

Урок №1.

Векторная и растровая графика

Информация взята с сайта: <http://www.mkgt.ru/files/material-static/practicum/teoriya/t1.htm>

Методы представления графических изображений

Растровая графика

Растровое изображение представляет из себя мозаику из очень мелких элементов — пикселей. Растровый рисунок похож на лист клетчатой бумаги, на котором каждая клеточка закрашена определённым цветом, и в результате такой раскраски формируется изображение.

Принцип растровой графики чрезвычайно прост. Он был изобретен и использовался людьми за много веков до появления компьютеров. Во-первых, это такие направления искусства, как мозаика, витражи, вышивка. В любой из этих техник изображение строится из дискретных элементов. Во-вторых, это рисование «по клеточкам» — эффективный способ переноса изображения с подготовительного картона на стену, предназначенную для фрески. Суть этого метода заключается в следующем. Картон и стена, на которую будет переноситься рисунок, покрываются равным количеством клеток, затем фрагмент рисунка из каждой клетки картона тождественно изображается в соответствующей клетке стены.

Создание изображения в растровом графическом редакторе (**Paint, Fractal Design Painter, Corel Photo-PAINT, Adobe PhotoShop**) похоже на работу художника, когда он пишет картину на настоящем холсте настоящими красками. Здесь компьютерный художник водит «кистью» — курсором мыши по «электронному полотну» — экрану, закрашивая каждый из пикселей рисунка в нужный цвет. Таким образом каждому пикселю присваивается цвет. Этот цвет закрепляется за определённым местом экрана и как бы «высыхает» подобно тому, как высыхает краска на настоящем холсте. Перемещение фрагмента изображения «снимает» краску с электронного холста и, следовательно, разрушает рисунок.

Растровая графика (Рис. 1.) работает с сотнями и тысячами пикселей, которые формируют рисунок. Пиксели «не знают», какие объекты (линии, эллипсы, прямоугольники и т. д.) они составляют.



Рис. 1. Растровое изображение (вверху увеличенный глаз)

В компьютерной графике термин «пиксель», вообще говоря, может обозначать разные понятия:

- наименьший элемент изображения на экране компьютера;
- отдельный элемент растрового изображения;
- точка изображения, напечатанного на принтере.

Поэтому, чтобы избежать путаницы, будем пользоваться следующей терминологией:

- **видеопиксель** — наименьший элемент изображения на экране;
- **пиксель** — отдельный элемент растрового изображения;
- **точка** — наименьший элемент, создаваемый принтером.

При этом для изображения одного пикселя на экране компьютера может быть использован один или несколько видеопикселей .

Экран дисплея разбит на фиксированное число видеопикселей , которые образуют графическую сетку (растр) из фиксированного числа строк и столбцов. Размер графической сетки обычно представляется в форме $N \times M$, где N — количество видеопикселей по горизонтали, а M — по вертикали. На современных дисплеях используются, например, такие размеры графической сетки: 640 x 480, 800 x 600, 1024 x 768, 1240 x 1024 и др. Изображение на экране дисплея создается путем избирательной засветки электронным лучом определенных видеопикселей экрана. Чтобы изображение могло восприниматься глазом, его необходимо составить из сотен или тысяч видеопикселей , каждый из которых должен быть подсвечен.

Достоинства растровой графики

1. Если размеры пикселей достаточно малы (приближаются к размерам видеопикселей), то растровое изображение выглядит не хуже фотографии (рис. 2). Таким образом, растровая графика эффективно представляет изображения фотографического качества.



Рис. 2. Растровое изображение, полученное с помощью цифровой фотокамеры

2. Компьютер легко управляет устройствами вывода, которые используют точки для представления отдельных пикселей. Поэтому растровые рисунки могут быть легко распечатаны на принтерах.

Недостатки растровой графики

1. В файле растрового изображения запоминается информация о цвете каждого видеопикселя в виде комбинации битов. Бит — наименьший элемент памяти компьютера, который может принимать одно из двух значений: включено или выключено. Наиболее простой тип изображения имеет только два цвета (например, белый и чёрный). В этом случае каждому видеопикселю соответствует один бит памяти (2^1). Если цвет видеопикселя определяется двумя битами, то мы имеем четыре (2^2) возможных комбинаций значений *включено /выключено*. Используя для значения выключено символ 0, а для включено — 1, эти комбинации можно записать так: 00, 01, 10, 11. Четыре бита памяти позволяют закодировать 16 (2^4) цветов, восемь бит — 2^8 или 256 цветов, 24 бита — 2^{24} или 16777216 различных цветовых оттенков.

Простые растровые картинки занимают небольшой объём памяти (несколько десятков или сотен килобайт). Изображения фотографического качества часто требуют несколько мегабайт. Например, если размер графической сетки — 1240 x 1024, а количество используемых цветов — 16777216, то объём растрового файла составляет около 4 Мб, так как информация о цвете видеопикселей в файле занимает

$1240 \times 1024 \times 24 = 30474240$ бит или
 $30474240 \text{ бит} : 8 = 3809280$ байт или
 $3809280 \text{ байт} : 1024 = 3720$ Кб или
 $3720 \text{ Кб} : 1024 = 3,63$ Мб.

Таким образом, для хранения растровых изображений требуется большой объём памяти.

Самым простым решением проблемы хранения растровых изображений является увеличение ёмкости запоминающих устройств компьютера. Современные жесткие и оптические диски предоставляют значительные объёмы памяти для данных. Обратной стороной этого решения является стоимость, хотя цены на эти запоминающие устройства в последнее время заметно снижаются.

Другой способ решения проблемы заключается в **сжатии графических файлов**, т. е. использовании программ, уменьшающих размеры файлов растровой графики за счет изменения способа организации данных. Существует несколько способов сжатия графических данных. В простейшем из них последовательность повторяющихся величин (в нашем случае — набор бит для представления видеопикселей) заменяется парой — единственной величиной и количеством её повторений. На рис. 3 показано, как подобным образом может быть сжата одна строка чёрно-белого растрового рисунка.

