МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»

Кафедра графики, конструирования и информационных
технологий в промышленном дизайне

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «ФОРМАТЫ ГРАФИЧЕСКИХ ДАННЫХ»

для обучающихся по направлению 54.03.01 «Дизайн»,
профиль «Промышленный дизайн» всех форм обучения

Воронеж 2021

УДК 681.3(07)
ББК30.18я7
Составители: А.В. Кузовкин, А.П. Суворов, Ю.С. Золототрубова

Методические рекомендации по выполнению лабораторных работ по дисциплине «Форматы графических данных» для обучающихся по направлению 54.03.01 «Дизайн», профиль «Промышленный дизайн» всех форм обучения / ФГБО ВО «Воронежский государственный технический университет»; сост.: А.В. Кузовкин, А.П. Суворов, Ю.С. Золототрубова. – Воронеж: Изд-во ВГТУ, 2021. – 37 с.

Приводится описание выполнения лабораторных работ по курсу «Форматы графических данных» для студентов обучающихся по направлению 54.03.01 «Дизайн», профиль «Промышленный дизайн» всех форм обучения

УДК 681.3(07)
ББК30.18я7

Рецензент - г.т.н., доцент Болдырев А.А.

Рекомендовано методическим семинаром кафедры ГКПД и методической комиссией ФИТКБ Воронежского государственного технического университета в качестве методических материалов

Введение

Если заглянуть в историю, то можно проследить, как с момента появления первых ЭВМ люди стремятся разнообразить способы общения человека и машины, приблизившись к уровню общения человека с человеком. Это общение было бы гораздо более ограниченным, если бы не использовало один из наиболее простых способов — язык изображений, образов. Сегодня графические изображения на экране монитора современного персонального компьютера стали для нас нормой, совершенно неотъемлемым атрибутом интерфейса. Спектр применения компьютерной графики, помимо средства интерфейса «человек-машина», чрезвычайно широк: от создания рекламных роликов, компьютерных мультфильмов и игр, кроя одежды, малых и монументальных форм дизайна, компьютерной живописи до визуализации результатов научных изысканий [10]. Можно с уверенностью сказать, что популярность Internet, и в частности WWW, во многом объясняется широким применением графики.

Рынок программного и аппаратного обеспечения компьютерной графики – один из самых динамичных. Об этом можно судить по объему литературы и числу сервисов Internet, посвященных так или иначе компьютерной графике.

Предметом данной работы является обширная область компьютерных наук, посвященная представлению данных в памяти ЭВМ в графической форме. Это самое общее определение, так как под данными можно понимать как непосредственно хранящееся в виде файла изображение в одном из графических форматов, так и протокол обмена командами между пользователем и ЭВМ (то, что мы называем графическим интерфейсом), и битовую последовательность, сформированную для вывода на экран или печатающее устройство. Методы и способы представления и манипуляции этим видом данных относятся к компетенции компьютерной графики.

В работе рассматриваются различные способы представления изображений в памяти ЭВМ, методы и алгоритмы растеризации и обработки растровых изображений, матричные преобразования на плоскости и в пространстве, методы и алгоритмы удаления скрытых линий и поверхностей. Кроме того, приводятся основы использования

графической библиотеки OpenGL, а также описываются базовые аппаратные средства, используемые при работе с изображениями.

Программный код, приведенный в пособии, создан в MS Visual Studio 2010 на языке C#.

1. Способы представления изображений в ЭВМ

Компьютерная (машинная) графика – область деятельности, изучающая создание, способы хранения и обработки изображений с помощью ЭВМ. Под ***интерактивной*** компьютерной графикой понимают раздел компьютерной графики, изучающий вопросы динамического управления со стороны пользователя содержанием изображения, его формой, размерами и цветом на экране с помощью интерактивных устройств взаимодействия. Кроме интерактивной в компьютерной графике выделяют разделы, изучающие методы работы с изображением на плоскости, так называемую ***2D графику***, и ***трехмерную (3D) графику***.

Трехмерное изображение отличается от двухмерного, тем, что строится исходя из математического описания некоторой трехмерной сцены. Математическое описание сцены чаще всего является моделью физических объектов в трехмерном пространстве. Таким образом, для получения трехмерного изображения требуется построить математическую модель сцены и объектов на ней, а далее визуализировать путем получения проекции с учетом освещения материалов и пр. В результате визуализации мы получим изображение на плоскости экрана или на выходе принтера.

Вопросы 2D, 3D графики, общего геометрического моделирования, связанные с визуализацией геометрических моделей входят в компетенцию ***компьютерной геометрии***.

На специализацию в отдельных областях указывают названия некоторых разделов: инженерная графика, научная графика, Webграфика, компьютерная полиграфия и прочие.

Кроме этого, по способу представления изображения в памяти ЭВМ, компьютерную графику разделяют на ***векторную, растровую*** и ***фрактальную***. Рассмотрим подробнее эти способы представления изображений, выделим их основные параметры и определим их достоинства и недостатки.

1.1. Растровое представление изображений

Что такое растровое изображение?

Возьмѐм фотографию (например, см. рис. 1.1). Конечно, она тоже состоит из маленьких элементов, но будем считать, что отдельные элементы мы рассмотреть не можем. Она представляется для нас, как реальная картина природы.

Теперь наложим на изображение прямоугольную сетку. Таким образом, разобьем изображение на прямоугольные элементы. Каждый прямоугольник закрасим цветом, преобладающим в нѐм (на самом деле программы при оцифровке генерируют некий «средний» цвет, т. е. если у нас была одна чѐрная точка и одна белая, то прямоугольник будет иметь серый цвет).

Как мы видим, изображение стало состоять из конечного числа прямоугольников определѐнного цвета. Эти прямоугольники называют *pixel* (от *PIX ELemen*t) – ***пиксел*** или ***пиксель***.

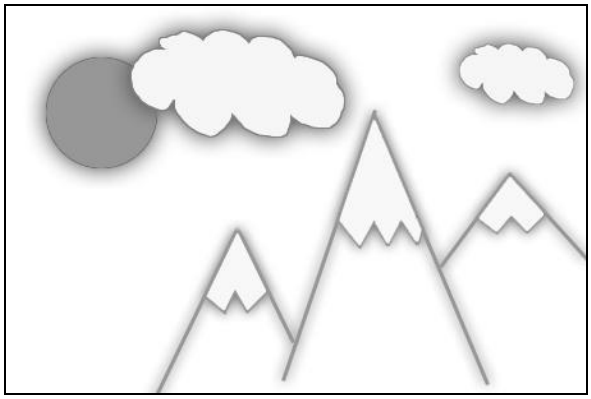

Рис. 1.1. Исходное изображение

Теперь каким-либо методом занумеруем цвета. Конкретная реализация этих методов нас пока не интересует. Для нас сейчас важно то, что каждый пиксель на рисунке стал иметь определѐнный цвет, обозначенный числом (рис. 1.2).

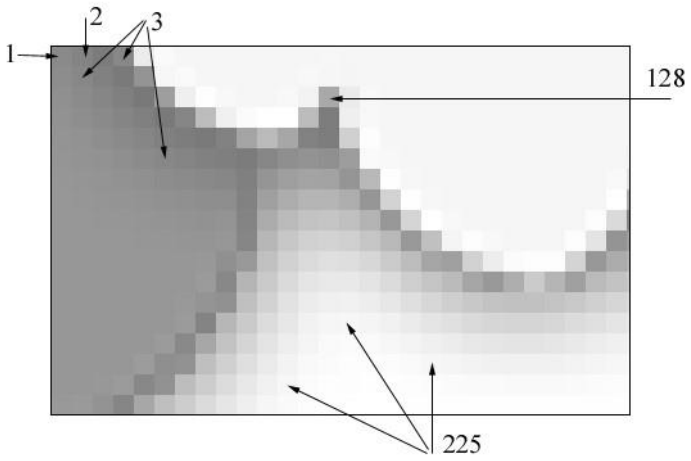

Рис. 1.2. Фрагмент оцифрованного изображения и номера цветов

Теперь пойдѐм по порядку (слева направо и сверху вниз) и будем в строчку выписывать номера цветов встречающихся пикселей. Получится строка примерно следующего вида:

1 2 8 3 212 45 67 45 127 4 78 225 34 ...

Вот эта строка и есть наши оцифрованные данные. Теперь мы можем сжать их (так как несжатые графические данные обычно имеют достаточно большой размер) и сохранить в файл.

Итак, под ***растровым*** (bitmap, raster) понимают способ представления изображения в виде совокупности отдельных точек (пикселей) различных цветов или оттенков. Это наиболее простой способ представления изображения, ибо таким образом видит наш глаз.

Достоинством такого способа является возможность получения фотореалистичного изображения высокого качества в различном цветовом диапазоне. Высокая точность и широкий цветовой диапазон требуют увеличения объема файла для хранения изображения и оперативной памяти для его обработки, что можно отнести к недостаткам. Вторым существенным недостатком является потеря качества изображения при его масштабировании.

