

Компьютерная графика

Курс лекций

Оглавление

ТЕМА № 1. ВВЕДЕНИЕ В КОМПЬЮТЕРНУЮ ГРАФИКУ	3
Определение и основные задачи компьютерной графики.....	3
История развития компьютерной (машинной) графики.....	5
Области применения компьютерной графики	9
ТЕМА № 2. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНОЙ ГРАФИКИ.....	11
Мониторы, классификация, принцип действия,	11
основные характеристики	11
Видеоадаптер.....	15
Принтеры, их классификация, основные характеристики.....	16
и принцип работы	16
Плоттеры (графопостроители)	18
Сканеры, классификация и основные характеристики	19
Дигитайзеры	20
Манипулятор «мышь»	20
Джойстики	22
Трекбол	22
Тачпад и трекпойнт	22
Средства диалога для систем виртуальной реальности	23
ТЕМА № 3. ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКИХ ДАННЫХ	24
Форматы графических файлов	24
Зрительный аппарат человека	27
Цветовые модели и их виды	29
Закон Грассмана (законы смешивания цветов)	30
Цветовая модель RGB	31
Цветовая модель HSB.....	32
Модель CMY (Cyan Magenta Yellow)	32
Цветовая модель CMYK	33
Цветовая модель Lab	34

Перцепционные цветовые модели	34
Черно-белый и полутоновый режим	35
Плашечные цвета	35
Кодирование цвета. Палитра	35
ТЕМА № 4. ФРАКТАЛЬНАЯ ГРАФИКА	36
Понятие фрактала и история появления фрактальной графики	36
Геометрические фракталы	38
Алгебраические фракталы	40
Стохастические фракталы	41
Системы итерируемых функций (IFS – Iterated Function Systems)	42
Фракталы и хаос	42
ТЕМА № 5. РАСТРОВАЯ ГРАФИКА	43
Растровая графика, общие сведения	43
Растровые представления изображений	44
Виды растров	45
Достоинства и недостатки растровой графики	47
О сжатии растровой графики	47
Геометрические характеристики растра	48
Форматы растровых графических файлов	50
Средства для работы с растровой графикой	51
ТЕМА № 6. ВЕКТОРНАЯ ГРАФИКА	52
Векторная графика, общие сведения	52
Элементы (объекты) векторной графики. Объекты и их атрибуты	53
Цвет в векторной графике	54
Структура векторной иллюстрации.	54
Достоинства векторной графики	55
Векторная графика в Интернете	57
Редактируемая двумерная графика для Интернета	58
Средства для создания векторных изображений	59
Растровая и векторная графика	60
ТЕМА № 7. ТРЕХМЕРНАЯ ГРАФИКА	60
Основные понятия трехмерной графики	60
Программные средства обработки трехмерной графики	62
ТЕМА № 8. БАЗОВЫЕ РАСТРОВЫЕ АЛГОРИТМЫ	63
Алгоритм вывода прямой линии	63
Простой пошаговый алгоритм	64
Алгоритм Брезенхема	64
Алгоритмы растровой графики	77

ТЕМА № 1. ВВЕДЕНИЕ В КОМПЬЮТЕРНУЮ ГРАФИКУ

Определение и основные задачи компьютерной графики. Области применения компьютерной графики. История развития компьютерной графики. Виды компьютерной графики.

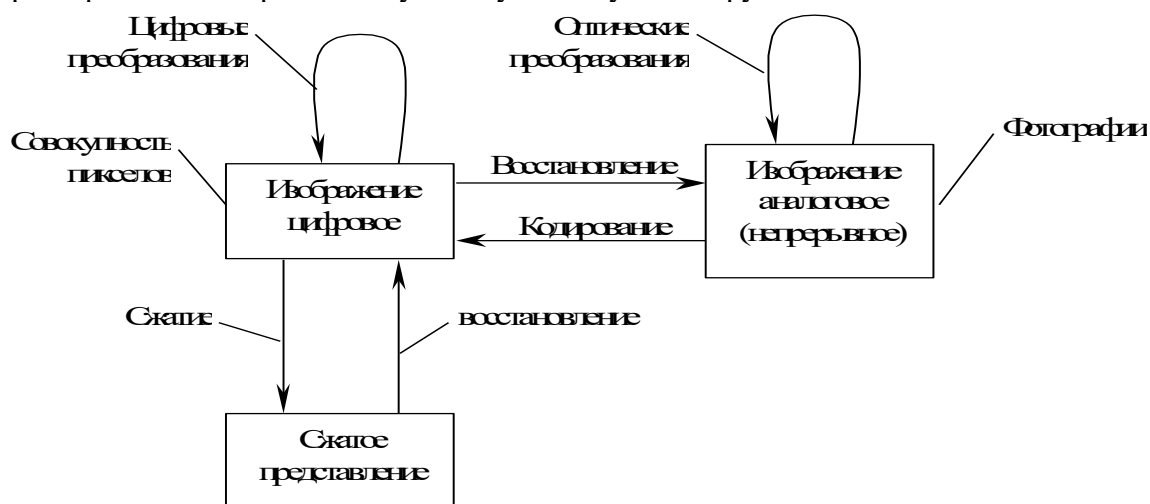
Определение и основные задачи компьютерной графики

При обработке информации, связанной с изображением на мониторе, принято выделять три основных направления: распознавание образов, обработку изображений и машинную графику.

Основная задача распознавания образов состоит в преобразовании уже имеющегося изображения на формально понятный язык символов. **Распознавание образов или система технического зрения (COMPUTER VISION)** – это совокупность методов, позволяющих получить описание изображения, поданного на вход, либо отнести заданное изображение к некоторому классу (так поступают, например, при сортировке почты). Одной из задач COMPUTER VISION является так называемая скелетизация объектов, при которой восстанавливается некая основа объекта, его «скелет».

Обработка изображений (IMAGE PROCESSING) рассматривает задачи в которых и входные и выходные данные являются изображениями. Например, передача изображения с устранением шумов и сжатием данных, переход от одного вида изображения к другому (от цветного к черно-белому) и т.д. Таким образом, под обработкой изображений понимают деятельность над изображениями (преобразование изображений). Задачей обработки изображений может быть как улучшение в зависимости от определенного критерия (реставрация, восстановление), так и специальное преобразование, кардинально изменяющее изображения.

При обработке изображений существуют следующие группы задач:



Ограничимся работой только с цифровым изображением. Цифровые преобразования по цели преобразования можно разделить на два типа:

- **реставрация изображения** – компенсирование имеющегося искажения (например, плохие условия фотосъемки);
- **улучшение изображения** – это искажение изображения с целью улучшения визуального восприятия или для преобразования в форму, удобную для дальнейшей обработки.

Компьютерная (машинная) графика (COMPUTER GRAPHICS) воспроизводит изображение в случае, когда исходной является информация неизобразительной природы. Например, визуализация экспериментальных данных в виде графиков, гистограмм или диаграмм, вывод информации на экран компьютерных игр, синтез сцен на тренажерах.

Компьютерная графика в настоящее время сформировалась как наука об аппаратном и программном обеспечении для разнообразных изображений от простых чертежей до реалистичных образов естественных объектов. Компьютерная графика используется почти во всех научных и инженерных дисциплинах для наглядности и восприятия, передачи информации. Применяется в медицине, рекламном бизнесе, индустрии развлечений и т. д. Без компьютерной графики не обходится ни одна современная программа. Работа над графикой занимает до 90% рабочего времени программистских коллективов, выпускающих программы массового применения.

Конечным продуктом компьютерной графики является изображение. Это изображение может использоваться в различных сферах, например, оно может быть техническим чертежом, иллюстрацией с изображением детали в руководстве по эксплуатации, простой диаграммой, архитектурным видом предполагаемой конструкции или проектным заданием, рекламной иллюстрацией или кадром из мультфильма.

Компьютерная графика – это наука, предметом изучения которой является создание, хранение и обработка моделей и их изображений с помощью ЭВМ, т.е. это раздел информатики, который занимается проблемами получения различных изображений (рисунков, чертежей, мультипликации) на компьютере.

В компьютерной графике рассматриваются следующие задачи:

- представление изображения в компьютерной графике;
- подготовка изображения к визуализации;
- создание изображения;
- осуществление действий с изображением.

Под компьютерной графикой обычно понимают автоматизацию процессов подготовки, преобразования, хранения и воспроизведения графической информации с помощью компьютера. Под графической информацией понимаются модели объектов и их изображения.

В случае, если пользователь может управлять характеристиками объектов, то говорят об интерактивной компьютерной графике, т.е. способность компьютерной системы создавать графику и вести диалог с человеком. В настоящее время почти любую программу можно считать системой интерактивной компьютерной графики.

Интерактивная компьютерная графика – это так же использование компьютеров для подготовки и воспроизведения изображений, но при этом пользователь имеет возможность оперативно вносить изменения в изображение непосредственно в процессе его воспроизведения, т.е. предполагается возможность работы с графикой в режиме диалога в реальном масштабе времени.

Интерактивная графика представляет собой важный раздел компьютерной графики, когда пользователь имеет возможность динамически управлять содержимым изображения, его формой, размером и цветом на поверхности дисплея с помощью интерактивных устройств управления.

Исторически первыми интерактивными системами считаются **системы автоматизированного проектирования (САПР)**, которые появились в 60-х годах. Они представляют собой значительный этап в эволюции компьютеров и программного обеспечения. В системе интерактивной компьютерной графики пользователь воспринимает на дисплее изображение, представляющее некоторый сложный объект, и может вносить изменения в описание (модель) объекта. Такими изменениями могут быть как ввод и редактирование отдельных элементов, так и задание числовых значений для любых параметров, а также иные операции по вводу информации на основе восприятия изображений.

Системы типа САПР активно используются во многих областях, например в машиностроении и электронике. Одними из первых были созданы САПР для проектирования самолетов, автомобилей, системы для разработки микросхем интегральных схем, архитектурные системы. Такие системы на первых порах функционировали на достаточно больших компьютерах. Потом распространилось использование быстродействующих компьютеров среднего класса с развитыми графическими возможностями – графических рабочих станций. С ростом мощностей персональных компьютеров все чаще САПР использовали на дешевых массовых компьютерах, которые сейчас имеют достаточные быстродействие и объемы памяти для решения многих задач. Это привело к широкому распространению систем САПР.

Сейчас становятся все более популярными **геоинформационные системы (ГИС)**. Это относительно новая для массовых пользователей разновидность систем интерактивной компьютерной графики. Они аккумулируют в себе методы и алгоритмы многих наук и информационных технологий. Такие системы используют последние достижения технологий баз данных, в них заложены многие методы и алгоритмы математики, физики, геодезии, топологии, картографии, навигации и, конечно же, компьютерной графики. Системы типа ГИС зачастую требуют значительных мощностей компьютера как в плане работы с базами данных, так и для визуализации объектов, которые находятся на поверхности Земли. Причем, визуализацию необходимо делать с различной степенью детализации – как для Земли в целом, так и в границах отдельных участков. В настоящее время заметно стремление разработчиков ГИС повысить реалистичность изображений пространственных объектов и территорий.

Работа с компьютерной графикой – одно из самых популярных направлений использования персонального компьютера, причем занимаются этой работой не только профессиональные художники и дизайнеры. На любом предприятии время от времени возникает необходимость в подаче рекламных объявлений в

газеты и журналы, в выпуске рекламной листовки или буклета. Иногда предприятия заказывают такую работу специальным дизайнерским бюро или рекламным агентствам, но часто обходятся собственными силами и доступными программными средствами.

Типичными для любой ГИС являются такие операции – ввод и редактирование объектов с учетом их расположения на поверхности Земли, формирование разнообразных цифровых моделей, запись в базы данных, выполнение разнообразных запросов к базам данных. Важной операцией является анализ с учетом пространственных, топологических отношений множества объектов, расположенных на некоторой территории.

История развития компьютерной (машинной) графики

Компьютерная графика насчитывает в своем развитии не более десятка лет, а ее коммерческим приложениям – и того меньше. **Андриес ван Дам** считается одним из отцов компьютерной графики, а его книги – фундаментальными учебниками по всему спектру технологий, положенных в основу машинной графики. Также в этой области известен **Айвэн Сазерленд**, чья докторская диссертация явилась теоретической основой машинной графики.

До недавнего времени экспериментирование по использованию возможностей интерактивной машинной графики было привилегией лишь небольшому количеству специалистов, в основном ученые и инженеры, занимающиеся вопросами автоматизации проектирования, анализа данных и математического моделирования. Теперь же исследование реальных и воображаемых миров через «призму» компьютеров стало доступно гораздо более широкому кругу людей.

Такое изменение ситуации обусловлено несколькими причинами. Прежде всего, в результате резкого улучшения соотношения стоимость / производительность для некоторых компонент аппаратуры компьютеров. Кроме того, стандартное программное обеспечение высокого уровня для графики стало широкодоступным, что упрощает написание новых прикладных программ, переносимых с компьютеров одного типа на другие.

Следующая причина обусловлена влиянием, которое дисплеи оказывают на **качество интерфейса** – средства общения между человеком и машиной, – обеспечивая максимальные удобства для пользователя. Новые, удобные для пользователя системы построены в основном на подходе **WYSIWYG** (аббревиатура от английского выражения «What you see is what you get» – «**Что видите, то и имеете**»), в соответствии с которым изображение на экране должно быть как можно более похожим на то, которое в результате печатается.

Большинство традиционных приложений машинной графики являются двумерными. В последнее время отмечается возрастающий коммерческий интерес к трехмерным приложениям. Он вызван значительным прогрессом в решении двух взаимосвязанных проблем: моделирования трехмерных сцен и построения как можно более реалистичного изображения. Например, в имитаторах полета особое значение придается времени реакции на команды, вводимые пилотом и инструктором. Чтобы создавалась иллюзия плавного движения, имитатор должен порождать чрезвычайно реалистичную картину динамически изменяющегося «мира» с частотой как минимум 30 кадров в секунду. В противоположность этому изображения, применяемые в рекламе и индустрии развлечений, вычисляются автономно, нередко в течение часов, с целью достичь максимального реализма или произвести сильное впечатление.

Развитие компьютерной графики, особенно на ее начальных этапах, в первую очередь связано с развитием технических средств и в особенности дисплеев:

- произвольное сканирование луча;
- растровое сканирование луча;
- запоминающие трубки;
- плазменная панель;
- жидкокристаллические индикаторы;
- электролюминисцентные индикаторы;
- дисплеи с эмиссией полей.

Произвольное сканирование луча. Дисплейная графика появилась, как попытка использовать электроннолучевые трубки (ЭЛТ) с произвольным сканированием луча для вывода изображения из ЭВМ. Как пишет Ньюмен "по-видимому, первой машиной, где ЭЛТ использовалась в качестве устройства вывода была ЭВМ Whirlwind-I (Ураган-I), изготовленная в 1950г. в Массачусетском технологическом институте. С этого эксперимента начался этап развития векторных дисплеев (дисплеев с **произвольным сканированием**

луча, каллиграфических дисплеев). На профессиональном жаргоне вектором называется отрезок прямой. Отсюда и происходит название «**векторный дисплей**».

При перемещении луча по экрану в точке, на которую попал луч, возбуждается свечение люминофора экрана. Это свечение достаточно быстро прекращается при перемещении луча в другую позицию (обычное время послесвечения – менее 0.1 с). Поэтому, для того чтобы изображение было постоянно видимым, приходится его перевыдавать (регенерировать изображение) 50 или 25 раз в секунду. Необходимость перевыдачи изображения требует сохранения его описания в специально выделенной памяти, называемой памятью регенерации. Само описание изображения называется дисплейным файлом. Понятно, что такой дисплей требует достаточно быстрого процессора для обработки дисплейного файла и управления перемещением луча по экрану.

Обычно серийные векторные дисплеи успевали 50 раз в секунду строить только около 3000–4000 отрезков. При большем числе отрезков изображение начинает мерцать, так как отрезки, построенные в начале очередного цикла, полностью погасают к тому моменту, когда будут строиться последние.

Другим недостатком векторных дисплеев является малое число градаций по яркости (обычно 2–4). Были разработаны, но не нашли широкого применения двух–трехцветные ЭЛТ, также обеспечивавшие несколько градаций яркости.

В векторных дисплеях легко стереть любой элемент изображения – достаточно при очередном цикле построения удалить стираемый элемент из дисплейного файла.

Текстовый диалог поддерживается с помощью алфавитно–цифровой клавиатуры. Косвенный графический диалог, как и во всех остальных дисплеях, осуществляется перемещением перекрестия (курсора) по экрану с помощью тех или иных средств управления перекрестием – координатных колес, управляющего рычага (джойстика), трекбола (шаровой рукоятки), планшета и т.д. Отличительной чертой векторных дисплеев является возможность непосредственного графического диалога, заключающаяся в простом указании с помощью светового пера объектов на экране (линий, символов и т.д.). Для этого достаточно с помощью фотодиода определить момент прорисовки и, следовательно, начала свечения люминофора) любой части требуемого элемента.

Первые серийные векторные дисплеи за рубежом появились в конце 60–х годов.

Растровое сканирование луча. Прогресс в технологии микроэлектроники привел к тому, с середины 70–х годов подавляющее распространение получили дисплеи с растровым сканированием луча.

Запоминающие трубки. В конце 60–х годов появилась запоминающая ЭЛТ, которая способна достаточно длительное время (до часа) прямо на экране хранить построенное изображение. Следовательно, не обязательна память регенерации и не нужен быстрый процессор для выполнения регенерации изображения. Стирание на таком дисплее возможно только для всей картинке в целом. Сложность изображения практически не ограничена. Разрешение, достигнутое на дисплеях на запоминающей трубке, такое же, как и на векторных или выше – до 4096 точек.

Текстовый диалог поддерживается с помощью алфавитно–цифровой клавиатуры, косвенный графический диалог осуществляется перемещением перекрестия по экрану обычно с помощью координатных колес.

Появление таких дисплеев с одной стороны способствовало широкому распространению компьютерной графики, с другой стороны представляло собой определенный регресс, так как распространялась сравнительно низкокачественная и низкоскоростная, не слишком интерактивная графика.

Плазменная панель. В 1966г. была изобретена плазменная панель, которую упрощенно можно представить как матрицу из маленьких разноцветных неоновых лампочек, каждая из которых включается независимо и может светиться с регулируемой яркостью. Ясно, что системы отклонения не нужно, не обязательна также и память регенерации, так как по напряжению на лампочке можно всегда определить горит она ли нет, т.е. есть или нет изображение в данной точке. В определенном смысле эти дисплеи объединяют в себе многие полезные свойства векторных и растровых устройств. К недостаткам следует отнести большую стоимость, недостаточно высокое разрешение и большое напряжение питания. В целом эти дисплеи не нашли широкого распространения.

Жидкокристаллические индикаторы. Дисплеи на жидкокристаллических индикаторах работают аналогично индикаторам в электронных часах, но, конечно, изображение состоит не из нескольких сегментов, а из большого числа отдельно управляемых точек. Эти дисплеи имеют наименьшие габариты и энергопотребление, поэтому широко используются в портативных компьютерах несмотря на меньшее разрешение, меньшую контрастность и заметно большую цену, чем для растровых дисплеев на ЭЛТ.

Электролюминисцентные индикаторы. Наиболее высокие яркость, контрастность, рабочий температурный диапазон и прочность имеют дисплеи на электролюминисцентных индикаторах. Благодаря достижениям в технологии они стали доступны для применения не только в дорогих высококласных системах, но и в общепромышленных системах. Работа таких дисплеев основана на свечении люминофора под воздействием относительно высокого переменного напряжения, прикладываемого к взаимноперпендикулярным наборам электродов, между которыми находится люминофор.

Дисплей с эмиссией поля. Дисплеи на электронно–лучевых трубках, несмотря на их относительно низкую дешевизну и широкое распространение, механически непрочны, требуют высокого напряжения питания, потребляют большую мощность, имеют большие габариты и ограниченный срок службы, связанный с потерей эмиссии катодами. Одним из методов устранения указанных недостатков, является создание плоских дисплеев с эмиссией поля с холодных катодов в виде сильно заостренных микроигл.

