

Министерство образования Республики Беларусь  
Учреждение образования «Витебский государственный университет  
имени П.М. Машерова»

# ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

*Курс лекций по дисциплине  
«Компьютерная графика и мультимедиа»*



Витебск

2010

Репозиторий ВГУ

## СОДЕРЖАНИЕ

<u>Введение.....</u>	<u>4</u>
<u>Тема 1. Основные понятия компьютерной графики.....</u>	<u>5</u>
<u>Растровая графика.....</u>	<u>7</u>
<u>Векторная графика.....</u>	<u>14</u>
<u>Фрактальная графика.....</u>	<u>19</u>
<u>Тема 2. Средства описания цвета.....</u>	<u>25</u>
<u>Цветовые модели.....</u>	<u>25</u>
<u>Аддитивные цветовые модели.....</u>	<u>28</u>
<u>Субтрактивные цветовые модели.....</u>	<u>32</u>
<u>Перцепционные цветовые модели.....</u>	<u>35</u>
<u>Тема 3. Типы графических изображений.....</u>	<u>38</u>
<u>Тема 4. Общая схема цифровой обработки изображений.....</u>	<u>42</u>
<u>Получение исходного, «сырого» изображения.....</u>	<u>42</u>
<u>Фильтрация изображения.....</u>	<u>43</u>
<u>Форматирование и индексирование изображения.....</u>	<u>46</u>
<u>Сжатие.....</u>	<u>47</u>
<u>Тема 5. Растровый графический редактор Adobe PhotoShop.....</u>	<u>48</u>
<u>Подготовка изображения для печати.....</u>	<u>53</u>
<u>Тема 6. Художественное оформление текста в векторных редакторах.....</u>	<u>55</u>
<u>Работа в векторном редакторе Adobe Illustrator.....</u>	<u>56</u>
<u>Набор текста.....</u>	<u>56</u>
<u>Оформление текста, не преобразованного в кривые.....</u>	<u>57</u>
<u>Асимметричное масштабирование.....</u>	<u>57</u>
<u>Тень.....</u>	<u>57</u>
<u>Отражение в воде.....</u>	<u>57</u>
<u>Полые буквы и оконтуривание снаружи.....</u>	<u>58</u>
<u>Текст в оболочке.....</u>	<u>58</u>
<u>Текст вдоль линии.....</u>	<u>59</u>
<u>Цветовые эффекты.....</u>	<u>59</u>
<u>Дополнительные возможности редактора.....</u>	<u>59</u>
<u>Оформление преобразованного в кривые текста.....</u>	<u>59</u>
<u>Цветовые эффекты.....</u>	<u>59</u>
<u>Скручивание.....</u>	<u>60</u>
<u>Фильтры.....</u>	<u>61</u>
<u>Тема 7. Графика в Web-дизайне.....</u>	<u>63</u>
<u>Литература.....</u>	<u>66</u>

## Введение

Компьютерная графика сформировалась как наука об аппаратном и программном обеспечении для подготовки, преобразования, хранения и воспроизведения графической информации с помощью ЭВМ. Представление данных на мониторе компьютера в графическом виде впервые было реализовано в середине 50-х годов для больших ЭВМ, применявшихся в научных и военных исследованиях. С тех пор графический способ отображения данных стал неотъемлемой принадлежностью подавляющего числа компьютерных систем и в настоящее время компьютерная графика используется во всех областях деятельности человека. Хотя компьютерная графика служит всего лишь инструментом, ее структура и методы основаны на передовых достижениях фундаментальных и прикладных наук: аналитической и начертательной геометрии, программирования для ЭВМ, методов вычислительной математики, статистики и множества других. Поэтому компьютерная графика является одной из наиболее бурно развивающихся отраслей информатики.

Целью преподавания дисциплины «Компьютерная графика и мультимедиа» для студентов специальности «Математика. Информатика» дневной и заочной формы обучения является ознакомление с основами современной компьютерной графики, освоение базовых понятий и методов компьютерной графики, изучение популярных графических программ и инструментов, позволяющих выполнять действия от самых простых операций по созданию графических примитивов до сложнейших дизайнерских решений любого уровня.

Предметом курса «Компьютерная графика и мультимедиа» являются методы и средства создания и обработки графических изображений с помощью современных программно-аппаратных вычислительных комплексов.

В процессе изучения дисциплины студенты приобретают практические навыки подготовки изображений к публикации, в том числе в электронном виде для размещения в сети Интернет, знакомятся с различными сферами применения методов и средств компьютерной графики в современном обществе, а также средствами электронно-вычислительной техники, применяемой для создания и обработки графических изображений.

Лекция 1. Основные понятия компьютерной графики. Растровая графика. Векторная графика. Фрактальная графика	6
Лекция 2. Средства описания цвета. Цветовые модели	6

Лекция 3. Типы графических изображений	2
Лекция 4. Общая схема цифровой обработки изображений. Фильтрация изображения. Сжатие	6
Лекция 5. Растровый графический редактор Adobe PhotoShop	4
Лекция 6. Художественное оформление текста в векторных редакторах	2
Лекция 7. Основы графического дизайна. Графика в Web-дизайне	4
Всего:	30

В курсе лекций представлены общие сведения о компьютерной графике, теоретические и практико-ориентированные материалы по изучению графических компьютерных программ PhotoShop и Adobe Illustrator. Материал содержит графические иллюстрации, поясняющие работу с описываемыми приложениями. Основными целями данной разработки являются знакомство будущих учителей информатики с теоретическими основами компьютерной графики и обучение их приемам создания, коррекции и оптимизации графических изображений. Материалы пособия могут быть полезны также учителям-предметникам для создания наглядных и дидактических материалов на основе векторной и растровой графики, используемых в образовательной деятельности.

## **Тема 1. Основные понятия компьютерной графики**

Современный мир немалозначим без компьютерной обработки графической информации. Обложки журналов, рекламная печатная продукция, газеты, рекламные ролики по телевидению, фильмы, презентации, компакт-диски, Интернет – это всё области применения компьютерной графики, основу которой составляет работа с цифровыми изображениями.

*Компьютерная графика* – это наука, предметом изучения которой является создание, хранение и обработка моделей и их изображений с помощью ЭВМ. В том случае, если пользователь может управлять характеристиками объектов, говорят об интерактивной компьютерной графике.

*Цифровым* может быть названо изображение, созданное с использованием компьютерной программы, либо изображение, преобразованное в электронную информацию для того, чтобы просматривать, редактировать и управлять им на экране компьютера.

Цифровое изображение можно создать, например, с помощью сканера или цифрового фотоаппарата, а затем отредактировать в программе для обработки изображений. Существуют также

специальные программы, позволяющие создавать цифровой рисунок с нуля, выбрав для этого необходимые электронные кисти и краску.

Устройства, преобразующие графические изображения в цифровую форму, называются *оцифровывающими* (сканеры, цифровые фотоаппараты).

Выделяют несколько направлений компьютерной графики:

- двухмерная графика;
- полиграфия;
- web-дизайн;
- мультимедиа;
- 3D-графика и компьютерная анимация;
- видеомонтаж;
- САПР и деловая графика.

Каждое из перечисленных направлений имеет свои отличительные особенности и технологические тонкости. Для каждого из них создано своё программное обеспечение, включающее разнообразные специальные программы – *графические редакторы*. Вне зависимости от направления применения, графический редактор должен включать в себя:

- инструменты рисования на компьютере;
- библиотеку готовых изображений;
- набор шрифтов;
- набор спецэффектов.

Кроме того, он должен быть совместим с другими графическими программами.

В зависимости от способа формирования изображений компьютерную графику принято подразделять на *растровую*, *векторную* и *фрактальную*.

Отдельным объектом считается трёхмерная графика, изучающая приёмы и методы построения объёмных моделей объектов в виртуальном пространстве. Как правило, в ней сочетаются векторный и растровый способы формирования изображений.

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

На стыке компьютерных, телевизионных и кино-технологий зародилась и стремительно развивается сравнительно новая область компьютерной графики – *анимация*. Заметное место в компьютерной графике отведено увлечениям. Появилось даже такое понятие как механизм графического представления данных. Рынок игровых программ имеет оборот в десятки миллиардов долларов и часто

инициализирует очередной этап совершенствования графики и анимации.

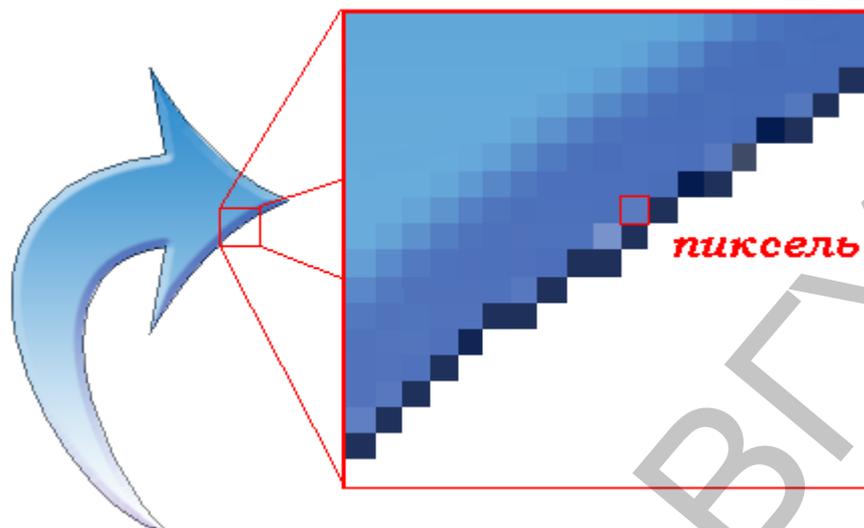
### **Растровая графика**

Существует несколько моделей представления изображений в компьютере. Все их можно свести к двум основным.

Первая модель кодирования графических изображений заключается в том, что изображение раскладывается на точки очень маленького размера, называемые пикселями. *Пиксель* (Pixel) – сокращение от Picture Element (Элемент рисунка). Изображение создается примерно так же, как в мозаике, и представляет собой множество цветных точек, расположенных на сетке. Сама сетка называется *растровой картой* (bitmap), а пиксель – её элемент.

Простейший случай – кодирование чёрно-белых графических изображений. Точка белого цвета кодируется нулём, точка чёрного цвета – единицей. Каждая точка в этом случае представляется одним битом.

Цветные изображения состояются из точек различных цветов. Каждая точка изображения в этом случае представляется несколькими битами. Количество цветов и оттенков, которые может принимать каждая точка изображения, характеризует качество изображения. Чем большим количеством оттенков характеризуется изображения, тем большее количество разрядов требуется для их описания. В зависимости от количества бит, отводимых для каждой точки, в изображении может присутствовать от двух до десятков миллионов цветов. Объём информации, описывающий цвет каждой точки, определяет *глубину цвета*. Для цветных рисунков требуется большой объём памяти. Если информация о каждой точке рисунка занимает 32 бита, то в полноцветном режиме можно представить до 16,7 миллионов цветов. Таким образом, чем качественнее изображение, тем больше размер файла.



**Рисунок 1.** Пиксель – цветная точка

Изображения, закодированные таким образом, называются *растрами*, или *растровыми изображениями*.

Растровое изображение характеризуется глубиной цвета, размером и разрешением. *Размер* описывает высоту и ширину изображения и измеряется в пикселях или других величинах. Самый простой способ измерения размера – в пикселях. Но если размер измеряется, например, в дюймах, то реальный размер изображения зависит от разрешения. *Разрешение* – это плотность размещения пикселей, формирующих изображение, или количество точек на единицу длины. Разрешение измеряется обычно в количестве точек на дюйм – dpi (Dot Per Inch) или пикселей на дюйм – ppi (Pixel Per Inch).

Поскольку основой растрового представления графики является пиксель (точка) с указанием цвета, то при описании рисунка приходится указывать цвет каждой точки, как изображения, так и фона. Изображение представляется в виде большого количества точек – чем их больше, тем визуально качественнее изображение и больше размер файла. Т.е. одна и та же картинка может быть представлена с лучшим или худшим качеством в соответствии с разрешением.



**Рисунок 2.** Растровое изображение в уменьшенном и увеличенном масштабе

Растровая графика зависит от разрешения, поскольку информация, описывающая изображение, прикреплена к сетке определенного размера. При редактировании растровой графики, качество ее представления может измениться. В частности, изменение размеров растровой графики может привести к «разлохмачиванию» краев изображения, поскольку пиксели будут перераспределяться на сетке. Вывод растровой графики на устройства с более низким разрешением, чем разрешение самого изображения, понизит его качество.

Средствами растровой графики принято иллюстрировать работы, требующие высокой точности в передаче цветов и полутонов. Однако размеры файлов растровых иллюстраций стремительно растут с увеличением разрешения. Например, фотография на рисунке 3 размером 3,2×3,2 дюйма (960×960 пикселей) и глубиной цвета 24 бита занимает около 156 килобайт, если использовать разрешение 300 dpi. Эта же фотография (рисунок 4) при разрешении 600 dpi занимает уже свыше 2 мегабайт.



**Рисунок 3.** Фотография размером около 156 КВ



**Рисунок 4.** Фотография размером более 2 МВ

Растровое представление обычно используют для изображений фотографического типа с большим количеством деталей или оттенков.

К сожалению, масштабирование таких картинок в любую сторону обычно ухудшает качество. При уменьшении количества точек теряются мелкие детали и деформируются надписи (правда, это может быть не так заметно при уменьшении визуальных размеров самой картинки – т.е. сохранении разрешения). Добавление пикселей приводит к ухудшению резкости и яркости изображения, так как новым точкам приходится давать оттенки, средние между двумя и более граничащими цветами.

Для растровых изображений понятие разрешения имеет особую важность. Различают

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

*Разрешение оригинала* измеряется в количестве точек на дюйм (или пикселей на дюйм) и зависит от требований к качеству изображения и размеру файла, способу оцифровки или методу создания исходной иллюстрации, избранному формату файла и другим параметрам. В общем случае действует правило: чем выше требования к качеству, тем выше должно быть разрешение оригинала.

Размер пикселя варьируется в зависимости от выбранного *экранного разрешения* (из диапазона стандартных значений), разрешения оригинала и масштаба отображения. Мониторы для обработки изображений с диагональю 20-21 дюйм (профессионального класса), как правило, обеспечивают стандартные экранные разрешения 640×480, 800×600; 1024×768; 1280×1024; 1600×1200; 1600×1280; 1920×1200; 1920×1600 точек. Расстояние между соседними точками люминофора у качественного монитора составляет 0,22-0,25 мм.

Размер точки растрового изображения как на твёрдой копии (бумага, плёнка), так и на экране зависит от применённого метода и параметров растривания оригинала. При растривании на оригинал как бы накладывается сетка линий, ячейки которой образуют элемент растра. Частота сетки растра изменяется числом линий на дюйм (lines per inch – lpi) и называется *линиатурой*.

Для экранной копии достаточно разрешения 72 dpi, для распечатки на цветном или лазерном принтере 150-200 dpi, для вывода на фотоэкспонирующем устройстве 200-300 dpi. Установлено эмпирическое правило, что при распечатке величина расширения оригинала должна быть в 1,5 раза больше, чем линиатура растра устройства вывода. В случае, если твёрдая копия будет увеличена по сравнению с оригиналом, эти величины следует умножить на коэффициент масштабирования.

Размер точки растра рассчитывается для каждого элемента и зависит от интенсивности тона в данной ячейке. Чем больше интенсивность, тем плотнее заполняется элемент растра. То есть, если в ячейку попал абсолютно чёрный цвет, размер точки растра совпадёт с размером элемента растра. В этом случае говорят о 100% заполняемости. Для абсолютно белого цвета значение заполняемости составит 0%. На практике заполняемость элемента на отпечатке обычно составляет от 3% до 98%. При этом все точки растра имеют одинаковую оптическую плотность, в идеале приближающуюся к абсолютно чёрному цвету. Иллюзия более тёмного тона создаётся за счёт увеличения размеров точек и, как следствие, сокращения пробельного поля между ними при одинаковом расстоянии между центрами элементов растра. Такой метод называют *растрированием с амплитудной модуляцией (АМ)*.

Существует и метод *растрирования с частотной модуляцией (ЧМ)*, когда интенсивность тона регулируется изменением расстояния между соседними точками одинакового размера. Таким образом, при частотно-модулированном растрировании в ячейках растра с разной интенсивностью тона находится разное число точек. Изображения, растрированные ЧМ-методом, выглядят более качественно, так как размер точек минимален и, во всяком случае, существенно меньше, чем средний размер точки при АМ-растрировании.

Ещё более повышает качество изображения разновидность ЧМ-метода, называемая стохастическим растрированием. В этом случае рассчитывается число точек, необходимое для отображения требуемой интенсивности тона в ячейке растра. Затем эти точки располагаются внутри ячейки на расстояниях, вычисленных квазислучайным методом (на самом деле используется специальный математический алгоритм). То есть регуляторная структура растра внутри ячейки, как и на изображении в целом, вообще отсутствует.

Поэтому при стохастическом ЧМ-растрировании теряет смысл понятие линиатуры растра, имеет значение лишь разрешающая способность устройства вывода. Такой способ требует больших затрат вычислительных ресурсов высокой точности полиграфического оборудования; он применяется в основном для художественных работ, при печати с числом красок, превышающим четыре.

Интенсивность тона (так называемую светлоту) принято подразделять на 256 уровней. Большее число градаций не воспринимается зрением человека и является избыточным. Меньшее число ухудшает восприятие изображения (минимально допустимым для качественной полутоновой иллюстрации принято значение 150 уровней). Нетрудно подсчитать, что для воспроизведения 256 уровней тона достаточно иметь размер ячейки растра  $256=16\times 16$  точек.