1.1.1. Параметры растровых изображений

Как уже говорилось ранее, растровое изображение представляется в памяти ЭВМ в виде матрицы отдельных пикселей. В этой связи возникает вопрос о том, каково должно быть число этих пикселей и какое число бит отводится для хранения одного пикселя, т. е. каковы

основные параметры растрового изображения – разрешение и глубина цвета.

Разрешение (resolution) — это степень детализации изображения, число пикселей (точек), отводимых на единицу площади. Поэтому имеет смысл говорить о разрешении изображения только применительно к какому-либо устройству ввода или вывода изображения. Например, пока имеется обычная фотография на твердом носителе, нельзя сказать о ее разрешении. Но как только мы попытаемся ввести эту фотографию в компьютер через сканер, нам необходимо будет определить разрешение оригинала, т. е. указать количество точек, считываемых сканером с одного квадратного дюйма.

Поскольку изображение можно рассматривать применительно к различным устройствам, то следует различать:

- разрешение оригинала;
- разрешение экранного изображения;
- разрешение печатного изображения.

Разрешение оригинала. Разрешение оригинала определяется при вводе изображения в компьютер и измеряется в точках на дюйм (*dots per inch - dpi*). При этом количество dpi определяет не число точек в квадратном дюйме, а количество точек на одной его стороне. Например, 300 dpi означает, что в квадратный дюйм изображения покрывается растровой сеткой 300x300 и после сканирования, изображение соответствующее, квадратному дюйму будет состоять из 90 000 пикселей.

В дальнейшем разрешение оригинала влияет на разрешение изображений выводимых на разных устройствах (принтерах, экранах мониторов).

Установка разрешения оригинала зависит от требований, предъявляемых к качеству изображения и размеру файла. В общем случае действует правило: чем выше требования к качеству, тем выше должно быть разрешение оригинала.

Для получения на экране изображения близкого к размеру оригинала обычно использует разрешения 72-75 dpi. Для вывода изображения в дальнейшем на печать и распознавания текста рекомендуется устанавливать разрешении 300-600 dpi. Если исходное изображение небольшого размера и его планируется увеличить и

вывести на печать, то в этом случае разрешение оригинала лучше устанавливать 600-1200 dpi. Сканирование слайдов, негативных фотопленок и качественных материалов для полиграфии требует установки величины разрешения 1200 и более dpi.

Разрешение экранного изображения. Для экранных копий изображения элементарную точку растра принято называть пикселем (*pixel*). Для измерения разрешения экранного изображения, кроме dpi, используют единица измерения *ppi* (*pixel per inch*). Размер пикселя, а значит и разрешение экранного изображения, варьируется в зависимости от выбранного разрешения экрана (из диапазона стандартных значений), разрешения оригинала и масштаба отображения. **Разрешение печатного изображения и понятие линиатуры.** Большинство находящихся в обращении печатающих устройств, от офсетных печатающих машин до простейших струйных принтеров, используют принципы полутонового растрирования.

Полутоновое растрирование (halftoning) – это способ имитации оттенков отдельными точками краски или тонера. Этот процесс основан на том, что печатающее устройство наносит на бумагу точки краски или тонера и располагает их в узлах регулярной прямоугольной сетки, которую иногда называют ***физическим растром***. Будем называть такие точки ***печатными точками***. Соседние точки физической сетки печатающего устройства объединяются в прямоугольники, которые называются ***полутоновыми ячейками*** (halftone cells). Из полутоновых ячеек образуется еще одна сетка, именуемая ***линейным растром*** (line screen). Линейный растр – это просто способ логической организации физического растра (рис. 1.3).

Частота линейного растра или количество полутоновых ячеек на единицу длины называется ***линиатурой*** и измеряется в линиях на дюйм (line per inch, lpi).

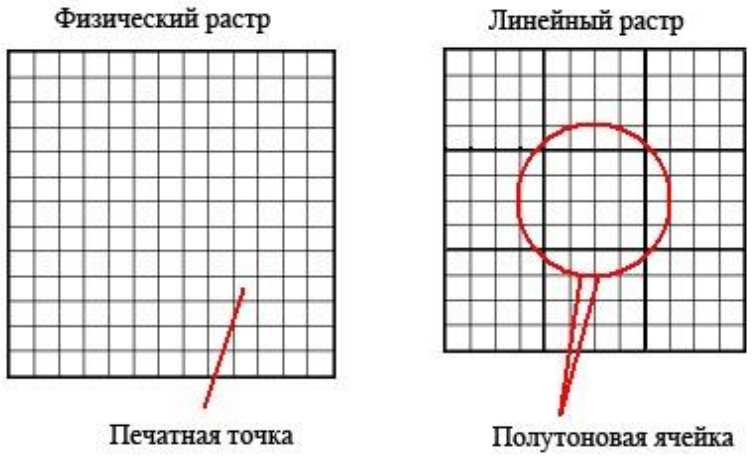

Рис. 1.3. Физический и линейный растры

При выводе на печать пиксели изображения представляются полутоновыми ячейками, а не точками физического растра печатающего устройства. Меняя заполнение полутоновых ячеек печатными точками, можно имитировать градации яркости пикселей изображения.

Рассмотрим простейшие методы растрирования оригинала в градациях серого цвета. Первым методом является метод ***растрированием с амплитудной модуляцией*** *(АМ)*, при котором иллюзия тона создается за счет формирования в центрах полутоновых ячеек, из печатных точек каких либо фигур (кругов, эллипсов, ромбов или квадратов) различного размера (рис. 1.4). Иллюзия более темного тона создается за счет увеличения радиальных размеров этих фигур и, как следствие, сокращения пробельного поля между ними при одинаковом расстоянии между центрами полутоновых ячеек.

Существует и метод ***растрирования с частотной модуляцией*** *(ЧМ)*, когда интенсивность тона регулируется изменением расстояния между соседними печатными точками одинакового размера. Таким образом, при частотно-модулированном растрировании в полутоновых ячейках с разной интенсивностью тона находится разное число печатных точек. Изображения, растрированные ЧМ-методом, выглядят более качественно, так как размер точек минимален и, во всяком случае, существенно меньше, чем средний размер фигуры при АМрастрировании. Еще более повышает качество изображения разновидность ЧМ-метода, называемая ***стохастическим растрированием***. В этом случае рассчитывается число точек,

необходимое для отображения требуемой интенсивности тона в ячейке растра. Затем эти точки располагаются внутри ячейки на расстояниях, вычисленных квазислучайным методом (на самом деле используется специальный математический алгоритм), т. е. регулярная структура растра внутри ячейки, как и на изображении в целом, вообще отсутствует. Поэтому при стохастическом ЧМ-растрировании теряет смысл понятие линиатуры растра, имеет значение лишь разрешающая способность устройства вывода. Такой способ требует больших затрат вычислительных ресурсов и высокой точности.

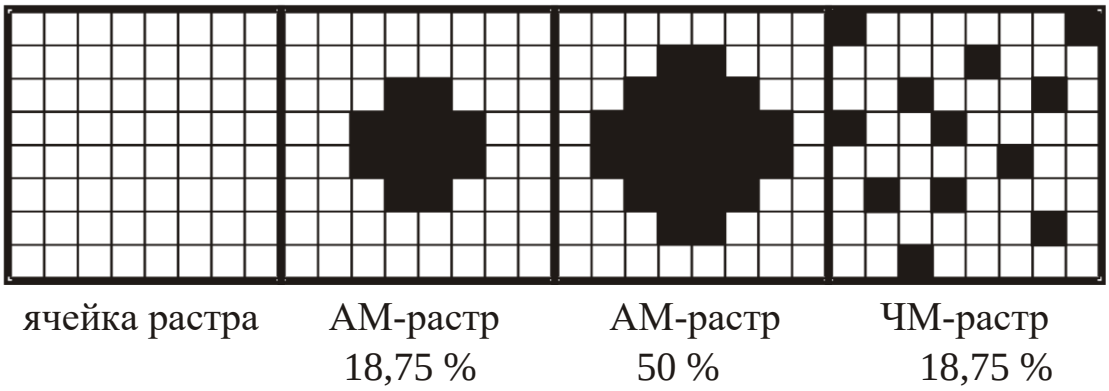

Рис. 1.4. Примеры амплитудной и частотной модуляции растра

Следующим параметром растрового изображения, который следует рассмотреть, является глубина цвета.

Глубина цвета (color depth) — это число бит, используемых для представления каждого пикселя изображения. С развитием вычислительных средств глубина цвета, хранимых в компьютере, изображений все время возрастала. Одним из первых распространенных стандартов мониторов являлся VGA, который поддерживал глубину цвета 8 бит для цветных изображений. Следующим шагом стало введение в компьютерах системы Macintosh стандарта HighColor, который кодировал цвет с глубиной 16 бит, что позволяло получить 65536 цветов. Сейчас наиболее используемым является 24-битный TrueColor, позволяющий кодировать около 16,7 млн. цветов. Однако необходимо отметить, что существуют графические системы использующие глубину цвета более чем 24 бита на пиксель.