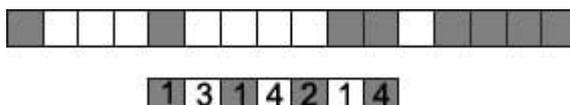


Рис. 3. Сжатие последовательности пикселей

Такой метод сжатия лучше всего работает с изображениями, которые содержат большие области однотонной закрашки, но намного хуже с его помощью сжимаются фотографии, так как в них почти нет длинных строк из одинаковых пикселей. Сильно насыщенные узорами изображения хорошо сжимаются методом **LZ W** (назван так по первым буквам фамилий его разработчиков — Lempel, Ziv и Welch). Объединённая группа экспертов по фотографии (Joint Photographic Experts Group) предложила метод **JPEG** для сжатия изображений фотографического качества.

2. Растровое изображение после масштабирования или вращения может потерять свою привлекательность. Например, области однотонной закрашки могут приобрести странный («муаровый») узор; кривые и прямые линии, которые выглядели гладкими, могут неюжи -

данно стать пилообразными. Если уменьшить, а затем снова увеличить до прежнего размера растровый рисунок, то он станет нечётким и ступенчатым (рис. 4), а закрашенные области могут быть искажены. Причина в том, что изменение размеров растрового изображения производится одним из двух способов:

- все пиксели рисунка одинаково изменяют свой размер (одновременно становятся больше или меньше);
- пиксели добавляются или удаляются из рисунка (это называется *выборкой пикселей в изображении*).

При первом способе масштабирование изображения не меняет количество входящих в него пикселей, но изменяется количество элементов (видеопикселей или точек), необходимых для построения отдельного пикселя (рис. 5), и при увеличении рисунка «ступенчатость» становится всё более заметной — каждая точка превращается в квадратик.

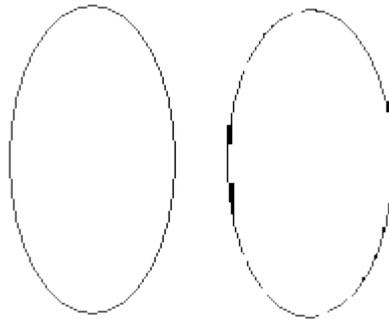
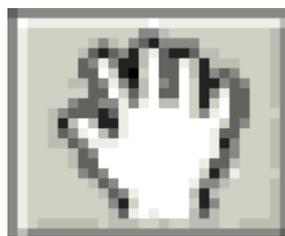
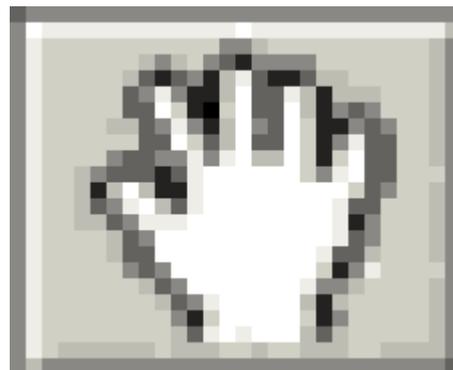


Рис. 4. Эффект, появляющийся при уменьшении, а затем увеличении растрового изображения (рисунок получен в графическом редакторе Paint)



Исходное изображение



Увеличенное изображение

Рис. 5. При изменении размеров изображения количество входящих в него пикселей не меняется

Выборка же растрового изображения может быть сделана двумя способами. Во-первых, можно просто продублировать или удалить необходимое число пикселей. Во-вторых, с помощью определенных вычислений программа может создать пиксели другого цвета, определяемого первоначальным пикселем и его окружением. При этом возможно исчезновение из рисунка мелких деталей и тонких линий, появление «муарового» узора или уменьшение резкости изображения (размытие).

Так как графический редактор Paint (стандартная программа WINDOWS) является растровым, то в нем легко продемонстрировать результаты масштабирования изображений и объяснить причины возникающих искажений.

Итак, растровые изображения имеют очень ограниченные возможности при масштабировании, вращении и других преобразованиях.

Векторная графика

В векторной графике изображения строятся из простых объектов — прямых линий, дуг, окружностей, эллипсов, прямоугольников, областей однотонного или изменяющегося цвета (заполнителей) и т. п., называемых примитивами. Из простых векторных объектов создаются различные рисунки (рис. 6).

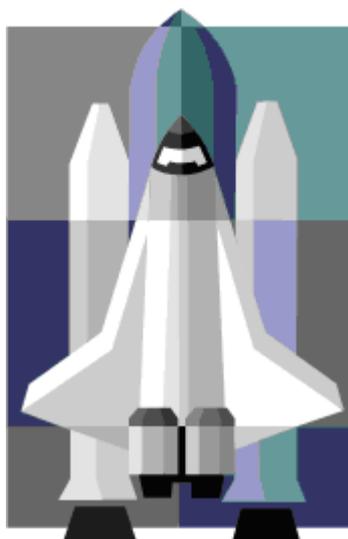


Рис. 6. Векторные изображения, созданные путем комбинации окружностей, прямоугольников, прямых и кривых линий

Комбинируя векторные объекты-примитивы и используя закрашку различными цветами, можно получить и более интересные иллюстрации (рис. 7).



Рис. 7. Рисунок, составленный из векторных примитивов

В трёхмерной компьютерной графике могут использоваться «пространственные» примитивы — куб, сфера и т. п.

Векторные примитивы задаются с помощью описаний. Например:

рисовать линию от точки A до точки B;

рисовать эллипс, ограниченный заданным прямоугольником.

Для компьютера подобные описания представляются в виде команд, каждая из которых определяет некоторую функцию и соответствующие ей параметры. Символические команды для приведённых выше примеров описаний в векторном формате **WMF** (Windows Metafile) записываются так:

MOVETO X1 , Y1	Установить текущую позицию (X1 , Y 1).
LINETO X2, Y2	Нарисовать линию от текущей позиции до позиции (X 2, Y 2).
ELLIPSE X3 , Y3 , X4, Y4	Нарисовать эллипс, ограниченный прямоугольником, где (X3, Y3) — координаты левого верхнего, а (X4, Y4) — правого нижнего угла этого прямоугольника.

Информация о цвете объекта сохраняется как часть его описания, т. е. в виде векторной команды (сравните: для растровых изображений хранится информация о цвете каждого видеопикселя).