Между разрешением оригинала, частотой растра и градацией уровней существует зависимость, описываемая формулой:

$$N = \left( \frac{dpi}{lpi} \right)^2 + 1 \quad lpi = \frac{dpi}{\sqrt{N - 1}}$$

Где  $N$  – число градаций уровней тона (оттенков),  $dpi$  – разрешение устройства вывода (отображения),  $lpi$  – линиятура растра. Единица в формуле соответствует абсолютно белому цвету, когда ячейка растра вообще не заполнена.

При выводе копии изображения на принтере или полиграфическом оборудовании линиятуру растра выбирают, исходя из компромисса между требуемым качеством, возможностями аппаратуры и параметрами печатных материалов. Для лазерных принтеров рекомендуемая линиятура составляет 65-100 lpi, для газетного производства 65-85 lpi, для книжно-журнального – 85-133 lpi, для художественных и рекламных работ – 133-300 lpi.

При печати изображений с наложением растров друг на друга, например многоцветных, каждый последующий растр поворачивается на определённый угол. Традиционными для цветной печати считаются углы поворота: 105° для голубой печатной формы, 75° для пурпурной, 90° для жёлтой и 45° для чёрной. При этом ячейка растра становится косоугольной, и для воспроизведения 256 градаций тона с линиятурой 150 lpi уже недостаточно расширения  $16 \times 150 = 2400 dpi$ . Поэтому для фотоэкспонирующих устройств профессионального класса принято минимальное стандартное разрешение 2540 dpi, обеспечивающее качественное растривание при разных углах поворота растра. Таким образом, коэффициент, учитывающий поправку на угол поворота растра, для цветных изображений составляет 1,06.

Одним из недостатков растровой графики является та называемая *пикселизация* изображений при их увеличении (если не приняты специальные меры). Раз в оригинале присутствует определённое количество точек, то при большем масштабе увеличивается и их размер, становятся заметны элементы растра. Для противодействия пикселизации принято заранее оцифровывать оригинал с разрешением, достаточным для качественной визуализации при масштабировании. Другой приём состоит в применении стохастического растра, позволяющего уменьшить эффект пикселизации в определённых пределах. Наконец, при масштабировании используют метод интерполяции, когда увеличение размера иллюстрации происходит не за счёт масштабирования точек, а путём добавления необходимого числа промежуточных точек.

Достоинства растровой графики.

- Техническая реализуемость (автоматизация) ввода (оцифровки) изобразительной информации.
- Редактирование фотореалистичных изображений (сцены природы, фотографии людей), возможность цветокоррекции, ретуши и создания спецэффектов на базе цифровых изображений.
- Создание коллажей, виньеток, фотомонтажа, живописных эффектов (туман, дымка, перспектива, размытость и так далее).
- Создание текстовых и фоновых эффектов в мультимедиа-продуктах.
- Изменение количества цветов изображения.
- Форматы файлов, предназначенные для сохранения точечных изображений, являются стандартными.

Недостатки растровой графики.

- Фиксированное положение пикселей.
- Необходимость на начальном этапе работы принципиального решения о разрешении (resolution) – количестве пикселей на дюйм изображения – и о глубине цвета – количестве цветовых битов на пиксель.
- Зависимость размера файла и качества изображения от пространственного и цветового разрешения (объем файла точечной графики = площадь изображения × разрешение × глубина цвета).
- Зависимость качества изображения от качества устройств вывода.
- При повороте на небольшой угол изображения, например, с четкими вертикальными линиями четкие линии превращаются в четкие «ступеньки».
- Невозможность увеличения рисунка для рассмотрения деталей (пикселизация визуально искажает иллюстрацию и делает ее грубой).
- Текст в растровой программе до недавнего времени представлял проблему; последние версии профессиональных растровых редакторов поддерживают возможность помещения текста в отдельные слои, которые сохраняют векторные свойства текста.

### **Векторная графика**

Если в растровой графике базовым элементом изображения является точка, то в векторной графике – *объект*. Объект описывается математически как единый, и потому объем данных для отображения объекта средствами векторной графики существенно меньше, чем в растровой графике.

С помощью комбинации нескольких объектов можно создавать новый объект. Объекты можно группировать для дальнейшего редактирования группы как единого объекта. Независимо от внешнего вида, все объекты векторной графики имеют ряд общих характеристик.

*Линия* – элементарный объект векторной графики. Как и любой объект, линия обладает свойствами: формой (прямая, кривая), толщиной, цветом, начертанием (сплошная, пунктирная). Замкнутые линии приобретают свойство заполнения – *заливка*. Охватываемое ими пространство может быть заполнено другими объектами (текстуры, карты) или выбранным цветом. Простейшая незамкнутая линия или сегмент ограничена двумя точками, именуемыми *узлами*. Узлы также имеют свойства, параметры которых влияют на форму конца линии и характер сопряжения с другими объектами.

Сегменты объекта образуют *контур*, который имеет толщину и цвет (их тоже можно изменять). Различают *замкнутые* и *разомкнутые* контуры.

У одного объекта не может быть различных заливок или соединительных линий различной толщины и разного цвета. Для создания сложных изображений требуется использовать множество объектов.

Все прочие объекты векторной графики состояются из линий. Например, куб можно составить из шести связанных прямоугольников, каждый из которых, в свою очередь, образован четырьмя связанными линиями. Можно представить куб как двенадцать связанных линий, образующих рёбра.

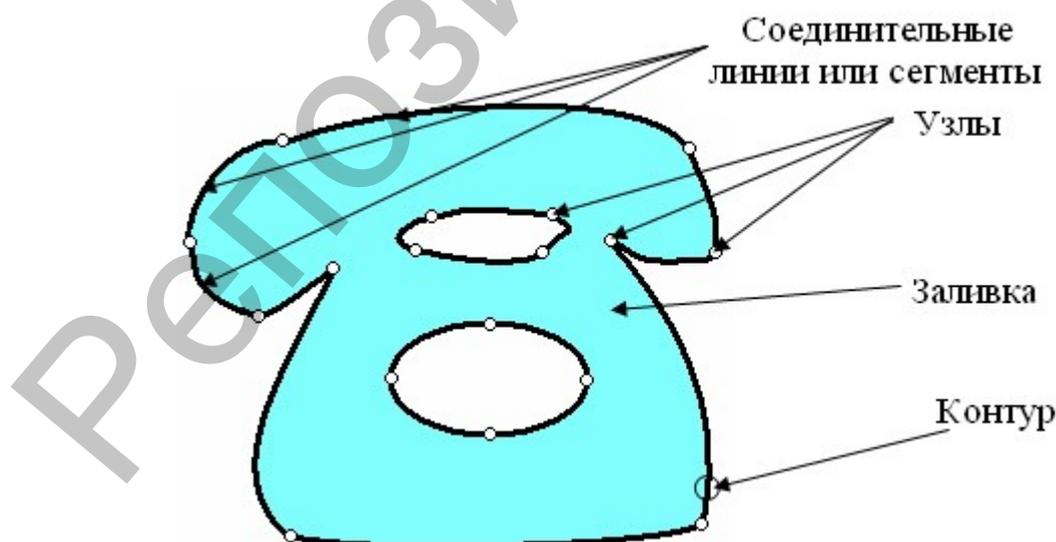


Рисунок 5. Пример объекта векторной графики

При векторном способе представления графических изображений геометрические фигуры (кривые и прямые линии),

составляющие рисунок, хранятся в памяти компьютера в виде математических формул и геометрических фигур (круг, прямоугольник, эллипс и т.п.). Подробнее остановимся на способах представления различных объектов в векторной графике.

*Точка.* Этот объект на плоскости представляется двумя числами  $(x, y)$ , указывающими его положение относительно начала координат.

*Прямая линия.* Ей соответствует уравнение  $y = kx + b$ . Указав параметры  $k$  и  $b$ , всегда можно отобразить бесконечную прямую линию в известной системе координат, то есть для задания прямой достаточно двух параметров.

*Отрезок прямой.* Он отличается тем, что требует для описания ещё двух параметров – например, координат  $x_1$  и  $x_2$  начала и конца отрезка.

*Кривая второго порядка.* К этому классу кривых относятся параболы, гиперболы, эллипсы, окружности, то есть все линии, уравнения которых содержат степени не выше второй. Кривая второго порядка не имеет точек перегиба. Прямые линии являются всего лишь частным случаем кривых второго порядка. Формула кривой второго порядка в общем виде может выглядеть так:

$$x^2 + a_1y^2 + a_2xy + a_3x + a_4y + a_5 = 0.$$

Таким образом, для описания бесконечной кривой второго порядка достаточно пяти параметров. Если требуется построить отрезок кривой, понадобятся ещё два параметра.

*Кривая третьего порядка.* Отличие этих кривых от кривых второго порядка состоит в возможном наличии точки перегиба. Например, график функции  $y = x^3$  имеет точку перегиба в начале координат. Именно эта особенность позволяет сделать кривые третьего порядка основой отображения природных объектов в векторной графике. Например, линии изгиба человеческого тела весьма близки к кривым третьего порядка. Все кривые второго порядка, как и прямые, являются частными случаями кривых третьего порядка.

В общем случае уравнение кривой третьего порядка можно записать так:

$$x^3 + a_1y^3 + a_2x^2y + a_3xy^2 + a_4x^2 + a_5y^2 + a_6xy + a_7x + a_8y + a_9 = 0.$$

Таким образом, кривая третьего порядка описывается девятью параметрами. Описание её отрезка потребует на два параметра больше.

*Кривые Безье.* Это особый, упрощённый вид кривых третьего порядка. Метод построения такой кривой основан на открытии математика Пьера Безье и опирается на тот факт, что для задания кривой достаточно использовать пару касательных, проведённых к отрезку линии в её окончаниях. Отрезки кривых Безье описываются

восемью параметрами, поэтому с ними удобно работать. На форму линии влияет угол наклона касательной и длина её отрезка.

Кроме положений начальной и конечной точки (т.е. узлов кривой), внешний вид кривой определяется кривизной, то есть её изогнутостью между двумя узлами. Кривизна определяется двумя параметрами кривой в каждом узле, которые графически представлены с помощью отрезков, выходящих из узлов. Эти отрезки называются *манипуляторами кривизны*.

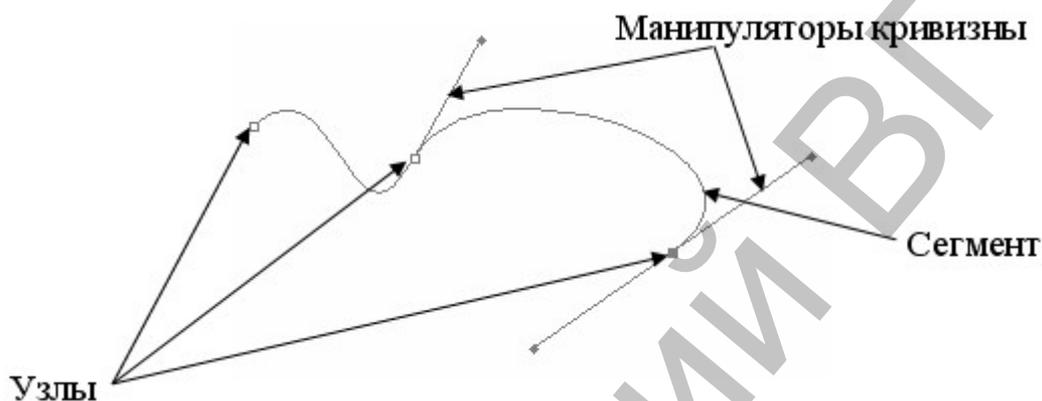
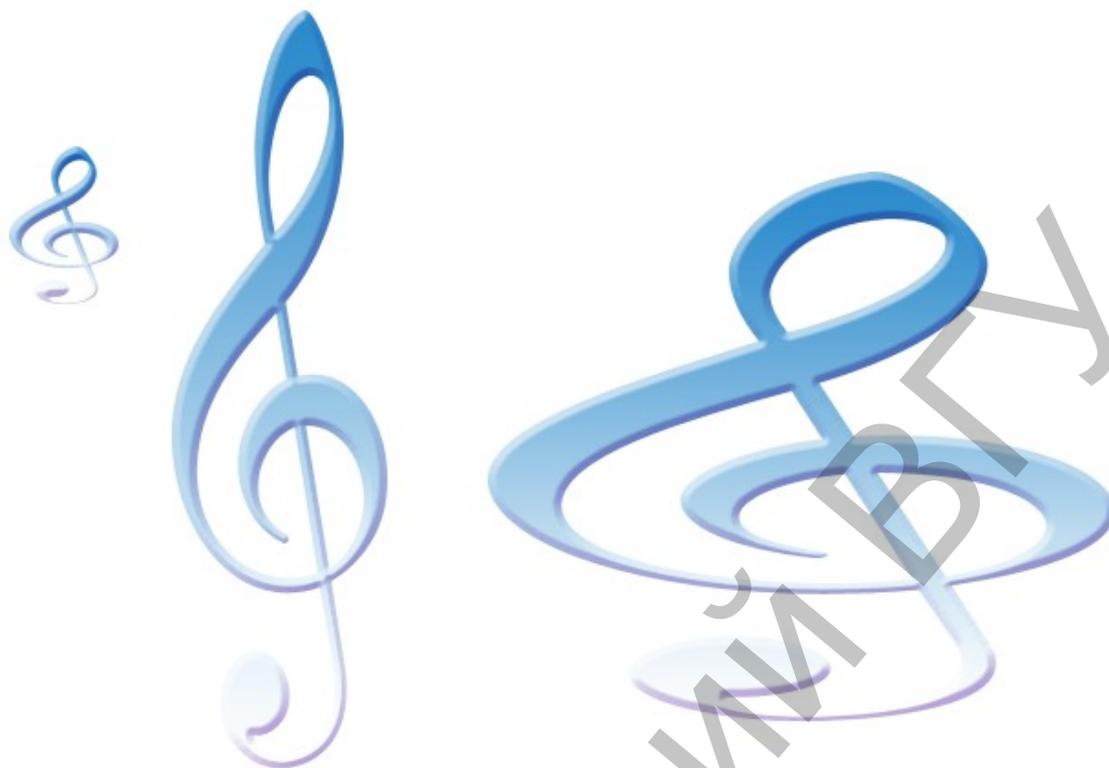


Рисунок 6. Кривая Безье

Первым параметром, определяющим кривизну, является наклон кривой при её входе в узел. Наклон манипулятора кривизны и показывает наклон кривой. Вторым параметром является степень кривизны, то есть, то, как быстро при удалении от узла кривая расходится с прямой, проведённой через узел с тем же наклоном. Степень кривизны определяется длиной манипулятора кривизны. Таким образом, координаты узлов, наклон и длина манипуляторов кривизны определяют внешний вид кривой Безье.

Если манипуляторы кривизны с обеих сторон сегмента имеют нулевую длину, то сегмент будет прямым. Увеличение длины манипулятора кривизны превратит сегмент в кривую.



**Рисунок 7.** Масштабирование векторного рисунка

При хранении векторных изображений размеры, кривизна и местоположение элементов изображения хранятся в виде числовых коэффициентов. Поэтому появляется возможность масштабировать изображения с помощью простых математических операций, например, простым умножением параметров графических элементов на коэффициент масштабирования. При этом качество изображения остаётся без изменений. На рисунке 7 приведен пример изменения векторного рисунка – изображения скрипичного ключа. Изображение можно вытягивать вертикально, горизонтально, увеличивать или уменьшать размер без потерь качества изображения.

Размеры графических файлов векторной графики имеют значительно меньший размер, чем файлы растровой графики.

Поскольку векторные рисунки состоят из кривых, описанных формулами, реалистичное изображение с их помощью получить трудно, для этого понадобилось бы слишком много элементов. Поэтому рисунки векторной графики не могут использоваться для кодирования фотографий.

Достоинства векторной графики.

- Всегда сохраняющаяся независимость объектов и невозможность совершить необратимые действия.
- Возможность неограниченного масштабирования изображения без потери качества и без увеличения размера файла.
- Высокая точность рисования.

- Экономия дискового пространства, меньший объем файлов (хранится не само изображение, а математическая формула объекта).
  - Богатые возможности текстовых эффектов.
  - Независимость от разрешения устройств вывода.
  - Возможность создания многостраничных документов с элементами логической разметки, интерактивные и анимационные возможности.
- Недостатки векторной графики.
- Сложно осуществить экспорт изображения из растрового формата в векторный.
  - Векторная графика ограничена в чисто живописных средствах, не позволяет получать фотореалистичные изображения.
  - Векторный принцип описания изображения не позволяет автоматизировать ввод графической информации, как это делает сканер для растровой графики.
  - В векторной графике невозможно применить обширную библиотеку эффектов (фильтров).
  - Процесс стандартизации слабо коснулся векторных форматов. Разработчики программ векторной графики предпочитают иметь дело только с собственными форматами.

### **Фрактальная графика**

Фрактальная графика, как и векторная, основана на математических вычислениях. Однако базовым элементом фрактальной графики является сама математическая формула, то есть в памяти компьютера никаких объектов не хранится и изображение строится исключительно по уравнениям. Таким способом строятся как простейшие регуляторные структуры, так и сложные иллюстрации, имитирующие природные ландшафты и трёхмерные объекты.

Слово *фрактал* (от лат. fractus) в переводе означает «состоящий из фрагментов, дроблёный». Впервые его использовал Бенуа Мандельброт в 1975 году для обозначения нерегулярных, но самоподобных структур, которыми он занимался. Рождение фрактальной геометрии принято связывать с выходом в 1977 году книги Мандельброта “*The Fractal Geometry of Nature*”. В его работах использованы научные результаты других ученых, работавших в период 1875 – 1925 годов в той же области (Пуанкаре, Фату, Жюлиа, Кантор, Хаусдорф).

Определение фрактала, данное Мандельбротом, звучит так: «Фракталом называется структура, состоящая из частей, которые в каком-то смысле подобны целому». Одним из основных свойств

фракталов является *самоподобие*. Объект называют самоподобным, когда увеличенные части объекта походят на сам объект и друг на друга. Можно сказать, что в простейшем случае небольшая часть фрактала содержит информацию обо всем фрактале.

