



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ им. М. В. ЛОМОНОСОВА

А. В. Боресков, Е. В. Шикин

КОМПЬЮТЕРНАЯ ГРАФИКА

**УЧЕБНИК И ПРАКТИКУМ
ДЛЯ ПРИКЛАДНОГО БАКАЛАВРИАТА**

*Рекомендовано Учебно-методическим отделом
высшего образования в качестве учебника
для студентов высших учебных заведений, обучающихся
по инженерно-техническим направлениям и специальностям*

**Книга доступна в электронной библиотечной системе
biblio-online.ru**

Москва ■ Юрайт ■ 2016

УДК 004.5+741(075.8)
ББК 32.973.26-018.2я73
Б82

Авторы:

Боресков Алексей Викторович — кандидат физико-математических наук, младший научный сотрудник кафедры нелинейных динамических систем и процессов управления факультета вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова;

Шикин Евгений Викторович — доктор физико-математических наук, профессор, заведующий кафедрой математических методов и информационных технологий в управлении факультета государственного управления, профессор кафедры общей математики факультета вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова.

Рецензенты:

Березин С. Б. — кандидат физико-математических наук, доцент кафедры математической физики факультета вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова;

Милонов Г. А. — старший преподаватель кафедры программного обеспечения систем РЭА Института кибернетики Московского государственного университета информационных технологий, радиотехники и электроники.

Боресков, А. В.

Б82 Компьютерная графика : учебник и практикум для прикладного бакалавриата / А. В. Боресков, Е. В. Шикин. — М. : Издательство Юрайт, 2016. — 219 с. — Серия : Бакалавр. Прикладной курс.

ISBN 978-5-9916-5468-5

В данном учебнике рассмотрены основные аспекты компьютерной графики, как чисто математические, так и алгоритмические, дано описание используемых понятий и основных алгоритмов. Отдельные главы посвящены использованию библиотек OpenGL и GLUT и современных пакетов компьютерной графики.

Соответствует актуальным требованиям Федерального государственного образовательного стандарта высшего образования.

Для студентов высших учебных заведений, обучающихся по инженерно-техническим направлениям и специальностям.

УДК 004.5+741(075.8)
ББК 32.973.26-018.2я73

ISBN 978-5-9916-5468-5

© Боресков А. В., Шикин Е. В., 2015
© ООО «Издательство Юрайт», 2016

Оглавление

Предисловие	5
Введение.....	7
Глава 1. Общие сведения о компьютерной графике.	
Область применения компьютерной графики	9
<i>Контрольные вопросы</i>	<i>11</i>
Глава 2. Растровая и векторная графика	12
<i>Контрольные вопросы</i>	<i>15</i>
Глава 3. Цвет и его представления в компьютере.....	16
<i>Контрольные вопросы и упражнения</i>	<i>26</i>
Глава 4. Математические основы работы в 2D.	
Преобразования на плоскости	27
4.1. Системы координат	27
4.2. Векторы.....	29
4.3. Матрицы	32
4.4. Преобразования на плоскости	34
4.5. Прямые на плоскости.....	40
4.6. Работа с векторами с помощью языка C++	41
<i>Контрольные вопросы и упражнения</i>	<i>45</i>
Глава 5. Математические основы трехмерной графики.	
Преобразования в пространстве. Проектирование.	
Однородные координаты	47
5.1. Векторы и матрицы.....	47
5.2. Плоскости и прямые.....	50
5.3. Преобразования	52
5.4. Линейное пространство	56
5.5. Кватернионы.....	57
5.6. Однородные координаты.....	61
5.7. Преобразование нормали	63
5.8. Проектирование.....	64
5.9. Переход между системами координат.....	67
5.10. Использование классов языка C++	69
<i>Контрольные упражнения</i>	<i>69</i>
Глава 6. Геометрическое моделирование.	
Операции над моделями. Слайны.....	71
6.1. Кривые в пространстве.....	72