Векторные команды сообщают устройству вывода о том, что необходимо нарисовать объект, используя *максимально возможное число элементов* (видеопикселей или точек). Чем больше элементов используется устройством вывода для создания объекта, тем лучше этот объект выглядит.

Кто же составляет последовательность векторных команд?

Для получения векторных изображений, как правило, используются программы иллюстративной графики (**CorelDraw, Adobe Illustrator, Macromedia Freehand**), которые широко применяются в области дизайна, технического рисования, а также для оформительских работ. Эти векторные программы предоставляют в распоряжение пользователя набор инструментов и команд, с помощью которых создаются рисунки. *Одновременно с процессом рисования специальное программное обеспечение формирует векторные команды, соответствующие объектам, из которых строится рисунок.*

Вероятнее всего, что пользователь такой программы никогда не увидит векторных команд. Однако знания о том, как описываются векторные рисунки, помогают понять достоинства и недостатки векторной графики.

Файлы векторной графики могут содержать растровые изображения в качестве одного из типов объектов (рис. 8).

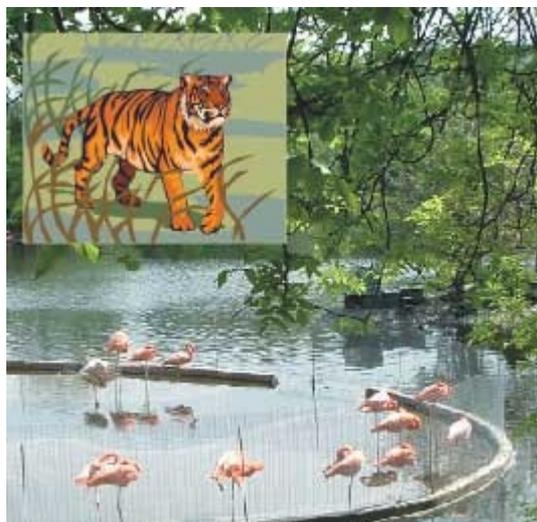


Рис. 8. Фотография, вставленная в документ векторного редактора

Большинство векторных программ позволяют только разместить растровый рисунок в векторной иллюстрации, изменить его размер, выполнить перемещение и поворот, обрезку, однако изменить в нём отдельные пиксели невозможно. Дело в том, что векторные изображения состоят из отдельных объектов, с которыми можно работать порознь. С растровыми же изображениями так поступать нельзя, так как пиксели нельзя классифицировать подобным образом (объектом здесь является весь растровый фрагмент в целом). Пиксель же обладает одним свойством — цветом. Поэтому в некоторых векторных редакторах к растровым объектам допускается применять специальные эффекты размытия и резкости, в основе которых лежит изменение цветов соседних пикселей.

Достоинства векторной графики

1. Векторные рисунки, состоящие из тысяч примитивов, занимают память, объём которой не превышает нескольких сотен килобайт. Аналогичный растровый рисунок требует памяти в 10-1000 раз больше. Таким образом, векторные изображения занимают относительно небольшой объём памяти.
2. Векторные объекты задаются с помощью описаний. Поэтому, чтобы изменить размер векторного рисунка, нужно исправить его описание. Например, для увеличения или уменьшения эллипса достаточно изменить координаты левого верхнего и правого нижнего угла прямоугольника, ограничивающего этот эллипс. И снова для рисования объекта будет использоваться максимально возможное число элементов (видеопикселей или точек). Следовательно, векторные изображения могут быть легко масштабированы без потери качества.

Замечание. В ряде случаев возможно преобразование растровых изображений в векторные. Этот процесс называется *трассировкой*. Программа трассировки растровых изображений отыскивает группы пикселей с одинаковым цветом, а затем создаёт соответствующие им векторные объекты. Однако получаемые результаты чаще всего нуждаются в дополнительной обработке.

Недостатки векторной графики

1. Прямые линии, окружности, эллипсы и дуги являются основными компонентами векторных рисунков. Поэтому до недавнего времени векторная графика использовалась для построения чертежей, диаграмм, графиков, а также для создания технических иллюстраций. С развитием компьютерных технологий ситуация несколько изменилась: сегодняшние векторные изображения по качеству приближаются к реалистическим. Однако векторная графика не позволяет получать изображений фотографического качества. Дело в том, что фотография — мозаика с очень сложным распределением цветов и яркостей пикселей и представление такой мозаики в виде совокупности векторных примитивов — достаточно сложная задача.
2. Векторные изображения описываются десятками, а иногда и тысячами команд. В процессе печати эти команды передаются устройству вывода (например, лазерному принтеру). При этом может случиться так, что на бумаге изображение будет выглядеть совсем иначе, чем хотелось пользователю, или вообще не распечатается. Дело в том, что принтеры содержат свои собственные процессоры, которые интерпретируют переданные им команды. Поэтому сначала нужно проверить, понимает ли принтер векторные команды данного стандарта, напечатав какой-нибудь простой векторный рисунок. После успешного завершения его печати можно уже печатать сложное изображение. Если же

принтер не может распознать какой-либо примитив, то следует заменить его другим — похожим, но понятным принтеру. Таким образом, векторные изображения иногда не печатаются или выглядят на бумаге не так, как хотелось бы.

Сравнение растровой и векторной графики

Критерий сравнения	Растровая графика	Векторная графика
Способ представления изображения	Растровое изображение строится из множества пикселей	Векторное изображение описывается в виде последовательности логических команд
Представление объектов реального мира	Растровые рисунки эффективно используются для представления реальных образов	Векторная графика не позволяет получать изображения фотографического качества
Качество редактирования изображения	При масштабировании и вращении растровых картинок возникают искажения	Векторные изображения могут быть легко преобразованы без потери качества
Особенности печати изображения	Растровые рисунки могут быть легко распечатаны на принтерах	Векторные рисунки иногда не печатаются или выглядят на бумаге не так, как хотелось бы

Особенности растровых и векторных программ

Графические программы — это инструменты компьютерного художника, с помощью которых он создаёт и редактирует изображения. В настоящее время существует много различных графических программ. Поэтому важно знать, какая программа наилучшим образом подходит для решения конкретной задачи. Улучшение качества изображений, а также монтаж фотографий выполняются в растровых программах. Для создания иллюстраций обычно используются векторные программы, которые также называют программами рисования.