*Фрактал* – это бесконечно самоподобная геометрическая фигура, каждый фрагмент которой повторяется при уменьшении масштаба. Масштабная инвариантность, наблюдаемая во фракталах, может быть либо точной, либо приближённой.

В основу метода построения фрактальных изображений положен принцип наследования от, так называемых, «родителей» геометрических свойств объектов-наследников. Другими словами, фрактал – это объект, отдельные элементы которого наследуют свойства родительских структур.

Фрактальная графика, как и векторная – вычисляемая, то есть, основана на математических вычислениях, но отличается от неё тем, что никакие объекты в памяти компьютера не хранятся. Создание фрактального рисунка или композиции состоит в генерации изображений путем математических расчетов, т.е. в программировании. Сама математическая формула является базовым элементом фрактальной графики. Изображение строится по уравнению (или по системе уравнений), поэтому ничего, кроме формулы, хранить не надо. Изменив коэффициенты в уравнении, можно получить совершенно другое изображение. Эта идея нашла использование в компьютерной графике благодаря компактности математического аппарата, необходимого для ее реализации. Так, с помощью нескольких математических коэффициентов можно задать линии и поверхности очень сложной формы.

Способность фрактальной графики моделировать образы живой природы вычислительным путем часто используют для автоматической генерации необычных иллюстраций. С точки зрения машинной графики, фрактальная геометрия незаменима при генерации искусственных облаков, гор, поверхности моря. Это способ легкого представления сложных неевклидовых объектов, образы которых весьма похожи на природные.

Чтобы представить все многообразие фракталов удобно прибегнуть к их общепринятой классификации.

*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.

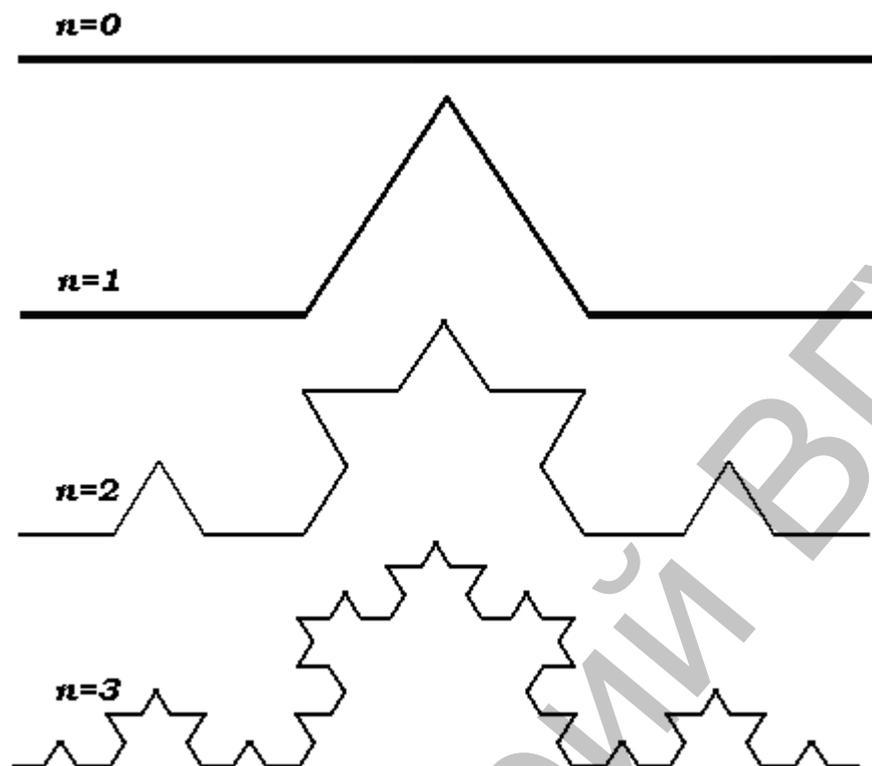


Рисунок 8. Поколения кривой Коха

Рассмотрим один из таких фрактальных объектов – триадную кривую Коха. Построение кривой начинается с отрезка единичной длины (рисунок 8) – это нулевое поколение кривой Коха ( $n = 0$ ). Далее каждое звено (в нулевом поколении один отрезок) заменяется на образующий элемент, обозначенный на рисунке 8 через  $n = 1$ . В результате такой замены получается следующее поколение кривой Коха. В 1-ом поколении – это кривая из четырех прямолинейных звеньев, каждое длиной по  $1/3$ . Для получения 2-го поколения прделываются те же действия – каждое звено заменяется на уменьшенный образующий элемент. Итак, для получения каждого последующего поколения, все звенья предыдущего поколения необходимо заменить уменьшенным образующим элементом. На рисунке 8 представлены три поколения кривой. При  $n$  стремящемся к бесконечности кривая Коха становится фрактальным объектом.

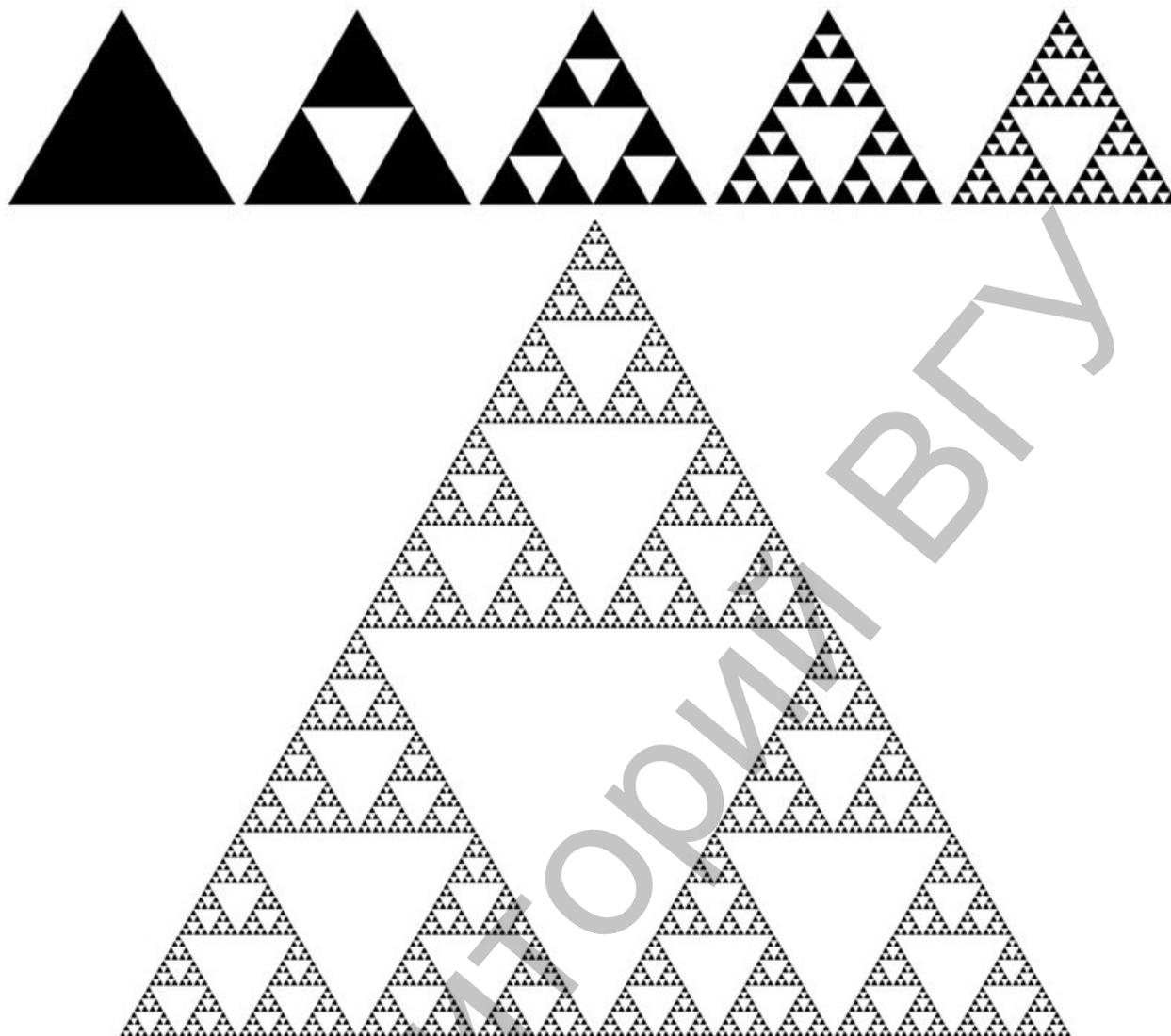


Рисунок 9. Треугольник Серпинского

На рисунке 9 приведен ещё один пример геометрического фрактала – треугольник Серпинского. В центре фрактальной фигуры находится её простейший элемент – равносторонний треугольник, который получил название «фрактальный». Затем, на среднем отрезке сторон строятся равносторонние треугольники со стороной, равной  $\frac{1}{3}a$  от стороны исходного фрактального треугольника. В свою очередь, на средних отрезках сторон полученных треугольников, являющихся объектами-наследниками первого поколения, выстраиваются треугольники-наследники второго поколения со стороной  $\frac{1}{9}a$  от стороны исходного треугольника.

Таким образом, мелкие элементы фрактального объекта повторяют свойства всего объекта. Полученный объект носит название *фрактальной фигуры*. Процесс наследования можно продолжать до бесконечности.

В машинной графике использование геометрических фракталов необходимо при получении изображений деревьев, кустов, береговой линии. Двухмерные геометрические фракталы используются для создания объемных текстур (рисунка на поверхности объекта).

*Алгебраические фракталы.* Это самая крупная группа фракталов. Получают их с помощью нелинейных процессов в  $n$ -мерных пространствах. Наиболее изучены двухмерные процессы. Интерпретируя нелинейный итерационный процесс, как дискретную динамическую систему, можно пользоваться терминологией теории этих систем: *фазовый портрет, установившийся процесс, аттрактор* и так далее.

Известно, что нелинейные динамические системы обладают несколькими устойчивыми состояниями. То состояние, в котором оказалась динамическая система после некоторого числа итераций, зависит от ее начального состояния. Поэтому каждое устойчивое состояние (или аттрактор) обладает некоторой областью начальных состояний, из которых система обязательно попадет в рассматриваемые конечные состояния. Таким образом, фазовое пространство системы разбивается на области притяжения аттракторов. Если фазовым является двухмерное пространство, то, окрашивая области притяжения различными цветами, можно получить цветовой фазовый портрет этой системы (итерационного процесса). Меняя алгоритм выбора цвета, можно получить сложные фрактальные картины с причудливыми многоцветными узорами. Неожиданностью для математиков стала возможность с помощью примитивных алгоритмов порождать очень сложные нетривиальные структуры.

В качестве примера рассмотрим множество Мандельброта. Алгоритм его построения достаточно прост и основан на простом итеративном выражении:

$$z_{i+1} = z_i^2 + C.$$

где  $z_i$  и  $C$  – комплексные переменные. Итерации выполняются для каждой стартовой точки  $C$  прямоугольной или квадратной области – подмножестве комплексной плоскости. Итерационный процесс продолжается до тех пор, пока  $z_i$  не выйдет за пределы окружности радиуса 2, центр которой лежит в точке  $(0,0)$ , (это означает, что аттрактор динамической системы находится в бесконечности), или после достаточно большого числа итераций (например, 200-500)  $z_i$  сойдется к какой-нибудь точке окружности. В зависимости от количества итераций, в течение которых  $z_i$  оставалась внутри окружности, можно установить цвет точки  $C$  (если  $z_i$  остается внутри окружности в течение достаточно большого количества итераций,

итерационный процесс прекращается и эта точка растра окрашивается в черный цвет).

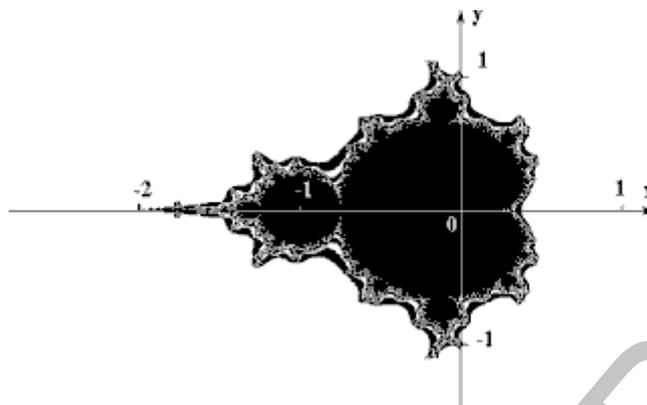


Рисунок 10. Множество Мандельброта

*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 так далее. Двумерные стохастические фракталы используются при моделировании рельефа местности и поверхности моря.

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

Фрактальная компьютерная графика позволяет создавать абстрактные композиции, где можно реализовать такие композиционные приёмы как, горизонтали и вертикали, диагональные направления, симметрию и асимметрию и др. Фактически благодаря фрактальной графике найден способ эффективной реализации сложных неевклидовых объектов, образы которых весьма похожи на природные. Геометрические фракталы на экране компьютера – это узоры, построенные самим компьютером по заданной программе. На рисунке 11 приведены примеры фрактальных изображений, полученных с помощью компьютера.

Геометрические фракталы на экране компьютера – это узоры, построенные самим компьютером по заданной программе. Помимо фрактальной живописи существуют *фрактальная анимация* и *фрактальная музыка*. Создавая фрактал, вы сами задаете форму рисунка математической формулой, исследуете сходимость процесса, варьируя его параметры, выбираете вид изображения и палитру цветов, то есть творите рисунок «с нуля». В этом одно из отличий фрактальных графических редакторов (и в частности – Painter) от прочих графических программ. Например, в Adobe Photoshop

изображение, как правило, «с нуля» не создается, а только обрабатывается. Другой самобытной особенностью фрактального графического редактора Painter (как и прочих фрактальных программ, например Art Dabbler) является то, что реальный художник, работающий без компьютера, никогда не достигнет с помощью кисти и карандаша тех возможностей, которые заложены в Painter программистами.

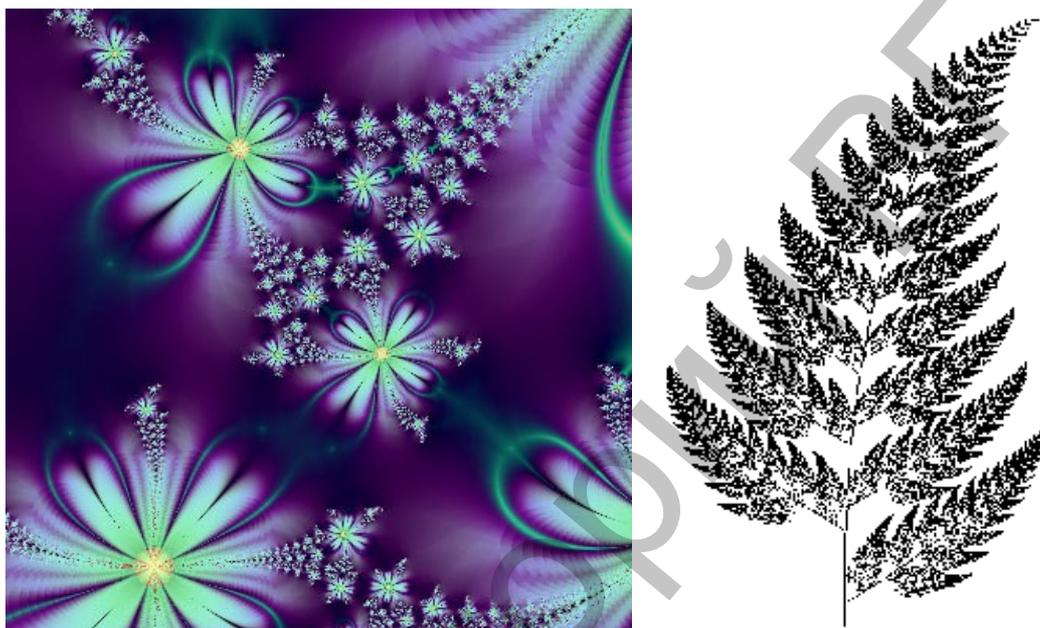


Рисунок 11. Фрактальные изображения

## Тема 2. Средства описания цвета

Каждый человек воспринимает цвет по-своему. Однако, субъективность в восприятии цвета человеком крайне нежелательна при обработке изображений. Кроме того, существуют различия в воспроизведении одного и того же цвета видеомониторами, принтерами и сканерами разных фирм-изготовителей. Возникает необходимость наличия объективных измерительных систем, позволяющих установить однозначное определение цветовых координат. Для этих целей разработаны специальные средства, включающие:

- цветовые модели;
- системы соответствия цветов;
- цветовые режимы.

### *Цветовые модели*

В основе создания цветовых моделей лежит использование универсальных языков, позволяющих реализовать способы точного описания цвета с помощью стандартных математических выражений.

В современных компьютерных программах манипуляции с цветом осуществляются с помощью цветовых моделей и режимов.

*Цветовые модели* (или цветовые пространства) предоставляют средства для концептуального и количественного описания цвета.

*Режим* – это способ реализации определенной цветовой модели в рамках конкретной графической программы.

Цветовые модели (color model) используются для математического описания определенных цветовых областей спектра. Большинство компьютерных цветовых моделей основано на использовании трех основных цветов, что соответствует восприятию цвета человеческим глазом. Каждому основному цвету присваивается определенное значение цифрового кода, после чего все остальные цвета определяются как комбинации основных цветов. Именно такой подход используют художники при создании картины на основе ограниченной палитры цветов.

Известно, что каждый пиксель растрового изображения содержит информацию о цвете. Любой векторный объект также содержит информацию о цвете его контура и заливки. Информация может занимать от одного до тридцати двух бит, в зависимости от глубины цвета.

