

# 1. Введение



Компьютерная графика – использование вычислительной техники для создания графических изображений, их отображения различными средствами и манипулирования ими.

Как следует из определения – компьютерным (цифровым) может быть названо изображение, созданное с помощью компьютерной программы.

Первоначально программисты научились получать рисунки в режиме символьной печати. На бумажных листах с помощью символов (звёздочек, точек, крестиков, букв и др.) получались рисунки, напоминающие мозаику. Так печатались графики функций, изображения течений жидкостей и газов, изображения электрических и магнитных полей т.д. С помощью символьной печати программисты умудрялись получать даже художественные изображения. В редком компьютерном центре стены не украшались распечатками с портретами Эйнштейна, репродукциями Джоконды и другой «машинной живописью».

Затем появились специальные устройства для графического вывода на бумагу – графопостроители (плоттеры). С помощью такого устройства на лист бумаги чернильным пером наносятся графические изображения: графики, диаграммы, технические чертежи и прочее.

Но настоящая революция в компьютерной графике произошла с появлением графических дисплеев. На экране графического дисплея стало возможным получать рисунки, чертежи в таком же виде, как на бумаге с помощью карандашей, красок, чертёжных инструментов.

Связь традиционной и компьютерной графики, с одной стороны, определяет

применение множительной техники. С другой стороны, можно найти еще одно объяснение возникновению термина «графика» применительно к работе художника-компьютерщика. Слово графика означает изображение линиями, штрихами, точками. А все графические компьютерные программы принципиально разделяются на два типа: векторные (изображение строится линиями) и растровые (изображение пятном из точек), то есть, каким бы сложным ни казалось изображение, созданное в компьютере, по своей сути, любое из них относится к графике.

Изменилась именно техника, и с появлением компьютера произошла настоящая революция в производстве изобразительной продукции.

## **2. Виды компьютерной графики**

Несмотря на то, что для работы с компьютерной графикой существует множество классов программного обеспечения, различают всего три вида компьютерной графики. Это растровая графика, векторная графика и фрактальная графика. Они отличаются принципами формирования изображения при отображении на экране монитора или при печати на бумаге.

Правомерна и другая классификация.

Двухмерная графика – изображение, имеющее два измерения, то есть лежащее на плоскости. Основа компьютерной графики, в том числе и трехмерной.

Трехмерная (3D) графика – построение на компьютере, с помощью специальных программ, пространственной модели, состоящей из простых и сложных геометрических форм, присвоение этой модели фактуры, цвета, степени прозрачности и матовости, придание ей и условной камере движения в виртуальном пространстве, расстановка в этом пространстве источников

света и, наконец, просчет выстроенной сцены. Применяется при создании компьютерных игр, рекламы и т.д.

### **3. Области применения компьютерной графики**

Современное применение компьютерной графики очень разнообразно. Рассмотрим основные области применения компьютерной графики.

Научная графика – это направление появилось самым первым. Назначение - визуализация (наглядное изображение) объектов научных исследований, графическая обработка результатов расчётов, проведение вычислительных экспериментов с наглядным представлением их результатов.

Деловая графика – область компьютерной графики, предназначенная для создания иллюстраций, часто используемых в работе различных учреждений. Плановые показатели, отчётная документация, статические сводки - вот объекты, для которых с помощью деловой графики создаются иллюстративные материалы. Чаще всего это графики, круговые и столбчатые диаграммы.

Конструкторская графика – используется в работе инженеров-конструкторов. Этот вид компьютерной графики является обязательным элементом систем автоматизации проектирования (САПР). Графика в САПР используется для подготовки технических чертежей проектируемых устройств. Графика в сочетании с расчётами позволяет проводить в наглядной форме поиск оптимальной конструкции, наиболее удачной компоновки деталей, прогнозировать последствия, к которым могут привести изменения в конструкции. Средствами конструкторской графики можно получать плоские изображения (проекции, сечения) и пространственные, трёхмерные изображения.

Полиграфия – совокупность технических средств для множественного репродуцирования текстового материала и графических изображений. Специалист, работающий в этой области, должен не только знать программы верстки и графические редакторы, но и разбираться в допечатной подготовке издания.

Web-дизайн – оформление web-страниц. Он играет такую же роль для сайта, как полиграфический дизайн и верстка для бумажного издания. Часто под web-дизайном понимают не только создание графических элементов для сайта, но и проектирование его структуры, навигации, то есть создание сайта целиком.

Мультимедиа – область компьютерной графики, связанная с созданием интерактивных приложений (дающих возможность активно влиять на содержимое и направленность), энциклопедий, справочных систем, обучающих программ и интерфейсов к ним.

#### **4. Настольные издательские системы**

Понятие настольная издательская система (Desktop Publishing) включает в себя все технические и программные аспекты компьютерной графики.

Можно выделить, как минимум, три основных уровня, наличие которых обеспечивает надежное функционирование системы.

- Аппаратный уровень (hardware level) представляет собой совокупность материальных элементов – устройств, с помощью которых происходят ввод, обработка, хранение, передача и вывод информации.
- Программный уровень (software level) – это совокупность информационных элементов (программ и их команд), с помощью которых осуществляется управление как собственно текстовой и

изобразительной информацией, так и аппаратным оборудованием.

- Пользовательский уровень (brainware level) представляет собой совокупность творческих индивидуумов, высококлассных специалистов и обычных пользователей, которые интегрируют свой творческий потенциал, а также аппаратный и программный уровни для создания творческих произведений.

#### **4.1. Аппаратный уровень**

В свою очередь состоит из следующих компонентов:

- устройства ввода информации (input devices) – обеспечивают преобразование любых видов информации на разнообразных носителях в цифровую форму, что создает условия для ее дальнейшей компьютерной обработки;
- устройства обработки, хранения и передачи информации (process, storage and transfer devices) – ядро аппаратного уровня;
- устройства вывода информации (output devices) – устройства, обеспечивающие перевод цифровой информации в форму, понятную и доступную человеку.

##### *Устройства ввода информации*

Главным признаком устройств ввода является преобразование одного вида информации (изображений на «твердом носителе») в иной вид, в нашем случае в цифровую форму.

К таким устройствам относятся:

- клавиатура (keyboard);
- мышь (mouse);

- трекбол (trackball);
- сканер (scanner);
- цифровая камера (digital camera);
- графический планшет (graphics tablet);
- дигитайзер (digitizer);
- и т.д.

### *Устройства обработки, хранения и передачи информации*

В большинстве случаев речь идет об аппаратном уровне компьютера, который представляет собой многоуровневую структуру.

Устройство обработки – процессор, видеопроцессор.

Устройства хранения информации:

- оперативное запоминающее устройство (RAM);
- память видеокарты;
- магнитные носители (FDD, HDD);
- оптические носители (CD, DVD и т.д.);
- магнитооптические носители (MO, MOD Drive);
- сменные диски и носители (flash-накопители);
- ленточные накопители (стример, mini DV);
- и т.д.

Устройства передачи – порты и другие компоненты.

### *Устройства вывода информации*

Устройства вывода выполняют функцию, обратную вводу информации, и

обеспечивают преобразование цифровой информации в понятный человеку вид – визуальные образы.

В зависимости от способа визуализации можно выделить два основных класса:

- средства электронной визуализации (мониторы, проекторы);
- средства физического вывода (струйные, лазерные и фотопринтеры, графопостроители, фотонаборные автоматы).

#### **4.2. Программный уровень**

Сам по себе компьютер не выполняет никаких действий. Любая последовательность действий определяется алгоритмами, заложенными в программах.

Можно выделить следующие основные классы программного обеспечения:

- редакторы растровой (пиксельной) графики;
- редакторы векторной графики;
- редакторы трехмерной графики;
- программы фрактальной графики;
- программы верстки;
- вспомогательные программы (операционная система, утилиты, plug-in, viewer, конверторы, браузеры, архиваторы и многие другие).

#### **4.3. Пользовательский уровень**

Эстетическая и художественная составляющие не являются частью технических систем. Человек, сидящий за компьютером, и есть пользовательский уровень.

## 5. Система координат и типы преобразования графической информации

### 5.1. Декартова система координат

Для того чтобы уметь синтезировать изображения на экране компьютера, необходимо предложить способ математического описания объектов в трехмерном пространстве или на плоскости. Окружающий нас мир с точки зрения практических приложений описывают как трехмерное евклидово пространство. Под описанием трехмерного объекта будем понимать знание о положении каждой точки объекта в пространстве в любой момент времени. Положение точек в пространстве удобно описывается с помощью декартовой системы координат.

Для того чтобы ввести декартову систему координат, проведем три направленные прямые линии, не лежащие в одной плоскости, которые называются осями, в трехмерном пространстве так, чтобы они пересекались в одной точке – начале координат. Выберем на этих осях единицу измерения. Тогда положение любой точки в пространстве будем описывать через координаты этой точки, которые представляют собой расстояния от начала координат до проекций точки на соответствующие оси координат. Проекцией точки на координатную ось называется точка пересечения плоскости, проходящей через заданную точку, и параллельной плоскости, образованной двумя другими осями координат. Например, на рис. 5.1 проекцией точки  $P$  на ось  $Ox$  является точка  $Q$ , которая принадлежит плоскости, параллельной плоскости  $zOy$ .

В общем случае оси системы координат могут располагаться под произвольными углами друг относительно друга. Для практических расчетов гораздо удобнее, когда эти оси расположены взаимно перпенди-



кулярно. Такая система координат называется ортогональной.

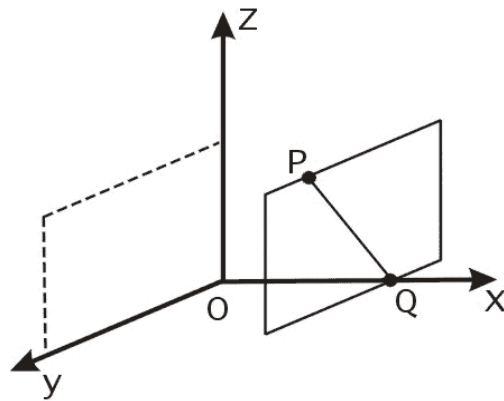


Рис. 5.1. Нахождение координаты  $x=Q$  точки  $P$

Таким образом, положение в пространстве точки  $P$  описывается ее координатами, что записывается как  $P=(x,y,z)$ . Взаимное расположение осей в ортогональной системе координат в трехмерном пространстве может быть двух видов. Проведем ось  $Ox$  слева направо, а ось  $Oy$  снизу вверх, как показано на рис. 5.2.

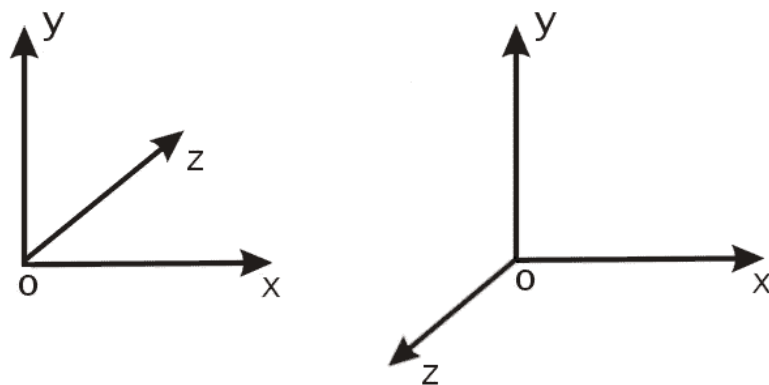


Рис. 5.2. Левосторонняя и правосторонняя системы координат

Ось  $Oz$  при этом может проходить как в направлении от наблюдателя в плоскость листа, так и от плоскости листа к наблюдателю. В первом случае система координат будет называться левой или левосторонней, а во втором случае – правой или правосторонней.

Декартовы координаты точек позволяют описывать статичное положение объектов в пространстве. Однако для проведения каких-либо действий над объектами необходимо иметь дополнительные математические конструкции. В качестве одной из таких конструкций применяют радиус-векторы.

Радиус-векторы обладают всеми свойствами векторов, но имеют одну особенность: начало радиус-вектора находится всегда в начале координат, а конец радиус-вектора лежит в некоторой точке пространства. Это свойство радиус-векторов позволяет поставить во взаимно однозначное соответствие всем точкам пространства соответствующие им радиус-векторы.

Рассмотрим теперь каким образом можно использовать координаты точек и радиус-векторы для описания прямых и плоскостей в трехмерном пространстве.

Под описанием прямой понимаем знание того, принадлежит ли точка с заданными координатами нашей прямой или нет. То есть, нужно получить некую математическую зависимость или уравнение прямой.

Во-первых, известно, что две различные точки определяют в пространстве прямую. Выберем в пространстве две точки  $P_1=(x_1, y_1, z_1)$  и  $P_2=(x_2, y_2, z_2)$  и соответствующие им радиус-векторы  $p_1, p_2$  и проведем через них прямую, как показано на рис. 5.3.

Проведем от точки  $P_1$  к точке  $P_2$  вектор  $p^* = p_2 - p_1$ . Тогда радиус-вектор  $p$ , определяющий некоторую точку на прямой, можно получить сложением,

например, вектора  $p_1$  и вектора  $p^*$ , умноженного на некоторое число  $\mu$ . Или  $p = p_1 - \mu p^*$ . Фактически мы уже получили уравнение прямой, но не через координаты двух точек на прямой, а другим способом, с помощью так называемых базового радиус-вектора  $p_1$  и направляющего радиус-вектора  $p^*$ . Преобразуем это уравнение к виду в котором используются только координаты двух исходных векторов  $p_1$  и  $p_2$ :

$$p = p_1 + \mu p^* = p_1 + \mu(p_2 - p_1) \Rightarrow p - p_1 = \mu(p_2 - p_1).$$

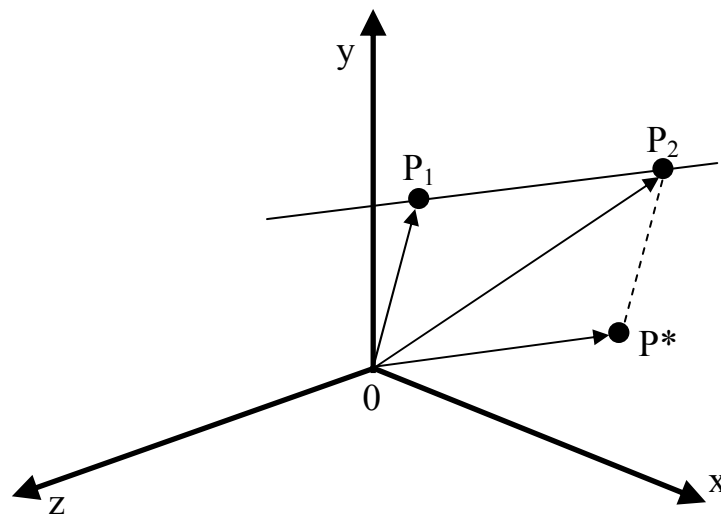


Рис. 5.3. Вывод уравнения прямой в трехмерном пространстве

Из этого векторного равенства получаем три равенства для соответствующих координат:

$$\begin{cases} x - x_1 = \mu(x_2 - x_1), \\ y - y_1 = \mu(y_2 - y_1), \\ z - z_1 = \mu(z_2 - z_1). \end{cases}$$

Попарно разделив эти уравнения друг на друга, для того чтобы избавиться от

коэффициента  $\mu$ , получаем следующую систему уравнений, определяющую прямую в трехмерном пространстве:

$$\begin{cases} (x-x_1)(y_2-y_1)=(x_2-x_1)(y-y_1), \\ (y-y_1)(z_2-z_1)=(y_2-y_1)(z-z_1), \\ (z-z_1)(x_2-x_1)=(z_2-z_1)(x-x_1). \end{cases}$$

Необходимо научиться управлять изображением на экране, вносить изменения в его положение, форму, ориентацию, размер. Для этих целей существуют специальные геометрические преобразования, которые позволяют изменять эти характеристики объектов в пространстве.

## 5.2. Двумерные матричные преобразования

Рассмотрим преобразования координат точек на плоскости. На рис. 5.4 точка  $A$  перенесена в точку  $B$ .

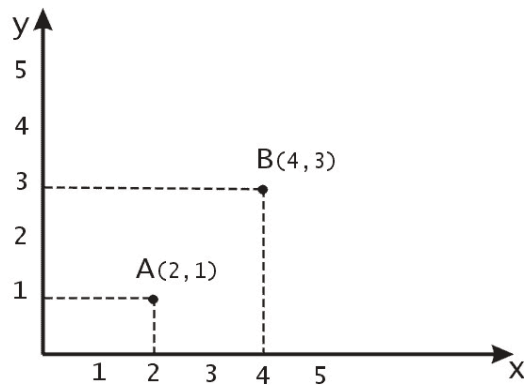


Рис. 5.4. Операция переноса или трансляции точки  $A$  в точку  $B$

Математически этот перенос можно описать с помощью вектора переноса  $\overline{AB}$ . Пусть  $\overline{R}$  радиус-вектор, соответствующий вектору переноса  $\overline{AB}$ .

Тогда переход из точки  $A$  в точку  $B$  будет соответствовать векторной записи  $\overline{B} = \overline{A} + \overline{R}$ . Отсюда получаем, что для переноса точки в новое положение необходимо добавить к ее координатам некоторые числа, которые представляют собой координаты вектора переноса:

$$\overline{B} = \overline{A} + \overline{R} = [A_x + R_x, A_y + R_y, A_z + R_z].$$

Масштабированием объектов называется растяжение объектов вдоль соответствующих осей координат относительно начала координат. Эта операция применяется к каждой точке объекта, поэтому можно также говорить о масштабировании точки. При этом, конечно, речь не идет об изменении размеров самой точки. Масштабирование достигается умножением координат точек на некоторые константы. В том случае, когда эти константы равны между собой, масштабирование называется однородным. На рис. 5.5 приведен пример однородного масштабирования треугольника  $ABC$ .