Таким образом, стартовав в 1950г., компьютерная графика к настоящему времени прошла путь от экзотических экспериментов до одного из важнейших, всепроникающих инструментов современной цивилизации, начиная от научных исследований, автоматизации проектирования и изготовления, бизнеса, медицины, экологии, средств массовой информации, досуга и кончая бытовым оборудованием.

Основные даты:

Дата	Событие	Технологии
1956	Первые эксперименты Бена Лапоски "oscillons" (США, с 1950г) и Герберта Франка (Германия). Эта дата была определена Jasia Reichardt как начало Компьютерного Искусства	
1957	Получено первое цифровое изображение в Национальном Бюро Стандартов (США)	
1958	Сэр Джон Уитни использует аналоговый компьютер для создания анимации (США)	
1959	Выставка "Экспериментальная эстетика" в музее Angewandte Kunst (Вена, Австрия), показ "oscillons" и т.д.	
1963	Проводится первое соревнование по компьютерному искусству, спонсором которого выступил американский журнал Computers and Automation. В 1965 г. Его выигрывает Майкл Нолл (США) и в 1966 Фрайдер Нейк (Германия). Выходит в свет первый созданный на компьютере фильм Эдварда Зайека (Bell labs, США) Чарльз Ксури создает свои первые компьютерные работы (США)	Иван Сазерленд представляет Sketchpad – программу для интерактивной работы с компьютерной графикой на конференции. Работа была начата в 1961г. В Массачусетском технологическом институте.
1965	Первая выставка цифрового искусства в Technische Hochschule в Штутгарте организованная Фрайдером Нейком, Майклом Ноллом и Джорджем Нисом (Германия) Первая выставка цифрового искусства в США в галерее Howard Wise в Нью Йорке. Были выставлены компьютерные работы Бела Джулса и Майкла Нолла (США) Три первые общественные выставки компьютерного искусства: 5–19 февраля, Generative Computergrafik. Georg Nees. Studien–Galerie des Studium Generale, TH Stuttgart. Открыта Максом Бенсом (Германия). 6–24 апреля, изображения созданные с помощью компьютера. Майкл Нолл, Бела Джулс, Howard Wise Gallery, Нью Йорк (США). 5–26 ноября, Computergrafik. Фрайдер Нейк, Джордж Нис. Галерея Wendelin Niedlich, Штутгарт. Открыта Максом Бенсом (Германия)	
1966	IBM присуждает звание Artist–in–Residence Сэру Джону Уитни	
1967	Эксперименты в искусстве и технологии начаты в Нью Йорке группой художников и техников, включая художника Robert Rauschenberg и инженера Billy Kluver (США)	

1968	<p>Cybernetic Serendipity: Выставка компьютерных технологий и искусства, в институте современного искусства, Лондон. Курирует Джеша Ричарт (директор ICA и автор Компьютер в Искусстве).</p> <p>Музей современного искусства приобретает работу Ксури "Hummingbird"</p> <p>Джон Лэнсдаун (архитектор) и Алан Сатклиф (пионер компьютерной музыки) создают Общество Компьютерного Искусства как подразделение Британского Компьютерного Общества</p> <p>Бруклинский музей (США) – эксперименты в цифровом искусстве.</p>	
1969	<p>SIGGRAPH, Special Interest Group on Computer Graphics сформировано с помощью ACM (the Association for Computing Machinery). В 1967г. по инициативе Сэма Матса и Андриас ван Дама (США) организован Special Interest Committee. Основана CTG (Computer Technique Group) (Япония)</p> <p>Generative Computer–Grafik публикуется первая докторская диссертация по компьютерному искусству Generative Computer–Grafik Джорджа Ниса, под руководством Макса Бенса представленную в Университете Штутгарта.(Германия)</p>	
1971	<p>Впервые в мире проводится персональная выставка работ по компьютерному искусству; Манфред Мор, Музей современного искусства, Франция, Париж.</p> <p>Герберт Фрэнк публикует 'Computer Graphics – Computer Art' (Германия)</p>	
1972–1973		Ричард Шуп создает SuperPaint, 8ми битную графическую программу в исследовательском центре Xerox Palo Alto (США)
1974	Фильм "Голод" Питера Фолдса получает Приз Жюри на Каннском фестивале кино за лучшую анимацию (Канада)	
1975	Фракталы – Бенуа Мандельбро (IBM, США)	
1976	Руфь Левитт публикует "Художник и компьютер" (США)	
1979	'Sunstone' анимация Эда Эмшвиллера (NYIT, США)	
1980		Фирма "Quantel" представляет Paintbox (Великобритания)
1983	Гарольд Коэн выставляет работу AARON в Tate gallery (Лондон, Великобритания)	Дэвид Эм работает в лаборатории Jet Propulsion в Калифорнийском Технологическом институте
1984	Питер Перлштайн использует графическую систему в Нью-Йоркском Технологическом институте	Продан первый компьютер Macintosh. Также рекламный ролик получает награду международного фестиваля рекламы фестиваля Clio

1986	Рисование светом – Дэвид Хокни, Говард Ходжкинс, Сэр Сидни Нолан и Ларри Риверс приглашены на ВВС для использования Qantel Paintbox на телевидении. (Великобритания) Энди Вархол использует Amiga для создания своего автопортрета и портрета певицы Деборы Харри (США) фильм "Luxo Jr" Джона Лассетера (фирма PIXAR) показан на конференции Siggraph (США)	Томасс и Джон Нолл, работая на Lucasfilm, пишут 24 бит-ную графическую программу Photoshop
1988	Первый международный симпозиум по электронному искусству в г. Утрехт (Германия) Кливлендская галерея, выставка Искусство и Компьютер, г. Мидлсбороу (Великобритания)	
1989	Выставка "Electronic Print" в музее Arnolfini в г. Бристоль. Под руководством Мартина Райзера. (Великобритания)	Релиз Photoshop для Macintosh (США)
1992	Первый Digital Salon в Нью-Йорке (США)	
1995	Первая конференция по CADE, Брайтон (Великобритания)	
1997	В Лондоне открыта Collville Place Gallery (Великобритания)	
1998	Вольфганг Лайзер основывает Музей Цифрового Искусства	

Области применения компьютерной графики

Область применения компьютерной графики не ограничивается одними художественными эффектами. Во всех отраслях науки, техники, медицины, в коммерческой и управленческой деятельности используются построенные с помощью компьютера схемы, графики, диаграммы, предназначенные для наглядного отображения разнообразной информации. Конструкторы, разрабатывая новые модели автомобилей и самолетов, используют трехмерные графические объекты, чтобы представить окончательный вид изделия. Архитекторы создают на экране монитора объемное изображение здания, и это позволяет им увидеть, как оно впишется в ландшафт.

Можно рассмотреть следующие области применения компьютерной графики.

Научная графика

Первые компьютеры использовались лишь для решения научных и производственных задач. Чтобы лучше понять полученные результаты, производили их графическую обработку, строили графики, диаграммы, чертежи рассчитанных конструкций. Первые графики на машине получали в режиме символьной печати. Затем появились специальные устройства – графопостроители (плоттеры) для вычерчивания чертежей и графиков чернильным пером на бумаге. Современная научная компьютерная графика дает возможность проводить вычислительные эксперименты с наглядным представлением их результатов.

Деловая графика

Деловая графика – область компьютерной графики, предназначенная для наглядного представления различных показателей работы учреждений. Плановые показатели, отчетная документация, статистические сводки – вот объекты, для которых с помощью деловой графики создаются иллюстративные материалы. Программные средства деловой графики включаются в состав электронных таблиц.

Конструкторская графика

Конструкторская графика используется в работе инженеров–конструкторов, архитекторов, изобретателей новой техники. Этот вид компьютерной графики является обязательным элементом САПР (систем автоматизации проектирования). Средствами конструкторской графики можно получать как плоские изображения (проекции, сечения), так и пространственные трехмерные изображения.

Иллюстративная графика

Иллюстративная графика – это произвольное рисование и черчение на экране компьютера. Пакеты иллюстративной графики относятся к прикладному программному обеспечению общего назначения. Простейшие программные средства иллюстративной графики называются графическими редакторами.

Художественная и рекламная графика

Художественная и рекламная графика – ставшая популярной во многом благодаря телевидению. С помощью компьютера создаются рекламные ролики, мультфильмы, компьютерные игры, видеоуроки, видеопрезентации. Графические пакеты для этих целей требуют больших ресурсов компьютера по быстродействию и памяти. Отличительной особенностью этих графических пакетов является возможность создания

реалистических изображений и "движущихся картинок". Получение рисунков трехмерных объектов, их повороты, приближения, удаления, деформации связано с большим объемом вычислений. Передача освещенности объекта в зависимости от положения источника света, от расположения теней, от фактуры поверхности, требует расчетов, учитывающих законы оптики.

Одним из первых известных фильмов был фильм «Звездные войны». Он был создан с помощью суперкомпьютера Сау. Этапы дальнейшего развития компьютерного кинематографа можно проследить по таким фильмам, как «Терминатор-2», «Вавилон 5», и др. До недавнего времени технологии компьютерной графики использовались для спецэффектов, создания изображений экзотических чудовищ, имитации стихийных бедствий и других элементов, которые являлись лишь фоном для игры живых актеров. В 2001 году вышел на экраны полнометражный кинофильм «Финальная фантазия», в котором все, включая изображения людей, синтезировано компьютером – живые актеры только озвучили роли за кадром.

Компьютерная анимация

Компьютерная анимация – это получение движущихся изображений на экране дисплея. Художник создает на экране рисунке начального и конечного положения движущихся объектов, все промежуточные состояния рассчитывает и изображает компьютер, выполняя расчеты, опирающиеся на математическое описание данного вида движения. Полученные рисунки, выводимые последовательно на экран с определенной частотой, создают иллюзию движения. Мультимедиа – это объединение высококачественного изображения на экране компьютера со звуковым сопровождением. Наибольшее распространение системы мультимедиа получили в области обучения, рекламы, развлечений.

Графика для Интернета

Появление глобальной сети Интернет привело к тому, что компьютерная графика стала занимать важное место в ней. Все больше совершенствуются способы передачи визуальной информации, разрабатываются более совершенные графические форматы, ощутимо желание использовать трехмерную графику, анимацию, весь спектр мультимедиа.

Виды компьютерной графики

Различают три вида компьютерной графики. Это растровая графика, векторная графика и фрактальная графика. Они отличаются принципами формирования изображения при отображении на экране монитора или при печати на бумаге.

Растровый метод – изображение представляется в виде набора окрашенных точек. Растровую графику применяют при разработке электронных (мультимедийных) и полиграфических изданий. Иллюстрации, выполненные средствами растровой графики, редко создают вручную с помощью компьютерных программ. Чаще всего для этой цели используют отсканированные иллюстрации, подготовленные художниками, или фотографии. В последнее время для ввода растровых изображений в компьютер нашли широкое применение цифровые фото– и видеокамеры.

Большинство графических редакторов, предназначенных для работы с растровыми иллюстрациями, ориентированы не столько на создание изображений, сколько на их обработку. В Интернете пока применяются только растровые иллюстрации.

Векторный метод – это метод представления изображения в виде совокупности отрезков и дуг и т. д. В данном случае вектор – это набор данных, характеризующих какой-либо объект.

Программные средства для работы с векторной графикой предназначены в первую очередь для создания иллюстраций и в меньшей степени для их обработки. Такие средства широко используют в рекламных агентствах, дизайнерских бюро, редакциях и издательствах. Оформительские работы, основанные на применении шрифтов и простейших геометрических элементов, решаются средствами векторной графики много проще.

Сравнительная характеристика растровой и векторной графики

Критерий сравнения	Растровая графика	Векторная графика
Способ представления изображения	Растровое изображение строится из множества пикселей	Векторное изображение описывается в виде последовательности команд
Представление объектов реального мира	Растровые рисунки эффективно используются для представления реальных образов	Векторная графика не позволяет получать изображения фотографического качества

Качество редактирования изображения	При масштабировании и вращении растровых картинок возникают искажения	Векторные изображения могут быть легко преобразованы без потери качества
Особенности печати изображения	Растровые рисунки могут быть легко напечатаны на принтерах	Векторные рисунки иногда не печатаются или выглядят на бумаге не так, как хотелось бы

Программные средства для работы с **фрактальной графикой** предназначены для автоматической генерации изображений путем математических расчетов. Создание фрактальной художественной композиции состоит не в рисовании или оформлении, а в программировании.

Фрактальная графика, как и векторная – вычисляемая, но отличается от неё тем, что никакие объекты в памяти компьютера не хранятся. Изображение строится по уравнению (или по системе уравнений), поэтому ничего, кроме формулы, хранить не надо.

Изменив коэффициенты в уравнении, можно получить совершенно другую картину. Способность фрактальной графики моделировать образы живой природы вычислительным путем часто используют для автоматической генерации необычных иллюстраций.

ТЕМА № 2. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНОЙ ГРАФИКИ

Устройства вывода графических изображений, их основные характеристики. Мониторы, классификация, принцип действия, основные характеристики. Видеоадаптер. Принтеры, их классификация, основные характеристики и принцип работы. Плоттеры (графопостроители). Устройства ввода графических изображений, их основные характеристики. Сканеры, классификация и основные характеристики. Дигитайзеры. Манипулятор «мышь», назначение, классификация. Джойстики. Трекбол. Тачпады и трекпойнты. Средства диалога для систем виртуальной реальности.

Мониторы, классификация, принцип действия, основные характеристики

Одной из наиболее важных составных частей персонального компьютера является его видеоподсистема, состоящая из монитора и видеоадаптера (обычно размещаемого на системной плате). Монитор предназначен для отображения на экране текстовой и графической информации, визуально воспринимаемой пользователем персонального компьютера. В настоящее время существует большое разнообразие типов мониторов. Их можно охарактеризовать следующими основными признаками:

По режиму отображения мониторы делятся на:

- Растровые дисплеи;
- Векторные дисплеи.

В векторных дисплеях с регенерацией изображения на базе электронно-лучевой трубки (ЭЛТ) используется люминофор с очень коротким временем послесвечения. Такие дисплеи часто называют дисплеями с произвольным сканированием. Из-за того, что время послесвечения люминофора мало, изображение на ЭЛТ за секунду должно многократно перерисовываться или регенерироваться. Минимальная скорость регенерации должна составлять, по крайней мере, 30 (1/с), а предпочтительнее 40–50 (1/с). Скорость регенерации меньшая 30 приводит к мерцанию изображения.

Кроме ЭЛТ, для векторного дисплея необходим дисплейный буфер и дисплейный контроллер. **Дисплейный буфер** – непрерывный участок памяти, содержащий всю информацию, необходимую для вывода изображения на ЭЛТ. Функция **дисплейного контроллера** заключается в том, чтобы циклически обрабатывать эту информацию со скоростью регенерации. Сложность рисунка ограничивается двумя факторами – размером дисплейного буфера и скоростью контроллера.

Растровое устройство можно рассматривать как матрицу дискретных ячеек (точек), каждая из которых может быть подсвечена. Таким образом, оно является точечно-рисующим устройством. Невозможно, за исключением специальных случаев, непосредственно нарисовать отрезок прямой из одной адресуемой точки или пиксела в матрице в другую адресуемую точку. Отрезок можно только аппроксимировать последо-

вательностями точек (пикселей), близко лежащих к реальной траектории отрезка.

Отрезок прямой из точек получится только в случае горизонтальных, вертикальных или расположенных под углом 45 градусов отрезков. Все другие отрезки будут выглядеть как последовательности ступенек. Это явление называется **лестничным эффектом** или «**зазубренностью**».

Чаще всего для графических устройств с растровой ЭЛТ используется буфер кадра. **Буфер кадра** представляет собой большой непрерывный участок памяти компьютера. Для каждой точки или пиксела в растре отводится как минимум один бит памяти. Эта память называется **битовой плоскостью**. Для квадратного раstra размером 512×512 требуется 2^{18} , или 262144 бита памяти в одной битовой плоскости. Из-за того, что бит памяти имеет только два состояния (двоичное 0 или 1), имея одну битовую плоскость, можно получить лишь черно-белое изображение. Битовая плоскость является цифровым устройством, тогда как растровая ЭЛТ – аналоговое устройство. Поэтому при считывании информации из буфера кадра и ее выводе на графическое устройство с растровой ЭЛТ должно происходить преобразование из цифрового представления в аналоговый сигнал. Такое преобразование выполняет цифро-аналоговый преобразователь (ЦАП).

По типу экрана мониторы делятся на:

- Дисплеи на основе ЭЛТ;
- Жидкокристаллические (ЖК);
- Плазменные.

Дисплеи на основе электронно-лучевой трубки

Чтобы понять принципы работы растровых дисплеев и векторных дисплеев с регенерацией, **нужно иметь представление о конструкции ЭЛТ** и методах создания видеоизображения.

На рисунке схематично показана ЭЛТ, используемая в видеомониторах.

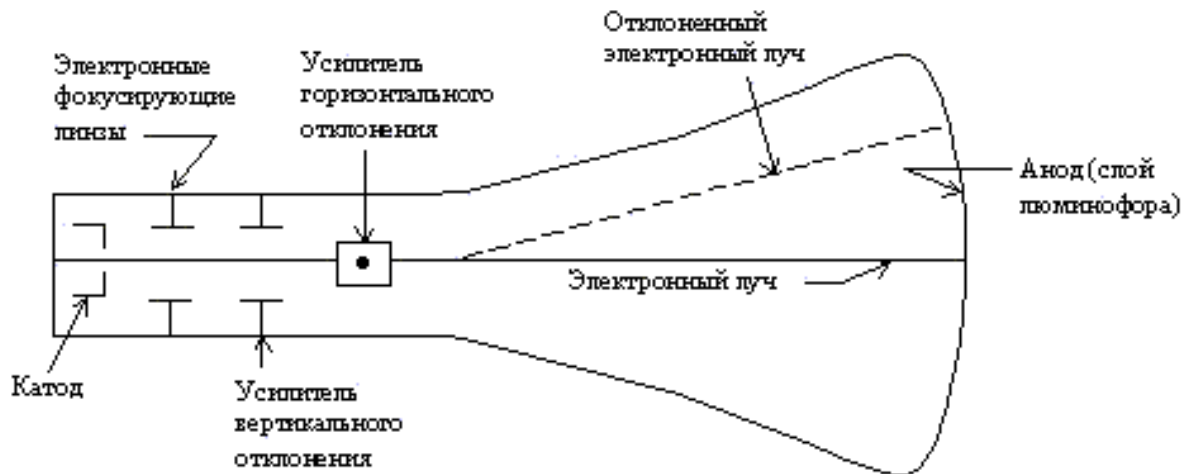


Рис. 7. Электронно-лучевая трубка

Катод (отрицательно заряженный) нагревают до тех пор, пока возбужденные электроны не создадут расширяющегося облака (электроны отталкиваются друг от друга, так как имеют одинаковый заряд). Эти электроны притягиваются к сильно заряженному положительному **аноду**. На внутреннюю сторону расширенного конца ЭЛТ нанесен **люминофор**. Облако электронов с помощью линз фокусируется с узкий, строго параллельный пучок, и луч дает яркое пятно в центре ЭЛТ. Луч отклоняется или позиционируется влево или вправо от центра и (или) выше или ниже центра с помощью усилителей горизонтального и вертикального отклонения. Именно в данный момент проявляется отличие векторных и растровых дисплеев. В векторном дисплее электронный луч может быть отклонен непосредственно из любой произвольной позиции в любую другую произвольную позицию на экране ЭЛТ (анод). Поскольку люминофорное покрытие нанесено на экран ЭЛТ сплошным слоем, в результате получается почти идеальная прямая. В отличие от этого в растровом дисплее луч может отклоняться только в строго определенные позиции на экране, образующие своеобразную мозаику. Эта мозаика составляет видеоизображение. Люминофорное покрытие на экране растровой ЭЛТ тоже не непрерывно, а представляет собой множество тесно расположенных мельчайших точек, куда может позиционироваться луч, образуя мозаику.