В компьютерной графике применяют понятие цветового разрешения (другое название – глубина цвета). Оно определяет метод кодирования цветовой информации для её воспроизведения на экране монитора. Для отображения чёрно-белого изображения достаточно двух бит (белый и чёрный цвета). Восемьразрядное кодирование позволяет отобразить 256 градаций цветового тона. Два байта определяют 65 536 оттенков (такой режим называют High Color). При 24-разрядном способе кодирования, возможно, определить более 16,5 миллионов цветов (режим называют True Color).

С практической точки зрения цветовому разрешению монитора близко понятие цветового охвата. Под ним подразумевается диапазон цветов, который можно воспроизвести с помощью того или иного устройства вывода (монитор, принтер, печатная машина и прочие).

Если мы работаем с черно-белыми изображениями, то цвет кодируется нулем или единицей. Никаких проблем в этом случае не возникает. Для несложных рисунков, содержащих 256 цветов или столько же градаций серого цвета, не трудно пронумеровать все используемые цвета. Но для изображений в истинном цвете, содержащих миллионы разных оттенков, простая нумерация не подходит. Для них разработаны несколько моделей представления цвета, помогающих однозначно определить любой оттенок.

Несмотря на то, что цветовые модели позволяют представить цвет математически, такое представление всегда будет казаться

несовершенным в силу отличия от нашего восприятия. Однако они удобны при использовании в компьютерных программах для однозначного определения выводимого цвета.

Любая цветовая модель должна отвечать трем требованиям.

- Реализовывать определения цвета некоторым стандартным способом, не зависящим от возможностей какого-либо конкретного устройства.
- Точно задавать диапазон воспроизводимых цветов, поскольку ни одно множество цветов не является бесконечным.
- Учитывать механизм восприятия цветов – излучение или отражение.

Современные графические пакеты располагают развитым интерфейсом для выбора необходимой цветовой модели и цвета внутри нее.

Большинство графических пакетов позволяют оперировать широким кругом цветовых моделей, часть из которых создана для специальных целей, а другая – для особых типов красок. На сегодняшний день существует множество цветовых моделей, например: CMY, CMYK, RGB, HSB, HLS, L\*a\*b, YIQ, YCC.

Перечисленные цветовые модели по принципу действия условно можно разделить на три класса:

- *аддитивные* (RGB), основанные на сложении цветов;
- *субтрактивные* (CMY, CMYK), основу которых составляет операция вычитания цветов (субтрактивный синтез);
- *перцепционные* (HSB, HLS, L\*a\*b, YCC), базирующиеся на восприятии.

В большинстве цветовых моделей для описания цвета используется трехмерная система координат. Она образует цветовое пространство, в котором цвет представляется в виде точки с тремя координатами. Г. Грассман вывел три закона для оперирования цветом в трехмерном пространстве.

1. *Трехмерность природы цвета.* Глаз реагирует на три различные цветовые составляющие, например:

- красный, зеленый и синий цвета;
- цветовой тон (доминирующая длина волны), насыщенность (чистота) и яркость (светлость).

2. *Четыре цвета всегда линейно зависимы*, то есть:

$$cC = rR + gG + bB,$$

где  $c$ ,  $r$ ,  $g$ ,  $b$  – весовые коэффициенты для каждой из составляющих цвета. Для смеси двух цветов  $(cC)_1$  и  $(cC)_2$  справедливо равенство:

$$(cC)_1 + (cC)_2 = (rR)_1 + (gG)_1 + (bB)_1 + (rR)_2 + (gG)_2 + (bB)_2,$$

свидетельствующее о том, что цвет смеси излучений  $C$  зависит только от их цвета, но не от спектрального состава.

Следствие: если цвет  $C_1$  равен цвету  $C$  и цвет  $C_2$  тоже равен цвету  $C$ , то следует, что цвет  $C_1$  равен цвету  $C_2$  независимо от структуры спектров энергии цветов  $C$ ,  $C_1$  и  $C_2$ .

3. *Цветовое пространство непрерывно.* Если в смеси трех цветов один непрерывно изменяется, а другие остаются постоянными, то цвет смеси будет меняться непрерывно.

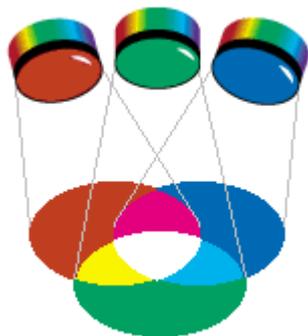
### Аддитивные цветовые модели

Известно, что солнечный свет можно разложить на отдельные цветные составляющие. Аддитивный цвет получается на основе законов Грассмана путем соединения лучей света разных цветов. В основе этого явления лежит тот факт, что большинство цветов видимого спектра могут быть получены путем смешивания в различных пропорциях трех основных цветовых компонентов (рисунок 12). Этими компонентами, которые в теории цвета иногда называются первичными цветами, являются *красный* (Red), *зеленый* (Green) и *синий* (Blue) цвета. При попарном смешивании первичных цветов образуются вторичные цвета: *голубой* (Cyan), *пурпурный* (Magenta) и *желтый* (Yellow). Следует отметить, что первичные и вторичные цвета относятся к базовым цветам.

*Базовыми цветами* называют цвета, с помощью которых можно получить практически весь спектр видимых цветов.

Большинство цветов, которые мы видим в окружающем нас мире, являются следствием отражения и поглощения света. Например, солнечный свет, падая на зеленую траву, частично поглощается, и отражается только его зеленая составляющая.

Собрав вместе в нужных пропорциях разноцветные лучи базовых цветов, мы получим луч белого цвета. Изменим немного пропорции – и у нас готов источник света заданного цвета. В телевизорах и компьютерных мониторах используется люминофор, который светится красным, зеленым и синим цветом. Смешав эти три цвета, можно получить разнообразные цвета и их оттенки. На этом и основана модель представления цвета *RGB*, названная так по начальным буквам входящих в нее цветов: *Red* – красный, *Green* – зеленый, *Blue* – синий. Любой цвет в этой модели представляется тремя числами, описывающими величину каждой цветовой составляющей. Черный цвет образуется, когда интенсивность всех трех составляющих равна нулю, а белый – когда их интенсивность максимальна.



**Рисунок 12.** Аддитивная цветовая модель RGB

RGB-модель возникла задолго до появления компьютера, и в каждой области ее применения появлялись свои термины. RGB-модель, использующая для построения первичные (или аддитивные) цвета, имеет еще одно название – *добавляющая модель*. Это название подчеркивает тот факт, что при добавлении света интенсивность цвета увеличивается.

Аддитивные цвета нашли широкое применение в системах освещения, видеосистемах, устройствах записи на фотопленку, мониторах, сканерах и цифровых камерах. Большинство компьютерного оборудования работает с использованием модели RGB, кроме того, эта модель очень проста. Этим объясняется ее широкое распространение.

К сожалению, в модели RGB теоретически невозможно получить некоторые цвета, например, насыщенный сине-зеленый, поэтому работать с моделью цвета RGB не всегда удобно. Кроме того, модель RGB сильно связана с реализацией ее на конкретных устройствах. В настоящее время достаточно распространенным является формат True Color, в котором каждая компонента представлена в виде байта, что дает 256 градаций для каждой компоненты:  $R = 0..255$ ,  $G = 0..255$ ,  $B = 0..255$ . Количество цветов в этом случае составляет  $256^3 = 16\,777\,216$ .

Математически цветовую модель RGB часто представляют в виде единичного куба с ортами:  $(1;0;0)$  – красный (ось X),  $(0;1;0)$  – зеленый (ось Y),  $(0;0;1)$  – синий (ось Z) и началом  $(0;0;0)$  – черный. В этом случае каждая пространственная точка однозначно определяется значениями координат X, Y и Z. Каждому цвету можно поставить в соответствие точку внутри куба. На рисунке 13 показан куб и также распределение цветов вдоль указанных векторов.

При использовании этой модели любой цвет может быть представлен в цветовом пространстве с помощью вектора, описываемого уравнением:

$$cC = rR + gG + bB$$

Это уравнение идентично уравнению свободного вектора в пространстве, рассматриваемому в векторной алгебре. При этом направление вектора характеризует *цветность*, а его модуль выражает *яркость*.

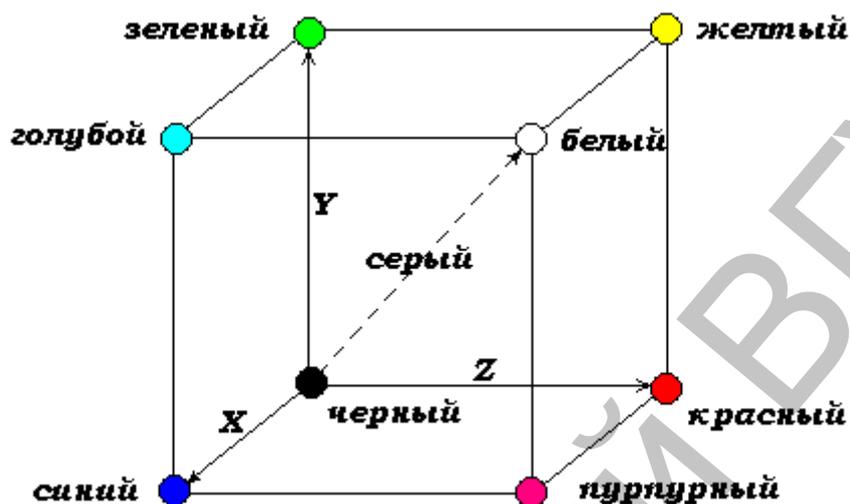


Рисунок 13. Схема RGB-модели в виде куба

Вершины куба, в которых отмечены зелёный, красный и синий цвета на рисунке 13, характеризуют максимальные интенсивности зеленого, красного и синего соответственно, а треугольник, который они образуют, называется треугольником Паскаля (см. рисунок 14).

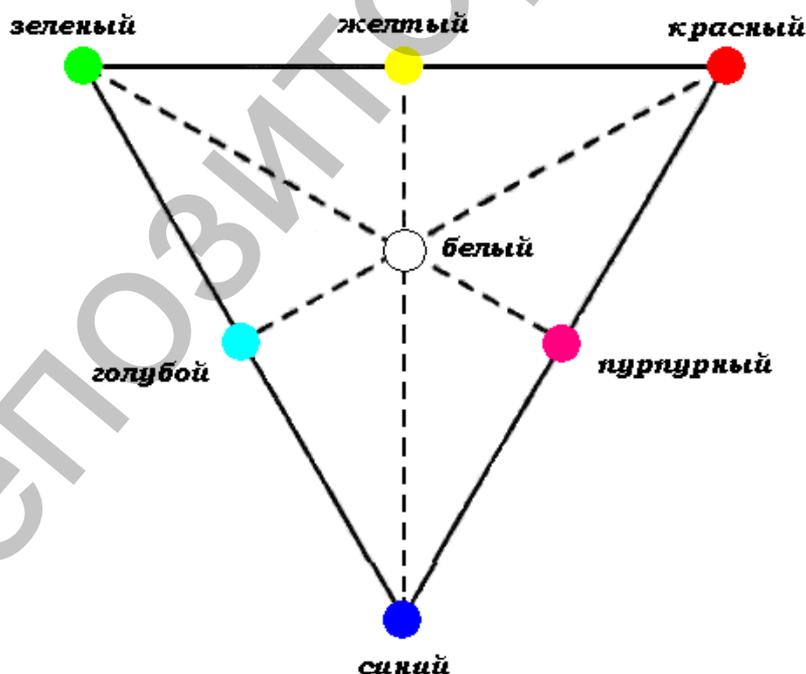
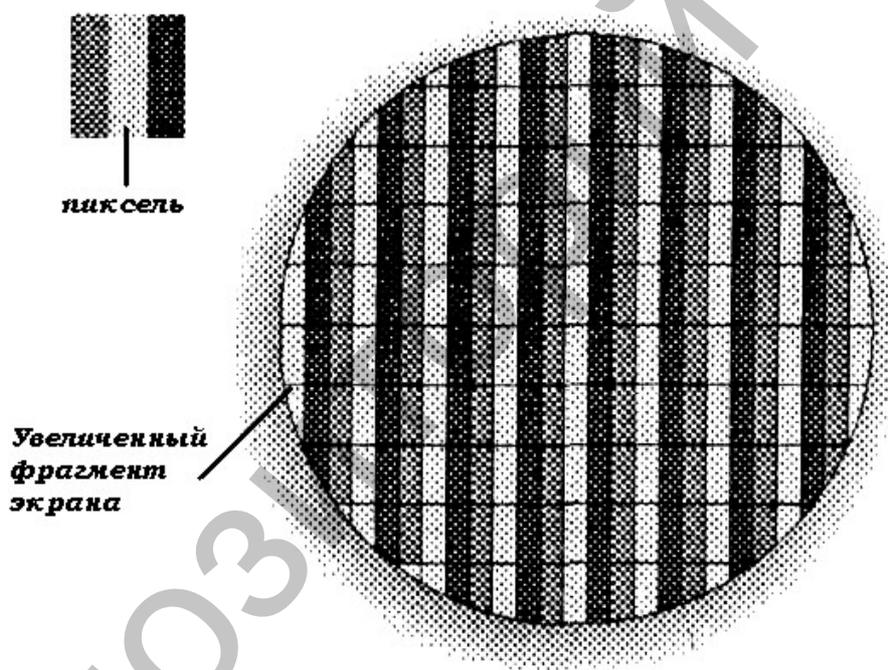


Рисунок 14. Колориметрическая схема получения цветового пространства RGB-модели с помощью базовых цветов. Стороны треугольника образуют множество спектрально-чистых цветов (локус)

На диагонали (ахроматической оси), соединяющей точки с координатами  $(R, G, B) = (0, 0, 0)$  и  $(R, G, B) = (255, 255, 255)$ ,

расположены различные градации серого, для которых значения красной, зеленой и синей составляющих одинаковы.

Как уже говорилось, мельчайший элемент изображения, воспроизводимый компьютером, называется пикселем. При работе с низким разрешением отдельные пиксели не видны. Однако если рассматривать белый экран включенного монитора через лупу, то будет видно, что он состоит из множества отдельных точек (маленьких электронных прожекторов) красного, зеленого и синего цветов, объединенных в RGB-элементы в виде триад основных точек. Цвет каждого выводимого пикселя (RGB-элементов изображения) получается в результате смешивания красного, синего и зеленого цветов входящих в него трех люминофорных точек. При просмотре изображения на экране с некоторого расстояния эти цветовые составляющие RGB-элементов сливаются, создавая иллюзию



результатирующего цвета.

**Рисунок 15.** Экран монитора состоит из множества триад маленьких точек красного, зелёного и синего цвета, называемых пикселями

Для назначения цвета и яркости точек, формирующих изображение монитора, следует задать значения интенсивностей для каждой из составляющих RGB-элемента (пикселя). В этом процессе значения интенсивностей используются для управления мощностью трех электронных прожекторов, возбуждающих свечение соответствующего типа люминофора. В то же время число градаций интенсивности определяет *цветовое разрешение*, или, иначе, *глубину цвета*. Цветовое разрешение характеризуют максимальное количество воспроизводимых цветов  $256^3 = 16\,777\,216$ .

Несмотря на то, что цветовая модель RGB достаточно проста и наглядна, при ее применении на практике возникают две серьезные проблемы:

- аппаратная зависимость;
- ограничение цветового охвата.

Первая проблема связана с тем, что цвет, возникающий в результате смешения цветовых составляющих RGB-элемента, зависит от типа люминофора. А поскольку в технологии производства современных кинескопов применяются разные типы люминофоров, то установка одних и тех же интенсивностей электронных лучей в случае различных люминофоров приведет к синтезу разного цвета. Например, если на электронный блок монитора подать определенную тройку RGB-значений, скажем  $R = 87$ ,  $G = 103$  и  $B = 215$ , то нельзя однозначно сказать, каков будет результат смешивания. Эти значения всего лишь задают интенсивности возбуждения трех люминофоров одного элемента изображения. Какой получится при этом цвет, зависит от спектрального состава излучаемого люминофором света. Поэтому в случае аддитивного синтеза для однозначного определения цвета наряду с установкой триады значений интенсивностей необходимо знать спектральную характеристику люминофора.

Существуют и другие причины, приводящие к аппаратной зависимости RGB-модели даже для мониторов, выпускаемых одним и тем же производителем. Это связано, в частности, с тем, что в процессе эксплуатации происходит старение люминофора и изменение эмиссионных характеристик электронных прожекторов. Для минимизации зависимости RGB-модели от аппаратных средств используются различные устройства и программы градуировки.

*Цветовой охват* (color gamut) – это диапазон цветов, который может различать человек или воспроизводить устройство независимо от механизма получения цвета (излучения или отражения).

*Ограниченность цветового охвата* объясняется тем, что с помощью аддитивного синтеза принципиально невозможно получить все цвета видимого спектра. В частности, некоторые цвета, такие как чистый голубой или чистый желтый, не могут быть точно воссозданы на экране. Но, несмотря на то, что человеческий глаз способен различать больше цветов, чем монитор, RGB-модели вполне достаточно для создания цветов и оттенков, необходимых для воспроизведения фотореалистических изображений на экране компьютера.

### **Субтрактивные цветовые модели**

В отличие от экрана монитора, воспроизведение цветов которого основано на излучении света, печатная страница может

только отражать цвет. Поэтому KRGB-модель в данном случае неприемлема. Вместо нее для описания печатных цветов используется модель CMY, базирующаяся на субтрактивных цветах.

*Субтрактивные цвета* в отличие от аддитивных цветов (той же RGB-модели) получаются вычитанием вторичных цветов из общего луча света. В этой системе белый цвет появляется как результат отсутствия всех цветов, тогда как их присутствие дает черный цвет.

В последнее время в качестве синонима термина «субтрактивная» иногда используют термин «исключающая». Происхождение этого названия связано с явлением отражения света от покрытой красителем поверхности, а также с тем фактом, что при добавлении красителей интенсивность света уменьшается, поскольку свет поглощается тем больше, чем больше красителя нанесено на поверхность.

