

ФГБОУ ВПО
«Воронежский государственный технический
университет»

О.Д. Козенков В.В. Ожерельев

ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

Утверждено Редакционно-издательским советом
университета в качестве учебного пособия

Воронеж 2013

УДК 004.92

Козенков О.Д. Основы компьютерной графики: учеб. пособие / О.Д. Козенков, В.В. Ожерельев. Воронеж: ФГБОУ ВПО «Воронежский государственный технический университет», 2013. 125 с.

В учебном пособии рассматриваются вопросы, относящиеся к растровой, векторной и фрактальной компьютерной графике. Учебное пособие призвано дать общее представление о предмете «Компьютерная графика» и ориентировано на студентов, для которых данный предмет не является профилирующим. С этой целью рассматриваются растр, алгоритмы растровой графики, математические основы растровой графики; математические основы векторной графики; цветовые модели, цветовые палитры; основные редакторы и форматы файлов растровой и векторной графики; фракталы и основы фрактальной графики; трехмерная графика; настольные издательские системы. Издание соответствует требованиям Федерального государственного образовательного стандарта высшего профессионального образования по направлению 150100 «Материаловедение и технологии материалов» (профиль «Физическое материаловедение»), дисциплине «Компьютерная графика».

Учебное пособие подготовлено в электронном виде в текстовом редакторе MS WORD XP 2003 и содержится в файле Осн. комп. граф. ФМ.doc.

Ил. 50. Библиогр.: 12 назв.

Рецензенты: кафедра высшей математики Воронежского института МВД России (нач. кафедры д-р физ.-мат. наук, проф. В.В. Меньших); д-р физ.-мат. наук, проф. И.Л. Батаронов

© Козенков О.Д., Ожерельев В.В., 2013

© Оформление. ФГБОУ ВПО

«Воронежский государственный технический университет», 2013

ВВЕДЕНИЕ

Компьютерная или машинная графика – это самостоятельная область научной и практической деятельности со своими задачами проблемами и спецификой, она охватывает все стороны формирования изображений с помощью компьютера. Компьютерная графика разрабатывает и использует программные системы, машинные языки, методы прикладной математики. Она предоставляет новые эффективные программные технические средства для проектировщиков, конструкторов, исследователей ученых и широкого круга обычных пользователей.

Современная компьютерная графика – это достаточно сложная, основательно проработанная и разнообразная научно-техническая дисциплина, которая сформировалась на базе синтеза таких наук, как аналитическая, прикладная, начертательная геометрия, программирование для ЭВМ, методы вычислительной математики и т.п. Некоторые ее разделы, такие как геометрические преобразования, способы описания кривых и поверхностей к настоящему времени уже исследованы достаточно полно. Ряд областей продолжает активно развиваться: методы растрового сканирования, удаление невидимых линий и поверхностей, моделирование цвета и освещенности, текстурирование, создание эффекта прозрачности и полупрозрачности и др.

Выделяют три вида компьютерной графики – это растровая, векторная и фрактальная. Кроме того, компьютерную графику можно разделить на: двухмерную (2D), то есть с плоской картинкой и трехмерную (3D), то есть с пространственной картинкой, имеющей три измерения.

Основные области применения компьютерной графики:

- научная графика;
- деловая графика;
- конструкторская графика;
- полиграфия;
- Web-дизайн;

- мультимедиа.

К основным задачам компьютерной (машинной) графики относятся:

- ввод (считывание) графической информации в ЭВМ;
- переработка информации в компьютере;
- вывод графической информации из ЭВМ и формирование изображений.

Таким образом, основные задачи компьютерной графики или как говорят машинной геометрии, т.е. автоматизированного геометрического моделирования и конструирования – это синтез и анализ геометрических объектов, и решение задач геометрического характера в ЭВМ.

Нетрудно представить, какое значение имеет и будет иметь компьютерная графика практически в любой области человеческой деятельности, начиная от технического проектирования и заканчивая искусством «для избранных».

Многие книги по компьютерной графике посвящены достаточно узким и специальным вопросам при этом не всегда имеется необходимая информация по общим вопросам предмета.

Поэтому цель данного учебного пособия – дать обзор основных результатов, полученных в области компьютерной графики, и познакомить читателя с математическими основами компьютерной графики, с оборудованием и алгоритмической реализацией задач компьютерной графики.

Данное пособие ориентировано на студентов, не специализирующихся в области компьютерной графики, и рассчитано на общее ознакомление с предметом.

Учебное пособие рассчитано на то, что читатель знаком с такими понятиями, как векторы и матрицы и с операциями над ними.

1. ГРАФИЧЕСКАЯ ПОДСИСТЕМА ЭВМ

1.1. Состав графической подсистемы ЭВМ

Графическая подсистема ЭВМ состоит из двух частей: аппаратной и программной. Аппаратная часть включает видеоадаптер (видеокарту), монитор и интерфейсы, обеспечивающие взаимодействие между видеоадаптером и чипсетом, а также между видеоадаптером и устройством отображения. Программная часть обеспечивает поддержку графических устройств и приложений на уровне базовой системы ввода-вывода (Basic input-output system, BIOS), операционной системы (ОС), драйверов и специализированных API – интерфейсов программирования приложений (application programming interface).

Графическая подсистема компьютера решает три группы задач:

1) задачи плоской (2D) графики – задачи создания, редактирования и отображения плоских изображений, в т.ч. задачи графического интерфейса пользователя (GUI - Graphic User Interface), который обеспечивает взаимодействие между различными программами и пользователем.

2) Задачи трехмерной (3D) графики. 3D-графика, как правило, представляет собой геометрические модели 3D-объектов, совокупность которых образует геометрическую модель некоторого трехмерного объема – трехмерную сцену.

Геометрические модели 3D-графики, как правило, создаются специальными программами, исполняемыми на центральном процессоре (ЦП), и хранятся в оперативной памяти (ОП) компьютера. В этом случае графическая подсистема компьютера используется для вывода и отображения трехмерных сцен. Современные видеоадаптеры способны при выполнении некоторых ограничений самостоятельно осуществлять создание и обработку геометрических моделей.

3) Задачи вывода и обработки видеографики. Графическая подсистема содержит аппаратные средства, обеспечивающие ускорение обработки, компрессии и декомпрессии видео

и работу с различными источниками видео (DVD, HDTV, PVR, Blu-ray и HD DVD).

1.2. Основные характеристики и классификация мониторов

Монитор (дисплей) – устройство, предназначенное для визуального отображения информации. Состоит из корпуса, блока питания, плат управления и экрана. Информация (видео-сигнал) для вывода на монитор поступает с видеоадаптера компьютера, либо с другого устройства, формирующего видеосигнал.

Современные мониторы являются растровыми устройствами: изображение на экране формируется из тысяч точек (пикселей).

Основные характеристики монитора:

- размер экрана (как правило, указывается длина диагонали в дюймах);
- соотношение сторон экрана;
- разрешение экрана (число пикселей по горизонтали и по вертикали);
- глубина цвета, определяется количеством бит на кодирование одного пикселя, от 1 (монохромный дисплей) до 32 бит;
- размер пикселя (зерна);
- время отклика пикселей;
- горизонтальный и вертикальный углы обзора;
- частота кадров.

По принципу формирования изображения мониторы делятся на:

- ЭЛТ (CRT, cathode ray tube) – мониторы на основе электронно-лучевой трубки;
- ЖК (LCD, liquid crystal display) – жидкокристаллические мониторы;
- Плазменные – на основе плазменной панели;
- Лазерные – на основе лазерной панели;

- OLED-монитор – на технологии OLED (organic light-emitting diode – органический светоизлучающий диод);

- Виртуальный ретинальный монитор — технология устройств вывода, формирующая изображение непосредственно на сетчатке глаза;

- Проектор.

По размерности отображения мониторы можно разделить на двухмерные (2D), формирующие одно изображение для обоих глаз, и трехмерные (3D), в которых для каждого глаза формируется отдельное изображение для получения эффекта объема.

Наиболее распространены мониторами в настоящее время являются мониторы на основе электронно-лучевой трубки (ЭЛТ) и жидкокристаллические (ЖК) мониторы.

1.3. Принцип работы ЭЛТ-монитора

Принципиально конструкция ЭЛТ для монитора аналогична конструкции телевизионного кинескопа (рис. 1.1). ЭЛТ представляет собой стеклянную колбу, дно которой покрыто слоем люминофора – вещества, излучающего свет при его бомбардировке электронами. Испускаемые нагретыми катодами электроны под воздействием высокой разности потенциалов фокусируются в пучки, ускоряются и направляются к экрану.

В цветных ЭЛТ используются три электронных пушки со своими катодами — по одному для каждого из основных цветов (Red – красный, Green – зеленый, Blue – синий), а слой люминофора составляют из близко расположенных точек группами по три, также в сочетании Red, Green, Blue — т.н. RGB – триада. Изменением тока каждого из трех электронных лучей можно добиться произвольного цвета элемента изображения, образуемого триадой точек.

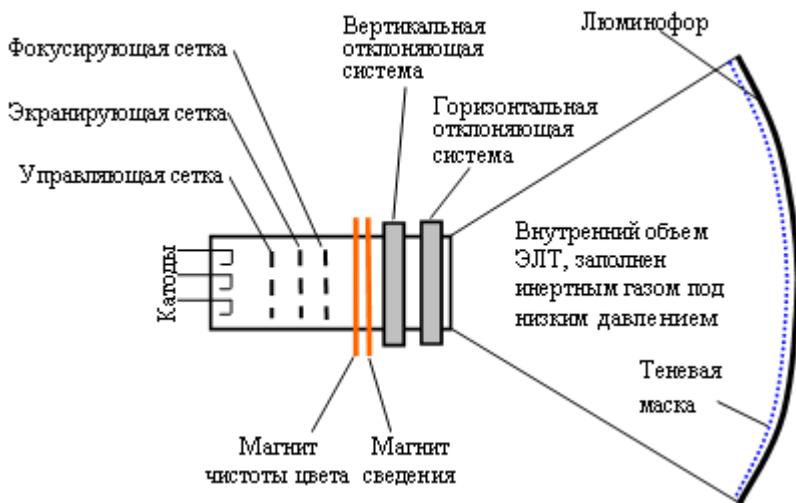


Рис. 1.1. Схема типичной цветной ЭЛТ

Изменяя напряжение, подаваемое на управляющую сетку, можно регулировать общую интенсивность электронных пучков (т.е. яркость свечения экрана). Напряжением, подаваемым на экранирующую сетку, осуществляется первичное ускорение электронов на пути к экрану, а фокусирующая сетка предназначена для «сжатия» электронных пучков, т.е. уменьшения их поперечного сечения. Сфокусированные и промодулированные по интенсивности электронные пучки с помощью магнитных полей, формируемых вертикальной и горизонтальной отклоняющими системами, направляются в различные точки на экране трубки.

Для точной фокусировки электронных лучей в цветных ЭЛТ непосредственно перед люминофорным покрытием располагают теневую маску, имеющую отверстия с размерами, близкими к размерам отдельной точки люминофора (рис. 1.3).

От точности наведения электронных пучков зависит такая характеристика монитора, как чистота цвета. Для повышения точности их юстировки предназначен установленный на горловине трубки магнит чистоты цвета. Кроме того, поскольку

ку тневая маска пропускает электроны только через микроскопические отверстия, все три пучка должны пересекаться именно в этих отверстиях. Для решения этой задачи предназначен установленный на горловине трубки магнит сведения. Регулируя его положение (или ток через обмотку в случае использования электромагнита), можно добиться точного сведения пучков в центре экрана (так называемое статическое сведение). Для сведения пучков по краям экрана (динамического сведения) используется катушка сведения, сигнал на которую подается со схемы управления разверткой.

В зависимости от конструкции цветоделительной маски различают ЭЛТ следующих типов (рис. 1.2):

- ЭЛТ с тневой маской;
- ЭЛТ с апертурной решеткой;
- ЭЛТ со щелевой маской.

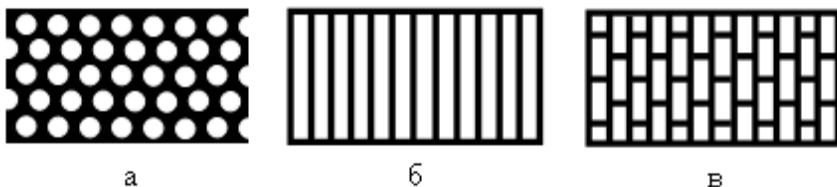


Рис. 1.2. Типы цветоделительной маски: а – тневая маска, б – апертурная решетка, в – щелевая маска

Тневая маска (рис. 1.2, а) выполнена в виде металлической пластины с круглыми отверстиями. Изготавливается маска в большинстве случаев из инвара (магнитный сплав железа (64%) с никелем (36%)). Этот материал характеризуется очень низким коэффициентом теплового расширения, что обеспечивает стабильность формы маски при разогреве в процессе электронной бомбардировки. Поверхность экрана у кинескопов с тневой маской обычно является частью сферы большого радиуса. Это сделано для того, чтобы электронный луч в центре экрана и по краям имел одинаковую толщину. За счет улучшения систем управления отклоняющей системой удастся

выпускать трубки с практически плоской поверхностью экрана.

Апертурная решетка (рис. 1.2, б) представляет собой тонкую фольгу, в которой выполнены частые вертикальные отверстия. Поперек размещают две-три нити, обеспечивающие жесткость конструкции. Люминофор на дне колбы также располагается в виде вертикальных чередующихся полосок разных цветов. Данная технология применяется фирмой Sony (кинескоп Trinitron), а также фирмами Mitsubishi и ViewSonic. Особенности технологии позволяют увеличить процент электронов, попадающих на люминофор, и добиться лучшей яркости изображения. В мониторах с апертурной решеткой поверхность экрана представляет собой часть цилиндра большого радиуса.

Щелевая маска – технология, разработанная компанией NEC и представляющая собой комбинацию теневой маски и апертурной решетки. В данном случае люминофорные элементы расположены в вертикальных эллиптических ячейках, а маска сделана из вертикальных линий (рис. 1.2, в).

Для формирования изображения в мониторе используются специальные сигналы. В цикле сканирования луч движется по зигзагообразной траектории от левого верхнего угла до правого нижнего (рис. 1.3). Прямой ход луча по горизонтали осуществляется сигналом строчной (горизонтальной) развертки, а по вертикали – кадровой (вертикальной) развертки.

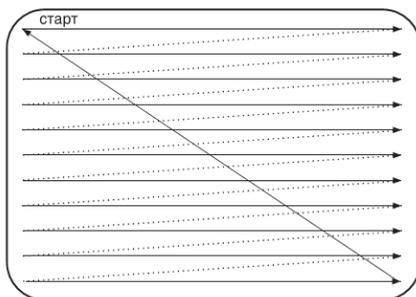


Рис. 1.3. Формирование раstra на экране монитора

1.4. Принцип работы ЖК-монитора

Экран ЖК-монитора состоит из двух стеклянных панелей (подложек), между которыми размещен слой жидкокристаллического вещества и представляет собой совокупность отдельных элементов - ЖК-ячеек, каждая из которых формирует 1 пиксел изображения. Поскольку ЖК-ячейка сама не генерирует свет, в ЖК-мониторах всегда используют подсветку. Для подсветки используют флуоресцентные лампы, характеризующиеся низким энергопотреблением.

ЖК-ячейка представляет собой электронно-управляемый светофильтр, основанный на эффекте поляризации световой волны. Вытянутость формы молекул (т.н. нематические молекулы) жидкокристаллического вещества приводит к их упорядоченной ориентации и возникновению оптической анизотропии: показатель преломления ЖК-вещества зависит от направления распространения световой волны. Другим важным свойством является наличие у молекул ЖК-вещества электрического дипольного момента, что дает возможность управлять их ориентацией внешним электрическим полем.

В ЖК-мониторах обычно используют ЖК-ячейки с твистированной (закрученной на 90^0) ориентацией молекул (рис. 1.4, а). На обе подложки нанесены специальные канавки, развернутые на 90^0 и задающие ориентацию молекул ЖК-вещества.

Подложки являются также поляризационными фильтрами: они пропускают световую волну только с линейной поляризацией. Верхняя подложка называется поляризатором, а нижняя анализатором. Векторы поляризации поляризатора и анализатора развернуты на 90^0 друг относительно друга.

В отсутствии внешнего поля (рис. 1.4, а) падающий на ячейку свет после прохождения через поляризатор приобретает определенную поляризацию, совпадающую с ориентацией молекул ЖК-вещества у поверхности поляризатора. По мере распространения световой волны по направлению к анализатору ее плоскость поляризации поворачивается на угол 90^0 . В

итоге свет свободно проходит через анализатор, поскольку плоскости поляризации света и анализатора совпадают.

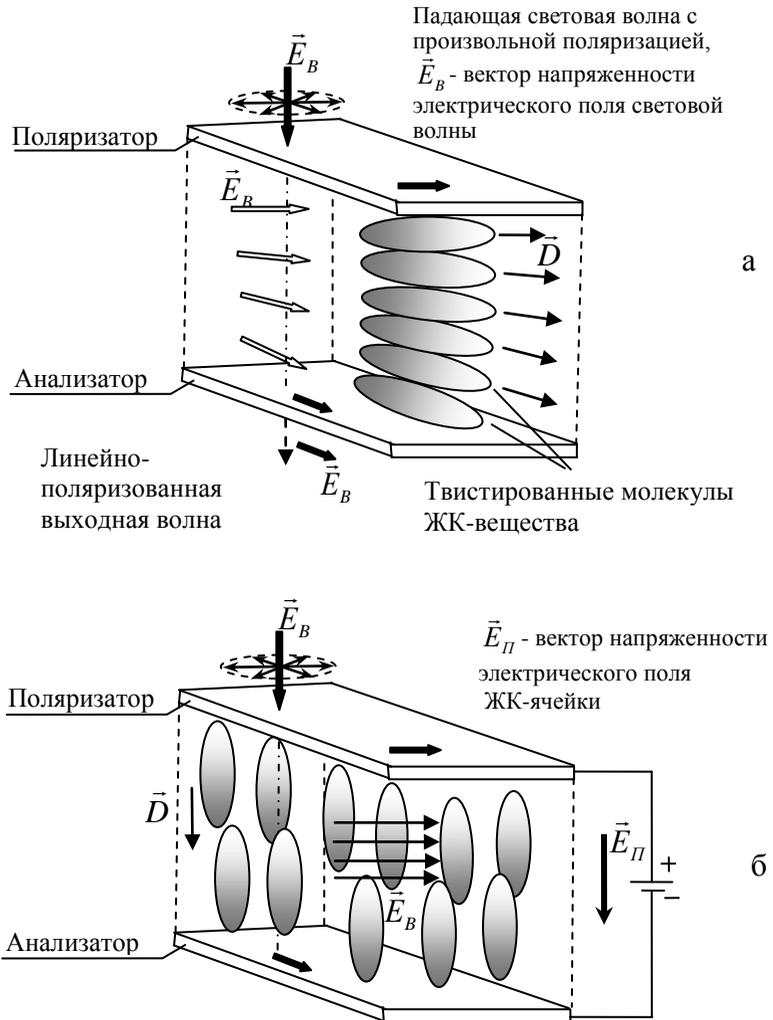


Рис. 1.4. Принцип действия ячейки ЖК-монитора

Если к подложкам приложить напряжение (3-10 В), то возникающее электрическое поле ориентирует молекулы параллельно силовым линиям поля (рис. 4, б). Твистированная структура ЖК-вещества исчезает, и при прохождении света поворота плоскости поляризации не происходит. В результате плоскость поляризации света не совпадает с плоскостью поляризации анализатора, и ЖК-ячейка оказывается непрозрачной.

С целью получения цветного изображения ЖК-ячейки объединяют в триады, каждая ячейка которой снабжена светофильтром трех основных цветов (красного, зеленого и синего).

Для повышения контрастности и быстродействия в современных ЖК-экранах применяют технологию TFT (thin film transistor – тонкопленочный транзистор). В этом случае каждая ЖК-ячейка располагается между контактами тонкопленочного транзистора, с помощью которого и подается напряжение на ЖК ячейку.

Преимуществами ЖК мониторов по сравнению с ЭЛТ состоит в существенно более низком энергопотреблении и меньших габаритах, а также в отсутствии вредного электромагнитного излучения.

1.5. Графические адаптеры

1.5.1. Структура графического адаптера

Современные графические адаптеры (видеоадаптеры, видеокарты) используют последние достижения трехмерной компьютерной графики, реализуемые на аппаратном уровне и программным способом. Выполнены в виде печатной платы (рис. 1.5) и вставляется в разъем расширения системной платы (AGP, PCI Express). Широкое распространение получили также и интегрированные в системную плату видеокарты — как в виде отдельного чипа, так и в качестве составляющей части северного моста чипсета или ЦП. Ведущими производителями графических адаптеров в настоящее время являются компании NVIDIA и ATI.



Рис. 1.5. Графический адаптер Nvidia GeForce 6600GT

Рассмотрим функциональные блоки, входящие в состав графического адаптера (рис. 1.6).