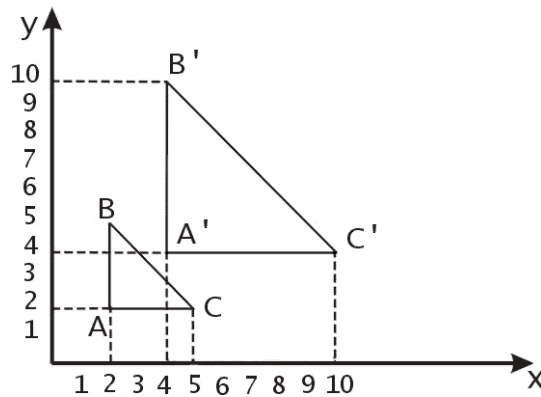


Рис. 5.5. Операция масштабирования

После применения операции однородного масштабирования с коэффициентом 2 он переходит в треугольник  $A'B'C'$ . Обозначим матрицу

масштабирования:

$$S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}.$$

Для точек  $A$  и  $A'$  операция масштабирования в матричном виде будет выглядеть следующим образом:

$$[x', y'] = [x, y] \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}.$$

Рассмотрим далее операцию вращения точки на некоторый угол относительно начала координат. На рис. 5.6 точка  $A = (x, y)$  переходит в точку  $B = (x', y')$  поворотом на угол  $\alpha$ .

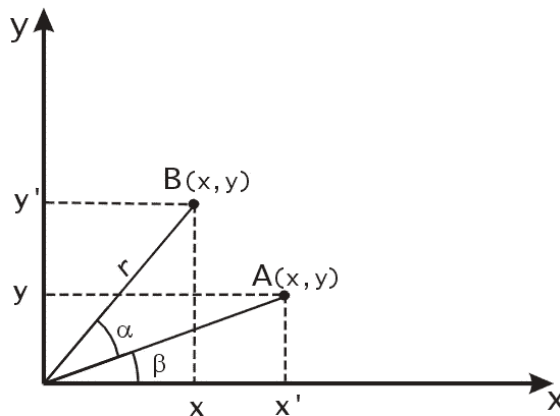


Рис. 5.6. Операция поворота точки  $A$  на угол  $\alpha$

Найдем преобразование координат точки  $A$  в точку  $B$ . Обозначим  $\beta$  угол, который составляет радиус-вектор  $\overline{OA}$  с осью  $Ox$ . Пусть  $r$  – длина радиус-вектора  $\overline{OA}$ , тогда

$$x' = r \times \cos(\alpha + \beta) = r(\cos \alpha \times \cos \beta - \sin \alpha \times \sin \beta),$$

$$y' = r \times \sin(\alpha + \beta) = r(\sin \alpha \times \cos \beta + \cos \alpha \times \sin \beta).$$

Так как  $\cos \beta = x/r$  и  $\sin \beta = y/r$ , то подставляя эти выражения в уравнения для  $x'$  и  $y'$ , получаем:

$$x' = x \times \cos \alpha - y \times \sin \alpha,$$

$$y' = x \times \sin \alpha + y \times \cos \alpha.$$

В матричном виде вращение точки  $A$  на угол  $\alpha$  выглядит следующим образом:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}.$$

### 5.3. Однородные координаты и матричное представление двумерных преобразований

В предыдущем параграфе были рассмотрены три вида преобразований точек на плоскости. Два из них – операции вращения и масштабирования – описываются в виде произведения матрицы на вектор, а третья – операция переноса – описывается как сумма двух векторов. В случае последовательного выполнения любой комбинации операций вращения и масштабирования результат легко можно записать в виде произведения матриц соответствующих преобразований. Это будет матрица результирующего поворота и масштабирования.

Очевидно, что удобнее применять результирующую матрицу вместо того,

чтобы каждый раз заново вычислять произведение матриц. Однако таким способом нельзя получить результирующую матрицу преобразования, если среди последовательности преобразований присутствует хотя бы один перенос. Матричное произведение в компьютерной графике также называют композицией. Было бы удобнее иметь математический аппарат, позволяющий включать в композиции преобразований все три выше указанные операции. При этом получился бы значительный выигрыш в скорости вычислений. Однородные координаты и есть этот математический аппарат.

Двумерный вектор  $(x, y)$  в однородных координатах записывается в виде  $(wx, wy, w)$ , где  $w \neq 0$ . Число  $w$  называется масштабным множителем. Для того чтобы из вектора, записанного в однородных координатах, получить вектор в обычных координатах, необходимо разделить первые две координаты на третью:  $(wx/w, wy/w, w/w) \rightarrow (x, y, 1)$ .

В общем случае осуществляется переход от  $n$ -мерного пространства к  $(n+1)$ -мерному. Это преобразование не единственное. Обратное преобразование называется проекцией однородных координат.

В однородных координатах преобразование центральной перспективы можно определить матричной операцией. Эта матрица записывается в виде:

$$\begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & k \end{bmatrix} = P$$

Покажем, что эта матрица определяет преобразование точки объекта, заданной в однородных координатах, в точку перспективной проекции



(также в однородных координатах). Пусть  $p = (x, y, z)$  – точка в трехмерном пространстве. Ее однородное представление  $v = (wx, wy, wz, w)$ . Умножим  $v$  на  $P$ :

$$vP = [wkx, wky, 0, w(z+k)] = [kx/(z+k), ky/(z+k), 0, 1].$$

Теперь точки двумерного пространства будут описываться трехэлементными вектор-строками, поэтому и матрицы преобразований, на которые будет умножаться вектор точки, будут иметь размеры  $3 \times 3$ . Запишем матричное преобразование операции переноса для однородных координат:

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}$$

$$\text{или } p' = pT(D_x, D_y), \text{ где } T(D_x, D_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}.$$

При последовательном переносе точки  $p$  в точку  $p'$  и затем в точку  $p''$  компоненты суммарного вектора переноса являются суммами соответствующих компонент последовательных векторов переноса. Рассмотрим, каковы будут элементы матрицы суммарного переноса. Пусть  $p' = pT(D_x, D_y)$ ,  $p'' = p'T(D'_x, D'_y)$ . Подставив первое уравнение во второе, получаем  $p'' = pT(D_x, D_y)p'T(D'_x, D'_y)$ . Матричное произведение, т.е. суммарный перенос, равен произведению соответствующих матриц переноса:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D'_x & D'_y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x + D'_x & D_y + D'_y & 1 \end{bmatrix}.$$

Запишем матричный вид операции масштабирования.

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Определим матрицу масштабирования  $S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ .

Так же, как последовательные переносы являются аддитивными, покажем, что последовательные масштабирования будут мультипликативными.

$$S(S_x, S_y) \times S(S'_x, S'_y) =$$

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S'_x & 0 & 0 \\ 0 & S'_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_x \times S'_x & 0 & 0 \\ 0 & S_y \times S'_y & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Для операции поворота матричный вид будет такой:

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Определим матрицу поворота  $R(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$ .

Аналогично двум предыдущим случаям, покажем, что матрица поворота остается таковой при последовательных поворотах.

$$\begin{aligned} R(\alpha)R(\beta) &= \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos(\alpha+\beta) & \sin(\alpha+\beta) & 0 \\ -\sin(\alpha+\beta) & \cos(\alpha+\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Таким образом, два, а значит и любое количество последовательных поворотов можно записать в виде одной матрицы суммарного поворота. Также легко заметить, что любая последовательность операций, включающая в себя перенос, масштабирование и вращение в однородных координатах, может быть представлена одной матрицей, которая является произведением матриц данных операций.

Рассмотрим, каким образом с помощью композиции матричных преобразований можно получить одно общее результирующее преобразование. Для этого будем использовать матрицы  $T$ ,  $S$  и  $R$ . С вычислительной точки зрения гораздо проще и быстрее применять матрицу уже готового преобразования вместо того, чтобы применять их последовательно одну за другой. К точке более эффективно применять одно результирующее преобразование, чем ряд преобразований друг за другом.

Для примера рассмотрим задачу поворота объекта на плоскости относительно некоторой произвольной точки  $p_0$ . Пока мы умеем поворачивать объекты только вокруг начала координат. Но можно представить эту задачу как последовательность шагов, на каждом из которых будет применяться только элементарная операция: перенос, масштабирование или вращение.

Вот эта последовательность элементарных преобразований (рис. 5.7):

1. Перенос, при котором точка  $p_0$  переходит в начало координат.
2. Поворот на заданный угол.
3. Перенос, при котором точка из начала координат возвращается в первоначальное положение  $p_0$ .

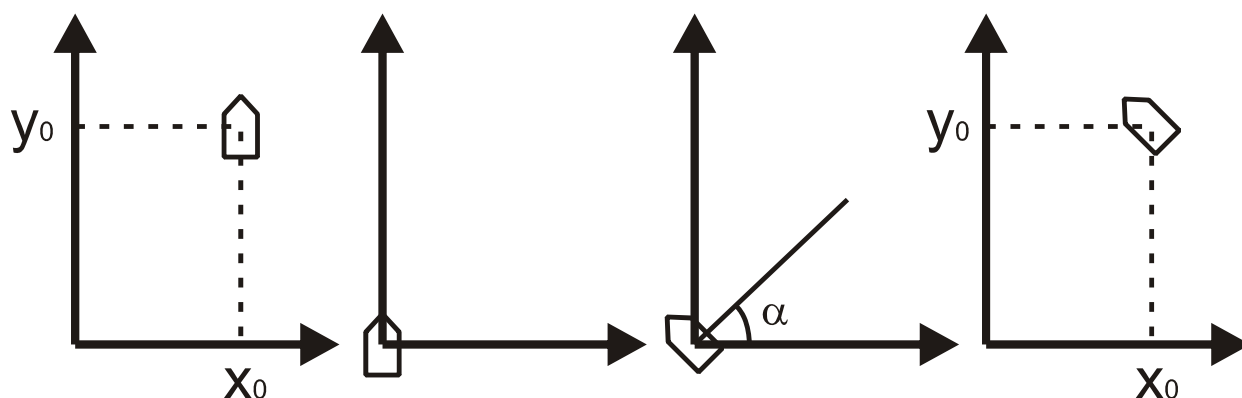


Рис. 5.7. Последовательность преобразований при повороте объекта вокруг точки  $p_0 = (x_0, y_0)$  на угол  $\alpha$

Точка  $p_0 = (x_0, y_0)$ . Первый перенос производится на вектор  $[-x_0, -y_0]$ , а обратный перенос - на вектор  $[x_0, y_0]$ .

#### 5.4. Трехмерные матричные преобразования

Подобно тому, как двумерные преобразования описываются матрицами размером  $3 \times 3$ , трехмерные преобразования могут быть представлены матрицами размером  $4 \times 4$ . Тогда трехмерная точка  $(x, y, z)$  записывается в однородных координатах как  $(wx, wy, wz, w)$ , где  $w \neq 0$ . Для получения декартовых координат надо первые три однородные координаты разделить на  $w$ .

Матрицы преобразований будем записывать в правосторонней системе координат. При этом положительный поворот определяется следующим образом. Если смотреть из положительной части оси вращения (например, оси  $Z$ ) в направлении начала координат, то поворот на  $90^\circ$  против часовой стрелки будет переводить одну положительную полуось в другую (ось  $X$  в  $Y$ , в соответствии с правилом циклической перестановки).

Заметим, что на практике удобнее применять левостороннюю систему координат, так как в этом случае удобнее интерпретировать тот факт, что точки с большими значениями  $Z$  находятся дальше от наблюдателя.

Запишем теперь матрицу трехмерного переноса. Аналогично двумерному случаю.

$$T(D_x, D_y, D_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ D_x & D_y & D_z & 1 \end{bmatrix}, \text{ при этом}$$

$$[x, y, z, 1] \cdot T(D_x, D_y, D_z) = [x + D_x, y + D_y, z + D_z, 1].$$

Операция масштабирования:

$$S(S_x, S_y, S_z) = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[x, y, z, 1] \cdot S(S_x, S_y, S_z) = [x \cdot S_x, y \cdot S_y, z \cdot S_z, 1].$$

Перейдем к операции поворота. Так как при двумерном повороте в плоскости  $xу$  координаты  $z$  остаются неизменными, то поворот вокруг оси  $z$  записывается так:

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Матрица поворота вокруг оси  $x$  имеет вид:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

и вокруг оси  $y$ :

$$R_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Обратите внимание на смену положения синуса угла с отрицательным знаком в матрице поворота вокруг оси  $y$ . Правильность этих матриц легко проверить поворотом одного из ортов на  $90^\circ$ , при этом он должен перейти в следующий по порядку орт на соответствующей координатной оси.

Обратные преобразования будут выражаться обратными матрицами. Для операции переноса надо лишь заменить знаки компонент вектора переноса на противоположные:

$$T^{-1}(D_x, D_y, D_z) = T(-D_x, -D_y, -D_z);$$

для операции масштабирования – на обратные значения:

$$S^{-1}(S_x, S_y, S_z) = s(1/S_x, 1/S_y, 1/S_z);$$

для поворота – выбором отрицательного угла поворота:

$$R^{-1}(\alpha) = R(-\alpha).$$

Результатом нескольких последовательных поворотов будет матрица

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Здесь верхняя матрица размером  $3 \times 3$  называется ортогональной. Важным ее свойством является то, что обратная к ней матрица является транспонированной:  $B^{-1} = B^T$ . Это полезно тем, что при вычислениях достаточно поменять индексы местами и обратное преобразование получается автоматически.

После перемножения любого числа матриц вида  $T$ ,  $S$  и  $R$  результирующая матрица всегда будет иметь вид:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}.$$

Здесь верхняя часть размером  $3 \times 3$  определяет суммарный поворот и масштабирование, а три коэффициента последней строки – суммарный перенос.



## 6. Цветовые модели

С физической точки зрения цвет – это набор определённых длин волн, отражённых от предмета или пропущенных сквозь прозрачный предмет. Однако сейчас нас интересует вопрос не о том, что такое цвет, какова его физическая природа, а то, как вообще на практике можно получить тот или иной цвет.

Цвета в природе редко являются простыми. Большинство цветов получаются смешением каких-либо других. Например, сочетание красного и синего даёт пурпурный цвет, синего и зелёного – голубой. Таким образом, путём смешения из небольшого количества простых цветов, можно получить множество (и при чём довольно большое) сложных (составных). Поэтому для описания цвета вводится понятие цветовой модели.



Цветовая модель – способ представления большого количества цветов посредством разложения их на простые составляющие.

Существует большое количество моделей. Рассмотрим только три основных.

### 6.1. Цветовая модель RGB

Множество цветов видны оттого, что объекты, их излучающие, светятся. К таким цветам можно отнести, например, цвета на экранах телевизора, монитора, кинопроектора. Цветов огромное количество, но из них выделено только три, которые считаются основными (первичными): это – красный, зелёный, синий.

При смешении двух основных цветов результирующий цвет осветляется: из смешения красного и зеленого получается желтый, из смешения

зеленого и синего получается голубой, синий и красный дают пурпурный. Если смешиваются все три цвета, в результате образуется белый. Такая модель цвета является аддитивной.

Модель, в основе которой лежат указанные цвета, носит название цветовой модели RGB – по первым буквам английских слов Red (Красный), Green (Зеленый), Blue (Синий).

Поскольку в модели используются три независимых значения, ее можно представить в виде трехмерной системы координат (рис. 6.1).

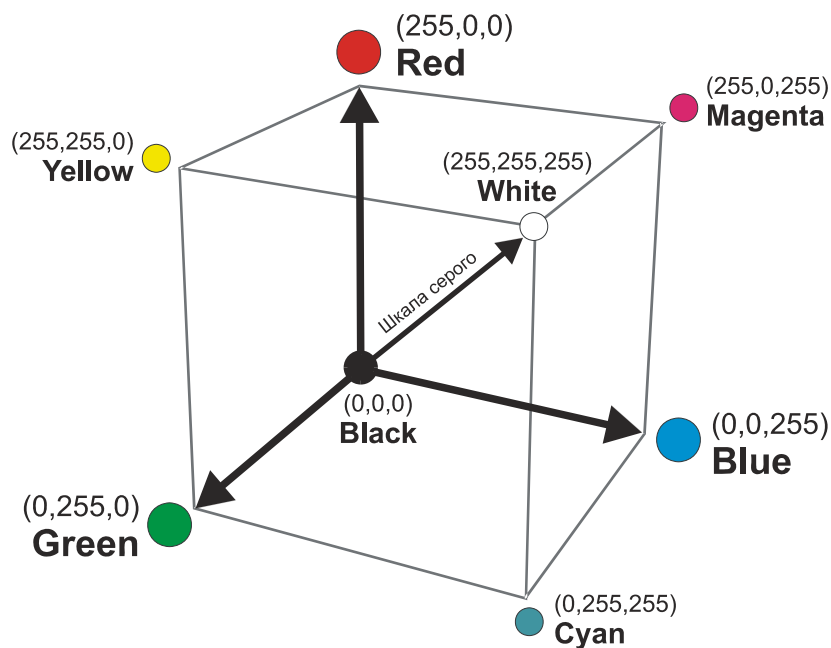


Рис. 6.1. Цветовая модель RGB

Каждая координата отражает вклад соответствующей составляющей в конкретный цвет в диапазоне от нуля до максимального значения. В результате получается некий куб, внутри которого и находятся все цвета, образуя цветное пространство модели RGB.

Объем такого куба (количество цифровых цветов) легко посчитать: поскольку на каждой оси можно отложить 256 значений, то 256 в кубе (или 2 в двадцать четвертой степени) дает число 16 777 216.

Важно отметить особые точки и линии этой модели.

Начало координат. В этой точке все составляющие равны нулю, излучение отсутствует, а это равносильно темноте, следовательно, начало координат — это точка черного цвета.

Точка, ближайшая к зрителю. В этой точке все составляющие имеют максимальное значение, что означает белый цвет.

Диагональ куба. На линии, соединяющей начало координат и точку, ближайшую к зрителю, располагаются серые оттенки: от черного до белого. Это происходит потому, что все три составляющих одинаковы и располагаются в диапазоне от нуля до максимального значения. Этот диапазон иначе называют серой шкалой (Grayscale). В компьютерных технологиях сейчас чаще всего используются 256 градаций (оттенков) серого. Хотя некоторые сканеры имеют возможность кодировать 1024 оттенков серого и выше.

Три вершины куба обозначают чистые исходные цвета, остальные три отражают двойные смешения исходных цветов.

Модель RGB является теоретической основой процессов сканирования и визуализации изображений на экране монитора.