Нанесение на бумагу трех базовых цветов: *голубого* (Cyan), *пурпурного* (Magenta) и *желтого* (Yellow) позволяет создать множество субтрактивных цветов. При печати на принтере на бумагу наносится цветная краска, которая отражает только свет определенного цвета. Все остальные цвета поглощаются (вычитаются) из солнечного света. На эффекте вычитания цветов построена модель представления цвета, называемая CMY. Эти буквы также взяты из названий цветов: *Cyan* – голубой, *Magenta* – пурпурный, *Yellow* – желтый. Строго говоря, Magenta не является пурпурным цветом. Точное название этого цвета – фуксин, но в компьютерной литературе и в программах принято называть этот цвет пурпурным.

Перечислим соотношения, связывающие аддитивные (красный, зеленый, синий) и субтрактивные (голубой, желтый, пурпурный) цвета:

*Зеленый + Синий = Голубой*

*Зеленый + Красный = Желтый*

*Красный + Синий = Пурпурный*

*Зеленый + Синий + Красный = Белый*

*Голубой + Желтый + Пурпурный = Черный*

Итак, что же происходит, когда на лист бумаги с нанесенным на него красителем падает белый свет? Если краситель голубой (сине-зеленый), то он поглощает из спектра комплиментарный ему красный цвет и отражает голубой. Соответственно пурпурный краситель поглощает комплиментарный ему зеленый цвет, а желтый краситель – синий цвет. Если при печати наложить друг на друга пурпурный и желтый цвета, то получится красный цвет, поскольку пурпурный краситель устранил зеленую составляющую, а желтый – синюю составляющую падающего цвета. Соответственно при печати с

наложением всех трех субтрактивных цветов результирующий цвет будет черным.

На основе приведенных выше рассуждений можно сформулировать *правило коррекции цветового дисбаланса при цветной печати*: если изображение имеет излишне синий оттенок, следует увеличить желтую составляющую, поскольку желтый поглощает синие составляющие. Соответственно избыточность зеленого цвета можно скорректировать увеличением пурпурной составляющей, а избыточность красного цвета – увеличением голубой составляющей.

Существует две наиболее распространенные версии субтрактивной модели: CMY и CMYK. Первая из них используется в том случае, если изображение или рисунок будут выводиться на черно-белом принтере, позволяющем заменять черный картридж на цветной (color upgrade). В ее основе лежит использование трех субтрактивных (вторичных) цветов: голубого (Cyan), пурпурного (Magenta) и желтого (Yellow). Теоретически при смешивании этих цветов на белой бумаге в равной пропорции получается черный цвет. Однако в реальном технологическом процессе получение черного цвета путем смешивания трех основных цветов для бумаги неэффективно по трем причинам.

- Невозможно произвести идеально чистые пурпурные, синие и желтые краски, поэтому цвет получается не чисто черным, а грязно-коричневым.
- На создание черного цвета с помощью модели CMY тратится в три раза больше краски.
- Любые цветные краски дороже обычных черных.

В силу перечисленных факторов при печати используется добавка дополнительного черного компонента цвета. Эта технология приводит также к улучшению качества теней и серых оттенков. Интенсивность каждого из четырех компонентов цвета может изменяться в диапазоне от 0 до 100 %.

Система CMYK широко применяется в полиграфии. Типографское оборудование работает исключительно с этой моделью, да и современные принтеры тоже используют красители четырех цветов. При печати на бумагу наносятся несколько слоев прозрачной краски, и в результате мы получаем цветное изображение, содержащее миллионы различных оттенков.

Системы RGB и CMYK удобны при работе с конкретным оборудованием, но не очень удобны для человеческого восприятия. Представив себе желаемый цвет, вы не сможете сказать, сколько в нем составляющих цветов той или иной модели.

СМΥК-модель имеет те же ограничения, что и RGB-модель: аппаратная зависимость и ограниченный цветовой диапазон.

В СМΥК-модели также нельзя точно предсказать результирующий цвет только на базе численных значений ее отдельных компонентов. В этом смысле она является даже более аппаратно-зависимой моделью, чем RGB. Это связано с тем, что в ней имеется большее количество дестабилизирующих факторов, чем в RGB-модели. К ним в первую очередь можно отнести вариацию состава цветных красителей, используемых для создания печатных цветов. Цветовое ощущение определяется еще и типом применяемой бумаги, способом печати и внешним освещением.

В силу того, что цветные красители имеют худшие характеристики по сравнению с люминофорами, цветовая модель СМΥК имеет более узкий цветовой диапазон по сравнению с RGB-моделью. В частности, она не может воспроизводить яркие насыщенные цвета, а также ряд специфических цветов, таких, например, как металлический или золотистый.

Об экранных цветах, которые невозможно точно воссоздать при печати, говорят, что они *лежат вне цветового охвата* (gamut alarm) СМΥК-модели. В большинстве графических пакетов под такими цветами понимаются цвета, которые могут быть представлены в формате RGB или HSB, но при этом не имеют печатных аналогов в цветовом пространстве СМΥК.

### **Перцепционные цветовые модели**

Следующая модель цвета основана на восприятии цвета человеком. Для человека основным инструментом индикации и воспроизведения цвета служит глаз. Этот естественный «инструмент» обладает цветовым охватом, намного превышающим возможности любого технического устройства.

Как уже было сказано, цветовые системы RGB и СМΥК являются аппаратно-зависимыми. Это значит, что воспроизводимый или создаваемый с помощью них цвет определяется не только составляющими модели, но и зависит от характеристик устройства вывода.

Для устранения аппаратной зависимости был разработан ряд так называемых *перцепционных* (иначе – интуитивных) цветовых моделей. В их основу заложено раздельное определение яркости и цветности. Такой подход обеспечивает ряд преимуществ:

- позволяет обращаться с цветом на интуитивно понятном уровне;

- значительно упрощает проблему согласования цветов, поскольку после установки значения яркости можно заняться настройкой цвета.

Модель HSB или ее ближайший аналог HSL представлены в большинстве современных графических пакетов. Из всех используемых в настоящее время моделей она наиболее точно соответствует способу восприятия цветов человеческим глазом и позволяет описывать цвета интуитивно ясным способом. В HSB-модели все цвета описываются с помощью комбинации трех базовых параметров:

- *цветовой тон* (H – *Hue*, значение измеряется в градусах от 0 до 360, поскольку здесь цвета радуги располагаются по кругу);
- *насыщенность* (S – *Saturation*, значения находятся в диапазоне [0..1]);
- *яркость* (V – *Brightness*, значения находятся в диапазоне [0..1]).

Например, при  $S = 0$ , т.е. на оси V – серые тона. Значение  $V = 0$  соответствует черному цвету. Белый цвет кодируется как  $S = 0, V = 1$ . Цвета, расположенные по кругу напротив друг друга, т.е. отличающиеся по H на  $180^\circ$ , являются дополнительными.

На рисунке 16 представлена цветовая палитра HSB-модели. Структурная схема этой модели представлена на рисунке 17.

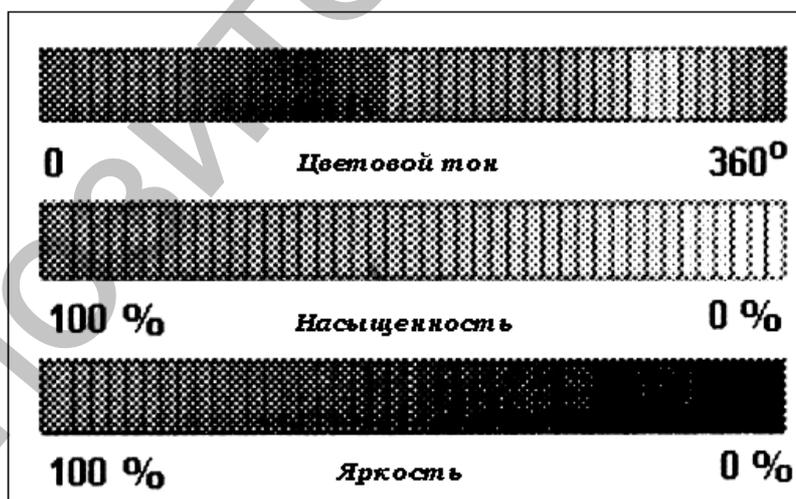
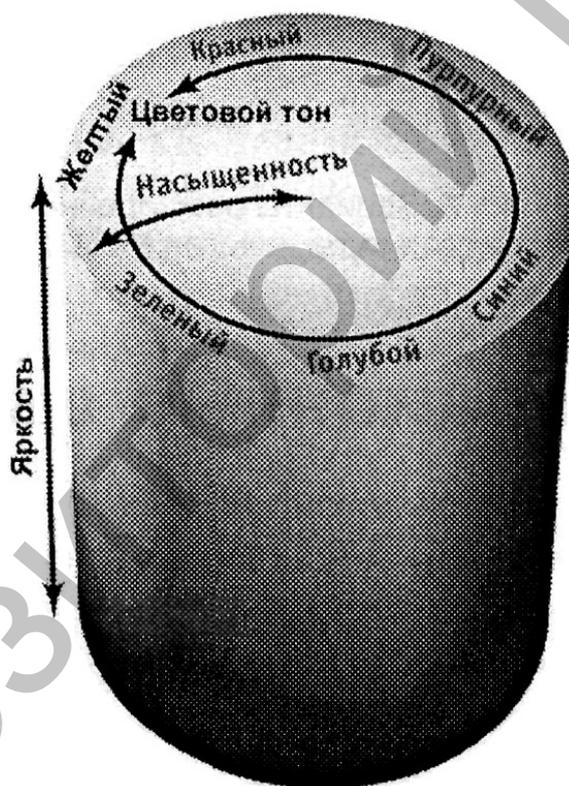


Рисунок 16. Палитра выбора цвета с помощью HSB-модели

Модель HSB не зависит от оборудования и удобна для восприятия человеком, поэтому с ней часто работают различные программы, в дальнейшем преобразуя цвета в модель RGB для показа на экране монитора или в модель CMYK – для печати на принтере. Кроме того, модель HSB удобно использовать при редактировании рисунков. Например, вы хотите заменить зеленый лист на желтый. В редактируемой фотографии достаточно поменять только цветовую

составляющую используемых цветов, не меняя яркость и насыщенность. Рисунок при этом не изменится, но примет иной оттенок.

Модель HSB в отличие от моделей RGB и CMYK носит абстрактный характер. Отчасти это связано с тем, что цветовой тон и насыщенность цвета нельзя измерить непосредственно. Любая форма ввода цветовой информации всегда начинается с определения красной, зеленой и синей составляющих, на базе которых затем с помощью математического пересчета получают компоненты HSB-модели. В результате эта цветовая модель имеет то же цветовое пространство, что и RGB-модель, а значит, и присущий ей недостаток – ограниченное цветовое пространство.



**Рисунок 17.** Структурная схема HSB-модели в виде цилиндра

Вместе с тем HSB-модель обладает по сравнению с RGB- и CMYK-моделями двумя важными преимуществами:

- Аппаратной независимостью. Задание составляющих этой модели в виде значений цветового тона, насыщенности и яркости позволяет однозначно определить цвет без необходимости учета параметров устройства вывода.
- Более простым и интуитивно понятным механизмом управления цветом.

Это связано с тем, что цветовой тон, насыщенность и яркость представляют собой независимые характеристики цвета. Например,

чистый красный цвет расположен на цветовом круге под углом  $0^\circ$ . Если нужно сместить красный тон к оранжевому тону, то следует лишь несколько увеличить угол, определяющий цветовой тон. Для получения более блеклого цвета достаточно лишь снизить насыщенность, а для придания ему большей яркости – соответственно увеличить значение яркости. Получить такие эффекты с помощью RGB-модели практически невозможно, поскольку значения ее цветовых компонентов очень сильно зависят друг от друга. Поэтому при изменении одной из ее составляющих, например красной, это повлияет не только на цветовой тон, но одновременно и на насыщенность и яркость.

Существуют и иные модели представления цвета, но в подавляющем большинстве случаев используются перечисленные выше.

Рассмотренные выше цветовые модели являются компонентным способом кодирования цвета.

Часто для описания оттенков используются фиксированные палитры, то есть список заданных цветов. В результате исследований определяют наиболее часто используемые цвета и помещают их в палитру. Каждый цвет изображения в этом случае кодируется индексом. Палитру можно воспринимать как таблицу цветов. Она устанавливает взаимосвязь между индексом цвета и его компонентами в выбранной цветовой модели. Поэтому такой способ кодирования цвета называется индексным.

### **Тема 3. Типы графических изображений**

В компьютерной графике применяют, по меньшей мере, три десятка форматов файлов для хранения изображений. Но лишь часть из них стала стандартом «де-факто» и применяется в подавляющем большинстве программ. Как правило, несовместимые форматы имеют файлы растровых, векторных, трёхмерных изображений, хотя существуют форматы, позволяющие хранить данные разных классов. Многие приложения ориентированы на собственные «специфические» форматы, перенос их файлов в другие программы вынуждает использовать специальные фильтры или экспортировать изображения в «стандартный» формат.

**TIFF** (Tagged Image File Format). Формат предназначен для хранения растровых изображений высокого качества (расширение имени файла .tif). Относится к числу широко распространённых, отличается переносимостью между платформами (IBM PC и Apple Macintosh), обеспечен поддержкой со стороны большинства графических, верстальных и дизайнерских программ. Предусматривает широкий диапазон цветового охвата – от

монохромного чёрно-белого до 32-разрядной модели цветоделения CMYK. Начиная с версии 6.0 в формате TIFF можно хранить сведения о масках изображений. Для уменьшения размера файла применяется встроенный алгоритм сжатия LZW.

**PSD** (PhotoShopDocument). Собственный формат программы AdobePhotoshop, один из наиболее мощных по возможностям хранения растровой графической информации. Позволяет запоминать параметры слоёв, каналов, степени прозрачности, множества масок. Поддерживаются 48-разрядное кодирование цвета, цветоделение и различные цветовые модели. Основной недостаток выражен в том, что отсутствие эффективного алгоритма сжатия информации приводит к большому объёму файлов.

**PCX**. Формат появился как формат хранения растровых данных программы PC PaintBrush фирмы Z-Soft и долгое время являлся одним из наиболее распространённых (расширение имени файла .pcx). В настоящее время он считается устаревшим.

**PhotoCD**. Формат разработан фирмой Kodak для хранения цифровых растровых изображений высокого качества (расширение имени файла .PCD). Сам формат хранения данных в файле называется ImagePac. Файл имеет внутреннюю структуру, обеспечивающую хранение изображения с фиксированными величинами разрешений, и потому размеры любых файлов лишь незначительно отличаются друг от друга и находятся в диапазоне 4-5 Мбайт. Каждому разрешению присвоен собственный уровень, отсчитываемый от так называемого базового (Base), составляющего 512×768 точек. Всего в файле пять уровней – от Base/16 (128×192 точек) до Base\*16 (2048×3072). При первичном сжатии исходного изображения применяется метод субдискретизации, практически без потери качества. Затем вычисляются разности Base-Base\*4 и Base\*16. Итоговый результат вписывается в файл. Чтобы воспроизвести информацию с высоким разрешением производится обратное преобразование. Для хранения информации о цвете использована цветовая модель YCC.

**Windows Bitmap**. Формат хранения растровых изображений в операционной системе Windows (расширение имени файла .bmp). Соответственно поддерживается всеми приложениями, работающими в этой среде.

**JPEG** (Joint Photographic Experts Group). Формат предназначен для хранения растровых изображений (расширение имени файла .jpg). Позволяет регулировать соотношение между степенью сжатия файла и качеством изображения. Применяемые методы сжатия основаны на удалении «избыточной» информации, поэтому формат рекомендуют использовать только для электронных публикаций

**GIF** (Graphics Interchange Format). Стандартизован в 1987 году как средство хранения сжатых изображений с фиксированным (256) количеством цветов (расширение имени файла .gif). Получил популярность в Интернете благодаря высокой степени сжатия. Последняя версия формата GIF89a позволяет выполнять чересстрочную загрузку изображений и создавать рисунки с прозрачным фоном. Ограниченные возможности по количеству цветов обуславливают его применение исключительно в электронных публикациях.

**PNG** (Portable Network Graphics). Сравнительно новый (1995 год) формат хранения изображений для их публикации в Интернете (расширение файла .png). Поддерживаются три типа изображений – цветные с глубиной 8 или 24 бита и чёрно-белое с градацией 256 оттенков серого. Сжатие информации происходит практически без потерь, предусмотрены 25 уровня альфа-канала, чересстрочная развёртка.

Формат [PNG](#) получил наибольшее признание из претендентов на роль наследника GIF. Возможностью работать с файлами этого формата уже обладают многие графические пакеты – например, Photoshop 4.0, а также популярный shareware-редактор [Paint Shop Pro](#). Web-браузеры Netscape Navigator и Microsoft Internet Explorer также поддерживают PNG.

Отличия формата PNG от GIF:

- PNG реализует открытый, незапатентованный алгоритм сжатия данных, дающий, тем не менее, лучшие результаты, чем GIF.
- В отличие от GIF с его максимумом 256 цветов, PNG позволяет хранить полноцветные изображения с 24 и даже 48 битами на пиксель.
- PNG реализует гораздо более эффективный алгоритм чересстрочности (которую в данном случае правильнее было бы называть «через-пиксельностью»). Достаточно сказать, что первый проход, дающий общее представление об изображении, занимает в PNG не одну восьмую исходного файла, как в GIF, а всего лишь одну шестьдесят четвертую – и, тем не менее, распознаваемость картинки при этом заметно лучше.
- PNG позволяет хранить полную информацию о степени прозрачности в каждой точке изображения в виде так называемого альфа-канала. Каждый пиксель PNG-файла, вне зависимости от его цвета и местоположения, может иметь любую градацию прозрачности – от нулевой (полная непрозрачность) до абсолютной невидимости.