6.2. Поверхности в пространстве	72
<i>Контрольные вопросы и упражнения</i>	<i>84</i>
Глава 7. Алгоритмические основы компьютерной графики.	
Растровые алгоритмы, определение видимости, закрашивание.....	85
7.1. Растровые алгоритмы	85
7.2. Определение видимости	92
7.3. Трассировка лучей.....	96
7.4. Метод буфера глубины.....	97
7.5. Методы упорядочивания	99
7.6. Расчет освещенности.....	106
7.7. Закрашивание.....	111
<i>Контрольные вопросы и упражнения</i>	<i>112</i>
Глава 8. Цифровая обработка изображений, фильтры.....	114
<i>Контрольные вопросы и упражнения</i>	<i>121</i>
Глава 9. Технические средства компьютерной графики	123
<i>Контрольные вопросы</i>	<i>128</i>
Глава 10. Библиотеки OpenGL, GLU и GLUT	129
10.1. Минимальное приложение на OpenGL.....	134
10.2. Вывод примитивов	138
10.3. Режимы.....	141
10.4. Работа с матрицами	142
10.5. Освещение	151
10.6. Работа с буфером трафарета	155
10.7. Работа с текстурами	158
10.8. Загрузка текстур с помощью библиотеки SOIL.....	172
10.9. Загрузка текстур с помощью библиотеки DevIL.....	174
10.10. Создание анимации с помощью библиотеки GLUT	175
10.11. Работа с OpenGL через библиотеку Qt.....	177
<i>Контрольные вопросы и упражнения</i>	<i>178</i>
Глава 11. Компьютерная анимация и мультимедиа	179
11.1. Основы анимации	179
11.2. Мультимедиа.....	185
<i>Контрольные вопросы и упражнения</i>	<i>187</i>
Глава 12. Основные графические пакеты.....	188
<i>Контрольные вопросы</i>	<i>195</i>
Глава 13. Основа работы с шейдерами в OpenGL	196
<i>Контрольные вопросы и упражнения</i>	<i>218</i>
Рекомендуемая литература.....	219

Предисловие

Данная книга представляет собой учебник по современной компьютерной графике, ориентированный на бакалавров. В ней рассматриваются основные аспекты компьютерной графики, как чисто математические, так и алгоритмические, дается описание используемых понятий и основных алгоритмов.

В книге содержатся многочисленные контрольные вопросы и упражнения, которые помогут читателю проверить свои знания, а также практические упражнения, требующие написания программ.

Предполагается, что читатель уже знаком с основами линейной алгебры, аналитической геометрии и математического анализа, поскольку в ходе изложения материала регулярно используются понятия из этих областей математики. Кроме того, приведены многочисленные примеры программ на языке C++, поэтому знание основ программирования и, в частности, этого языка крайне желательно (однако по возможности избегается использование сложных понятий и возможностей C++).

Рассматривается широкий спектр различных понятий компьютерной графики, включая математические и алгоритмические основы, а также использование библиотек OpenGL и GLUT. Книга посвящена не только описанию различных алгоритмов, в отдельной главе дано краткое описание современных пакетов компьютерной графики. Достаточно подробно описать основные пакеты в рамках данного учебника не представляется возможным, поскольку они весьма сложны и полное описание любого из них заняло бы книгу, гораздо большую, чем та, которую вы держите в руках. Поэтому материал главы носит скорее ознакомительный характер, и для освоения того или иного пакета рекомендуется обратиться к специализированным книгам и интернет-ресурсам.

Одна из глав посвящена основам компьютерной анимации и мультимедиа, она дает базовые сведения об этих быстро развивающихся и довольно обширных направлениях.

В результате изучения данной книги студент должен:

знать

- предмет, цели и задачи компьютерной графики;
- основные понятия компьютерной графики;
- основные алгоритмы компьютерной графики;

уметь

- решать задачи, связанные с преобразованием координат и проектированием;
- решать задачи, связанные с цветами и переходами между различными цветовыми пространствами;
- выбирать подходящие методы для решения основных задач компьютерной графики;
- применять полученные теоретические знания к конкретным задачам компьютерной графики;
- пользоваться библиотеками OpenGL, GLU, GLUT и писать простые программы для рендеринга объектов с применением текстурирования и освещения;

владеть

- методами и алгоритмами, используемыми в компьютерной графике;
- современной терминологией, применяемой в компьютерной графике;
- навыками поиска информации, необходимой для решения задач компьютерной графики.

Введение

Практически каждый человек в своей жизни постоянно сталкивается с компьютерной графикой, в современном мире она играет очень важную роль. Именно благодаря компьютерной графике мы можем «вживую» увидеть еще не существующие предметы и устройства, ознакомиться с их работой. Средства визуализации крайне важны для инженеров и архитекторов, огромную роль играет компьютерная графика в рекламе и индустрии развлечений. Именно с помощью компьютерной графики созданы, например, такие мультфильмы, как «Храбрая сердцем» и «Как приручить дракона», потрясающие визуальные эффекты для фильмов «Властелин колец» и «Хоббит».

Без компьютерной графики было бы невозможным создание многих из компьютерных игр, таких как Crysis или Metro 2033. Именно технологии современной компьютерной графики (вместе с многократно выросшими за последние годы ресурсами персональных компьютеров и игровых консолей) приближают нынешние компьютерные игры по качеству изображения к кинофильмам.