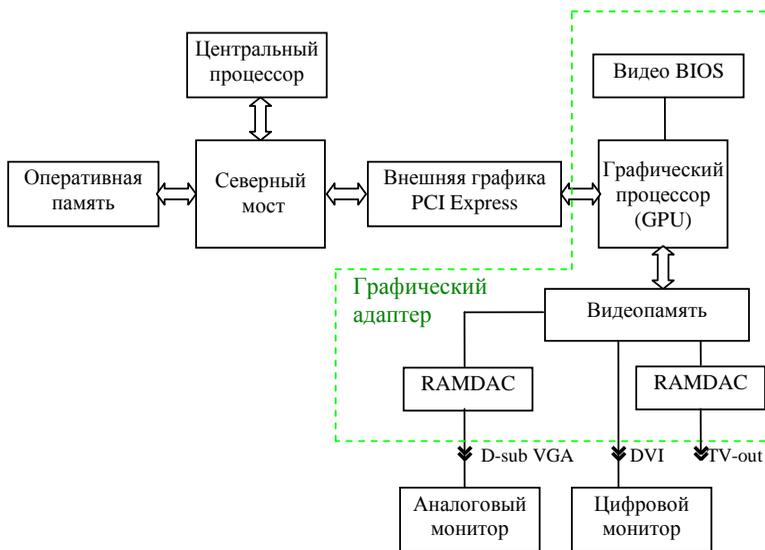


Рис. 1.6 Структурная схема графического адаптера

Графический процессор (GPU – Graphics processor unit) – графическая многопроцессорная система, предназначенная для выполнения графического рендеринга (термином ренде-

ринг в к компьютерной графике обозначают процесс получения изображения по модели с помощью компьютерной программы).

Благодаря специализированной конвейерной архитектуре современные графические процессоры намного эффективнее в обработке графической информации, чем ЦП. Графический процессор в современных видеоадаптерах применяется в качестве ускорителя трёхмерной графики.



Рис. 1.7. Графический процессор nVidia GeForce 3 Ti 200

Как и ЦП, GPU характеризуются сложной внутренней архитектурой, рабочей частотой графического ядра, технологическими нормами, по которым изготовлена микросхема и другими параметрами.

GPU могут применяться не только в составе дискретной видеокарты, но и быть встроенными в северный мост либо в гибридный процессор.

Современные GPU могут применяться не только для графических вычислений, но и для вычислений общего характера. Так, компанией NVIDIA разработана технология CUDA (Compute Unified Device Architecture) – программно-аппаратная архитектура параллельных вычислений на GPU.

Video BIOS – микросхема, установленная на плате видеоадаптера, в которой хранятся программы, обеспечивающие инициализацию видеокарты, поддержку простейшего интерфейса пользователя, базовые компоненты драйвера и другие необходимые компоненты.

Видеопамять. Важную роль в повышении производительности видеоадаптера играют характеристики видеопамяти, определяемые ее типом, частотой работы, величиной задержек, шириной шины памяти. ЦП компьютера направляет данные в видеопамять, а графический процессор видеокарты считывает оттуда информацию. Кроме того, в видеопамяти хранятся кадровый буфер и промежуточные данные, необходимые графическому процессору.

Объем памяти современных видеокарт может достигать 6 Гб (NVIDIA QUADRO 6000), а пиковая скорость передачи данных может составлять более 300 Гб/с. Объем видеопамяти, установленной на карте, важен в первую очередь для обработки трехмерных изображений с текстурами высокого разрешения при большой глубине цвета. Большая емкость памяти необходима для быстрого вывода трехмерных изображений с высоким разрешением, поскольку в видеопамять при этом загружается огромный объем дополнительной информации, прежде всего массивы текстур.

RAMDAC. Графический процессор, завершив обработку изображения, передает информацию либо в цифроаналоговый преобразователь (RAMDAC) для вывода на аналоговый монитор, либо в схему формирования цифрового сигнала TDMS (а через нее на цифровой видеовыход DVI) для вывода на цифровой монитор.

RAMDAC состоит из памяти с произвольным доступом (RAM) и цифроаналогового преобразователя (DAC — Digital to Analog Converter).

Аналоговые компоненты видеокарты обеспечивают стабильное питание микросхем и формирование сигналов нужной формы на аналоговых выходах (на монитор и телевизор).

Интерфейс видеокарты обеспечивает сопряжение с серверным мостом чипсета. Высокая производительность видеоадаптеров требует использования специализированных интерфейсов, обеспечивающих высокую скорость обмена данными с системной памятью.

Получившему широкое распространение интерфейсу AGP (Accelerated Graphic Port – ускоренный графический порт) в настоящее время пришел на смену последовательный интерфейс PCI Express, обеспечивающий максимальную пропускную способность шины 4000 Мбайт/с.

1.5.2. Технологии SLI и CrossFire

Для параллельного использования двух видеоадаптеров с целью повышения производительности в 3D приложениях компанией NVIDIA разработана технология SLI (Scalable Link Interface – масштабируемый соединительный интерфейс). Для ее использования необходимо наличие двух видеокарт с поддержкой SLI и системной платы с чипсетом, поддерживающим SLI (можно использовать переходную плату SLI-bridge для связи видеокарт и соответствующий драйвер).

Технология SLI поддерживает два режима работы пары видеокарт: Split Frame Rendering (SFR) и Alternate Frame Rendering (AFR). В режиме SFR происходит разделение кадра на две части, за рендеринг каждой из которых отвечает отдельный видеоадаптер. В режиме AFR одна видеокарта обрабатывает четные кадры, а другая – нечетные.

Позднее аналогичная технология, получившая название CrossFire, была представлена и компанией ATI.

1.5.3. Графические API

Интерфейсы программирования приложений (API) предоставляют разработчикам аппаратного и программного обеспечения средства создания драйверов и программ, работающих на большом числе платформ. Использование API избавляет разработчиков программ от необходимости работать с низкоуровневыми командами конкретной видеокарты, облегчая процесс создания программ и делая их универсальными.

В настоящее время основными графическими API, поддерживаемыми большинством видеоадаптеров, являются OpenGL и DirectX.

OpenGL (Open Graphics Library – открытая графическая библиотека) – спецификация, определяющая платформонезависимый программный интерфейс для написания приложений, использующих двухмерную и трёхмерную компьютерную графику. Используется для создания игр, систем автоматизированного проектирования (САПР), виртуальной реальности, визуализации в научных исследованиях.

DirectX (компания Microsoft) используется в игровых приложениях, работающих под управлением операционной системы Windows.

2. ЦВЕТОВЫЕ МОДЕЛИ

2.1. Свет и особенности человеческого цветовосприятия

Видимый свет представляет собой электромагнитные волны с длиной волны $\lambda = 400..700$ нм. Энергия световой волны пропорциональна квадрату амплитуды A .

На практике обычно приходится иметь дело не со строго монохроматическим светом, а с набором волн, обладающих различными частотами и амплитудами. Такой поток световых волн можно характеризовать энергетической спектральной кривой $F(\lambda)$ (рис. 2.1), где само значение $F(\lambda)$ представляет собой энергетический вклад волн с длиной волны λ в суммарный световой поток.



Рис. 2.1. Энергетическая спектральная кривая светового потока

Во внутренней оболочке глаза – сетчатке – имеются светочувствительные рецепторы двух различных типов: т.н. палочки и колбочки. В них энергия света преобразуется в сигнал, который передается по зрительному нерву в мозг.

Палочки обладают широкой спектральной кривой чувствительности, они реагируют только на суммарную энергию света. Они ответственны за восприятие яркостной составляющей света. Палочки содержат пигмент с максимальной чувствительностью для длины волны ≈ 510 нм (зеленая часть спектра).

Колбочки сосредоточены в центре сетчатки, в желтом теле напротив зрачка. Они обладают узкой спектральной чувствительностью и именно они ответственны за цветовосприятие.

Максимум суммарной чувствительности палочек и колбочек (рис. 2.2) приходится на 550 нм, что соответствует желто-зеленому свету.

Известен факт избирательной чувствительности зрения человека к волнам различной длины: человек лучше всего видит зеленый свет, хуже – красный и еще хуже – синий.

Это связано с тем, что колбочки бывают трех типов, отвечающих за чувствительность к длинным, средним и коротким волнам. На рис. 2.3 представлены графики функций чувствительности для всех трех типов колбочек. Видно, что у одной из них пик чувствительности приходится на волны с короткой длиной волны (синий цвет, В, Blue), у другой - на волны средней длины волны (желто-зеленый цвет, G, Green), а у третьей - на волны с большой длиной волны (красный цвет, R, Red).

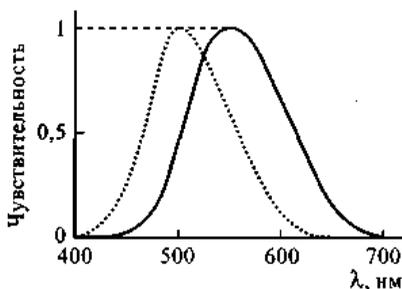


Рис. 2.2. Чувствительность палочек (пунктирная кривая) и суммарная чувствительность палочек и колбочек (сплошная кривая)

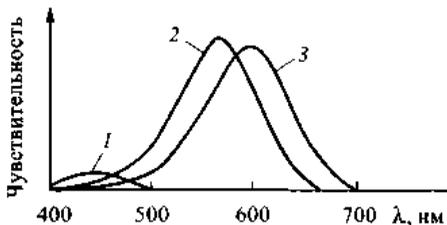


Рис. 2.3. Функции чувствительности трех типов колбочек: 1 – синих, 2 – зеленых, 3 – красных

2.2. Цветовая модель RGB

Для описания цвета в компьютерной графике вводится понятие цветовой модели.

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

Модель, в основе которой лежат красный, зеленый и синий цвета называется цветовой моделью RGB – по первым буквам английских наименований соответствующих цветов: Red – красный, Green – зеленый, Blue – синий.

Так как в модели RGB используются три независимых значения интенсивностей трех основных цветов, то ее можно представить в трехмерной системе координат, как показано на рис. 2.4.

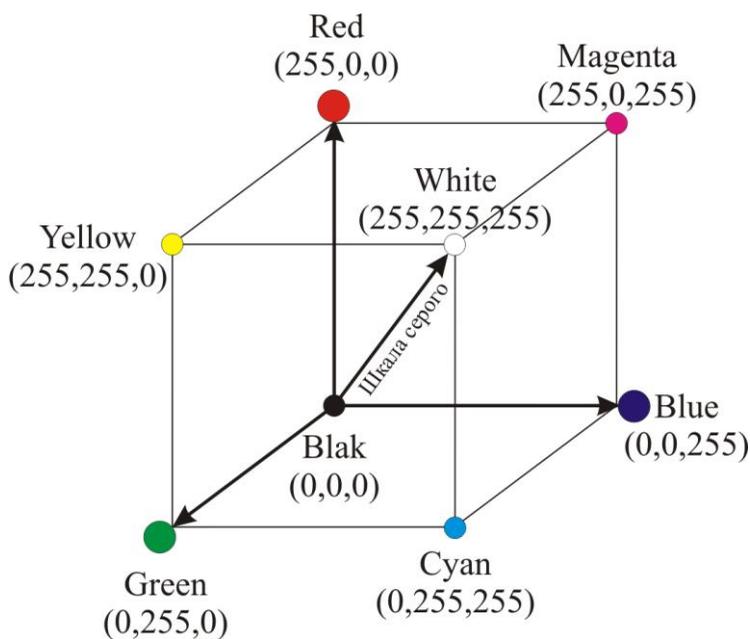


Рис. 5.4. Графическое представление цветовой модели RGB

Каждая координата отражает вклад соответствующей составляющей в конкретный цвет в диапазоне от нуля до максимального значения. В результате получается цветовой куб, внутри которого находятся все возможные цвета, образуя цветное пространство модели RGB. Объем такого куба, соответствует количеству дискретных цветов. Не трудно посчитать: так как по каждой из трех осей основных цветов можно отложить 256 значений интенсивностей, то число цветов $256^3 = 16777216$.

Отметим особые точки и линии модели RGB:

- Начало координат. В этой точке все составляющие равны нулю, то есть излучение отсутствует, это соответствует темноте – черному цвету. Итак, начало координат – черный цвет;

- Точка, лежащая на пространственной диагонали куба, исходящей из начала координат. В этой точке все основные цвета имеют максимально возможные значения, что соответствует белому цвету;

- Пространственная диагональ куба, исходящая из начала координат. На этой линии располагаются оттенки серого – от черного до белого. Это происходит потому, что интенсивности всех трех основных цветов на пространственной диагонали куба одинаковы и их значения изменяются от нуля до максимально возможных. Эту линию называют шкалой серого – Grayscale;

- Три вершины куба обозначают чистые исходные цвета интенсивности, которых изменяются от нуля до максимального значения по соответствующим осям;

- Оставшиеся три вершины куба – это результат двойного смешения основных цветов с максимальной интенсивностью – Cyan, Magenta, Yellow;

- Плоскости Red-Green, Green-Blue, Red-Blue – дают цвета от двойного смешения соответствующих основных цветов с различной интенсивностью.

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

2.3. Цветовая модель СМΥК

Цветовая модель СМΥК показана на рис. 2.5. Данная цветовая модель в отличие от модели RGB описывает отражаемые цвета. К отражаемым относятся цвета, которые не излучаются, а используют белый цвет, вычитая из него определенные цвета при отражении. Такие цвета называются субтрактивными (вычитательными), поскольку они остаются после вычитания основных аддитивных цветов. Существует три основных субтрактивных цвета: голубой – Cyan, пурпурный – Magenta, желтый – Yellow. Эти цвета составляют так называемую полиграфическую триаду. При печати красками этих цветов на белой бумаге поглощаются красная, зеленая и синяя составляющие белого цвета, таким образом, что большая часть видимого цветового спектра может быть репродуцирована на бумаге.

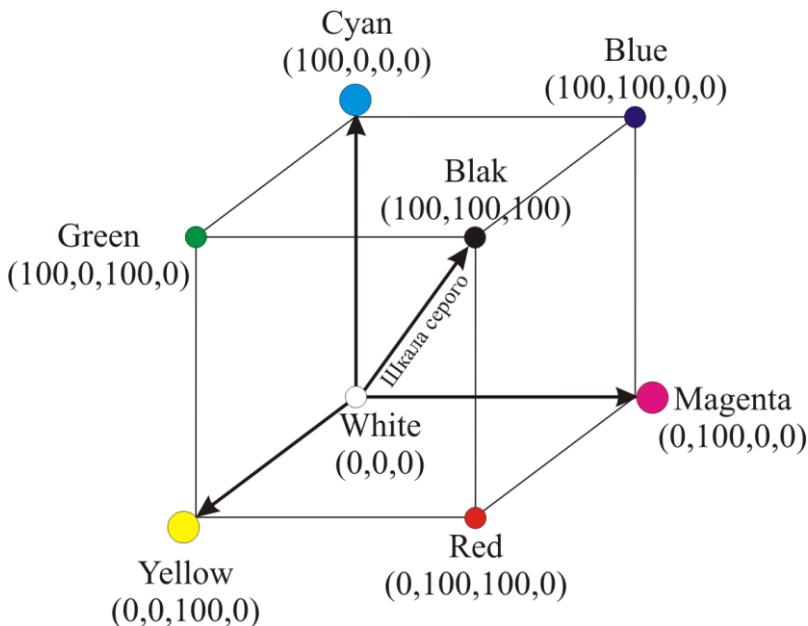


Рис. 2.5. Графическое представление цветовой модели СМΥК

При смешении двух субтрактивных составляющих результирующий цвет затемняется. При смешении всех трех должен получиться черный цвет. При полном отсутствии краски остается белый цвет – цвет бумаги.

В итоге получается, что нулевые значения составляющих дают белый цвет, а максимальные значения должны давать черный цвет, равные значения основных составляющих лежат на пространственной диагонали цветового куба и дают оттенки серого цвета. Кроме того, так же как и в модели RGB, имеются чистые субтрактивные цвета и их двойные сочетания.

Основная проблема модели CMYK заключается в том, что она должна описывать реальные полиграфические краски, которые далеко не так идеальны, как цветовой луч.

Полиграфические краски имеют примеси и технологически достаточно трудно изготовить их со строго заданным цветом. Кроме того, и бумага, используемая при печати должна быть идеально белой. По этим причинам полиграфические краски не могут перекрыть весь цветовой диапазон. Не идеальность красок приводит к тому, что при смешении трех основных цветов, которое должно давать черный цвет, получается какой-то неопределенный грязный темный цвет. Это скорее темно-коричневый, нежели глубокий черный цвет.

Для компенсации этого недостатка в число основных полиграфических красок была внесена черная краска. Именно она добавила последнюю букву в название модели CMYK, хотя и не совсем обычно: С – это Cyan (голубой), М – это Magenta (пурпурный), Y – это Yellow (желтый), К – это black (черный), то есть от слова взята последняя буква.

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

Модели RGB и CMYK, хотя и связаны друг с другом, однако их взаимные переходы друг в друга (конвертирование) не происходят без потерь, поскольку цветовой охват у них разный. Речь идет лишь о том, чтобы уменьшить потери до при-

емлемого уровня. Это вызывает необходимость сложных калибровок всех аппаратных частей, составляющих работу с цветом: сканера, который осуществляет ввод изображения, монитора, по которому судят о цвете и корректируют его параметры, выводного устройства, которое создает оригиналы для печати, печатного станка, которые выполняет последнюю стадию размножения оригиналов для печати.

Для устранения аппаратной зависимости был разработан ряд так называемых перцепционных, то есть интуитивных цветовых моделей. В их основу заложено раздельное определение яркости и цветности.

2.4. Цветовая модель HSB

На цветовом круге, показанном на рис. 2.6, основные цвета моделей RGB и CMY находятся в следующей зависимости: каждый цвет расположен напротив дополняющего его комплементарного цвета, при этом комплементарный цвет находится между цветами, с помощью которых он был получен. Например, сложение зеленого и красного цветов дает желтый цвет.

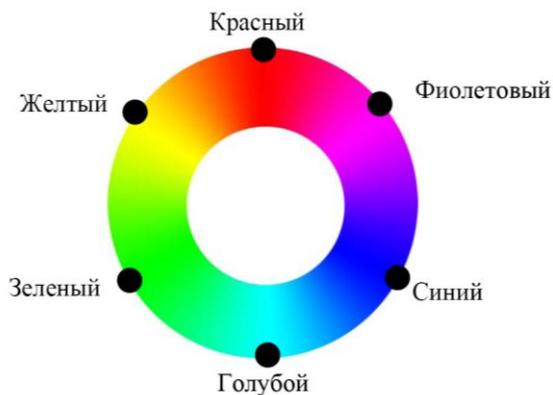


Рис.2.6. Цветовой круг

Чтобы усилить какой-либо цвет, нужно ослабить дополняющий его цвет, то есть цвет, расположенный напротив него

на цветовом круге. Например, чтобы изменить общее световое решение в сторону голубых тонов, следует снизить в нем содержание красного цвета.

По краю цветового круга располагаются так называемые спектральные цвета или цветовые тона (Hue), которые определяются длиной световой волны, отраженной от непрозрачного объекта или прошедшей через прозрачный объект. Цветовой тон характеризуется положением на цветовом круге и определяется величиной угла в диапазоне от 0° до 360° . Эти цвета обладают максимальной насыщенностью, то есть синий цвет еще синее быть уже не может.

Следующим параметром является насыщенность цвета – Saturation. Это параметр цвета, определяющий его чистоту.

Уменьшение насыщенности цвета означает его разбеление. Цвет с уменьшением насыщенности становится пастельным, блеклым, размытым. На модели все одинаково насыщенные цвета располагаются на концентрических окружностях, то есть можно говорить об одинаковой насыщенности, например, зеленого и пурпурного цветов, и чем ближе к центру круга, тем все более разбавленные цвета получаются. В самом центре круга любой цвет максимально разбеливается и становится белым. Поэтому работу с параметром насыщенности можно характеризовать как добавление в спектральный цвет определенного процента белой краски.

Еще одним параметром является яркость цвета (Brightness) – это параметр цвета, определяющий освещенность или затемненность цвета. Уменьшение яркости цвета означает его зачернение. Поэтому работу с параметром яркости можно характеризовать как добавление в спектральный цвет определенного процента черной краски.

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

Эта модель уже гораздо ближе к традиционному пониманию работы с цветом. Можно определять сначала цветовой тон (Hue), а затем насыщенность (Saturation) и яркость (Bright-

ness). Такая модель получила название по первым буквам приведенных выше английских слов – HSB. Она показана на рис. 2.7.

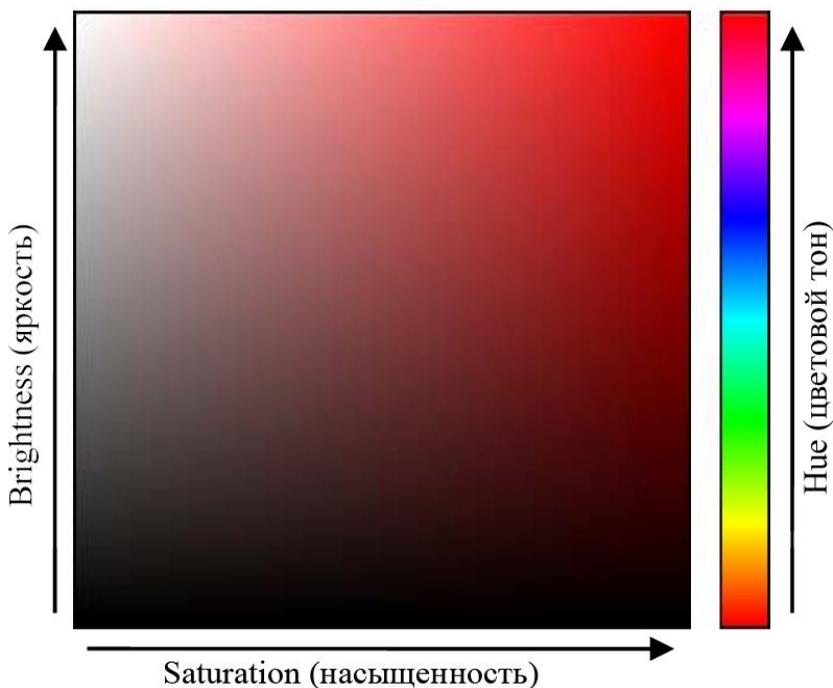


Рис. 2.7. Цветовая модель HSB

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

3. ЦВЕТОВЫЕ ПАЛИТРЫ

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

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

3.1. Индексная палитра

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

3.2. Фиксированная палитра

В тех случаях, когда цвет закодирован двумя байтами – режим High Color, на экране возможно отображение 65 тысяч цветов. Разумеется это не все возможные цвета, а лишь одна двести пятьдесят шестая доля общего непрерывного спектра цветов, доступных в режиме True Color. В таком изображении каждый двухбайтный код тоже выражает какой-то цвет из об-

щего спектра. Но в данном случае нельзя приложить к файлу индексную палитру, в которой было бы записано – какой код, какому цвету соответствует, так как в этой таблице было бы 65 тысяч записей, и ее размер составил бы сотни тысяч байтов. Вряд ли есть смысл прикладывать к файлу таблицу, которая по размеру может быть больше самого файла. В этом случае используют понятие фиксированной палитры. Ее не надо прикладывать к файлу, так как в любом графическом файле, имеющем шестнадцатиразрядное кодирование цвета, один и тот же код выражает один и тот же цвет.