Чтобы сгладить различия в яркости дисплеев, существующие между разными типами компьютеров (и, соответственно, между создаваемыми на них изображениями), PNG позволяет хранить в файле исходный коэффициент яркости того дисплея, на котором изображение было создано. Это дает возможность программам просмотра перед выводом на экран регулировать яркость изображения, приспособив ее к особенностям данной видеосистемы (так называемая гамма-коррекция).

Если PNG уступает GIF невозможностью хранить несколько изображений в одном файле и создавать что-либо похожее на GIF-анимацию.

**WMF** (Windows Meta File). Формат хранения векторных изображений операционной системы Windows (расширение имени файла .wmf). По определению поддерживается всеми приложениями этой системы. Однако отсутствие средств для работы со стандартизованными палитрами, принятыми в полиграфии, и другие недостатки ограничивают его применение.

**EPS** (Encapsulated Post Script). Формат описания как векторных, так и растровых изображений на языке Post Script фирмы Adobe, фактическом стандарте в области доредакционных процессов и полиграфии (расширение имени файла .eps). Так как язык PostScript фирмы Adobe является универсальным, в файле могут одновременно храниться векторная и растровая графика, шрифты, маски, параметры калибровки оборудования, цветовые профили. Для отображения на экране векторного содержимого используется формат WMF, а растрового – TIFF. Но экранная копия лишь в общих чертах отображает реальное изображение, что является существенным недостатком EPS. Действительное изображение можно увидеть лишь на выходе выводного устройства, с помощью специальных программ просмотра или после преобразования файла в формат PDF в приложениях Acrobat Reader, Acrobat Exchange.

**PDG** (Portable Document Format). Формат описания документов, разработанный фирмой Adobe (расширение файла .pdf). Хотя этот формат в основном предназначен для хранения документа целиком, его впечатляющие возможности позволяют обеспечить эффективное представление изображений. Формат является аппаратно-независимым, поэтому вывод изображений допустим на любых устройствах – от экрана монитора до фотоэкспонирующего устройства. Мощный алгоритм сжатия со средствами управления итоговым разрешением изображения обеспечивает компактность файлов при высоком качестве иллюстраций.

## **Тема 4. Общая схема цифровой обработки изображений**

В процессе обработки изображений принято выделять следующие этапы.

1. Получение исходного, «сырого» изображения.
2. Фильтрация изображения.
3. Перевод изображения в необходимую цветовую модель.
4. Форматирование и индексирование изображения.
5. Разбивка на блоки.
6. Обработка графической информации, содержащейся в блоках.
7. Последовательное сжатие.
8. Энтропийное сжатие.

Данное деление весьма условно, но дает общую картину процесса обработки. Некоторые этапы, например, 5, 7 или 8 можно пропустить. Перед каждым этапом, возможно, будет необходима специальная фильтрация.

### **Получение исходного, «сырого» изображения**

Изображения для обработки условно можно разбить на четыре класса:

- Естественные, полученные путем сканирования, захвата теле- или видео кадра, съемкой цифровой аппаратурой.
- Изображения, нарисованные с использованием графического редактора на компьютере, назовем их компьютерными рисунками.
- Трехмерные сцены, синтезированные с помощью специальных программ, таких как: САД'ы (AutoCAD, ArchiCAD), 3D-генераторы (3D Studio, LightWave) и так далее.
- Изображения-визуализация данных, полученных как результат некоторого эксперимента, опыта, измерения (энцефалограмма, сейсмографическая карта и пр.).

Естественные изображения имеют некомпьютерное происхождение. В них почти нет резких цветовых переходов. Компьютерные рисунки, как в прочем и любые другие, подразделяются на два типа: растровые и векторные. Трехмерные сцены вынесены в отдельный класс, так как в процессе их создания (например, прямой или обратной трассировкой луча, методом излучательности) можно получить дополнительные данные (характеристики прямого и диффузного отражения света, преломления объектов сцены) и использовать их при дальнейшей обработке. Изображения, как результат опыта и т.п. необходимо обработать, с целью выявить его особые характеристики, например,

выделить часть изображения лежащую в заданном спектре и т.п. В данном разделе рассматриваются в основном растровые изображения.

### **Фильтрация изображения**

Понятие фильтрации в данном случае весьма обширно, и включает в себя любое преобразование графической информации. Фильтрация может быть задана не только в виде формулы, но и в виде алгоритма, её реализующего.

Человек запоминает графическую информацию, в основном, в виде трех ее составляющих.

- *Низкочастотные составляющие изображения.* Они несут информацию о локализации объектов, составляющих изображения. Эта составляющая наиболее важна, так как связке глаз мозг уделяет первостепенное внимание.
- *Высокочастотные составляющие изображения.* Они отвечают за цветовые перепады – контуры изображения. Увеличивая их, мы повышаем резкость изображения.
- *Текстуры изображения.* Чтобы понятно объяснить, что это такое проведем небольшой эксперимент. Расслабьтесь, вспомните интерьер вашего дома, например, письменный стол. Вы знаете его очертания, местоположение, цвет – это низкочастотные характеристики; вспомнили его заостренные углы, небольшую царапину где-нибудь ближе к его кромке – это высокочастотные составляющие. Также Вы знаете, что стол деревянный, но не можете в точности рассказать обо всех мельчайших деталях его поверхности, хотя общие характеристики (коричневый с темными впадинами, две области расхождения концентрических эллипсов от сучков) – наверняка. В данном случае в скобках – описание текстуры. Можно трактовать текстуру как характеристику участков в контурах изображения.

Большинство фильтров (filters или plug-ins) предназначено для создания специальных эффектов. Использование художественных эффектов позволяет превратить обычную фотографию в настоящее произведение искусства. Фильтры и спецэффекты представляют собой небольшие программы, которые автоматически вычисляют значения и характеристики каждого пикселя изображения и затем модифицируют их в соответствии с новыми значениями.

Большинство современных графических программ поддерживают возможности применения фильтров, разработанных третьими фирмами под стандарт Adobe Photoshop. Такие модули называются подключаемыми. Использование таких модулей значительно расширяет функциональные возможности программы.

Пример использования фильтров и спецэффектов приведен на рисунке 18.



**Рисунок 18.** Результат использования фильтров: слева – исходная фотография, справа – изображение, полученное после применения фильтров и спецэффектов

При обработке изображений наиболее часто применяются такие эффекты, как размытие, увеличение резкости, тиснение, акварельный эффект.

Процесс *размытия* сводится к перераспределению цветов изображения и смягчению резких границ (см. рисунок 19).



**Рисунок 19.** Применение к фотографии (слева) фильтра Размытие (изображение справа)

Процесс увеличения *резкости*, наоборот, подчёркивает различия между цветами смежных пикселей и выделяет незаметные детали.

В результате процесса *тиснения* изображение преобразуется так, что фигуры внутри изображения выглядят словно выдавленные на металлической поверхности (см. рисунок 20).



**Рисунок 20.** Применение к фотографии (слева) фильтра Тиснение (изображение справа)



**Рисунок 21.** Применение к фотографии (слева) Акварельного эффекта (изображение справа)

*Акварельный эффект* превращает фотографическое изображение в как будто бы нарисованную красками картинку (см. рисунок 21).

Получение спецэффектов не представляет особого труда с алгоритмической точки зрения. В основе каждого из спецэффектов лежит матрица чисел, которую называют *ядром свёртки*. Матрица размером  $3 \times 3$  содержит три строки по три числа в каждой. Каждое из чисел, элементов матрицы, представляет собой цвет преобразуемого пикселя. Для преобразования каждого пикселя изображения необходимо выполнить следующий алгоритм.

1. Значение цвета пикселя умножается на число в центре ядра (элемент  $a_{22}$  матрицы).
2. Выполняется умножение восьми значений цветов пикселей, окружающих центральный пиксель, на соответствующие им коэффициенты ядра с последующим суммированием всех девяти значений. В результате получается новое значение цвета преобразуемого пикселя.
3. Для каждого пикселя изображения повторяется процесс, включающий выполнение шагов 1 и 2.

Применение данного алгоритма к изображению принято называть *фильтрацией*.

### Форматирование и индексирование изображения

Будем рассматривать изображение как прямоугольную матрицу  $A = \{a_{i,j}\}$  с  $N$  столбцами и  $M$  строками, где  $N$  – ширина изображения в пикселях,  $M$  – высота изображения в пикселях. Рассмотрим основные форматы, применяемые в компьютерной обработке изображений:

- *Черно-белый*. Каждый элемент матрицы представлен одним битом. Если он равен единице, то он отождествляется с черным цветом, если равен нулю – с белым. Это самый простой формат, он применяется при печати газет, распознавании текстов и подписей.
- *Grayscale* (градации серого). Отличие данного формата от предыдущего в том, что для каждого элемента матрицы отводится 8 битов (байт). Это позволит нам использовать  $2^8 = 256$  уровней серого цвета. Если  $a_{i,j} = 0$ , то имеем белый цвет, с возрастанием до 255 мы будем терять яркость и при  $a_{i,j} = 255$  получим черный цвет. В промежутке от 0 до 255 будут располагаться серые цвета по правилу: чем ближе значение к 255, тем чернее будет серый. Данный формат позволяет получать довольно качественные черно-белые изображения. Значения  $a_{i,j}$  содержат обратную яркость, то есть значения  $(1 - L) \times 255$ , где  $L$  – яркость, которая может быть получена, например из RGB цветowych изображений по формуле:

$$L = aR + bG + cB,$$

где  $R, G, B$  лежат в интервале  $[0;1]$ , а веса  $a, b, c$  обладают свойством:  $a + b + c = 1$ .

Иногда, для хранения Grayscale-изображений используют на точку 4-7 и 16 битов. В таком случае мы имеем 16-128 или 65536 оттенков серого цвета.

- *Многоканальные*. В данном случае  $a_{i,j}$  представлен в виде вектора с координатами используемой цветовой модели. Обычно вектор трехмерный, так как природа глаза реагирует на три различных цветовых составляющих. Каждый компонент вектора чаще всего занимает байт. Рассмотрим наиболее распространенные многоканальные форматы:

Название	Соотношение бит	1-ый компонент	2-ой компонент	3-ий компонент
RGB-Truecolor	8:8:8	Красный 0-255	Зеленый 0-255	Синий 0-255
RGB-Highcolor	5:6:5/5:5:5	Красный 0-31	Зеленый 0-31	Синий 0-31
RGB-Extended	12:12:12/ 16:16:16	Красный 0-4095/ 0-65535	Зеленый 0-4095/ 0-65535	Синий 0-4095/ 0-65535

CMY	8:8:8	Голубой 0-255	Пурпурный 0-255	Желтый 0-255
L*a*b	8:8:8	Яркость 0-255	Канал А 0-100%	Канал В 0-100%
YIQ	8:8:8	Яркость 0-255	Синфазный 0-255	Интегрированный 0-255
HLS	8:8:8	Тон 0-3600	Яркость 0-100%	Насыщенность 0-100%
HSB	8:8:8	Тон 0-3600	Насыщенность 0-100%	Яркость 0-100%

Встречаются четырех и более мерные вектора, например, модель CMYK, она применяется, когда имеются четыре основных цветовых красителя. Двумерные модели называют дуплексами. Их применяют в полиграфии, например, при печати стандартного Grayscale-изображения, реально в промышленности оно будет выполнено лишь приблизительно в 50 градациях серого, а для повышения числа градаций вводят вторую краску.

- *Индексированный.* Для уменьшения объемов изображения или для использования определенных цветов используют данный формат. Элемент матрицы  $a_{ij}$  является указателем на таблицу цветов. Число используемых цветов равно  $2K$ , где  $K$  – количество бит, используемых для хранения элемента матрицы. Цвета в указываемой таблице могут кодироваться другим числом бит. Например, в 256 цветовых режимах видеоадаптеров выбирается 256 цветов из 262144 возможных, так как выбираемые цвета представляются в RGB- формате и для каждой цветовой компоненты кодируется 6-ю битами. Существует много методов преобразования многоканальных изображений в индексированные (Error diffusion, ближайшего цвета и др.).

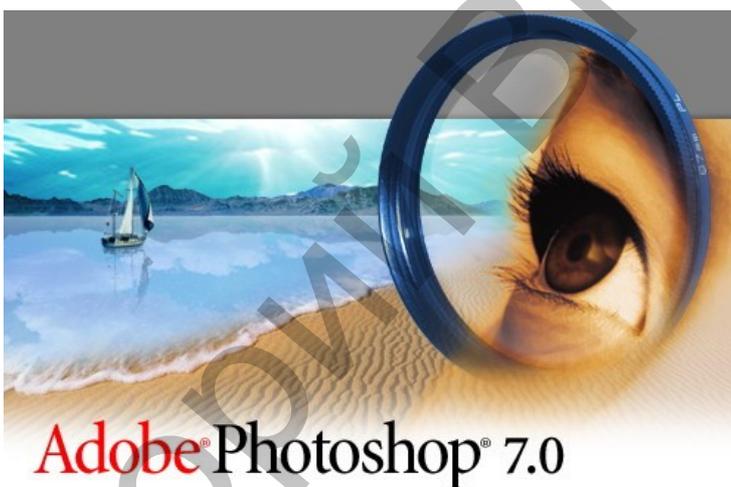
### Сжатие

Изображения в машинном представлении представляются в двумерной матрице  $N$  на  $M$ , где  $N$  – ширина,  $M$  – высота. При сканировании обычно используют разрешение от 72 до 2400 dpi. Наиболее часто – 300 dpi. Если взять лист бумаги 21/29 см с изображением и отсканировать его в модели RGB-Truecolor, то несжатое изображение будет занимать приблизительно 27300000 байтов или 26 Мбайт. Обычно в базах данных применяют изображения порядка от 320×240 до 640×480. Но и они занимают 76 до 900 Кбайт. А что, если таких изображений сотни, тысячи? Для таких больших объемов изображений существуют методы сжатия. Существуют методы сжатия, характерные только для изображений и методы сжатия, применимые для любых массивов данных.

Методы статического сжатия предполагают, что массив данных для сжатия целиком сформирован. Методы статического сжатия часто подразделяют на последовательное и энтропийное. *Последовательное сжатие* использует в работе наличие повторяющихся участков. *Энтропийное* используется с целью сокращения к минимуму избыточности информации. Последовательное применение этих методов позволяет получить хороший результат.

## Тема 5. Растровый графический редактор Adobe PhotoShop

В обширном классе программ для обработки растровой графики особое место занимает пакет Photoshop компании Adobe. По сути дела, сегодня он является стандартом в компьютерной графике, и все другие программы неизменно сравнивают именно с ним. Пакет Photoshop предоставляет профессиональные возможности в обработке изображений с целью печати, для демонстрации на экранах компьютеров и для использования в Web-дизайне.



Отличительные особенности пакета Photoshop – стабильность, надёжность, универсальность и адекватность цветопередачи. Используя возможности этой программы можно научиться рационально и грамотно исправлять недостатки цифровых изображений, например, реставрировать старые и поврежденные фотографии, доводить до требуемого уровня иллюстрации, рисунки и слайды, создавать необходимые композиции, коллажи и так далее.

Главный недостаток программы – она требует много оперативной и дисковой памяти при работе с большими изображениями и несколькими слоями.

Одна из последних разработок фирмы Adobe – программа ImageReady, которая является доработкой Photoshop для Интернета. В этом приложении развиты функции, применяемые в Web-дизайне. В ImageReady очень удобно и просто решена GIF-анимация, а также система сохранения многослойных PSD-файлов интернетовских форматов.

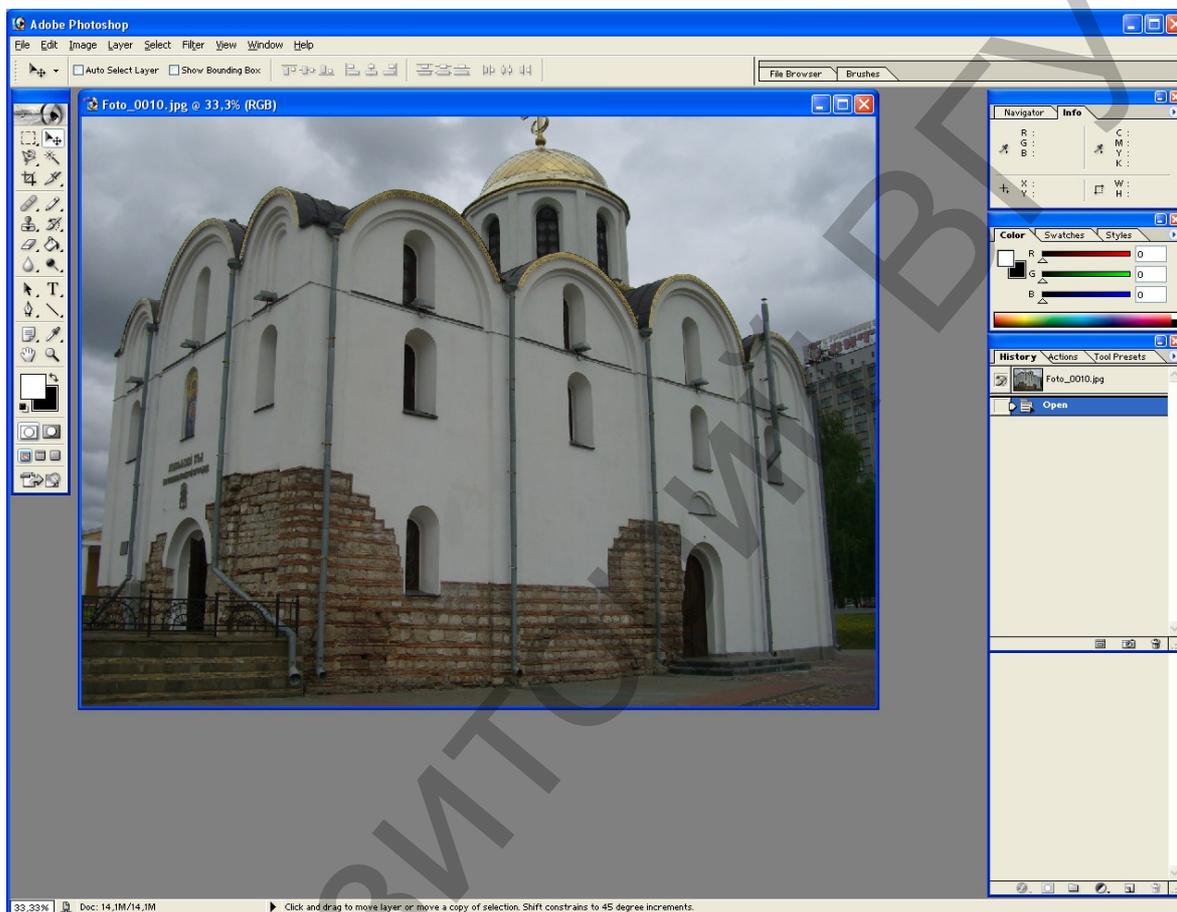
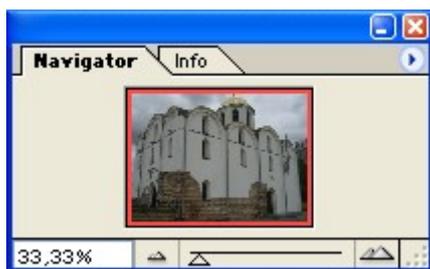


Рисунок 22. Окно редактора Photoshop

Главные элементы управления программы Adobe Photoshop сосредоточены в строке меню и панели инструментов. Особую группу составляют диалоговые окна – *инструментальные палитры*.