Компьютерная графика развивается быстрыми темпами, постоянно появляются новые методы и алгоритмы, позволяющие показывать сложные и захватывающие эффекты, затрачивая для этого все меньше и меньше вычислительных ресурсов.

К числу красивых визуальных эффектов, появившихся в последние годы, можно отнести реалистичный рендеринг волос и меха, учет подповерхностного рассеивания для аккуратного рендеринга лица. Все эти и многие другие эффекты можно легко заметить, посмотрев мультфильмы последних лет таких компаний, как Pixar и DreamWorks.

При этом компьютерная графика позаимствовала много понятий и элементов из самых разных областей знания. Так, понятие цвета тесно связано с функционированием зрительной системы человека. Огромную роль в компьютерной графике играют двух- и трехмерные координаты и преобразования на плоскости и в пространстве.

Для построения изображений, практически не отличающихся от фотографий, используют последние достижения физики отно-

сительно отражения и рассеивания света, позволяющие строить физически корректные и реалистично выглядящие модели освещения.

Для построения реалистичных изображений приходится разрабатывать различные алгоритмы и способы оптимизации, позволяющие получать изображения требуемого качества за сравнительно небольшое время. Но даже с учетом подобных оптимизаций для построения графики высокого качества зачастую требуются очень мощные вычислительные ресурсы, приводящие к использованию кластеров из нескольких тысяч персональных компьютеров для расчетов. Все это требует крайне сложного программного обеспечения.

Сейчас существует большое количество различных графических пакетов, таких как Adobe PhotoShop, Autodesk 3DS Max, Maya и многих других. Эти мощные и весьма дорогостоящие программные пакеты ориентированы на решение графических задач определенных классов (или типов). В то же время существуют и различные opensource («открытое программное обеспечение») решения, которые являются бесплатными, поддерживают различные операционные системы и обладают большим сообществом пользователей. Так, при поддержке Blender Foundation регулярно создаются opensource мультфильмы: пользователь может скачать не только сам мультфильм, но и все использованные в нем модели, анимации, текстуры и т.п. При этом для моделирования и анимации используется пакет Blender, а непосредственно для рендеринга — пакет Yafray. Для многих других графических пакетов также есть их бесплатные (opensource) аналоги, такие как GIMP, Blender, Paint.net.

Глава 1

ОБЩИЕ СВЕДЕНИЯ

О КОМПЬЮТЕРНОЙ ГРАФИКЕ.

ОБЛАСТЬ ПРИМЕНЕНИЯ

КОМПЬЮТЕРНОЙ ГРАФИКИ

Компьютерная графика — это часть информатики (*computer science*), занимающаяся как созданием новых, так и обработкой уже существующих изображений.

Компьютерная графика является сравнительно молодой дисциплиной: первые попытки использовать компьютер не только непосредственно для научных расчетов, но и для отображения информации, относятся к началу 1950-х гг. Далее, с одной стороны, компьютеры становились все быстрее, увеличивался объем оперативной памяти, а с другой — развивались средства отображения информации. Появились графические плоттеры (графопостроители), графические дисплеи, причем разрешение последних все время увеличивалось.

Все это способствовало бурному росту компьютерной графики. Причем, если в самом начале получаемые изображения носили явно искусственный (или «компьютерный») вид, то в дальнейшем качество получаемых изображений сильно повысилось и стало практически невозможно отличить построенное на компьютере изображение от фотографии. В то же время появилась возможность создавать графику в режиме реального времени, что привело к разработке широкого спектра различных интерактивных приложений, включающих в себя компьютерные игры. В результате компьютерная графика распространилась в сфере кино и рекламы. Большинство сложных спецэффектов оказалось гораздо дешевле и проще сделать целиком на компьютере, более того, многие графические эффекты можно сделать только таким способом.

Компьютерная графика играет огромную роль в современном кинематографе, причем не только для реализации различных спецэффектов, но и, например, для нелинейного видеомонтажа.

С появлением достаточно дешевых персональных компьютеров стали активно создаваться различные компьютерные игры, которые также играют важную роль в развитии компьютерной графики. В отличие от кино, где используют мощные компьютеры и кластеры компьютеров, графика для компьютерных игр строилась с помощью довольно слабых в техническом отношении персональных компьютеров. Кроме того, для игр также требовалось построение изображений с заданной частотой кадров, в противном случае игра просто была бы никому не нужна. Результат уступает по качеству тому, что мы видим в кино, но для игр этого бывает достаточно. В последнее время, однако, наблюдается тенденция постоянного повышения и совершенствования качества графики в играх, оно приближается к кинематографическому.