Для лучшего понимания, что такое разрешение и глубина цвета, приведем простой пример. Вы решили отсканировать Вашу фотографию размером 10□15 см. чтобы затем обработать и распечатать на цветном принтере. Для получения приемлемого качества печати необходимо разрешение не менее 300 dpi. Считаем: 10 см = 3,9 дюйма; 15 см = 5,9 дюймов.

По вертикали: 3,9 * 300 = 1170 точек.

По горизонтали: 5,9 * 300 = 1770 точек.

Итак, число пикселей растровой матрицы 1170 * 1770 = 2 070 900.

Теперь решим, сколько цветов мы хотим использовать. Для чернобелого изображения используют обычно 256 градаций серого цвета для каждого пикселя, или 1 байт. Получаем, что для хранения нашего изображения надо 2 070 900 байт или 1,97 Мб.

Для получения качественного цветного изображения надо не менее 256 оттенков для каждого базового цвета. В модели RGB соответственно их 3: красный, зеленый и синий. Получаем общее количество байт – 3 на каждый пиксель. Соответственно, размер хранимого изображения возрастает в три раза и составляет 5,92 Мб.

Для создания макета для полиграфии фотографии сканируют с разрешением 600 dpi, следовательно, размер файла вырастает еще вчетверо.

Задание: зная, что размер экрана в пикселях 800□600, а разрешение 72 ppi, установить реальные размеры экрана в сантиметрах.

1.2. Векторное представление изображений

Для ***векторной*** графики характерно разбиение изображения на ряд графических примитивов – точки, прямые, ломаные, дуги, полигоны. Таким образом, появляется возможность хранить не все точки изображения, а координаты узлов примитивов и их свойства (цвет, связь с другими узлами и т. д.).

Вернемся к изображению на рис. 1.1. Взглянем на него по-другому. На изображении легко можно выделить множество простых объектов — отрезки прямых, ломанные, эллипс, замкнутые кривые. Представим себе, что пространство рисунка существует в некоторой координатной системе. Тогда можно описать это изображение, как совокупность

простых объектов, вышеперечисленных типов, координаты узлов которых заданы вектором относительно точки начала координат (рис. 1.5).

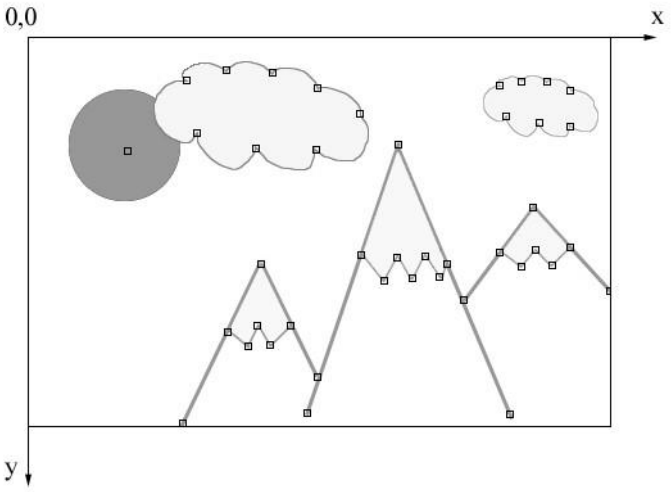

Рис. 1.5. Векторное изображение и узлы его примитивов

Проще говоря, чтобы компьютер нарисовал прямую, нужны координаты двух точек, которые связываются по кратчайшей прямой. Для дуги задается радиус и т. д. Таким образом, векторная иллюстрация – это набор геометрических примитивов.

Важной деталью является то, что объекты задаются независимо друг от друга и, следовательно, могут перекрываться между собой.

При использовании векторного представления изображение хранится в памяти как база данных описаний примитивов. Основные графические примитивы, используемые в векторных графических редакторах: точка, прямая, кривая Безье, эллипс (окружность), полигон (прямоугольник). Примитив строится вокруг его узлов (nodes). Координаты узлов задаются относительно координатной системы макета.

А изображение будет представлять из себя массив описаний – нечто типа:

```
отрезок (20,20-100,80);
окружность(50,40-30); кривая_Безье
(20,20-50,30-100,50).
```

Каждому узлу приписывается группа параметров, в зависимости от типа примитива, которые задают его геометрию относительно узла. Например, окружность задается одним узлом и одним параметром –

радиусом. Такой набор параметров, которые играют роль коэффициентов и других величин в уравнениях и аналитических соотношениях объекта данного типа, называют аналитической моделью примитива. Отрисовать примитив – значит построить его геометрическую форму по его параметрам согласно его аналитической модели.

Векторное изображение может быть легко масштабировано без потери деталей, так как это требует пересчета сравнительно небольшого числа координат узлов. Другой термин – «object-oriented graphics».

Самой простой аналогией векторного изображения может служить аппликация. Все изображение состоит из отдельных кусочков различной формы и цвета (даже части растра), «склеенных» между собой. Понятно, что таким образом трудно получить фотореалистичное изображение, так как на нем сложно выделить конечное число примитивов, однако существенными достоинствами векторного способа представления изображения, по сравнению с растровым, являются:

- векторное изображение может быть легко масштабировано без потери качества, так как это требует пересчета сравнительно небольшого числа координат узлов;
- графические файлы, в которых хранятся векторные изображения, имеют существенно меньший, по сравнению с растровыми, объем (порядка нескольких килобайт).

На самом деле размер векторного изображения зависит от количества объектов на изображении. И чем ближе качество векторного рисунка будет приближаться к фотореалистичному изображению, тем большей размер будет у файла.

Сферы применения векторной графики очень широки. В полиграфике – от создания красочных иллюстраций до работы со шрифтами. Все, что мы называем машинной графикой, 3D-графикой, графическими средствами компьютерного моделирования и САПР – все это сферы приоритета векторной графики, ибо эти ветви дерева компьютерных наук рассматривают изображение исключительно с позиции его математического представления.

Как видно, векторным можно назвать только способ описания изображения, а само изображение для нашего глаза всегда растровое. Таким образом, задачами векторного графического редактора являются

растровая прорисовка графических примитивов и предоставление пользователю сервиса по изменению параметров этих примитивов. Все изображение представляет собой базу данных примитивов и параметров макета (размеры холста, единицы измерения и т. д.). Отрисовать изображение – значит выполнить последовательно процедуры прорисовки всех его деталей.

С другой стороны, если изображение состоит из простых объектов, то для его хранения в векторном виде необходимо не более нескольких килобайт.

1.3. Представление изображений с помощью фракталов

В последние время фракталы стали очень популярны. Большую роль в этом сыграла книга франко-американского математика Бенуа Мандельброта "Фрактальная геометрия природы", изданная в 1975 году. Что же такое фрактал?

Фрактал (лат. fractus — дробленый, сломанный, разбитый) — сложная геометрическая фигура, обладающая свойством самоподобия, то есть составленная из нескольких частей, каждая из которых подобна всей фигуре целиком. Свойство самоподобия характерно для многих природных объектов. Таким свойством обладают, например, ветки деревьев, снежинки, границы облаков и морских побережий, трещины в камнях, структуры некоторых веществ, полученных с помощью электронного микроскопа и т. д. ***Фрактальная геометрия*** позволяет описать и получить изображения таких природных объектов с помощью математических средств. В компьютерной графике фракталы, могут использоваться не только для генерации изображений сложных объектов, но и для сжатия изображений.

Для классификации фракталов часто используют деление на следующие классы:

- геометрические фракталы;
- алгебраические фракталы;
- стохастические фракталы.

Существуют и другие классификации фракталов, например деление фракталов на детерминированные (алгебраические и геометрические) и недетерминированные (стохастические). Подробно рассмотрим геометрические и алгебраические фракталы.

1.3.1. Геометрические фракталы

Фракталы этого класса самые наглядные. В двухмерном случае их получают с помощью некоторой ломаной (или поверхности в трехмерном случае), называемой генератором. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяется на ломануюгенератор, в соответствующем масштабе. В результате бесконечного повторения этой процедуры, получается геометрический фрактал.

Рассмотрим один из таких фрактальных объектов - ***триадную кривую Коха***. Построение кривой начинается с отрезка единичной длины (рис. 1.6) - это 0-е поколение кривой Кох. Далее каждое звено (в нулевом поколении один отрезок) заменяется на *образующий элемент*, обозначенный на рис.1 через **n=1**. В результате такой замены получается следующее поколение кривой Кох. В 1-ом поколении - это кривая из четырех прямолинейных звеньев, каждое длиной по **1/3**. Для получения 3-го поколения проделываются те же действия - каждое звено заменяется на уменьшенный образующий элемент. Итак, для получения каждого последующего поколения, все звенья предыдущего поколения необходимо заменить уменьшенным образующим элементом. Кривая **n**-го поколения при любом конечном **n** называется ***предфракталом***. На рис. 1.6 представлены пять поколений кривой. При **n** стремящемся к бесконечности кривая Кох становится фрактальным объектом.