3.3. Безопасная палитра

Безопасную палитру используют в web-графике. Поскольку скорость передачи данных в Интернете пока оставляет желать лучшего, для оформления web-страниц не применяют графику, имеющую кодирование цвета выше 8-разрядного. При этом возникает проблема, связанная с тем, что создатель web-страницы не имеет ни малейшего понятия о том, на какой модели компьютера и под управлением каких программ будет просматриваться его произведение. В связи с чем, было принято следующее решение. Все наиболее популярные программы для просмотра web-страниц – браузеры, заранее настроены на некоторую одну фиксированную палитру. Если разработчик web-страницы при создании иллюстраций будет применять только эту палитру, то он может быть уверен, что пользователи всего мира правильно увидят его рисунок.

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

Фиксированная цветовая палитра, жестко определяющая индексы для кодирования 216 цветов, называется безопасной палитрой.

4. АЛГОРИТМЫ РАСТРОВОЙ ГРАФИКИ

4.1. Растровые представления изображений

Цифровое изображение – это набор точек (пикселей) изображения. Пиксель – сокращение от *pix element*, наименьший логический элемент двумерного цифрового изображения в растровой графике, а также физический элемент светочувствительной матрицы и элемент матрицы дисплеев, формирующих изображение. Пиксель представляет собой неделимый объект, характеризуемый определённым цветом. Растровое компьютерное изображение состоит из пикселей, расположенных по строкам и столбцам. Чем больше пикселей на единицу площади содержит изображение, тем более оно детально.

Каждая точка цифрового изображения характеризуется координатами x и y и яркостью $V(x,y)$ (Рис. 4.1.). Это дискрет-

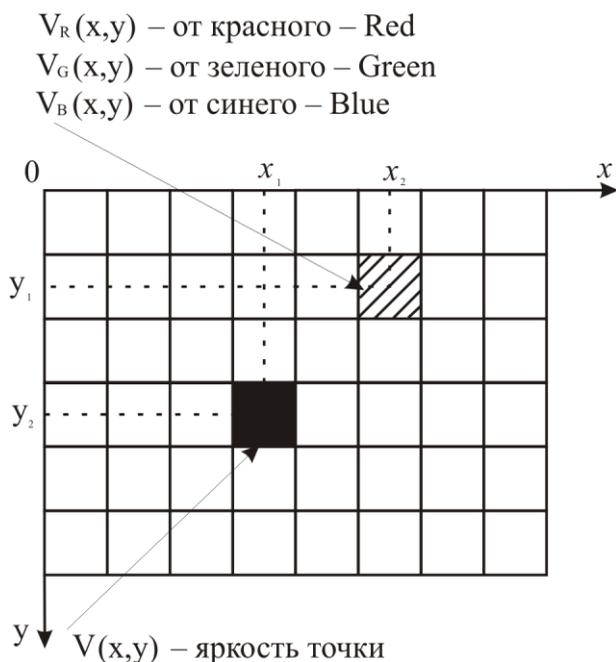


Рис. 4.1. Координаты и яркость черно-белого и цветного пикселей

ные, обычно целые величины. В случае черно-белого изображения каждый пиксель характеризуется двумя координатами x и y и яркостью $V(x,y)$. В случае цветного изображения, каждый пиксель характеризуется координатами x и y и тремя яркостями основных цветов, комбинируя которые можно получить большое количество различных оттенков: V_R – яркость красного, V_B – яркость синего, V_G – яркость зеленого. На градацию яркости в черно-белом изображении обычно отводится 1 байт, причем 0 – черный цвет, а 255 – белый (максимальная интенсивность). 1 байт – это 8 бит. Одна ячейка памяти содержит 1 бит информации. Так как физически ячейка может находиться в двух состояниях, на пример: отсутствие заряда или магнитного момента в ячейке или их наличие, то, соответственно, этим двум физическим состояниям приписываются значения – 0 или 1. Не трудно посчитать, что для двух ячеек число возможных комбинаций равно 4, для трех – 9, для четырех – 16 и так далее. То есть число возможных комбинаций располагается по степеням двойки от числа ячеек. Из-за этой особенности в компьютерах используется двоичная система счисления. Не трудно видеть, что для 1 байта = 8 бит число возможных комбинаций составит $2^8 = 256$, для 12 бит $2^{12} = 4096$.

Для цветного изображения обычно отводится по 1 байту на градации яркостей каждого из трех основных цветов. Возможно кодирование градаций яркостей другим количеством битов – 4 или 12, но человеческий глаз способен различать только 8 бит градаций на каждый цвет, хотя специальные устройства могут потребовать и более точную передачу цветов.

4.2. Растр

Растр – изображение, построенное из отдельных растровых элементов (точек), как правило, расположенных регулярно. В телевидении, технике видеозаписи, в системах отображения графической информации растр – последовательность параллельных друг другу видеострок на экране электронно-

лучевой трубки или иного оборудования, возникающая в результате работы системы развёртки.

На рис. 4.1 изображен растр, элементами которого являются квадраты. Такой растр называется квадратным. Именно такие растры являются наиболее часто используемыми. Хотя принципиально возможно использование в качестве растрового элемента фигуры любой другой формы, которая удовлетворяет следующим требованиям:

- форма всех растровых элементов должна быть одинаковой;

- растровые элементы должны полностью покрывать поверхность без наезжания и дырок.

В качестве растровых элементов, кроме квадрата или прямоугольника, возможно использование равносторонних треугольников или правильных шестиугольников (рис. 4.2). Возможно, построить растры, используя неправильные треугольники или многоугольники при условии, что они удовлетворяют требованиям, предъявляемым к форме растровых элементов, однако практический смысл в подобных построениях отсутствует, так как создает неоправданные технологические трудности в процессе изготовления растровых элементов сложной формы.

4.3. Построение линий в прямоугольном и гексагональном растре

В качестве прямоугольного растра рассмотрим квадратный растр. Это не меняет принципиальным образом сути рассуждений, но значительно облегчает их.

В квадратном растре построение прямой линии можно осуществить двумя способами:

- 1) Восемьсвязная линия. При этом соседние пиксели образующие линию могут находиться в одном из восьми возможных положений, как это показано на рис. 4.3. Недостаток такого способа построения линии – это малая ее толщина при угле 45° .

2) Четырехсвязная линия. Соседние пиксели, образующие линию, могут находиться в одном из четырех возможных положений, как это показано на рис. 4.4. Недосток такого построения – это избыточная толщина линии при угле в 45° .

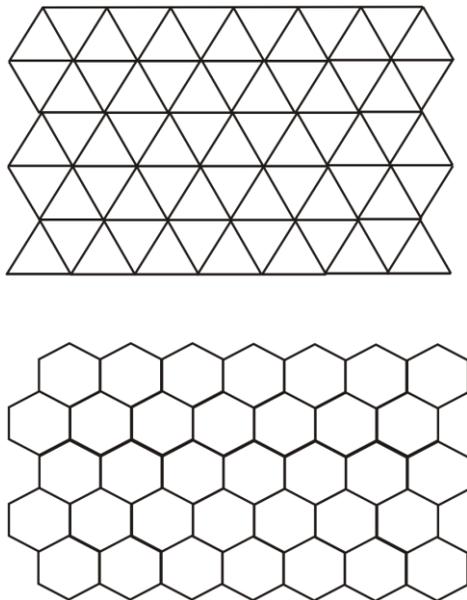


Рис. 4.2 Пример построения растров с треугольными и гексагональными растровыми элементами

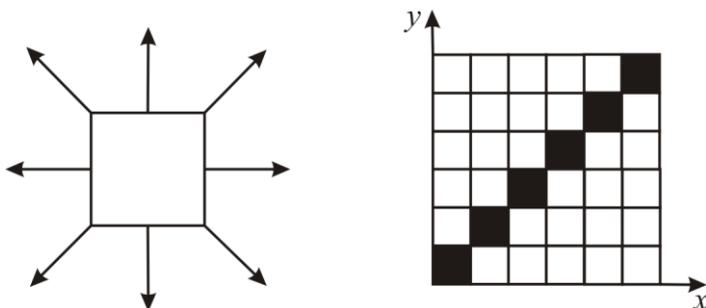


Рис. 4.3. Построение восьмисвязной линии в квадратном растре

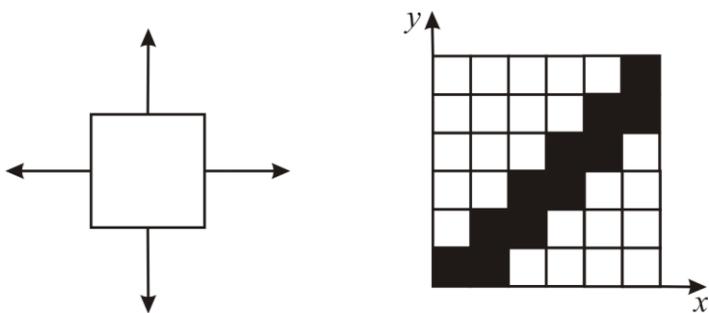


Рис. 4.4. Построение четырехсвязной линии в квадратном растре

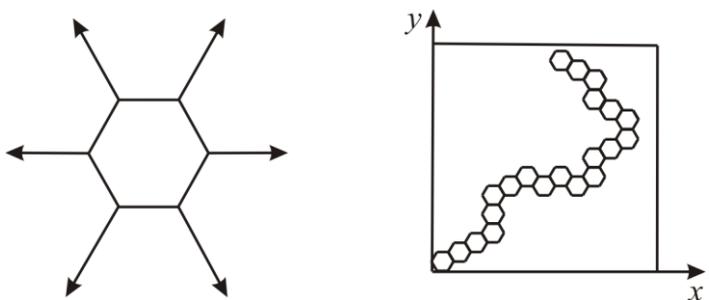


Рис. 4.5. Построение шестисвязной линии в гексагональном растре

В гексагональном растре строятся шестисвязные линии (рис. 4.5). Такие линии более стабильны по ширине, то есть дисперсия ширины линии меньше, чем в квадратном растре.

Мы рассмотрели квадратный (прямоугольный) и гексагональный растр. Естественно возникает вопрос. Какой растр лучше, то есть, каким образом можно оценить качество раstra?

Одним из способов оценки качества раstra является передача по каналу связи кодированного, с учетом используемого раstra, изображения с последующим восстановлением и визуальным анализом достигнутого качества. Экспериментально

и математически доказано, что гексагональный растр несколько лучше квадратного (прямоугольного), так как обеспечивает наименьшее отклонение изображения от оригинала. Однако разница между квадратным и гексагональным растром не велика и не имеет отличия принципиального характера, в то время как технологически изготовление квадратного растра намного проще.

4.4. Моделирование гексагонального растра

Гексагональный растр, обладающий определенными преимуществами перед прямоугольным, возможно смоделировать на основе квадратного. Для этого достаточно представить шестиугольник в виде прямоугольника. Определим, какие пропорции должен иметь элемент гексагонального изображения. Как следует из рис. 4.6 элемент гексагонального изображения может быть представлен в виде прямоугольника со сторонами пропорциональными расстоянию между центрами гексагонов составляющих его при одинаковом количестве гексагональных элементов по оси x и y . Исходя из этих условий, определим параметры элемента гексагонального изображения в виде соотношения сторон прямоугольника:

$$\frac{l_x}{l_y} = \frac{\frac{a}{2} 2 \tan 30^\circ}{\frac{3}{2} a} = \frac{2}{\sqrt{3}} \approx 1,15 \quad (4.1)$$

Таким образом, исходя из выражения (5.1) возникает принципиальная возможность моделирования гексагонального растра на основе прямоугольного. Для этого существуют следующие способы:

1) Аппаратный способ моделирования гексагонального растра показан на рис. 4.7. Этот способ реализуется путем задержки каждой нечетной строки изображения на 1 пиксель.

При этом необходимо растянуть изображение на экране таким образом, чтобы было выполнено условие (1).

2) Программный способ построения гексагонального раstra в квадратном. Этот способ осуществляется на основе модели показанной на рис 4.8 исключительно программными средствами.

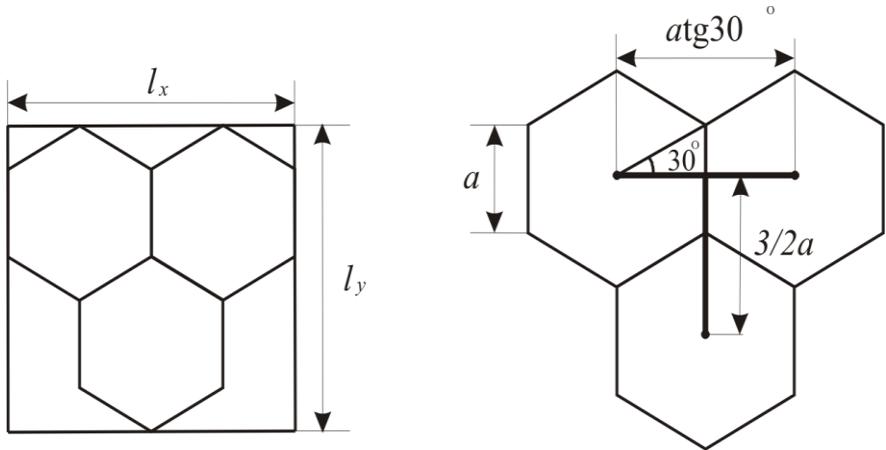


Рис. 4.6. Определение пропорций элементов прямоугольного раstra для моделирования гексагонального изображения

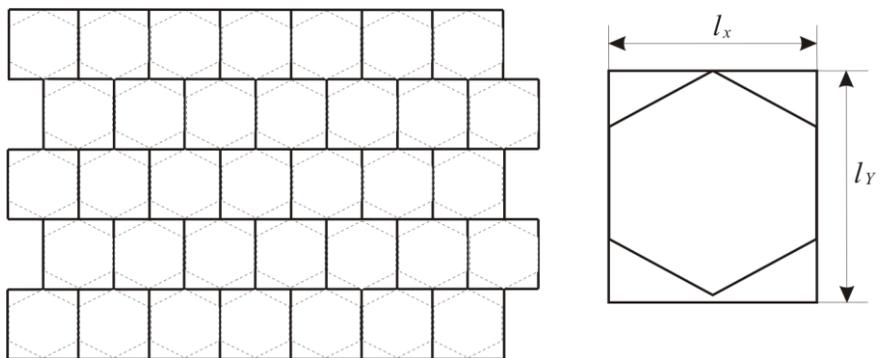


Рис. 4.7. Построение гексагонального раstra на прямоугольном аппаратном методе

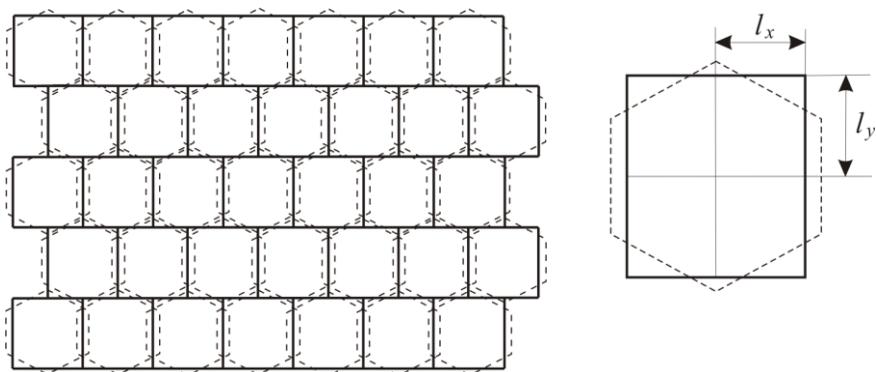


Рис. 4.8. Построение гексагонального растра в квадратном программными средствами

Тот факт, что гексагональный растр не используется на практике, объясняется следующими причинами:

- некоторое усложнение алгоритмов для гексагонального растра по сравнению с квадратным;
- преимущества гексагонального растра не слишком велики;
- историческая ориентация на прямоугольный (квадратный) растр.

4.5. Построение линии в квадратном растре

Так как экран растрового дисплея с электронно-лучевой трубкой представляет собой матрицу дискретных элементов (пикселей), каждый из которых может быть подсвечен, то, разумеется, невозможно непосредственно провести отрезок из одной произвольной точки в другую. За исключением тех случаев, когда все точки отрезка совпадают с положением пикселей. Процесс определения пикселей, наилучшим образом аппроксимирующих заданный отрезок, называется разложением в растр. В сочетании с процессом построчной визуализации изображения этот процесс известен как преобразование растровой развертки. Для горизонтальных вертикальных и наклоненных под углом 45° отрезков выбор растровых элементов

очевиден. При любой другой ориентации выбрать нужные пиксели труднее. Процесс выбора пикселей наилучшим образом аппроксимирующих отрезок показан на рис. 4.9.

Для построения линии в квадратном растре необходимо соблюдать общие требования к изображению отрезка:

- концы отрезка должны находиться в заданных точках;
- отрезки должны выглядеть прямыми;
- яркость вдоль отрезка должна быть постоянной и не зависеть от его длины и наклона.

Очевидно, что ни одно из этих требований не может быть выполнено в полном объеме на растровом дисплее в силу его дискретности. Изображение строится из пикселей конечных размеров поэтому:

- концы отрезка в общем случае располагаются на пикселях, лишь наиболее близких к требуемым позициям и только в

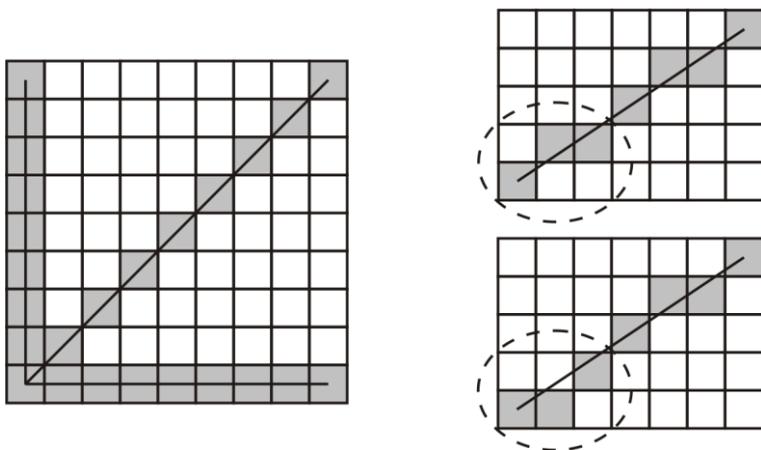


Рис. 4.9. Примеры разложения отрезков в растр

частных случаях координаты концов отрезков могут точно совпадать с координатами пикселей;

- отрезок аппроксимируется набором пикселей и лишь в частных случаях вертикальных, горизонтальных и отрезков проведенных под углом 45° они будут выглядеть прямыми,

причем гладкими прямыми, без ступенек, только для горизонтальных и вертикальных отрезков (рис. 4.9);

- яркость для различных отрезков и даже вдоль отрезка в общем случае различна, так как, например, расстояние между центрами пикселей для вертикального отрезка и отрезка проведенного под углом в 45° различно (рис.4.9);

- для произвольных отрезков наряду со ступенчатостью и переменной яркостью возникает проблема выбора пикселей, наилучшим образом аппроксимирующим отрезок (рис.4.9, выделенные области).

Объективное улучшение аппроксимации достигается увеличением разрешения дисплея, но в силу существенных технологических проблем разрешение для растровых систем не может быть бесконечно велико. Обычно разрешение составляет величину порядка 1280×1024 . Субъективное улучшение аппроксимации основано на психофизиологических особенностях зрения и, в частности, может достигаться простым уменьшением размеров экрана. Другие способы субъективного улучшения качества аппроксимации основаны на различных программных ухищрениях по «размыванию» резких границ изображения.

4.6. Параметрический алгоритм рисования линии

Пусть необходимо провести линию из точки (x_1, y_1) в точку (x_2, y_2) с линейной интерполяцией по яркости (рис. 4.10). Любую точку этой линии можно представить в следующем виде:

$$\begin{aligned}x_{i+1} &=]x_i + t(x_2 - x_1)[\\y_{i+1} &=]y_i + t(x_2 - x_1)[\\V_{i+1} &=]V_i + t(x_2 - x_1)[\end{aligned} \quad (4.2)$$

Где $t \in [0, 1]$, $] [$ – знак округления до целого.

$$t = \frac{1}{\max\{|x_2 - x_1|, |y_2 - y_1|\}} = \frac{1}{N - 1}; \quad (4.3)$$

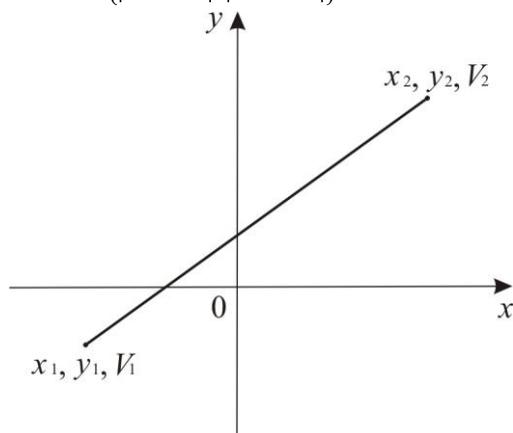


Рис. 4.10. Линия, проведенная из точки (x_1, y_1) с интенсивностью V_1 в точку (x_2, y_2) с интенсивностью V_2 .