Из вышесказанного можно сделать вывод, что наиболее яркой и заметной сферой применения компьютерной графики является индустрия развлечений — спецэффекты для фильмов, компьютерные мультфильмы и игры. Однако использование компьютерной графики отнюдь не ограничивается этой сферой. Например, она довольно широко применяется для создания привлекающей внимание рекламы — как в виде отдельных изображений, так и видеороликов.

Еще одной областью использования компьютерной графики является архитектура, где очень важно иметь возможность увидеть, как будет выглядеть здание до того, как оно будет построено. Аналогичная потребность часто возникает у инженеров — определить вид будущего изделия и прямо по ходу работы изменить его характеристики (например, материал).

Важную роль компьютерная графика играет и в сфере образования: она дает возможность увидеть сложные (и часто скрытые от глаз) процессы во всех деталях. Нужна она и при создании различных тренажеров, например для обучения пилотов. Сфера применения компьютерных обучающих систем растет с каждым годом, и далеко не последнюю роль в этом играет компьютерная графика.

Интересная область применения компьютерной графики — визуализация результатов медицинских исследований. Здесь обычно подразумевается построение объемных изображений, полученных в результате томографии или других похожих процедур. При этом крайне важна возможность управлять визуализацией, задавая различные параметры, области отсечения и т.п.

В целом, можно сказать, что компьютерная графика проникает во все новые и новые направления, позволяя представить информацию в визуальной форме, которая наиболее удобна для восприятия.

Контрольные вопросы

1. Назовите основные области применения компьютерной графики.
2. Приведите примеры использования компьютерной графики в своей специальности.
3. Приведите примеры использования компьютерной графики в последних фильмах, которые вы смотрели.

Глава 2

РАСТРОВАЯ И ВЕКТОРНАЯ ГРАФИКА

Одним из важнейших понятий в компьютерной графике является то, как мы представляем и храним графическую информацию; в первую очередь это относится к изображениям. На ранних стадиях развития компьютерной графики, когда компьютеры были маломощными и обладали очень небольшим объемом оперативной памяти, наиболее распространенным способом представления изображений был векторный.

При *векторном* способе отображения/хранения все данные представлены как наборы отрезков, дуг и т.п., т.е. линейных примитивов. Такой способ позволяет хранить и отображать чертежи с высокой точностью и требует относительно небольшого объема памяти.

Для отображения векторных данных на электронно-лучевых трубках (CRT, ЭЛТ) луч вместо традиционной развертки просто пробегает все имеющиеся примитивы один за другим. Это давало достаточно точное отображение данных, однако обладало и рядом серьезных ограничений. Одним из них было то, что время, за которое луч в ЭЛТ обходил все изображение, увеличивалось в зависимости от сложности изображения, в результате чего появлялось заметное его мерцание. Кроме того, векторное отображение оказалось плохо приспособленным для передачи сложной информации: например, для того чтобы отобразить сплошной треугольник, приходилось выводить его на экран как ряд параллельных отрезков для создания иллюзии того, что мы видим треугольник (рис. 2.1).

По мере удешевления оперативной памяти и роста быстродействия компьютеров все чаще стали прибегать к *растровому* способу отображения информации. При растровом способе отображения весь экран выводящего устройства представляет собой прямоугольную матрицу из $M \times N$ элементов. Каждый из таких элементов получил название *пиксела* (от англ. *picture element*). Пиксел обладает своим цветом, который может принимать значения из заданного множества цветов (рис. 2.2).

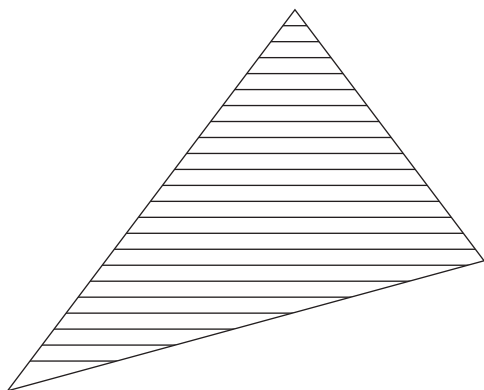


Рис. 2.1. Векторное представление треугольника с помощью набора отрезков

Цвет каждого пиксела задавался фиксированным числом бит, которые хранились в специально отведенном участке памяти, получившем название видеопамати. Число бит на один пиксел определяет, сколько различных цветов одновременно можно отобразить на экране. Соответственно, присутствовал блок, который с заданной частотой сканировал эту память и по ее содержимому генерировал видеосигнал.