Экран жидкокристаллического дисплея (ЖКД) состоит из двух стеклянных пластин, между кото-

рыми находится масса, содержащая жидкие кристаллы, которые изменяют свои оптические свойства в зависимости от прикладываемого электрического заряда. Жидкие кристаллы сами не светятся, поэтому ЖКД нуждаются в подсветке или во внешнем освещении.

Основным достоинством ЖКД являются их габариты (экран плоский). К недостаткам можно отнести недостаточное быстродействие при изменении изображения на экране, что особенно заметно при перемещении курсора мыши, а также зависимость резкости и яркости изображения от угла зрения.

Жидкокристаллические дисплеи

ЖК – дисплеи обладают неоспоримыми преимуществами перед конкурирующими устройствами отображения:

1. **Размеры.** ЖК–дисплеи отличаются малой глубиной и небольшой массой и поэтому их более удобно перемещать и устанавливать, чем ЭЛТ–мониторы, у которых размер в глубину приблизительно равен ширине.

2. **Энергопотребление.** ЖК–дисплей потребляет меньшую мощность, чем ЭЛТ–монитор с сопоставимыми характеристиками.

3. **Удобство для пользователя.** В ЭЛТ электронные лучи при развертке движутся по экрану, обновляя изображение. Хотя в большинстве случаев можно установить такую частоту регенерации (число обновлений экрана электронными лучами в секунду), что изображение выглядит стабильным, некоторые пользователи все же воспринимают мерцание, способное вызвать быстрое утомление глаз и головную боль. На экране ЖК–дисплея каждый пиксел либо включен, либо выключен, так что мерцание отсутствует. Кроме того, для ЭЛТ–мониторов характерно в небольших количествах электромагнитное излучение; в ЖК–мониторах такого излучения нет.

Недостаток – высокая цена

Еще одно достижение, благодаря которому может произойти снижение цен в скором времени, – усовершенствование технологии панелей на **супертвистированных нематических кристаллах** (dual supertwist nematic, DSTN). DSTN–дисплеи всегда были дешевле, чем ЖК–устройства на тонкопленочных транзисторах, но несколько уступали им по качеству: DSTN–дисплеи не обеспечивают такой контрастности и четкости, как матрицы TFT, а их медленная реакция приводит к мерцанию и появлению паразитных (повторных) изображений на экране, особенно при отображении движущихся объектов. Однако фирма Sharp, крупнейший поставщик DSTN–панелей, недавно провела презентацию панели, в которой используется разработанная ею технология HSA (высококонтрастная адресация).

HSA–панели обеспечивают такую же контрастность изображения, как TFT–матрицы, и почти не уступают им по скорости реакции при воспроизведении видео. Фирма Arithmos разработала процессор визуализации для DSTN–панелей, который позволяет еще более улучшить качество изображения. Таким образом, для пользователей, ограниченных в средствах, DSTN–дисплей может оказаться хорошим компромиссным решением.

В ЖК–дисплеях угол обзора не только мал, но и асимметричен: обычно он составляет 45 градусов по горизонтали и +15...–30 по вертикали. Излучающие дисплеи, такие как электролюминесцентные, плазменные и на базе ЭЛТ, как правило, имеют конус обзора от 80 до 90 по обеим осям. Хотя в последнее время на рынке появились модели ЖК–дисплеев с увеличенным углом обзора 50–60 градусов.

Представитель Hitachi Тим Паттон (Tim Patton) считает, что в традиционных ЖК–дисплеях наблюдается зависимость контрастности и цвета изображения от угла зрения. Эта проблема обострялась по мере увеличения размеров ЖК–дисплеев и приобретения ими способности воспроизводить больше цветов.

Hitachi при создании своего нового дисплея SuperTFT воспользовалась иной технологией – IPS. Как известно, в обычных ЖК–дисплеях молекулы жидкого кристалла меняют свою ориентацию с горизонтальной на вертикальную под воздействием электрического поля, а адресующие электроды помещаются на две расположенные друг против друга стеклянные подложки. В IPS(in–plane switching) – дисплеях, наоборот, происходит чередование двух углов в горизонтальной плоскости, причем оба электрода находятся на одной из подложек. В результате угол обзора, как по горизонтальной, так и по вертикальной оси достигает 70 градусов.

Плазменные дисплеи

Газоплазменные мониторы состоят из двух пластин, между которыми находится газовая смесь, светящаяся под воздействием электрических импульсов. Такие мониторы не имеют недостатков, присущих ЖКД, однако их нельзя использовать в переносных компьютерах с аккумуляторным и батарейным питанием, так как они потребляют большой ток.

Размер по диагонали (расстояние от левого нижнего до правого верхнего угла экрана) приводится в дюймах. Наиболее распространены мониторы с диагональю 14". Однако работать с монитором с диагональю 15" намного удобнее, а для работы с графическими пакетами, издательскими системами и системами автоматизированного проектирования необходимы мониторы с диагональю не меньше 17";

– **теневая маска экрана**. Единицей измерения является расстояние между отверстиями маски в мм. Чем меньше это расстояние и чем больше отверстий, тем выше качество изображения. Этот параметр часто отождествляют с **зерном экрана** монитора, однако это справедливо не во всех случаях;

– **разрешение**, измеряется в пикселах (точках), помещающихся по горизонтали и вертикали видимой части экрана. В настоящее время наиболее распространены мониторы с расширением не менее 1024*768 пикселей;

– **кинескоп**. Наиболее предпочтительны следующие типы кинескопов: Black Trinitron, Black Matrix и Black Planar. Данные кинескопы очень контрастны, дают отличное изображение, однако их люминофор чувствителен к свету, что может сократить срок службы монитора. К тому же при работе с контрастным монитором быстрее устают глаза;

– **потребляемая мощность**. У мониторов с диагональю 14" потребляемая мощность не должна превышать 60 Вт, иначе повышается вероятность теплового перегрева монитора, что сокращает срок его службы. У более крупных мониторов потребляемая мощность соответственно выше;

– **антибликовое** покрытие. Для дешевых мониторов используют пескоструйную обработку поверхности экрана. При этом качество изображения ухудшается. В дорогих мониторах на поверхность экрана наносится специальное химическое вещество, обладающее антибликовыми свойствами; – защитные свойства монитора. В настоящее время распространены мониторы с низким уровнем излучения (LR–мониторы). Они отвечают нормам стандарта MPRI или MPR II.

По цветности мониторы делятся на:

- Цветные;
- Монохромные.

Частота кадров (обычно от 50 до 100 Гц).

Все современные аналоговые мониторы условно можно разделить на следующие типы:

- с фиксированной частотой развертки;
- с несколькими фиксированными частотами;
- и многочастотные (мультичастотные).

Мультичастотные мониторы обладают способностью настраиваться на произвольные значения частот синхронизации из некоторого заданного диапазона, например, 30–64 кГц для строчной и 50–100 Гц для кадровой развертки. Разработчиками мониторов данного типа является фирма NEC. В названии таких мониторов присутствует слово Multisync. Эти мониторы относятся к наиболее распространенному типу мониторов с электронно–лучевой трубкой.

Видеодиапазон (обычно от 65 до 200 МГц).

Видеосигнал:

- Цифровой;
- Аналоговый.

Под цифровыми мониторами понимаются устройства отображения зрительной информации на основе электронно–лучевой трубки, управляемой цифровыми схемами. К цифровым относятся монохромные мониторы, снабженные видеоадаптерами стандартов MDA и Hercules, цветные RGB–мониторы, предназначенные для подключения к видеоадаптеру стандарта EGA. Монохромные мониторы способны отображать на экране только темные и светлые точки, иногда точки могут различаться интенсивностью. Hercules–мониторы имеют разрешение до 728*348 пикселей, небольшие габариты и вес. Блок развертки монитора получает синхроимпульсы от соответствующего видеоадаптера. RGB–мониторы способны отображать 16 цветов, однако разрешение экрана у них меньше, чем у Hercules–мониторов.

Электронно–лучевая трубка мониторов данного типа управляется **аналоговыми сигналами** поступающими от видеоадаптера. Принцип работы электронно–лучевой трубки монитора такой же, как у телевизионной трубки. Аналоговые мониторы способны поддерживать разрешение стандарта VGA (640*480) пикселей и выше.

Прочие характеристики: функции управления растром, система энергосбережения, защита от излучения, вес, габариты, потребляемая мощность.

Видеоадаптер

Работой монитора руководит специальная плата, которую называют **видеоадаптером (видеокартой)**. Вместе с монитором видеокарта создает видеоподсистему персонального компьютера. В первых компьютерах видеокарты не было.

Видеоадаптер имеет вид отдельной платы расширения, которую вставляют в определенный слот материнской платы (в современных ПК это слот AGP). **Видеоадаптер** выполняет функции видеоконтроллера, видеопроцессора и видеопамяти.

Сформированное графическое изображение хранится во внутренней памяти **видеоадаптера**, которая называется **видеопамятью**. Необходимая емкость видеопамяти зависит от заданной разрешающей способности и палитры цветов, поэтому для работы в режимах с высокой разрешающей способностью и полноцветной гаммой нужно как можно больше видеопамяти. Если еще недавно типичными были видеоадаптеры с 2-4 Мбайт видеопамяти, то уже сегодня нормальной считается емкость в 32-64 Мбайт. Большинство современных видеокарт обладает возможностью расширения объема видеопамяти до 128 Мбайт. Видеопамять, как правило, строится на микросхемах динамической памяти с произвольным доступом (DRAM), обладающих большим объемом. Видеопамять доступна процессору как обычная оперативная память.

Основные характеристики:

- режим работы (текстовый и графический);
- воспроизведение цветов (монохромный и цветной);
- число цветов или число полутонов (в монохромном);
- разрешающая способность (число адресуемых на экране монитора пикселей по горизонтали и по вертикале);
- разрядность шины данных, определяющая скорость обмена данными с системной шиной и т.д.

Важнейшей характеристикой является емкость видеопамяти, она определяет количество хранимых в памяти пикселей и их атрибутов.

В зависимости от количества поддерживаемых цветовых оттенков различают следующие режимы работы видеоадаптеров:

- 16 цветов;
- 256 цветов;
- High Color (16 бит);
- True Color (24 бит);
- True Color (32 бит).

Первоначально IBM PC выпускались с черно-белым адаптером MDA (Monochrome Display Adapter). Максимальное разрешение составляло 640×350 точек, графические возможности отсутствовали. Следующим был также черно-белый видеоадаптер «Геркулес», выпущенный фирмой Hercules Computer Technology, Inc. Этот адаптер обеспечивает разрешение 720×350 с графическими возможностями.

Первым цветным видеоадаптером фирмы IBM стал CGA (Color Graphics Adapter). Разрешение его мало (320×200), цветов мало (до 4). Максимальное разрешение составляет 640×200.

Затем был выпущен видеоадаптер EGA – (Enhanced Graphics Adapter). Он обеспечивает разрешение 640×200 при 16 цветах из 64. Максимальное разрешение 640×350.

Первым видеоадаптером со сравнительно приемлемыми характеристиками стал VGA (Video Graphics Array) с максимальным разрешением до 800×600 при 256 цветах.

Затем фирма IBM разработала видеоадаптер 8514/A, имевшего параметры более соответствующие сложившимся потребностям – 1024×768 при 256 цветах.

Последняя разработка фирмы IBM – видеоадаптер XGA (eXtended Graphics Array) с не самыми современными возможностями 1024×768 при 256 цветах. Он на уровне регистров совместим с VGA.

Многие фирмы выпускают улучшенные версии VGA под названиями Super VGA и Ultra VGA, но общий стандарт отсутствует.

Фирма Texas Instruments предложила стандарт на программный интерфейс с интеллектуальными видеоадаптерами, использующими графические процессоры TMS 340xx (TIGA–стандарт, Texas Instruments Graphics Architecture). В настоящее это самые мощные видеоадаптеры для IBM PC.

Таким образом, существуют следующие видеоконтроллеры:

- Hercules – монохромный графический адаптер;
- MDA – монохромный дисплейный адаптер;
- MGA – монохромный графический адаптер;

тер;

- CGA – цветной графический адаптер;
- EGA – улучшенный графический адаптер;
- VGA – видеографический адаптер (видеографическая матрица);
- SVGA – улучшенный видеографический адаптер;
- PGA – профессиональный графический адаптер;

Для повышения быстродействия графических подсистем IBM PC выпускаются специальные типы адаптеров – графические акселераторы. Графические акселераторы содержат собственные процессоры, которые специализированы для выполнения графических преобразований, поэтому изображения обрабатываются быстрее, чем с использованием универсального ЦП ПЭВМ.

Акселераторы, кроме типа и возможностей графического процессора, различаются по следующим основным параметрам:

- памятью для сохранения изображений. В некоторых случаях используется обычная динамическая память DRAM, но обычно используется специализированная видеопамять VRAM;
- используемой шиной. В настоящее время обычно используется PCI;
- шириной регистров. Чем шире регистр, тем большее число пикселей можно обработать за одну команду. В настоящее время ширина – 64 бита.

Принтеры, их классификация, основные характеристики и принцип работы

Принтеры являются наиболее популярными устройствами вывода информации для персональных компьютеров.

По технологии печати принтеры можно разделить на:

- игольчатые (матричные);
- струйные;
- лазерные.

Матричные принтеры

Матричные принтеры до последнего времени являлись основным стандартным устройством вывода для персональных компьютеров, поскольку струйные принтеры работали еще неудовлетворительно, а цена лазерных была достаточно высока. И в настоящее время игольчатые принтеры применяются достаточно часто.

Достоинства матричных (игольчатых) принтеров:

- удовлетворительная скорость печати;
- универсальность, заключающаяся в способности работать с любой бумагой;
- низкая стоимость печати.

Игольчатый принтер формирует знаки несколькими иглами, расположенными в головке принтера. Бумага втягивается с помощью вала, а между бумагой и головкой принтера располагается красящая лента. При ударе иголки по этой ленте на бумаге остается закрашенный след. Иголки, расположенные внутри головки, обычно активизируются электромагнитным методом. Головка движется по горизонтальной направляющей и управляется шаговым двигателем. Так как напечатанные знаки внешне представляют собой матрицу, а воспроизводит эту матрицу игольчатый принтер, то часто его называют **матричным принтером**. Среди матричных принтеров существуют 9–игольчатые и 24–игольчатые.

В головке **9–игольчатого** принтера находятся 9 иглоков, которые, как правило, располагаются вертикально в один ряд. Благодаря горизонтальному движению головки принтера и активизации отдельных иглоков напечатанный знак образует как бы матрицу, причем отдельные буквы, цифры и знаки «заложены» внутри принтера в виде бинарных кодов.

В **24–игольчатом** принтере используется технология последовательного расположения иглоков в два ряда по 12 штук. Вследствие того, что иглоки в соседних рядах сдвинуты по вертикали, точки на распечатке перекрываются таким образом, что их невозможно различить. Имеется возможность перемещения головки дважды по одной и той же строке, чтобы знаки пропечатывались еще раз с небольшим смещением. Такое качество печати обозначают как **LQ (Letter Quality – машинописное качество)**, в этом режиме скорость печати уменьшается незначительно, так как головка печатает при движении слева направо и справа налево. Изготовители обычно указывают теоретическую скорость печати, то есть максимально возможную скорость

чернового режима, при этом качество печати не играет роли. LQ–печать для игольчатых принтеров длится дольше. Еще дольше печатается графика, потому что при этом набор знаков не читается из внутренней памяти (ROM) принтера, а каждая печатаемая точка должна рассчитываться.

Единица измерения скорости печати – **cps (символ в секунду)**. Игольчатые принтеры оборудованы внутренней памятью (буфером) до 64 Кбайт и более, который принимает данные от персонального компьютера. Игольчатый принтер – механическое устройство, а работа механических узлов всегда сопровождается шумом.

Качество печати сильно зависит от **разрешения принтера**, т.е. количества точек, которое печатается на одном дюйме – **dpi**. Данная характеристика играет роль, в основном, при работе принтера в графическом режиме.

Струйные принтеры

Принцип работы **струйных принтеров** напоминает игольчатые принтеры. Вместо иглонок здесь применяются тонкие **сопла**, которые находятся в головке принтера. В этой головке установлен резервуар с жидкими чернилами, которые через сопла как микрочастицы переносятся на материал носителя. Число сопел находится в диапазоне от 16 до 64, а иногда и до нескольких сотен.

Для хранения чернил используются два метода:

- головка принтера объединена с резервуаром для чернил; замена резервуара с чернилами одновременно связана с заменой головки;
- используется отдельный резервуар, который через систему капилляров обеспечивает чернилами головки принтера.

В основе принципа действия струйных принтеров лежат:

- пьезоэлектрический метод;
- метод газовых пузырей.

Для реализации **пьезоэлектрического метода** в каждое сопло установлен плоский пьезокристалл, связанный с диафрагмой. Под воздействием электрического тока происходит деформация пьезоэлемента. При печати, находящийся в трубке пьезоэлемент, сжимая и разжимая трубку, наполняет капиллярную систему чернилами. Чернила, которые отжимаются назад, перетекают обратно в резервуар, а чернила, которые выдавлились наружу, образуют на бумаге точки. Струйные принтеры с использованием данной технологии выпускают фирмы Epson, Brother и др.

Метод газовых пузырей базируется на термической технологии. Каждое сопло оборудовано нагревательным элементом, который, при пропускании через него тока, за несколько микросекунд нагревается до температуры около 500 градусов. Возникающие при резком нагревании газовые пузыри стараются вытолкнуть через выходное отверстие сопла порцию (каплю) жидких чернил, которые переносятся на бумагу. При отключении тока нагревательный элемент остывает, паровой пузырь уменьшается, и через входное отверстие поступает новая порция чернил. Данная технология используется в изделиях фирм Hewlett–Packard и Canon.

Цветные струйные принтеры имеют более высокое качество печати по сравнению с игольчатыми цветными принтерами и невысокую стоимость по сравнению с лазерными. **Цветное изображение** получается за счет использования (наложения друг на друга) четырех основных цветов. Уровень шума струйных принтеров значительно ниже, чем у игольчатых, поскольку его источником является только двигатель, управляющий перемещением печатающей головки. При черновой печати скорость струйного принтера значительно выше, чем у игольчатого. При печати с качеством LQ скорость составляет 3–4 (до 10) страницы в минуту. **Качество печати зависит от количества сопел в печатающей головке** – чем их больше, тем выше качество. **Большое значение имеет качество и толщина бумаги**. Выпускается специальная бумага для струйных принтеров, но можно печатать на обычной бумаге плотностью от 60 до 135 г/кв.м. В некоторых моделях для быстрого высыхания чернил применяется подогрев бумаги. Разрешение струйных принтеров при печати графики составляет от 300*300 до 720*720 dpi.

Основной недостаток струйного принтера: возможность засыхания чернил внутри сопла, что приводит к необходимости замены печатающей головки.

Лазерные принтеры

Лазерные принтеры обеспечивают более высокое качество печати по сравнению со струйными и игольчатыми принтерами. Однако стоимость печати выше, особенно при использовании цветных лазерных принтеров. Таким образом, для получения высококачественной черно–белой печати целесообразно использовать лазерный принтер, а для получения цветного изображения можно использовать цветной струйный

принтер. В лазерных принтерах используется механизм печати, применяемый в ксероксах.

Основным элементом является вращающийся барабан для переноса изображения на бумагу, представляющий собой металлический цилиндр, покрытый тонкой пленкой фотопроводящего полупроводника. По поверхности барабана равномерно распределяется статический заряд. Для этого служит тонкая проволока или сетка – **коронирующий провод**. Высокое напряжение, подаваемое на этот провод вызывает возникновение вокруг него светящейся ионизированной области – короны. Лазер, управляемый микроконтроллером, генерирует тонкий световой луч, отражающийся от вращающегося зеркала. Этот луч, падая на барабан, изменяет его электрический заряд в точке падения. Таким образом, на барабане возникает скрытая копия изображения. Далее на барабан наносится **тонер** – мельчайшая красящая пыль. Под действием статического заряда эти мелкие частицы притягиваются к поверхности барабана в точках, подвергшихся экспозиции, и формируют изображение. Бумага втягивается с подающего лотка и с помощью системы валиков перемещается к барабану. Перед барабаном бумаге сообщается статический заряд. Бумага соприкасается с барабаном и притягивает, благодаря своему заряду, частички тонера от барабана. Для фиксации тонера бумага вновь заряжается и пропускается между двумя роликами с температурой 180 градусов. Затем барабан разряжается, очищается от прилипших частиц и готов для нового процесса печати. Фирма OKI выпускает лазерный принтер, в котором вместо лазера используется неподвижная диодная строка, описывающая не каждую точку, а целую строку.