## **6.2. Цветовая модель СМΥΚ**

Модель описывает отражаемые цвета. К отражаемым относятся цвета, которые сами не излучают, а используют белый свет, вычитая из него определенные цвета. Такие цвета называются субтрактивными

(вычитательными), поскольку они остаются после вычитания основных аддитивных. Существует три основных субтрактивных цвета: голубой, пурпурный, желтый (рис. 6.2).

Эти цвета составляют так называемую полиграфическую триаду. При печати красками этих цветов поглощаются красная, зеленая и синяя- составляющие белого цвета таким образом, что большая часть видимого цветового спектра может быть репродуцирована на бумаге.

При смешении двух субтрактивных составляющих результирующий цвет затемняется, а при смешении всех трех должен получиться черный цвет. При полном отсутствии краски остается белый цвет (белая бумага).

В итоге получается, что нулевые значения составляющих дают белый цвет, максимальные значения должны давать черный, их равные значения – оттенки серого, кроме того, имеются чистые субтрактивные цвета и их двойные сочетания.

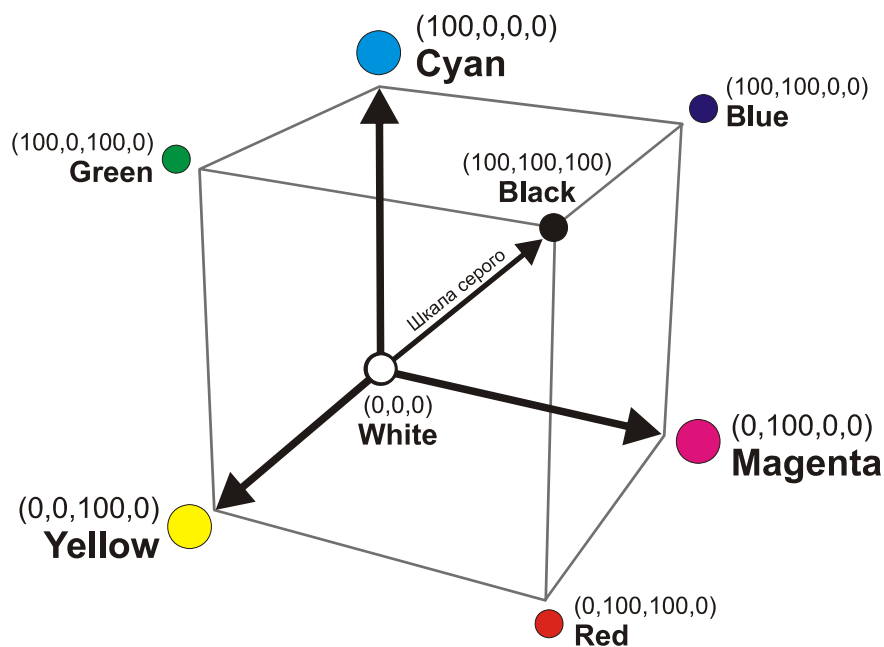


Рис. 6.2. Цветовая модель CMYK

Основная проблема заключается в том, что данная модель призвана описывать реальные полиграфические краски, которые далеко не так идеальны, как цветной луч. Они имеют примеси, поэтому не могут полностью перекрыть весь цветовой диапазон, а это приводит, в частности, к тому, что смешение трех основных красок, которое должно давать черный цвет, дает какой-то неопределенный (грязный) темный цвет, и это скорее темно-коричневый, чем глубокий черный цвет.

Для компенсации этого недостатка в число основных полиграфических красок была внесена черная краска. Именно она добавила последнюю букву в название модели CMYK, хотя и не совсем обычно: С – это Cyan (Голубой), М – это Magenta (Пурпурный), Y – Yellow (Желтый), К – это black (Черный), т.е. от слова взята не первая, а последняя буква.

Существует еще один вариант расшифровки CMYK – Cyan, Magenta, Yellow, Key color. Причем key color (ключевой цвет) может быть любым.

Таким образом, модели RGB и CMYK, хотя и связаны друг с другом, однако их взаимные переходы друг в друга (конвертирование) не происходят без потерь, поскольку цветовой охват у них разный. И речь идет лишь о том, чтобы уменьшить потери до приемлемого уровня. Это вызывает необходимость очень сложных калибровок всех аппаратных частей, составляющих работу с цветом: сканера (он осуществляет ввод изображения), монитора (по нему судят о цвете и корректируют его параметры), выводного устройства (оно создает оригиналы для печати), печатного станка (выполняющего конечную стадию).

Для устранения аппаратной зависимости был разработан ряд так называемых перцепционных (интуитивных) цветовых моделей. В их основу заложено отдельное определение яркости и цветности.

### 6.3. Цветовая модель HSB

Если модель RGB наиболее удобна для компьютера, а модель CMYK – для типографий, то модель HSB наиболее удобна для человека.

На цветовом круге (рис. 6.3) основные цвета моделей RGB и CMY находятся в такой зависимости: каждый цвет расположен напротив дополняющего его (комплементарного) цвета, при этом он находится между цветами, с помощью которых он получен. Например, сложение зеленого и красного цветов дает желтый.

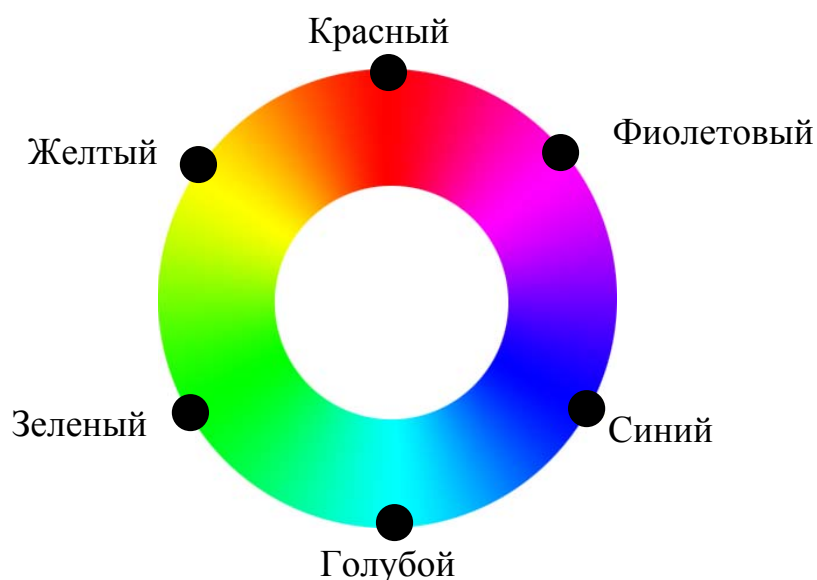


Рис. 6.3. Цветовой круг

Чтобы усилить какой-либо цвет, нужно ослабить дополняющий его цвет (расположенный напротив него на цветовом круге). Например, чтобы изменить общее цветовое решение в сторону голубых тонов, следует снизить в нем содержание красного цвета.

По краю этого цветового круга располагаются так называемые спектральные цвета или цветовые тона (Hue), которые определяются длиной

световой волны, отраженной от непрозрачного объекта или прошедшей через прозрачный объект. Цветовой тон характеризуется положением на цветовом круге и определяется величиной угла в диапазоне от 0 до 360 градусов. Эти цвета обладают максимальной насыщенностью, т. е. синий цвет еще синее быть уже не может.

Следующим параметром является насыщенность цвета (Saturation) — это параметр цвета, определяющий его чистоту.

Уменьшение насыщенности цвета означает его разбеливание. Цвет с уменьшением насыщенности становится пастельным, блеклым, размытым. На модели все одинаково насыщенные цвета располагаются на концентрических окружностях, т.е. можно говорить об одинаковой насыщенности, например, зеленого и пурпурного цветов, и чем ближе к центру круга, тем все более разбеленные цвета получают. В самом центре любой цвет максимально разбеливается и становится белым цветом.

Поэтому работу с параметром насыщенности можно характеризовать как добавление в спектральный цвет определенного процента белой краски.

Еще одним параметром является яркость цвета (Brightness) – это параметр цвета, определяющий освещенность или затемненность цвета. Уменьшение яркости цвета означает его зачернение.

Поэтому работу с параметром яркости можно характеризовать как добавление в спектральный цвет определенного процента черной краски.

В общем случае, любой цвет получается из спектрального цвета добавлением определенного процента белой и черной красок, т. е. фактически серой краски.

Эта модель уже гораздо ближе к традиционному пониманию работы с цветом. Можно определять сначала цветовой тон (Hue), а затем

насыщенность (Saturation) и яркость (Brightness). Такая модель получила название по первым буквам приведенных выше английских слов — HSB (рис. 6.4).

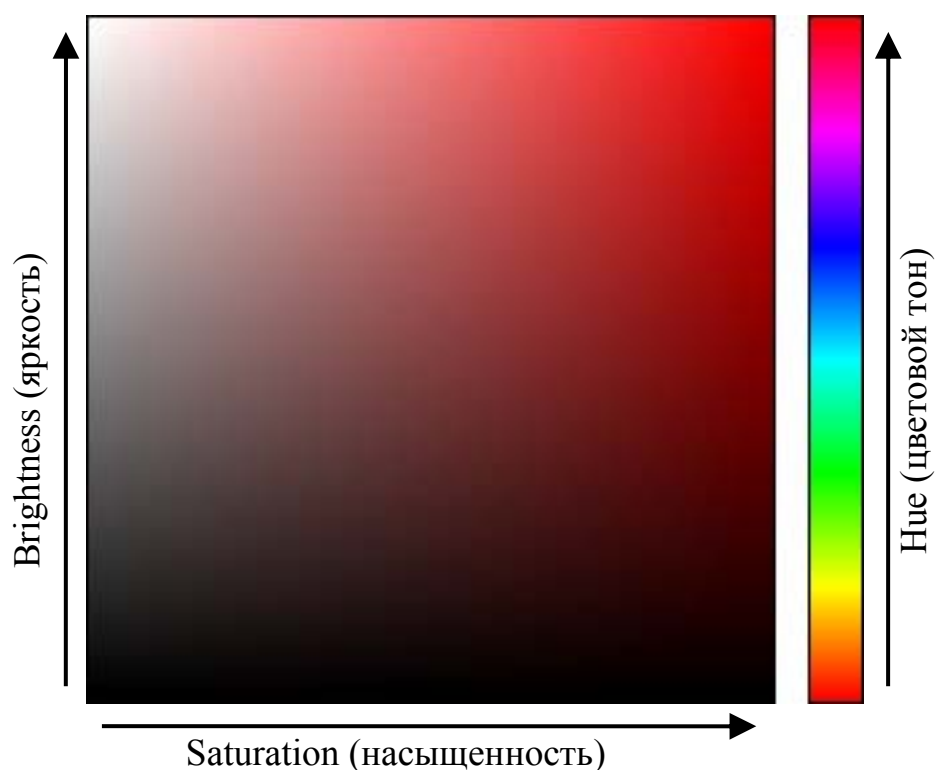


Рис. 6.4. Цветовая модель HSB

Модель HSB неплохо согласуется с восприятием человека: цветовой тон является эквивалентом длины волны света, насыщенность — интенсивности волны, а яркость — количества света.

Недостатком этой модели является необходимость преобразовывать ее в модель RGB для отображения на экране монитора или в модель CMYK для получения полиграфического оттиска.



## 7. Растровая (пиксельная) графика

Растровую графику применяют при разработке электронных и полиграфических изданий. Иллюстрации, выполненные средствами растровой графики, редко создают вручную с помощью компьютерных программ. Чаще используют отсканированные иллюстрации, подготовленные художником на бумаге, или фотографии. В последнее время для ввода растровых изображений в компьютер нашли широкое применение цифровые фото- и видеокамеры. Соответственно, большинство растровых графических редакторов ориентированы не столько на создание изображений, сколько на их обработку.



Растр – прямоугольная сетка точек, формирующая изображение на экране компьютера.

Каждая точка растра характеризуется двумя параметрами: своим положением на экране и своим цветом, если монитор цветной, или степенью яркости, если монитор черно-белый.

Растровая графика представляет изображения в виде массива цифр. Поэтому при большом увеличении все изображения выглядят как мозаика (сетка), состоящая из мельчайших ячеек (рис. 7.1).

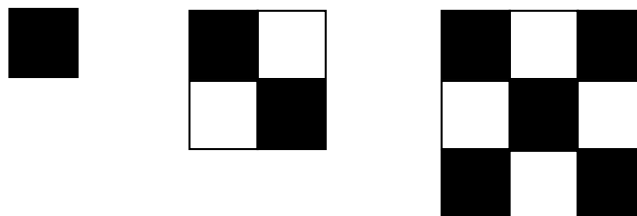


Рис. 7.1. Простые пиксельные изображения

Сама сетка получила название растровой карты (bitmap), а элементы, ее формирующие, называются пикселями.



Пиксель (pixel) – мельчайший элемент изображения, воспроизводимый компьютером.

Слово пиксель произошло от сокращения picture element (элемент изображения) и замены буквы p на x.

Отличительными особенностями пикселя являются его однородность (все пиксели по размеру одинаковы) и неделимость (внутри пикселя не может быть никаких более мелких элементов).

Если пиксели достаточно малы, глаз воспринимает «пиксельную мозаику» как одно целое изображение (рис. 7.2, а).

При масштабировании растровых изображений возникают искажения – ступеньки (aliasing) (рис. 7.2, б).

**aliasing**



а)

б)

в)

Рис. 7.2: Растровое изображение: а – в натуральном масштабе;  
б – в увеличенном масштабе; в – в увеличенном масштабе с применением  
anti-aliasing

Растровые редакторы позволяют частично убрать эти искажения за счет применения специальных алгоритмов обработки. Эта операция называется anti-aliasing (рис. 7.2, в).

## 7.1. Разрешение растровой графики

Качество цифрового изображения определяется многими параметрами. Одним из ключевых является понятие разрешения.



Разрешение (resolution) – количество дискретных элементов на единицу длины.

За единицу длины был принят дюйм (inch), равный 25,4 мм. Как было рассмотрено выше, дискретный элемент это пиксель. Таким образом, разрешение можно определить как количество пикселей в дюйме, обычно оно обозначается как ppi, что является сокращением от словосочетания pixels per inch (пикселей в каждом дюйме).

Однако в реальной жизни связанная с разрешением терминология не так однозначна. В зависимости от устройства, на котором выводится изображение, возможно использование следующих единиц измерения разрешения:

- spi (sample per inch) – элементов на дюйм;
- dpi (dot per inch) – точек на дюйм;
- ppi (pixel per inch) – пикселей на дюйм;
- lpi (line per inch) – линий на дюйм.

## 7.2. Виды разрешения

В компьютерной графике с понятием разрешения обычно происходит больше всего путаницы, поскольку приходится иметь дело сразу с несколькими свойствами разных объектов. Следует четко различать: разрешение экрана, разрешение печатающего устройства и разрешение изображения. Все эти понятия относятся к разным объектам. Друг с другом эти виды разрешения никак не связаны, пока не потребуется узнать, какой физический размер будет иметь картинка на экране монитора, отпечаток на бумаге или файл на жестком диске.

*Разрешение экрана* – это свойство компьютерной системы (зависит от монитора и видеокарты) и операционной системы (зависит от настроек). Разрешение экрана измеряется в пикселях и определяет размер изображения, которое может поместиться на экране целиком.

*Разрешение принтера* – это свойство принтера, выражающее количество отдельных точек, которые могут быть напечатаны на участке единичной длины. Оно измеряется в единицах dpi (точки на дюйм) и определяет размер изображения при заданном качестве или, наоборот, качество изображения при заданном размере.

*Разрешение изображения* – это свойство самого изображения. Оно измеряется в пикселях на дюйм и задается при создании изображения в графическом редакторе или при сканировании. Значение разрешения хранится в файле изображения и неразрывно связано с другим его свойством – физическим размером.

Физический размер изображения может измеряться как в пикселях, так и в единицах длины (миллиметрах, сантиметрах, дюймах).

### 7.3. Кодирование изображения

Рассмотрим процесс кодирования (оцифровки) изображения.

Изображение (оригинал) разбивается на дискретные элементы (пиксели), и так как это происходит в декартовой системе координат, то элементы имеют форму прямоугольника.

Оригинал помещается в декартовы координаты, по нижнему краю располагается ось  $X$ , по левому ось  $Y$  (рис. 7.3). По обеим осям устанавливаются одинаковые интервалы отсчета.

Сетка дискретизации является матрицей, ячейки которой пронумерованы и в них можно записывать значения.

Матрица, которая создается средствами графической программы, называется *map* (карта), а так как в нее записываются биты информации, она получила название битовая карта (*bitmap*).

Перейдем к заполнению битовой карты.

Рассмотрим случай с изображением, состоящим из двух цветов (рис. 7.4, а), т.е. необходимо закодировать оригинал по двум состояниям.

Там, где в оригинале дискретные ячейки имеют белый цвет, в соответствующие им ячейки матрицы (битовой карты) записываются нули, а где черный – единицы.

После полного заполнения всех ячеек, можно считать, что процесс оцифровки (кодирования) оригинала завершился и полученная таблица (рис. 7.4, б) может быть записана на любом цифровом устройстве.