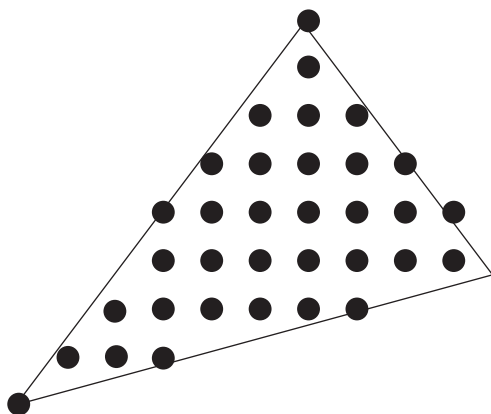


Рис. 2.2. Растровое представление треугольника

Первое время, в связи с небольшим объемом видеопамати, число бит на один пиксел было очень небольшим — от одного до восьми, поэтому часто использовали специальную таблицу цветов — палитру. Биты для каждого пиксела определяли индекс цвета в палитре, но сами цвета можно было выбирать из очень

большого набора. Это позволило обеспечить постоянную скорость отображения информации вне зависимости от сложности изображения.

Однако разрешение дисплея заметно влияет на качество изображения и занимаемое им место в памяти. Для того чтобы на растровом устройстве можно было отображать различные базовые примитивы — отрезки, дуги и т.п., их необходимо было *растеризовать*, т.е. превратить в набор пикселей.

Сейчас на современных графических процессорах (GPU, Graphics Processing Unit) растеризация осуществляется аппаратно: приложение просто посылает на вход GPU набор примитивов, заданных своими вершинами.

Помимо представления изображений на экране также возникает задача их хранения. Изображения можно хранить как в векторной, так и в растровой форме, независимо от того, на какое устройство их будут выводить.

Большинство распространенных форматов хранения изображений (например, .bmp, .png, .gif, .jpeg) являются растровыми, т.е. хранят изображение в виде набора пикселей. Из-за того что изображения бывают довольно большими, многие из этих форматов прибегают к различным видам сжатия информации, позволяющим сохранить изображение, используя при этом заметно меньший объем памяти.

Способы сжатия графической (в том числе и видео-) информации делятся на два принципиально разных класса — *сжатие без потерь* и *сжатие с потерями*. Форматы, использующие сжатие без потерь (такие как .bmp, .png, .gif), хранят изображение таким образом, что после его распаковки мы получаем в точности такое изображение, каким оно было до сжатия. Сжатие с потерями (.jpeg) вносит небольшие искажения в изображение, позволяющие очень сильно его сжать. Степень сжатия при этом во много раз выше, чем при первом способе. Степенью сжатия можно управлять: чем сильнее мы сожмем изображение, тем больше искажений в него внесем. Обычно искажения вносят таким образом, чтобы они были практически незаметны глазу.

Кроме растровых, существуют и векторные форматы хранения изображений, обычно они хранят изображения в виде набора примитивов или команд для вывода этих примитивов. К числу таких форматов относятся, например, .wmf и .svg.

Важным преимуществом векторного формата хранения изображения является то, что мы всегда можем растеризовать изображение для конкретного разрешения (точек на единицу длины, обычно — точек на дюйм, *dpi, dots per inch*). Разрешение может

очень сильно различаться и у хороших принтеров достигать нескольких тысяч точек на дюйм. Поэтому при построении изображения, которое по своему виду скорее векторное (например, графики, диаграммы), лучше его сохранять в векторном формате. Такое сохранение обеспечит получение качественного изображения независимо от разрешения раstra. Это связано с тем, что хотя при выводе векторного изображения все равно будет происходить растрезизация, но она будет привязана к разрешению конкретного устройства, на которое будет происходить вывод.

Таким образом, несмотря на то что практически все современные графические устройства являются растровыми, в целом ряде случаев более выгодно хранить данные в векторном представлении (если исходные данные носят векторный характер).

С точки зрения математики можно рассматривать векторное представление как непрерывное, а растровое — как дискретное, т.е. заданное на конкретной растровой сетке. Из этого следует, что растровому представлению свойственны так называемые ошибки дискретизации, или *алиасинг* (*aliasing*), с которыми приходится бороться, прибегая к специальным алгоритмам.

Контрольные вопросы

1. Какие способы представления изображений вы знаете?
2. Назовите несколько форматов для хранения изображений. Какие из них являются растровыми, а какие — векторными?
3. Что такое сжатие изображений с потерями и без потерь?
4. Приведите примеры форматов для хранения изображения со сжатием с потерями и без потерь.
5. Назовите основные преимущества векторного и растрового способов хранения.
6. Если нужно подготовить на компьютере диаграмму, то какой способ представления изображения лучше использовать для получения хорошего качества при печати?