В **цветном лазерном принтере** изображение формируется на **светочувствительной фотоприемной ленте** последовательно для каждого из 4-х основных цветов. Лист печатается за четыре прохода: имеются четыре емкости для тонеров и от двух до четырех узлов проявления. **Схема управления** включает процессор, память большого объема и иногда, особенно при функционировании в сети, винчестер. Стоимость цветного лазерного принтера значительно выше, чем черно-белого, а скорость печати – ниже.

Лазерные принтеры со средними возможностями печатают 4–8 страниц в минуту. Высокопроизводительные сетевые лазерные принтеры могут печатать до 20 и более страниц в минуту. При печати сложных графических изображений время печати больше.

Разрешение по вертикали (соответствует шагу барабана) составляет от 1/300 до 1/600 дюйма. Разрешение по горизонтали определяется точностью наведения лазерного луча и количеством точек в строке и составляет, как правило, от 1/300 до 1/1200 дюйма.

Лазерный принтер обрабатывает целые странички, что связано с большим количеством вычислений. Минимальный объем памяти лазерного принтера не менее 1 Мбайт. Наиболее часто используется память от 2 до 4 Мбайт. Цветные принтеры требуют для работы еще большую память. Память лазерного принтера может быть увеличена путем установки специальных карт с DRAM или SIMM модулями. Большинство лазерных принтеров могут печатать на бумаге формата А4, реже – А3. Некоторые принтеры могут печатать на обеих сторонах листа, но они стоят существенно дороже.

Плоттеры (графопостроители)

Плоттер (графопостроитель) – это устройство автоматического построения диаграмм или других изображений на бумаге, пластике, фоточувствительном материале или ином носителе путем черчения, гравирования, фоторегистрации или иным способом.

Различают:

- **планшетные графопостроители** для формата А3 – А2, с фиксацией листа электростатическим способом и пишущим узлом, перемещающимся в двух координатах (на плоскости);
- **барабанные графопостроители** с носителем, закрепляемым на вращающемся барабане;
- **рулонные или роликовые графопостроители** с чертежной головкой, перемещающейся в одном направлении при одновременном перемещении носителя в перпендикулярном ему направлении. Ширина бумаги формата А1 или А0. Такие плоттеры используют рулоны бумаги длиной до нескольких десятков метров.

Плоттеры изготавливаются в **напольном и настольном** исполнении.

По принципу построения изображения подразделяются на:

- **векторные графопостроители** (Создают изображение с помощью шариковых, перьевых рапидографов, фломастера, карандаша. В настоящее время практически сняты с производства);
- **растровые графопостроители** (Наследуя конструктивные особенности принтеров, создают изображение путем строчного воспроизведения).

По способу печати растровые графопостроители подразделяются на:

- *электростатические* графопостроители с электростатическим принципом воспроизведения;
- *струйные* графопостроители, основанными на принципе струйной печати (выдавливании красящего вещества через сопла форсунок);
- *лазерные* графопостроители, воспроизводящие изображение с использованием луча лазера;
- *светодиодные* графопостроители, отличающиеся от лазерных способом перенесения изображения с барабана на бумагу;
- *термические* графопостроители;
- *микрофильм–плоттеры, фотоплоттеры* с фиксацией изображения на светочувствительном материале.

Основные конструктивные и эксплуатационные характеристики графопостроителей: формат воспроизводимого изображения–оригинала, размер рабочего поля, точность, разрешение растровых графопостроителей (обычно в пределах 300–2500 dpi), скорость прорисовки или изготовления единицы продукции заданного формата, наличие или отсутствие собственной памяти (буфера), программное обеспечение (драйверы, программы растеризации), наличие сетевой платы. Некоторые модели графопостроителей комплектуются или могут оснащаться насадками, дополняющими их функциями сканера.

Большинство струйных аппаратов обеспечивают печать графических файлов формата TIFF, BMP, PCX.

Сканеры, классификация и основные характеристики

Сканер это устройство ввода в персональный компьютер цветного и черно–белого изображения с бумаги, пленки и т.п.

Принцип действия сканера заключается в преобразовании оптического сигнала, получаемого при сканировании изображения световым лучом, в электрический, а затем в цифровой код, который передается в компьютер.

Сканеры разделяют на:

- **черно–белые сканеры** могут в простейшем случае различать только два значения – черное и белое, что вполне достаточно для чтения штрихового кода (более сложные сканеры различают градации серого цвета);
- **цветные сканеры** работают на принципе сложения цветов, при котором цветное изображение получается путем смешения трех цветов: красного, зеленого и синего. Технически это реализуется двумя способами:
 - при сканировании цветной оригинал освещается не белым светом, а последовательно красным, зеленым и синим. Сканирование осуществляется для каждого цвета отдельно, полученная информация предварительно обрабатывается и передается в компьютер;
 - в процессе сканирования цветной оригинал освещается белым светом, а отраженный свет попадает на CCD–матрицу через систему специальных фильтров, разлагающих его на три компонента: красный, зеленый, синий, каждый из которых улавливается своим набором фотоэлементов.

А также сканеры делятся на:

- **Ручные сканеры** – это относительно недорогие устройства небольшого размера, удобны для оперативного сканирования изображений из книг и журналов. Ширина полосы сканирования обычно не превышает 105 мм, стандартное разрешение 300–400 dpi. К недостаткам ручного сканера можно отнести зависимость качества сканирования от навыков пользователя и невозможность одновременного сканирования относительно больших изображений.
- В **барабанном сканере** сканируемый оригинал располагается на вращающемся барабане. В настоящее время используются только в типографском производстве.
- В **листовых сканерах** носитель с изображением протягивается вдоль линейки, на которой расположены CCD–элементы. Ширина изображения как правило составляет формат А4, а длина ограничена возможностями используемого компьютера (чем больше изображение, тем больше размер файла, где хранится его цифровая копия).
- **Планшетные сканеры** осуществляют сканирование в автоматическом режиме. Оригинал располагается в сканере на стеклянном листе, под которым головка чтения с CCD–элементами сканирует изображение построчно с равномерной скоростью. Размеры сканируемых изображений зависят от размера сканера и могут достигать размеров большого чертежного листа (А0). Специальная слайд–приставка позволяет сканировать слайды и негативные пленки. Аппаратное разрешение планшетных сканеров достигает 1200 dpi.

Сканеры подключаются к персональному компьютеру через **специальный контроллер** (для планшетных сканеров это чаще всего SCSI контроллер). Сканер всегда должен иметь соответствующий драйвер, так как только ограниченное число программных приложений имеет встроенные драйверы для общения с определенным классом сканеров.

Дигитайзеры

Дигитайзер предназначен для профессиональных графических работ. С помощью специального программного обеспечения он позволяет преобразовывать движение руки оператора в формат векторной графики. Первоначально дигитайзер был разработан для приложений систем автоматизированного проектирования, так как в этом случае необходимо определять и задавать точное значение координат большого количества точек. В отличие от мыши дигитайзер способен точно определять и обрабатывать абсолютные координаты.

Дигитайзер состоит из специального планшета являющегося рабочей поверхностью и, кроме этого, выполняющего разнообразные функции управления соответствующим программным обеспечением, и светового пера или, чаще, кругового курсора, являющихся устройствами ввода информации.

Одной из разновидностей дигитайзера является **графический** или **рисовальный планшет**. Он представляет собой панель, под которой расположена электромагнитная решетка. Если провести по его поверхности специальным пером, то на экране монитора появится штрих. В планшете реализован принцип абсолютного позиционирования: изображение, нарисованное в левом нижнем углу планшета, появится в левом нижнем углу экрана монитора. Обычно рисовальные планшеты имеют размеры коврика для мыши, но рабочая поверхность несколько меньше.

Имеются планшеты, обладающие чувствительностью к нажиму, с помощью которых, регулируя нажим, можно получать на экране линии различной толщины.

Специальная пластмассовая пленка, прилагаемая к планшету, позволяет копировать подложенные под нее изображения на бумажных оригиналах. Планшеты подключаются к последовательному порту персонального компьютера.

Графический планшет может иметь различные форматы: от A2 – для профессиональной деятельности и меньше – для более простых работ.

Компания Immersion разработала уникальную механическую технологию оцифровки, которая компактна, доступна и легка в использовании. Каждое соединение использует цифровые оптические датчики, работа которых не зависит от любого относящегося к окружению влияния. Результат – универсальная система, которая может работать практически в любой среде и сканировать объекты из любого материала. Но кроме этого, есть и другие технологии трехмерного сканирования:

- Ультразвуковое сканирование.
- Магнитное сканирование.
- Лазерные сканеры.

Из всех трехмерных технологий сканирования, **ультразвуковые системы** наименее точны, наименее надежны и наиболее восприимчивы к геометрическим искажениям. Вследствие того, что скорость звука зависит от воздушного давления, температуры и других атмосферных условий, эффективность ультразвуковых систем может изменяться вместе с погодой. Кроме того, они восприимчивы к работе различного оборудования, даже шуму ламп дневного света.

Магнитные трехмерные цифровые преобразователи работают на том же принципе, что и «ультразвуковые системы», т.е. используют магнитное поле. Они невосприимчивы к атмосферным изменениям и очень чувствительны к искажениям от близлежащего металла или магнитных полей. Металлические стулья, платы, компьютеры или другое оборудование, размещенные близко от магнитного цифрового преобразователя, исказит данные. Кроме того, такие системы нельзя использовать для оцифровки объектов с металлическими частями.

Лазерные сканеры в 10–100 раз дороже, чем системы механической оцифровки, такие, как MicroScribe–3D. Системы, использующие лазеры, имеют много ограничений. Объекты с отражающими или яркими поверхностями, большие объекты и объекты с вогнутыми поверхностями, которые затеняют прямой путь лазерного луча – главная проблема для лазерных систем.

Манипулятор «мышь»

Наряду с клавиатурой мышь является важнейшим средством ввода информации. В современных программах, имеющих сложную графическую оболочку, мышь является основным инструментом управления программой.

По принципу действия мыши делятся на:

- *Механические;*
- *Оптико–механические;*
- *Оптические.*

Подавляющее число компьютерных мышек используют **оптико–механический принцип** кодирования перемещения. С поверхностью стола соприкасается тяжелый, покрытый резиной шарик сравнительно большого диаметра. Ролики, прижатые к поверхности шарика, установлены на перпендикулярных друг другу осях с двумя датчиками. Датчики, представляющие собой **оптопары** (светодиод–фотодиод), располагаются по разные стороны дисков с прорезями. Порядок, в котором освещаются фоточувствительные элементы, определяет направление перемещения мыши, а частота приходящих от них импульсов – скорость. Хороший механический контакт с поверхностью обеспечивает специальный коврик.

Более точного позиционирования курсора позволяет добиться **оптическая мышь**. Для нее используется специальный коврик, на поверхности которого нанесена мельчайшая сетка из перпендикулярных друг другу темных и светлых полос. Расположенные в нижней части мыши две оптопары освещают коврик и по числу пересеченных при движении линий определяют величину и скорость перемещения. Оптические мыши не имеют движущихся частей и лишены такого присущего оптико–механическим мышам недостатка, как перемещение курсора мыши рывками из–за загрязнения шарика. Разрешающая способность применяемого в мыши устройства считывания координат составляет 400 dpi (Dot per Inch точек на дюйм) и выше, превосходя аналогичные значения для механических устройств.

Для оптимального функционирования мышь должна передвигаться по ровной поверхности. Лучше всего подходят специальные коврики. Указатель мыши передвигается по экрану синхронно с движением мыши по коврику. Устройством ввода мыши являются кнопки (клавиши). Большинство мышей имеют две кнопки, существуют также 3–х кнопочные мыши и имеющие большее количество кнопок.

Одной из важных характеристик мыши является ее разрешение, которое измеряется в dpi. Разрешение определяет минимальное перемещение, которое способен почувствовать контроллер мыши. Чем больше разрешение, тем точнее позиционируется мышь, тем с более мелкими объектами можно работать. Нормальное разрешение мыши лежит в диапазоне от 300 до 900 dpi. В усовершенствованных мышах используют переменный баллистический эффект скорости, заключающийся в том, что при небольших перемещениях скорость смещения курсора – небольшая, а при значительных перемещениях – существенно увеличивается. Это позволяет эффективнее работать в графических пакетах, где приходится обрабатывать мелкие детали.

По принципу передачи информации мыши делятся на:

- *Последовательные* (Serial Mouse), подключаемые к последовательному порту COM;
- *Параллельные* (Bus Mouse), использующие системную шину. Bus Mouse подключается к специальной карте расширения, входящей в комплект поставки мыши.

Параллельные мыши предпочтительнее в тех системах, где к компьютеру требуется подключить много периферийных устройств, особенно занимающих последовательные порты, и где компьютер подвержен конфликтам прерываний периферийных устройств (Bus Mouse не использует прерывания).

Существует несколько стандартов **последовательных мышей**. Самым распространенным является стандарт MS–Mouse. Альтернативными стандартами являются PC–Mouse, используемый для трехкнопочных мышей фирмы Genius, и редко используемый PS/2. MS–Mouse и совместимые с ней PC–Mouse для работы требуют установки соответствующих драйверов. Большинство программного обеспечения для персональных компьютеров ориентировано на MS–Mouse. Стандарт PS/2 не требует подключения драйверов.

К основным тенденциям развития современных мышей можно отнести постепенный переход на шину USB, а также поиски в области эргономических усовершенствований. К ним можно отнести беспроводные (Cordless) мыши, работающие в радио– или инфракрасном диапазоне волн, а также мыши с дополнительными кнопками. Наиболее удачными решениями являются наличие между двумя стандартными кнопками колесика (мышь Microsoft IntelliMouse) или качающейся средней кнопки (мыши Genius NetMouse NetMouse Pro), которые используются для быстрой прокрутки документа под Windows 95/98/NT.

К наиболее известным производителям мышей относятся компании Genius, Logitech, Microsoft, Mitsumi и др.

Джойстики

Джойстик является координатным устройством ввода информации и наиболее часто применяется в области компьютерных игр и компьютерных тренажеров.

Джойстики бывают:

- Аналоговые (обычно используются в компьютерных тренажерах);
- Цифровые (в игровых компьютерах).

Аналоговые джойстики обеспечивают более точное управление, что очень важно для программных приложений, в которых объекты должны точно позиционироваться. Для удобства работы конструкция джойстика должна быть достаточно прочной и устойчивой. Джойстик подключают к внешнему разъему карты расширения, имеющей соответствующий порт.

Для того чтобы подключить джойстик к компьютеру, нужен игровой порт. **Игровой порт (или адаптер)** может быть расположен на плате асинхронного последовательного адаптера, на плате мультипорта или на отдельной плате. Иногда игровой порт может быть расположен и на системной плате компьютера.

Джойстик подключается к компьютеру через игровой порт. К одному игровому порту может быть подключено два джойстика. Процедура подключения джойстика весьма проста. Все что нужно – это вставить разъем на конце шнура джойстика в разъем игрового порта. Этот разъем внешне напоминает разъем последовательного порта, но имеет 15 выводов.

Работа с джойстиком не требует подключения дополнительного драйвера. Достаточно настроить вашу игровую программу на его использование. Заметим, что далеко не все игры могут работать с джойстиком. Обычно это эмуляторы полета на самолетах, вертолетах и космических кораблях.

Трекбол

Трекбол (Trackball) – это устройство ввода информации, которое можно представить в виде перевернутой мыши с шариком большого размера. Принцип действия и способ передачи данных трекбола такой же, как и мыши. Наиболее часто используется оптико–механический принцип регистрации положения шарика. Подключение трекбола, как правило, осуществляется через последовательный порт.

Основные отличия от мыши:

- стабильность положения за счет неподвижного корпуса;
- не нужна площадка для движения, так как позиция курсора рассчитывается по вращению шарика.

Первое устройство подобного типа было разработано компанией Logitech. Миниатюрные трекболы получили сначала широкое распространение в портативных ПК. Встроенные трекболы могут располагаться в самых различных местах корпуса ноутбука, внешние крепятся специальным зажимом, а к интерфейсу подключаются кабелем. Большого распространения в ноутбуках трекболы не получили из-за своего недостатка постепенного загрязнения поверхности шара и направляющих роликов, которые бывает трудно очистить и, следовательно, вернуть трекболу былую точность. Впоследствии их заменили тачпады и трекпойнты.

Тачпад и трекпойнт

Трекпойнт (TrackPoint) – координатное устройство, впервые появившееся в ноутбуках IBM, представляет собой миниатюрный джойстик с шершавой вершиной диаметром 5–8 мм. Трекпойнт расположен на клавиатуре между клавишами и управляется нажатием пальца.

Тачпад (TouchPad) представляет собой чувствительную контактную площадку, движение пальца по которой вызывает перемещение курсора. В подавляющем большинстве современных ноутбуков применяется именно это указательное устройство, имеющее не самое высокое разрешение, но обладающее самой высокой надежностью из-за отсутствия движущихся частей.

TouchPad поддерживает следующие протоколы: PS/2; RS-232; ADB – протокол, используемый компьютерами семейства Apple Macintosh.

В каждом из этих случаев TouchPad поддерживает индустриальный стандарт «mouse» плюс собственные, специфические, расширенные протоколы. Поддержка «mouse» означает, что, подключив к компьютеру TouchPad, вы сразу можете использовать ее как обычную «мышку», без инсталляции ее собственного драйвера. После этого вы инсталлируете драйвер и получаете целый набор дополнительных возможностей.

Дальнейшим развитием TouchPad является TouchWriter – панель TouchPad с повышенной чувствительностью, одинаково хорошо работающая как с пальцем, так и со специальной ручкой и даже с ногтем. Эта панель позволяет вводить данные привычным для человека образом – записывая их ручкой. Кроме того, ее можно использовать для создания графических изображений или для подписывания документов. Для желающих писать китайскими иероглифами, можно порекомендовать установить на компьютер пакет QuickStroke, который позволит вводить иероглифы, непосредственно рисуя их на панели. Причем программа, по мере ввода, предлагает готовые варианты иероглифов.

Оба эти устройства предполагают наличие определенной тренировки для обращения с ними, однако по надежности и малогабаритности остаются вне конкуренции.

Средства диалога для систем виртуальной реальности

В системах виртуальной реальности, в отличие от обычных приложений компьютерной графики, как правило, требуется вывод и ввод трехмерной координатной информации, как для управления положениями синтезируемых объектов, так и для определения координат частей тела оператора и направления его взгляда.

Спейсбол. Одним из первых появилось устройство спейсбол (space ball), представляющее собой конструктивное объединение мышки и небольшого трекбола. Мышка перемещается оператором по столу и обеспечивает ввод двух координат. Ввод третьей координаты обеспечивается вращением шарика трекбола большим пальцем руки.

Для манипулирования объектами в трехмерном пространстве часто используется техника виртуальной сферы. Управляемый объект окружается (воображаемой) сферой. Для перемещения сферы используется мышь, а вращение сферы и заключенного в нее объекта обеспечивается вращением шарика трекбола.

Head Mounted Display. В системах виртуальной реальности используются устройства вывода в виде монтируемых на голове дисплеев (Head Mounted Display – HMD) с бинокулярным всенаправленным монитором (Binocular Omni-Orientation Monitor – BOOM) со средствами отслеживания положения головы (head tracking) и даже отслеживанием положения глаза (eye tracking). Это требуется для создания эффекта «погружения» со стереоскопическим изображением и оперативным изменением сцены при поворотах головы и/или глаз.