$|x_2 - x_1|, |y_2 - y_1|$ – длина линии в пикселях по оси x и по оси y соответственно, N – длина линии в пикселях.

Аналогичные вычисления можно провести и через приращения координат.

$$\begin{aligned} \Delta x &= \frac{x_2 - x_1}{N - 1} \\ \Delta y &= \frac{y_2 - y_1}{N - 1} \\ \Delta V &= \frac{V_2 - V_1}{N - 1} \end{aligned} \quad (4.4)$$

Значения приращений считаются в начале функции и не входят в цикл построения линии на экране дисплея, за счет чего повышается быстродействие алгоритма.

К недостаткам данного алгоритма следует отнести:
- необходимость работы с вещественными числами;

- в алгоритме имеется операция деления, что значительно усложняет аппаратную организацию и увеличивает время работы алгоритма.

К достоинствам данного алгоритма можно отнести:

- простоту программной реализации;
- простоту реализации линейной интерполяции по яркости.

4.7. Алгоритм Брезенхема рисования линии

В 1965 году Брезенхемом был предложен простой целочисленный алгоритм для растрового построения отрезка. В алгоритме используется управляющая переменная d_i , которая на каждом шаге пропорциональна разности между s и t – расстояниям от прямой до ближайших пикселей S и T . На рис. 4.11. показан i -ый шаг, когда пиксель P_{i-1} уже найден как ближайший к реальному изображаемому отрезку, и теперь требуется определить, какой из пикселей должен быть установлен следующим T_i или S_i .

Если $s < t$, то S_i ближе к отрезку и необходимо выбрать его. В противном случае ближе к отрезку будет пиксель T_i и выбирается он. Другими словами, если $s - t < 0$, то выбирается S_i , если $s - t > 0$, то выбирается T_i .

Изображаемый отрезок проводится из точки (x_1, y_1) в точку (x_2, y_2) . Пусть первая точка находится ближе к началу координат, тогда перенесем обе точки отрезка начальную и конечную так, чтобы начальная точка отрезка находилась в начале координат, при этом конечная окажется в положении $(\Delta x, \Delta y)$ относительно начальной, где $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$. Уравнение прямой теперь имеет вид:

$$y = x \frac{\Delta y}{\Delta x}. \quad (4.5)$$

Из рисунка следует, что

$$s = \frac{\Delta y}{\Delta x}(r+1) - q; \quad (4.6)$$

$$t = q + 1 - \frac{\Delta y}{\Delta x}(r+1);$$

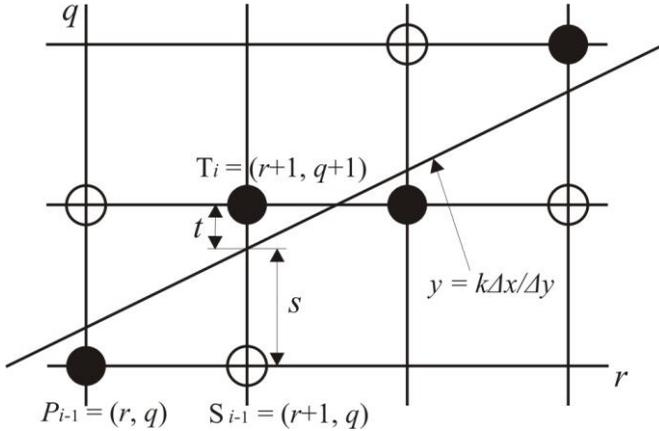


Рис. 4.11. Схема выбора пикселей в алгоритме Брезенхе-ма

поэтому

$$s - t = 2 \frac{\Delta y}{\Delta x}(r+1) - 2q - 1. \quad (4.7)$$

Помножим обе части уравнения на Δx :

$$\Delta x(s - t) = 2(r\Delta y - q\Delta x) + 2\Delta y - \Delta x. \quad (4.8)$$

Так как $\Delta x > 0$, то величину $\Delta x(s - t)$ можно использовать в качестве критерия для выбора пикселя.

Обозначим $d_i = \Delta x(s - t)$. Учитывая, что $r = x_{i-1}$, $q = y_{i-1}$ запишем (4.8) в виде:

$$d_i = 2(x_{i-1}\Delta y - y_{i-1}\Delta x) + 2\Delta y - \Delta x. \quad (4.9)$$

Для последующего шага запишем:

$$\Delta_i = d_{i+1} - d_i = 2\Delta y(x_i - x_{i-1}) - 2\Delta x(y_i - y_{i-1}). \quad (4.10)$$

Тогда, для критерия d_i можно записать:

1). Если $d_i \geq 0$, то выбираем T_i , при этом необходимо учесть, что $x_i - x_{i-1} = 1$ и $y_i - y_{i-1} = 1$ тогда

$$\Delta_i = 2(\Delta y - \Delta x); \quad (4.11)$$

2). Если $d_i < 0$, то выбираем S_i , при этом необходимо учесть, что $x_i - x_{i-1} = 0$ и $y_i - y_{i-1} = 1$ тогда

$$\Delta_i = 2\Delta y. \quad (4.12)$$

Таким образом, получена интерактивная формула для вычисления критерия d_i . Начальное значение, которого определится как:

$$d_1 = 2\Delta y - \Delta x. \quad (4.13)$$

Блок схема алгоритма Брезенхема рисования линии показана на рис. 4.12.

Линейная интерполяция по яркости при построении отрезка по алгоритму Брезенхема получается параметрическим способом:

$$\begin{aligned} V &= V_{i-1} + \Delta V; \\ \Delta V &= \frac{V_2 - V_1}{N - 1}, \end{aligned} \quad (4.14)$$

где N – длина отрезка в пикселях.

4.8. Алгоритм Брезенхема генерации окружности

Существуют различные алгоритмы рисования окружности, однако Брезенхем разработал наиболее эффективный алгоритм генерации окружности. Здесь используется тот же принцип, что и рассмотренный ранее для рисования линии.

Рассмотрим сегмент окружности с центром в начале координат. На каждом шаге выбираем точку, ближайшую к реальной окружности. В качестве ошибки возьмем величину

$$D(P_i) = (x_i^2 + y_i^2) - R^2. \quad (4.15)$$

На рис. 2.14, показаны различные возможные способы прохождения истинной окружности через сетку пикселей. Пусть пиксель P_{i-1} уже найден как ближайший к реальной окружности, и теперь требуется определить, какой из пикселей

должен быть установлен следующим: T_i или S_i . Для этого определим точку, которая соответствует минимальной ошибке:

$$D(S_i) = ((q+1)^2 + p^2) - R^2,$$

$$D(T_i) = ((q+1)^2 + (p^2 + 1)) - R^2.$$
(4.16)

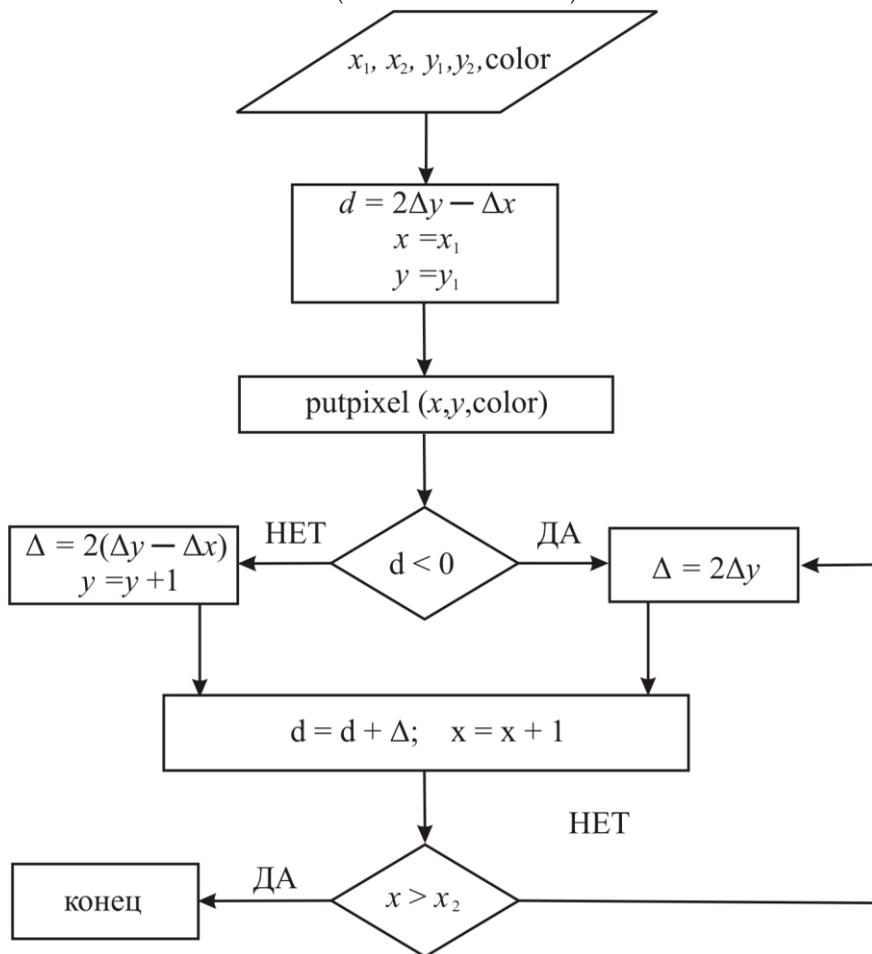


Рис. 4.12. Блок схема алгоритма Брезенхема рисования линии

Если

$$|D(S_i)| < |D(T_i)|, \quad (4.17)$$

то выбираем S_i , в противном случае – T_i . Введем критерий

$$d_i = |D(S_i)| - |D(T_i)|, \quad (4.18)$$

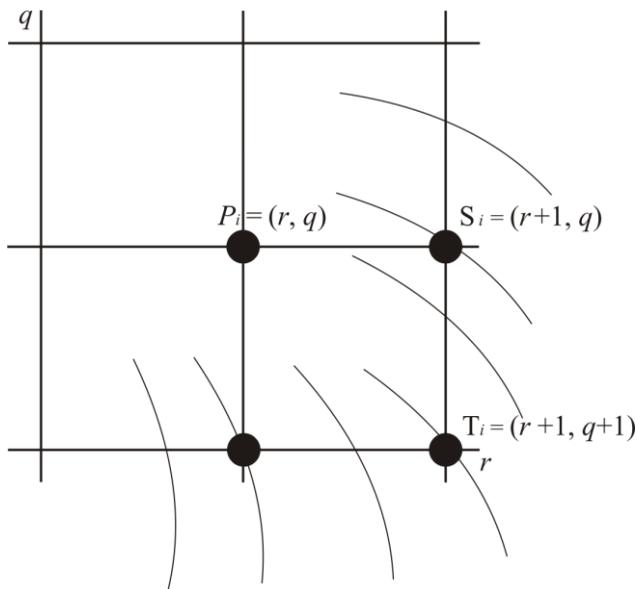


Рис. 4.13. Возможные способы прохождения окружности через сетку пикселей

тогда S_i выбирается при $d_i < 0$, если $d_i > 0$, то выбирается – T_i . Если рассматривается только часть окружности, на пример, дуга окружности от 0° до 45° , то $D(S_i) > 0$, так как точка S_i лежит за пределами окружности, а $D(T_i) < 0$, так как T_i находится внутри окружности, поэтому

$$d_i = D(S_i) + D(T_i). \quad (4.19)$$

Алгебраические вычисления, аналогичны тем, которые проводились для линии. Эти вычисления приводят к результату:

$$d_i = 3 - 2R. \quad (4.20)$$

Если выбираем S_i (когда $d_i < 0$), то

$$\Delta_i = 4x_{i-1} + 6; \quad (4.21)$$

если выбираем T_i (когда $d_i \geq 0$), то

$$\Delta_i = 4(x_{i-1} \cdot y_{i-1}) + 10. \quad (4.22)$$

Блок схема этого алгоритма изображена на рис. 4.14.

Данный алгоритм хорош тем, что в нем отсутствуют операции с плавающей точкой, а также операции деления и извлечения корня.

4.9. Отсечение по полю вывода

На рис. 1.16 показана плоская сцена и отсекающее окно регулярной формы. Окно задается левым (Л), правым (П), верхним (В), и нижним (Н) ребрами. Регулярным отсекающим окном является прямоугольник, стороны которого параллельны осям координат объектного пространства или осям координат экрана. Целью алгоритма отсечения является определение тех точек, отрезков или частей отрезков, которые лежат внутри отсекающего окна. Эти точки, отрезки или их части остаются для визуализации внутри окна. Все, что не попадает в отсекающее окно, отбрасывается.

Так как в обычных сценах или картинках необходимо отсекал большое количество отрезков или точек, эффективность алгоритмов отсечения представляет большой интерес. Очень часто подавляющее большинство отрезков или точек лежит целиком внутри или вне отсекающего окна. Соответственно очень важно уметь быстро отбирать отрезки подобные ab , или точки подобные p , и отбрасывать отрезки, подобные ij , или точки, подобные q показанные на рис. 2.16.

Точки, лежащие внутри отсекающего окна, удовлетворяют условию:

$$\begin{aligned} x_{\hat{E}} &\leq x \leq x_{\hat{I}}; \\ y_{\hat{I}} &\leq y \leq y_{\hat{A}}. \end{aligned} \quad (4.23)$$

Знак равенства указывает на то, что точки, лежащие на границе окна, считаются находящимися внутри него. Отрезок лежит внутри окна и, следовательно, является видимым, если обе его концевые точки лежат внутри окна, например отрезок ab на рис. 2.16. Однако если оба конца отрезка лежат вне окна, то отрезок необязательно лежит целиком вне окна, например от-

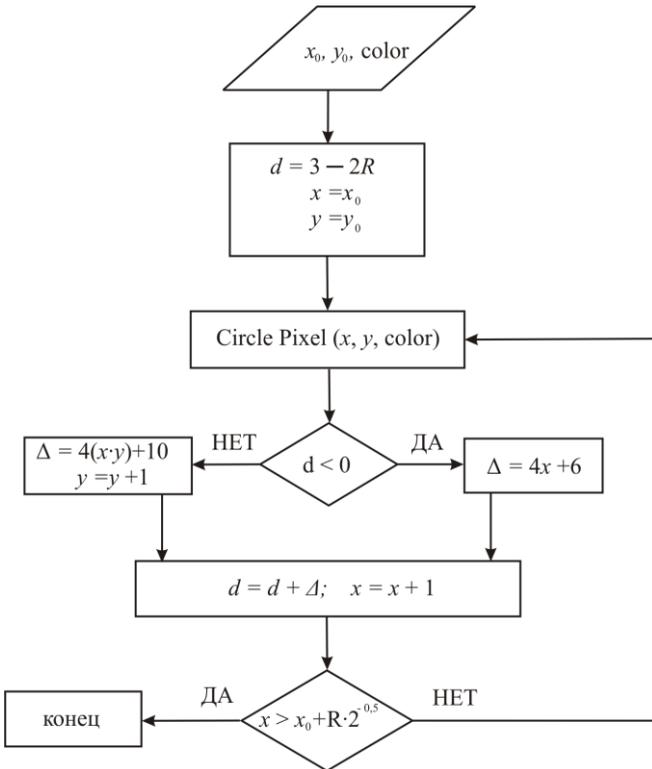


Рис. 4.14. Блок схема алгоритма Брезенхема генерации окружности

резок gh на рис. 2.16. Если же оба конца отрезка лежат справа, слева, выше или ниже окна, то этот отрезок целиком лежит вне окна, а значит невидим. Проверка последнего условия полностью устранил все отрезки, помеченные ij на рис. 4.15. Но данная проверка не устранил, ни отрезка gh , который видим частично, ни отрезка kl , который целиком невидим.

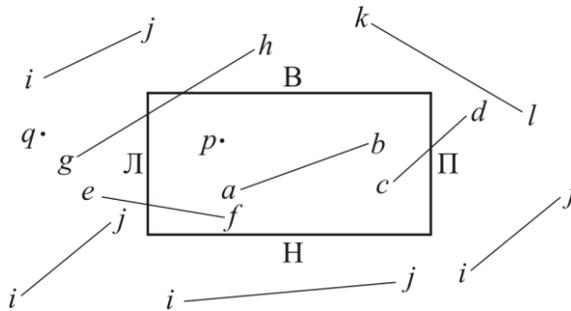


Рис. 4.15. Отсечение по полю вывода

4.9.1. Алгоритм Коэна-Сазерленда

Пусть a и b – концы отрезка, тогда можно предложить алгоритм, определяющий все полностью видимые и большинство невидимых отрезков. Это алгоритм Коэна-Сазерленда.

Для решения задачи определения принадлежности отрезка полю вывода используется следующий метод. Пространство разбивается на 9 областей, каждая из областей кодируется бинарным четырех битным кодом рис. 4.16. Для каждого отрезка определяются коды его концов (K_1 , K_2) затем проводится экспресс анализ полученных результатов.

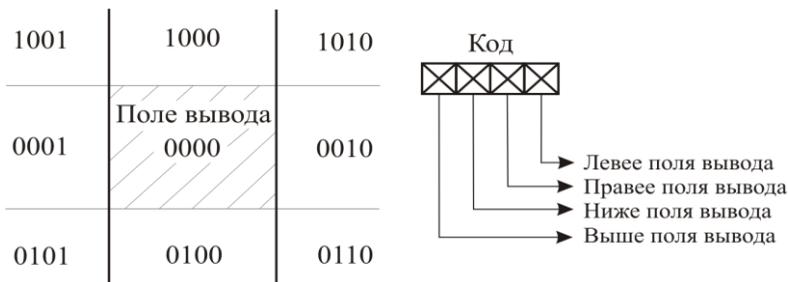


Рис. 4.16. Кодирование пространства бинарным четырех битным кодом

Если $K_1(x_1, y_1, z_1, u_1)$ почленно умножить на $K_2(x_2, y_2, z_2, u_2)$, (где x, y, z, u – двоичный бинарный код) и это произведение не равно нулю, то отрезок целиком лежит вне поля вывода и отбрасывается целиком.

Если $K_1(x_1, y_1, z_1, u_1) = K_2(x_2, y_2, z_2, u_2) = 0$, то отрезок полностью лежит в поле вывода и отсечение не нужно.

Если $K_1(x_1, y_1, z_1, u_1)$ почленно умножить на $K_2(x_2, y_2, z_2, u_2)$, и это произведение равно нулю, то отрезок может частично лежать внутри поля вывода и необходимо произвести отсечение по полю вывода.

4.9.2. Алгоритм отсечения по прямоугольной области

Если почленное произведение $K_1(x_1, y_1, z_1, u_1)$ на $K_2(x_2, y_2, z_2, u_2)$ равно нулю, то необходимо провести отсечение по полю вывода. Отсечение отрезка по границам поля вывода проводится последовательно по всем сторонам ограничивающим пространство поля вывода как это показано на рис. 4.17.

На рис. 4.17 жирной линией выделена граница, по которой происходит отсечение отрезка в данный момент: а – отсечение по правой стороне; б – отсечение по верхней стороне; в – отсечение по левой стороне; г – отсечение по нижней стороне; д – отрезок после отсечения. Необходимо отметить, что точки отрезка, лежащие на границе поля вывода, принадлежат полю вывода.

На каждом шаге процесса отсечения по полю вывода вычисляются новые координаты точек пресечения прямой с границей поля вывода.

Найдем формулы для вычисления новых координат начала и конца отрезка после отсечения по полю вывода с исполь-

зованием линейной интерполяцией по яркости. Формулы для вычисления новых координат записываются следующим образом:

а

б

$$q(x_0, y_0, V_q)$$

$$q(x_q, y_0, V_q)$$

$$y_q = y_s + \frac{x_0 - x_s}{x_p - x_s} (y_p - y_s)$$

$$x_q = x_s + \frac{y_0 - y_s}{y_p - y_s} (x_p - x_s)$$

$$V_q = V_s + \frac{x_0 - x_s}{x_p - x_s} (V_p - V_s)$$

$$V_q = V_s + \frac{y_0 - y_s}{y_p - y_s} (V_p - V_s)$$

Вывод формул для вычисления новых координат отрезка после отсечения по границам поля вывода проиллюстрирован на рис. 4.18. Случай а – показан на рис. 4.18 а; случай б – показан на рис. 4.18 б.