Любая графическая программа содержит набор инструментов для работы с изображениями. Инструмент «Кривая» («Кисть» или «Карандаш») предназначен для рисования прямых и кривых линий. Инструменты «Прямоугольник», «Эллипс», «Многоугольник» используются для построения геометрических фигур. Закраска выполняется инструментом «Заливка». Для создания надписей и заголовков используется инструмент «Текст». При работе с изображением часто возникает необходимость увеличить его фрагмент, чтобы лучше рассмотреть мелкие детали. В этом случае нужно воспользоваться инструментом «Масштаб». Несмотря на то, что растровые и векторные программы могут использовать одинаковые инструменты, способ представления создаваемых ими изображений различен.

В графических программах реализованы возможности, позволяющие перемещать, копировать, удалять, масштабировать, зеркально отражать, вращать отдельные части изображений. Прежде, чем выполнить операцию над фрагментом изображения, его необходимо выделить. В векторных программах выделяют **объекты** (векторные

примитивы), а в растровых — **области** (наборы пикселей). Чтобы выделить объект, достаточно щёлкнуть по нему мышью. Выделение же области — более сложная задача, так как в этом случае необходимо точно указать, какая группа пикселей составляет область (например, цветок или яблоко). Вот почему в растровых программах встречаются разнообразные инструменты выделения. Некоторые из них используются для выделения областей простой формы (прямоугольников или эллипсов), другие — для областей со сложной криволинейной границей.

Так как основное понятие растровой графики — пиксель, большинство инструментов и команд растровых программ изменяют яркость и цветовые оттенки отдельных пикселей. Это даёт возможность улучшать резкость изображений, осветлять или затемнять отдельные его фрагменты, а также удалять небольшие дефекты (морщинки, царапины и т. д.).

Основное понятие векторной графики — **объект**. Поэтому векторные программы содержат команды упорядочивания, взаимного выравнивания, пересечения объектов, исключения одних объектов из других. Таким образом, можно создавать новые объекты сложной формы из более простых.

Как правило, в растровых и векторных программах имеются средства для получения эффекта объёма. Однако трёхмерные образы реальных объектов (персонажи, интерьеры и т. д.) следует создавать в программах трёхмерного моделирования.

Системы цветов в компьютерной графике

В этом разделе:

- излучаемый и отражённый свет в компьютерной графике;
- формирование цветовых оттенков на экране монитора;
- формирование цветовых оттенков при печати изображений.

Для описания цветовых оттенков, которые могут быть воспроизведены на экране компьютера и на принтере, разработаны специальные средства — *цветовые модели* (или *системы цветов*). Чтобы успешно применять их в компьютерной графике, необходимо:

- понимать особенности каждой цветовой модели
- уметь определять тот или иной цвет, используя различные цветовые модели
- понимать, как различные графические программы решают вопрос кодирования цвета
- понимать, почему цветовые оттенки, отображаемые на мониторе, достаточно сложно точно воспроизвести при печати.

Мы видим предметы потому, что они излучают или отражают свет.

Свет — электромагнитное излучение.

Цвет характеризует действие излучения на глаз человека. Таким образом, лучи света, попадая на сетчатку глаза, производят ощущение цвета.

Излучаемый свет — это свет, выходящий из источника, например, Солнца, лампочки или экрана монитора.

Отражённый свет — это свет, «отскочивший» от поверхности объекта. Именно его мы видим, когда смотрим на какой-либо предмет, не являющийся источником света.

Излучаемый свет, идущий непосредственно от источника к глазу, сохраняет в себе все цвета, из которых он создан. Но этот свет может измениться при отражении от объекта (рис. 1).

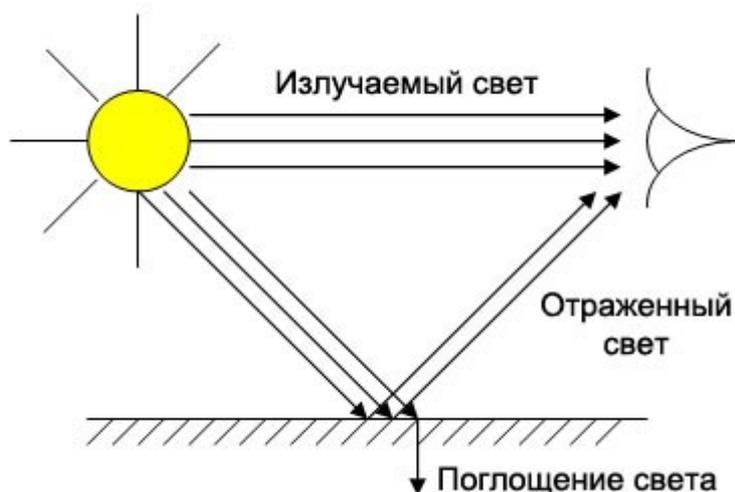


Рис. 1. Излучение, отражение и поглощение света

Подобно Солнцу и другим источникам освещения, монитор излучает свет. Бумага, на которой печатается изображение, отражает свет. Так как цвет может получиться в процессе излучения и в процессе отражения, то существуют два противоположных метода его описания: *системы аддитивных и субтрактивных цветов*.

Система аддитивных цветов

Если с близкого расстояния (а ещё лучше с помощью лупы) посмотреть на экран работающего монитора или телевизора, то нетрудно увидеть множество мельчайших точек красного (**Red**), зелёного (**Green**) и синего (**Blue**) цветов. Дело в том, что на поверхности экрана расположены тысячи фосфоресцирующих цветных точек, которые бомбардируются электронами с большой скоростью. Цветовые точки излучают свет под воздействием электронного луча. Так как размеры этих точек очень малы (около 0,3 мм в диаметре), соседние разноцветные точки сливаются, формируя все другие цвета и оттенки, например:

красный + зеленый = желтый,

красный + синий = пурпурный,

зеленый + синий = голубой,

красный + зеленый + синий = белый.