Используемые в HMD жидкокристаллические дисплеи обычно невысокого разрешения (до 417×277 пикселей). Сравните это с 1280×1024 и 1600×1200 для настольных систем или с разрешением монитором для телевидения высокой четкости (ТВВЧ) – 1920×1035 и 1920×1135. Поэтому ведутся интенсивные исследования по созданию средств отображения для систем виртуальной реальности, обладающих высоким разрешением при приемлемых значениях электромагнитных наводок. Одна из таких систем, использующих миниатюрные монохромные прецизионные электронные трубки и жидкокристаллические затворы, обеспечивает разрешение до 2000×2000. Интересное решение заключается в формировании изображения лазером непосредственно на сетчатке, но эти предложения пока далеки от коммерческой реализации.

Отслеживание положения головы обеспечивается либо механическими рычажными системами, либо комплектом инфракрасных или электромагнитных датчиков.

Power Glove, Data Glove, Data Suit. Непосредственный ввод геометрической информации о положении частей тела с поддержкой тактильной и даже силовой обратной связи обеспечивается перчатками и костюмами данных.

Дешевая перчатка данных – Power Glove, используемая для игр, обеспечивает только четыре уровня данных.

В более усовершенствованной перчатке данных Data Glove фирмы VPL (см. рис.) для определения углов сгибания пальцев используются оптические волокна. Для обеспечения тактильной обратной связи используются пневматические активаторы.

Были эксперименты обеспечения тактильной обратной связи за счет вибрации пьезокристаллов.

Более точный ввод координатной информации обеспечивают системы с использованием механического рычажного экзоскелета руки (Exos Dexterous Handmaster) и датчиками углов сгибания пальцев на основе эффекта Холла. Системы с экзоскелетом позволяют обеспечить и силовую обратную связь.

Более простой прибор, использующий силовую обратную связь, был разработан фирмой Digital и представляет собой рукоятку, подобную рукоятке газа в мотоцикле, которая может менять свое сопротивление скручиванию.

Проблема в обеспечении тактильной и силовой обратной связи состоит в том, что пользователь реа-

гирует на воздействия и вносит изменения быстрее, чем система сможет среагировать. Для хорошего ощущения объекта система тактильной обратной связи должна обеспечивать скорость порядка 100–300 Гц, что почти на порядок выше обычной скорости перезаписи визуальной информации.

Пиджак данных (Date Suit) по принципам работы подобен перчатке данных и отличается только количеством датчиков.



Рис. Перчатка данных фирмы VPL

ТЕМА № 3. ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКИХ ДАННЫХ

Форматы графических файлов. Понятие цвета. Зрительный аппарат человека, для восприятия цвета. Аддитивные и субтрактивные цвета в компьютерной графике. Понятие цветовой модели и режима. Закон Грассмана. Пиксельная глубина цвета. Черно-белый режим. Полутоновый режим. Виды цветовых моделей (RGB, CMYK, HSB, Lab), их достоинства и недостатки. Кодирование цвета.

Форматы графических файлов

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

TIFF (Tagged Image File Format). Формат предназначен для хранения растровых изображений высокого качества (расширение имени файла .TIF). На сегодняшний день является одним из самых распространенных и надежных, его поддерживают практически все программы. TIFF является лучшим выбором при импорте растровой графики в векторные программы и издательские системы. Предусматривает широкий диапазон цветового охвата – от монохромного черно-белого до модели *СМУК*.

PSD (PhotoShop Document). Собственный формат программы Adobe Photoshop (расширение имени файла .PSD), один из наиболее мощных по возможностям хранения растровой графической информации. Позволяет запоминать параметры слоев, каналов, степени прозрачности, множества масок. Поддерживаются 48-разрядное кодирование цвета, цветоделение и различные цветовые модели. Основной недостаток выражен в том, что отсутствие эффективного алгоритма сжатия информации приводит к большому объему файлов.

PCX. Формат появился как формат хранения растровых данных программы PC PaintBrush фирмы Z-Soft и является одним из наиболее распространенных (расширение имени файла .PCX). Отсутствие возможности хранить цветоделенные изображения, недостаточность цветовых моделей и другие ограничения привели к утрате популярности формата. В настоящее время считается устаревшим.

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

Формат файла JPEG (Объединенная экспертная группа по фотографии, произносится «джейпег») был разработан компанией C-Cube Microsystems как эффективный метод хранения изображений с большой глубиной цвета, например, получаемых при сканировании фотографий с многочисленными едва уловимыми оттенками цвета. Самое большое отличие формата JPEG от других форматов состоит в том, что в JPEG используется алгоритм сжатия с потерями информации. Алгоритм сжатия без потерь так сохраняет информацию об изображении, что распакованное изображение в точности соответствует оригиналу. При сжатии с потерями приносится в жертву часть информации об изображении, чтобы достичь большего коэффициента сжатия. Распакованное изображение JPEG редко соответствует оригиналу абсолютно точно, но очень часто эти различия столь незначительны, что их едва можно обнаружить.

Процесс сжатия изображения JPEG достаточно сложен и часто для достижения приемлемой производительности требует специальной аппаратуры. Вначале изображение разбивается на квадратные блоки со стороной размером 8 пикселей. Затем производится сжатие каждого блока отдельно за три шага.

Коэффициент архивации в JPEG может изменяться в пределах от 2 до 200 раз. Широкое применение JPEG сдерживается тем, что он оперирует 24-битными изображениями. Поэтому для того, чтобы с приемлемым качеством посмотреть картинку на обычном мониторе в 256-цветной палитре, требуется применение соответствующих алгоритмов и, следовательно, определенное время. Кроме того, если имеющиеся у вас изображения, допустим, в 8-битном формате GIF перевести в 24-битный JPEG, а потом обратно в GIF для просмотра, то потеря качества произойдет дважды при обоих преобразованиях.

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

PNG (Portable Network Graphics). Сравнительно новый (1995 год) формат хранения изображений для их публикации в Интернете (расширение имени файла .PNG). Поддерживаются три типа изображений – цветные с глубиной 8 или 24 бита и черно-белое с градацией 256 оттенков серого. Сжатие информации происходит практически без потерь, предусмотрены 254 уровня альфа-канала, чересстрочная развертка.

WMF (Windows MetaFile). Формат хранения векторных изображений операционной системы Windows (расширение имени файла .WMF). По определению поддерживается всеми приложениями этой системы. Однако отсутствие средств для работы со стандартизированными цветовыми палитрами, принятыми в полиграфии, и другие недостатки ограничивают его применение (WMF искажает цвет, не может сохранять ряд параметров, которые могут быть присвоены объектам в различных векторных редакторах).

EPS (Encapsulated PostScript). Формат описания как векторных, так и растровых изображений на языке PostScript фирмы Adobe, фактическом стандарте в области донепечатных процессов и полиграфии

(расширение имени файла .EPS). Так как язык PostScript является универсальным, в файле могут одновременно храниться векторная и растровая графика, шрифты, контуры обтравки (маски), параметры калибровки оборудования, цветовые профили. Для отображения на экране векторного содержимого используется формат *WMF*, а растрового – *TIFF*. Но экранная копия лишь в общих чертах отображает реальное изображение, что является существенным недостатком *EPS*. Действительное изображение можно увидеть лишь на выходе выводного устройства, с помощью специальных программ просмотра или после преобразования файла в формат PDF в приложениях Acrobat Reader, Acrobat Exchange.

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

BMP (Windows Device Independent Bitmap). Формат BMP является родным форматом Windows, он поддерживается всеми графическими редакторами, работающими под ее управлением. Применяется для хранения растровых изображений, предназначенных для использования в Windows и, по сути, больше ни на что не пригоден. Способен хранить как индексированный (до 256 цветов), так и RGB-цвет.

CDR (CorelDRAW Document). Формат известен в прошлом низкой устойчивостью и плохой совместимостью файлов, тем не менее, пользоваться CorelDRAW чрезвычайно удобно.

Понятие цвета и его характеристики

Мы смотрим на предметы и, характеризуя их, говорим примерно следующее: он большой, мягкий, светло-голубого цвета. При описании чего-либо в большинстве случаев упоминается цвет, так как он несет огромное количество информации. На самом деле тело не имеет определенного цвета. Само понятие цвета тесно связано с тем, как человек (человеческий взгляд) воспринимает свет; можно сказать, что цвет зарождается в глазу.

Цвет – чрезвычайно сложная проблема, как для физики, так и для физиологии, т.к. он имеет как психофизиологическую, так и физическую природу. Восприятие цвета зависит от физических свойств света, т.е. электромагнитной энергии, от его взаимодействия с физическими веществами, а также от их интерпретации зрительной системой человека. Другими словами, цвет предмета зависит не только от самого предмета, но также и от источника света, освещающего предмет, и от системы человеческого видения. Более того, одни предметы отражают свет (доска, бумага), а другие его пропускают (стекло, вода). Если поверхность, которая отражает только синий свет, освещается красным светом, она будет казаться черной. Аналогично, если источник зеленого света рассматривать через стекло, пропускающее только красный свет, он тоже покажется черным.

Самым простым является **ахроматический цвет**, т.е. такой, какой мы видим на экране черно-белого телевизора. При этом белыми выглядят объекты, ахроматически отражающие более 80% света белого источника, а черными – менее 3%. Единственным атрибутом такого цвета является интенсивность или количество. С интенсивностью можно сопоставить скалярную величину, определяя черное, как 0, а белое как 1.

Если воспринимаемый свет содержит длины волн в произвольных неравных количествах, то он называется **хроматическим**.

При субъективном описании такого цвета обычно используют три величины: цветовой тон, насыщенность и светлота. Цветовой тон позволяет различать цвета, такие как красный, зеленый, желтый и т.д. (это основная цветовая характеристика). **Насыщенность** характеризует чистоту, т.е. степень ослабления (разбавления, осветления) данного цвета белым светом, и позволяет отличать розовый цвет от красного, изумрудный от ярко-зеленого и т.д. Другими словами, по насыщенности судят о том, насколько мягким или резким кажется цвет. **Светлота** отражает представление об интенсивности, как о факторе, не зависящем от цветового тона и насыщенности (интенсивность (мощность) цвета).

Обычно встречаются не чистые монохроматические цвета, а их смеси. В основе трехкомпонентной теории света лежит предположение о том, что в центральной части сетчатки глаза находятся три типа чувствительных к цвету колбочек. Первый воспринимает зеленый цвет, второй – красный, а третий – синий цвет. Относительная чувствительность глаза максимальна для зеленого цвета и минимальна для синего. Если на все три типа колбочек воздействует одинаковый уровень энергетической яркости, то свет кажется бе-

лым. Ощущение белого цвета можно получить, смешивая любые три цвета, если ни один из них не является линейной комбинацией двух других. Такие цвета называют основными.

Человеческий глаз способен различать около 350 000 различных цветов. Это число получено в результате многочисленных опытов. Четко различимы примерно 128 цветовых тонов. Если меняется только насыщенность, то зрительная система способна выделить уже не так много цветов: мы можем различить от 16 (для желтого) до 23 (для красного и фиолетового) таких цветов.

Таким образом, для характеристики цвета используются следующие атрибуты:

- **Цветовой тон.** Можно определить преобладающей длиной волны в спектре излучения. Цветовой тон позволяет отличать один цвет от другого – например, зеленый от красного, желтого и других.
- **Яркость.** Определяется энергией, интенсивностью светового излучения. Выражает количество воспринимаемого света.
- **Насыщенность** или чистота тона. Выражается долей присутствия белого цвета. В идеально чистом цвете примесь белого отсутствует. Если, например, к чистому красному цвету добавить в определенной пропорции белый цвет, то получится светлый бледно-красный цвет.

Указанные три атрибута позволяют описать все цвета и оттенки. То, что атрибутов именно три, является одним из проявлений трехмерных свойств цвета.

Большинство людей различают цвета, а те, кто занимается компьютерной графикой, должны четко чувствовать разницу не только в цветах, но и в тончайших оттенках. Это очень важно, так как именно цвет несет в себе большое количество информации, которая ничуть не уступает в важности ни форме, ни массе, ни другим параметрам, определяющим каждое тело.

Факторы, влияющие на внешний вид конкретного цвета:

- источник света;
- информация об окружающих предметах;
- ваши глаза;

Правильно подобранные цвета могут, как привлечь внимание к желаемому изображению, так и оттолкнуть от него. Это объясняется тем, что в зависимости от того, какой цвет видит человек, у него возникают различные эмоции, которые подсознательно формируют первое впечатление от видимого объекта.

Цвет в компьютерной графике нужен для того, что:

- он несет в себе определенную информацию об объектах. Например, летом деревья зеленые, осенью – желтые. На черно-белой фотографии определить пору года практически невозможно, если на это не указывают какие-либо другие дополнительные факты.
- цвет необходим также для того, чтобы различать объекты.
- с его помощью можно вывести одни части изображения на первый план, другие же увести в фон, то есть акцентировать внимание на важном – композиционном – центре.
- без увеличения размера при помощи цвета можно передать некоторые детали изображения.
- в двумерной графике, а именно таковую мы видим на мониторе, так как он не обладает третьим измерением, именно при помощи цвета, точнее оттенков, имитируется (передается) объем.
- цвет используется для привлечения внимания зрителя, создания красочного и интересного изображения.

Любое компьютерное изображение характеризуется, кроме геометрических размеров и разрешения (количество точек на один дюйм), максимальным числом цветов, которые могут быть в нем использованы. Максимальное количество цветов, которое может быть использовано в изображении данного типа, называется **глубиной цвета**. Кроме полноцветных, существуют типы изображений с различной глубиной цвета – черно-белые штриховые, в оттенках серого, с индексированным цветом. Некоторые типы изображений имеют одинаковую глубину цвета, но различаются по цветовой модели.

Зрительный аппарат человека

Системы отображения графической информации воздействуют на зрительный аппарат человека, поэтому с необходимостью должны учитывать как физические, так и психофизиологические особенности зрения.

На рис. показан поперечный размер глазного яблока человека.

Свет попадает в глаз через роговицу и фокусируется хрусталиком на внутренний слой глаза, называемый **сетчаткой**.

Сетчатка глаза содержит два принципиально различных типа фоторецепторов – палочки, обладающие широкой спектральной кривой чувствительности, вследствие чего они не различают длин волн и, следовательно, цвета, и колбочки, характеризующиеся узкими спектральными кривыми и поэтому обладающие цветовой чувствительностью.

Колбочек существует три типа отличающихся фоточувствительным пигментом. Колбочки обычно называют "синими", "зелеными" и "красными" в соответствии с наименованием цвета, для которого они оптимально чувствительны.

Выдаваемое колбочкой значение является результатом интегрирования спектральной функции с весовой функцией чувствительности.

Светочувствительные клетки, известные как колбочки и палочки, формируют слой клеток в задней части сетчатки.

Колбочки и палочки содержат зрительные пигменты. Зрительные пигменты очень похожи на любые другие пигменты, в том, что они поглощают свет и степень поглощения зависит от длины волны. Важное свойство зрительных пигментов состоит в том, что когда зрительный пигмент поглощает фотон света, то изменяется форма молекулы и в то же самое время происходит переизлучение света. Пигмент при этом изменился, измененная молекула поглощает свет менее хорошо чем прежде, т.е. как часто говорят, "отбеливается". Изменение формы молекулы и переизлучение энергии некоторым, пока еще не вполне ясным образом, инициируют светочувствительную клетку к выдаче сигнала.



Рис. Поперечный разрез глаза

Светочувствительные клетки, известные как колбочки и палочки, формируют слой клеток в задней части сетчатки.

Колбочки и палочки содержат зрительные пигменты. Зрительные пигменты очень похожи на любые другие пигменты, в том, что они поглощают свет и степень поглощения зависит от длины волны. Важное свойство зрительных пигментов состоит в том, что когда зрительный пигмент поглощает фотон света, то изменяется форма молекулы и в то же самое время происходит переизлучение света. Пигмент при этом изменился, измененная молекула поглощает свет менее хорошо чем прежде, т.е. как часто говорят, "отбеливается". Изменение формы молекулы и переизлучение энергии некоторым, пока еще не вполне ясным образом, инициируют светочувствительную клетку к выдаче сигнала.

Информация от светочувствительных рецепторов (колбочек и палочек) передается другим типам

клеток, которые соединены между собой. Специальные клетки передают информацию в зрительный нерв. Таким образом волокно зрительного нерва обслуживает несколько светочувствительных рецепторов, т.е. некоторая предварительная обработка изображения выполняется непосредственно в глазу, который по сути представляет собой выдвинутую вперед часть мозга.

Область сетчатки, в которой волокна зрительного нерва собираются вместе и выходят из глаза, лишена светочувствительных рецепторов и называется слепым пятном.

Таким образом, свет должен вначале пройти два слоя клеток, прежде чем он воздействует на колбочки и палочки. Причины для такого обратного устройства сетчатки не полностью поняты, но одно из объяснений состоит в том, что расположение светочувствительных клеток в задней части сетчатки позволяет любому паразитному непоглощенному свету попасть на клетки находящиеся непосредственно позади сетчатки, которые содержат черный пигмент – **меланин**. Клетки, содержащие меланин, также помогают химически восстанавливать светочувствительный визуальный пигмент в колбочках и палочках после того, как они были отбелены на свету.

Интересно отметить, что природа создала целый ряд конструкций глаза. При этом глаза у всех позвоночных похожи на глаза человека, а глаза у беспозвоночных либо сложные (фасеточные) как у насекомых, либо недоразвитые в виде светочувствительного пятна. Только у осьминогов глаза устроены как у позвоночных, но светочувствительные клетки находятся непосредственно на внутренней поверхности глазного яблока, а не как у нас позади других слоев, занимающихся предварительной обработкой изображения. Поэтому, возможно, особого смысла в обратном расположении клеток в сетчатке нет. А это просто один из экспериментов природы.

Цветовые модели и их виды

Наука о цвете – это довольно сложная и широкомасштабная наука, поэтому в ней время от времени создаются различные цветовые модели, применяемые в той либо иной области. Одной из таких моделей и является **цветовой круг**.

Многим известно о том, что существует 3 первичные цвета, которые невозможно получить и которые образуют все остальные. **Основные цвета** – это желтый, красный и синий. При смешивании желтого с красным получается оранжевый, синего с желтым – зеленый, а красного с синим – фиолетовый. Таким образом, можно составить круг, который будет содержать все цвета. Он представлен на рис. и называется **большим кругом Освальда**.

Наряду с кругом Освальда есть еще и **круг Гете**, в котором основные цвета расположены в углах равностороннего треугольника, а дополнительные – в углах перевернутого треугольника.

Друг напротив друга расположены контрастные цвета.

Для описания излучаемого и отраженного цвета используются разные математические модели – **цветовые модели** (цветовое пространство), т.е. – это способ описания цвета с помощью количественных характеристик. Цветовые модели могут быть **аппаратно-зависимыми** (их пока большинство, RGB и CMYK в их числе) и **аппаратно-независимыми** (модель Lab). В большинстве «современных» визуализационных пакетов (например, в Photoshop) можно преобразовывать изображение из одной цветовой модели в другую.

В цветовой модели (пространстве) каждому цвету можно поставить в соответствие строго определенную точку. В этом случае цветовая модель – это просто упрощенное геометрическое представление, основанное на системе координатных осей и принятого масштаба.

Основные цветовые модели:

- RGB;
- CMY (Cyan Magenta Yellow);

- CMYK (Cyan Magenta Yellow Key, причем Key означает черный цвет);
- HSB;
- Lab;
- HSV (Hue, Saturation, Value);
- HLS (Hue, Lightness, Saturation);
- и другие.

В цифровых технологиях используются, как минимум четыре, основных модели: RGB, CMYK, HSB в различных вариантах и Lab. В полиграфии используются также многочисленные библиотеки плашечных цветов.

Цвета одной модели являются дополнительными к цветам другой модели. **Дополнительный цвет** – цвет, дополняющий данный до белого. Дополнительный для красного – голубой (зеленый+синий), дополнительный для зеленого – пурпурный (красный+синий), дополнительный для синего – желтый (красный+зеленый) и т.д.

По принципу действия перечисленные цветовые модели можно условно разить на три класса:

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

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

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

Для получения новых цветов с помощью аддитивного синтеза можно использовать и различные комбинации из двух основных цветов, варьирование состава которых приводит к изменению результирующего цвета.