**Палитры.** *Палитра* (palette) – способ вывода и управления информацией посредством специальных окон, которые можно перемещать, сворачивать, разворачивать и убирать. Photoshop содержит множество палитр. Доступ к ним открывается из пункта меню *Window*.

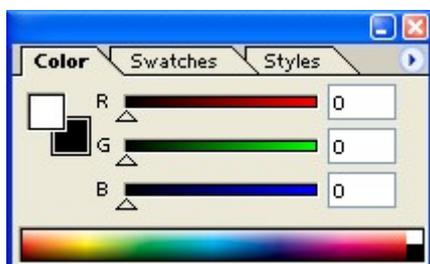
Рассмотрим характеристики некоторых инструментальных палитр.



**Navigator** (Навигатор) позволяет быстро просмотреть различные фрагменты изображения и изменить масштаб просмотра



**Info** (Инфо) информирует о текущих координатах указателя мыши, показывает размер текущей выделенной области, указывает цветовые параметры точки в разных цветовых системах. В зависимости от выбранного инструмента позволяет определить размеры, расстояния, углы поворота



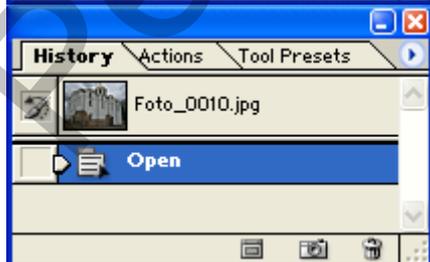
**Color** (Цвет) показывает значения текущих цветов переднего и заднего планов, которые можно отредактировать перемещением ползунков соответствующих компонентов цветовой модели



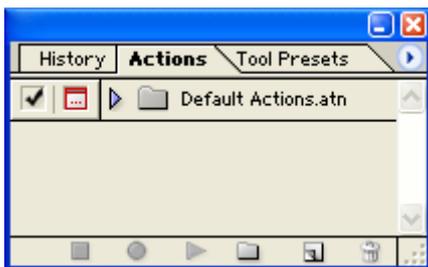
**Swatches** (Каталог) содержит набор допустимых для использования цветов. Позволяет добавлять в набор новые и удалять ненужные цвета



**Styles** (Стили) включает варианты заливок рабочей области



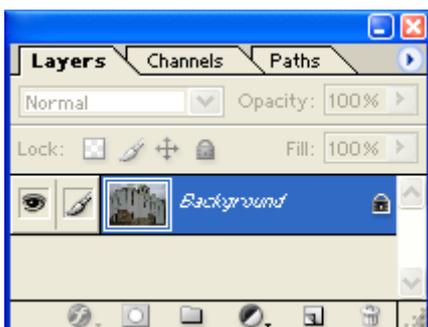
**History** (Протокол) служит для отображения всех операций, сделанных пользователем с помощью команд строки меню или инструментами из Панели инструментов (Tool Bar)



**Actions** (Операции) позволяет создавать макрокоманды. Эти макросы можно записать, выполнить, редактировать и сохранить в виде файла



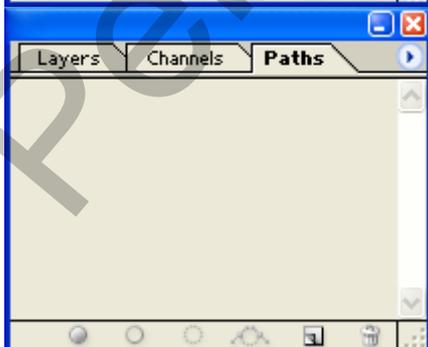
**Tool Presets** (Переустановки инструментов) предназначена для установки собственных настроек для любого инструмента с возможностью последующего сохранения его в качестве пользовательского инструмента. Доступ к пользовательским инструментам осуществляется через панель инструментов или диалоговое окно установок (Presets), в котором отражаются все предварительные установки



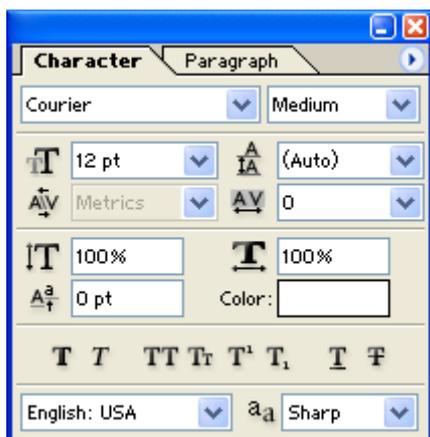
**Layers** (Слой) используется для определения параметров слоёв. Здесь перечислены все слои изображения, начиная с верхнего и заканчивая фоновым



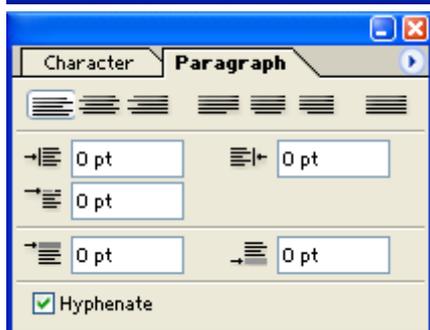
**Channels** (Каналы) предназначена для создания и редактирования каналов



**Paths** (Контур) Содержит список созданных контуров



**Character** (Символ) используется для установки и перенастройки основных атрибутов текста – таких, как гарнитура, кегль, интерлиньяж, кренинг, трекинг, отклонение от базовой линии и высота символов



**Paragraph** (Абзац) используется для работы с большими объемами текста. Среди них – настройка межсловных и межбуквенных интервалов (пробелов), задание отступов от краёв абзаца и отбивки между абзацами, изменение способов выравнивания текста и разгонка текста по всей ширине полосы

Первичное получение оригинала происходит либо через меню *File* (Файл) командой *Open* (Открыть), либо командой *Import* (Импорт). Импортом называют получение изображения от внешнего источника – сканера, цифровой фотокамеры. Связь графического редактора с внешними устройствами обеспечивается через программный интерфейс TWAIN, устанавливающий стандарт на параметры обмена данными с источниками изображений.

Панель инструментов является одним из основных средств работы с изображениями. Все инструменты, представленные на панели, делятся на *простые* и *альтернативные*. Основные инструменты видны непосредственно в главном окне программы. Значки альтернативных инструментов помечены маленьким треугольником (разворачивающая кнопка). Если при нажатой кнопке мыши задержать указатель на таком значке, откроется линейка значков с вариантами инструмента.

Все инструменты Photoshop можно условно разделить на четыре группы:

- инструменты рисования и закрашивания;
- инструменты для выделения и перемещения выделенных областей;
- текстовые инструменты;
- инструменты редактирования и ретуши.

С помощью таких инструментов, как *Кисть* (Paintbrush), *Аэрограф* (Airbrush), *Перо* (Pen) и *Карандаш* (Pencil), можно рисовать

и раскрашивать изображения на манер живописца, начинающего с чистого «холста». Все названные инструменты допускают настройку целого ряда параметров в соответствии с потребностями пользователя: ширины мазка, степени размывания кромок, жесткости и мягкости.

Вероятно, наибольшую мощь программе Photoshop дает использование слоев, предоставляющих возможность объединять несколько изображений и создавать монтажи, работая лишь над одной из составных частей изображения. Каждый слой полностью независим от остальных, что позволяет изменять степень его прозрачности и редактировать, оставляя другие слои изображения неприкосновенными.

### **Подготовка изображения для печати**

В Adobe Photoshop заложены возможности получения на экране информации, сигнализирующей о наличии в изображении цветов, не поддерживаемых моделью CMYK.



**Рисунок 23.** Изображение в цветовом пространстве RGB

Если изображение создано и обрабатывалось с использованием пространства RGB (см. рисунок 23), необходимо проверить наличие в нём цветов, выходящих за пределы печатного диапазона CMYK. Для этого необходимо воспользоваться командой *View / Proof Setup / Working CMYK*.

Далее необходимо выделить все области изображения, где используются цвета из модели RGB, но отсутствующие в таком же виде в пространстве CMYK: команда *Select / Color Range*. В диалоговом окне *Color Range* из списка *Select* необходимо выбрать *Out Of Gamut*. Области с отсутствующими в CMYK-модели цветами будут выделены (см. рисунки 24, 25).



**Рисунок 24.** Выделение областей цвета, выходящего за пространство цветов RGB



**Рисунок 25.** Выделение серым цветом областей, выходящих за пределы диапазона RGB

Для того, чтобы оценить, как будет выглядеть изображение при печати CMYK, необходимо воспользоваться командой *View / Prof Colors*.

Для того, чтобы все цвета, выходящие за пределы диапазона CMYK-модели, отображались серым цветом, используется команда *View / Gamut Warning*.

Если команда *Gamut Warning* активна, можно использовать выделенную область и команду *Image / Adjustments / Hue/Saturation* для устранения проблем, связанных с цветами, выходящими за пределы диапазона. Перемещая ползунок *Saturation* влево (тем самым уменьшая насыщенность цветов), можно уменьшить области цветов, лежащих вне диапазона CMYK. Однако, если чрезмерно использовать уменьшение насыщенности цветов, то изображение получится очень тусклым.

## Тема 6. Художественное оформление текста в векторных редакторах

Когда речь идёт о шрифтах, подразумевается, как правило, что речь идёт о шрифтах TrueType или PostScript, то есть о шрифтах в контурном описании. В *векторных шрифтах* очертания символов описываются отрезками прямых.

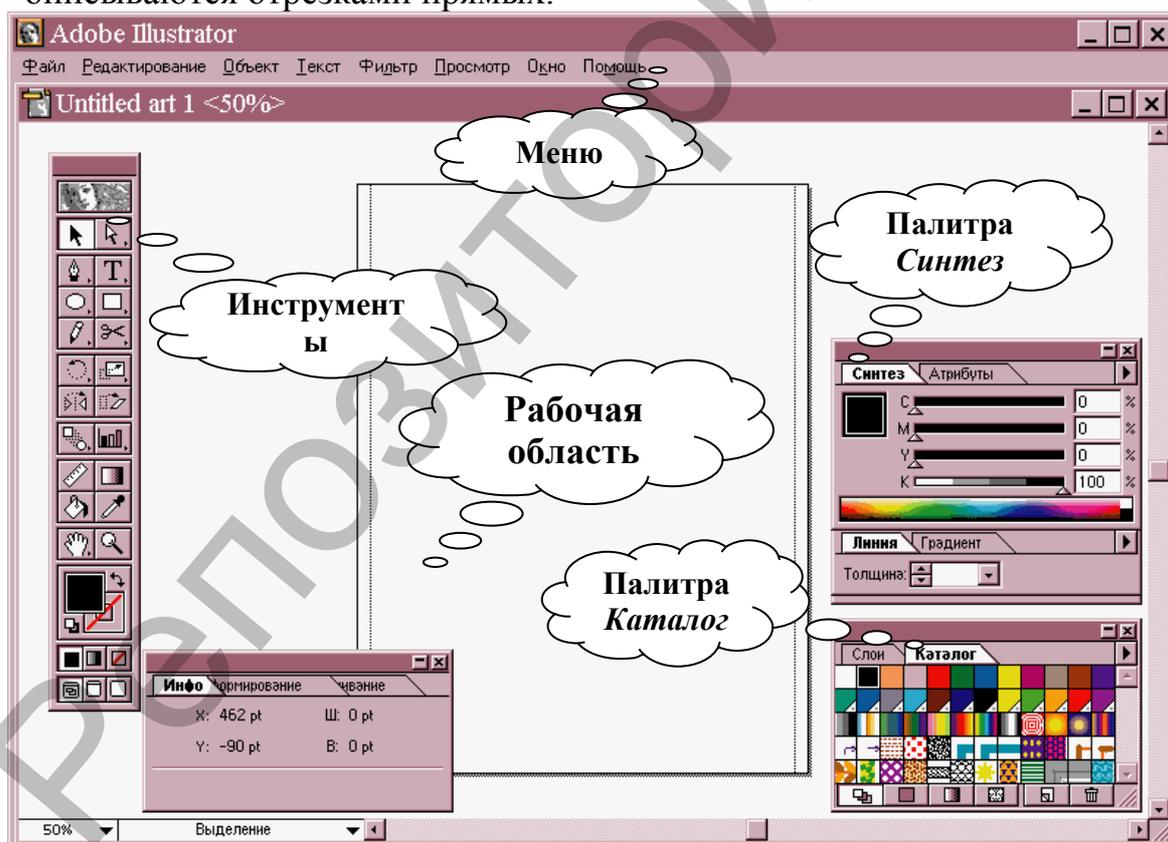


Рисунок 26. Окно программы Adobe Illustrator

Имеем ли мы дело с истинными векторными шрифтами, шрифтами формата PostScript или TrueType либо с текстом, преобразованным в кривые, – всегда речь идёт об определённом способе математического описания формы. Типы кривых,

используемых для описания, различаются. Для векторных шрифтов это прямые, для TrueType – кривые второго порядка, для PostScript-шрифтов – кривые третьего порядка. Для пользователя совершенно не важен тип описания, существенно лишь то, что во всех этих случаях исходное изображение не имеет разрешения. Конечно, принтер и монитор воспроизводят не непрерывную линию, а набор точек. Поэтому при выдаче на экран или печати происходит *растеризация*, то есть перевод описания формы как набора линий в описание её как совокупности отдельных точек.

### **Работа в векторном редакторе Adobe Illustrator**

Этот редактор относится к группе наиболее предпочитаемых профессионалами инструментов для оформления текста. Adobe Illustrator является векторным редактором, поэтому большую часть художественных эффектов здесь можно применять только к тексту, преобразованному в кривые. Редактировать текст после этого невозможно. По этой причине Adobe Illustrator – не самый удобный инструмент для оформления надписей. На рисунке 26 представлено окно редактора Adobe Illustrator.

#### **Набор текста**

Для набора текста предназначено несколько инструментов: *Текст*, *Текст в области*, *Текст вдоль контура*, *Вертикальный текст*, *Вертикальный текст в области*, *Вертикальный текст вдоль контура*. Вся линейка инструментов появится на экране, если нажать кнопку первоначально присутствующего инструмента (по умолчанию это *Текст*).

Для набора коротких надписей используется точечный текст. Для этого выбирается инструмент *Текст*, выполняется щелчок в нужной точке и начинается набор текста. Для набора большого участка текста выбирается тот же инструмент *Текст*, в нужном месте чертится прямоугольник. Текст будет вводиться в созданную рамку с автоматическим переводом строки. Сама рамка на печать не выводится.

Изменение параметров символов и абзацев производится из меню *Текст*. Для удобства работы можно вывести на экран палитру *Символ* (для этого можно воспользоваться комбинацией клавиш **Ctrl + T** или командой *Текст / Символ*). В палитре настраивается гарнитура и начертание (ниспадающее меню *Шрифт*),



кегель и итерлиньяж (два окна в среднем ряду), кернинг и трекинг (нижний ряд).

В палитре *Абзац* устанавливается выравнивание, абзацные отступы, «красная строка».

В меню *Текст* предлагается множество возможностей, характерных, скорее для издательской системы, чем для графического редактора: поиск и замена текста, поиск шрифта, обтекание текстом иллюстрации и так далее. В последних версиях Adobe Illustrator есть даже проверка русской орфографии и расстановка переносов в русском тексте.

## Оформление текста, не преобразованного в кривые

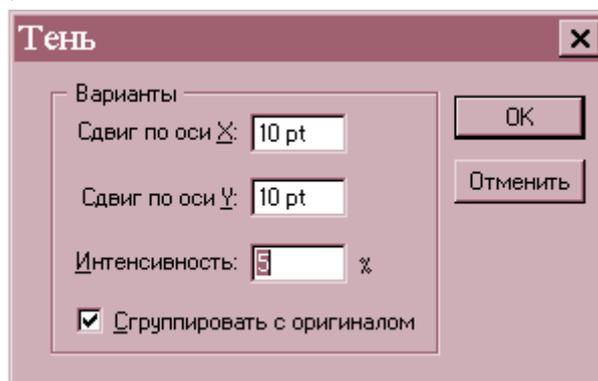
### Асимметричное масштабирование

Инструментом *Выделение*  выделите текст. Перейдите к инструменту *Размер*  Курсор примет форму крестика. Это означает, что можно установить неподвижную точку объекта. Совместите эту точку с началом строки и выполните однократное нажатие. Курсор примет вид стрелки. Для растяжения текста по горизонтали потяните по горизонтали за конец строки. Для растяжения в обоих направлениях потяните за верхнюю или боковую сторону. Чтобы растянуть текст на заданный процент, используйте меню масштабирования, которое появится после двойного щелчка по инструменту *Размер*.