Компьютер может точно управлять количеством света, излучаемого через каждую точку экрана. Поэтому, изменяя интенсивность свечения цветных точек, можно создать большое многообразие оттенков.

Таким образом, аддитивный (add — присоединять) цвет получается при объединении (суммировании) лучей трех основных цветов — красного, зеленого и синего. Если интенсивность каждого из них достигает 100%, то получается белый цвет. Отсутствие всех трех цветов дает черный цвет. Систему аддитивных цветов, используемую в компьютерных мониторах, принято обозначать аббревиатурой **RGB**.

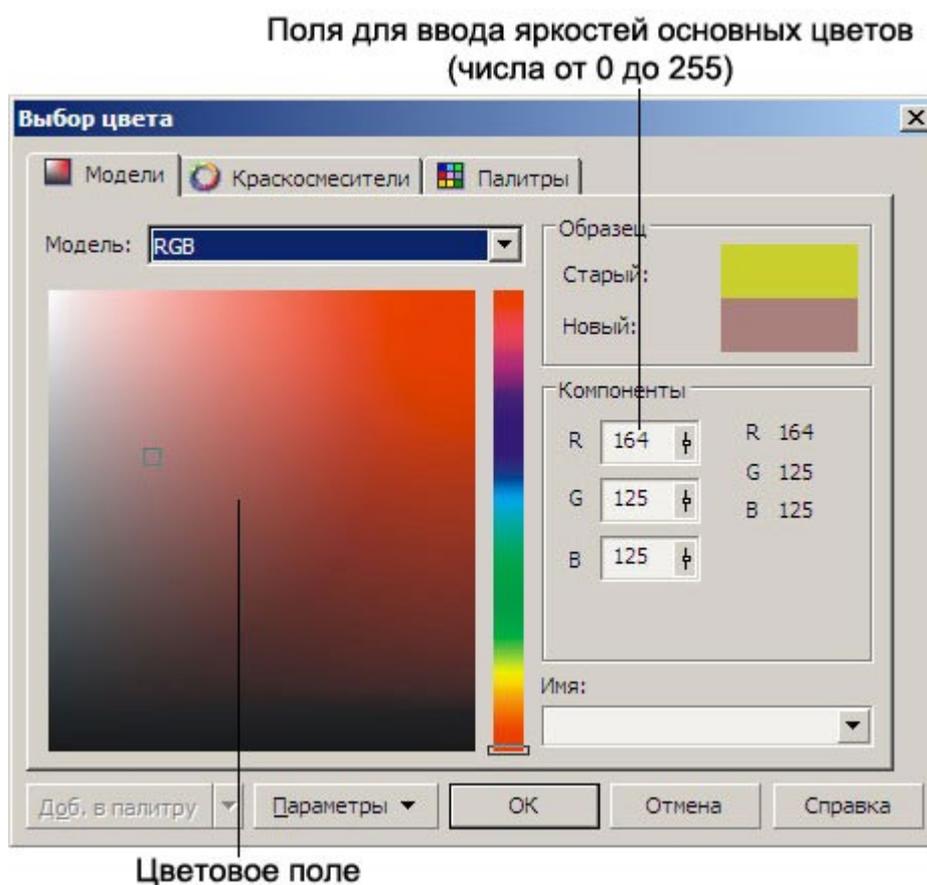


Рис. 2. Диалоговое окно программы CorelDraw для формирования цвета в системе RGB

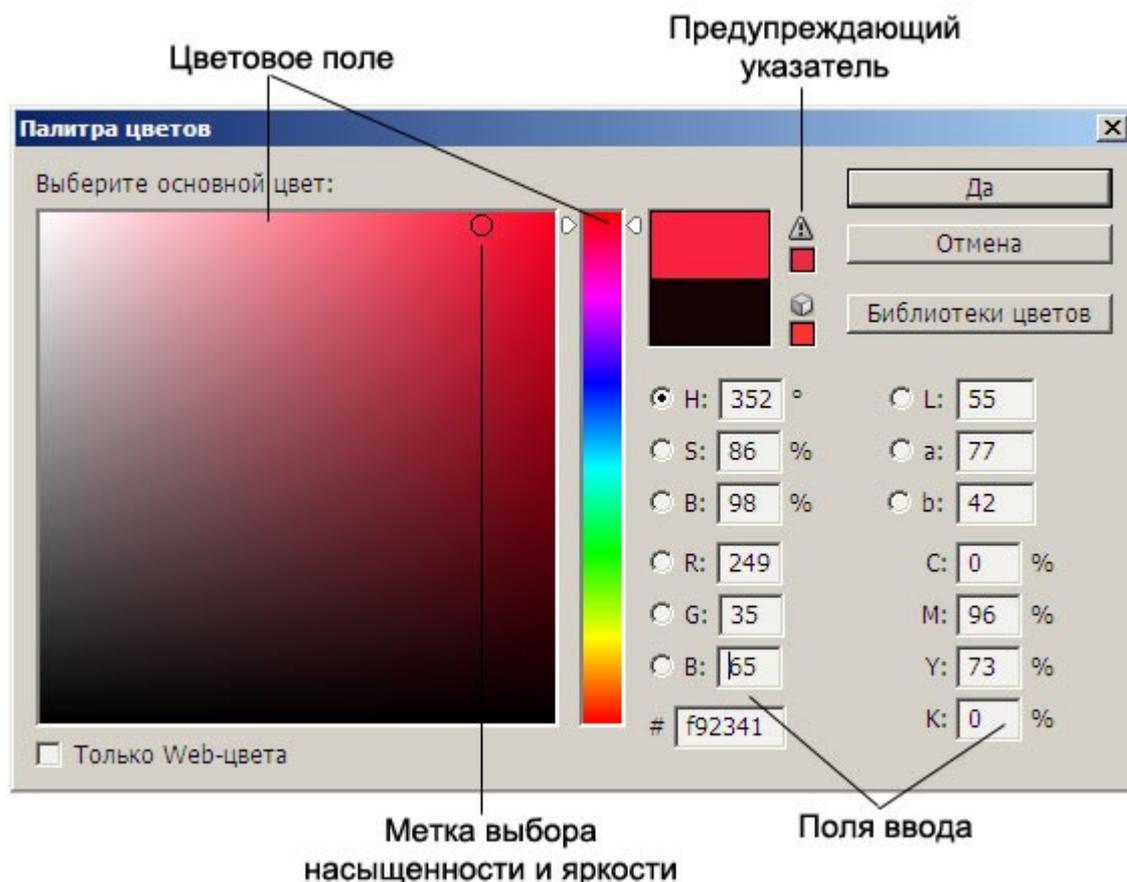


Рис. 3. Диалоговое окно для выбора цвета в программе **Adobe Photoshop**

В большинстве программ для создания и редактирования изображений пользователь имеет возможность сформировать свой собственный цвет (в дополнение к предлагаемым палитрам), используя красную, зеленую и синюю компоненты. Как правило, графические программы позволяют комбинировать требуемый цвет из 256 оттенков красного, 256 оттенков зеленого и 256 оттенков синего. Как нетрудно подсчитать, $256 \times 256 \times 256 = 16,7$ миллионов цветов. Вид диалогового окна для задания произвольного цветового оттенка в разных программах может быть различным (рис. 2,3,4).

Таким образом, пользователь может выбрать готовый цвет из встроенной палитры или создать свой собственный оттенок, указав в полях ввода значения яркостей R, G и B для красной, зеленой и синей цветовых составляющих в диапазоне от 0 до 255 (рис. 2,3,4).

Далее вновь созданный цвет может быть использован для рисования и закрашивания фрагментов изображения.

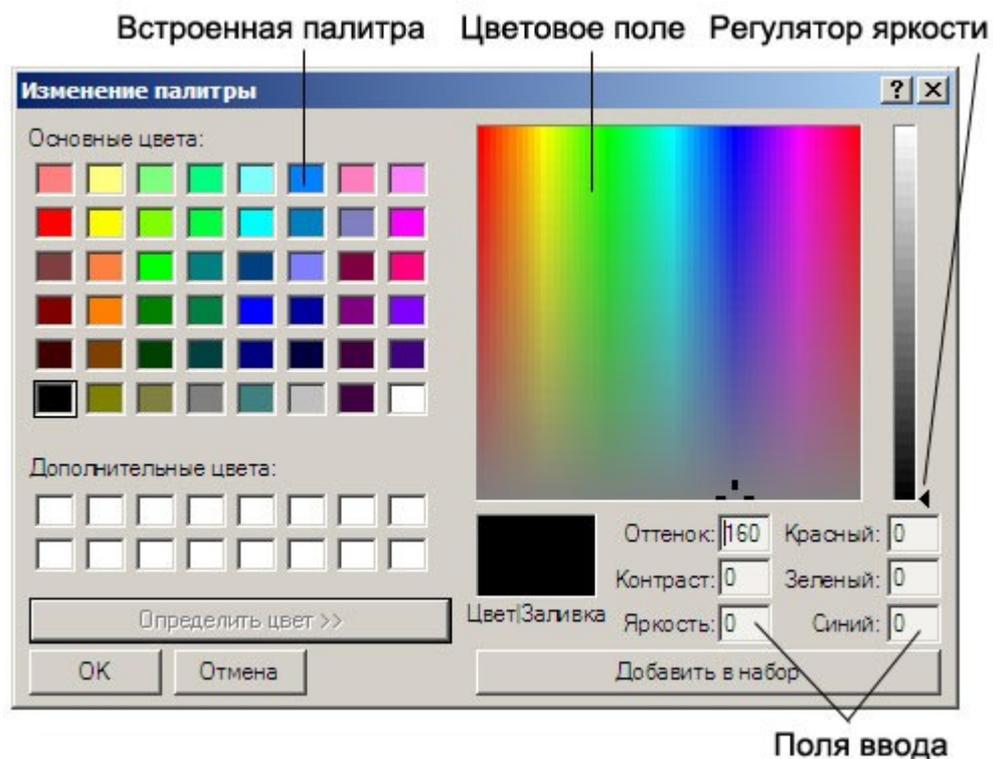


Рис. 4. Диалоговое окно для выбора цвета в графическом редакторе **Paint** (стандартная программа WINDOWS)