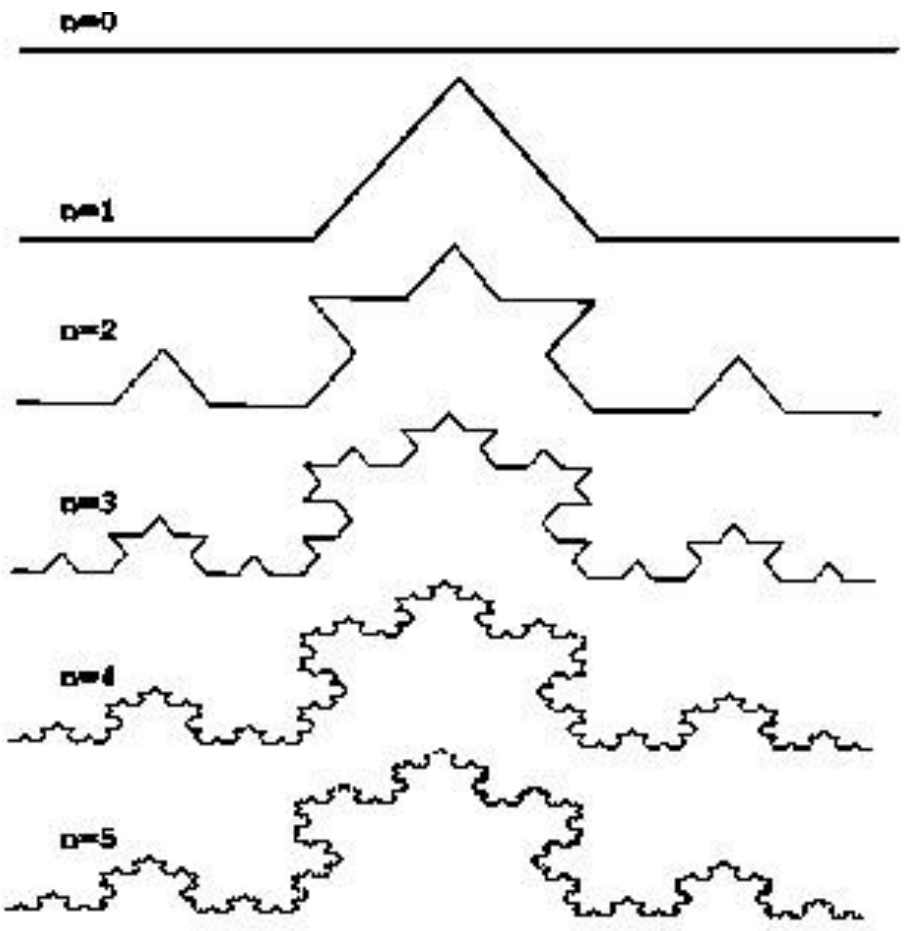

Рис. 1.6. Построение триадной кривой Коха

Три копии кривой Коха, построенные (остриями наружу) на сторонах правильного треугольника, образуют замкнутую кривую, называемую ***снежинкой Коха***.

Алгоритм построение фрактала можно описать словесно, как показано на примере построение кривой Коха, либо математически, как будет показано в следующем разделе для алгебраических фракталов. Но существует еще один способ описания алгоритма построения геометрических фракталов с помощью ***L-систем (системы Линдемайера)***. *L*-система это формальная грамматика, используемая для моделирования процессов роста и развития растений.

Изначально *L*-системы были введены при изучении формальных языков, а также использовались в биологических моделях селекции. С их помощью можно строить многие известные самоподобные фракталы, включая снежинку Коха и ковер Серпинского.

L-система определяется как кортеж $G=(V, w, P)$, где V – алфавит, представляющий набор символов, содержащих элементы которые могут быть представлены графически, w – аксиома, которая представляет собой строку символов из V, определяющая начальное состояния системы, P - представляет собой набор правил производства новой строки, путем замены символов текущего состояния системы на ряд новых. Правила грамматики *L*-системы применяются итеративно, начиная с начального состояния.

Рассмотрим *L*-систему, где алфавит состоит всего из двух символов $V=\{a,b\}$. Аксиома $w=a$. Правила производства описываются как $p1$: $a \rightarrow ab$, $p2$: $b \rightarrow ab$. Следуя итеративному принципу, каждое поколение кривой удваивается на каждом этапе: a, ab, $abab$, $abababab$. Графически a и b отображаются как два равных отрезка, соединенных под прямым углом (рис. 1.7).

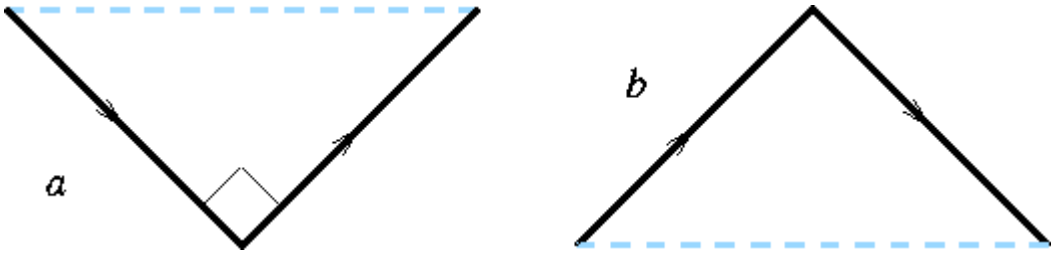

Рис. 1.7. Графическое представление алфавита *L*-системы

Графическое представление правил $p1$ и $p2$, приведено на рис. 1.8. Таким образом, правила описывают замену каждого из отрезков двумя другими, соединенных под прямым углом. При этом, каждый раз, меняется угол поворота новых отрезков относительно исходного.

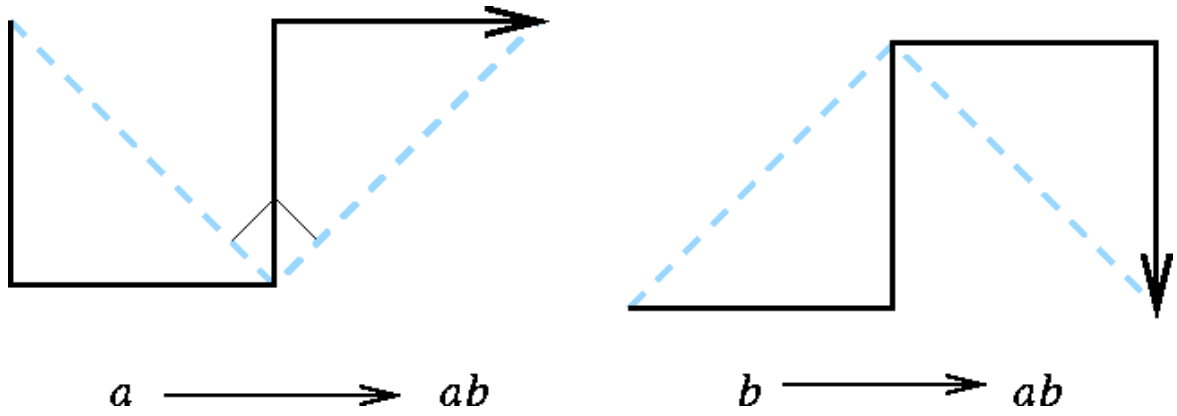

Рис. 1.8. Графическое представление правил *L*-системы

На рис. 1.9 изображены пять первых итераций процесса построения фрактала.