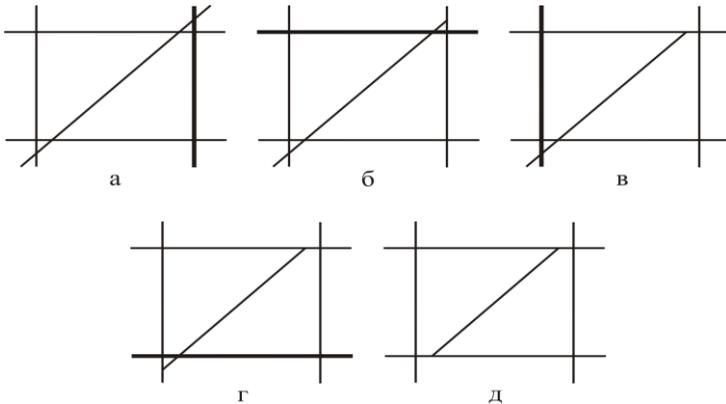


Рис. 4.17. Последовательность отсечения отрезка по границам поля вывода

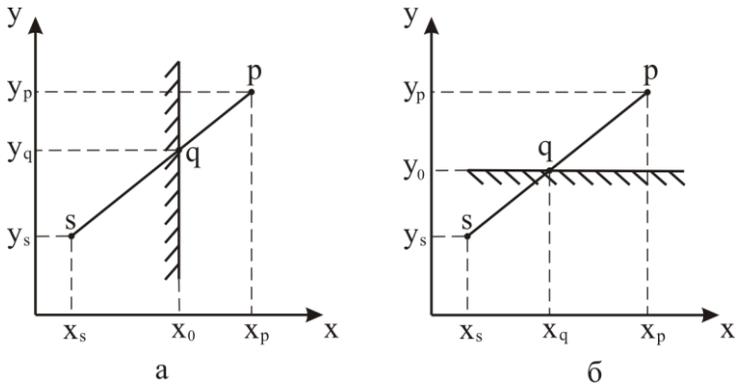


Рис. 4.18. К выводу формул для расчета координат отрезка после отсечения

5. СИСТЕМА КООРДИНАТ И ТИПЫ ПРЕОБРАЗОВАНИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

5.1. Декартова система координат

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

Для того чтобы ввести декартову систему координат, проведем в трехмерном пространстве три направленные прямые линии, которые называются осями. Причем они не должны лежать в одной плоскости и должны пересекаться в одной точке – начале координат. Выберем единицы измерения по осям. Тогда положение любой точки в пространстве будет описываться через ее координаты, которые представляют собой расстояния от начала координат до проекции точки на соответствующие оси координат. Проекцией точки на координатную ось называется точка пересечения плоскости, проходящей через заданную точку, и параллельной плоскости образованной, двумя другими осями координат. Например, на рис 2.1 проекцией точки P на ось Ox является точка Q , которая принадлежит плоскости, параллельной плоскости zOy .

В общем случае оси системы координат могут располагаться под произвольными углами относительно друг друга. Для практического применения гораздо удобнее, когда эти оси расположены взаимно перпендикулярно друг к другу. Такая система координат называется ортогональной.

Таким образом, положение в пространстве точки P описывается ее координатами, что записывается как $P(x,y,z)$. Взаимное расположение осей в ортогональной системе координат

в трехмерном пространстве может быть двух видов. Проведем ось Ox слева на право, а ось Oz снизу вверх, как показано на рис.2.2. Ось Oy при этом может проходить как в направлении от наблюдателя перпендикулярно плоскости листа, так и от плоскости листа к наблюдателю. В первом случае система координат называется левой или левосторонней, а во втором случае – правой или правосторонней.

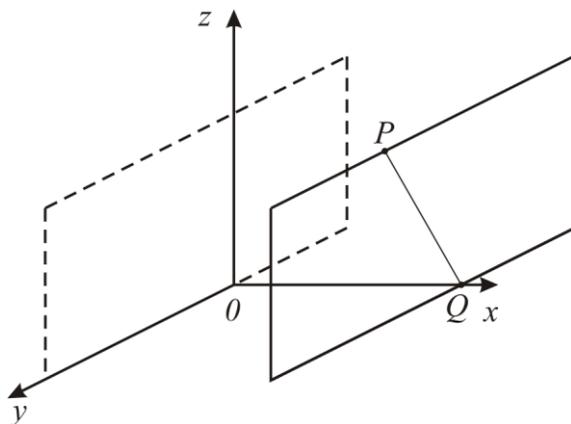


Рис. 5.1. Определение координаты x пространственной точки P

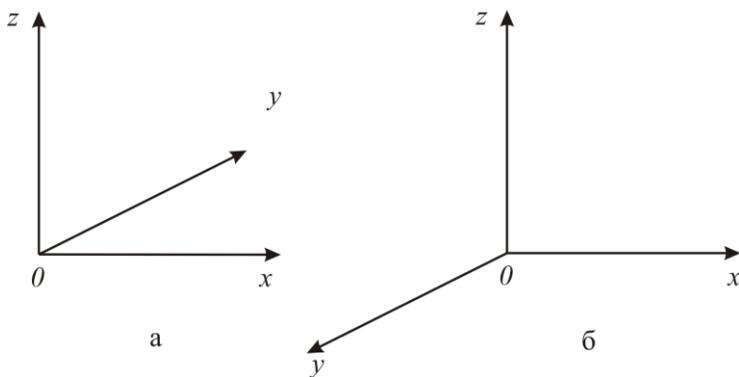


Рис. 5.2. Левосторонняя (а) и правосторонняя (б) системы координат

Декартовы координаты точек позволяют описывать статичное положение объектов в пространстве. Однако для проведения каких-либо действий над объектами необходимо иметь дополнительные математические конструкции. В качестве одной из таких конструкций используются радиус-векторы. Они обладают всеми свойствами векторов, но имеют одну особенность. Начало радиус-вектора всегда находится в начале координат, а конец радиус-вектора лежит в некоторой точке пространства. Это свойство радиус векторов позволяет поставить во взаимное однозначное соответствие всем точкам пространства соответствующие им радиус-векторы.

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

Известно, что две различные точки в пространстве определяют прямую, как кратчайшее расстояние между ними. Выберем в пространстве две точки $P_1(x_1, y_1, z_1)$ и $P_2(x_2, y_2, z_2)$ и соответствующие им радиус векторы \vec{p}_1, \vec{p}_2 и проведем прямую через точки P_1 и P_2 как показано на рис. 5.3.

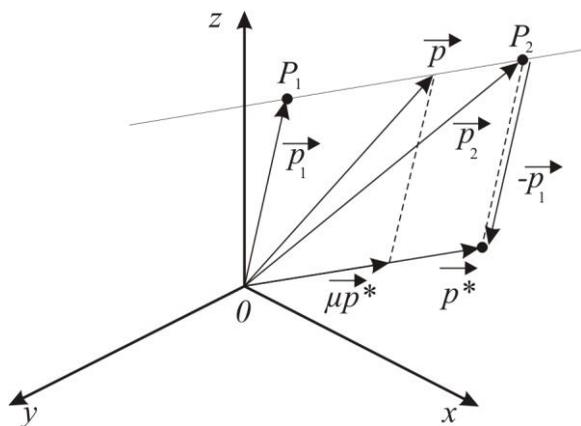


Рис. 5.3. К выводу уравнения прямой в трехмерном пространстве

Проведем от точки P_1 к точке P_2 вектор $\overrightarrow{p^*} = \overrightarrow{p_2} - \overrightarrow{p_1}$. Тогда радиус-вектор \overrightarrow{p} , определяющий некоторую точку на прямой, можно получить сложением например вектора $\overrightarrow{p_1}$ и вектора $\overrightarrow{p^*}$, умноженного на некоторое число μ . Или $\overrightarrow{p} = \overrightarrow{p_1} + \mu\overrightarrow{p^*}$. Фактически уравнение прямой уже получено, но не через координаты двух точек на прямой, а с помощью базового радиус-вектора $\overrightarrow{p_1}$ и направляющего радиус-вектора $\overrightarrow{p^*}$. Преобразуем это уравнение к виду в котором используются только координаты двух исходных векторов $\overrightarrow{p_1}$ и $\overrightarrow{p_2}$:

$$\overrightarrow{p} = \overrightarrow{p_1} + \mu\overrightarrow{p^*} = \overrightarrow{p_1} + \mu(\overrightarrow{p_2} - \overrightarrow{p_1}) \Rightarrow \overrightarrow{p} - \overrightarrow{p_1} = \mu(\overrightarrow{p_2} - \overrightarrow{p_1}). \quad (5.1)$$

Из последнего векторного равенства получаем три скалярных равенства для соответствующих координат:

$$\begin{aligned} x - x_1 &= \mu(x_2 - x_1), \\ y - y_1 &= \mu(y_2 - y_1), \\ z - z_1 &= \mu(z_2 - z_1). \end{aligned} \quad (5.2)$$

Попарно разделим эти три уравнения друг на друга, для того чтобы избавиться от коэффициента μ , в результате получим систему уравнений, определяющую прямую в трехмерном пространстве:

$$\begin{aligned} (x - x_1)(y_2 - y_1) &= (x_2 - x_1)(y - y_1), \\ (y - y_1)(z_2 - z_1) &= (y_2 - y_1)(z - z_1), \\ (z - z_1)(x_2 - x_1) &= (z_2 - z_1)(x - x_1). \end{aligned} \quad (5.3)$$

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

5.2. Двумерные матричные преобразования

Рассмотрим преобразование координат точек на плоскости. Перенесем точку A в точку B как показано на рис. 5.4.

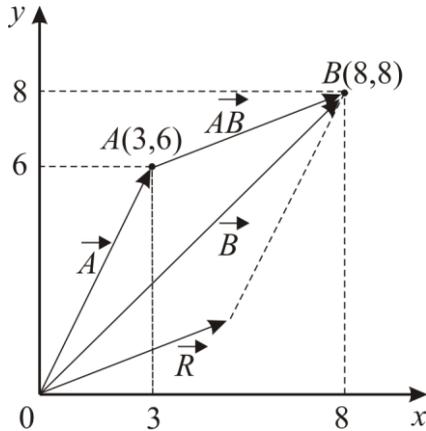


Рис. 5.4. Перенос или трансляция точки A в точку B

С точки зрения математики этот перенос или трансляцию можно описать с помощью вектора переноса \vec{AA} . Пусть \vec{R} радиус-вектор, соответствующий вектору переноса \vec{AA} . Тогда переход из точки A в точку B будет соответствовать векторной записи:

$$\vec{A} = \vec{A} + \vec{R}. \quad (5.4)$$

Откуда следует, что для переноса точки в новое положение следует добавить к ее координатам некоторые числа, которые представляют собой координаты вектора переноса:

$$\vec{B} = \vec{A} + \vec{R} = [A_x + R_x, A_y + R_y, A_z + R_z]. \quad (5.5)$$

Масштабированием объектов называется растяжение объектов вдоль соответствующих координатных осей относительно начала координат. Эта операция применяется к каждой точке объекта, поэтому можно также говорить о масштабировании точки. Однако при этом, конечно, речь не идет об изменении размеров самой точки. Масштабирование достигается умножением координат точек на некоторые константы – ко-

эффиценты масштабирования. В том случае, когда коэффициенты масштабирования равны между собой, масштабирование называется однородным. На рис 2.5 приведен пример однородного масштабирования треугольника ABC .

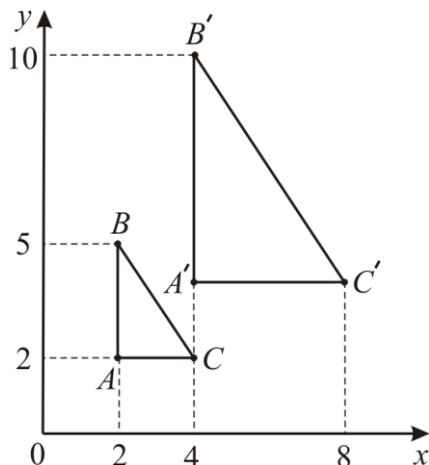


Рис. 5.5. Операция масштабирования с коэффициентом масштабирования $S=2$

После применения операции однородного масштабирования с коэффициентом $S=2$ треугольник ABC переходит в подобный треугольник $A'B'C'$. Обозначим матрицу масштабирования:

$$S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}. \quad (5.6)$$

Где S_x – коэффициент масштабирования по оси x , S_y - коэффициент масштабирования по оси y .

Для точек A и A' операция масштабирования в матричном виде запишется следующим образом:

$$\begin{aligned} [x', y'] &= [x, y] \times \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} = \\ &= [(x \cdot S_x + y \cdot 0), (x \cdot 0 + y \cdot S_y)] = [x \cdot S_x, y \cdot S_y] \end{aligned} \quad (5.7)$$

Рассмотрим далее операцию вращения точки на некоторый угол α относительно начала координат. Как показано на рис. 5.6 точка $A(x,y)$ в результате поворота на угол α переходит в положение $B(x',y')$.

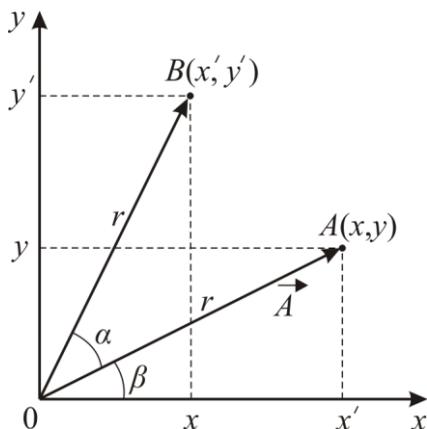


Рис. 5.6. Иллюстрация поворота точки A на угол α

Найдем преобразование точки A в положение B . Обозначим через β угол, который составляет радиус вектор \vec{A} с осью Ox . Пусть r – длина радиус-вектора \vec{A} , тогда:

$$\begin{aligned} x' &= r \cdot \cos(\alpha + \beta) = r(\cos \alpha \cdot \cos \beta - \sin \alpha \cdot \sin \beta), \\ y' &= r \cdot \sin(\alpha + \beta) = r(\sin \alpha \cdot \cos \beta + \cos \alpha \cdot \sin \beta). \end{aligned} \quad (5.8)$$

Так как $\cos \beta = \frac{x}{r}$, а $\sin \beta = \frac{y}{r}$, то подставляя эти выражения в уравнение (3.8) для x' и y' , получим:

$$\begin{aligned} x' &= x \cdot \cos \alpha - y \cdot \sin \alpha, \\ y' &= x \cdot \sin \alpha + y \cdot \cos \alpha. \end{aligned} \quad (5.9)$$

В матричном виде операция вращения точки A на угол α в положение B , как показано на рисунке 2.6, записывается следующим образом:

$$\begin{aligned}
 [x', y'] &= [x, y] \times \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} = \\
 &= \left[\underbrace{(x \cdot \cos \alpha - y \cdot \sin \alpha)}_{x'}, \underbrace{(x \cdot \sin \alpha + y \cdot \cos \alpha)}_{y'} \right].
 \end{aligned}
 \tag{5.10}$$

Последнее выражение в квадратных скобках представляют собой координаты точки B после поворота точки A на угол α , как следует из выражения (5.9).

5.3. Однородные координаты и матричное представление двумерных преобразований

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

Очевидно, что удобнее применять результирующую матрицу последовательных преобразований вместо того, чтобы каждый раз заново вычислять произведение матриц. В компьютерной графике матричное произведение называется композицией. Однако таким способом нельзя получить результирующую матрицу преобразования, если среди последовательности преобразований присутствует, хотя бы одна операция переноса.

Удобно иметь математический аппарат, позволяющий включать в композиции преобразований все три рассмотренные операции. При этом получается значительный выигрыш в скорости вычислений. В качестве такого математического аппарата используются однородные координаты.

Так двумерный вектор (x, y) в однородных координатах записывается в виде $(\omega x, \omega y, \omega)$, где $\omega \neq 0$. Число ω называется масштабным множителем. Для того чтобы из вектора, записанного в однородных координатах, получить вектор в обычных координатах, необходимо разделить три однородных координаты на третью:

$$\left(\frac{\omega x}{\omega}, \frac{\omega y}{\omega}, \frac{\omega}{\omega} \right) \rightarrow (x, y, 1). \quad (5.11)$$

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

В однородных координатах преобразование центральной перспективы определяется матричной операцией, которая записывается в виде:

$$D = \begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & k \end{bmatrix} \quad (5.12)$$

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

Пусть $p = (x, y, z)$ – точка в трехмерном пространстве. Ее однородное представление $v = (\omega x, \omega y, \omega z, \omega)$. Умножим v на P :

$$\begin{aligned} v \cdot p &= (\omega x, \omega y, \omega z, \omega) \times \begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & k \end{bmatrix} = \\ &= [\omega k x, \omega k y, 0, \omega(z+k)]. \end{aligned} \quad (5.13)$$

Последнее почленно поделим на $\omega(z+k)$, тогда:

$$v \cdot p = \left[\frac{kx}{(z+k)}, \frac{ky}{z+k}, 0, 1 \right]. \quad (5.14)$$

Или в трехмерном пространстве координаты точки перспективной проекции p' запишутся:

$$p' = \left[\frac{kx}{(z+k)}, \frac{ky}{(z+k)}, 0 \right]. \quad (5.15)$$

Полученное выражение определяет преобразование точки объекта, в точку перспективной проекции.

Использование однородных координат позволяет описывать точки двумерного пространства трехэлементными вектор-строками, соответственно и матрицы преобразований, на которые будут умножаться вектор-строки точек, также будут иметь размеры 3×3 .

Запишем матричное преобразование операции переноса точки A в точку B в двумерном пространстве в однородных координатах. Операция переноса показана на рис. 5.7.

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix} = [(x + D_x), (y + D_y), 1]. \quad (5.16)$$

Нетрудно видеть, что координаты точки после операции переноса определятся следующим образом:

$$\begin{aligned} x' &= x + D_x, \\ y' &= y + D_y. \end{aligned} \quad (5.17)$$

При этом компоненты вектора переноса запишутся:

$$\vec{R} = [D_x, D_y]. \quad (5.18)$$

Обозначим:

$$\begin{aligned} p &= [x, y, 1], \\ p' &= [x', y', 1]. \end{aligned} \quad (5.19)$$

Тогда можно записать:

$$p' = pT(D_x, D_y). \quad (5.20)$$

Где

$$T(D_x, D_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}. \quad (5.21)$$

Матрица двумерного переноса в однородных координатах.

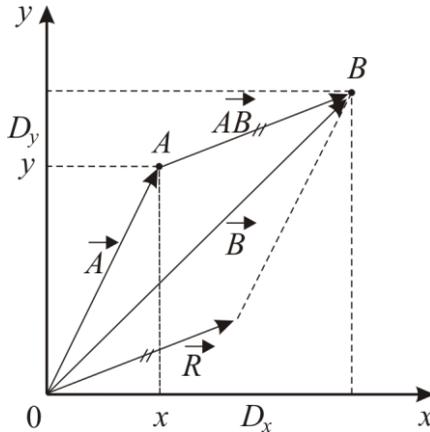


Рис. 5.7. Результат матричной операции переноса в двумерном пространстве в однородных координатах

При последовательном переносе точки p в точку p' и затем в точку p'' компоненты суммарного вектора переноса являются суммами соответствующих компонент последовательных векторов переноса. Пусть $p' = pT(D_x, D_y)$, $p'' = p'T(D'_x, D'_y)$. Тогда подставив первое уравнение во второе получим:

$$p'' = pT(D_x, D_y)p'T(D'_x, D'_y). \quad (5.22)$$

Таким образом, матричное произведение, то есть суммарный перенос, равен произведению соответствующих матриц переноса:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D'_x & D'_y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x + D'_x & D_y + D'_y & 1 \end{bmatrix}. \quad (5.23)$$

Запишем матричное преобразование операции масштабирования точки A в точку A' в двумерном пространстве в однородных координатах. Операция масштабирования показана на рис. 5.5.

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = [x \cdot S_x, y \cdot S_y, 1]. \quad (5.24)$$

Где S_x, S_y – коэффициенты масштабирования по соответствующим координатным осям.

Нетрудно видеть, что координаты точки после операции масштабирования определяются следующим образом:

$$\begin{aligned} x' &= xS_x, \\ y' &= yS_y. \end{aligned} \quad (5.25)$$

Обозначим:

$$\begin{aligned} p &= [x, y, 1], \\ p' &= [x', y', 1]. \end{aligned} \quad (5.26)$$

Тогда можно записать:

$$p' = pS(S_x, S_y). \quad (5.27)$$

Где

$$S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (5.28)$$

матрица двумерного масштабирования в однородных координатах.

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

Пусть происходит последовательное масштабирование точки p в точку p' и затем в точку p'' . Тогда $p' = pS(S_x, S_y)$, $p'' = p'S(S'_x, S'_y)$. Подставив первое уравнение во второе, получим:

$$p'' = pS(S_x, S_y)p'S(S'_x, S'_y). \quad (5.29)$$

Таким образом, матричное произведение, то есть результирующее масштабирование, равно произведению соответствующих матриц масштабирования, причем результирующее масштабирование по соответствующим осям определяется как произведение соответствующих коэффициентов масштабирования:

$$\begin{aligned} S(S_x, S_y) \times (S'_x, S'_y) &= \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} S'_x & 0 & 0 \\ 0 & S'_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} S_x \cdot S'_x & 0 & 0 \\ 0 & S_y \cdot S'_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (5.30)$$

Запишем матричное преобразование операции поворота относительно начала координат точки A в точку B в двумерном пространстве в однородных координатах. Операция поворота показана на рис. 5.6.

$$\begin{aligned} [x', y', 1] &= [x, y, 1] \times \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &= [(x \cdot \cos \alpha - y \cdot \sin \alpha), (x \cdot \sin \alpha + y \cdot \cos \alpha), 1]. \end{aligned} \quad (5.31)$$

Где α – угол поворота.

Нетрудно видеть, что координаты точки после операции поворота определяются следующим образом:

$$\begin{aligned}x' &= (x \cdot \cos \alpha - y \cdot \sin \alpha), \\y' &= (x \cdot \sin \alpha + y \cdot \cos \alpha).\end{aligned}\tag{5.32}$$

Обозначим:

$$\begin{aligned}p &= [x, y, 1], \\p' &= [x', y', 1].\end{aligned}\tag{5.33}$$

Тогда можно записать:

$$p' = pR(\alpha).\tag{5.34}$$

Где

$$R(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix},\tag{5.35}$$

матрица двумерного поворота в однородных координатах.

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

Пусть происходит последовательный поворот точки p в точку p' на угол α и затем в точку p'' на угол β . Тогда $p' = pR(\alpha)$, $p'' = p'R(\beta)$. Подставив первое уравнение во второе, получим:

$$p'' = pR(\alpha)p'R(\beta).\tag{5.36}$$

Таким образом, результирующий поворот, равен произведению соответствующих матриц поворота, причем он определяется как суммарный поворот на угол $(\alpha + \beta)$:

$$\begin{aligned}R(\alpha)R(\beta) &= \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & 0 \\ -\sin(\alpha + \beta) & \cos(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.\end{aligned}\tag{5.37}$$

То есть любое количество последовательных поворотов можно записать в виде матрицы суммарного поворота. Кроме того, любая последовательность операций, включающая в себя перенос, масштабирование и поворот относительно начала координат, может быть представлена одной матрицей, которая определяется как произведение матриц данных операций.