Глава 3

ЦВЕТ И ЕГО ПРЕДСТАВЛЕНИЯ В КОМПЬЮТЕРЕ

Одним из базовых терминов в компьютерной графике является *цвет*. Далее мы будем постоянно сталкиваться с ним, поэтому необходимо дать ему корректное определение. Несмотря на то что цвет является базовым понятием, интуитивно понятным каждому, строго определить его довольно сложно.

Есть несколько вариантов подхода к цвету, и наиболее очевидным из них является подход, основанный на физике. Как известно, свет — это поток электромагнитных волн. Попробуем дать определение цвета подобным образом.

Определим поток электромагнитных волн, содержащий волны различной длины λ , с помощью *плотности энергии* (SPD, *spectral power distribution*) — функции $I(\lambda)$. Задав длину волны λ и некоторую малую величину $\Delta\lambda$, можно найти общую энергию, которая заключена в волнах с длинами от λ до $\lambda + \Delta\lambda$, с помощью следующего интеграла:

$$\Delta I = \int_{\lambda}^{\lambda + \Delta\lambda} I(\lambda) d\lambda.$$

На рис. 3.1 приведена спектральная функция для дневного света, называемая D_{65} .

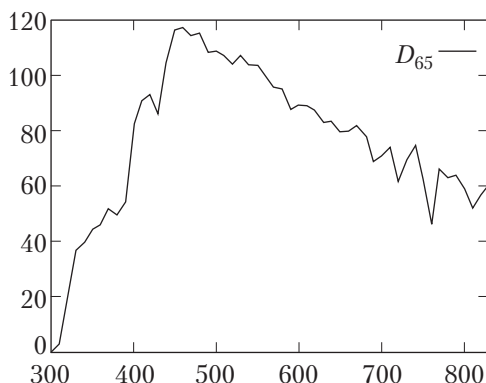


Рис. 3.1. Спектральная функция для дневного света

Кажется, что мы можем сопоставить каждому цвету свою спектральную функцию $I(\lambda)$ и определять далее цвет с ее помощью. Такой подход обладает удобным свойством *аддитивности*: если мы хотим сложить два цвета (например, точка освещается двумя источниками разных цветов), то просто складываем соответствующие спектральные функции. Однако в действительности такой подход для представления цветов не работает. Одно и то же ощущение цвета (как его воспринимает человек) вызывается бесконечным количеством различных спектральных функций. Такие спектральные функции, вызывающие одно и то же ощущение цвета, называются *метамерами*. Подход, основанный на представлении цвета при помощи спектральной кривой, не подходит, поскольку мы даже не можем однозначно сопоставить каждому цвету единственную спектральную кривую.

Из этого следует, что для построения модели цвета мы должны опираться в первую очередь на то, как работает человеческий глаз. Можно представить глаз человека как оптический прибор, в котором есть свой аналог объектива — хрусталик — и светочувствительный элемент — сетчатка (рис. 3.2). С помощью глаза мы можем видеть не все электромагнитные волны, а только те, длина которых находится примерно в диапазоне 400–700 нм.

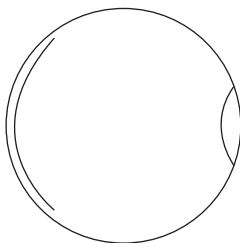


Рис. 3.2. Устройство человеческого глаза

Хрусталик проектирует видимое изображение на сетчатку, покрытую светочувствительными клетками — палочками и колбочками. Палочки, расположенные в основном по периферии сетчатки, обладают крайне высокой светочувствительностью, но при этом слабо различают длину волны. Именно палочки отвечают за так называемое ночное зрение и позволяют видеть при плохом освещении, но при этом мы не различаем цвета.

Если необходимо разглядеть что-то в темноте, то лучше не смотреть прямо на объект, а постараться увидеть его боковым зрением. При этом изображение попадет на периферию сетчатки, где находится большое количество палочек, обладающих высокой светочувствительностью.

Колбочки, расположенные в основном в центре сетчатки, обладают низкой светочувствительностью и узким диапазоном воспринимаемых длин волн. Именно за счет этих клеток мы различаем цвета, однако в силу их низкой светочувствительности мы плохо различаем цвета в темноте. Существуют три типа колбочек, чувствительных соответственно к коротким, средним и длинным волнам. Эти длины примерно соответствуют красному, зеленому и синему цветам. На рис. 3.3 приведены нормализованные чувствительности колбочек к волнам различных длин.