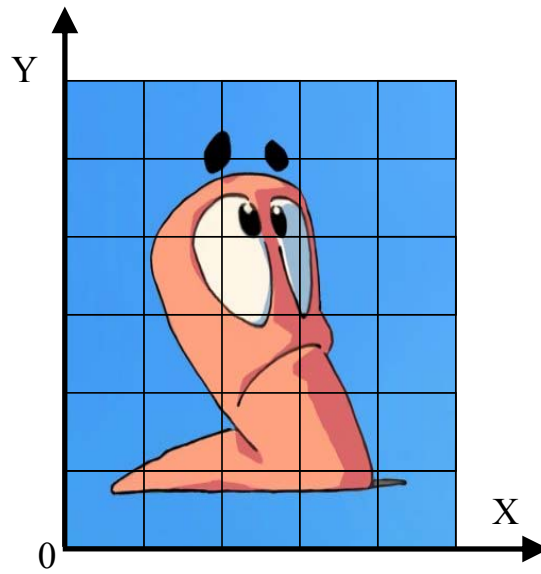
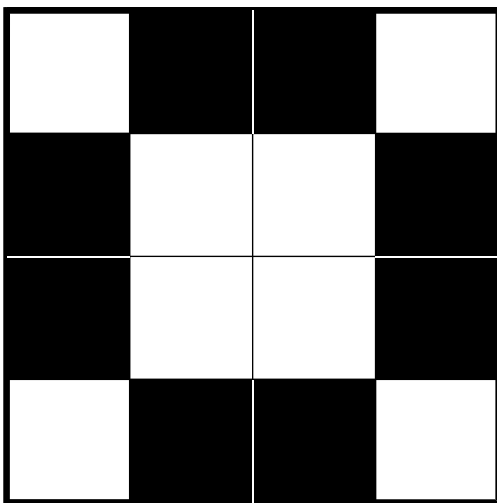


Рис. 7.3. Изображение в декартовых координатах



а)

0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0

б)

Рис. 7.4. Заполнение матрицы битами:

а – изображение, состоящее из двух цветов;

б – кодирование оригинала

#### 7.4. Глубина цвета

Далеко не все изображения мы можем свести к черно-белому варианту, более того, таких изображений как раз значительно меньше. Существует огромное количество изображений, которые представляют собой непрерывный тон (плавный переход между оттенками серого и цветовыми оттенками). Не говоря уже об окружающей реальности.

Первоначально рассмотрим только тоновые изображения, к которым относятся, прежде всего, классическая черно-белая фотография, а также некоторые художественные техники: рисунок карандашом, акварель и масляная живопись, некоторые виды гравюр и т.д.

Здесь история развития технических систем шла, в общем, путем увеличения количества градаций тона (цвета): сначала использовались 16 градаций, потом – 64, и наконец – 256 градаций. В настоящее время существуют системы, позволяющие оперировать 1024 градациями.

Как все виды информации, тоновые уровни также были привязаны к необходимости оперировать совокупностью байтов. А поскольку байт – это восемь бит, или восемь разрядов, количество возможных градаций тона можно рассчитать следующим образом:  $2^8 = 256$ .

Следовательно, таблица квантования, предназначенная для передачи непрерывного тона, состоит из 256 строк, т.е. 256 двоичных кодов, в диапазоне от 00000000 до 11111111.

Таким образом, начальному коду (то есть 0 в десятичной системе счисления) ставим в соответствие черный цвет, а конечному (то есть 255 в десятичной системе счисления) - белый цвет, остальные 254 кода будут соответствовать оттенкам серого: от темного, близкого к черному, до очень светлого,

близкого к белому.

Теперь стоит вернуться к виртуальной цифровой матрице, которая выше была определена как битовая карта.

Вспомним, что в каждую ячейку матрицы записывается бит информации, т.е. одно двоичное число (0 или 1).

Теперь, в связи с тоновыми изображениями, ситуация изменилась: каждый пиксель тонового изображения требует для своего представления не одного двоичного числа, а восьми, т.е. целого байта информации. И требуется сообразить, где разместить дополнительные семь разрядов.

Проблема решается за счет добавления новых битовых карт по числу дополнительных разрядов. Эти дополнительные карты располагаются как бы в пространстве, а именно в глубину (рис. 7.5).

И, соответственно, общее количество битовых карт (а по сути разрядов или двоичных цифр) определяет глубину таблицы квантования. В английской терминологии этот параметр получил название *color depth*, что дословно означает цветовая глубина, а в русском языке прижилась форма глубина цвета.



Глубина цвета – количество бит, приходящихся на один пиксель (bpp). Определяет количество бит, или разрядов, с помощью которых составляются коды потенциальных значений тона или цвета.



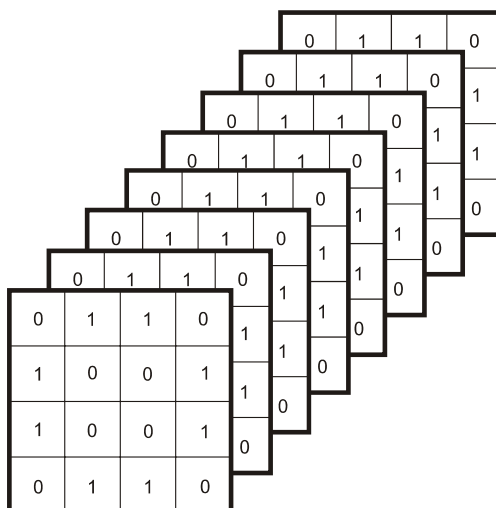


Рис. 7.5. Дополнительные битовые карты

Глубина цвета – это важнейший параметр цифровой графики, поэтому он должен иметь количественную меру. Следовательно, необходимо определить и принять соответствующую единицу измерения.

Глубина цвета измеряется в количестве разрядов, или битов, которые отводятся на каждый пиксель изображения, т.е. общее количество битовых карт и будет единицей измерения глубины цвета.

Если используется тоновое изображение, то глубина цвета такого изображения традиционно равна восьми битам, поэтому такое изображение называют 8-bit image (восьмибитным изображением), кроме того, у него есть специальное название: grayscale (серая шкала).

Перейдем к полноцветным изображениям.

Для того чтобы оцифровать и сохранить цветовую информацию, все технические системы используют цветовую фильтрацию: пропускают цветовой поток через три цветных фильтра (красный, зеленый и синий). В результате этой процедуры создаются три изображения в градациях серого (grayscale). В этих изображениях каждый пиксель описывается восемью двоичными разрядами, в сумме это составит 24 бита, т.е. полноцветные

изображения называются 24-bit image.

Цветовые составляющие в программе организуются в виде так называемых каналов, каждый из которых представляет собой изображение в градациях серого со значениями яркости от 0 до 255 (рис. 7.6).



Рис. 7.6. Палитра каналов программы Adobe Photoshop

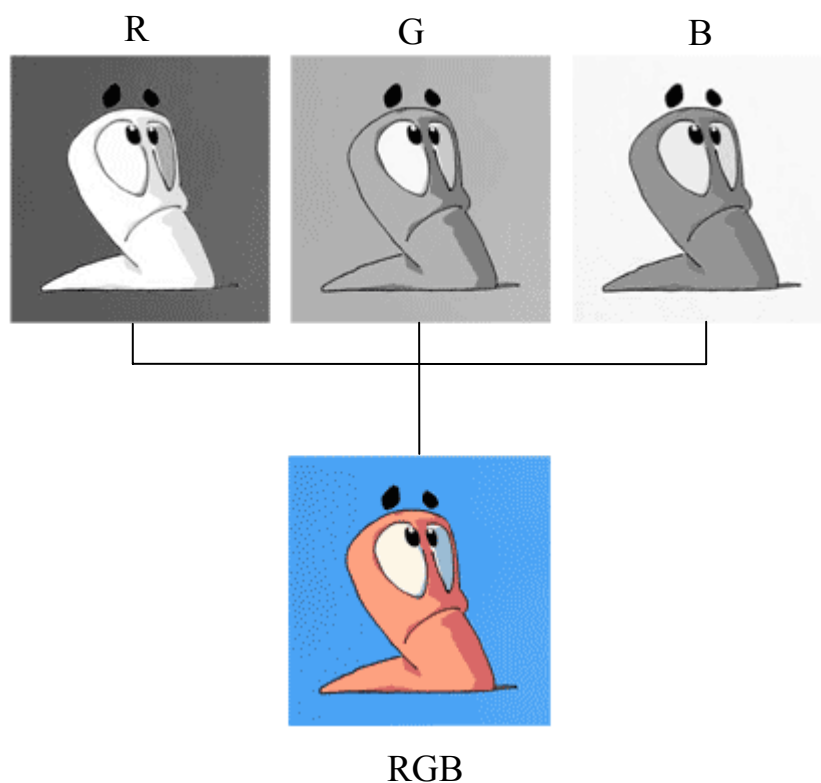


Рис. 7.7. Совмещение тоновых градаций каналов

Совмещение тоновых градаций всех каналов и определяет цвет изображения (рис. 7.7).

## 7.5. Цветовые палитры



Цветовая палитра – это таблица данных, в которой хранится информация о том, каким кодом закодирован тот или иной цвет. Эта таблица создается и хранится вместе с графическим файлом.

Самый удобный для компьютера способ кодирования цвета— 24-разрядный, True Color. В этом режиме на кодирование каждой цветовой составляющей R (красной), G (зеленой) и B (синей) отводится по одному байту (8 бит). Яркость каждой составляющей выражается числом от 0 до 255, и любой цвет из 16,5 миллионов компьютер может воспроизвести по трем кодам. В этом случае цветовая палитра не нужна, поскольку в трех байтах и так достаточно информации о цвете конкретного пикселя.

### *Индексная палитра.*

Существенно сложнее обстоит дело, когда изображение имеет только 256 цветов, кодируемых одним байтом. В этом случае каждый цветовой оттенок представлен одним числом, причем это число выражает не цвет пикселя, а индекс цвета (его номер). Сам же цвет разыскивается по этому номеру в сопроводительной цветовой палитре, приложенной к файлу. Такие цветовые палитры еще называют индексными палитрами. Разные изображения могут иметь разные цветовые палитры. Например, в одном изображении зеленый цвет может кодироваться индексом 64, а в другом изображении этот индекс может быть отдан розовому цвету.

### *Фиксированная палитра.*

В тех случаях, когда цвет изображения закодирован двумя байтами (режим High Color), на экране возможно изображение 65 тысяч цветов. Разумеется, это не все возможные цвета, а лишь одна двести пятьдесят шестая доля общего непрерывного спектра красок, доступного в режиме True Color. В таком изображении каждый двухбайтный код тоже выражает какой-то цвет из общего спектра. Но в данном случае нельзя приложить к файлу индексную палитру, в которой было бы записано – какой код какому цвету соответствует, поскольку в этой таблице было бы 65 тысяч записей, и ее размер составил бы сотни тысяч байтов. Вряд ли есть смысл прикладывать к файлу таблицу, которая может быть по размеру больше самого файла. В этом случае используют понятие фиксированной палитры. Ее не надо прикладывать к файлу, поскольку в любом графическом файле, имеющем шестнадцатиразрядное кодирование цвета, один и тот же код всегда выражает один и тот же цвет.

### *Безопасная палитра.*

Термин безопасная палитра используют в web-графике. Поскольку скорость передачи данных в Интернете пока оставляет желать лучшего, для оформления web-страниц не применяют графику, имеющую кодирование цвета выше 8-разрядного.

При этом возникает проблема, связанная с тем, что создатель web-страницы не имеет ни малейшего понятия о том, на какой модели компьютера и под управлением каких программ будет просматриваться его произведение.

В связи с этим было принято следующее решение. Все наиболее популярные программы для просмотра web-страниц (браузеры) заранее настроены на некоторую одну фиксированную палитру. Если разработчик web-страницы при создании иллюстраций будет применять только эту палитру, то он может

быть уверен, что пользователи всего мира увидят рисунок правильно.

В этой палитре не 256 цветов, как можно было бы предположить, а лишь 216. Это связано с тем, что в Интернете работают люди с разными компьютерами, а не только с IBM PC, и не все компьютеры могут воспроизводить 256 цветов.

Такая фиксированная палитра, жестко определяющая индексы для кодирования 216 цветов, называется безопасной палитрой.

## **7.6. Основные редакторы растровой графики**

Существует множество различных графических редакторов. Некоторые из них, такие как Adobe PhotoShop, предназначены для профессиональной работы с графикой. Это коммерческие продукты, которые стоят немалых денег. Другие, например Paint, встроенный в ОС Windows, доступны для работы даже маленькому ребенку, но и возможности у них не велики.

В ОС Linux на данный момент представлен весь спектр графических редакторов. Первое место среди них, бесспорно, занимает мощнейший редактор GIMP, обладающий инструментами как для создания десятков разновидностей форматов растровой графики, так и средствами подготовки векторной графики. В его состав входит уникальный редактор фрактальной графики, позволяющий создавать настоящие произведения искусства. Следует также отметить, что как и большинство Linux-приложений, редактор GIMP распространяется бесплатно, что также является немаловажным плюсом в пользу его применения.

### *Редактор GIMP*

GIMP – это мощный профессиональный графический редактор с массой вспомогательных программ. Само название GIMP является аббревиатурой

GNU Manipulation Image Program и переводится на русский язык как программа обработки изображений.

Первая версия GIMP была выпущена в феврале 1996 г. и оказала большое влияние как первая действительно профессиональная и бесплатная программа обработки изображений, которая могла конкурировать с большими коммерческими графическими редакторами. GIMP постоянно развивается: появляются версии программы с новыми функциональными возможностями.

Разработчики программы постарались включить в нее все, начиная с основных средств рисования и редактирования, без которых не обходится ни один графический редактор, и заканчивая самыми современными средствами обработки изображений. GIMP включает все те средства, которые есть в программе Photoshop, и «понимает» формат файлов, используемый в этой программе, так что пользователи Photoshop могут легко перейти на использование GIMP. Есть версия этого редактора для ОС Windows.

Имеются в ОС Linux и другие графические редакторы, более специализированные для работы с тем или иным форматом графики. Среди них:

- XPaint – редактор растровой графики, похожий на Paint, но содержащий массу дополнительных возможностей;
- KPaint – встроенный в KDE редактор растровой графики, менее функционален чем XPaint;
- KDE Icon Editor – редактор растровой графики, предназначенный для создания пиктограмм.

Сайт программы: [www.gimp.org](http://www.gimp.org); [www.gimp.ru](http://www.gimp.ru).

### *Редактор Paint*

Этот редактор является одним из самых простых графических редакторов в среде MS Windows. Он появился в версии Windows 95 и с тех пор не изменился. Paint (в переводе с английского означает краска) представляет собой простой однооконный редактор растровой графики, который, тем не менее, позволяет создать достаточно сложный рисунок.

Программа включает средства для построения прямых и кривых линий, эллипсов и окружностей, прямоугольников, квадратов и многоугольников (как контурных, так и закрашенных). Есть инструменты для выделения фрагмента рисунка, заливки замкнутой области цветом, а также инструменты, имитирующие рисование кистью и пульверизатором. Имеется возможность создания надписи и задания толщины линии.

Доступны и некоторые операции преобразования рисунка, а именно: зеркальное отображение относительно горизонтальной и вертикальной осей, инвертирование и замена цветов, сжатие, растяжение и наклон. Однако в Paint совершенно отсутствуют разного рода эффекты и фильтры. Кроме того, этот редактор поддерживает всего несколько форматов файлов.

### *Редактор Adobe Photoshop*

Среди графических редакторов является стандартом де-факто. Производится, распространяется и поддерживается компанией Adobe.

Это мощный и гибкий инструмент, удовлетворяющий требованиям почти любого пользователя – от новичка, знакомящегося с компьютерной графикой, до опытного профессионала-дизайнера. Богатые возможности по обработке графических файлов всех популярных форматов, наличие удобного и интуитивно понятного интерфейса, принцип открытой архитектуры, позволяющий другим компаниям свободно разрабатывать

дополнительные модули делают Photoshop безусловным лидером.

Для работы с анимированной графикой пользователю предоставляется отдельная программа ImageReady, включенная в дистрибутив Photoshop.

Сайт программы: [www.adobe.com/products/photoshop](http://www.adobe.com/products/photoshop).

### *Редактор Paint Shop Pro*

По лицензии Shareware распространяется хорошо известный во всем мире графический редактор Paint Shop Pro, предлагаемый компанией Jasc Software.

Paint Shop Pro поддерживает все распространенные форматы графических файлов, в том числе и PSD, принадлежащий Adobe Photoshop, его главному конкуренту. В распоряжение пользователя попадает весьма объемная библиотека заранее разработанных эффектов и фильтров (их больше 70). Имеется также возможность создавать собственные эффекты и сохранять их в специальном файле для применения в будущем.

Для работы с анимированной графикой пользователю предоставляется отдельная программа Animation Shop, устанавливаемая из одного дистрибутива с самим редактором.

Незарегистрированная копия позволяет свободно редактировать графические файлы всех форматов. Желание компании получить с пользователя оплату выражается во время работы с программой не слишком назойливыми сообщениями. После окончания тридцатидневного срока, предоставленного на то, чтобы войти во вкус, пользователю, не успевшему убедиться во всех преимуществах Paint Shop Pro, дается своеобразный бонус: он может бесплатно поработать с программой еще два месяца. Но вот после этого программа уже окончательно перестает функционировать.

Сайт программы: [www.jasc.com/products/paintshoppro](http://www.jasc.com/products/paintshoppro).



### *Редактор PhotoImpact*

Вполне полноценную, пробную версию программы PhotoImpact предлагает скачать из Интернета и проверить на практике компания Ulead Systems, известная своими графическими и мультимедийными продуктами.

PhotoImpact позволяет весьма качественно улучшать фотографии, сделанные при плохом освещении, неплохо справляется с автоматической подгонкой цветов объектов в соответствии с фоном, имеет большое количество встроенных фильтров, тщательно разработанных с учетом баланса удобства и функциональности.

При использовании незарегистрированной версии, программа успешно и полноценно обрабатывает графические файлы всех распространенных форматов. В то же время компания отмечает, что пробная версия «слегка облегчена» изъятием аниматора, образцов рисунков, некоторых текстур и т.д. По окончании месячного срока со дня установки программа полностью перестает функционировать, и пользователю остается или отказаться от нее, или заплатить за регистрацию.

Сайт программы: [www.ulead.com/pi/](http://www.ulead.com/pi/).

### *Редактор Corel PHOTO-PAINT*

Еще один стандарт растровой графики, который вместе с PhotoShop охватывает большинство пользователей, работающих с графическими пакетами. Он предоставляет большие возможности не только в области создания изображений, но, благодаря большому количеству корректирующих инструментов и настроек, также позволяет работать с уже готовыми изображениями, фотографиями и иллюстрациями. Причем данный редактор позволяет работать не только над композицией, но и над цветом, что является немаловажным и наиболее часто используемым в повседневной практике.

Corel Photo Paint входит в состав пакета Corel Graphics Suite.

Сайт программы: [www.corel.com](http://www.corel.com).

### *Редактор Macromedia Fireworks*

Программу Macromedia Fireworks можно отнести к разряду универсальных web-редакторов, потому что с помощью Fireworks можно:

- создавать графические изображения, используя векторные и растровые инструменты;
- добавлять интерактивные элементы и анимацию;
- сохранять в формате HTML and Images, экспортировать для работы в других программах.