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

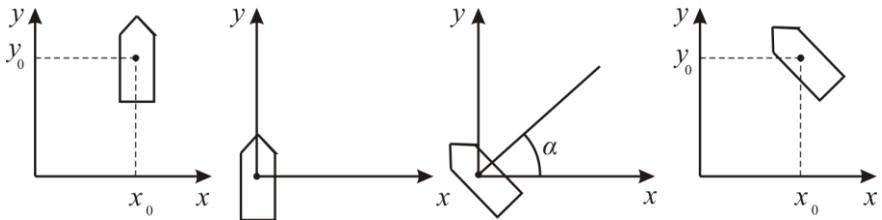


Рис. 5.8. Последовательность преобразований при повороте объекта вокруг точки $p_0(x_0, y_0)$ на угол α

Для примера рассмотрим задачу поворота объекта на плоскости относительно некоторой произвольной точки $p_0(x_0, y_0)$. Нам известна операция вращения только относительно начала координат. Но данную задачу можно представить как последовательность шагов, на каждом из которых будет использоваться только одна из элементарных операций: перенос, масштабирование или поворот относительно начала координат. Данная последовательность элементарных преобразований проиллюстрирована на рис. 5.8.

1. Перенос, при котором точка p_0 переходит в начало координат.

2. Поворот на заданный угол.

3. Перенос, при котором точка из начала координат возвращается в первоначальное положение p_0 .

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

5.4. Трехмерные матричные преобразования

Подобно тому, как двумерные преобразования в однородных координатах описываются матрицами размером 3×3 , трехмерные преобразования в однородных координатах могут быть представлены матрицами размером 4×4 . Трехмерная точка с координатами (x, y, z) записывается в однородных координатах как $(\omega x, \omega y, \omega z)$, где $\omega \neq 0$. Для получения декартовых координат также как и в двумерном случае необходимо разделить однородные координаты на ω .

Матрицы преобразований будем записывать в правосторонней системе координат. При этом положительный поворот определяется следующим образом. Если смотреть из положительной части оси вращения (например, оси z) в направлении начала координат, то ближайший поворот (на 90°) против часовой стрелки будет переводить одну положительную полуось в другую (в данном примере ось x в ось y , в соответствии с правилом циклической перестановки).

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

Запишем матрицу трехмерного переноса в однородных координатах по аналогии с двумерным случаем.

$$T(D_x, D_y, D_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ D_x & D_y & D_z & 1 \end{bmatrix}, \quad (5.38)$$

при этом

$$[x, y, z, 1] \cdot T(D_x, D_y, D_z) = [x + D_x, y + D_y, z + D_z, 1]. \quad (5.39)$$

Запишем матрицу трехмерного масштабирования в однородных координатах по аналогии с двумерным случаем.

$$S(S_x, S_y, S_z) = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.40)$$

при этом

$$[x, y, z, 1] \cdot S(S_x, S_y, S_z) = [x \cdot S_x, y \cdot S_y, z \cdot S_z, 1]. \quad (5.41)$$

Перейдем к операции поворота в однородных координатах. Так как при двумерном повороте в плоскости xu координаты z остаются неизменными, то поворот вокруг оси z записывается аналогично двумерному повороту:

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.42)$$

Матрица поворота вокруг оси x имеет вид:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.43)$$

Матрица поворота вокруг оси y имеет вид:

$$R_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha & 0 \\ \sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.44)$$

При повороте вокруг оси y в матрице произошла смена положения синуса угла с отрицательным знаком.

Правильность записанных матриц легко проверить поворотом одного из ортов на 90° , при этом он должен перейти в следующий по порядку орт на соответствующей координатной оси. Так при повороте вокруг оси z (Рис. 5.9 а) матрица поворота при повороте на угол 90° запишется:

$$R_z(90^\circ) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.45)$$

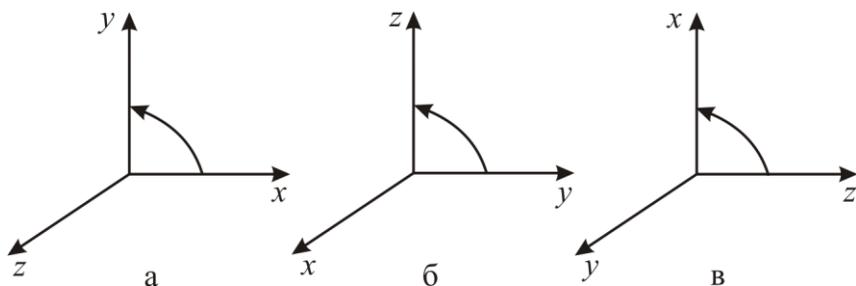


Рис. 5.9. Порядок перехода ортов при повороте на 90° вокруг оси z (а), оси x (б), оси y (в)

Нетрудно видеть, что если теперь перемножить вектор-строку орта $x = [1,0,0,1]$ на матрицу $R_z(90^\circ)$, то получим орт $y = [0,1,0,1]$, то есть при повороте на 90° орт x переходит в орт y . Аналогично при повороте вокруг оси x (рис. 5.9 б) матрица поворота на угол 90° запишется:

$$R_x(90^\circ) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.46)$$

Нетрудно видеть, что если теперь перемножить вектор-строку орта $y = [0,1,0,1]$ на матрицу $R_x(90^\circ)$, то получим орт $z = [0,0,1,1]$, то есть при повороте на 90° орт y переходит в орт z . Аналогично при повороте вокруг оси y матрица поворота на угол 90° запишется:

$$R_y(90^\circ) = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.47)$$

Нетрудно видеть, что если теперь перемножить вектор-строку орта $z = [0,0,1,1]$ на матрицу $R_y(90^\circ)$, то получим орт $x = [1,0,0,1]$, то есть при повороте на 90° орт z (рис. 2.9 в) переходит в орт x . Если бы знаки синусов не были бы изменены, то результат перемножения вектор-строки $z = [0,0,1,1]$ на матрицу поворота на угол 90° с противоположными знаками синусов, дал бы орт с противоположным направлением $-x = [-1,0,0,1]$.

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

$$T^{-1}(D_x, D_y, D_z) = T(-D_x, -D_y, -D_z). \quad (5.48)$$

Где $T^{-1}(D_x, D_y, D_z)$ – матрица обратного переноса.

Для операции масштабирования надо заменить коэффициенты масштабирования на их обратные значения:

$$S^{-1}(S_x, S_y, S_z) = S\left(\frac{1}{S_x}, \frac{1}{S_y}, \frac{1}{S_z}\right). \quad (5.49)$$

Для операции поворота необходимо поменять знак угла поворота:

$$R^{-1}(\alpha) = R(-\alpha). \quad (5.50)$$

Результатом нескольких последовательных поворотов вокруг различных осей будет матрица вида:

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.51)$$

Здесь верхняя матрица B размером 3×3 является ортогональной.

$$B = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (5.52)$$

Важным ее свойством является то, что обратная к ней матрица является транспонированной: $B^{-1} = B^T$.

$$B^{-1} = B^T = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix}. \quad (5.53)$$

Это свойство полезно тем, что при вычислениях достаточно поменять индексы местами и обратное преобразование получается автоматически.

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

$$A_p = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}. \quad (5.54)$$

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

6. ОСНОВНЫЕ РЕДАКТОРЫ РАСТРОВОЙ ГРАФИКИ

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

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

6.1. Редактор GIMP

GIMP – это мощный графический редактор с массой вспомогательных программ. Само название GIMP является аббревиатурой GNU Manipulation Image Program и переводится на русский язык как программа обработки изображений.

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

Разработчики программы постарались включить в нее все, начиная с основных средств рисования и редактирования, без которых не обходится ни один графический редактор, и

заканчивая самыми современными средствами обработки изображений. Типичные задачи, которые можно решать при помощи GIMP, включают в себя создание графики и логотипов, масштабирование и кадрирование фотографий, раскраска, комбинирование изображений с использованием слоёв, ретуширование и преобразования изображений в различные форматы. GIMP включает в себя все те средства, которые есть в программе Photoshop, и «понимает» формат файлов, используемый в этой программе, так что пользователи Photoshop могут легко перейти на использование GIMP. Имеется версия редактора GIMP для операционной системы Windows.

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

- XPaint – редактор растровой графики, похожий на Paint, но содержащий массу дополнительных возможностей;

- KPaint – встроенный в KDE редактор растровой графики, менее функционален, чем XPaint;

- KDE Icon Editor – редактор растровой графики, предназначенный для создания пиктограмм.

Сайт программы: www.gimp.org; www.gimp.ru.

6.2. Редактор Paint

Этот редактор является одним из самых простых графических редакторов в среде ОС Windows. Он появился в версии Windows 95 и с тех пор не изменился. Paint, в переводе с английского – краска, представляет собой простой однооконный редактор растровой графики, который, тем не менее, позволяет создать достаточно сложный рисунок. Графический редактор Paint ориентирован на процесс «рисования» изображения и комбинирования готовых фрагментов, а не на обработку готовых изображений, таких как отсканированные фотографии.

Программа Paint включает в себя средства для построения прямых и кривых линий, эллипсов и окружностей, прямоугольников, квадратов и многоугольников, как контурных, так и закрашенных. Есть инструменты для выделения фрагмента

рисунка, заливки замкнутой области цветом, а также инструменты имитирующие рисование кистью и пульверизатором. Имеется возможность создания надписи и задания толщины линии.

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

6.3. Редактор Adobe Photoshop

Среди графических редакторов является стандартом де-факто. Производится, распространяется и поддерживается компанией Adobe. Этот графический редактор предназначен для работы с растровыми изображениями.

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

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

Сайт программы: www.adobe.com/products/photoshop.

6.4. Редактор Paint Shop Pro

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

Paint Shop Pro поддерживает все распространенные форматы графических файлов, в том числе и PSD, принадлежащий

Adobe Photoshop, его главному конкуренту. В распоряжение пользователя попадает весьма объемная библиотека заранее разработанных эффектов и фильтров, которых более 70. Имеется также возможность создавать собственные эффекты и сохранять их в специальном файле для применения в будущем.

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

Незарегистрированная копия позволяет свободно редактировать графические файлы всех форматов. Желание компании получить с пользователя оплату выражается во время работы не слишком назойливыми сообщениями. После окончания тридцатидневного срока, дается своеобразный бонус, пользователь может бесплатно поработать с программой еще два месяца. Но вот после этого программа уже окончательно перестает функционировать.

Сайт программы: www.jacs.com/products/paintshoppro.

6.5. Редактор Corel PHOTO-PAINT

Еще один стандарт растровой графики, который вместе с Photoshop охватывает большинство пользователей, работающих с графическими пакетами. Графический редактор Corel Photo-Paint отличается исключительной мощностью функций и огромным спектром поддерживаемых графических форматов. Этот редактор представляет собой отличное средство для создания и редактирования растровых изображений и их композиций, коррекции и обработки цифровых фотографий. Он предоставляет большие возможности не только в области создания изображений, но, благодаря большому количеству корректирующих инструментов и настроек, также позволяет работать с уже готовыми изображениями, фотографиями и иллюстрациями. Причем данный редактор позволяет работать не только над композицией, но и над цветом, что является немаловажным и наиболее часто используемым в повседневной практике. Corel Photo Paint входит в состав пакета Corel Graphics Suite.

Сайт программы: www.corel.com.

6.6. Редактор Macromedia Fireworks

Macromedia Fireworks представляет собой комплексное решение для создания и оптимизации высококачественной "легкой" графики и веб-изображений. Fireworks включает в себя многообразие креативных инструментов и различные функции интерактивного дизайна. Программа предназначена для веб-дизайнеров и разработчиков, которым нужно быстро создавать, редактировать и оптимизировать изображения для веб-сайтов или мобильных устройств.

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

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

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

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

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

Сайт программы:

www.macromedia.com/software/fireworks.

7. ФОРМАТЫ ФАЙЛОВ РАСТРОВОЙ ГРАФИКИ

Формат графического файла – способ представления и расположения графических данных на внешнем носителе.

Знание файловых форматов и их возможностей является одним из ключевых факторов в допечатной подготовке изданий, подготовке изображений для web и в компьютерной графике вообще. Каждый из утвердившихся форматов прошел естественный отбор и доказал свою жизнеспособность. Все они имеют какие-то характерные особенности и возможности, делающие их незаменимыми в работе.

7.1. Формат PCX

Формат графических файлов, разработанный фирмой ZSoft для работы с графическим редактором PS Paintbrush, поддерживается множеством графических редакторов, программами сканирования изображений, настольными издательскими системами, программами рассылки факсов. Однако: не поддерживает изображения с оттенками серого или таблицы коррекции шкалы серого; не поддерживает цвета CMYK или другие системы, отличные от RGB; многочисленные варианты, особенно при работе с цветами, могут делать работу с файлом невозможной; неудобная схема сжатия в действительности может увеличивать размеры некоторых файлов.

Серьезным недостатком PCX формата наряду с ограниченностью цветовой палитры является наличие многочисленных версий. Последнее обусловлено тем, что PCX стандарт является открытым и используется многими независимыми разработчиками программного обеспечения.

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

7.2. Формат BMP

Разработка фирмы Microsoft для операционной системы Windows. BMP – сокращение от слова bitmap. Он поддержива-

ется всеми графическими редакторами, работающими под управлением этой операционной системы.

Представляет собой чрезвычайно простую структуру и служит для описания и визуализации небольших пиктограмм – иконок (icons), применяемых в графических интерфейсах. Способен хранить как индексированный (до 256 цветов), так и RGB-цвет (16700 оттенков). Возможно применение сжатия по принципу RLE. Существует разновидность формата BMP для операционной системы OS/2.

7.3. Формат GIF

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

Существуют две разновидности формата. Первой официальной версией GIF была GIF87a. Затем вышла новая усовершенствованная версия формата, обладающая дополнительными возможностями и получившая название GIF89a. Обе эти разновидности поддерживают технологию чередования строк (interlacing) при помощи которой изображение может появиться на экране путем постепенного увеличения четкости. Это особенно полезно, когда изображения выводятся на экран по мере поступления данных. Формат GIF89a позволяет добиваться еще целого ряда сильных визуальных эффектов: маски прозрачности; включение нескольких изображений; глобальные цветовые палитры для нескольких изображений; анимация – попеременный показ нескольких изображений; включение текстовых комментариев; настраиваемые задержки показа и чередование – interlacing; ожидание реакции пользователя.

Разработчики программ, использующих формат GIF, должны заплатить за лицензию на алгоритм сжатия LZW, применяемый в GIF.

7.4. Формат PNG

PNG произносится как «пинг» - аббревиатура словосочетания Portable Network Graphics. Как видно из названия, этот формат предназначен специально для передачи изображений по сетям.

Спецификация 1,0 формата PNG выглядит впечатляюще: как и GIF, формат поддерживает сжатие без потерь, чересстрочность (interlacing), прозрачность цвета (transparency), хранение пользовательских данных (например, комментариев). Но, в отличие от GIF, он может сохранять truecolor-изображения с глубиной цвета до 48 бит на пиксель, изображения в шкале серого цвета с глубиной до 16 бит на пиксель, поддерживает альфа-канал и гамма-индикатор. Кроме того, PNG может использоваться бесплатно. Правда, PNG 1,0 не может хранить несколько изображений в одном файле, и, следовательно, анимация, одна из самых привлекательных черт формата GIF, в PNG пока не доступна. Существует два подформата: PNG8 и PNG24. Цифры в названии обозначают максимально возможную глубину цвета. В PNG24 реализована поддержка 256 градаций прозрачности, за счет дополнительно альфа-канала с 256 градациями серого.

7.5. Формат JPEG

Формат JPEG – Joint Photographic Experts Group повсеместно используется для отображения фотографий и других тоновых изображений в www и в других сетях.

В отличие от формата GIF, формат JPEG сохраняет всю информацию о цвете. Кроме того, формат JPEG использует очень эффективный алгоритм уплотнения, который нередко дает значительное сокращение объема файла за счет удаления избыточной информации (от 10% до 60% от начального объема файла), не влияющей на видимое качество отображения документа. При открытии JPEG-файла происходит его автоматическая распаковка. Недостатком JPEG-уплотнения является то, что оно приводит к частичной потере данных: если упаковать изображение, а затем вновь распаковать его, то результат не

будет идентичным оригиналу. Между качеством изображения и степенью уплотнения существует обратная зависимость: чем более высокое качество будет задано для результирующего изображения, тем менее компактным будет упакованный файл.

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

7.6. Формат PSD

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

Начиная с версии 3.0, Adobe добавила поддержку слоев и контуров, поэтому формат версии 2,5 и ранее выделяется в отдельный подформат. Для совместимости с ним в более поздних версиях Photoshop имеется возможность включить режим добавления в файл одного базового слоя, в котором слиты все слои.

7.7. Формат TIFF

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

Основное преимущество TIFF-формата заключается в том, что он поддерживает очень большие изображения, файлы, содержащие несколько изображений (слоев), и разнообразные

методы сжатия. Хранит данные в структурированном виде, что позволяет приложениям осуществлять быстрый доступ к различным фрагментам большого изображения.

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

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

Важным достоинством формата является возможность его переноса на различные платформы.

7.8. Формат RAW

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

7.9. Формат DjVu

Графический формат DjVu (произносится дежавю) разработан фирмой AT&T в первую очередь для размещения в Internet отсканированных изображений. Его уникальной особен-

ностью является чрезвычайная компактность при хранении изображений в высоком разрешении (300 dpi и выше). Таким образом, DjVu является уникальным инструментом для открытия Internet-доступа к фондам библиотек.

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

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

Формат поддерживает гиперссылки и многостраничные документы.

8. ВЕКТОРНАЯ ГРАФИКА

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

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

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

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

Разумеется, в растровой графике тоже существуют линии, но там они рассматриваются как комбинации точек. Для каждой точки линии в растровой графике отводится одна или несколько ячеек памяти. Чем больше цветов имеют точки растрового изображения, тем больше ячеек памяти им выделяется. Соответственно, чем длиннее растровая линия, тем больше памяти она занимает. В векторной графике объем памяти, занимаемый линией, не зависит от размеров линии, так как линия представляется в виде формулы, а точнее говоря в виде нескольких параметров. Чтобы не происходило с этой линией, меняются только ее параметры, хранящиеся в ячейках памяти. Количество же ячеек остается неизменным.

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

При редактировании элементов векторной графики изменяются параметры прямых и изогнутых линий, описывающих форму этих элементов. Можно переносить элементы, менять

их размер, форму и цвет, но это не отразится на качестве их визуального представления. Векторная графика не зависит от разрешения, то есть может быть показана в разнообразных выходных устройствах с различным разрешением без потери качества.

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

Основными достоинствами векторной графики являются:

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

К основным недостаткам векторной графики можно отнести:

- сложность экспорта из растрового формата в векторный формат;
- невозможность применения обширной библиотеки эффектов, используемых при работе с растровыми изображениями.

8.1. Математические основы векторной графики

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

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

8.1.1. Прямая линия

Известно, что для задания прямой линии достаточно двух параметров. Обычно прямая линия описывается уравнением

$$y = kx + b. \quad (8.1)$$

Зная параметры k и b , всегда можно нарисовать бесконечную прямую линию в известной системе координат.

8.1.2. Кривая второго порядка

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

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

$$x^2 + a_1 y^2 + a_2 xy + a_3 x + a_4 y + a_5 = 0. \quad (8.2)$$

Из выражения (8.2) следует, что для описания бесконечной кривой второго порядка достаточно пяти параметров, то есть пяти коэффициентов a_i ($i = 1, 2, 3, 4, 5$).

8.1.3. Кривая третьего порядка

Отличительная особенность кривых третьего порядка состоит в том, что они могут иметь точку перегиба, в отличие от кривых второго порядка.

Кривые третьего порядка хорошо соответствуют тем линиям, которые мы видим в повседневной жизни, например, линиям изгиба человеческого тела. По этой причине в качестве основных объектов векторной графики используют именно такие линии. Все прямые и кривые второго порядка являются частными случаями кривых третьего порядка.

В общем случае уравнение кривой третьего порядка записывается так:

$$x^3 + a_1 y^3 + a_2 x^2 y + a_3 x y^2 + a_4 x^2 + a_5 y^2 + a_6 x y + a_7 x + a_8 y + a_9 = 0. \quad (8.3)$$

Из выражения (8.3) следует, что для описания кривой третьего порядка достаточно девяти параметров, то есть девяти коэффициентов a_i .

8.1.4. Кривые Безье

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

Вопрос о построении аппроксимирующего многочлена привлек многих математиков. Среди них одну из главных ролей сыграл выдающийся ученый Сергей Натанович Берштейн, который окончил Харьковский университет, учился в Сорбонне, а в начале XX века предложил новое доказательство теоремы Вейерштрасса с помощью теории вероятностей. В этом случае необходимый полином строится в явном виде, а не параметрически. Именно данный полином и стал основой сплайновых кривых, в частности NURBS-кривых и кривых Безье.

К сожалению, нашему великому соотечественнику не очень повезло, так как чаще известны имена людей, применивших открытия, чем авторов этих открытий. То же самое произошло и в данном случае. Кривые Сергея Натановича Берштейна, известны всему миру под именем кривых Безье.

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

Метод построения кривых Безье основан на использовании пары касательных, проведенных к линии в точках ее окончания, как это показано на рис. 8.1.

Для построения кривой требуется четыре контрольные точки. Но кривая физически проходит только через две из них, они получили название опорных. Одна из этих точек называется начальной, то есть *start point*, а другая – конечной, то есть

end point. Две точки остаются в стороне от кривой, они получили название управляющих, то есть control point. Для того чтобы их не «потерять», в программах векторной графики управляющие точки соединяются с опорными точками линией.