### Тень

Данный эффект выполняется автоматически при выборе фильтра *Тень*. Для этого воспользуйтесь командой *Фильтр / Стилизация / Тень*.

Текст, разумеется, предварительно должен быть выделен.



### Отражение в воде

Для копирования объекта переместите его при нажатой клавише **Alt**. Переворачивание производится при помощи инструмента *Зеркало* – Щёлкните по кнопке инструмента, поставьте фиксированную точку под надписью и протяните стрелку при нажатой кнопке **Shift** (это обеспечит горизонтальность оси). Как только кнопка будет отпущена, надпись перевернётся.

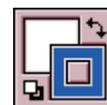
Для наклона используйте инструмент *Наклон*. Установите неподвижную точку в начало текста и наклоняйте его, взявшись

стрелкой за противоположный угол. Отражённые буквы можно залить цветом при помощи инструмента *Заливка*. Цвета для заливки настраиваются в меню *Синтез*. К сожалению текст, не преобразованный в линии, заполнить цветовым градиентом нельзя.



### Полые буквы и оконтуривание снаружи

Для получения этого эффекта выделите текст, щёлкните по пиктограмме *Линия* в нижней части меню инструментов. Она должна переместиться на передний план, перед пиктограммой *Заливка*. После этого перейдите к палитре *Синтез* и задайте цвет контура, в окне *Линия* установите толщину контура. Затем щёлкните по пиктограмме *Заливка* и в окне *Синтез* задайте белый цвет заливки.



### Текст в оболочке

В данном случае не происходит модификации контура букв, текст просто заполняет контур.



С помощью инструментов *Перо* или *Карандаш* создайте контур и замкните его. Для этого начальную и конечную точки выделите инструментом *Частичное выделение* и выполните команду *Объект / Контур / Соединить*. Можно воспользоваться уже замкнутыми контурами – эллипсом или прямоугольником, построение которых осуществляется одноименными инструментами.

Среди текстовых инструментов выберите инструмент *Текст в области* или *Вертикальный текст в области*. Щёлкните по контуру и введите текст. Затем выделите его и, если необходимо, выровняйте по центру. Для этого воспользуйтесь командой *Текст / Абзац*. Контур в данном случае не является отдельным объектом – его нельзя залить цветом, он не виден на печати и так далее.

### Текст вдоль линии

Для создания данного эффекта нарисуйте линию, используя инструменты *Перо* или *Карандаш*. Из подменю *Инструменты* для работы с текстом выберите *Текст вдоль контура* или *Вертикальный текст вдоль контура*.

Щёлкните по кривой и наберите текст. Если необходимо, выделите текст и измените его параметры.

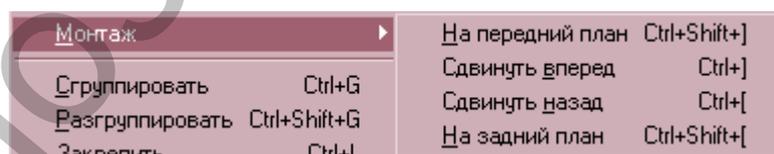
### Цветовые эффекты

Для заполнения текста цветом можно использовать готовые орнаменты из палитры *Каталог*. Выделите текст и щёлкните по нужному орнаменту в палитре *Каталог*. Таким способом оформлены следующие примеры надписей.



### Дополнительные возможности редактора

Для регулирования порядка наложения объектов друг на друга используйте варианты, предложенные в меню команды *Объект / Монтаж*, которые представлены на рисунке.



### Оформление преобразованного в кривые текста

Перечисленные ниже эффекты можно применять только к тексту, преобразованному в кривые. Для этого используется команда *Текст / Преобразовать в контуры*. При этом каждая буква становится отдельным объектом. Чтобы работать с заголовком, как с целым, его необходимо сгруппировать. Для этого используется команда *Объект / Сгруппировать*.

### Цветовые эффекты

Для заполнения текста цветом можно использовать градиентные заливки из палитры *Каталог*. Выделите текст, предварительно

преобразованный в кривые, и выберите вариант градиента в палитре *Каталог*. Таким способом оформлена надпись «Ну, погоди!».

**НУ, ПОГОДИ!**

### Скручивание

Преобразованный в контуры текст можно скручивать. Кнопка  (Скручивание) скрыта за кнопкой *Вращение* и появляется на экране при длительном нажатии последней. Как обычно, ставим неподвижную точку, затем переводим курсор в противоположный конец надписи и вращаем её. Если неподвижную точку расположить в середине надписи, то образуются S-образные формы, если неподвижную точку совместить с началом надписи – дугообразные. Ниже приведены примеры скручивания.

Текст до скручивания (гарнитура – *Amelia\_DG*, начертание – *Regular*, размер – 48 pt, с тенью):

Реклама, просто реклама...

Неподвижная точка была установлена в середине надписи:

Реклама, просто реклама...

Неподвижная точка была установлена в конце надписи:

Реклама, просто реклама...

Неподвижная точка была установлена в начале надписи:

Реклама, просто реклама

### Фильтры

К преобразованному в кривые тексту можно применять фильтры, как к графическому объекту. Группа фильтров *Искажение* включает: *Втягивание и раздувание*, *Зигзаг*, *Каракули и помарки*, *Огрубление*, *Свободное искажение* и *Скручивание*.

Исходный текст (гарнитура – KabelSTT Book, начертание – Bold):

Фирма "У Карлсона"

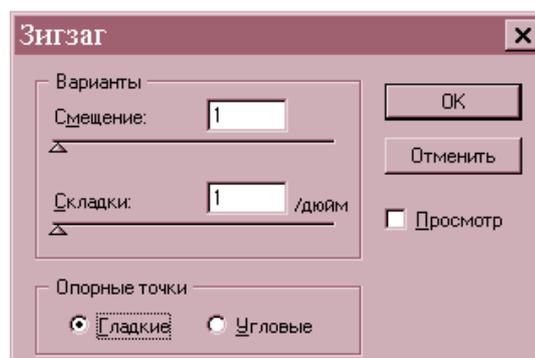
Текст после применения фильтра *Втягивание*:

Фирма "У Карлсона"

Текст после применения фильтра *Раздувание*:

Фирма "У Карлсона"

Фильтр *Зигзаг* устанавливает на контуре дополнительные точки и делает его неровным. Смещение следует устанавливать минимальным, иначе текст после такого искажения станет неразборчивым. Угловые опорные



точки делают контур зазубренным, гладкие – создают сглаженные выпуклости. Приведём пример.

Исходный текст (гарнитура – AGCrownStyle, начертание – Normal, размер – 48):

**ЗИГЗАГ УДАЧИ**

Текст после использования фильтра *Зигзаг*, в случае, когда заданы гладкие опорные точки:

**ЗИГЗАГ УДАЧИ**

С помощью фильтра *Каракули и помарки* можно создать иллюзию письма дрожащей рукой. При аккуратном подборе вертикальных и горизонтальных искажений оформление смотрится довольно реалистично.

Исходный текст (гарнитура – ArtScript):

*Здравствуй, милый дедушка!*

Текст после применения фильтра *Каракули*:

*Здравствуй, милый дедушка!*

Текст после дополнительного использования эффекта *Помарки*:

*Здравствуй, милый дедушка!*

Использование фильтра *Огрубление* позволяет получить результат, довольно схожий с результатом, который достигается с помощью предыдущего фильтра. Приведём пример применения фильтра *Огрубление*.

Исходный текст (гарнитура – Bangkok):

**Сказка - ложь, да в ней намёк**

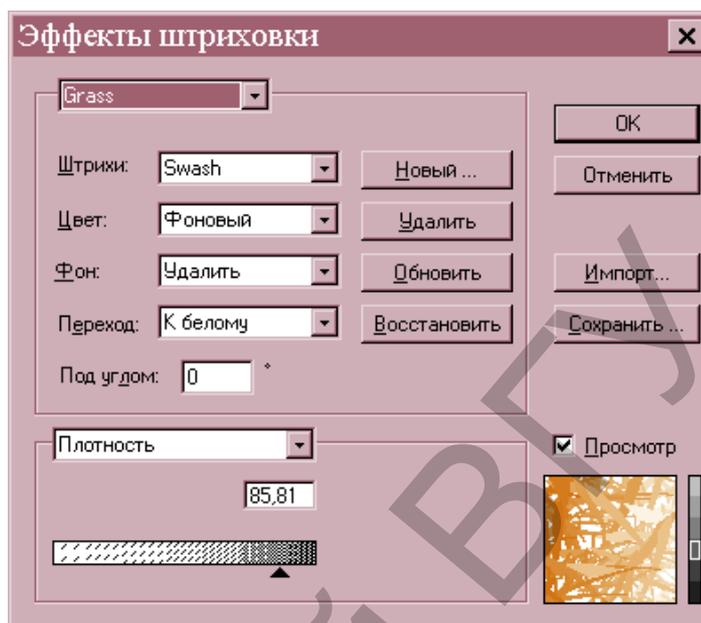
Текст после применения фильтра *Огрубление*:

**Сказка - ложь, да в ней намёк**

Результат применения фильтра *Скручивание* примерно совпадает с тем, что достигается при применении инструмента *Скручивание*. Разница в том, что неподвижная точка всегда привязана к центру объекта.

Интересные результаты даёт использование фильтра *Штриховка*. В меню фильтра можно настраивать стиль заполнения, тип штриха (линии различных типов, точки и так далее), плотность заполнения и насыщенность самих штрихов.

Для примера возьмём следующий фрагмент текста (гарнитура – CyrillicHeavy):



# Осенний дождь

Результаты применения фильтра *Штриховка*:

Осенний  
дождь  
стучит  
в окно

## Тема 7. Графика в Web-дизайне

Первые сайты, появившиеся еще в эпоху становления сети Интернет, представляли собой строгие текстовые документы, главным и единственным компонентом которых являлось их содержательное информационное наполнение. Сегодня трудно себе представить Web-страницу без графики, достаточно сложно встретить во Всемирной сети неиллюстрированный проект (чаще всего это либо литературные тексты, либо технические спецификации и справочники).

Графические компоненты Web-страниц по назначению можно условно разделить на три обширные категории:

- *иллюстративная графика*, которая включает дополняющие текст фотографии, пояснительные рисунки, чертежи и схемы;
- *функциональная графика*, которая представляет собой элементы управления сайтом (кнопки навигации, элементы интерактивных форм и счетчики);
- *декоративная графика* – элементы дизайна страницы, включенные в нее «для красоты» и не несущие информационной нагрузки (фоновые рисунки, линии-разделители, выполненные в виде графических файлов заголовки и многое другое).

Пока ещё большинство пользователей Интернета на нашей планете применяют соединение по коммутируемым телефонным каналам при помощи модема. Поскольку это весьма медленный способ связи, время загрузки графического изображения в клиентский браузер в данном случае должно быть как можно меньшим. Именно поэтому в сети Интернет используют два наиболее популярных стандарта, в которых хранится 90% всей графики, представленной в сети, это GIF и JPEG. В них заложены различные алгоритмы сжатия изображения с потерей качества, благодаря использованию которых удается значительно уменьшить размер целевого файла.

Благодаря возможностям алгоритма сжатия LZW стандарт GIF позволяет значительно сокращать объем итогового графического файла по сравнению с исходным изображением. Достигается это методом смешения сходных оттенков в один. Если, например, в составе рисунка имеется участок, состоящий из нескольких сходных полутонов, к примеру, голубого, светло-голубого и темно-голубого цвета, они будут кодированы одним оттенком – голубым. Информация об изображении в файле стандарта GIF записывается построчно, то есть представляет собой массив описаний строк высотой в один пиксель. Именно это свойство GIF, а также то, что данный формат оперирует фиксированной, так называемой индексированной палитрой, причем число цветов в этой палитре не превышает 256, явилось основой для появления двух простых правил, применяющихся в современном Web-дизайне.

Стандарт GIF используется в документах HTML только для отображения так называемой бизнес-графики: диаграмм, логотипов, кнопок, разделительных линий, других элементов оформления страницы. Для размещения на Web-сайте фотографий, репродукций картин и изображений с большим количеством цветов и цветовых переходов используется стандарт JPEG. Другими словами данный закон «Web-мастерства» можно сформулировать так: *если*

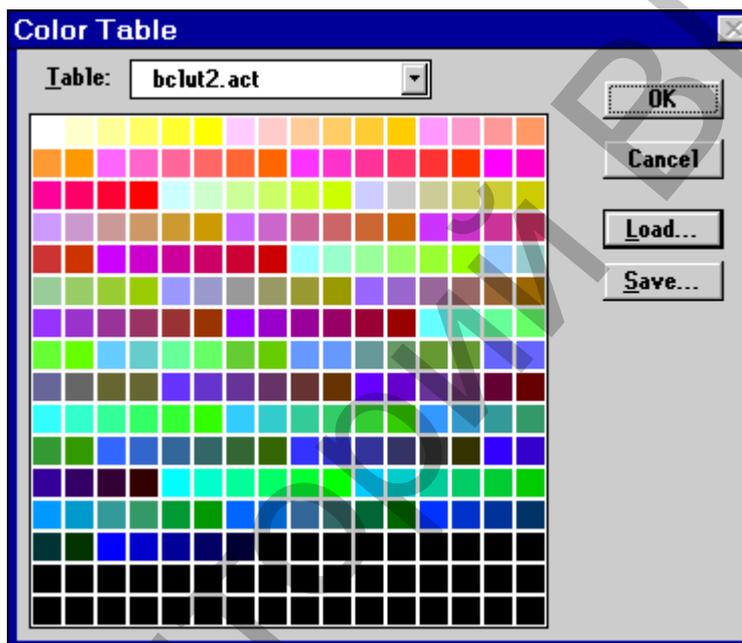
*изображение рисованное, его следует представлять в стандарте GIF, во всех остальных случаях лучше воспользоваться JPEG.*

Подготавливая рисунок для сохранения его в формате GIF, необходимо избегать следующих художественных приемов: градиентных заливок, размытий, постепенных цветовых переходов с множеством оттенков, а также графических фильтров, обеспечивающих неравномерное смешение нескольких цветов на одном участке изображения, например, эффектов изменения интенсивности освещения, подобных фильтру «блик» редактора Adobe Photoshop.

Это правило продиктовано тем, что алгоритм замещения схожих оттенков одним в формате GIF далеко не всегда корректно работает (точнее было бы сказать, работает некорректно почти всегда). Поэтому участки со множеством различных оттенков на небольшом физическом пространстве рисунка после сохранения изображения в индексированной палитре будут выглядеть смазанными и «грязными». Этого можно избежать, применяя в своей иллюстрации по возможности однотонные и контрастные цвета. Одно из замечательных свойств стандарта GIF – его уникальная особенность, названная разработчиками «interlace», или, по-русски, «чересстрочность». Она позволяет загружать картинку с сервера в клиентский браузер не целиком, а частями, причем процедура считывания файла выглядит следующим образом: сначала на экране отображаются первая, пятая и десятая строки, составляющие изображение, затем – вторая, шестая и одиннадцатая и так далее. Таким образом, для пользователя создается иллюзия постепенной загрузки графического элемента: картинка как бы медленно «проявляется» на странице, что иногда бывает очень полезно при включении в документ изображений большого размера – психологически зрителю легче дождаться полной прорисовки иллюстрации, чем несколько минут скучать перед пустым экраном. К тому же еще до полной загрузки файла пользователь может оценить приблизительное содержимое картинки и решить, стоит ли ему дожидаться ее полной прорисовки или нет.

Вторая особенность связана с составом цветовых палитр. Цветовая палитра RGB зависит от выбранного цветового разрешения – 246, 16 или 8 бит. В последнем случае цветовая палитра называется индексной, потому что каждый цветовой оттенок кодируется одним числом, которое выражает не цвет пикселя, а индекс (номер) цвета. Таким образом, к файлу цветного изображения, созданного в индексной палитре, должна быть приложена сама палитра, так как программе обработки компьютерной графики неизвестно, какая именно палитра была использована.

Изображения, подготавливаемые для публикации в Интернете, принято создавать в так называемой безопасной палитре цветов. Она является вариантом рассмотренной выше индексной палитры. Но так как файлы изображений в Web-графике должны иметь минимальный размер, необходимо было отказаться от включения в их состав индексной палитры. Для этого была принята единая фиксированная палитра цветов, названная «безопасной», то есть обеспечивающей правильное отображение цветов на любых устройствах (в программах), поддерживающих единую палитру. Безопасная палитра



содержит всего 216 цветов, что связано с ограничениями, накладываемыми требованиями совместимости компьютеров.

**Рисунок 27.** Единая фиксированная палитра цветов, «безопасная палитра»

## Литература

1. Буковецкая О.А. Дизайн текста: шрифт, эффекты, цвет – М.: ДМК, 2000.
2. Залогова Л.А. Компьютерная графика. Практикум. – М.: Лаборатория Базовых Знаний, 2005.
3. Миронов Д.Ф. Компьютерная графика в дизайне. – СПб.: Питер, 2004.
4. Петров М.Н., Молочков В.П. Компьютерная графика. – СПб.: Питер, 2004.
5. Пореев В.Н. Компьютерная графика. – СПб.: БХВ-Петербург, 2002. – 432 с.
6. Тозик В.Т. 3DS MAX. Трехмерное моделирование и анимация. – СПб.: БХВ., 2005.

7. Коцюбинский А.О., Грошев С.В. Компьютер для художника. – М.: ТРИУМФ, 1999.

Репозиторий ВГУ