В программе CorelDRAW цветовая модель RGB дополнительно представляется в виде трёхмерной системы координат (рис. 2), в которой нулевая точка соответствует чёрному цвету. Оси координат соответствуют основным цветам, а каждая из трёх координат в диапазоне от 0 до 255 отражает «вклад» того или иного основного цвета в результирующий оттенок. Перемещение указателей («ползунков») по осям системы координат влияет на изменение значений в полях ввода, и наоборот. На диагонали, соединяющей начало координат и точку, в которой все составляющие имеют максимальный уровень яркости, располагаются оттенки серого цвета — от чёрного до белого (оттенки серого цвета получаются при равных значениях уровней яркости всех трёх составляющих).

Так как бумага не излучает свет, цветовая модель **RGB** не может быть использована для создания изображения на печатаемой странице.

Система субтрактивных цветов

В процессе печати свет отражается от листа бумаги. Поэтому для печати графических изображений используется система цветов, работающая с отраженным светом — система субтрактивных цветов (subtract — вычитать).

Белый цвет состоит из всех цветов радуги. Если пропустить луч света через простую призму, он разложится в цветной спектр. Красный, оранжевый, жёлтый, зелёный, голубой, синий и фиолетовый цвета образуют видимый спектр света. Белая бумага при освещении отражает все цвета, окрашенная же бумага поглощает часть цветов, а остальные — отражает. Например, листок красной бумаги, освещённый белым светом, выглядит красным именно потому, что такая бумага поглощает все цвета, кроме красного. Та же

красная бумага, освещённая синим цветом, будет выглядеть чёрной, так как синий цвет она поглощает.

В системе субтрактивных цветов основными являются голубой (**Cyan**), пурпурный (**Magenta**) и жёлтый (**Yellow**). Каждый из них поглощает (вычитает) определённые цвета из белого света, падающего на печатаемую страницу. Вот как три основных цвета могут быть использованы для получения чёрного, красного, зелёного и синего цветов:

голубой + пурпурный + жёлтый = чёрный,

голубой + пурпурный = синий,

жёлтый + пурпурный = красный,

жёлтый + голубой = зелёный.