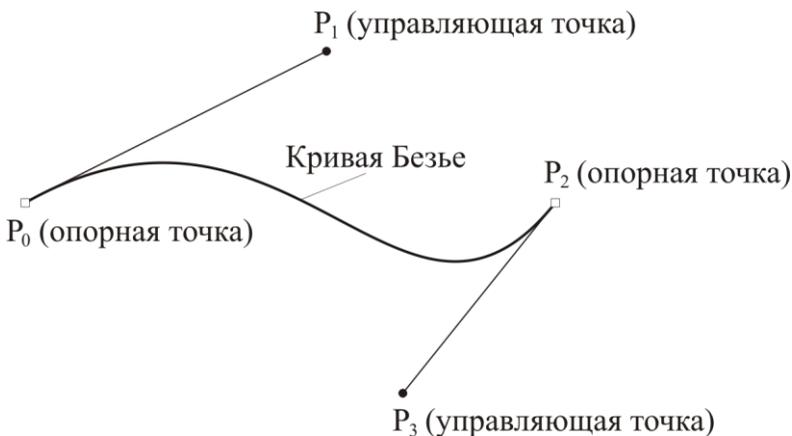


Рис. 8.1. Канонический вид кривой Безье

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

В программах векторной графики существует единственный способ – это интерактивное перемещение опорных точек, как это показано на рис. 7.2. Если перемещается начальная или конечная точки, то кривая соответствующим образом меняется, а именно вытягивается или сжимается, как упругая резинка. Перемещение управляющих точек изменяет кривизну соответствующей части кривой Безье.

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

В каждом сегменте можно добавлять опорные точки, которые тоже позволяют изменять форму кривой. Добавление

новых опорных точек в пределах одного сегмента кривой не противоречит тому условию, что отдельные кривые соединяются в цепь. Просто кривая Безье добавляется не к концу контура, а размещается внутри уже имеющегося контура.

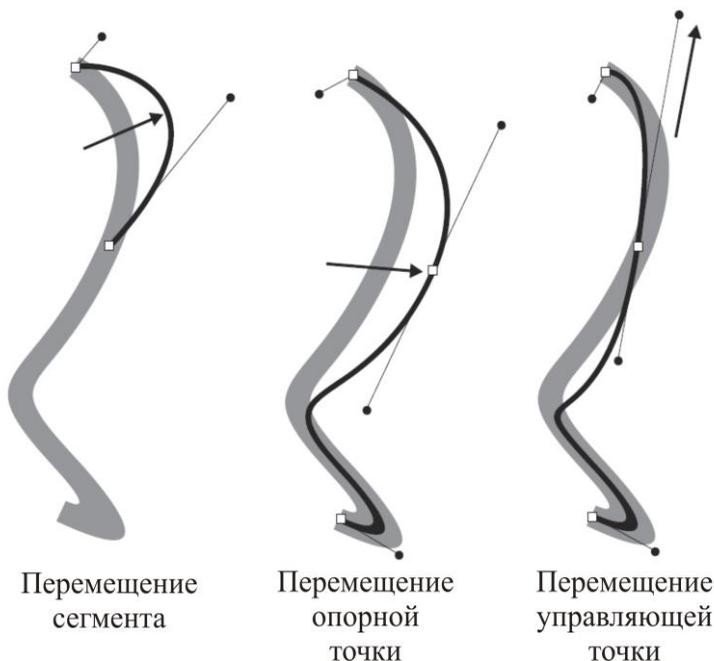


Рис. 8.2. Изменение канонической формы кривой Безье перемещением опорных точек

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

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

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

8.2. Типы опорных точек

8.2.1. Точка перегиба

Первый тип опорной точки показан на рис. 7.3. Данная опорная точка соединяет два сегмента, обеспечивая независимость управляющих точек, по направлению и длине друг от друга. Такое состояние сегментов называется изгиб. А опорная точка называется – точкой перегиба.

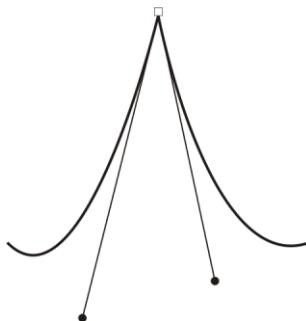


Рис. 8.3. Пример сочленения двух элементарных сегментов точкой перегиба

8.2.2. Гладкая опорная точка

Угловое сочленение сегментов – изгиб далеко не всегда обеспечивает достижение поставленной цели. Например, для создания окружности необходимо обеспечить соединение, которое в черчении и геометрии называют гладким сопряжением, когда одна кривая плавно переходит в другую. Такое сочленение обеспечивает гладкая опорная точка – smooth. Пример сочленения с гладкой опорной точкой показан на рис.7.4.

8.2.3. Симметричная опорная точка

В программе Corel Draw предусмотрен подвид гладкого сочленения, который называется симметричный узел – *symm*, от слова *symmetrical*.

Суть гладкого сочленения симметричной опорной точкой состоит в том, что управляющие линии фиксируют не только

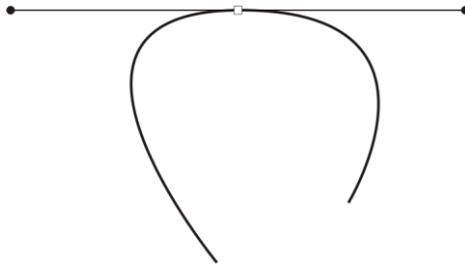


Рис.8.4. Пример сочленения двух элементарных сегментов гладкой опорной точкой

по направлению, но и по величине (длина направляющих линий одинакова). Если одну из них увеличивать или уменьшать, то вторая будет синхронно повторять изменение размеров вслед за первой.

8.2.4. Тангенциальная опорная точка

Это случай гладкого сочленения прямолинейного и криволинейного сегментов. Пример такого сочленения показан на рис. 7.5. Такая точка получила название тангенциальной - contact point. При выделении такая точка обозначается треугольником.

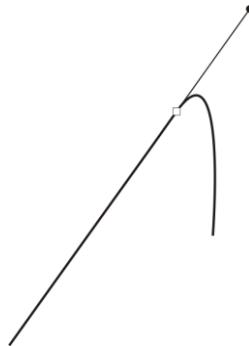


Рис.8.5. Пример сочленения двух элементарных сегментов тангенциальной опорной точкой

Логика этой точки заключается в следующем: для того чтобы криволинейный сегмент гладко сопрягался с прямой линией, касательная криволинейного сегмента должна совпадать с продолжением прямого сегмента. Поэтому управляющая точка криволинейного сегмента способна двигаться только вдоль этой касательной.

9. ОСНОВНЫЕ РЕДАКТОРЫ ВЕКТОРНОЙ ГРАФИКИ

9.1. Редактор XFig

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

Стандартный набор геометрических фигур включает в себя: окружность и эллипс с возможностями их создания по двум различным начальным условиям – радиусу или диаметру, замкнутые и незамкнутые линии и ломанные, правильные и неправильные многоугольники, дуги и различные прямоугольники. Объекты можно создавать, удалять, перемещать или изменять. Можно менять атрибуты объектов, такие как цвет, тип линии. Доступно 35 шрифтов для изображения текста.

9.2. Редактор Adobe Illustrator

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

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

Сайт программы: www.adobe.com/products/illustrator.

9.3. Редактор CorelDRAW

На сегодняшний день, наиболее распространенный редактор векторной графики. Графический редактор CorelDRAW предназначен для работы с векторной графикой и является несомненным лидером среди аналогичных программ. Популярность CorelDRAW объясняется большим набором средств создания и редактирования графических образов, удобным интерфейсом и высоким качеством получаемых изображений. Особенно удобен CorelDRAW при создании иллюстраций, со-

стоящих из множества рисунков, фотографий и надписей. Расположить в нужных местах компоненты изображения с помощью CorelDRAW чрезвычайно просто. Помимо обработки векторной графики, в этом пакете существует обработчик растровой графики – Corel PHOTO-PAINT, трассировщик изображений – CorelTRACE, редактор шрифтов, подготовки текстур и создания штрихкодов, а также огромные коллекции с изображениями – CorelGallery.

Сайт программы: www.corel.com.

9.4. Редактор Macromedia FreeHand

Был разработан фирмой Aldus, позднее права перекупила Macromedia. Арсенал изобразительных средств FreeHand позволяет решать самые сложные задачи дизайна на высоком профессиональном уровне. Программа FreeHand универсальна и может быть использована как для создания книжных иллюстраций, так и для решения задач в области Web-дизайна. Пакет обеспечивает высокое удобство работы с кистями, эффекты выдавливания и трехмерное вращение векторных объектов, может работать в многостраничном режиме. Кроме того, FreeHand полностью интегрирован с Flash и другими продуктами MX-линейки Macromedia.

Сайт программы: www.macromedia.com/software/freehand.

10. ФОРМАТЫ ФАЙЛОВ ВЕКТОРНОЙ ГРАФИКИ

10.1. Формат PostScript

PS – Adobe PostScript. PostScript – язык описания страниц, то есть язык управления лазерными принтерами фирмы Adobe. PostScript соединил в себе лучшие возможности принтеров и плоттеров. Подобно плоттерам, PostScript предоставляет возможность вывода высококачественной векторной графики и единый язык управления, который может быть использован любым производителем принтеров. Подобно матричным принтерам, PostScript предлагает удобные возможности по печати растровой графики и текста. В отличие от тех и других, PostScript может совмещать все эти типы вывода на одной странице, давая намного больше гибкости, чем до этого имел любой принтер или плоттер. Файлы этого формата с расширением .ps или, реже, .prn получаются с помощью функции Print to file графических программ при использовании драйвера PostScript-принтера. Такие файлы содержат в себе сам документ (только то, что располагалось на страницах), все связанные файлы, использованные шрифты, а также другую информацию: цветodelение, дополнительные платы, полутоновый растр для каждой платы, линиатуру растра и прочие данные для выводного устройства. Если файл закрыт правильно, то не имеет значения, на какой платформе он делался, были ли использованы шрифты True Type или Adobe Type 1. Нужно только учитывать версию языка. Недавно Adobe выпустила PostScript level 4. Тема языка PostScript – отдельный большой разговор. Ниже приводятся еще два формата непосредственно с ним связанных.

10.2. Формат EPS

EPS – Encapsulated PostScript – упрощенный PostScript. Не может содержать в одном файле более одной страницы, не сохраняет ряд установок для принтера. Как и в файлы печати PostScript, в EPS записывают конечный вариант работы, хотя такие программы как Adobe Illustrator, Photoshop и Macromedia

FreeHand могут использовать его как рабочий. EPS используется для передачи векторов и растра в издательские системы, создается почти всеми программами, работающими с графикой.

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

10.3. Формат PDF

Формат PDF – Portable Document Format, предложен фирмой Adobe как независимый от платформы формат, в котором могут быть сохранены иллюстрации, как векторные, так и растровые, и текст, со множеством шрифтов и гипертекстовых ссылок. Для работы с этим форматом компания Adobe выпустила пакет Acrobat. Acrobat Distiller переводит в PDF PostScript-файлы, Acrobat Exchange позволяет их редактировать.

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

10.4. Формат AI

AI – Adobe Illustrator Document – Adobe Illustrator. Она может содержать в одном файле только одну страницу, имеет маленькое рабочее поле – всего 3×3 метра. В целом несколько

уступает FreeHand CorelDRAW по иллюстративным возможностям, тем не менее, отличается наибольшей стабильностью и совместимостью с PostScript. AI поддерживают почти все программы так или иначе связанные с векторной графикой. Формат Illustrator является наилучшим посредником при передаче векторов из одной программы в другую, с PC на Macintosh и назад. Кроме того, если вы работаете в основном в Photoshop, то Illustrator станет лучшим помощником, так как имеет с Photoshop одинаково организованный интерфейс и горячие клавиши. Photoshop понимает форматы Illustrator (AI и EPS).

10.5. Формат FH

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

10.6. Формат CDR

Формат файла CDR – векторное изображение или рисунок, созданный с помощью программы CorelDRAW. Данный формат файла разработан компанией Corel для использования в собственных программных продуктах.

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

11. ФРАКТАЛЬНАЯ ГРАФИКА

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

Самыми известными и наглядными фрактальными объектами являются деревья. От каждой ветки отделяются меньшие, похожие на нее, от тех – еще меньшие и так далее. По отдельной ветке можно проследить свойства всего дерева. Фрактальными свойствами обладают многие природные объекты. Так снежинка при увеличении оказывается фракталом. Фрактальность прослеживается не только в кристаллах и растениях. Если посмотреть на береговую линию моря на картах все более крупного масштаба, то становятся видны все новые изгибы и изломы, которые по своей форме похожи на более крупные.

Роль фракталов в компьютерной графике на сегодняшний день достаточно велика. Они незаменимы когда, например, требуется, с помощью нескольких коэффициентов, задать линии и поверхности очень сложной формы. С точки зрения компьютерной графики, фрактальная геометрия позволяет генерировать искусственные облака, горы, леса, деревья, поверхность моря. Фактически фрактальная геометрия дает способ простого представления сложных неевклидовых объектов, образы которых весьма похожи на природные.

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

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

Такое определение фракталу дал Мандельброт.

Фрактал можно определить как бесконечно самоподобную структуру.

11.1. Классификация фракталов

11.1.1. Геометрические фракталы

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

Примером такого фрактального объекта является триадная кривая Коха, построение которой показано на рис. 11.1.

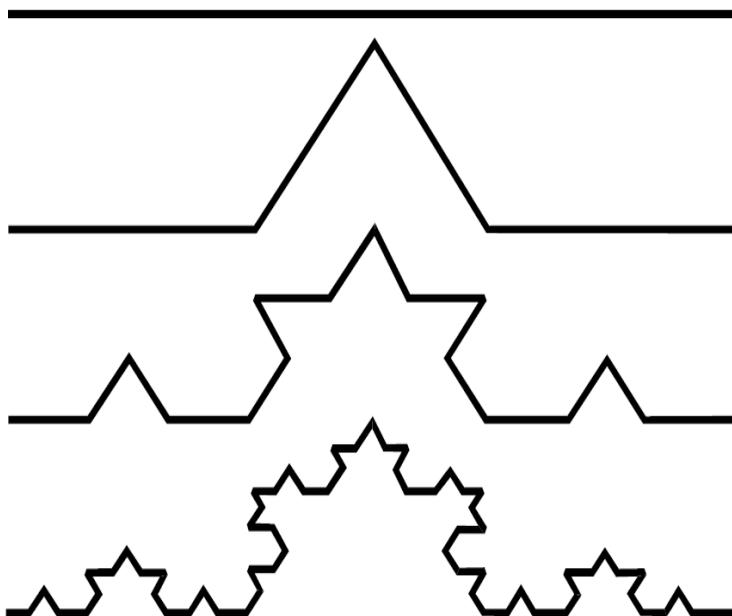


Рис.11.1. Построение триадной кривой Коха

Построение кривой начинается с отрезка единичной длины – это нулевое поколение кривой Коха. Далее каждое звено

(в нулевом поколении один отрезок) заменяется на образующий элемент. В результате такой замены получается следующее поколение кривой Коха. В первом поколении это кривая из четырех линейных звеньев, каждое длиной по $1/3$. Для получения третьего поколения проделывают те же действия – каждое звено заменяется на уменьшенный образующий элемент. Итак, для получения каждого последующего поколения, все звенья предыдущего поколения необходимо заменить уменьшенным образующим элементом. Кривая n -ого порядка при любом конечном n называется предфракталом.

На рис.11.1 представлены три поколения кривой. При n стремящемся к бесконечности кривая Коха становится фрактальным объектом.

В компьютерной графике использование геометрических фракталов необходимо при получении изображений деревьев, кустов, леса, облаков, гор, береговой линии моря и других природных объектов. Двумерные геометрические фракталы используются для создания объемных текстур.

11.1.2. Алгебраические фракталы

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

Методов получения алгебраических фракталов несколько. В качестве примера рассмотрим множество Мандельброта рис.10.2. Алгоритм его построения достаточно прост и основан на простом итеративном выражении: $Z[i+1] = Z[i] \cdot Z[i] + C$, 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 ите-

раций, 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 раstra окрашивается в черный цвет).

Вышеописанный алгоритм дает приближение к так называемому множеству Мандельброта. Множеству Мандельброта принадлежат точки, которые в течение бесконечного числа итераций не уходят в бесконечность (точки, имеющие черный цвет). Точки, принадлежащие границе множества (именно там возникает сложные структуры) уходят в бесконечность за конечное число итераций, а точки, лежащие за пределами множества, уходят в бесконечность через несколько итераций (белый фон).

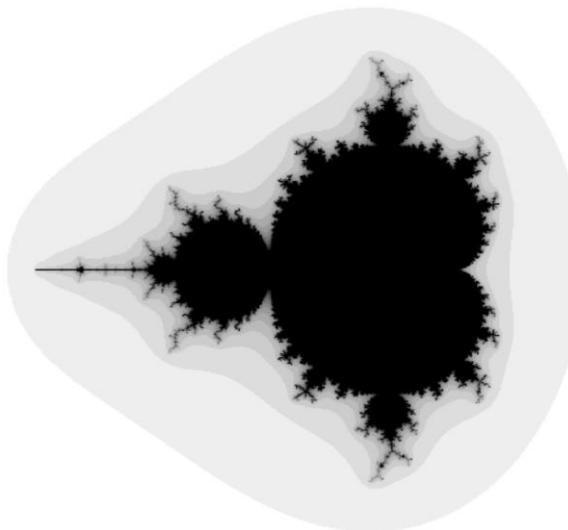


Рис.11.2. Множество Мандельброта

11.1.3. Стохастические фракталы

Еще одним известным классом фракталов являются стохастические фракталы. Они получаются в том случае, если в

итерационном процессе случайным образом менять какие-либо параметры. При этом получаются объекты очень похожие на природные: несимметричные деревья, изрезанные береговые линии и т.д. (рис.11.3).

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

Способность фрактальной графики моделировать образы природы вычислительным путем используется в компьютерной графике для автоматической генерации необычных иллюстраций достаточно хорошо отражающих реальную действительность.

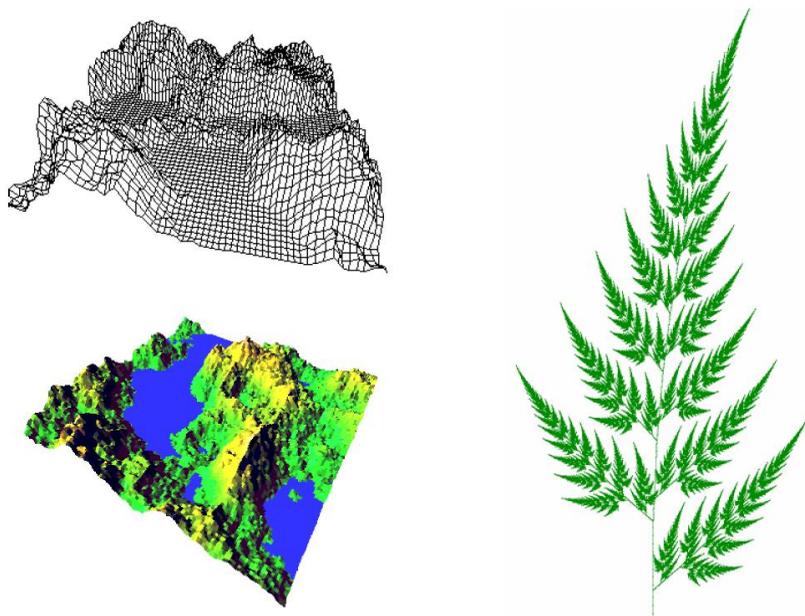


Рис.11.3. Примеры стохастических фракталов

12. ТРЕХМЕРНАЯ 3D-ГРАФИКА

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

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

- визуальные спецэффекты для кино- и видеоиндустрии;
- телевизионная реклама;
- интерактивные игры;
- промышленный и архитектурный дизайн;
- научная, медицинская и судебная визуализация;
- компьютерные тренажеры и обучающие программы.

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

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

12.1. Типы пространств

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

Чаще всего программы трехмерного моделирования предоставляют следующие варианты пространств:

- пространство объекта – предназначено для моделирования (описания) формы объекта в его собственной (локальной) системе координат безотносительно к тому, где он будет размещен на сцене, как ориентирован и масштабирован. У каждого объекта существует своя собственная система координат;

- мировое пространство используется для размещения объектов на сцене, осуществления аффинных трансформаций (перемещения, поворота и масштабирования объектов), описания освещения сцены, вычисления столкновений между объектами при моделировании динамики их движения и т.п. Это единое пространство для всех объектов сцены;

- видовое пространство ассоциировано с виртуальным наблюдателем (обычно камерой) или определенной проекцией сцены (например, фронтальным видом) и описывает ту часть сцены, которая доступна для просмотра и работы в видовом окне;

- экранное пространство – это пространство (плоскость), в котором отображаются аксонометрические или перспективные проекции объектов на плоскость поверхности монитора;

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

12.2. Моделирование объектов

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

- первые из них используются для моделирования объектов вещественного мира (персонажей, предметов);

- вторые применяются для придания сцене реалистичности (например, правильного освещения), для моделирования физических сил, действующих на объекты (например, гравитации или порывов ветра).