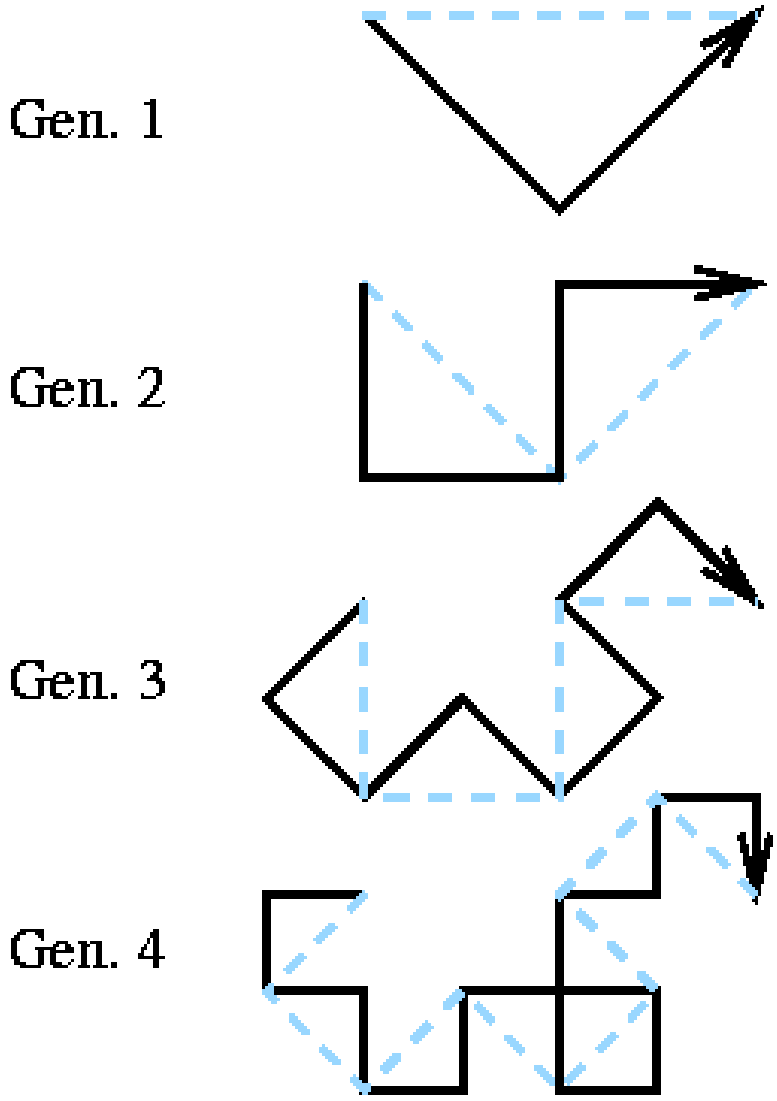

Рис. 1.9. Первые итерации построения фрактальной кривой

Такая предельная фрактальная кривая (при числе итераций стремящимся к бесконечности) называется ***драконом Хартера-Хейтуэя*** и представлена на рис. 1.10.

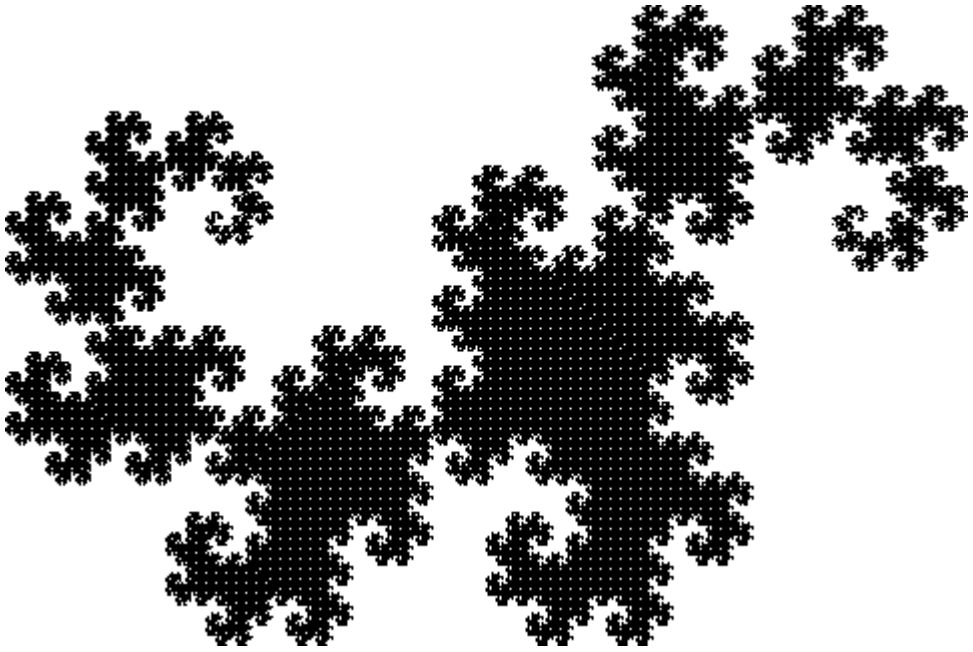

Рис. 1.10. Дракон Хартера-Хейтуэя

Некоторые из геометрических фракталов можно построить исходя из первоначально закрашенной плоской фигуры. Примером этого может служить метод построения треугольника Серпинского. Процесс построения можно описать следующим образом.

Равносторонний треугольник M_0 делится прямыми, параллельными его сторонам, на 4 равных равносторонних треугольника. Из треугольника удаляется центральный треугольник. Получается множество M_1, состоящее из 3 оставшихся треугольников "первого ранга". Поступая точно так же с каждым из треугольников первого ранга, получим множество M_2, состоящее из 9 равносторонних треугольников второго ранга. Продолжая этот процесс бесконечно, получим бесконечную последовательность M_0, M_1,…, M_n,…пересечение членов которой есть треугольник Серпинского (рис. 1.11).

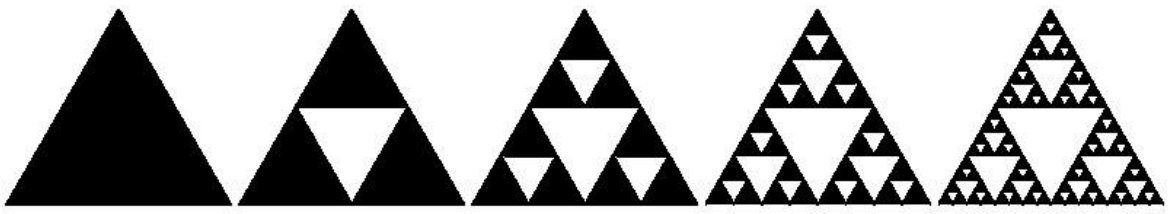

Рис. 1.11. Построение салфетки Серпинского

2. Представление цвета в компьютере

2.1. Свет и цвет

Понятия света и цвета в компьютерной графике тесно связаны и являются основополагающими.

Свет может рассматриваться либо как электромагнитная волна, либо как поток фотонов. Одной из характеристик электромагнитной волны является ее длинна λ. Видимый свет имеет длину волн в диапазоне 400-700 нм. Свет принимается либо непосредственно от источника, например, от экрана монитора, либо косвенно при отражении от поверхности объекта или преломлении в нем.

На практике мы редко сталкиваемся со светом какой то определенной длинны волны. Напротив, видимый свет практически всегда состоит из сочетания фотонов разных длин волн.

Источник или объект является ***ахроматическим***, если наблюдаемый свет содержит все видимые длины волн в приблизительно равных количествах. Ахроматический источник кажется белым, а отраженный или преломленный ахроматический свет — белым, черным или серым. Белыми выглядят объекты, ахроматически отражающие более 80% света белого источника, а черными — менее 3%.
Промежуточные значения дают различные оттенки серого.

Если воспринимаемый свет содержит длины волн в произвольных неравных количествах, то он называется ***хроматическим***.

Монохроматический - это такой свет, который имеет одну длину волны или частоту.

Понятие цвета тесно связано с тем, как человек воспринимает свет. Можно сказать, что ощущение цвета формируется человеческим мозгом в результате анализа электромагнитного излучения (света), попадающего на сетчатку глаз.

Считается, что в глазе человека существует три группы цветовых рецепторов (колбочек), каждая из которых чувствительна к определенному диапазону длин волны. Каждая группа формирует один из трех основных цветов: красный, зеленый, синий.

Если длины волн светового потока сконцентрированы у верхнего края видимого спектра (около 700 Нм), то свет воспринимается как красный. Если длины волн сконцентрированы у нижнего края видимого спектра (около 400 Нм), то свет воспринимается как синий. Если длины

волн сконцентрированы в середине видимого спектра (около 550 Нм), то свет воспринимается как зеленый.

С помощью экспериментов, построенных на этой гипотезе, были получены кривые реакции глаза, показанные на рис. 2.1.

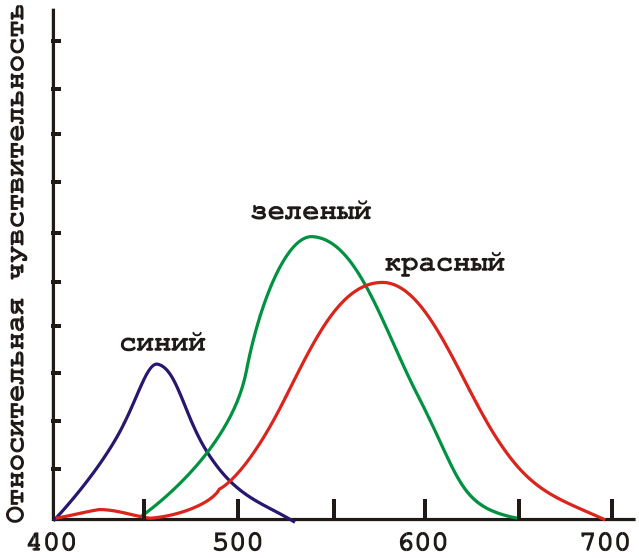

Рис. 2.1. Кривые реакции глаза

При описании цвета используют три его субъективных атрибута: ***цветовой тон***, ***насыщенность*** и ***светлоту***. Разделение признака цвета на эти взаимосвязанные компоненты есть результат мысленного процесса, существенно зависящего от навыка и обучения.