Смешивая основные цвета в разных пропорциях на белой бумаге, можно создать большое многообразие оттенков.

Белый цвет получается при отсутствии всех трёх основных цветов. Высокое процентное содержание голубого, пурпурного и жёлтого образует чёрный цвет. Точнее, чёрный цвет должен получиться теоретически, в действительности же из-за некоторых особенностей типографских красок смесь всех трёх основных цветов даёт грязно-коричневый тон, поэтому при печати изображения добавляется ещё чёрная краска (**Black**).

Систему субтрактивных цветов обозначают аббревиатурой **СМУК** (чтобы не возникла путаница с **Blue**, для обозначения **Black** используется символ **K**).

Процесс четырёхцветной печати можно разделить на два этапа.

1. Создание на базе исходного рисунка четырёх составляющих изображений голубого, пурпурного, жёлтого и чёрного цветов.
2. Печать каждого из этих изображений одного за другим на одном и том же листе бумаги.

Разделение цветного рисунка на четыре компоненты выполняет специальная программа цветоделения. Если бы принтеры использовали систему **СМУ** (без добавления чёрной краски), преобразование изображения из системы **RGB** в систему **СМУ** было бы очень простым: значения цветов в системе **СМУ** — это просто инвертированные значения системы **RGB**. На схеме «цветовой круг» (рис. 5) показана взаимосвязь основных цветов моделей **RGB** и **СМУ**. Смесь красного и зелёного даёт жёлтый, жёлтого и голубого — зелёный, красного и синего — пурпурный и т. д.



Рис. 5. Цветовой круг показывает взаимосвязь моделей **RGB** и **CMY**

Таким образом, цвет каждого треугольника на рис. 5 определяется как сумма цветов смежных к нему треугольников. Но из-за необходимости добавлять чёрную краску, процесс преобразования становится значительно сложнее. Если цвет точки определялся смесью цветов **RGB**, то в новой системе он может определяться смесью значений **CMY** плюс ещё включать некоторое количество чёрного цвета. Для преобразования данных системы **RGB** в систему **CMYK** программа цветоделения применяет ряд математических операций. Если пиксель в системе **RGB** имел чистый красный цвет (100% R, 0% G, 0% B), то в системе **CMYK** он должен иметь равные значения пурпурного и жёлтого (0 % C, 100% M, 100% Y, 0% K).

В приведённой здесь таблице для примера представлено описание нескольких цветов с использованием моделей **RGB** и **CMYK** (диапазон изменения составляющих цвета — от 0 до 255).

Таблица 1

Цвет	RGB	CMYK
золотой	202 R, 153 G, 51B	38C, 105M, 204 Y, 3K
тёмно-фиолетовый	102 R, 51 G, 104B	97C, 183M, 67 Y, 67K
оранжевый	255 R, 102 G, 0B	0C, 187M, 253 Y, 0K
коричневый	153 R, 102 G, 51B	58C, 134M, 174 Y, 29K

Важно то, что вместо сплошных цветных областей программа цветоделения создаёт растры из отдельных точек (рис. 6), причём эти точечные растры слегка повернуты друг относительно друга так, чтобы точки разных цветов не накладывались одна поверх другой, а располагались рядом.

Маленькие точки различных цветов, близко расположенные друг к другу, кажутся сливающимися вместе. Именно так наши глаза воспринимают результирующий цвет.

Таким образом, система **RGB** работает с излучаемым светом, а **CMYK** — с отражённым. Если необходимо распечатать на принтере изображение, полученное на мониторе, специальная программа выполняет преобразование одной системы цветов в другую. Но в системах **RGB** и **CMYK** различна природа получения цветов. Поэтому цвет, который мы видим на мониторе, достаточно трудно точно повторить при печати. Обычно на экране цвет выглядит несколько ярче по сравнению с тем же самым цветом, выведенным на печать.

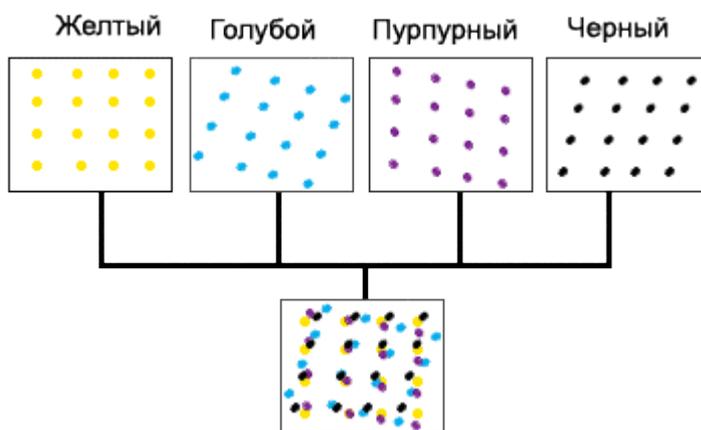


Рис. 6. Точечные растры для четырёхцветной печати

Всё множество цветов, которые могут быть созданы в цветовой модели, называется цветовым диапазоном. Диапазон **RGB** шире диапазона **CMYK**. Это означает, что цвета, созданные на экране, не всегда можно воспроизвести при печати. Поэтому в некоторых графических программах предусмотрены диапазонные предупреждающие указатели. Они появляются в том случае, если цвет, созданный в модели **RGB**, выходит за рамки диапазона **CMYK**. В **Adobe Photoshop** в виде предупреждающего указателя используется маленький восклицательный знак (рис. 3). Когда появляется подобное предупреждение, можно просто щёлкнуть на нём левой кнопкой мыши, тем самым заставив **Adobe Photoshop** заменить данный цвет на ближайший по спектру цвет из модели **CMYK**.

Существуют программы (например, **CorelDraw** и **Adobe Photoshop**), которые позволяют создавать на экране рисунки не только в системе **RGB**, но и в цветах **CMYK**. Для создания произвольного цвета в системе **CMYK** необходимо указать процентное содержание каждого основного цвета аналогично тому, как это делается при работе с **RGB**-моделью. Тогда, глядя на экран, пользователь сможет увидеть, как рисунок будет выглядеть при печати.