Программа предназначена для создания и редактирования как векторной, так и растровой графики для web.

С ее помощью можно создавать интерактивные элементы интерфейса: различные кнопки и выпадающие меню, нарезать картинки на части (слайсы/slices) для дальнейшей генерации HTML-кода.

Использование программы вместе с другими продуктами Macromedia (Dreamweaver, Flash) обеспечивает комплексность процесса приготовления Web-продукции, хотя и использование Fireworks отдельно дает прекрасные результаты.

Сайт программы: [www.macromedia.com/software/fireworks](http://www.macromedia.com/software/fireworks).

## **7.7. Форматы файлов растровой графики**

Знание файловых форматов и их возможностей является одним из ключевых факторов в допечатной подготовке изданий, подготовке изображений для web и в компьютерной графике вообще. Каждый из утвердившихся

форматов прошел естественный отбор и доказал свою жизнеспособность. Все они имеют какие-то характерные особенности и возможности, делающие их незаменимыми в работе.

### *Формат РСХ*

Формат графических файлов разработанный фирмой ZSoft для работы с графическим редактором PC Paintbrush. Поддерживается множеством графических редакторов, программами сканирования изображений, настольными издательскими системами, программами рассылки факсов. Наряду с ограниченностью цветовой палитры, другим серьезным недостатком РСХ формата является наличие многочисленных версий. Последнее обусловлено тем, что РСХ стандарт является открытым и используется многими независимыми разработчиками программного обеспечения.

Формат РСХ использует простейший способ сжатия изображений, позволяющий выполнять быструю перезапись изображения из файла в видеопамять и обратно.

### *Формат BMP*

Разработка фирмы Microsoft для операционной системы Windows (сокращение от слова bitmap). Он поддерживается всеми графическими редакторами, работающими под управлением этой операционной системы.

Представляет собой чрезвычайно простую структуру и служит для описания и визуализации небольших пиктограмм – иконок (icons), применяемых в графических интерфейсах. Способен хранить как индексированный (до 256 цветов), так и RGB-цвет (16 700 оттенков). Возможно применение сжатия по принципу RLE. Существует разновидность формата BMP для операционной системы OS/2.

## *Формат GIF*

Формат GIF (Graphics Interchange Format) является одним из двух форматов файлов, наиболее часто используемых для изображений в Internet, который поддерживается почти всеми web-браузерами. Этот формат позволяет отобразить не более 256 цветов, поэтому его лучше всего использовать для черно-белых графических изображений, цветных картинок и рисунков, содержащих крупные фрагменты одного цвета.

Существует две разновидности формата. Первая официальная версия GIF была GIF87a, названная так по году выхода официальной документации. Затем вышла новая усовершенствованная версия формата, обладающая дополнительными возможностями и получившая название GIF89a. Обе эти разновидности поддерживают технологию чередования строк (interlacing) при помощи которой изображение может отрисовываться на экране путем постепенного увеличения четкости. Это особенно полезно, когда изображения выводятся на экран по мере поступления данных, как в случае с программами просмотра (браузерами) World Wide Web. Кроме того, формат GIF89a позволяет добиваться еще целого ряда сильных визуальных эффектов:

- маски прозрачности;
- включение нескольких изображений;
- глобальные цветовые палитры для нескольких изображений;
- анимация (попеременный показ нескольких изображений);
- включение текстовых комментариев;
- настраиваемые задержки показа и чередование (interlacing);
- ожидание реакции пользователя.

Разработчики программ, использующих формат GIF (читающих и

записывающих файлы в этом формате), должны заплатить за лицензию на алгоритм сжатия LZW, примененный в GIF.

### *Формат PNG*

PNG (произносится как «пинг») — аббревиатура словосочетания Portable Network Graphics. Как видно из названия, этот формат предназначен специально для передачи изображений по сетям. Формат PNG мог бы сделать революцию в web-дизайне, если бы Microsoft Internet Explorer поддерживал его.

Спецификация 1.0 формата PNG выглядит впечатляюще: как и GIF, формат поддерживает сжатие без потерь, чересстрочность (interlacing), прозрачность цвета (transparency), хранение пользовательских данных (например, комментариев). Но, в отличие от GIF, он может сохранять truecolor-изображения с глубиной цвета до 48 бит на пиксель, изображения в шкале серого цвета с глубиной до 16 бит на пиксель, поддерживает альфа-канал и гамма-индикатор. Кроме того, PNG может использоваться абсолютно бесплатно. Правда, PNG 1.0 не может хранить несколько изображений в одном файле, и, следовательно, анимация, одна из самых привлекательных черт формата GIF, в PNG пока не доступна.

Существует два подформата: PNG8 и PNG24. Цифры в названии обозначают максимально возможную глубину цвета. В PNG24 реализована поддержка 256 градаций прозрачности, за счет дополнительного альфа-канала с 256 градациями серого.

### *Формат JPEG*

Формат JPEG (Joint Photographic Experts Group) повсеместно используется для отображения фотографий и других тоновых изображений в WWW и в других сетях.

В отличие от формата GIF, формат JPEG сохраняет всю информацию о цвете. Кроме того, формат JPEG использует очень эффективный алгоритм уплотнения, который нередко дает значительное сокращение объема файла за счет удаления избыточной информации (от 10 до 60 процентов от начального объема файла), не влияющей на видимое качество отображения документа. При открытии JPEG-файла происходит его автоматическая распаковка. Недостатком JPEG-уплотнения является то, что оно приводит к частичной потере данных: если упаковать изображение, а затем вновь распаковать его, то результат не будет идентичным оригиналу. Между качеством изображения и степенью уплотнения существует обратная зависимость: чем более высокое качество вы зададите для результирующего изображения, тем менее компактным будет упакованный файл.

Существует три подформата JPEG: обычный, optimized (файлы несколько меньше, но не поддерживаются старыми программами) и progressive (чересстрочное отображение, аналог interlaced в gif). Некоторые приложения позволяют хранить изображение в jpeg в режиме СМΥК и включать в файл обтравочные контуры. Однако использовать JPEG для полиграфических нужд категорически не рекомендуется. Взаимодействие регулярной структуры блоков 8x8 пикселей, получающихся в результате компрессии, с не менее регулярной структурой типографского растра в итоге приводит к образованию муара.

### *Формат PSD*

Формат PSD (Photoshop Document) является внутренним для Photoshop, и позволяет хранить информацию о контурах, каналах, слоях, векторных надписях. Поддерживаются все цветовые модели и любая глубина цвета. Используется сжатие без потерь.

Начиная с версии 3.0, Adobe добавила поддержку слоев и контуров, поэтому

формат версии 2.5 и ранее выделяется в отдельный подформат. Для совместимости с ним в более поздних версиях Photoshop имеется возможность включить режим добавления в файл одного базового слоя, в котором слиты все слои.

### *Формат TIFF*

Формат TIFF (Tagged Image File Format) обеспечивает очень высокое качество изображений. Разработан фирмой Aldus Corporation для хранения графических изображений высокого разрешения.

Основное преимущество TIFF-формата заключается в том, что он поддерживает очень большие изображения, файлы, содержащие несколько изображений (слоев), и разнообразные методы сжатия. Хранит данные в структурированном виде, что позволяет приложениям осуществлять быстрый доступ к различным фрагментам большого изображения.

Спецификация TIFF 5.0 определяет несколько различных подмножеств формата, называемых классами. Класс В поддерживает двухцветные изображения, класс G – изображения с различными оттенками серого, P – изображения с палитрой, R – полноцветные изображения.

Спецификация TIFF 6.0 определяет Baseline TIFF, который должен поддерживать все программы чтения TIFF-файлов. Baseline TIFF включает минимальную поддержку всех четырех классов, имеет большое число различных дополнительных особенностей.

Важным достоинством формата является его переносимость на разные платформы.

### *Формат RAW*

Формат RAW (переводится как – сырой) сравнительно молод, в файлах сохраняется информация, которая непосредственно передается с ПЗС-

матрицы цифровой камеры, минуя все возможные процессы обработки изображения (интерполяцию, повышение резкости, урезание цветности). Для открытия подобного файла в графическом редакторе необходимо проинсталлировать специальное дополнение, которое обычно поставляется вместе с драйверами аппарата. В целом RAW более предпочтителен, чем TIFF, для записи кадров в цифровых камерах, во-первых, потому, что позволяет сохранять изначальную цветовую информацию снимка без использования встроенных алгоритмов, а во-вторых, потому что файлы формата RAW занимают примерно в два раза меньший объем памяти, нежели TIFF.

### *Формат DjVu*

Графический формат DjVu (произносится Дежавю) разработан фирмой AT&T в первую очередь для размещения в Internet отсканированных изображений. Его уникальной особенностью является чрезвычайная компактность при хранении изображений в высоком разрешении (300 DPI и выше). Таким образом, DjVu является уникальным инструментом для открытия Internet-доступа к фондам библиотек.

Для просмотра необходимо либо установить плагин для браузера, либо воспользоваться одной из программ просмотра графических файлов, поддерживающей формат DjVu.

При сжатии текст и контрастные рисунки сохраняются с разрешением 300 dpi, все остальное считается фоном и сохраняется с пониженным разрешением. Это позволяет хорошо сжать электронный документ без потери его читаемости.

Формат поддерживает гиперссылки и многостраничные документы.



## 8. Векторная графика

Программные средства для работы с векторной графикой предназначены, в первую очередь, для создания иллюстраций и в меньшей степени для их обработки.

Принципы векторной графики основаны на отличном от пиксельной графики математическом аппарате и имеют целью построение линейных контуров, составленных из элементарных кривых, описываемых математическими уравнениями.



Векторная графика – вид компьютерной графики, в котором изображение представляется в виде совокупности отдельных объектов, описанных математически.

Как в растровой графике основным элементом изображения является точка, так в векторной графике основным элементом изображения является линия (при этом не важно, прямая это линия или кривая).

Разумеется, в растровой графике тоже существуют линии, но там они рассматриваются как комбинации точек. Для каждой точки линии в растровой графике отводится одна или несколько ячеек памяти (чем больше цветов могут иметь точки, тем больше ячеек им выделяется). Соответственно, чем длиннее растровая линия, тем больше памяти она занимает. В векторной графике объем памяти, занимаемый линией, не зависит от размеров линии, поскольку линия представляется в виде формулы, а точнее говоря, в виде нескольких параметров. Что бы мы ни делали с этой линией, меняются только ее параметры, хранящиеся в ячейках памяти. Количество же ячеек остается неизменным.

Линия — это элементарный объект векторной графики. Все, что есть в векторной иллюстрации, состоит из линий. Простейшие объекты объединяются в более сложные. Например, объект четырехугольник можно рассматривать как четыре связанные линии.

При редактировании элементов векторной графики изменяются параметры прямых и изогнутых линий, описывающих форму этих элементов. Можно переносить элементы, менять их размер, форму и цвет, но это не отразится на качестве их визуального представления. Векторная графика не зависит от разрешения, т.е. может быть показана в разнообразных выходных устройствах с различным разрешением без потери качества.

Векторный формат более компактный, но он совершенно не пригоден для хранения фотографических изображений. В этом формате задавать их математически было бы очень громоздко. А вот рисунки и чертежи гораздо удобнее и практичнее делать именно в векторном виде.

Основными достоинствами векторной графики являются:

- изменение масштаба без потери качества и практически без увеличения размеров исходного файла;
- огромная точность (до сотой доли микрона);
- небольшой размер файла по сравнению с растровыми изображениями;
- высокое качество печати;
- отсутствие проблем с экспортом векторного изображения в растровое;
- возможность редактирования каждого элемента изображения в отдельности.

Основные недостатки:

- сложность экспорта из растрового в векторный формат;

- невозможность применения обширной библиотеки эффектов, используемых при работе с растровыми изображениями.

## 8.1. Математические основы векторной графики

В основе векторной графики лежат математические представления о свойствах геометрических фигур. Как мы сказали выше, простейшим объектом векторной графики является линия. Поэтому в основе векторной графики лежит, прежде всего, математическое представление линии. Рассмотрим несколько видов линий, но начнем с точки.

### *Точка*

Точка на плоскости задается двумя числами  $(x, y)$ , определяющими ее положение относительно начала координат.

### *Прямая линия*

Известно, что для задания прямой линии достаточно двух параметров. Обычно график прямой линии описывается уравнением  $y=kx+b$ . Зная параметры  $k$  и  $b$ , всегда можно нарисовать бесконечную прямую линию в известной системе координат.

### *Кривая второго порядка*

К кривым второго порядка относятся параболы, гиперболы, эллипсы, окружности и другие линии, уравнения которых не содержат степеней выше второй.

Отличаются кривые второго порядка тем, что не имеют точек перегиба. Самая общая формула кривой второго порядка может выглядеть, например, так:

$$x^2 + a_1 y^2 + a_2 xy + a_3 x + a_4 y + a_5 = 0.$$

Как видите, пяти параметров вполне достаточно для описания бесконечной кривой второго порядка.

### *Кривая третьего порядка*

Отличительная особенность этих более сложных кривых состоит в том, что они могут иметь точку перегиба.

Кривые третьего порядка хорошо соответствуют тем линиям, которые мы наблюдаем в живой природе, например линиям изгиба человеческого тела, поэтому в качестве основных объектов векторной графики используют именно такие линии. Все прямые и кривые второго порядка (например, окружности или эллипсы) являются частными случаями кривых третьего порядка.

В общем случае уравнение кривой третьего порядка можно записать так:

$$x^3 + a_1 y^3 + a_2 x^2 y + a_3 xy^2 + a_4 x^2 + a_5 y^2 + a_6 xy + a_7 x + a_8 y + a_9 = 0.$$

Видно, что для записи кривой третьего порядка достаточно девяти параметров.

### *Кривые Безье*

Рисовать кривую третьего порядка по заданным коэффициентам ее уравнения — занятие не слишком интересное. Для упрощения этой процедуры в векторных редакторах применяют не любые кривые третьего порядка, а их особый вид, называемый кривыми Безье.

Вопрос о построении аппроксимирующего многочлена привлек многих математиков. Среди них одну из решающих ролей сыграл выдающийся ученый Сергей Натанович Бернштейн, который окончил Харьковский университет,

учился в Сорбонне, а в начале XX века предложил новое доказательство теоремы Вейерштрасса с помощью теории вероятностей. В этом случае необходимый полином строится в явном виде (не параметрически). Именно данный полином и стал основой сплайновых кривых, в частности NURBS-кривых и кривых Безье.

К сожалению, нашему великому соотечественнику не очень повезло, поскольку чаще известны имена людей, применивших открытия, чем их авторов. То же случилось с кривыми, которые известны всему миру под именем кривых Безье.

Отрезки кривых Безье — это частный случай отрезков кривых третьего порядка. Они описываются не одиннадцатью параметрами, как произвольные отрезки кривых третьего порядка, а лишь восемью, и потому работать с ними удобнее.

Метод построения кривой Безье основан на использовании пары касательных, проведенных к линии в точках ее окончания (рис. 8.1).

Для построения кривой требуется 4 контрольные точки. Но кривая физически проходит только через две из них, они получили название опорных. Одна из этих точек называется начальной (start point), а другая – конечной (end point). Две точки остаются в стороне, они получили название управляющих (control point). Для того чтобы их не «потерять», в программах векторной графики управляющие точки соединяются с опорными точками линией.

Как же изменить форму канонической кривой Безье, чтобы с ее помощью получить огромное многообразие форм, из которых можно составить объект любой сложности?

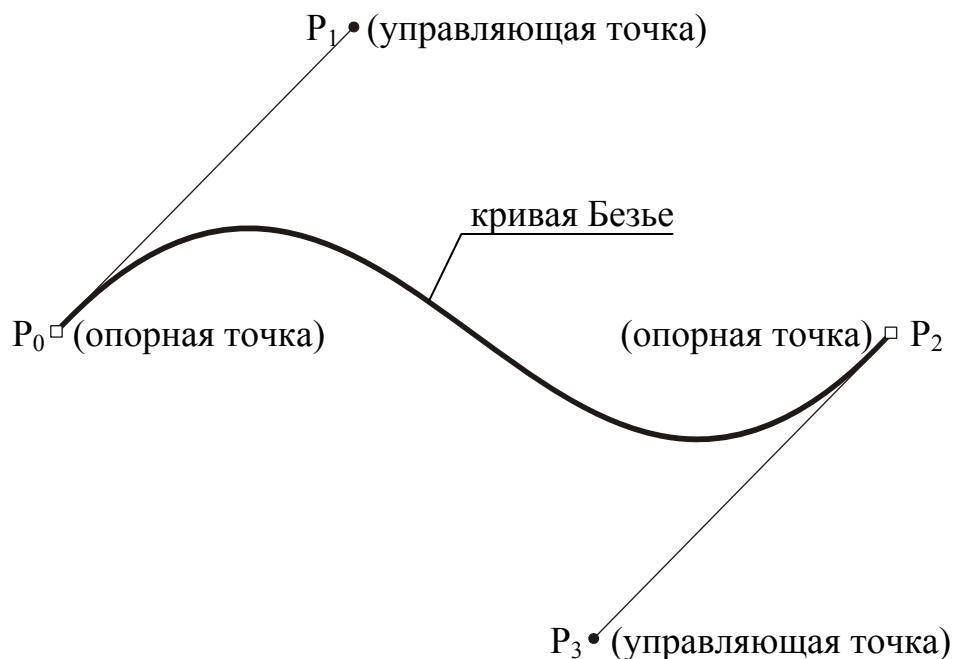


Рис. 8.1. Канонический вид кривой Безье

В программах векторной графики существует единственный способ – это интерактивное перемещение опорных и управляющих точек (рис. 8.2). Если перемещаются начальная или конечная точки, то кривая станет соответствующим образом изменяться (вытягиваться или сжиматься как упругая резинка). Перемещение управляющих точек изменяет кривизну соответствующей части кривой Безье.

Таким образом, с помощью перемещения этих четырех точек получают неограниченное количество форм кривой Безье, которая может быть всего-навсего одним отдельным сегментом сложного векторного контура.

В каждом сегменте можно добавлять опорные точки, которые тоже позволяют изменять форму кривой. Добавление новых опорных точек в пределах одного сегмента кривой не противоречит тому условию, что отдельные кривые соединяются в цепь. Просто кривая Безье добавляется не к концу контура, а размещается внутри уже имеющегося контура.