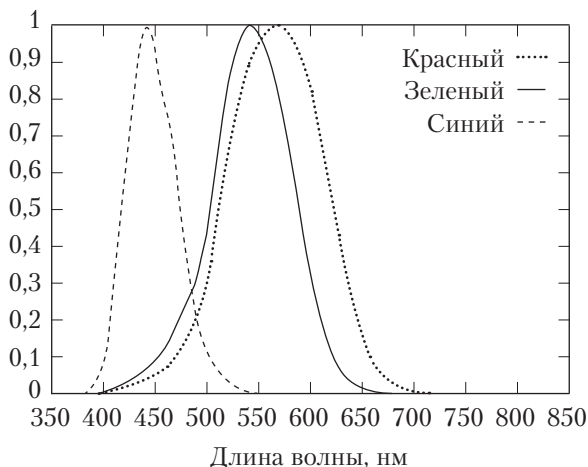


Рис. 3.3. Графики чувствительности колбочек к волнам различных длин

Таким образом, мы приходим к следующему принципу: каждый цвет будем представлять как сумму трех базовых цветов — красного (*Red*, r), зеленого (*Green*, g) и синего (*Blue*, b) — с заданными весами. Именно эта тройка весов (r , g , b) и будет задавать цвет.

Соответствующая модель получила название *RGB*, и множество всех представимых цветов является единичным кубом (рис. 3.4). Таким образом, каждый цвет представляется как трехмерный вектор, компоненты которого определяют доли красного, зеленого и синего цветов, которые нужно смешать, чтобы получить заданный цвет.

В этой цветовой модели черный цвет представлен тройкой чисел $(0, 0, 0)$, белый — тройкой $(1, 1, 1)$, а диагональ куба, соединяющая соответствующие этим цветам вершины куба, соответствует оттенкам серого цвета.

Легко понять, что эта модель представления цвета также является аддитивной. При этом мы не определили сами базовые цвета:

в зависимости от выбора этих базовых цветов будет меняться сам вектор, представляющий какой-либо конкретный цвет, т.е. эта модель должна быть привязанной к конкретным базовым цветам.

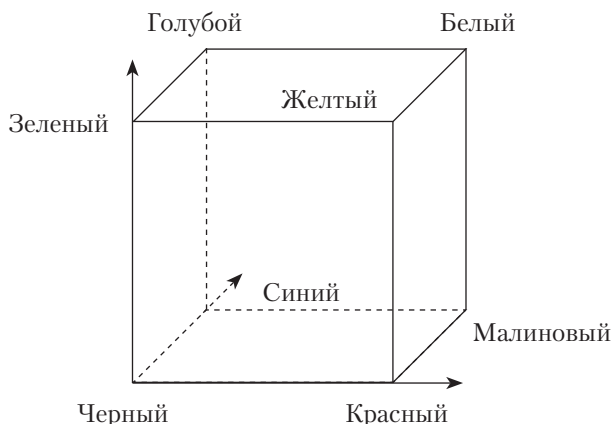


Рис. 3.4. Множество всех цветов для модели *RGB*

Другим, гораздо более серьезным, недостатком модели *RGB* является следующий. В 1920-х гг. У. Д. Райт и Дж. Гилд провели серию опытов с привлечением групп наблюдателей. Задачей этих опытов было получение *RGB*-представлений для большой группы цветов, в первую очередь чистых спектральных (т.е. состоящих из волн с одной и той же длиной волны) цветов.

В проводимых опытах наблюдателям показывали две точки: одна соответствовала разлагаемому цвету, а другая строилась как результат наложения точек от трех источников — красного, зеленого и синего цветов. Задачей наблюдателей было добиться идентичности цветов обеих точек путем изменения пропорций красного, зеленого и синего.

В результате этих опытов выяснилось, что далеко не все цвета можно представить в виде суммы красного, зеленого и синего. Более того, чистые спектральные цвета также нельзя представить таким образом. Однако при этом обнаружилось, что если к разлагаемому цвету прибавить определенный базовый цвет, то получившийся в результате цвет уже можно разложить на сумму базовых. Фактически это соответствует тому, что для представления некоторых цветов иногда получаются отрицательные веса.

Поскольку использование отрицательных весов крайне неудобно, в 1931 г. была создана модель *CIE XYZ* — стандарт представления цвета. В этой модели каждый видимый человеком цвет однозначно представлялся с помощью тройки неотрицательных

чисел (X, Y, Z) , причем Y -компонента соответствовала воспринимаемой человеком яркости.