Система «Тон — Насыщенность — Яркость»

Системы цветов **RGB** и **CMYK** базируются на ограничениях, накладываемых аппаратным обеспечением (мониторами компьютеров и типографскими красками). Более интуитивным способом описания цвета является его представление в виде тона (**Hue**), насыщенности (**Saturation**) и яркости (**Brightness**). Для такой системы цветов используется аббревиатура **HSB**. *Тон* — конкретный оттенок цвета: красный, желтый, зеленый, пурпурный и т. п. *Насыщенность* характеризует «чистоту» цвета: уменьшая насыщенность, мы «разбавляем» его белым цветом. *Яркость* же зависит от количества черной краски, добавленной к данному цвету: чем меньше черноты, тем больше яркость цвета. Для отображения на мониторе компьютера система **HSB** преобразуется в **RGB**, а для печати на принтере — в систему **CMYK**. Можно создать произвольный цвет, указав в

полях ввода H, S и B значения для тона, насыщенности и яркости из диапазона от 0 до 255 (рис. 3,4,7).

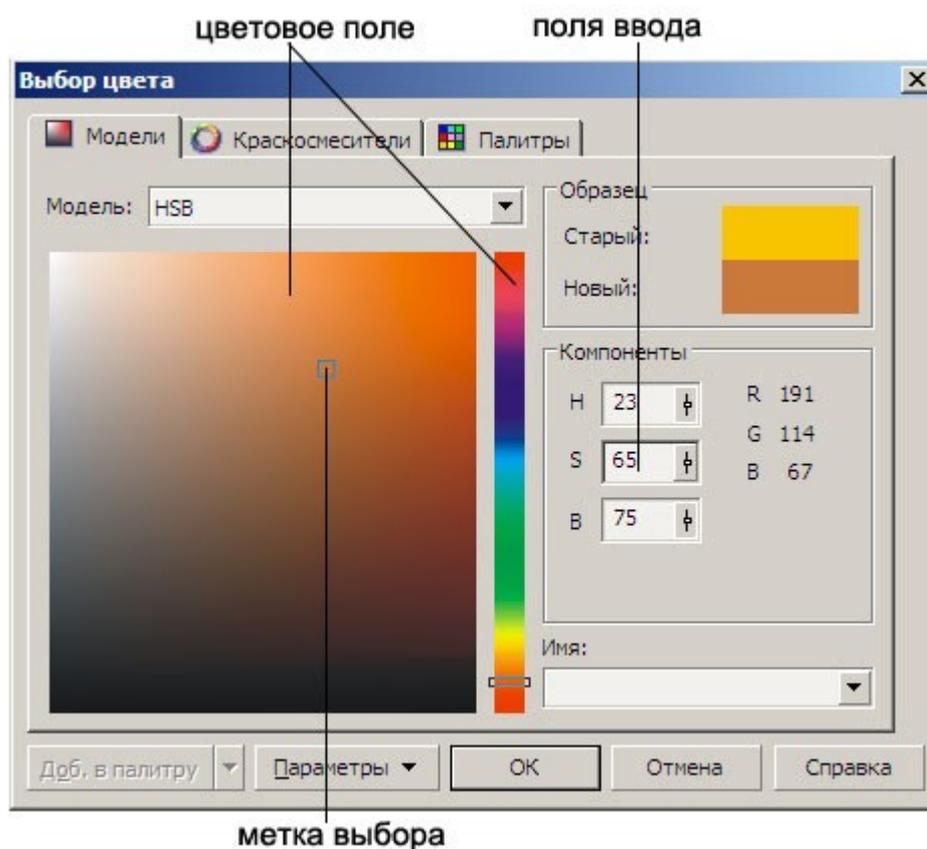


Рис. 7. Диалоговое окно программы **CorelDraw** для формирования цвета в системе **HSB**

Кроме того, пользователь может выбрать цветовой тон, щёлкнув мышью в соответствующей точке цветового поля (рис. 3, 4, 7).

В результате перемещения метки в виде маленького квадратика (рис. 7) по внутренней части цветового прямоугольника изменяются насыщенность и яркость выбранного тона. В левом верхнем углу прямоугольника цвет становится максимально размытым (почти белым). В правом нижнем углу яркость его цвета минимальна. В **Adobe PhotoShop** (рис. 3) изменение насыщенности и яркости выполняется в результате перемещения метки в виде окружности внутри цветового поля. В графическом редакторе **Paint** цветовое поле позволяет выбирать тон и насыщенность (контраст), а яркость устанавливается с помощью специального регулятора (рис. 4).

Контрольные вопросы

1. В чём состоит отличие излучаемого и отражённого света?
2. Какие методы описания цвета вам известны?
3. Как формируется цвет в системе цветов RGB?

4. Как сформировать свой собственный цвет при работе в каком-нибудь графическом пакете?
5. Почему система цветов RGB не может быть использована для создания изображений на печатаемой странице?
6. Какие базовые цвета используются для формирования цвета в системе цветов CMYK?
7. В чём заключается процесс четырёхцветной печати?
8. Почему цвета, созданные на экране, не всегда можно воспроизвести при печати?
9. Как описывается цвет в системе цветов HSB?
1. В чём состоит принцип растровой графики?
2. Что обозначают понятия пиксель, видеопиксель, точка?
3. Почему растровая графика эффективно представляет изображения фотографического качества?
4. Почему для хранения растровых изображений требуется большой объём памяти?
5. Почему растровое изображение искажается при масштабировании?
6. Как хранится описание векторных изображений?
7. Кто составляет последовательность векторных команд?
8. Почему векторные изображения могут быть легко масштабированы без потери качества?
9. Почему векторная графика не позволяет получать изображений фотографического качества?
10. Для решения каких задач используются растровые программы?
11. Почему векторные программы называют программами рисования?
12. Почему в растровых и векторных программах выделение фрагментов изображения выполняется по-разному?
13. Какие программы предоставляют возможность улучшать резкость изображения, осветлять или затемнять отдельные его фрагменты?