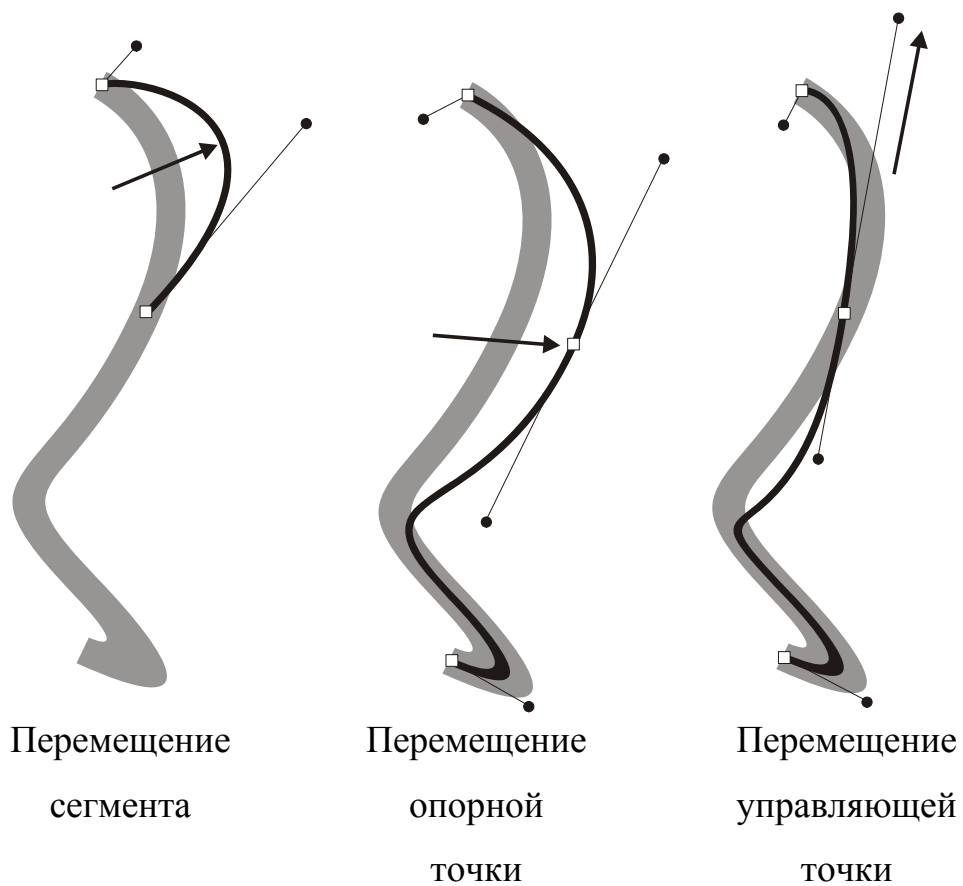


Рис. 8.2. Способы изменения формы сегмента

По сути дела любая векторная конструкция (векторный контур или векторная форма) создается из векторных сегментов, каждый из которых идентичен отдельной элементарной кривой Безье.

Отсюда следует, что между ними образуются соединительные точки, которые иногда называются узлами (например, nodes – в графическом редакторе Corel Draw).

Для поддержки соотношения между элементарными сегментами существуют разные типы опорных точек.

## 8.2. Типы опорных точек

### *Точки перегиба*

Первый тип опорной точки, который соединяет два сегмента, обеспечивает независимость управляющих точек по направлению и длине друг от друга. Такое состояние сегментов называется изгиб (рис. 8.3).

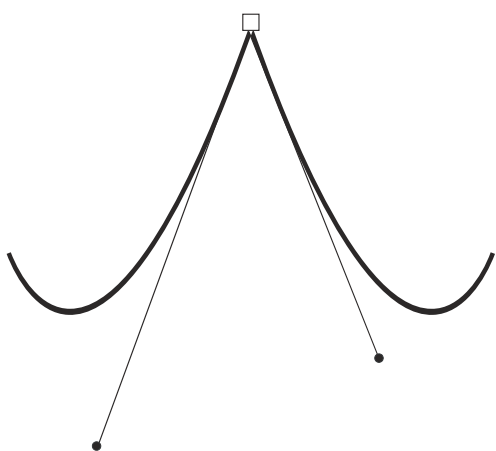


Рис. 8.3. Пример точки перегиба

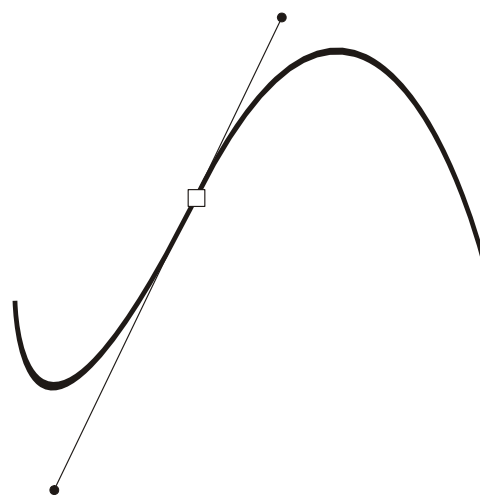


Рис. 8.4. Пример гладкой точки

### *Гладкая опорная точка*

Угловое сочленение сегментов (изгиб) далеко не всегда разумно и выгодно. Например, для создания окружности необходимо обеспечить соединение, которое в черчении и в геометрии называют гладким сопряжением, когда одна кривая плавно переходит в другую. Такое сочленение обеспечивает гладкая опорная точка (smooth) (рис. 8.4).



### *Симметричная опорная точка*

У программы CorelDRAW предусмотрен подвид гладкого сочленения, который называется симметричный узел (symm от слова "symmetrical") (рис. 8.5). Суть его состоит в том, что управляющие линии фиксируют не только по направлению, но и по величине (длина направляющих всегда одинакова). Если одну из них увеличивать или уменьшать, другая будет синхронно повторять это действие.

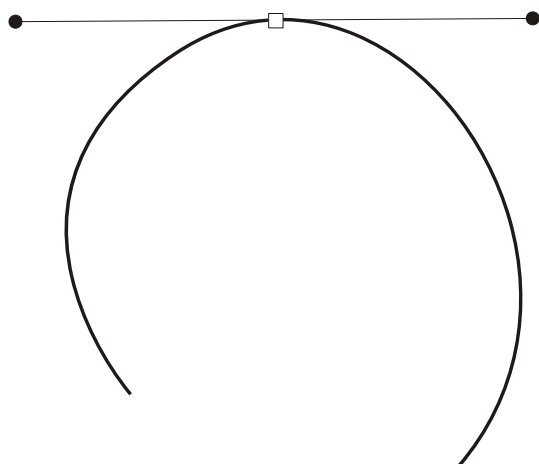


Рис. 8.5. Симметричная опорная точка.

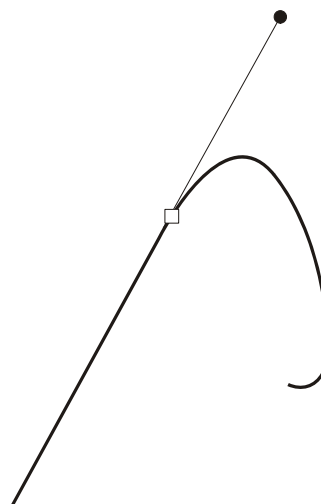


Рис. 8.6. Тангенциальная точка.

### *Тангенциальная опорная точка*

В свою очередь, у программы FreeHand в отдельный вид опорных точек выделен случай гладкого сочленения прямолинейного и криволинейного сегментов (рис. 8.6). Такая точка получила название тангенциальной (connector point). При выделении такая точка обозначается треугольником.

Логика этой точки заключается в следующем: для того чтобы криволинейный сегмент гладко сопрягался с прямой линией, касательная криволиней-

ного сегмента должна совпасть с продолжением прямого сегмента. Поэтому управляющая точка криволинейного сегмента способна двигаться только вдоль этой касательной.

Типы опорных точек можно суммировать в виде табл. 8.1.

Таблица 8.1

Типы опорных точек в различных векторных программах

Тип опорной точки	Adobe Illustrator	Macromedia FreeHand	CoreIDRAW
Угловая	Corner anchor point	Corner point	Cups node
Гладкая	Smooth anchor point	Curve point	Smooth node
Тангенциальная	—	Connector point	—
Симметричная	—	—	Summ node

### 8.3. Основные редакторы векторной графики

#### *Редактор XFig*

Этот графический редактор предназначен для работы в операционной системе Linux и встроен в графическую оболочку KDE.

Стандартный набор геометрических фигур включает в себя: окружность и эллипс с возможностями их создания по двум различным начальным условиям – радиусу или диаметру, замкнутые и незамкнутые линии и ломаные, правильные и неправильные многоугольники, дуги и различные прямоугольники.

### *Редактор Adobe Illustrator*

Ветеран векторной графики – пакет Illustrator – когда-то представлял собой выдающееся достижение в этой области и послужил моделью, которая легла в основу многих программ.

Adobe Illustrator по своей мощности эквивалентен растровому редактору Adobe Photoshop: имеет аналогичный интерфейс, позволяет подключать различные фильтры и эффекты, понимает многие графические форматы, в том числе .cdr (Corel Draw) и .swf (Flash).

Сайт программы: [www.adobe.com/products/illustrator](http://www.adobe.com/products/illustrator).

### *Редактор CorelDRAW*

На сегодняшний день, наиболее распространенный редактор векторной графики. Помимо обработки векторной графики, в этом пакете существует обработчик растровой графики (Corel PHOTO-PAINT), трассировщик изображений (CorelTRACE), редактор шрифтов, подготовки текстур и создания штрихкодов, а также огромные коллекции с изображениями (CorelGallery).

Сайт программы: [www.corel.com](http://www.corel.com).

### *Редактор Macromedia FreeHand*

Был разработан фирмой Aldus, позднее права перекупила Macromedia. Пакет обеспечивает высокое удобство работы с кистями, эффекты выдавливания и трехмерное вращение векторных объектов, может работать в многостраничном режиме. Кроме того, FreeHand полностью интегрирован с Flash и другими продуктами MX-линейки Macromedia.

Сайт программы: [www.macromedia.com/software/freehand](http://www.macromedia.com/software/freehand).

## 8.4. Форматы файлов векторной графики

### *Формат PostScript*

PS (Adobe PostScript) PostScript – язык описания страниц (язык управления лазерными принтерами) фирмы Adobe. Файлы этого формата с расширением .ps или, реже, .ppl получают с помощью функции Print to file графических программ при использовании драйвера PostScript-принтера. Такие файлы содержат в себе сам документ (только то, что располагалось на страницах), все связанные файлы, использованные шрифты, а также другую информацию: цветоделение, дополнительные платы, полутоновый растр для каждой платы, линиатуру растра и прочие данные для выводного устройства. Если файл закрыт правильно – не имеет значения на какой платформе он делался, были ли использованы шрифты True Type или Adobe Type 1. Нужно только учитывать версию языка. Недавно Adobe выпустила PostScript level 4. Тема языка PostScript - отдельный большой разговор. Ниже приводятся еще два формата, непосредственно с ним связанных.

### *Формат EPS*

EPS (Encapsulated PostScript) – упрощенный PostScript. Не может содержать в одном файле более одной страницы, не сохраняет ряд установок для принтера. Как и в файлы печати PostScript, в EPS записывают конечный вариант работы, хотя такие программы как Adobe Illustrator, Photoshop и Macromedia FreeHand могут использовать его как рабочий. EPS используется для передачи векторов и растра в издательские системы, создается почти всеми программами, работающими с графикой.

Изначально EPS разрабатывался как векторный формат, позднее появилась его растровая разновидность – Photoshop EPS.

## *Формат PDF*

Формат PDF (Portable Document Format) предложен фирмой Adobe как независимый от платформы формат, в котором могут быть сохранены иллюстрации (векторные и растровые) и текст, причем со множеством шрифтов и гипертекстовых ссылок. Для работы с этим форматом компания Adobe выпустила пакет Acrobat. Acrobat Distiller переводит в PDF PostScript-файлы, Acrobat Exchange позволяет их редактировать: устанавливать внутренние ссылки, ссылки на внешние звуковые и видеофайлы, Web-ссылки.

Первоначальная задача PDF – передача по сети в сжатом виде проиллюстрированных и отформатированных документов - сегодня значительно расширена. Версия 3 способна сохранять все установки для выводного устройства, записанные в PostScript-файле. Exchange позволяет восстанавливать из файлов PDF PostScript. Таким образом, получается порой серьезный выигрыш во времени, так как PDF можно передавать через e-mail вместо посылного. Кроме того, в PDF можно быстро передавать клиенту полноценные эскизы. PDF позволяет не заботиться о наличии необходимых шрифтов у получателя - все подгружается прямо в файл. По адресу <http://www.adobe.com/acrobat> можно скачать бесплатно распространяемую утилиту Adobe Acrobat Reader, которая позволяет читать документы и распечатывать их на принтере, но не дает возможности создавать или изменять их.

## *Формат AI*

AI (Adobe Illustrator Document) Adobe Illustrator – может содержать в одном файле только одну страницу, имеет маленькое рабочее поле - всего 3x3 метра. В целом несколько уступает FreeHand и CorelDRAW по иллюстративным возможностям, тем не менее, отличается наибольшей

стабильностью и совместимостью с PostScript. AI поддерживают почти все программы так или иначе связанные с векторной графикой. Формат Illustrator является наилучшим посредником при передаче векторов из одной программы в другую, с PC на Macintosh и назад. Кроме того, если вы работаете в основном в Photoshop (Web-дизайн, например), то Illustrator станет лучшим помощником, так как имеет с Photoshop одинаково организованный интерфейс и горячие клавиши. Photoshop понимает форматы Illustrator (AI и EPS) напрямую.

### *Формат FH*

FH (FreeHand Document, последняя цифра в расширении указывает на версию программы) ничем особенным не выделяется. Формат понимает только сам FreeHand, Illustrator для Macintosh и несколько программ от Macromedia. 7-я и 8-я версии имеют полную кроссплатформенную совместимость. Поддерживает многостраничность. Некоторые эффекты FreeHand несовместимы с PostScript.

### *Формат CDR*

CDR (CorelDRAW Document). Формат известен в прошлом низкой устойчивостью и плохой совместимостью файлов, тем не менее, пользоваться CorelDRAW чрезвычайно удобно, он имеет неоспоримое лидерство на платформе PC. Многие программы на PC (FreeHand, Illustrator, PageMaker) могут импортировать файлы CDR. 7-ю и 8-ю версии CorelDRAW можно без натяжек назвать профессиональными. В файлах этих версий применяется компрессия для векторов и раstra отдельно, могут внедряться шрифты, файлы CDR имеют огромное рабочее поле 45x45 метров (этот параметр важен для наружной рекламы); начиная с 4-й версии, поддерживается многостраничность.

## 9. Фрактальная графика

Понятия фрактал и фрактальная геометрия, появившиеся в конце 70-х, с середины 80-х гг. прочно вошли в обиход математиков и программистов. Слово фрактал образовано от латинского *fractus* и в переводе означает состоящий из фрагментов. Оно было предложено Бенуа Мандельбротом в 1975 г. для обозначения нерегулярных, но самоподобных структур, которыми он занимался.

Самыми известными фрактальными объектами являются деревья: от каждой ветки ответвляются меньшие, похожие на нее, от тех — еще меньшие и так далее. По отдельной ветке математическими методами можно проследить свойства всего дерева. Фрактальными свойствами обладают многие природные объекты: снежинка при увеличении оказывается фракталом; по фрактальным алгоритмам растут кристаллы и растения. Если посмотреть на береговую линию моря на картах все более крупного масштаба, то становятся видны все новые изгибы и изломы, похожие на более крупные.

Роль фракталов в машинной графике сегодня достаточно велика. Они приходят на помощь, например, когда требуется, с помощью нескольких коэффициентов, задать линии и поверхности очень сложной формы. С точки зрения машинной графики, фрактальная геометрия незаменима при генерации искусственных облаков, гор, поверхности моря. Фактически найден способ легкого представления сложных неевклидовых объектов, образы которых весьма похожи на природные.

Одним из основных свойств фракталов является самоподобие. В самом простом случае небольшая часть фрактала содержит информацию о всем фрактале.



Фрактал – структура, состоящая из частей, которые в каком-то смысле подобны целому.

Такое определение фракталу дал Мандельброт.

## 9.1. Классификация фракталов

### *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 ломаную-генератор, в соответствующем масштабе. В результате бесконечного повторения этой процедуры, получается геометрический фрактал.

Примером такого фрактального объекта является триадная кривая Кох (рис. 9.1).

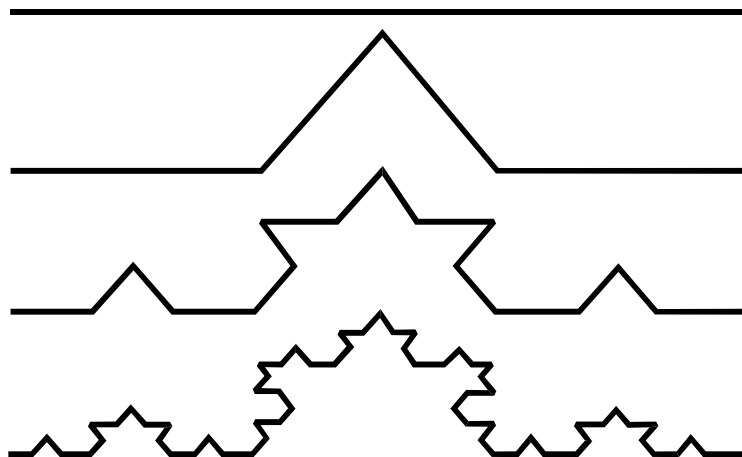


Рис. 9.1. Построение триадной кривой Кох.



Построение кривой начинается с отрезка единичной длины – это нулевое поколение кривой Кох. Далее каждое звено (в нулевом поколении один отрезок) заменяется на образующий элемент. В результате такой замены получается следующее поколение кривой Кох. В первом поколении - это кривая из четырех прямолинейных звеньев, каждое длиной по  $1/3$ . Для получения третьего поколения прodelываются те же действия - каждое звено заменяется на уменьшенный образующий элемент. И так, 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. 9.1 представлены три поколения кривой. При  $n$  стремящемся к бесконечности кривая Кох становится фрактальным объектом.