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

12.2.1. Геометрические объекты

Краткий перечень типов геометрических объектов:

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

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

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

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

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

- поверхности Безье – это математически гладкие поверхности, описываемые расположением вершин Безье. Эти вер-

шины определяют их кривизну при помощи дополнительных управляющих точек на концах касательных к поверхности векторов;

- NURBS-поверхности – это наиболее универсальный и эффективный способ моделирования неоднородных криволинейных поверхностей. Такие поверхности описываются в особом четырехмерном гомогенном (однородном) пространстве, в котором каждая управляющая вершина, кроме трех координат x , y и z , имеет еще и дополнительную весовую характеристику;

- составные объекты – представляют собой комбинацию двух или более смоделированных заранее заготовок. В зависимости от того, какое именно составное тело создается, заготовками могут служить кривые или объемы (поверхности);

- системы частиц – это объекты, генерирующие по заданному алгоритму частицы с определенной формой, начальной скоростью, сроком жизни и другими характеристиками. Такие анимационные объекты используют для моделирования дождя, пузырьков газа в жидкости, осколков взрывающихся снарядов и тому подобных образцов объективной реальности;

- динамические объекты – позволяют моделировать объекты, реагирующие на приложенные к ним внешние силы: пружины и амортизаторы. Используются при моделировании динамики движения объектов.

12.2.2. Негеометрические объекты

К ним относятся:

- источники света – используются при моделировании наружного и интерьерного освещения. Различные типы источников реализуют различные алгоритмы распространения света;

- камеры – позволяют полностью контролировать отображение объектов в плоскости кадра. Важнейшей характеристикой является фокусное расстояние объектива камеры, определяющее ее поле зрения. Оба этих параметра взаимосвязаны и измеряются в миллиметрах и градусах соответственно. Еще одной важной характеристикой камеры являются плоско-

сти отсечения, ограничивающие видимую по глубине – расстоянию от наблюдателя, часть сцены;

- системы сочленений – это структуры, состоящие из иерархически связанных «костей», описывающих сложную кинематику движения моделируемого объекта (например, человека или животного);

- Искривители пространства – реализуют динамические воздействия внешних сил на объекты, это – своеобразные силовые поля, влияющие на определенные объекты. Примерами могут служить волновая деформация поверхности или ударная волна, разбрасывающая фрагменты объекта в пространстве.

12.2.3. Материалы и карты

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

Можно выделить следующие свойства поверхностей, которые определяют результат взаимодействия материала со светом:

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

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

12.2.4. Текстурирование материалов

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

В качестве карт могут быть использованы изображения, сохраненный в файлах различных форматов (BMP, TIF, JPG, EPS), либо процедурные текстуры, являющиеся наборами правил быстрого построения нужного узора.

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

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

11.2.5. Дополнительные свойства материалов

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

12.2.6. Анимация

Компьютерная анимация – вид анимации, создаваемый при помощи компьютера. На сегодня получила широкое применение, как в области развлечений, так и в производственной, научной и деловой сферах. Являясь производной от компьютерной графики, анимация наследует те же способы создания изображений: векторная графика, растровая графика, фрактальная графика, трёхмерная графика (3D).

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

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

По принципу анимирования можно выделить несколько видов компьютерной анимации:

- анимация по ключевым кадрам. Расстановка ключевых кадров производится аниматором. Промежуточные же кадры генерирует специальная программа. Этот способ наиболее близок к традиционной рисованной анимации, только роль фазовщика берет на себя компьютер, а не человек;
 - запись движения. Данные анимации записываются специальным оборудованием с реально двигающихся объектов и переносятся в компьютер. Распространённый пример такой техники – Motion capture (захват движений). Актеры в специальных костюмах совершают движения, которые записываются камерами и анализируются специальным программным обеспечением. Итоговые данные о перемещении суставов и конечностей актеров применяют к трёхмерным скелетам виртуальных персонажей. Такой же метод используют для переноса мимики живого актера на его трёхмерный аналог в компьютере;
 - процедурная анимация полностью или частично рассчитывается компьютером. Сюда можно выделить следующие её виды:
 - симуляция физического взаимодействия твёрдых тел;
 - имитация движения систем частиц, жидкостей и газов;
 - имитация взаимодействия мягких тел (ткани, волосы);
 - расчёт движения иерархической структуры связей (скелета персонажа) под внешним воздействием (Ragdoll).
 - имитация автономного (самостоятельного) движения персонажа. Примером такой системы является программа Euphoria;
 - программируемая анимация. Широкое применение в сети получили два языка, с помощью которых программируются движения анимируемых объектов: Java-Script – браузерный язык и Action-Script – язык работы с приложениями Flash;
- Преимущество программируемой анимации – в уменьшении размера исходного файла, недостаток – нагрузка на процессор клиента;
- конструкторы анимаций. Для создания анимированных изображений существует множество программ как платных, так и бесплатных: Adobe Photoshop – платная, GIMP (чаще исполь-

зуется на Linux) – бесплатная, Adobe Flash Professional – платная, CoffeeCup – условно бесплатная;

- создание анимации с помощью цифрового фотоаппарата.

Программное обеспечение, позволяющее задействовать цифровой фотоаппарат для съёмки анимации, применяется также часто, как и ставшие привычными 3D- или 2D-пакеты. Любая программа такого типа обеспечивает управление цифровым фотоаппаратом через компьютер и работу с полученными кадрами.

12.2.7. Рендеринг

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

Визуализация – один из наиболее важных разделов в компьютерной графике, и на практике он тесным образом связан с остальными. Обычно программные пакеты трехмерного моделирования и анимации включают в себя также и функцию рендеринга. Существуют отдельные программные продукты, выполняющие рендеринг.

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

Компьютерная программа, производящая рендеринг, называется рендером или рендерером.

12.2.7.1. Методы рендеринга

Разработано множество алгоритмов визуализации, которые составляют четыре группы методов, более эффективных, чем моделирование всех лучей света, освещающих сцену:

- растеризация (rasterization) совместно с методом сканирования строк (scanline rendering). Визуализация производится проецированием объектов сцены на экран без рассмотрения эффекта перспективы относительно наблюдателя.

- Ray casting (рейкастинг). Сцена рассматривается, как наблюдаемая из определённой точки. Из точки наблюдения на объекты сцены направляются лучи, с помощью которых определяется цвет пикселя на двумерном экране. При этом лучи прекращают своё распространение (в отличие от метода обратного трассирования), когда достигают любого объекта сцены либо её фона. Возможно использование каких-либо очень простых способов добавления оптических эффектов. Эффект перспективы получается естественным образом в случае, когда бросаемые лучи запускаются под углом, зависящим от положения пикселя на экране и максимального угла обзора камеры.
- трассировка лучей (ray tracing) похожа на метод бросания лучей. Из точки наблюдения на объекты сцены направляются лучи, с помощью которых определяется цвет пикселя на двумерном экране. Но при этом луч не прекращает своё распространение, а разделяется на три компонента, луча, каждый из которых вносит свой вклад в цвет пикселя на двумерном экране: отражённый, теневой и преломленный. Количество таких разделений на компоненты определяет глубину трассирования и влияет на качество и фотореалистичность изображения. Благодаря своим концептуальным особенностям, метод позволяет получить очень фотореалистичные изображения, но при этом он очень ресурсоёмкий, и процесс визуализации занимает значительные периоды времени.
- трассировка пути (path tracing) содержит похожий принцип трассировки распространения лучей, однако этот метод является самым приближенным к физическим законам распространения света. Также является самым ресурсоёмким.

Передовое программное обеспечение обычно совмещает в себе несколько техник, чтобы получить достаточно качественное и фотореалистичное изображение за приемлемые затраты вычислительных ресурсов.

12.2.7.2. Математическое обоснование рендеринга

Реализация механизма рендеринга всегда основывается на физической модели. Ключом к теоретическому обоснова-

нию моделей рендеринга служит уравнение рендеринга. Оно является наиболее полным формальным описанием части рендеринга, не относящейся к восприятию конечного изображения. Все модели представляют собой какое-то приближённое решение этого уравнения.

$$L_0(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) L_i(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}'$$

Неформальное толкование таково: Количество светового излучения (L_0), исходящего из определённой точки в определённом направлении есть собственное излучение L_e и отражённое излучение L_i . Отражённое излучение есть сумма по всем направлениям приходящего излучения, умноженного на коэффициент отражения n из данного угла ω . Объединяя в одном уравнении приходящий свет с исходящим в одной точке, это уравнение составляет описание всего светового потока в заданной системе.

14. НАСТОЛЬНЫЕ ИЗДАТЕЛЬСКИЕ СИСТЕМЫ

Настольная издательская система (Desktop Publishing) включает в себя все технические и программные аспекты компьютерной графики. Она представляет собой – комплекс компьютерных аппаратных и программных средств, служащих для предпечатной подготовки оригинал-макетов печатной продукции (книг, газет, журналов, буклетов и т. п.). Обеспечивает набор текстов и формул, встраивание рисунков, проверку орфографии и проверку качества печатной продукции. При изготовлении исходных материалов для печати обычно используют текстовый редактор Word для набора текстов и графические пакеты для создания иллюстраций. Наиболее распространены следующие настольные издательские системы:

- Corel Ventura – одна из первых систем, ориентирована на издание многостраничных публикаций. Обладает большими возможностями по размещению и оформлению текста. Последняя версия распространяется с пакетом Corel Draw позволяющим создавать высококачественные иллюстрации.

- PageMaker – популярная программа с многочисленными функциями, предназначена для подготовки газетных и книжных изданий.

- QuarkXPress – настольная издательская система для профессиональной работы по оформлению изданий любой сложности и содержания.

Выделяют три основных уровня обеспечивающих надежное функционирование настольной издательской системы:

- аппаратный уровень (hardware level) представляет собой совокупность устройств, с помощью которых происходит ввод, обработка, хранение, передача и вывод информации;

- программный уровень (software level) представляет собой совокупность информационных элементов (программ и их команд), с помощью которых осуществляется управление как текстовой и изобразительной информацией, так и аппаратным оборудованием;

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

14.1. Аппаратный уровень

Состоит из следующих компонентов:

- устройства ввода информации (input devices) – обеспечивают преобразование информации с разнообразных носителей в цифровую форму, что дает возможность ее дальнейшей компьютерной обработки;

- устройства обработки, хранения и передачи информации (process, storage and transfer devices) являются ядром аппаратного уровня;

- устройства вывода информации (output devices) – обеспечивают перевод цифровой информации в форму, понятную и доступную человеку в виде картинки на экране монитора или на бумаге.

14.1.1. Устройства ввода информации

Главная функция устройств ввода информации – преобразование какого-либо вида информации в цифровую форму.

К таким устройствам относятся: клавиатура (keyboard); мышь (mouse); трекбол (trackball); сканер (scanner); цифровая камера (digital camera); графический планшет (graphics tablet); дигитайзер (digitizer); и т.д.

14.1.2. Устройства обработки, хранения и передачи информации

Чаще всего речь идет об аппаратном уровне компьютера, который представляет собой многоуровневую структуру.

Устройства обработки – процессор, видеопроцессор.

Устройства хранения информации: оперативное запоминающее устройство (RAM); память видеокарты; магнитные носители (FDD, HDD); оптические носители (CD, DVD и т.д.);

магнитооптические носители (MO, MOD Drive); сменные диски и носители (flash-накопители); ленточные накопители (стример, мини DV); и т.д.

Устройства передачи – порты и другие компоненты.

14.1.3. Устройства вывода информации

Выполняют функцию обратную вводу информации и обеспечивают преобразование цифровой информации в понятный человеку вид.

Выделяют два основных способа визуализации информации: средства электронной визуализации (мониторы, проекторы); средства физического вывода (струйные, лазерные, фотопринтеры, графопостроители, фотонаборные автоматы).

14.2. Программный уровень

Любая последовательность действий, выполняемых компьютером, определяется алгоритмами, заложенными в программах. Соответственно, выделяют следующие основные классы программного обеспечения: редакторы растровой графики; редакторы векторной графики; редакторы трехмерной графики; программы фрактальной графики; программы верстки; вспомогательные программы (операционная система, утилиты, plug-in, viewer, конверторы, браузеры, архиваторы и многие другие).

14.3. Пользовательский уровень

Чаще всего эстетическая и художественная составляющая не являются частью технических систем. По этой причине пользовательский уровень – это человек, работающий за компьютером с его индивидуальным эстетическим и художественным восприятием.

ЗАКЛЮЧЕНИЕ

В пособии рассмотрены отдельные вопросы, относящиеся к теоретической части дисциплины «Компьютерная графика», которые позволяют получить общее представление о предмете. Практическое освоение прикладных пакетов графических программ осуществляется в процессе лабораторных и практических занятий.

Читатели, у которых появился особый интерес к данному предмету, могут обратиться к специальной литературе по компьютерной графике и получить более глубокие и детальные познания по интересующему их конкретному вопросу или предмету в целом.

Авторы выражает признательность рецензентам и редактору за внимательное прочтение рукописи и сделанные замечания, которые способствовали улучшению качества пособия.

ГЛОССАРИЙ

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

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

Grayscale (серая шкала или шкала серого цвета) - одноканальное представление изображения с 256 уровнями (градациями) серого цвета. На каждый пиксель изображения отводится по 1 байт (8 бит). В этой модели различные оттенки серого задаются с помощью изменения процентного содержания двух цветов: черного и белого.

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

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

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

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

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

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

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

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

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

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

Двумерная (2D) графика – изображение, имеющее два измерения, то есть лежащее в плоскости.

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

Интерактивность (interactive) – свойство программного обеспечения, обеспечивающее реакции со стороны программы в ответ на какие-либо действия пользователя.

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

помощью программного обеспечения обладающего свойством интерактивности.

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

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

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

Контраст (contrast) – отношение яркости абсолютно белой и абсолютно черной точек экрана. Степень тонового различия между областями изображения.

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

Масштабирование объекта – растяжение (сжатие) объекта вдоль соответствующих осей относительно начала координат.

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

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

Пиксель (pixel) – мельчайший элемент изображения, воспроизводимый компьютером. Отличительными особенностями пикселя являются его однородность (все пиксели одина-

ковы по размеру) и неделимость (внутри пикселя не может быть никаких более мелких элементов).

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

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

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

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

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

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

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

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

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

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

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

Трехмерная (3D) графика – построение на компьютере с помощью специальных программ пространственной модели,

состоящей из простых и сложных геометрических форм; присвоение этой модели фактуры, цвета, степени прозрачности и матовости; придание ей и условной камере движения в виртуальном пространстве; расстановка в этом пространстве источников света; и, наконец, просчет выстроенной сцены.

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

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

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

Цвет – набор определенных длин волн, отраженных от предмета или пропущенных сквозь прозрачный предмет.

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

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

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Васильев В.Е. Компьютерная графика: учеб. пособие. /В.Е. Васильев, А.В. Морозов. – СПб.: СЗТУ, 2005. – 101 с.
2. Божко А.Н. Компьютерная графика: учебное пособие / А.Н. Божко, Д.М. Жук, В.Б. Маничев. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2007. – 392 с.
3. Колесниченко О.В. Аппаратные средства РС / О.В. Колесниченко, И.В. Шишигин, В.Г. Соломенчук. – СПб.: БХВ-Петербург, 2010. – 800 с.
4. Бэйн С. Эффективная работа: CorelDRAW 10. /С. Бэйн. – СПб.: Питер, 2002. – 784 с.
5. Ганзбург А. Периферийные устройства /А. Ганзбург. – СПб.: Питер, 2001. – 448 с.
6. Гук М. Аппаратные средства IBM PC Энциклопедия. /М. Гук. – СПб.: Питер, 2000. – 816 с.
7. Казанцев А.В. Основы компьютерной графики: в 2ч. Ч.1. Математический аппарат компьютерной графики (электронная версия) /А.В. Казанцев. – Казань, 2001. – 62 с.
8. Ким Ли. 3D Studio Max 4/5. Для дизайнера. Искусство трехмерной анимации /Ли. Ким. – Киев: ООО «ТИД «ДС», 2002. 848 с.
9. Петров М.Н. Компьютерная графика: Учебник для вузов /М.Н. Петров, В.П. Молочков. – СПб.: Питер, 2003.–736 с.
10. Пономаренко С.И. Adobe Illustrator 10 /С.И. Пономаренко. – СПб.: БХВ-Петербург, 2003. - 688 с.
11. Шушан Р. Дизайн и компьютер /Р. Шушан, Д. Райт, Л. Льюис. – М.: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd», 1997. – 544 с.
12. Яцюк О.Г. Компьютерные технологии в дизайне. Эффективная реклама /О.Г. Яцюк, Э.Т. Романычева. – СПб.: БХВ-Петербург, 2001. – 432 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1. Графическая подсистема ЭВМ	5
1.1. Состав графической подсистемы ЭВМ.....	5
1.2. Основные характеристики и классификация мониторов.....	6
1.3. Принцип работы ЭЛТ монитора.....	7
1.4. Принцип работы ЖК-монитора.....	11
1.5. Графические адаптеры.....	13
1.5.1. Структура графического адаптера.....	13
1.5.2. Технологии SLI и CrossFire.....	17
1.5.3. Графические API.....	17
2. ЦВЕТОВЫЕ МОДЕЛИ.....	19
2.1. Свет и особенности человеческого цветовосприятия.....	19
2.2. Цветовая модель RGB.....	21
2.3. Цветовая модель CMYK.....	23
2.4. Цветовая модель HSB.....	25
3. ЦВЕТОВЫЕ ПАЛИТРЫ.....	28
3.1. Индексная палитра.....	28
3.2. Фиксированная палитра.....	28
3.3. Безопасная палитра.....	29
4. АЛГОРИТМЫ РАСТРОВОЙ ГРАФИКИ	30
4.1. Растровые представления изображений.....	30
4.2. Растр.....	31
4.3. Построение линий в прямоугольном и гексагональном растре.....	32
4.4. Моделирование гексагонального растра.....	35
4.5. Построение линии в квадратном растре.....	37
4.6. Параметрический алгоритм рисования линии... ..	39
4.7. Алгоритм Брезенхема рисования линии.....	41
4.8. Алгоритм Брезенхема генерации окружности... ..	43
4.9. Отсечение по полю вывода.....	46
4.9.1. Алгоритм Козна-Сазерленда.....	48
4.9.2. Алгоритм отсечения по прямоугольной	

	области.....	49
5.	СИСТЕМА КООРДИНАТ И ТИПЫ ПРЕОБРАЗОВАНИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ	52
	5.1. Декартова система координат.....	52
	5.2. Двумерные матричные преобразования.....	56
	5.3. Однородные координаты и матричное представление двумерных преобразований.....	59
	5.4. Трехмерные матричные преобразования.....	67
6.	ОСНОВНЫЕ РЕДАКТОРЫ РАСТРОВОЙ ГРАФИКИ.....	72
	6.1. Редактор GIMP.....	72
	6.2. Редактор Paint.....	73
	6.3. Редактор Adobe Photoshop.....	74
	6.4. Редактор Paint Shop Pro.....	74
	6.5. Редактор Corel PHOTO-PAINT.....	75
	6.6. Редактор Macromedia Fireworks.....	76
7.	ФОРМАТЫ ФАЙЛОВ РАСТРОВОЙ ГРАФИКИ...	77
	7.1. Формат PCX.....	77
	7.2. Формат BMP.....	77
	7.3. Формат GIF.....	78
	7.4. Формат PNG.....	79
	7.5. Формат JPEG.....	79
	7.6. Формат PSD.....	80
	7.7. Формат TIFF.....	80
	7.8. Формат RAW.....	80
	7.9. Формат DjVu.....	81
8.	ВЕКТОРНАЯ ГРАФИКА.....	83
	8.1. Математические основы векторной графики...	84
	8.1.1. Прямая линия.....	85
	8.1.2. Кривая второго порядка.....	85
	8.1.3. Кривая третьего порядка.....	85
	8.1.4. Кривые Безье.....	86
	8.2. Типы опорных точек.....	89
	8.2.1. Точка перегиба.....	89
	8.2.2. Гладкая опорная точка.....	89
	8.2.3. Симметричная опорная точка.....	89

8.2.4. Тангенциальная опорная точка.....	90
9. ОСНОВНЫЕ РЕДАКТОРЫ ВЕКТОРНОЙ ГРАФИКИ.....	92
9.1. Редактор XFig.....	92
9.2. Редактор Adobe Illustrator.....	92
9.3. Редактор CorelDRAW.....	92
9.4. Редактор Macromedia FreeHand.....	93
10. ФОРМАТЫ ФАЙЛОВ ВЕКТОРНОЙ ГРАФИКИ.....	94
10.1. Формат PostScript.....	94
10.2. Формат EPS.....	94
10.3. Формат PDF.....	95
10.4. Формат AI.....	95
10.5. Формат FH.....	96
10.6. Формат CDR.....	96
11. ФРАКТАЛЬНАЯ ГРАФИКА.....	97
11.1. Классификация фракталов.....	98
11.1.1. Геометрические фракталы.....	98
11.1.2. Алгебраические фракталы.....	99
11.1.3. Стохастические фракталы.....	100
12. ТРЕХМЕРНАЯ 3D-ГРАФИКА.....	102
12.1. Типы пространств.....	102
12.2. Моделирование объектов.....	103
12.2.1. Геометрические объекты.....	104
12.2.2. Негеометрические объекты.....	105
12.2.3. Материалы и карты.....	106
12.2.4. Текстурирование материалов.....	106
12.2.5. Дополнительные свойства материалов...	107
12.2.6. Анимация.....	107
12.2.7. Рендеринг.....	109
11.2.7.1. Методы рендеринга.....	109
12.2.7.2. Математическое обоснование рендеринга.....	110
14. НАСТОЛЬНЫЕ ИЗДАТЕЛЬСКИЕ СИСТЕМЫ.....	112
14.1. Аппаратный уровень.....	113
14.1.1. Устройства ввода информации.....	113

14.1.2. Устройства обработки, хранения и передачи информации.....	113
14.1.3. Устройства вывода информации.....	114
14.2. Программный уровень.....	114
14.3. Пользовательский уровень.....	114
ЗАКЛЮЧЕНИЕ.....	115
ГЛОССАРИЙ	116
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	121

Учебное издание

Козенков Олег Дмитриевич
Ожерельев Виктор Вадимович

ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

В авторской редакции

Подписано к изданию 13.06.2013

Объем данных 5,1 Мб.

ФГБОУ ВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14