Таким образом, цветовые модели (цветовое пространство) представляют средства для концептуального и количественного описания цвета. Цветовой режим – это способ реализации определенной цветовой модели в рамках конкретной графической программы.

Закон Грассмана (законы смешивания цветов)

В большинстве цветовых моделей для описания цвета используется трехмерная система координат. Она образует цветовое пространство, в котором цвет можно представить в виде точки с тремя координатами. Для оперирования цветом в трехмерном пространстве Т. Грассман вывел три закона (1853г):

1. Цвет трехмерен – для его описания необходимы три компоненты. Любые четыре цвета находятся в линейной зависимости, хотя существует неограниченное число линейно независимых совокупностей из трех цветов.

Иными словами, для любого заданного цвета можно записать такое цветовое уравнение, выражающее линейную зависимость цветов.

Первый закон можно трактовать и в более широком смысле, а именно, в смысле трехмерности цвета. Необязательно для описания цвета применять смесь других цветов, можно использовать и другие величины – но их обязательно должно быть три.

2. Если в смеси трех цветовых компонент одна меняется непрерывно, в то время, как две другие остаются постоянными, цвет смеси также изменяется непрерывно.

3. Цвет смеси зависит только от цветов смешиваемых компонент и не зависит от их спектральных составов.

Смысл третьего закона становится более понятным, если учесть, что один и тот же цвет (в том числе и цвет смешиваемых компонент) может быть получен различными способами. Например, смешиваемая компонента может быть получена, в свою очередь, смешиванием других компонент.

Цветовая модель RGB

Это одна из наиболее распространенных и часто используемых моделей. Она применяется в приборах, излучающих свет, таких, например, как мониторы, прожекторы, фильтры и другие подобные устройства.

Данная цветовая модель базируется на трех основных цветах: Red – красном, Green – зеленом и Blue – синем. Каждая из вышеперечисленных составляющих может варьироваться в пределах от 0 до 255, образуя разные цвета и обеспечивая, таким образом, доступ ко всем 16 миллионам (полное количество цветов, представляемых этой моделью равно $256 \cdot 256 \cdot 256 = 16\,777\,216$).

Эта модель **аддитивная**. Слово аддитивная (сложение) подчеркивает, что цвет получается при сложении точек трех базовых цветов, каждая своей яркости. Яркость каждого базового цвета может принимать значения от 0 до 255 (256 значений), таким образом, модель позволяет кодировать 256^3 или около 16,7 млн цветов. Эти тройки базовых точек (светящиеся точки) расположены очень близко друг к другу, так что каждая тройка сливается для нас в большую точку определенного цвета. Чем ярче цветная точка (красная, зеленая, синяя), тем большее количество этого цвета добавится к результирующей (тройной) точке.

При работе с графическим редактором Adobe PhotoShop можно выбирать цвет, полагаясь не только на то, что мы видим, но при необходимости указывать и цифровое значение, тем самым иногда, особенно при цветокоррекции, контролируя процесс работы.

Таблица

Значения некоторых цветов в модели RGB

Цвет	R	G	B
Красный (red)	255	0	0
Зеленый (green)	0	255	0
Синий (blue)	0	0	255
Фуксин (magenta)	255	0	255
Голубой (cyan)	0	255	255
Желтый (yellow)	255	255	0
Белый (white)	255	255	255
Черный (black)	0	0	0

Данная цветовая модель считается **аддитивной**, то есть при увеличении яркости отдельных составляющих будет увеличиваться и яркость результирующего цвета: если смешать все три цвета с максимальной интенсивностью, то результатом будет белый цвет; напротив, при отсутствии всех цветов получается черный.

Модель является аппаратно–зависимой, так как значения базовых цветов (а также точка белого) определяются качеством примененного в мониторе люминофора. В результате на разных мониторах одно и то же изображение выглядит неодинаково.



Рис. Модель RGB

Система координат RGB – куб с началом отсчета (0,0,0), соответствующим черному цвету (см. рис. [0.4.1](#)). Максимальное значение RGB – (1,1,1) соответствует белому цвету.

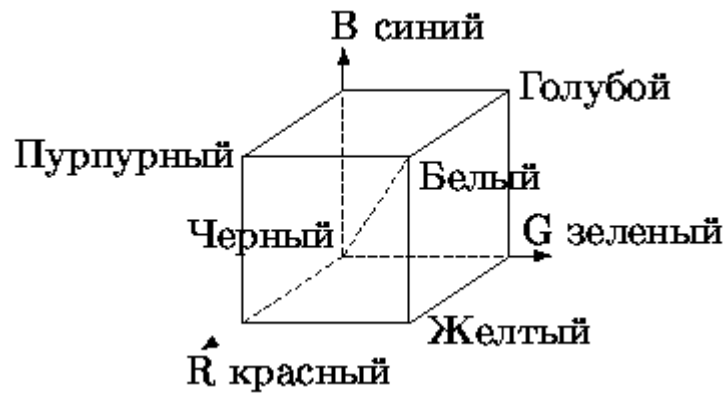


Рис. Цветовой куб модели RGB

Несомненными **достоинствами** данного режима является то, что он позволяет работать со всеми 16 миллионами цветов, а **недостаток** состоит в том, что при выводе изображения на печать часть из этих цветов теряется, в основном самые яркие и насыщенные, также возникает проблема с синими цветами.

Модель RGB – это аддитивная цветовая модель, которая используется в устройствах, работающих со световыми потоками: сканеры, мониторы.

Цветовая модель HSB

Здесь заглавные буквы не соответствуют никаким цветам, а символизируют тон (цвет), насыщенность и яркость (Hue Saturation Brightness). Предложена в 1978 году. Все цвета располагаются по кругу, и каждому соответствует свой градус, то есть всего насчитывается 360 вариантов – **H** определяет частоту света и принимает значение от 0 до 360 градусов (красный – 0, желтый – 60, зеленый – 120 градусов и так далее), т.е. любой цвет в ней определяется своим цветом (тоном), насыщенностью (то есть добавлением к нему белой краски) и яркостью.

Насыщенность определяет, насколько ярко выраженным будет выбранный цвет. 0 – серый, 100 – самый яркий и чистый из возможных вариантов.

Параметр яркости соответствует общепризнанному, то есть 0 – это черный цвет.

Такая цветовая модель намного беднее рассмотренной ранее RGB, так как позволяет работать всего лишь с 3 миллионами цветов.

Эта модель аппаратно-зависимая и не соответствует восприятию человеческого глаза, так как глаз воспринимает спектральные цвета как цвета с разной яркостью (синий кажется более темным, чем красный), а в модели HSB им всем приписывается яркость 100%.

Насыщенность (Saturation) – это параметр цвета, определяющий его чистоту. Отсутствие (серых) примесей (чистота кривой) соответствует данному параметру. Уменьшение насыщенности цвета означает его разбеливание. Цвет с уменьшением насыщенности становится пастельным, блеклым, размытым. На модели все одинаково насыщенные цвета располагаются на концентрических окружностях, т.е. можно говорить об одинаковой насыщенности, например, зеленого и пурпурного цветов, и чем ближе к центру круга, тем все более разбеленные цвета получают. В самом центре любой цвет максимально разбеливается, проще говоря, становится белым цветом.

Работу с насыщенностью можно характеризовать как добавление в спектральный цвет определенного процента белой краски. Чем больше в цвете содержание белого, тем ниже значение насыщенности, тем более блеклым он становится.

Яркость (Brightness) – это параметр цвета, определяющий освещенность или затемненность цвета. Амплитуда (высота) световой волны соответствует этому параметру. Уменьшение яркости цвета означает его зачернение. Работу с яркостью можно характеризовать как добавление в спектральный цвет определенного процента черной краски. Чем больше в цвете содержание черного, тем ниже яркость, тем более темным становится цвет.

Модель HSB – это пользовательская цветовая модель, которая позволяет выбирать цвет традиционным способом.

Модель CMY (Cyan Magenta Yellow)

В этой модели основные цвета образуются путем вычитания из белого цветов основных аддитивных цветов модели RGB.

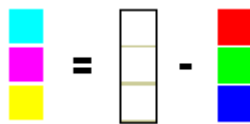


Рис. Получение модели CMY из RGB

Цвета, использующие белый свет, вычитая из него определенные участки спектра называются субтрактивными. Основные цвета этой модели: голубой (белый минус красный), фуксин (в некоторых книгах его называют пурпурным) (белый минус зеленый) и желтый (белый минус синий). Эти цвета являются полиграфической триадой и могут быть легко воспроизведены полиграфическими машинами. При смешении двух субтрактивных цветов результат затемняется (в модели RGB было наоборот). При нулевом значении всех компонент образуется белый цвет (белая бумага). Эта модель представляет отраженный цвет, и ее называют моделью **субтрактивных основных цветов**. Данная модель является основной для полиграфии и также является аппаратно-зависимой.



Рис. Модель CMY

Система координат CMY – тот же куб, что и для RGB, но с началом отсчета в точке с RGB координатами (1,1,1), соответствующей белому цвету. Цветовой куб модели CMY показан на рис. [0.4.2](#).

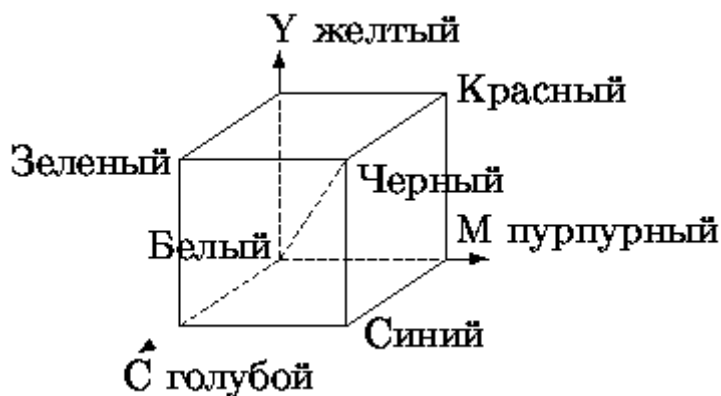


Рис. 0.4.2: Цветовой куб модели CMY

Цветовая модель CMYK

Это еще одна из наиболее часто используемых цветовых моделей, нашедших широкое применение. Она, в отличие от аддитивной RGB, является **субтрактивной** моделью.

Модель **CMYK** (Cyan Magenta Yellow Key, причем Key означает черный цвет) – является дальнейшим улучшением модели CMY и уже четырехканальна. Поскольку реальные типографские краски имеют примеси, их цвет не совпадает в точности с теоретически рассчитанным голубым, желтым и пурпурным. Особенно трудно получить из этих красок черный цвет. Поэтому в модели CMYK к триаде добавляют черный цвет. Почему – то в названии цветовой модели черный цвет зашифрован как К (от слова Key – ключ). Модель CMYK является «эмпирической», в отличие от теоретических моделей CMY и RGB. Модель является аппаратно-зависимой.

Основные цвета в субтрактивной модели отличаются от цветов аддитивной. **Cyan** – голубой, **Magenta** – пурпурный, **Yellow** – желтый. Так как при смешении всех вышеперечисленных цветов идеального черного не получится, то вводится еще один дополнительный цвет – черный, который позволяет добиваться большей глубины и используется при печати прочих черных (как, например, обычный текст) объектов.

Цвета в рассматриваемой цветовой модели были выбраны такими не случайно, а из-за того, что го-

лужбой поглощает лишь красный, пурпурный – зеленый, желтый – синий.

В отличие от аддитивной модели, где отсутствие цветовых составляющих образует черный цвет, в субтрактивной все наоборот: если нет отдельных компонентов, то цвет белый, если они все присутствуют, то образуется грязно-коричневый, который делается более темным при добавлении черной краски, которая используется для затемнения и других получаемых цветов. При смешивании отдельных цветовых составляющих можно получить следующие результаты:

Голубой + Пурпурный = Синий с оттенком фиолетового, который можно усилить, изменив пропорции смешиваемых цветов.

Пурпурный + Желтый = Красный. В зависимости от соотношения входящих в него составляющих он может быть преобразован в оранжевый или розовый.

Желтый + Голубой = Зеленый, который может быть преобразован при использовании тех же первичных цветов как в салатовый, так и в изумрудный.

Следует помнить, что если вы готовите изображение к печати, то следует все-таки работать с CMYK, потому что в противном случае то, что вы увидите на мониторе, и то, что получите на бумаге, будет отличаться настолько сильно, что вся работа может пойти насмарку.

Модель CMYK – это субтрактивная цветовая модель, которая описывает реальные красители, используемые в полиграфическом производстве.

Цветовая модель Lab

Цветовая модель Lab была разработана Международной комиссией по освещению (CIE) с целью преодоления существенных недостатков вышеизложенных моделей, в частности она призвана стать аппаратно независимой моделью и определять цвета без оглядки на особенности устройства (сканера, монитора, принтера, печатного станка и т. д.).

Такую модель предпочитают в основном профессионалы, так как она совмещает достоинства как CMYK, так и RGB, а именно обеспечивает доступ ко всем цветам, работая с достаточно большой скоростью.

На вопрос, почему же такой моделью пользуются в основном профессионалы, можно ответить лишь то, что она отличается несколько необычным и непривычным построением, и понять принцип ее действия порой несколько сложнее описанных ранее.

Построение цветов здесь, так же как и в RGB, базируется на слиянии трех каналов. На этом, правда, все сходство заканчивается.

Название она получила от своих базовых компонентов **L**, **a** и **b**. Компонент **L** несет информацию о яркостях изображения, а компоненты **a** и **b** – о его цветах (т. е. **a** и **b** – хроматические компоненты). Компонент **a** изменяется от зеленого до красного, а **b** – от синего до желтого. Яркость в этой модели отделена от цвета, что удобно для регулирования контраста, резкости и т.д. Однако, будучи абстрактной и сильно математизированной эта модель остается пока что неудобной для практической работы.

Поскольку все цветовые модели являются математическими, они легко конвертируются одна в другую по простым формулам. Такие конверторы встроены во все "приличные" графические программы.

Перцепционные цветовые модели

Для дизайнеров, художников и фотографов основным инструментом индикации и воспроизведения цвета служит глаз. Этот естественный «инструмент» обладает цветовым охватом, намного превышающим возможности любого технического устройства, будь то сканер, принтер или фотоэкспонирующее устройство вывода на пленку.

Как было показано ранее, используемые для описания технических устройств цветовые системы RGB и CMYK являются аппаратнозависимыми. Это значит, что воспроизводимый или создаваемый с помощью них цвет определяется не только составляющими модели, но и зависит от характеристик устройства вывода.

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

- позволяет обращаться с цветом на интуитивно понятном уровне;
- значительно упрощает проблему согласования цветов, поскольку после установки значения яркости можно заняться настройкой цвета.

Прототипом всех цветовых моделей, использующих концепцию разделения яркости и цветности, является HSV–модель. К другим подобным системам относятся HSI, HSB, HSL и YUV. Общим для них является то, что цвет задается не в виде смеси трех основных цветов – красного, синего и зеленого, а определяется путем указания двух компонентов: цветности (цветового тона и насыщенности) и яркости.

Черно-белый и полутоновый режим

Черно-белый режим. Это обычный черно-белый режим, который полностью лишен цвета, в нем есть только белый, черный и градации серого. Ничего особенно нового сказать о данной цветовой модели невозможно, так как она состоит из одного канала, который полностью соответствует изображению и выглядит как обычная черно-белая фотография.

Художники и разработчики программного обеспечения иногда называют этот режим монохромной графикой, растровой графикой, или графикой с однобитовым разрешением.

Для отображения черно-белого изображения используются только два типа ячеек: черные и белые. Поэтому для запоминания каждого пиксела требуется только 1 бит памяти компьютера. Областям исходного изображения, имеющим промежуточные оттенки, назначаются черные или белые пикселы, поскольку других оттенков для этой модели не предусмотрено.

Этот режим можно использовать для работы с черно-белыми изображениями, полученными сканированием черно-белых чертежей и гравюр, а также иногда при выводе цветных изображений на черно-белую печать.

Полутоновый режим. Такой способ реализации изображения базируется на специфике восприятия изображения человеческим глазом, для которого область изображения, заполненная крупными точками, ассоциируется с более темными тонами и, наоборот, область, заполненная точками меньшего размера, воспринимается как более светлая. Режим Halftone поддерживается большинством принтеров.

Полутоновые изображения представляют собой однобитовые изображения с непрерывным тоном, которые реализуются с помощью конгломерата *точек* разного размера и формы.

Плашечные цвета

В некоторых типах полиграфической продукции используются всего два-три цвета, которые печатаются смесовыми красками, которые называются плашечными цветами (spot colors). В частности, к такой продукции относятся бланки, визитки, приглашения, прайс-листы и прочая акцидентная продукция. Каждый плашечный цвет репродуцируется с помощью отдельной печатной формы (плашки).

Для осуществления печати такой продукции дизайнер должен представить в типографию отдельные полосы оригинал-макетов с плашками на каждый смесовой цвет и крестами привода и приложить образцы цвета («выкраски») для каждой полосы.

Для того чтобы унифицировать использование таких цветов создают цветовые библиотеки.

В частности, известная фирма Pantone, которая является владельцем и разработчиком одноименной библиотеки, начиналась с того, что химик Ло-уренс Герберт создал совокупность различных цветов, составляемых из восьми красок, и напечатал альбом этих цветов, каждый из которых имел свой номер. С тех пор эта идея получила самое широкое развитие, цветовые библиотеки используются в самых разных областях и в первую очередь в компьютерной графике и полиграфии. Появилось множество других компаний, выпускающих другие стандартизированные библиотеки цветов (например, TRUMATCH SWATCHING SYSTEM, FOCOLTONE COLOUR SYSTEM, TOYO 88 ColorFinder1050 System и ANPA-COLOR system и т. д.).

Цветовой набор Process Color System Guide охватывает более 3000 цветов, получаемых при полиграфической печати, с рецептами процентного соотношения 16 базовых цветов для цветовой модели CMYK.

Кодирование цвета. Палитра

Для того чтобы компьютер имел возможность работать с цветными изображениями, необходимо представлять цвета в виде чисел – кодировать цвет. Способ кодирования зависит от цветовой модели и формата числовых данных в компьютере.

Для модели RGB каждая из компонент может представляться числами, ограниченными некоторым диапазоном – например, дробными числами от 0 до 1 либо целыми числами от 0 до некоторого максимального значения. В настоящее время достаточно распространенным является формат True Color, в котором каждая компонента представлена в виде байта, что дает 256 градаций для каждой компоненты: **R** = 0...255, **G** = 0...255, **B** = 0...255. Количество цветов составляет $256 \times 256 \times 256 = 16.7$ млн (2^{24}).

Такой способ кодирования цветов можно назвать **компонентным**. В компьютере коды изображений **True Color** представляются в виде троек байтов, либо упаковываются в длинное целое (четырехбайтное) – 32 бита.

При работе с изображениями в системах компьютерной графики часто приходится искать компромисс между качеством изображения (требуется как можно больше цветов) и ресурсами, необходимыми для хранения и воспроизведения изображения, исчисляемыми, например, объемом памяти (надо уменьшать количество бит на пиксел).

Кроме того, некоторое изображение само по себе может использовать ограниченное количество цветов. Например, для черчения может быть достаточно двух цветов, для человеческого лица важны оттенки розового, желтого, пурпурного, красного, зеленого; а для неба – оттенки голубого и серого. В этих случаях использование полноцветного кодирования цвета является избыточным.

При ограничении количества цветов используют **палитру**, представляющую набор цветов, важных для данного изображения. Палитру можно воспринимать как таблицу цветов. Палитра устанавливает взаимосвязь между кодом цвета и его компонентами в выбранной цветовой модели.

Компьютерные видеосистемы обычно предоставляют возможность программисту установить собственную палитру.

Каждый цвет изображения, использующего палитру, кодируется индексом, который будет определять номер строки в таблице палитры. Поэтому такой способ кодирования цвета называют **индексным**.