Наиболее важный атрибут цвета – ***цветовой тон (hue)*** – ассоциируется в человеческом сознании с обусловленностью окраски предмета определенным типом пигмента, краски, красителя. Тон определяется характером распределения излучения в спектре видимого света. Именно тон определяет название цвета, например «красный», «синий», «зелѐный».

Насыщенность (saturation) характеризует степень, уровень, силу выражения цветового тона. Этот атрибут в человеческом сознании связан с количеством (концентрацией) пигмента, краски, красителя. Можно сказать, что насыщенность цвета показывает, насколько данный цвет отличается от монохроматического («чистого») излучения того же светового тона. Насыщенность характеризует степень ослабления (разбавления) данного цвета белым и позволяет отличать розовый от красного, голубой от синего. Считается, что серые тона

(ахроматические) не имеют насыщенности и различаются лишь по светлоте.

Светлота (lightness, value) – это различимость участков, сильнее или слабее отражающих свет. Уровень светлоты окрашенных объектов определяется при сравнении их с ахроматическими объектами и при выявлении степени их приближения к белому цвету, отражающему максимум света.

Со светлотой тесно связано физическое понятие ***яркости света (luminance)***. Чем больше яркость, тем больше светлота. Поэтому можно сказать, что светлота есть мера ощущения яркости. С другой стороны, во многих источниках, вместо понятия светлота часто используют субъективное понятие ***яркости (brightness)***.

2.2. Цветовые модели и пространства

Как видим из вышеизложенного, описание цвета может опираться на составление любого цвета на основе основных цветов или на такие понятия, как светлота, насыщенность, цветовой тон. Применительно к компьютерной графике описание цвета также должно учитывать специфику аппаратуры для ввода/вывода изображений. В связи с необходимостью описания различных физических процессов воспроизведения цвета были разработаны различные цветовые модели. Цветовые модели позволяют с помощью математического аппарата описать определенные цветовые области спектра. Цветовые модели описывают цветовые оттенки с помощью смешивания нескольких основных цветов.

Основные цвета разбиваются на оттенки по яркости (от темного к светлому), и каждой градации яркости присваивается цифровое значение (например, самой темной – 0, самой светлой – 255). Считается, что в среднем человек способен воспринимать около 256 оттенков одного цвета. Таким образом, любой цвет можно разложить на оттенки основных цветов и обозначить его набором цифр – цветовых координат.

Таким образом, при выборе цветовой модели можно определять обычно трехмерное цветовое координатное пространство, внутри которого каждый цвет представляется точкой. Такое пространство называется пространством цветовой модели.

Профессиональные графические программы обычно позволяют оперировать с несколькими цветовыми моделями, большинство из

которых создано для специальных целей или особых типов красок: CMY, CMYK, CMYK256, RGB, HSB, HLS, L*a*b, YIQ, Grayscale (Оттенки серого) и Registration color. Некоторые из них используются редко, диапазоны других перекрываются.

3. Графические файловые форматы

Как уже говорилось ранее, при хранении растровых изображений, как правило, приходится иметь дело с файлами большого размера. В этой связи важной задачей является выбор соответствующего формата файла.

Форматов графических файлов существует великое множество и выбор приемлемого отнюдь не является тривиальной задачей. Для облегчения выбора воспользуемся классификациями. **По типу хранимой графической информации:**

- растровые (TIFF, GIF, BMP, JPEG);
- векторные (AI, CDR, FH7, DXF);
- смешанные/универсальные (EPS, PDF).

Следует учитывать, что файлы практически любого векторного формата позволяют хранить в себе и растровую графику. Однако часто это приводит к искажениям в цветопередаче, поэтому если изображение не содержит векторных объектов, то предпочтительнее использовать растровые форматы.

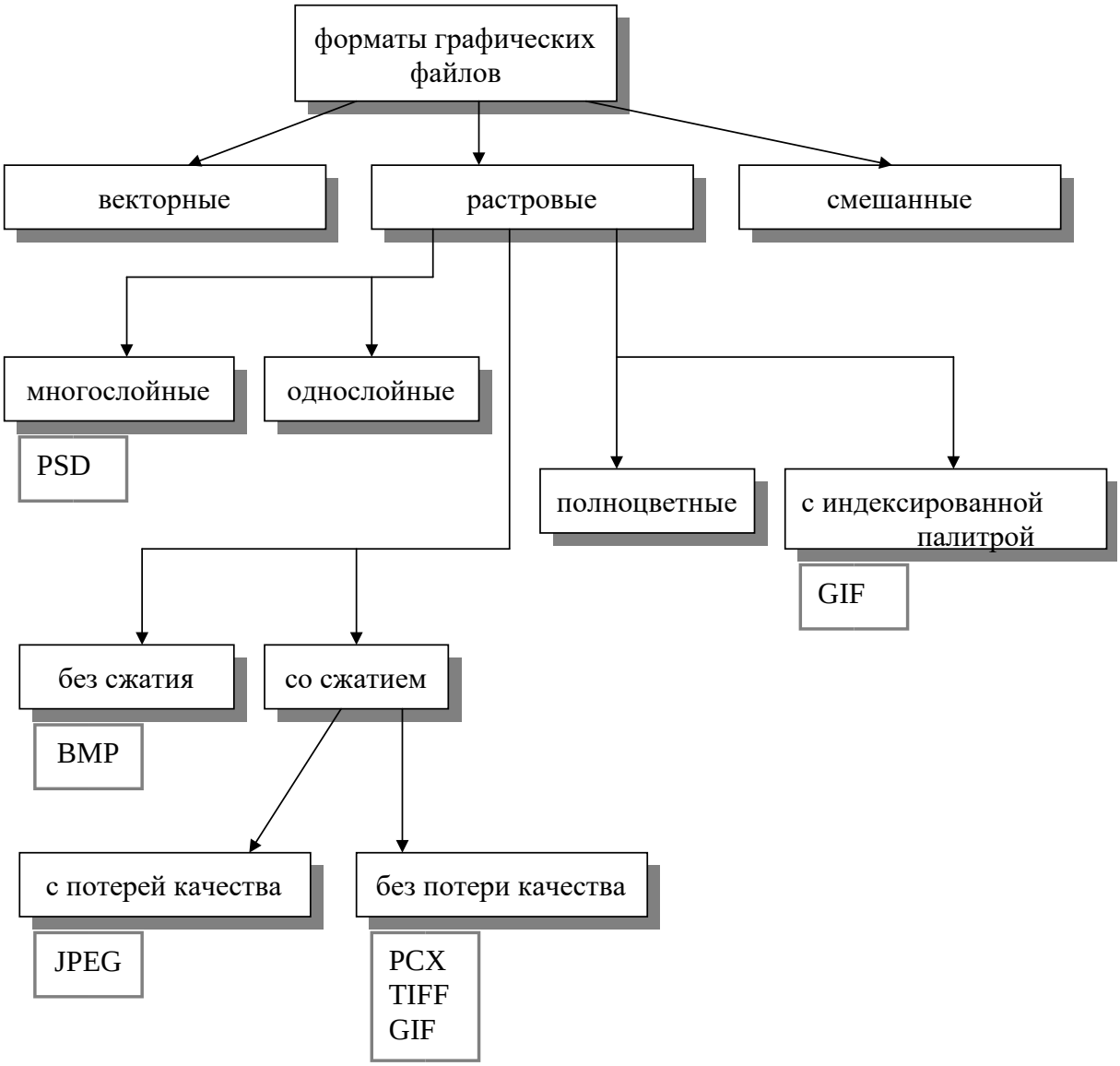

Рис. 3.1. Графические файловые форматы

Скажем несколько слов о наиболее популярных графических форматах.

3.1. BMP

BMP (от англ. BitMap Picture) — формат хранения растровых изображений. BMP был создан компанией Microsoft и широко используется в операционных системах семейства Windows.

Глубина цвета в данном формате может быть 1, 2, 4, 8, 16, 24, 32, 48 бит на пиксель, максимальные размеры изображения 65535×65535 пикселей. Однако, глубина 2 бит официально не поддерживается.

Формат BMP является примером хранения ***полноцветных изображений***. В этом случае, цвета пикселей можно определять, явно

задавая несколько параметров цвета. Например, в RGB-модели конечный цвет каждого пикселя определяется тремя слагаемыми для трех основных цветов.

В формате BMP есть поддержка сжатия по ***алгоритму RLE***. Алгоритм RLE или ***алгоритм кодирования повторов*** оперирует сериями данных, то есть последовательностями, в которых один и тот же символ встречается несколько раз подряд. При кодировании строка одинаковых символов, составляющих серию, заменяется строкой, которая содержит сам повторяющийся символ и количество его повторов.