В машинной графике использование геометрических фракталов необходимо при получении изображений деревьев, кустов, береговой линии. Двумерные геометрические фракталы используются для создания объемных текстур.

### *Алгебраические фракталы*

Это самая крупная группа фракталов. Свое название они получили за то, что их строят на основе алгебраических формул, иногда весьма простых.

Методов получения алгебраических фракталов несколько. Один из методов представляет собой многократный (итерационный) расчет функции  $Z_{n+1}=f(z_n)$ , где  $Z$  – комплексное число, а  $f$  – некая функция. Расчет данной функции продолжается до выполнения определенного условия. И когда это условие выполнится - на экран выводится точка. При этом значение функции для разных точек комплексной плоскости может иметь разное поведение:

- с течением времени стремится к бесконечности;
- стремится к 0;
- принимает несколько фиксированных значений и не выходит за их пределы;
- поведение хаотично, без каких-либо тенденций.

Примером этого вида фракталов является множество Мандельброта (рис. 9.2).

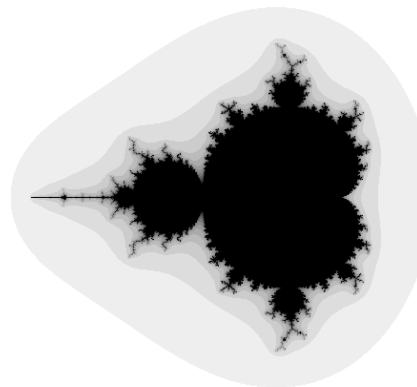


Рис. 9.2. Множество Мандельброта

### *Стохастические фракталы*

Еще одним известным классом фракталов являются стохастические фракталы, которые получаются в том случае, если в итерационном процессе случайным образом менять какие-либо его параметры. При этом получаются объекты очень похожие на природные - несимметричные деревья, изрезанные береговые линии и т.д. (рис. 9.3). Двумерные стохастические фракталы используются при моделировании рельефа местности и поверхности моря.

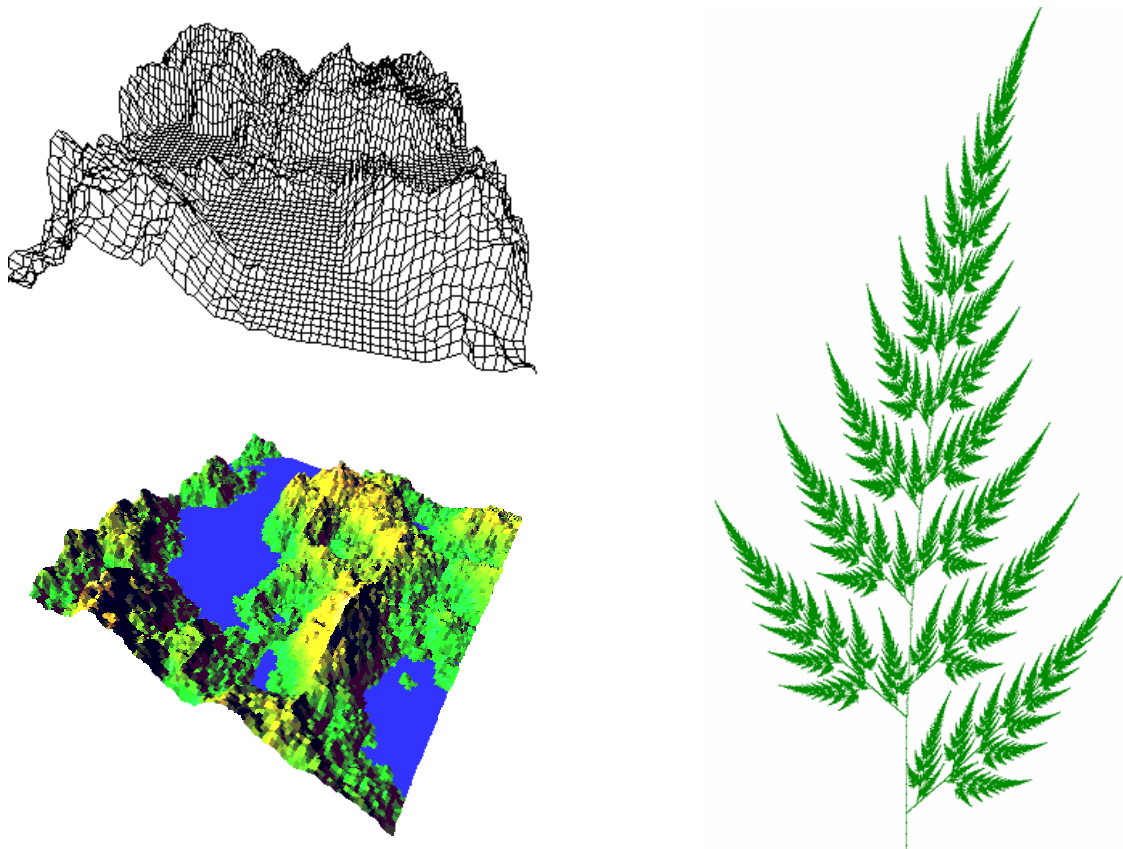


Рис. 9.3. Стохастические фракталы

Способность фрактальной графики моделировать образы живой природы вычислительным путем часто используют для автоматической генерации необычных иллюстраций.

## 10. Трехмерная (3D) графика

Наряду с программами традиционной 2D-графики в последнее десятилетие широчайшее распространение и популярность получили программы 3D-моделирования, анимации и визуализации. При этом такие известные программные решения, как 3D Studio MAX или Maya, являются, по сути, гибридными графическими пакетами. С одной стороны, они предоставляют дизайнеру возможность манипулирования 2D- и 3D-векторными объектами, с другой, результатом работы (финальной визуализации) является пиксельное (растровое) изображение — отдельный кадр или видеоролик.

В силу специфики 3D-моделирования и возможности работать с анимацией такие программы занимают особое место в линейке графических программ. Если попытаться позиционировать пакеты 3D-графики с точки зрения их целевой функции, можно выделить следующие занимаемые ими сегменты рынка.

- Визуальные спецэффекты для кино- и видеоиндустрии.
- Телевизионная реклама.
- Интерактивные игры.
- Промышленный и архитектурный дизайн.
- Научная, медицинская и судебная визуализация.
- Компьютерные тренажеры и обучающие программы.

Следует отметить, что использование пакетов трехмерной графики предъявляет повышенные требования как к аппаратно-программным средствам используемого компьютера, так и к уровню знаний работающего с ними дизайнера.

Все программы 3D-графики, прежде всего, позволяют использовать декартову (картезианскую) систему координат.

### **10.1. Типы пространств**

В зависимости от задачи и этапа работы можно выбирать различные типы пространств и связанных с ними координатных систем.

Чаще всего программы трехмерного моделирования предоставляют следующие варианты пространств.

- Пространство объекта – предназначено для моделирования (описания) формы объекта в его собственной (локальной) системе координат безотносительно того, где он будет размещен на сцене, как ориентирован или масштабирован. У каждого объекта существует своя собственная система координат.
- Мировое пространство используется для размещения объектов на сцене, осуществления аффинных трансформаций (перемещения, поворота и масштабирования объектов), описания освещения сцены, вычисления столкновений между объектами при моделировании динамики их движения и т. п. Это единое пространство для всех объектов сцены.
- Видовое пространство ассоциировано с виртуальным наблюдателем (обычно камерой) или определенной проекцией сцены (например, фронтальным видом) и описывает ту часть сцены, которая доступна для просмотра и работы в видовом окне.
- Экранное пространство – это пространство (плоскость), в котором отображаются аксонометрические или перспективные проекции объектов на плоскость поверхности монитора.

- UVW-параметрическое пространство используется при математическом моделировании сложных кривых и поверхностей (например, NURBS-объектов) или для задания UVW-координат текстурирования поверхностей.

## 10.2. Моделирование объектов

В первом приближении все создаваемые трехмерные объекты можно разделить на геометрические и негеометрические.

- Первые из них используются для моделирования объектов вещественного мира (персонажей, предметов).
- Вторые же применяются для придания сцене реалистичности (например, правильного освещения), для моделирования физических сил, действующих на объекты (например, гравитации или порывов ветра).

Другими словами, геометрические объекты будут присутствовать в визуализированном кадре явно (как кривые или поверхности), а негеометрические — опосредствованно (в виде бликов, теней, ускорений).

### *Геометрические объекты*

Вот краткий перечень типов геометрических объектов:

- Сплайн – гладкая кривая, которая проходит через две или более контрольных точек, управляющих формой сплайна. Два из наиболее общих типов сплайнов - кривые Безье (Bezier curves) и В-сплайны (B-spline curves). Типичным примером сплайнов являются также неоднородные рациональные В-сплайны (Non-Uniform Rational B-Spline - NURBS).

Сплайны состоят из вершин (vertices) и сегментов (segments).

Каждая вершина сплайна имеет касательные векторы (tangents), снабженные на концах управляющими точками, или маркерами (handles). Маркеры касательных векторов управляют кривизной сегментов сплайна при входе в вершину, которой принадлежат касательные векторы, и выходе из нее. В зависимости от свойств касательных векторов различают следующие типы вершин: С изломом (Corner), Сглаженная (Smooos), Безье (Bezier) и Безье с изломом (Bezier Corner).

Сплайны могут служить заготовками для построения поверхностей или их можно использовать в качестве траектории движения.

- Полигональные объекты – это полигональные примитивы, описываемые набором динамически изменяемых параметров (например, длин, радиусов), или полигональные сетки, определяемые как набор граней, ограниченных ребрами, попарно соединяющими вершины.



Polygon (многоугольник) – плоская фигура, ограниченная со всех сторон ломаной линией. Треугольники, то есть простые трехсторонние многоугольники формируют основу, каркас объектов в трехмерной среде.

- Поверхности Безье – это математически гладкие поверхности, описываемые расположением вершин Безье. Эти вершины определяют их кривизну при помощи дополнительных управляющих точек на концах касательных к поверхности векторов.
- NURBS-поверхности – это наиболее универсальный и эффективный способ моделирования неоднородных криволинейных поверхностей. Такие поверхности описываются в особом четырехмерном гомогенном (однородном) пространстве, в котором каждая управляющая вершина,

кроме трех координат  $x$ ,  $y$  и  $z$ , имеет еще и дополнительную весовую характеристику.

- Составные объекты – представляют собой комбинацию двух или более смоделированных заранее заготовок. В зависимости от того, какое именно составное тело создается, заготовками могут служить кривые или объемы (поверхности).
- Системы частиц – это объекты, генерирующие по заданному алгоритму частицы с определенной формой, начальной скоростью, сроком жизни и другими характеристиками. Такие анимационные объекты используются для моделирования дождя, пузырьков газа в жидкости, осколков взрывающихся снарядов и тому подобных образцов объективной реальности.
- Динамические объекты – позволяют моделировать объекты, реагирующие на приложенные к ним внешние силы: пружины и амортизаторы. Используются при моделировании динамики движения объектов.

### *Негеометрические объекты*

- Источники света – используются при моделировании наружного и интерьерного освещения. Различные типы источников реализуют различные алгоритмы распространения света.
- Камеры – позволяют полностью контролировать отображение объектов в плоскости кадра. Важнейшей характеристикой является фокусное расстояние объектива камеры, определяющее ее поле зрения. Оба этих параметра взаимосвязаны и измеряются в миллиметрах и градусах соответственно. Еще одной важной характеристикой камеры являются плоскости отсечения, ограничивающие видимую по глубине (расстоянию от наблюдателя) часть сцены.



- Системы сочленений – это структуры, состоящие из иерархически связанных «костей», описывающих сложную кинематику движения моделируемого объекта (например, человека).
- Искривители пространства – реализуют динамические воздействия внешних сил на объекты, это – своеобразные силовые поля, влияющие на определенные объекты. Примерами могут служить волновая деформация поверхности или ударная волна, разбрасывающая фрагменты объекта в пространстве.

### *Материалы и карты*

Материалы определяют визуальные свойства поверхностей, т. е. описывают то, как поверхность объекта взаимодействует с освещением сцены.

Следующие свойства поверхностей определяют взаимодействие материала со светом:

- цвет;
- прозрачность;
- глянецовость;
- коэффициент преломления.

Важно помнить, что для получения финальной визуализации фотореалистического качества нужно, помимо прочего, очень корректно задавать свойства материала.

### *Текстурирование материалов*

Использование текстурных карт (декоративных узоров, maps) позволяет придать материалам дополнительную реалистичность (например, вид мощенной плиткой мостовой или портмоне из крокодиловой кожи).

В качестве карт могут быть использованы изображения, сохра-

ненные в файлах различных форматов (BMP, TIF, JPG, EPS), либо процедурные текстуры, являющиеся наборами правил быстрого построения нужного узора.



Текстурирование – основной метод моделирования поверхностей наложением на них изображений, называемых текстурой.

Текстура – побитовое отображение поверхностей, отсканированное или нарисованное, придающее поверхности реалистичный вид.

### *Дополнительные свойства материалов*

В зависимости от конкретной программы трехмерного моделирования предлагается различная совокупность дополнительных свойств материалов. Например, пакет 3D Studio MAX позволяет моделировать динамику твердых тел, что приводит в этом случае к необходимости задать коэффициенты упругости материала, силы трения покоя и трения скольжения.

### *Анимация*

Базовым принципом компьютерной анимации (как, собственно говоря, и любой другой) является быстрая смена последовательности кадров, фиксирующих промежуточные фазы движения, перед глазами наблюдателя.

Под движением подразумевается как непосредственно перемещение или поворот объекта в пространстве сцены, так и любое изменение его формы, цвета и т. п. Кадры должны сменять друг друга при просмотре с достаточно высокой скоростью, иначе у наблюдателя не создастся иллюзии непрерывности происходящих изменений.

## *Рендеринг*

Итоговая визуализация (rendering) – заключительный этап работы со сценой.

К этому этапу сцена будет содержать информацию о геометрии объектов, их материалах и освещении. Задача модуля визуализации состоит в том, чтобы вычислить цвет каждого пикселя итогового изображения, основываясь на информации о моделях и выбранном положении виртуального наблюдателя (камеры).

Цвет каждой точки на поверхности отрисовываемого объекта вычисляется исходя из физических свойств материала и освещающего его света.



Рендеринг – процесс интерпретации всего объекта и данных о его освещении для создания, завершенной картины в том виде, в каком она должна выглядеть на перспективе с выбранной точки зрения.

## **Заключение**

В данном пособии рассмотрены только вопросы, относящиеся к теоретической части дисциплины «Компьютерная графика». Работа в прикладных пакетах изучается в процессе лабораторных и практических занятий.

В заключение, авторы выражают признательность рецензентам и редактору за внимательное прочтение рукописи и замечания, способствовавшие улучшению качества предлагаемого пособия.

## Библиографический список

1. Бэйн С. Эффективная работа: CorelDRAW 10. – СПб.: Питер, 2002. – 784 с.
2. Ганзбург А. Периферийные устройства. – СПб.: Питер, 2001. – 448 с.
3. Гук М. Аппаратные средства IBM PC Энциклопедия.– СПб.: Питер, 2000. – 816 с.
4. Казанцев А. В. Основы компьютерной графики: В2ч. Ч 1. Математический аппарат компьютерной графики (электронная версия). – Казань, 2001. – 62 с.
5. Ким Ли. 3D Studio MAX 4/5. Для дизайнера. Искусство трехмерной анимации. – Киев: ООО «ТИД «ДС», 2002. – 848 с.
6. Панкратова Т. FreeHand 9: Учебный курс.– СПб.: Питер, 2001. – 448 с.
7. Петров М. Н., Молочков В. П. Компьютерная графика: Учебник для вузов. – СПб.: Питер, 2003. – 736 с.
8. Пономаренко С. И. Adobe Illustrator 10. – СПб.: БХВ-Петербург, 2003. – 688 с.
9. Пономаренко С. И. Пиксел и вектор. Принципы цифровой графики. – СПб.: БХВ-Петербург, 2002. – 496 с.
10. Тайц А. М., Тайц А. А. Самоучитель Adobe Photoshop 7. – СПб.: БХВ-Петербург, 2002. – 688 с.
11. Шушан Р., Райт Д., Льюис Л. Дизайн и компьютер. – М.: Издательский отдел «Русская Редакция» ТОО «Channel Trading Ltd.», 1997. – 544 с.
12. Яцюк О. Г., Романычева Э. Т. Компьютерные технологии в дизайне. Эффективная реклама. – СПб.: БХВ-Петербург, 2001. – 432 с.

## Глоссарий

**Bitmap** (битовая карта) – способ представления изображения, в котором каждому пикселю соответствует несколько двоичных разрядов, характеризующих его цветность.

**Brightness** (яркость) – характеристика цвета, определяющая интенсивность цвета. Используется в цветовой модели HSB. Измеряется в канделах на квадратный метр ( $\text{кд}/\text{м}^2$ ) и определяется при абсолютно белом изображении на экране (регулировки яркости и контраста настроены на максимум).

**Grayscale** (серая шкала, шкала серого и т.д.) – одноканальное представление изображения с 256 уровнями (градациями) серого цвета. На каждый пиксель изображения отводится 1 байт (8 бит). В этой модели различные оттенки серого задаются с помощью разного процентного содержания двух цветов: черного и белого.

**High Color** – 16-разрядное представление цвета (65 тысяч цветов).

**Hue** (цветовой тон, оттенок) – цвет, получаемый из любого исходного цвета путем незначительного добавления черного, белого или любого другого цвета. Основная характеристика цвета, отличающая его от других цветов, например, оранжевый от синего, фиолетовый от розового и т.д. Используется в модели HSB.

**NURBS** (Non-Uniform Rational Basis-Splines) – математические поверхности, кривизна которых определяется положением контрольных точек, а положение любой точки поверхности вычисляется по математическим формулам. Большинство современных САПР и систем компьютерной анимации поддерживают моделирование с использованием NURBS.