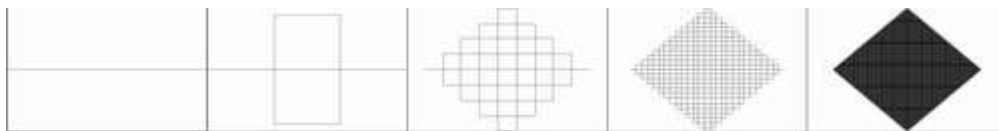
ТЕМА № 4. ФРАКТАЛЬНАЯ ГРАФИКА

Понятие фрактала и история появления фрактальной графики. Понятие размерности и ее расчет. Геометрические фракталы. Алгебраические фракталы. Системы итерированных функций. Стохастические фракталы. Фракталы и хаос.

Понятие фрактала и история появления фрактальной графики

Вы, наверное, часто видели довольно хитроумные картины, на которых непонятно что изображено, но все равно необычность их форм завораживает и привлекает внимание. Как правило, это хитроумные формы не поддающиеся, казалось бы, какому-либо математическому описанию. Вы, к примеру, видели узоры на стекле после мороза или, к примеру, хитроумные кляксы, оставленные на листе чернильной ручкой, так вот что-то подобное вполне можно записать в виде некоторого алгоритма, а, следовательно, доступно объясниться с компьютером. Подобные множества называют **фрактальными**. Фракталы не похожи на привычные нам фигуры, известные из геометрии, и строятся они по определенным алгоритмам, а эти алгоритмы с помощью компьютера можно изобразить на экране. Вообще, если все слегка упростить, то фракталы – это некое преобразование многократно примененное к исходной фигуре.

Первые идеи фрактальной геометрии возникли в 19 веке. **Кантор** с помощью простой рекурсивной (повторяющейся) процедуры превратил линию в набор несвязанных точек (так называемая **Пыль Кантора**). Он брал линию и удалял центральную треть и после этого повторял то же самое с оставшимися отрезками. **Пеано** нарисовал особый вид линии (см. рис). Для ее рисования Пеано использовал следующий алгоритм.



На первом шаге он брал прямую линию и заменял ее на 9 отрезков длиной в 3 раза меньшей, чем длина исходной линии (Часть 1 и 2 рисунка 1). Далее он делал то же самое с каждым отрезком получившейся линии. И так до бесконечности. Ее уникальность в том, что она заполняет всю плоскость. Доказано, что для каждой точки на плоскости можно найти точку, принадлежащую линии **Пеано**. **Кривая Пеано** и **пыль Кантора** выходили за рамки обычных геометрических объектов. Они не имели четкой размерности. **Пыль Кантора** строилась вроде бы на основании одномерной прямой, но состояла из точек, а **кривая Пеано** строилась на основании одномерной линии, а в результате получалась плоскость. Во многих других областях науки появлялись задачи, решение которых приводило к странным результатам, на подобие описанных (Броуновское движение, цены на акции).

Вплоть до 20 века шло накопление данных о таких странных объектах, без какой либо попытки их систематизировать. Так было, пока за них не взялся **Бенуа Мандельброт** – отец современной фрактальной геометрии и слова **фрактал**. Работая в IBM математическим аналитиком, он изучал шумы в электронных схемах, которые невозможно было описать с помощью статистики. Постепенно сопоставив факты, он пришел к открытию нового направления в математике – фрактальной геометрии.

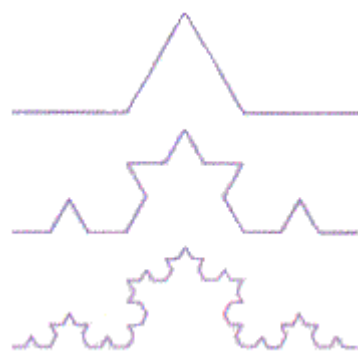
Сам Мандельброт вывел слово **fractal** от латинского слова *fractus*, что означает разбитый (поделенный на части). И одно из определений фрактала – это геометрическая фигура, состоящая из частей и которая может быть поделена на части, каждая из которых будет представлять уменьшенную копию целого (по крайней мере, приблизительно).

Как только **Мандельброт** открыл понятие **фрактала**, оказалось, что мы буквально окружены ими. Фрактальны слитки металла и горные породы, фрактальны расположение ветвей, узоры листьев, капиллярная система растений; кровеносная, нервная, лимфатическая системы в организмах животных, фрактальны речные бассейны, поверхность облаков, линии морских побережий, горный рельеф...

Чтобы представить себе фрактал понаглядней рассмотрим пример, приведенный в книге Б.Мандельброта «Фрактальная геометрия природы» ставший классическим – «Какова длина берега Британии?». Ответ на этот вопрос не так прост, как кажется. Все зависит от длины инструмента, которым мы будем пользоваться. Померив берег с помощью километровой линейки мы получим какую-то длину. Однако мы пропустим много небольших заливчиков и полуостровов, которые по размеру намного меньше нашей линейки. Уменьшив размер линейки до, скажем, 1 метра – мы учтем эти детали ландшафта, и, соответственно длина берега станет больше. Пойдем дальше и измерим длину берега с помощью миллиметровой линейки, мы тут учтем детали, которые больше миллиметра, длина будет еще больше. В итоге ответ на такой, казалось бы, простой вопрос может поставить в тупик кого угодно – длина берега Британии бесконечна.

Основное свойство фракталов – **самоподобие**. Любой микроскопический фрагмент фрактала в том или ином отношении воспроизводит его глобальную структуру. В простейшем случае часть фрактала представляет собой просто уменьшенный целый фрактал.

Отсюда основной рецепт построения фракталов: возьми простой мотив и повторяй его, постоянно уменьшая размеры. В конце концов выйдет структура, воспроизводящая этот мотив во всех масштабах.



Берем отрезок и среднюю его треть переламаываем под углом 60 градусов. Затем повторяем эту операцию с каждой из частей получившейся ломаной – и так до бесконечности. В результате мы получим простейший фрактал – **триадную кривую**, которую в 1904 году открыла математик **Хельга фон Кох**.

Если на каждом шаге не только уменьшать основной мотив, но также смещать и поворачивать его, можно получить более интересные и реалистически выглядящие образования, например, лист папоротника или даже целые их заросли. А можно построить весьма правдоподобный фрактальный рельеф местности и покрыть её очень симпатичным лесом. В 3D Studio Max, например, для генерации деревьев используется фрактальный алгоритм. И это не исключение – большинство текстур местности в современных компьютерных играх представляют фракталы. Горы, лес и облака на картинке – фракталы.

Файлы фрактальных изображений имеют расширение *fif*. Обычно файлы в формате *fif* получаются несколько меньше файлов в формате *jpg*, но бывает и наоборот. Самое интересное начинается, если рассматривать картинки со все большим увеличением. Файлы в формате *jpg* почти сразу демонстрируют свою дискретную природу – появляется пресловутая лесенка. А вот *fif* файлы, как и положено фракталам, с ростом увеличения показывают все новую степень детализации структуры, сохраняя эстетику изображения.

Понятие размерности и ее расчет

В своей повседневной жизни мы постоянно встречаемся с размерностями. Мы прикидываем длину дороги, узнаем площадь квартиры и т.д. Это понятие вполне интуитивно ясно и, казалось бы, не требует разъяснения. Линия имеет размерность 1. Это означает, что, выбрав точку отсчета, мы можем любую точку на этой линии определить с помощью 1 числа – положительного или отрицательного. Причем это касается всех линий – окружность, квадрат, парабола и т.д.

Размерность 2 означает, что любую точку мы можем однозначно определить двумя числами. Не надо думать, что двумерный – значит плоский. Поверхность сферы тоже двумерна (ее можно определить с помощью двух значений – углов наподобие ширины и долготы).

Если смотреть с математической точки зрения, то размерность определяется следующим образом: для одномерных объектов – увеличение в два раза их линейного размера приводит к увеличению размеров (в данном случае длины) в два раза (2^1).

Для двумерных объектов увеличение в два раза линейных размеров приводит к увеличению размера (например, площадь прямоугольника) в четыре раза (2^2).

Для 3-х мерных объектов увеличение линейных размеров в два раза приводит к увеличению объема в восемь раз (2^3) и так далее.

Таким образом, размерность D можно рассчитать исходя из зависимости увеличения «размера» объекта S от увеличения линейных размеров L . $D = \log(S)/\log(L)$. Для линии $D = \log(2)/\log(2) = 1$. Для плоскости $D = \log(4)/\log(2) = 2$. Для объема $D = \log(8)/\log(2) = 3$.

Рассчитаем размерность для кривой Пеано. Исходная линия, состоящая из трех отрезков длины X , заменяется на 9 отрезков втрое меньшей длины. Таким образом, при увеличении минимального отрезка в 3 раза длина всей линии увеличивается в 9 раз и $D = \log(9)/\log(3) = 2$ – двумерный объект.

Когда размерность фигуры получаемой из каких-то простейших объектов (отрезков) больше размерности этих объектов – мы имеем дело с фракталом.

Геометрические фракталы

Именно с них и начиналась история фракталов. Этот тип фракталов получается путем простых геометрических построений. Обычно при построении этих фракталов поступают так: берется «затравка» – аксиома – набор отрезков, на основании которых будет строиться фрактал. Далее к этой «затравке» применяют набор правил, который преобразует ее в какую-либо геометрическую фигуру. Далее к каждой части этой фигуры применяют опять тот же набор правил. С каждым шагом фигура будет становиться все сложнее и сложнее, и если мы проведем бесконечное количество преобразований – получим **геометрический фрактал**.

Рассмотренная ранее **кривая Пеано** является геометрическим фракталом. На рис. ниже приведены другие примеры геометрических фракталов (слева направо Снежинка Коха, Лист, Треугольник Серпинского).

Из этих геометрических фракталов очень интересным и довольно знаменитым является – **снежинка Коха**. Строится она на основе равностороннего треугольника. Каждая линия которого заменяется на 4 линии каждая длиной в $1/3$ исходной. Таким образом, с каждой итерацией длина кривой увеличивается на треть. И если мы сделаем бесконечное число итераций – получим фрактал – снежинку Коха бесконечной длины. Получается, что наша бесконечная кривая покрывает ограниченную площадь.

Размерность снежинки Коха (при увеличении снежинки в 3 раза ее длина возрастает в 4 раза)

$$D = \log(4)/\log(3) = 1.2619...$$

Для построения геометрических фракталов хорошо приспособлены так называемые **L-Systems**. Суть этих систем состоит в том, что имеется определенный набор символов системы, каждый из которых обозначает определенное действие и набор правил преобразования символов.

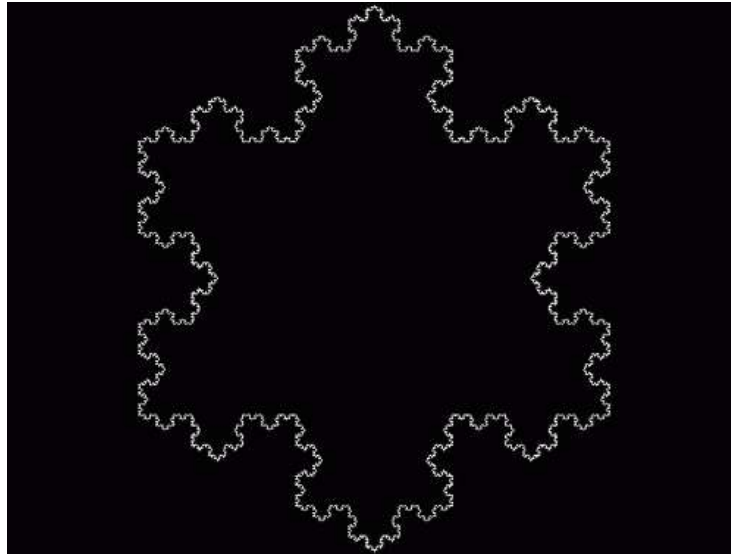


Рис. Снежинка Коха

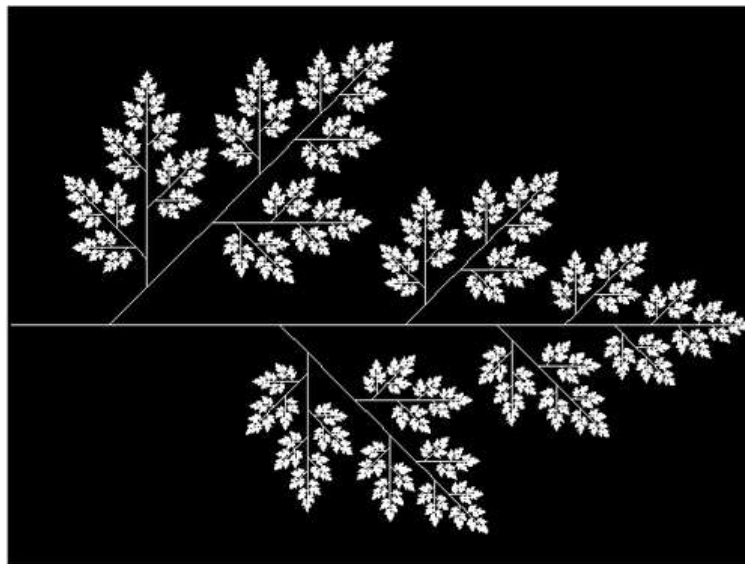


Рис. Лист

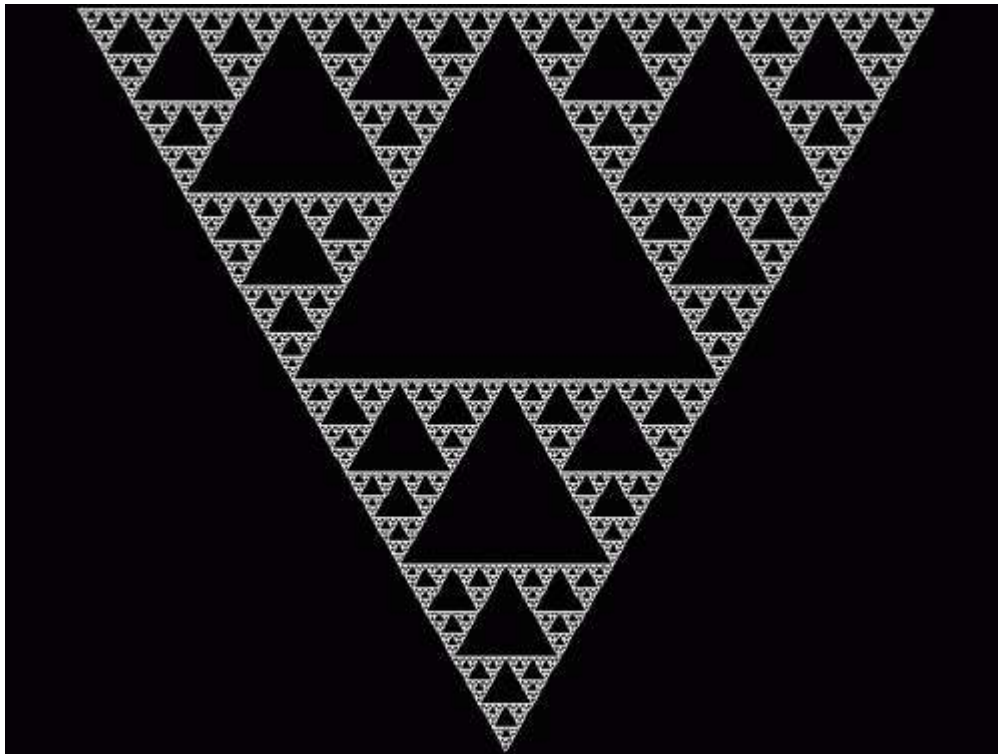


Рис. Треугольник Серпинского

Алгебраические фракталы

Вторая большая группа фракталов – **алгебраические**. Свое название они получили за то, что их строят, на основе алгебраических формул иногда весьма простых. Методов получения алгебраических фракталов несколько. Один из методов представляет собой многократный (итерационный) расчет функции $Z_{n+1}=f(Z_n)$, где Z – комплексное число, а f некая функция. Расчет данной функции продолжается до выполнения определенного условия. И когда это условие выполнится – на экран выводится точка. При этом значения функции для разных точек комплексной плоскости может иметь разное поведение:

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

Чтобы проиллюстрировать алгебраические фракталы обратимся к классике – **множеству Мандельброта**.

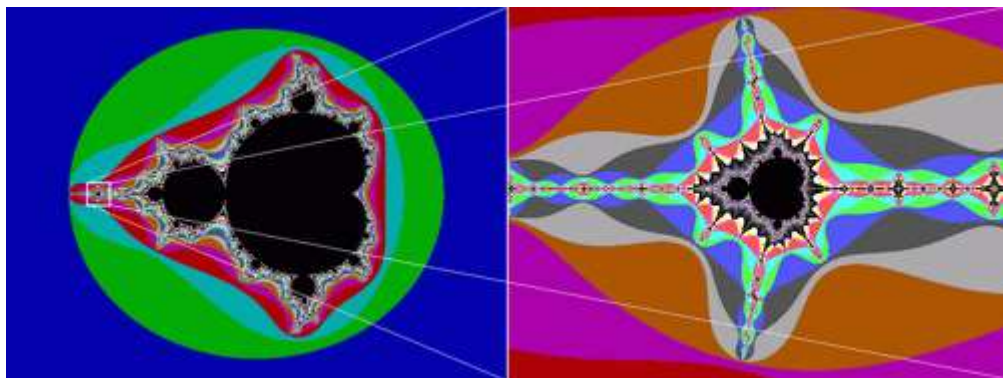


Рис. Множество Мандельброта

Для его построения нам необходимы комплексные числа. Комплексное число – это число, состоящее из двух частей – действительной и мнимой, и обозначается оно $a+bi$. Действительная часть a это обычное число в нашем представлении, а bi – мнимая часть. i – называют мнимой единицей, потому, что если мы возведем i в квадрат, то получим -1 .

Комплексные числа можно складывать, вычитать, умножать, делить, возводить в степень и извлекать корень, нельзя только их сравнивать. Комплексное число можно изобразить как точку на плоскости, у кото-

рой координата X это действительная часть a , а Y это коэффициент при мнимой части b .

Функционально множество Мандельброта определяется как $Z_{n+1}=Z_n*Z_n+C$. Для построения множества Мандельброта воспользуемся алгоритмом на Бейсике.

```

For a=-2 to 2 ' для всех действительных a от -2 до 2
For b=-2 to 2 ' для всех мнимых b от -2 до 2
C=a+bi
Z0=0+0i
'Принадлежит множеству Мандельброта
Lake=True
'Повторяем 255 раз (для режима 256 цветов)
For iteration=1 to 255
Zn=Z0*Z0+C
'Проверили – не принадлежит
If abs(Zn)>2 then Lake=False: Exit For
Z0=Zn
Next
'Нарисовали черную точку,принадлежащую "озеру" Мандельброта.
If Lake=True Then PutPixel(a,b,BLACK)
' Нарисовали точку не принадлежащую множеству или лежащую на границе.
Else PutPixel(a, b, iteration)
Next
Next

```

А теперь опишу программку словами. Для всех точек на комплексной плоскости в интервале от $-2+2i$ до $2+2i$ выполняем некоторое достаточно большое количество раз $Z_n=Z_0*Z_0+C$, каждый раз проверяя абсолютное значение Z_n . Если это значение больше 2, что рисуем точку с цветом равным номеру итерации на котором абсолютное значение превысило 2, иначе рисуем точку черного цвета. Все множество Мандельброта в полной красе у нас перед глазами.

Черный цвет в середине показывает, что в этих точках функция стремится к нулю – это и есть **множество Мандельброта**. За пределами этого множества функция стремится к бесконечности. А самое интересное это границы множества. Они то и являются фрактальными. На границах этого множества функция ведет себя непредсказуемо – хаотично.

Меняя функцию, условия выхода из цикла можно получать другие фракталы. Например, взяв вместо выражения $C=a+bi$ выражение $Z_0=a+bi$, а C присваивать произвольные значения мы получим **множество Жюлиа**, тоже красивый фрактал.

Для множества Мандельброта тоже проявляется самоподобие.

Стохастические фракталы

Типичный представитель данного класса фракталов «Плазма».