Рассмотрим изображение, состоящее из строки белых и черных пикселей. Опишем эту строку как:

WWWWBWWWWBBBWWWWBWWWW

Здесь B представляет чѐрный пиксель, а W обозначает белый. Если мы применим RLE кодирование к этой строке, то получим следующее:

4W1B4W3B4W1B4W

Файлы формата BMP могут иметь расширения .bmp, .dib и .rle. DIB означает ***аппаратно-независимый растр*** (Device Independent Bitmap). При использовании этого формата программист может получить доступ ко всем элементам структур, описывающих изображение, при помощи обычного указателя. Но эти данные не используются для непосредственного управления экраном, так как они всегда хранятся в системной памяти, а не в специализированной видеопамяти. Формат пикселя в оперативной памяти может отличаться от того формата, который должен заноситься в видеопамять для индикации точки такого же цвета. Например, в DIB-формате может использоваться 24 бита для задания пикселя, а графический адаптер в этот момент может работать в режиме HighColor с цветовой глубиной 16 бит. При этом ярко-красная точка в аппаратно-независимом формате будет задаваться тремя байтами 0x0000ff, а в видеопамяти — словом 0xF800.

DDB означает ***аппаратно-зависимый растр*** (Device Dependent Bitmap, DDB). Этот формат всегда содержит цветовые коды, совпадающие с кодами видеобуфера, но храниться он может как в системной, так и в видеопамяти. В обоих случаях он содержит только коды цвета в том формате, который обеспечит пересылку изображения из ОЗУ в видеопамять при помощи простого копирования.

Таким образом, достоинством формата BMP является простота обращения к отдельным пикселям на изображении, что может быть использовано при написании демонстрационных программ по компьютерной графике. К недостатком нужно отнести сравнительно большие размеры файлов, хранящих изображения в формате BMP, вследствие не совершенства алгоритма RLE.

Кроме поддержки полноцветных изображений формат BMP может обеспечивать хранения изображений с использованием ***индексированной палитры***. Второй подход заключается в том, что в первой части файла, хранящего изображение, хранится ***«палитра»***, в которой с помощью одной из цветовых моделей кодируются цвета, присутствующие на изображении. А вторая часть, которая непосредственно описывает пиксели изображения, фактически состоит из индексов в палитре.

Формат BMP может использовать режим индексирования цветов при следующих значениях глубины цвета: 1 бит (2 цвета), 2 бита (4 цвета), 4 бита (16 цветов), 8 бит (256 цветов).

Благодаря использованию палитры имеется возможность адаптировать изображение к цветам, присутствующим на изображении. В таком случае изображение ограниченно не заданными цветами, а максимальным количеством одновременно используемых цветов.

Достоинством палитры является возможность существенно сократить размер файла с изображением. Недостатком является возможность потери цветов при ограниченном размере палитры.

3.2. TIFF

TIFF (англ. Tagged Image File Format) — формат хранения растровых графических изображений. Изначально был разработан компанией Aldus в сотрудничестве с Microsoft. TIFF был выбран в качестве основного графического формата операционной системы Mac OS X. В настоящее время авторские права на спецификации формата принадлежат компании Adobe.

Принцип хранения данных в TIFF основан на использовании специальных маркеров (тэгов) в сочетании с битовыми последовательностями кусков растра.

Формат TIFF поддерживает большую глубину цвета: 8, 16, 32 и 64 бит на канал при целочисленном кодировании, а также 32 и 64 бит на

канал при представлении значения пикселя числами с плавающей запятой.

Структура формата гибкая и позволяет сохранять изображения в режиме цветов с палитрой, а также в различных цветовых пространствах: бинарном (двуцветном), полутоновом, с индексированной палитрой, RGB, CMYK, YCbCr, CIE Lab. Помимо традиционных цветов CMY формат поддерживает цветоделение с большим числом красок, в частности систему Hexachrome компании Pantone. В систему Hexachrome, известную как CMYKOG, добавлены оранжевые и зеленые краски для лучшего представления цветов при печати.

Помимо прочих достоинств формат TIFF позволяет сохранять растровые изображения с компрессией без потери качества. Этот формат поддерживает сжатие без потери качества по ***алгоритму LZWкомпрессии***.

Алгоритм Лемпеля — Зива — Велча (Lempel-Ziv-Welch, LZW) — это универсальный алгоритм сжатия данных без потерь данных. Алгоритм на удивление прост. Если в двух словах, то LZW-сжатие заменяет строки символов некоторыми кодами. Это делается без какоголибо анализа входного текста. Вместо этого при добавлении каждой новой строки символов просматривается уже существующая таблица строк. Сжатие происходит, когда код заменяет строку символов. Коды, генерируемые LZW-алгоритмом, могут быть любой длины, но они должны содержать больше бит, чем единичный символ. Первые 256 кодов (когда используются 8-битные символы) по умолчанию соответствуют стандартному набору символов. Остальные коды соответствуют обрабатываемым алгоритмом строкам. Простой метод может работать с 12-битными кодами. Значения кодов 0 - 255 соответствуют отдельным байтам, а коды 256 - 4095 соответствуют подстрокам.

Приведем пример LZW кодирования применительно к изображению. Так, если в изображении имеются наборы из розового, оранжевого и зелѐного пикселей, повторяющиеся 50 раз, LZW выявляет это, присваивает данному набору отдельное число (например, 7) и затем сохраняет эти данные 50 раз в виде числа 7. Метод LZW, так же, как и RLE, лучше действует на участках однородных, свободных от шума

цветов, он действует гораздо лучше, чем RLE, при сжатии произвольных графических данных, но процесс кодирования и распаковки происходит медленнее.

Кроме LZW-компрессии формат TIFF поддерживает следующие алгоритмы сжатия:

- RLE;
- LZ77;
- ZIP;
- JBIG;
- JPEG;
- CCITT Group 3, CCITT Group 4.

При этом JPEG является просто инкапсуляцией формата JPEG в формат TIFF. Формат TIFF позволяет хранить изображения, сжатые по стандарту JPEG, без потерь данных (JPEG-LS).

3.3. GIF

Первая версия формата GIF (Graphics Interchange Format, «Формат для обмена графической информацией») была разработана в 1987 г. специалистами компьютерной сети CompuServe. Этот формат сочетает в себе редкий набор достоинств, неоценимых при той роли, которую он играет в WWW. Сам по себе формат содержит уже достаточно хорошо упакованные графические данные.

Формат GIF использует для хранения изображений индексированную палитру, ограничивающую количество цветов 256 значениями. Размер палитры может быть и меньше. Если в изображении используется, скажем, 64 цвета (2^6), то для хранения каждого пикселя будет использовано ровно шесть бит и ни битом больше.

Поскольку GIF использует для сжатия LZW алгоритм, то степень сжатия графической информации сильно зависит от уровня ее повторяемости и предсказуемости, а иногда еще и от ориентации картинки. Так как LZW метод сканирует изображение по строкам, то, к примеру, плавный переход цветов (градиент), направленный сверху вниз, сожмется куда лучше, чем тех же размеров градиент, ориентированный слева направо, а последний – лучше, чем градиент по диагонали.

Изменив порядок следования данных в файле, создатели GIFa заставили картинку рисоваться не только сверху вниз, но и, если можно

так выразиться, «с глубины к поверхности», – то есть становиться все четче и детальнее по мере подхода из сети новых данных.

Для этого файл с изображением тасуется при записи так, чтобы сначала шли все строки пикселей с номерами, кратными восьми (первый проход), затем четырем (второй проход), потом двум и, наконец, последний проход – все оставшиеся строки с нечетными номерами. Во время приема и декодирования такого файла каждый следующий проход заполняет «дыры» в предыдущих, постепенно приближая изображение к исходному состоянию. Поэтому такие изображения были названы ***чересстрочными (interlaced)***.

Другой полезной возможностью формата является использование прозрачности и поддержка анимационных изображений. Для создания анимации используется нескольких статичных кадров, а также информация о том, сколько времени каждый кадр должен быть показан на экране.

3.4. PNG

PNG (portable network graphics) - растровый формат хранения графической информации, использующий сжатие без потерь по алгоритму Deflate.

PNG поддерживает три основных типа растровых изображений:

- Полутоновое изображение (с глубиной цвета 16 бит);
- Цветное индексированное изображение (палитра 8 бит для цвета глубиной 24 бит);
- Полноцветное изображение (с глубиной цвета 48 бит).

Формат PNG спроектирован для замены устаревшего и более простого формата GIF, а также, в некоторой степени, для замены значительно более сложного формата TIFF. Алгоритм сжатия Deflate является свободным в отличие от защищенного патентами и соответственно платного LZW, используемого в GIF. В отличии от LZW кодирования, которое позволяет эффективно сжимать горизонтальные одноцветные области, с Deflate сжатием можно забыть про эти ограничения.

