TIFF изменчива, что позволяет ему хранить разнообразные типы изображений.

Файлы TIFF начинаются с небольшого заголовка, за которым следует указатель на первый набор помеченных полей. Помеченные поля собираются в каталоги файла изображения (КФИ), который просто содержит счетчик количества помеченных полей и указатель на первое из них. Длина помеченного поля равна 12 байтов, поэтому считывающая программа может быстро найти каждое из них.

Заголовок файла. Заголовок файла TIFF составляет 8 байтов. Первые 2 байта указывают, какой формат у файла, следующие 2 байта содержат номер версии, последние 4 байта заголовка указывают на первый КФИ. Обычно первый КФИ должен следовать сразу после заголовка, но в файле TIFF все локализуется с помощью указателей.

Каталог файла изображения. Содержит 2-байтовый счетчик общего количества помеченных полей, за которыми следуют сами помеченные поля. Двухбайтовая длина ограничивает предел каждого КФИ 256-ю помеченными полями.

Помеченные поля. Длина помеченного поля равна 12 байтов. Первые 2 байта — код метки (тип помеченного поля), следующие 2 байта содержат тип данных, на которые указывает поле. Тип данных позволяет программе узнать, сколько байтов нужно читать для каждого отдельного элемента данных. Всего пять различных типов: 1-байтовый код символа ASCII; 1-байтовое целое число; 2-байтовое целое число; 4-байтовое целое число, 8-байтовое дробное число.

Следующие 4 байта показывают, на сколько различных элементов данных указывает помеченное поле. Если данные занимают в общем 4 байта или

меньше, то элементы данных содержатся в последних 4 байтах помеченного поля. Если же данные занимают больше 4 байтов, то последние 4 байта помеченного поля используются в роли указателя на данные.

Сжатие. Файлы данного формата способны хранить пиксельные данные, сжатые с помощью разнообразных схем, например, кодированием по методу Хаффмана.