**PostScript** – язык описания страниц (язык управления лазерными принтерами) фирмы Adobe.

**Saturation** (насыщенность) – характеристика цвета, определяющая чистоту цвета, т.е. степень видимого отличия хроматического (цветного) цвета от ахроматического (серого) цвета. Используется в цветовой модели HSB.

**True Color** – 24-разрядное представление цвета (16,7 миллиона цветов).

**Анимация** (animation) – искусственное представление движения в кино, на телевидении или в компьютерной графике, путем отображения последовательности рисунков или кадров с частотой, при которой обеспечивается целостное зрительное восприятие образов.

**Аддитивный цвет** (от англ. add — добавлять, складывать) – получается при соединении лучей света разных цветов.

**Безопасная палитра** – состоит из всех возможных сочетаний 6 оттенков красного, зеленого и синего (RGB) с равномерным шагом. Таких оттенков  $6 \times 6 \times 6 = 216$  (остальные 40 используются в качестве системных). При использовании цветов этой палитры можно быть уверенным в том, что каждый цвет будет передан без искажений. Основная область применения – web-дизайн.

**Векторная графика** – вид компьютерной графики, в котором изображение представляется в виде совокупности отдельных объектов, описанных математически.

**Глубина цвета** (color depth) – количество бит, приходящихся на один пиксель (bpp). Определяет количество бит, или разрядов, с помощью которых составляются коды потенциальных значений тона или цвета.

**Двухмерная (2D) графика** – изображение, имеющее два измерения, то есть лежащее на плоскости.

**Индексная палитра** – табличный набор цветов (не более 256), который сформирован из исходного изображения либо из некоторой цветовой палитры и используется для окрашивания данного изображения с целью уменьшения размера его файла или создания художественного эффекта.

**Интерактивность** (interactive) – свойство программного обеспечения, обеспечивающее реакции со стороны программы в ответ на какие-либо действия пользователя. Реализуется за счет применения графического интерфейса пользователя.

**Интерактивным** называется приложение, результат работы которого зависит от пользователя, который может изменять как результат, так и представление результата. Обработка данных в режиме двухстороннего диалогового взаимодействия пользователя и компьютера, обмена между ними последовательностью запросов (вопросов) и ответов (приглашений) с целью управления процессом обработки. Реализуется с помощью программного обеспечения обладающего свойством интерактивности.

**Канал** (channel) – компьютерная форма отображения каждой составляющей цветовой модели, аналог цветоделенной формы в полиграфическом процессе.

**Комплементарный**, дополнительный (complementary color) цвет – два цвета называются комплементарными (дополняющими друг друга), если при смешении они дают белый цвет.

**Компьютерная графика** – использование вычислительной техники для создания графических изображений, их отображения различными средствами и манипулирования ими.

**Контраст** (contrast) – отношение яркости абсолютно белой и абсолютно

черной точек экрана. Степень тонового различия между областями изображения.

**Кривая Безье** – полиномиальная кривая, задаваемая набором определяющих точек. Кривые Безье записываются в памяти компьютера в виде математических формул, поэтому рисунки, полученные с помощью этих кривых, обеспечивают возможность масштабирования без потери качества изображения.

**Масштабирование объекта** – растяжение объекта вдоль соответствующих осей относительно начала координат.

**Мультимедиа** – область компьютерной графики, связанная с созданием интерактивных приложений (дающих возможность активно влиять на содержимое и направленность), энциклопедий, справочных систем, обучающих программ и интерфейсов к ним.

**Настольная издательская система (Desktop Publishing)** – понятие включает в себя все технические и программные аспекты компьютерной графики.

**Пиксель (pixel)** – мельчайший элемент изображения, воспроизводимый компьютером. Отличительными особенностями пикселя являются его однородность (все пиксели по размеру одинаковы) и неделимость (внутри пикселя не может быть никаких более мелких элементов).

**Полигон** (от англ. polygon – многоугольник) – плоская фигура, ограниченная со всех сторон ломаной линией. Треугольники, то есть простые трехсторонние многоугольники формируют основу, каркас объектов в трехмерной среде.

**Полигональные объекты** – это полигональные примитивы, описываемые набором динамически изменяемых параметров (например, длин, радиусов), или полигональные сетки, определяемые как набор граней, ограниченных



ребрами, попарно соединяющими вершины.

**Полиграфия** – совокупность технических средств для множественного репродуцирования текстового материала и графических изображений.

**Разрешение (resolution)** – количество дискретных элементов на единицу длины.

**Растр** – прямоугольная сетка точек, формирующая изображение на экране компьютера.

**Растровая графика** – формат графического представления объекта в виде множества точек.

**Рендеринг (rendering)** – процесс интерпретации всего объекта и данных о его освещении для создания завершенной картины в том виде, в каком она должна выглядеть на перспективе с выбранной точки зрения.

**Сплайн** – гладкая кривая, которая проходит через две или более контрольных точек, управляющих формой сплайна.

**Субтрактивный цвет** (от англ. subtract — вычитать) – получается вычитанием цвета из общего луча отраженного света.

**Текстура** – побитовое отображение поверхностей, отсканированное или нарисованное, придающее поверхности реалистичный вид.

**Текстурирование** – основной метод моделирования поверхностей наложением на них изображений, называемых текстурой.

**Трехмерная (3D) графика** – построение на компьютере с помощью специальных программ пространственной модели, состоящей из простых и сложных геометрических форм присвоение этой модели фактуры, цвета, степени прозрачности и матовости придание ей и условной камере движения в виртуальном пространстве расстановка в этом пространстве источников

света; и, наконец, просчет выстроенной сцены.

**Формат файла** (file format) – определяет способ организации данных на носителях информации. Форматы файлов определяются разработчиками программ в процессе их создания, затем они, если находят широкое распространение и поддержку, подлежат стандартизации.

**Фрактал** (от латинского fractus – состоящий из фрагментов) – структура, состоящая из частей, которые в каком-то смысле подобны целому.

**Чересстрочный** (interlaced) – в настоящее время существует две наиболее распространенные системы вывода картинки на экран: «сплошная, нечересстрочная» (Non-interlaced) и «чересстрочная». В «чересстрочной» системе сначала отображаются все нечетные горизонтальные строки исходного изображения, а затем - все четные строки этого же изображения. Такой режим позволяет вдвое снизить нагрузку на электронику (частота обновления в два раза меньше), но создает эффект мерцания изображения. В «сплошной» системе наоборот все строки (нечетные и четные) отображаются в том порядке, в каком они идут в исходном изображении. Это дает более качественную картинку, но повышает требования к электронике.

**Цвет** – набор определённых длин волн, отражённых от предмета или пропущенных сквозь прозрачный предмет.

**Цветовая модель** – способ представления большого количества цветов посредством разложения их на простые составляющие.

**Цветовая палитра** – это таблица данных, в которой хранится информация о том, каким кодом закодирован тот или иной цвет. Эта таблица создается и хранится вместе с графическим файлом.

# Предметный указатель

## A

Adobe ...45, 48, 50, 51, 57, 69, 70, 71, 72, 87, 89  
Adobe Acrobat ..... 72  
Adobe Acrobat Distiller ..... 72  
Adobe Acrobat Exchange..... 72  
Adobe Acrobat Reader ..... 72  
Adobe Illustrator ..... 69, 70, 71, 72, 73, 87  
Adobe ImageReady ..... 51  
Adobe PhotoShop..... 48, 52  
AI (Adobe Illustrator Document)..... 72  
aliasing ..... 37  
Animation Shop ..... 51  
anti-aliasing ..... 37, 38

## B

bitmap..... 37, 40, 54  
BMP ..... 54, 85  
bpp (bit per pixel)..... 43, 89  
brainware level (пользовательский уровень  
настольной издательской системы) ..... 8  
brightness (яркость)..... 34, 35, 88

## C

CDR (CorelDRAW Document) ..... 73  
CMYK (цветовая модель) 30, 31, 32, 33, 35, 57  
color depth (глубина цвета) ..... 43, 89  
control point (управляющая точка) ..... 64  
Corel..... 52, 53, 66, 70  
Corel Graphics Suite ..... 53  
Corel PHOTO-PAINT ..... 52, 70  
CorelDRAW..... 66, 70  
CorelTRACE..... 70

## D

Desktop Publishing (настольная издательская  
система).....7, 91  
DjVu.....59  
dpi (dot per inch).....38, 39

## E

end point (конечная точка) .....64  
EPS (Encapsulated PostScript) ..... 71, 73, 85

## F

FH (FreeHand Document) ..... 73

## G

GIF (Graphics Interchange Format).....55, 56, 57  
GIF89a (Graphics Interchange Format).....55  
GIMP (GNU Manipulation Image Program) ..48,  
49  
grayscale .....30, 44, 88

## H

hardware level (аппаратный уровень  
настольной издательской системы).....7  
High Color .....47, 88  
HSB (цветовая модель).....33, 35, 88, 89  
hue (цветовой тон).....33, 34, 88

## I

inch (дюйм) .....38  
input devices (устройства ввода информации)  
.....8

interlacing..... 55, 56

## **J**

JPEG (Joint Photographic Experts Group) 56, 57

## **L**

lpi (line per inch) ..... 38

## **M**

Macromedia..... 53, 69, 70, 71, 73

Macromedia Dreamweaver ..... 53

Macromedia Fireworks ..... 53

Macromedia Flash..... 53, 70

Macromedia FreeHand 68, 69, 70, 71, 72, 73, 87

Maya..... 79

## **N**

NURBS ..... 64, 81, 82, 88

## **O**

output devices (устройства вывода информации)..... 8

## **P**

Paint Shop Pro..... 51

Paintbrush..... 54

PCX..... 54

PDF (Portable Document Format) ..... 72

PhotoImpact ..... 52

pixel (picture element) ..... 37, 38, 91

PNG (Portable Network Graphics) ..... 56

PNG24 (Portable Network Graphics 24 bit) ... 56

PNG8 (Portable Network Graphics 8 bit) ..... 56

polygon..... 82

PostScript ..... 71, 72, 73, 89

ppi (pixel per inch) ..... 38

process, storage and transfer devices

(устройства обработки, хранения и передачи информации).....8

PSD (Photoshop Document) .....51, 57

## **R**

RAW .....58

rendering .....86, 92

resolution (разрешение).....38, 92

RGB (цветовая модель) ..28, 29, 30, 32, 33, 35, 54, 89

## **S**

saturation (насыщенность) .....34, 35, 89

software level (программный уровень настольной издательской системы).....7

spi (sample per inch) .....38

start point (начальная точка) .....64

## **T**

TIFF (Tagged Image File Format).....58, 59

transparency (прозрачность).....56

True Color.....46, 47, 89

## **U**

Ulead Systems.....52

UVW-параметрическое пространство.....81

## **W**

web-дизайн.....7, 56, 73, 89

## **X**

XFig .....69

XPaint .....49

<b>А</b>		<b>И</b>	
аддитивная цветовая модель .....	29	индексная палитра .....	46, 90
алгебраические фракталы.....	76	интерактивное приложение.....	7, 91
анимация .....	85, 89	источники света.....	83
аппаратный уровень настольной издательской системы.....	7, 8	<b>К</b>	
аппроксимирующий многочлен.....	63	камера.....	9, 59, 83, 86
<b>Б</b>		комплементарный цвет.....	33
безопасная палитра .....	47, 89	компьютерная графика ...	1, 2, 4, 19, 39, 53, 87, 89, 90
битовая карта .....	40, 43, 44, 88	конвертирование .....	32
<b>В</b>		кривая Безье.....	63, 64, 65, 66, 81, 91
вектор переноса .....	15, 20, 26	кривая второго порядка .....	62
векторная графика.....	5, 10, 48, 60, 61, 62, 63, 64, 65, 69, 70	кривая третьего порядка.....	63
видовое пространство .....	80	<b>Л</b>	
восьмибитовое изображение .....	44	левосторонняя система координат .....	24
вращение точки .....	17, 18	<b>М</b>	
<b>Г</b>		масштабирование объектов.....	16, 17
геометрические объекты.....	81	масштабный множитель .....	19
геометрические фракталы .....	75	матрица масштабирования .....	17, 21
гладкая опорная точка .....	67	матрица результирующего поворота и масштабирования .....	18
глубина цвета.....	42, 43, 44, 56, 57, 89	матричная операция.....	19
графопостроитель.....	4, 10	матричное преобразование .....	20, 22
<b>Д</b>		матричное произведение .....	19, 20
двумерные матричные преобразования .....	15	мировое пространство .....	80
декартова система координат.....	11	множество Мандельброта .....	77
декартовы координаты точек .....	13	моделирование объектов .....	81
динамические объекты .....	83	монитор .....	5, 28, 30, 32, 35, 36, 39, 80
дисплей.....	4	мультимедиа .....	7, 91
дюйм .....	38, 39	<b>Н</b>	
		настольная издательская система.....	7
		насыщенность цвета .....	34

негеометрические объекты..... 83

## О

однородные координаты..... 18, 19, 22, 24

операция переноса..... 15, 18, 20, 26

опорная точка ..... 65, 66, 67, 68, 69

ортогональная система координат..... 12, 27

ось системы координат ..... 11

отражаемые цвета..... 30

## П

перцепционная цветовая модель..... 32

пиксель .....37, 38, 40, 43, 44, 56, 57, 88, 89, 91

пиксельная (растровая) графика ..... 10, 60

плоттер ..... 4

поверхности Безье ..... 82

полигональные объекты ..... 82, 91

полноцветное изображение ..... 44, 45, 58

пользовательский уровень настольной  
издательской системы..... 8, 10

последовательный перенос..... 20

правосторонняя система координат ..... 24

преобразование координат точек..... 15

программное обеспечение ..... 5, 10, 54, 90

программный уровень настольной  
издательской системы..... 7, 10

проекция однородных координат ..... 19

пространственная модель ..... 5, 92

пространство объекта ..... 80

## Р

радиус-вектор ..... 13, 17

разрешение..... 38, 39, 58, 59, 61, 92

разрешение изображения..... 39

разрешение принтера ..... 39

разрешение экрана ..... 39

растеризация ..... 40

растр .....36, 57, 71, 73, 92

растровая (пиксельная) графика .....5, 36, 60

растровая карта.....37

редактор ..... 7, 38, 48, 49, 50, 51, 52, 53, 69, 70

результующая матрица..... 18, 27

результующее преобразование .....22

рендеринг .....86, 92

## С

серая шкала.....44, 45, 88

симметричная опорная точка .....68

система координат ..... 11, 12, 13, 29, 80

системы сочленений .....84

системы частиц.....83

сканер .....9, 32

составные объекты .....83

сплайн.....81, 92

сплайновая кривая.....64

стохастические фракталы.....77, 78

субтрактивные цвета.....30, 31

суммарный перенос .....20, 27

## Т

таблица квантования.....42, 43

тангенциальная опорная точка .....68

текстура.....85, 92

текстурирование.....84, 85, 92

тоновое изображение .....42, 43, 44

точка перегиба .....63, 67

трехмерная графика .....10, 79

трехмерные матричные преобразования .....24

триадная кривая Кох .....75

## У

уравнение прямой .....13, 14

устройства ввода информации.....8

устройства вывода информации .....8, 9

устройства обработки, хранения и передачи информации ..... 8, 9

## **Ф**

физический размер изображения ..... 39

фиксированная палитра ..... 47, 48

формат файла ..... 49, 53, 71, 93

фрактал ..... 74, 75, 76, 77, 93

фрактальная графика ..... 5, 10, 48, 78

## **Ц**

цветовая модель...28, 29, 30, 31, 32, 33, 35, 57, 88, 89, 90, 93

цветовая модель CMYK ..30, 31, 32, 33, 35, 57

цветовая модель HSB.....33, 35, 88, 89

цветовая модель RGB28, 29, 30, 32, 33, 35, 54, 89

цветовая палитра .....46, 55, 93

цветовой охват.....32

цветовой тон .....33, 34, 35, 88

## **Э**

экранное пространство .....80

## **Я**

яркость цвета .....34

## Оглавление:

Предисловие.....	3
1. Введение.....	4
2. Виды компьютерной графики.....	5
3. Области применения компьютерной графики .....	6
4. Настольные издательские системы .....	7
4.1. Аппаратный уровень.....	8
4.2. Программный уровень.....	10
4.3. Пользовательский уровень.....	10
5. Система координат и типы преобразования графической информации...	11
5.1. Декартова система координат.....	11
5.2. Двумерные матричные преобразования .....	15
5.3. Однородные координаты и матричное представление двумерных преобразований .....	18
5.4. Трехмерные матричные преобразования .....	24
6. Цветовые модели.....	28
6.1. Цветовая модель RGB .....	28
6.2. Цветовая модель CMYK.....	30
6.3. Цветовая модель HSB.....	33
7. Растровая (пиксельная) графика.....	36
7.1. Разрешение           растровой   графики.....	38



7.2. Виды разрешения .....	39
7.3. Кодирование изображения.....	40
7.4. Глубина цвета .....	42
7.5. Цветовые палитры .....	46
7.6. Основные редакторы растровой графики.....	48
7.7. Форматы файлов растровой графики.....	53
8. Векторная графика .....	60
8.1. Математические основы векторной графики.....	62
8.2. Типы опорных точек.....	67
8.3. Основные редакторы векторной графики .....	69
8.4. Форматы файлов векторной графики .....	71
9. Фрактальная графика .....	74
9.1. Классификация фракталов .....	75
10. Трехмерная (3D) графика .....	79
10.1. Типы пространств .....	80
10.2. Моделирование объектов.....	81
Заключение .....	86
Библиографический список .....	87
Глоссарий .....	88
Предметный указатель.....	94

Виталий Евгеньевич Васильев

Алексей Валентинович Морозов

## Компьютерная графика

Учебное пособие

Редактор Т. В. Шабанова

Сводный темплан 2005 г.

Лицензия ЛР № 020308 от 14.02.97

Санитарно-эпидемиологическое заключение № 78.01.07.953.П.005641.11.03 от 21.11.2003 г.

---

Подписано в печать

. Формат 60x84 1/16

Б. кн.- журн. П. л.6,25. Б. л.3,125. РТП РИО СЗТУ.

Тираж 100. Заказ

---

Северо-Западный государственный заочный технический университет

РИО СЗТУ, член Издательско-полиграфической ассоциации университетов  
России

191186, Санкт-Петербург, ул. Миллионная, д. 5