На рис. 3.5 приведены весовые функции для определения цвета по его спектральной функции.

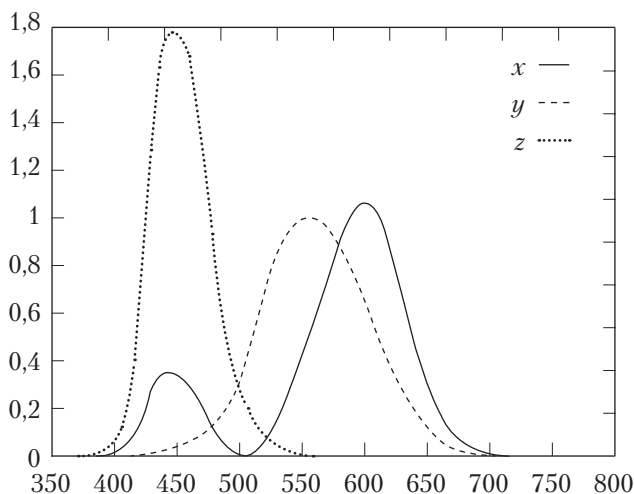


Рис. 3.5. Весовые функции для модели XYZ

Для получения XYZ-координат цвета по спектральной функции используют следующие формулы:

$$X = \int I(\lambda)k_x(\lambda); Y = \int I(\lambda)k_y(\lambda); Z = \int I(\lambda)k_z(\lambda),$$

где $k_x(\lambda)$, $k_y(\lambda)$ и $k_z(\lambda)$ — определенные стандартом весовые функции.

Модель CIE XYZ является стандартом для представления цвета, и все остальные цветовые модели выражаются через нее. В частности, существует стандарт для телевидения высокой четкости (так называемый Rec 709), который явно задает используемые цвета (красный, зеленый и синий) их XYZ-координатами:

$$\begin{pmatrix} r_{709} \\ g_{709} \\ b_{709} \end{pmatrix} = \begin{pmatrix} 3,240479 & -1,537150 & -0,498535 \\ -0,969256 & 1,875992 & 0,041556 \\ 0,055648 & -0,204043 & -0,204043 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}.$$

В моделях CIE XYZ и RGB информация о цвете «размазана» по всем трем компонентам. Соответственно, существуют цветовые модели, в которых имеет место явное деление координат на яркость и так называемые хроматические координаты, задаю-

щие цвет без учета яркости. Одной из таких моделей является Yxy , определяемая по следующим формулам:

$$x = \frac{X}{X+Y+Z}; \quad y = \frac{Y}{X+Y+Z}.$$

Хроматические координаты x и y полностью определяют цвет без учета его яркости. На рис. 3.6 приведена фигура (называемая хроматической диаграммой), которая содержит хроматические координаты всех видимых человеком цветов. На этом изображении дуга соответствует чистым спектральным цветам, а отрезок — цветам, получающимся при смешивании красного и синего цветов.

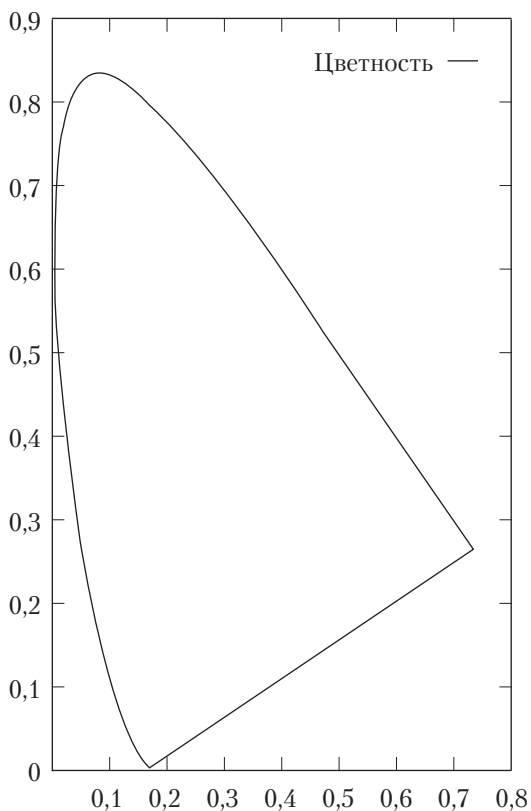


Рис. 3.6. Хроматическая диаграмма

Если у нас есть два разных цвета, то им соответствуют две точки на хроматической диаграмме. Рассмотрим все множество цветов, которые получаются путем смешивания этих двух заданных цве-