В формате GIF один из цветов в палитре может быть объявлен «прозрачным». В этом случае в программах, которые поддерживают прозрачность GIF (например, большинство современных браузеров) сквозь пиксели, окрашенные «прозрачным» цветом будет виден фон.

Однако создатели PNG пошли еще дальше, разработав технологию ***альфа-канала***. Альфа-канал позволяет добиться эффекта частичной прозрачности пикселей. Также, в PNG поддерживается гаммакоррекция. ***Гамма-коррекция*** - коррекция функции яркости в зависимости от характеристик устройства вывода. Повышение показателя гамма-коррекции позволяет повысить контрастность, разборчивость тѐмных участков изображения, не делая при этом чрезмерно контрастными или яркими светлые детали снимка.

Кроме этого PNG формат имеет следующие преимущества перед GIF:

- практически неограниченное количество цветов в изображении (GIF использует в лучшем случае 8-битный цвет);
- двумерная чересстрочная развѐртка;
- возможность расширения формата пользовательскими блоками.

К недостатком формата, можно отнести, что PNG изначально был предназначен лишь для хранения одного изображения в одном файле. Поэтому данный формат не поддерживает анимацию. Для решения этой проблемы создана модификация формата APNG.

3.5. JPEG

JPEG (Joint Photographic Experts Group, по названию организациираразработчика) - один из популярных графических форматов, основанный на алгоритме сжатия JPEG и применяемый для хранения фотоизображений и подобных им изображений.

Алгоритм JPEG разработан группой экспертов из Международной организации по стандартизации (ISO) специально для сжатия полноцветных 24-битовых изображений.

Процесс сжатия по схеме JPEG состоит из нескольких шагов. На первом шаге производится преобразование изображения из цветовой модели RGB в модель YUV, основанной на характеристиках яркости и цветности.

В модели YUV, Y-компонента – светлота (яркость). Компоненты U и V содержат информацию о цвете.

На следующем после преобразования шаге изображение разделяется на квадратные участки размером 8x8 пикселей. После этого над каждым участком производится дискретное косинус-

преобразование (ДКП). При этом выполняется анализ каждого блока, разложение его на составляющие цвета и подсчет частоты появления каждого цвета.

Если говорить научным языком, то JPG использует для сохранения ряды Фурье и при больших степенях сжатия просто отбрасывает члены ряда высшего порядка.

Можно сказать, что JPEG хранит скорость изменения цвета от пикселя к пикселю. Лишнюю с его точки зрения цветовую информацию он отбрасывает, усредняя некоторые значения. Чем выше уровень компрессии, тем больше данных отбрасывается и тем ниже качество

На следующем этапе изображение представляется строками чисел, которые можно сжимать дальше. Поскольку в результате предыдущей обработки строки содержат много нулей, последняя стадия кодирования выполняется по алгоритму Хаффмана, что дает хорошие результаты и позволяет получить файл в 10–500 раз меньше, чем BMP.

В дополнении к стандарту JPEG, ориентированным прежде всего на сжатие с потерями, был разработан стандарт на сжатие без потерь — JPEG-LS (в котором, однако, предусмотрен также режим сжатия с ограниченными потерями).

Алгоритм сжатия, лежащий в основе JPEG-LS, использует адаптивное предсказание значения текущего пикселя по окружению, включающему уже закодированные пиксели.

К недостаткам сжатия по стандарту JPEG следует отнести появление на восстановленных изображениях при высоких степенях сжатия характерных артефактов (заметных искажений изображения).

3.6. PDF

Формат PDF (Portable Document Format) предложен фирмой Adobe как независимый от платформы формат, в котором могут быть сохранены и иллюстрации (векторные и растровые), и текст, причем со множеством шрифтов и гипертекстовых ссылок. Для достижения продекларированной в названии переносимости размер PDF-файла должен быть малым. Для этого используется компрессия (для каждого вида объектов применяется свой способ). Например, растровые изображения записываются в формате JPEG.

Для работы с этим форматом компания Adobe выпустила пакет Acrobat. Бесплатная утилита Acrobat Reader позволяет читать документы и распечатывать их на принтере, но не дает возможности создавать или изменять их. Acrobat Distiller переводит в этот формат PostScript-файлы. Многие программы (Adobe PageMaker, CorelDraw, FreeHand) позволяют экспортировать свои документы в PDF, а некоторые – еще и редактировать графику, записанную в этом формате. Обычно в этом формате хранят документы, предназначенные только для чтения, но не для редактирования. Файл в формате PDF содержит все необходимые шрифты. Это удобно и позволяет не передавать шрифты для вывода (передача шрифтов не вполне законна с точки зрения авторского права).

PostScript

Это язык описания страниц, предназначеный для формирования изображений произвольной сложности и вывода их на печать. Для этого в языке имеется широкий набор графических операторов, используемых в произвольной комбинации. Все графические операторы языка, формирующие изображение, можно разделить на три группы. Это:

- **векторная графика**, позволяющая рисовать прямые линии, дуги, кривые произвольного размера, ориентации, ширины, закрашивать площади любого размера, формы, цвета; цвет для линий или заливок может задаваться в любом из цветовых пространств языка; любой описанный на языке контур может быть границей клипирования изображения; контур клипирования задаѐт границы рисуемого изображения;
- **работа с текстом** – для вывода текста произвольного размера в различных гарнитурах, размещая его с произвольной ориентацией в произвольном месте страницы; текст полностью интегрирован с графикой – все текстовые символы трактуются как графические фигуры и могут обрабатываться любым из графических операторов;
- **растровые изображения** позволяют выводить на листе сканированные рисунки или фотографии с масштабированием и ориентацией, источником растра может быть как текущий файл, содержащий программу на PostScripte, так и внешний; считывание цветовых слоев может вестись как из одного файла, так и из нескольких сепарированных. Изображение, описываемое на языке

PostScript, никак не зависит от разрешающей способности выходного устройства и его цветовой глубины (числа цветов). Приближение к конкретным разрешающим возможностям выходного устройства – это процесс, не связанный с описанием изображения на языке PostScript, и выполняется для каждого выходного устройства по-своему. В этом и заключается устройство-независимость языка PostScript. Качество изображения определяется конкретным выходным устройством, его физическими ограничениями.

Формирование изображения на выходном устройстве является двухступенчатым процессом:

1. Приложение генерирует устройство независимое изображение на языке PostScript.
2. Система обработки изображения (интерпретатор) интерпретирует изображение (программу) и приближает его к характеристикам конкретного выходного устройства.

Литература

1. Божко А. Н. Компьютерная графика : [учебное пособие для вузов] / А. Н. Божко, Д. М. Жук, В. Б. Маничев. – М: Изд-во МГТУ им. Н. Э. Баумана, 2007. – 392 с.
2. Гринченко В.Т., Мацыпура В.Т., Снарский А.А. Введение в нелинейную динамику. Хаос и фракталы. - 2-е изд. Издательство: ЛКИ, 2007 г. — 264 с.
3. Дегтярев В. М. Компьютерная геометрия и графика. – М: Академия 2010 г. 192 с.
4. Краснов М. В. OpenGL. Графика в проектах Delphi. — СПб.: БХВПетербург, 2001. — 352 с.
5. М. Домасев, С. Гнатюк. Цвет. Управление цветом, цветовые расчеты и измерения. – СПб: Питер 2009 г. 224 с.
6. Никулин Е. А. Компьютерная геометрия и алгоритмы машинной графики. — СПб: БХВ-Петербург, 2003. — 560 с.
7. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ. - М.: Мир, 1989. – 512 с.
8. Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ. – М.: Мир, 2001. – 604 с.
9. Тихомиров Ю. Программирование трехмерной графики. – СПб: BHV – Санкт-Петербург, 1998. – 256 с.
10. Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики: В 2-х кн., Кн. 1. / Пер. с англ. – М.: Мир, 1985 – 368 с.
11. Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики: В 2-х кн., Кн. 2. / Пер. с англ. – М.: Мир, 1985 – 368 с.
12. Шикин Е. В., Боресков А. В. Компьютерная графика. Полигональные модели. – М.: ДИАЛОГ-МИФИ, 2000. – 464 с.

Оглавление

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «ФОРМАТЫ ГРАФИЧЕСКИХ ДАННЫХ»

для обучающихся по направлению 54.03.01 «Дизайн», профиль «Промышленный дизайн» всех форм обучения

Составители:
Кузовкин Алексей Викторович
Суворов Александр Петрович
Золототрубова Юлия Сергеевна

Подписано в печать 04.09.2021
Формат 60x84 1/8 Бумага для множительных аппаратов Уч.-изд. л. 3,3 Усл. печ. л. 3,0.

ФГБОУ ВО «Воронежский государственный технический университет»
396026 Воронеж, Московский просп., 14

Участок оперативной полиграфии издательства ВГТУ
396026 Воронеж, Московский просп., 14