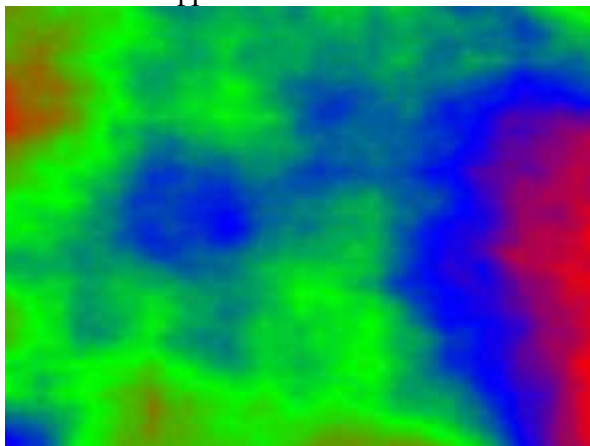
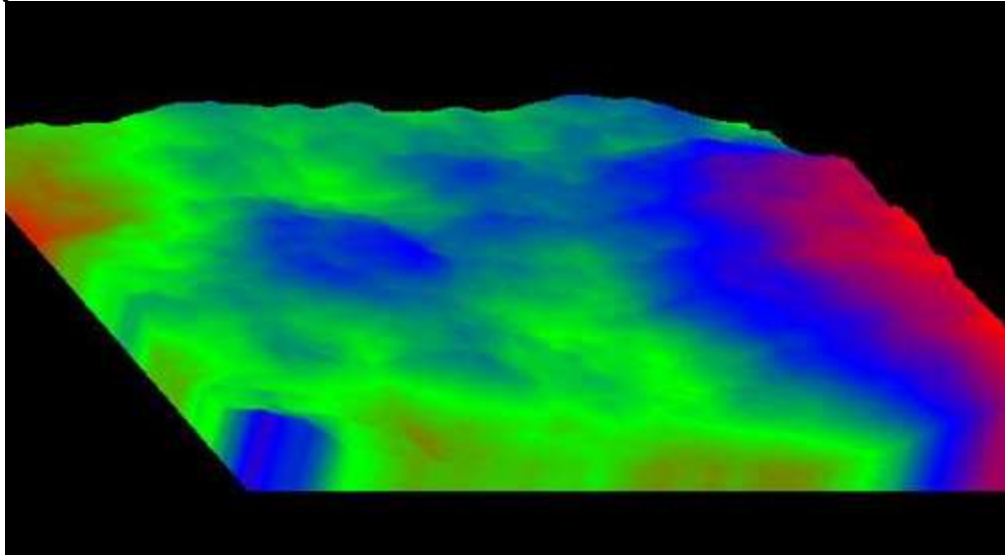


Рис. Плазма

Для ее построения возьмем прямоугольник и для каждого его угла определим цвет. Далее находим центральную точку прямоугольника и раскрашиваем ее в цвет равный среднему арифметическому цветов по углам прямоугольника плюс некоторое случайное число. Чем больше случайное число – тем более «рва-

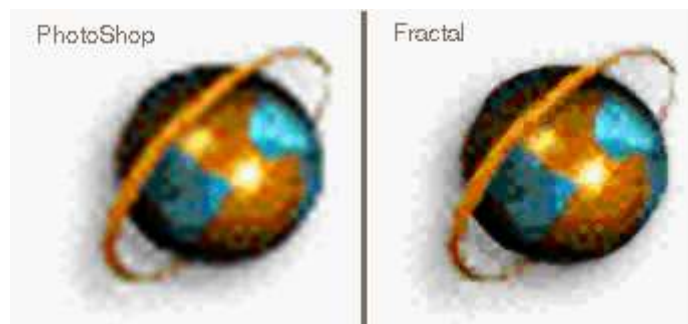
ным» будет рисунок. Если, например, сказать, что цвет точки это высота над уровнем моря, то получим вместо плазмы – горный массив. Именно на этом принципе моделируются горы в большинстве программ. С помощью алгоритма, похожего на плазму строится карта высот, к ней применяются различные фильтры, накладываем текстуру.



Системы итерируемых функций (IFS – Iterated Function Systems)

Эта группа фракталов получила широкое распространение благодаря работам Майкла Барнсли из технологического института штата Джорджия. Он пытался кодировать изображения с помощью фракталов. Запатентовав несколько идей по кодированию изображений с помощью фракталов, он основал фирму «Iterated Systems», которая через некоторое время выпустила первый продукт «Images Incorporated», в котором можно было изображения переводить из растровой формы во фрактальную FIF.

Это позволяло добиться высоких степеней сжатия. При низких степенях сжатия качество рисунков уступало качеству формата JPEG, но при высоких картинка получалась более качественными. В любом случае этот формат не прижился, но работы по его усовершенствованию ведутся до сих пор. Ведь этот формат не зависит от разрешения изображения. Так как изображение закодировано с помощью формул, то его можно увеличить до любых размеров и при этом будут появляться новые детали, а не просто увеличится размер пикселей.



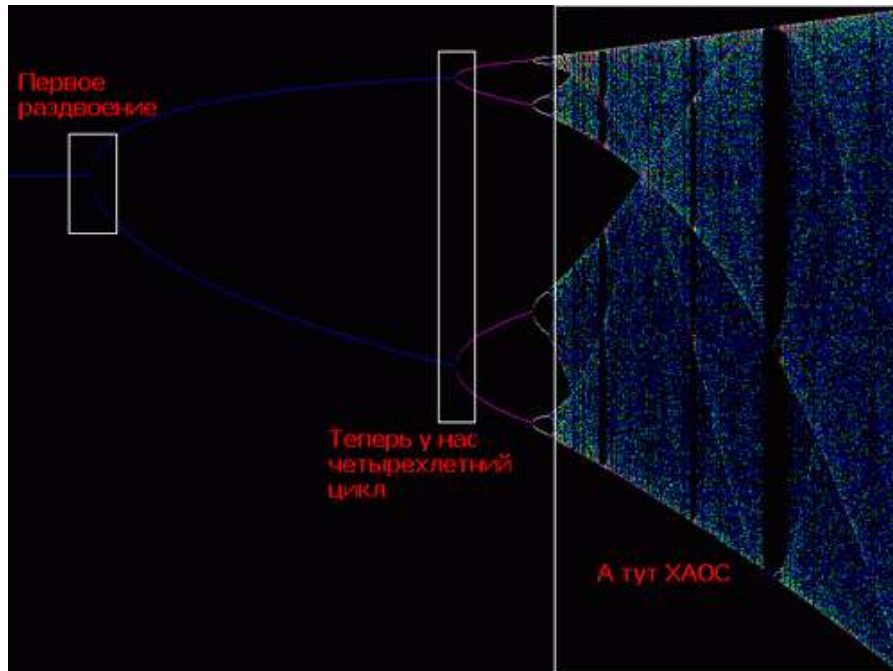
Если в L-systems (алгебраических фракталах) речь шла о замене прямой линии неким полигоном, то в IFS мы в ходе каждой итерации заменяем некий полигон (квадрат, треугольник, круг) на набор полигонов, каждый из которых подвергнут аффинным преобразованиям. При аффинных преобразованиях исходное изображение меняет масштаб, параллельно переносится вдоль каждой из осей и вращается на некоторый угол.

Фракталы и хаос

Понятие фрактал неразрывно связано с понятием хаос. Хаос – это отсутствие предсказуемости. Хаос возникает в динамических системах, когда для двух очень близких начальных значений система ведет себя совершенно по-разному. Пример хаотичной динамической системы – погода (метеорологи шутят: «Взмах крыла бабочки в Техасе приводит к урагану во Флориде»).

Хорошо проиллюстрировать хаотичное поведение можно с помощью так называемого logistic equation $x = c * x(1 - x)$. Пришло это выражение из биологии, т.к. это грубая модель популяции животных. Так

вот при исследовании поведения этой функции выяснилась интересная ее особенность. Если s – фактор роста популяции находится в пределах от 1 до 3, то через некоторое количество итераций популяция стабилизируется.



При $s=3$ наша функция раздваивается – через определенное число итераций приходим к ситуации, когда высокая популяция в один год сменяется низкой в следующий и значение выражения как бы скачет между двумя значениями.

При $s=3.45$ она раздваивается снова и у нас уже имеется четырехлетний цикл.

Далее при росте s функция раздваивается все быстрее и быстрее: при $s=3.54$, $s=3.564$, $s=3.569$...

И в точке 3.57 начинается хаос. Значения выражения не имеют какой либо периодичности или структуры. На рисунке изображена зависимость поведения функции от величины s .

ТЕМА № 5. РАСТРОВАЯ ГРАФИКА

Растровая графика, общие сведения. Растровые представления изображений. Виды растров. Факторы, влияющие на количество памяти, занимаемой растровым изображением. Достоинства и недостатки растровой графики. Геометрические характеристики раstra (разрешающая способность, размер раstra, форма пикселей). Количество цветов растрового изображения. Средства для работы с растровой графикой.

Растровая графика, общие сведения

Компьютерное растровое изображение представляется в виде прямоугольной матрицы, каждая ячейка которой представлена цветной точкой.

Основой **растрового** представления графики является **пиксель** (точка) с указанием ее цвета. При описании, например, красного эллипса на белом фоне необходимо указать цвет *каждой* точки эллипса и фона. Изображение представляется в виде большого количества точек – чем их больше, тем визуально качественнее изображение и больше размер файла. Т.е. одна и даже картинка может быть представлена с лучшим или худшим качеством в соответствии с количеством точек на единицу длины – *разрешением* (обычно, точек на дюйм – dpi или пикселей на дюйм – ppi).

Растровые изображения напоминают лист клетчатой бумаги, на котором любая клетка закрашена либо черным, либо белым цветом, образуя в совокупности рисунок. **Пиксел** – основной элемент растровых изображений. Именно из таких элементов состоит растровое изображение, т.е. растровая графика описывает изображения с использованием цветных точек (**пиксели**), расположенных на сетке.

При редактировании растровой графики Вы редактируете **пиксели**, а не **линии**. Растровая графика зависит от разрешения, поскольку информация, описывающая изображение, прикреплена к сетке опреде-

ленного размера. При редактировании растровой графики, качество ее представления может измениться. В частности, изменение размеров растровой графики может привести к «разлохмачиванию» краев изображения, поскольку пиксели будут перераспределяться на сетке. Вывод растровой графики на устройства с более низким разрешением, чем разрешение самого изображения, понизит его качество.

Кроме того, качество характеризуется еще и количеством цветов и оттенков, которые может принимать каждая точка изображения. Чем большим количеством оттенков характеризуется изображение, тем большее количество разрядов требуется для их описания. Красный может быть цветом номер 001, а может и – 00000001. Таким образом, чем качественнее изображение, тем больше размер файла.

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

С помощью растровой графики можно отразить и передать всю гамму оттенков и тонких эффектов, присущих реальному изображению. Растровое изображение ближе к фотографии, оно позволяет более точно воспроизводить основные характеристики фотографии: освещенность, прозрачность и глубину резкости.

Чаще всего растровые изображения получают с помощью сканирования фотографий и других изображений, с помощью цифровой фотокамеры или путем "захвата" кадра видеосъемки. Растровые изображения можно получить и непосредственно в программах растровой или векторной графики путем преобразования векторных изображений.

Распространены форматы **.tif, .gif, .jpg, .png, .bmp, .psx** и др.

Растровые представления изображений

Пиксел – основной элемент растровых изображений. Именно из таких элементов состоит растровое изображение.

Цифровое изображение – это совокупность пикселей. Каждый пиксел растрового изображения характеризуется координатами x и y и яркостью $V(x,y)$ (для черно-белых изображений). Поскольку пиксели имеют дискретный характер, то их координаты – это дискретные величины, обычно целые или рациональные числа. В случае цветного изображения, каждый пиксел характеризуется координатами x и y , и тремя яркостями: яркостью красного, яркостью синего и яркостью зеленого цветов (V_R, V_B, V_G). Комбинируя данные три цвета можно получить большое количество различных оттенков.

Заметим, что в случае, если хотя бы одна из характеристик изображения не является числом, то изображение относится к виду **аналоговых**. Примерами аналоговых изображений могут служить галограммы и фотографии. Для работы с такими изображениями существуют специальные методы, в частности, оптические преобразования. В ряде случаев аналоговые изображения переводят в цифровой вид. Эту задачу осуществляет Image Processing.

Цвет любого пиксела растрового изображения запоминается с помощью комбинации битов. Чем больше битов для этого используется, тем больше оттенков цветов можно получить. Под градацию яркости обычно отводится 1 байт (256 градаций), причем 0 – черный цвет, а 255 – белый (максимальная интенсивность). В случае цветного изображения отводится по байту на градации яркостей всех трех цветов. Возможно кодирование градаций яркости другим количеством битов (4 или 12), но человеческий глаз способен различать только 8 бит градаций на каждый цвет, хотя специальная аппаратура может потребовать и более точную передачу цветов. Цвета, описываемые 24 битами, обеспечивают более 16 миллионов доступных цветов и их часто называют естественными цветами.

В цветовых палитрах каждый пиксел описан кодом. Поддерживается связь этого кода с таблицей цветов, состоящей из 256 ячеек. Разрядность каждой ячейки – 24 разряда. На выходе каждой ячейки по 8 разрядов для красного, зеленого и синего цветов.

Цветовое пространство, образуемое интенсивностями красного, зеленого и синего, представляют в виде цветового куба (см. рис. 1.).

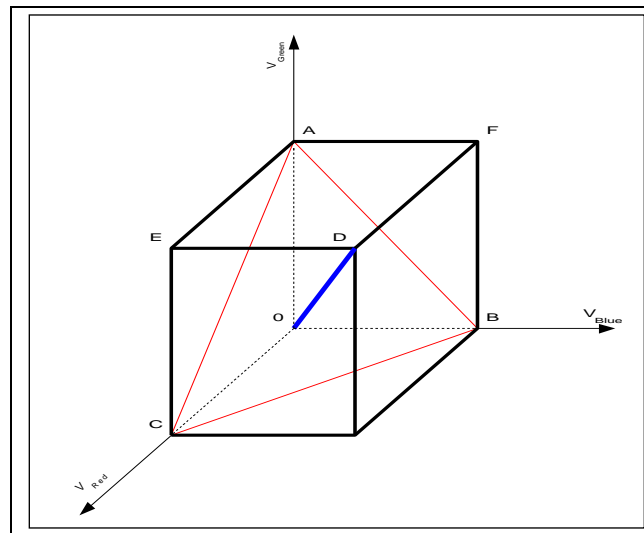


Рис. 1. Цветовой Куб

Вершины куба A, B, C являются максимальными интенсивностями зеленого, синего и красного соответственно, а треугольник, которые они образуют, называется **треугольником Паскаля**. Периметр этого треугольника соответствует максимально насыщенным цветам. Цвет максимальной насыщенности содержит всегда только две компоненты. На отрезке OD находятся оттенки серого, причем точка O соответствует черному, а точка D белому цвету.

Виды растров

Растр – это порядок расположения точек (растровых элементов). На рис. 2. изображен растр, элементами которого являются квадраты, такой растр называется **прямоугольным**, именно такие растры наиболее часто используются.

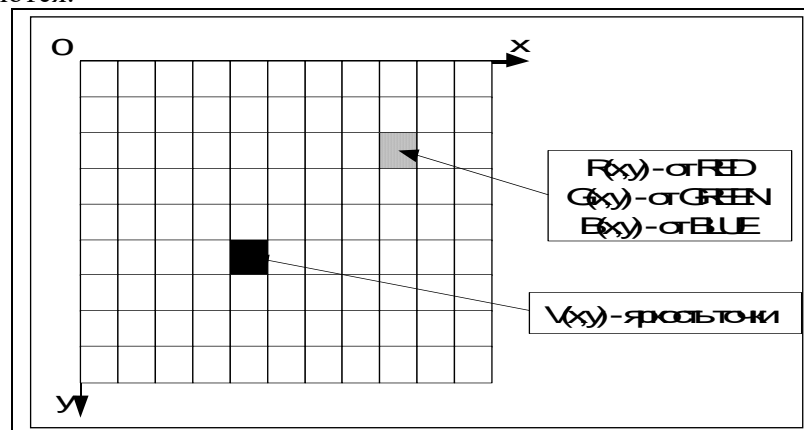


Рис. 2.

Хотя возможно использование в качестве растрового элемента фигуры другой формы: треугольника, шестиугольника; соответствующего следующим требованиям:

- все фигуры должны быть одинаковые;
- должны полностью покрывать плоскость без наезжания и дырок.

Так в качестве растрового элемента возможно использование равностороннего треугольника рис. 3, правильного шестиугольника (гексаэдра) рис. 4. Можно строить растры, используя неправильные многоугольники, но практический смысл в подобных растрах отсутствует.

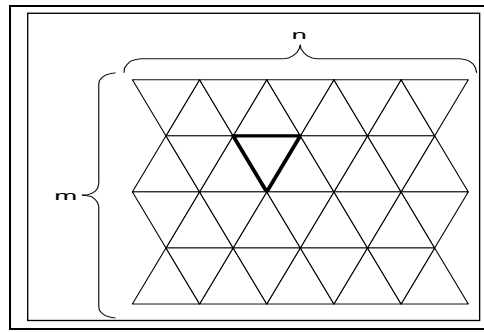


Рис. 3. Треугольный растр

Рассмотрим способы построения линий в прямоугольном и гексагональном растре.

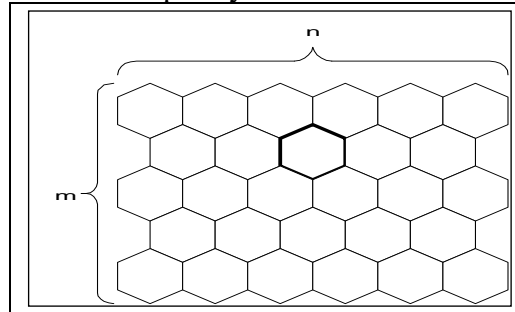


Рис. 4. «Гексагональный растр»

В прямоугольном растре построение линии осуществляется двумя способами:

- 1) Результат – восьмисвязная линия. Соседние пиксели линии могут находиться в одном из восьми возможных (см. рис. 5а) положений. Недосток – слишком тонкая линия при угле 45° .
- 2) Результат – четырехсвязная линия. Соседние пиксели линии могут находиться в одном из четырех возможных (см. рис. 5б) положений. Недосток – избыточно толстая линия при угле 45° .

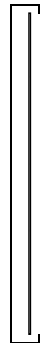


Рис. 5. Построение линии в прямоугольном растре

В гексагональном растре линии шестисвязные (см. рис. 6) такие линии более стабильны по ширине, т.е. дисперсия ширины линии меньше, чем в квадратном растре.



Рис. 6. Построение линии в гексагональном растре

Одним из способов оценки растра является передача по каналу связи кодированного, с учетом используемого растра, изображения с последующим восстановлением и визуальным анализом достигнутого качества. Экспериментально и математически доказано, что гексагональный растр лучше, т.к. обеспечивает наименьшее отклонение от оригинала. Но разница не велика.

Моделирование гексагонального растра. Возможно построение гексагонального растра на основе

квадратного. Для этого гексаугольник представляют в виде прямоугольника.

Факторы, влияющие на количество памяти, занимаемой растровым изображением

Файлы растровой графики занимают большое количество памяти компьютера. Некоторые картинки занимают большой объем памяти из-за большого количества пикселей, любой из которых занимает некоторую часть памяти. Наибольшее влияние на количество памяти занимаемой растровым изображением оказывают три факта:

- размер изображения;
- битовая глубина цвета;
- формат файла, используемого для хранения изображения.

Существует прямая зависимость размера файла растрового изображения. Чем больше в изображении пикселей, тем больше размер файла. Разрешающая способность изображения на величину файла никак не влияет. Разрешающая способность оказывает эффект на размер файла только при сканировании или редактировании изображений.

Связь между битовой глубиной и размером файла непосредственная. Чем больше битов используется в пикселе, тем больше будет файл. Размер файла растровой графики сильно зависит от формата выбранного для хранения изображения. При прочих равных условиях, таких как размеры изображения и битовая глубина существенное значение имеет схема сжатия изображения. Например, BMP файл имеет, как правило, большие размеры, по сравнению с файлами PCX и GIF, которые в свою очередь больше JPEG файла.

Многие файлы изображений обладают собственными схемами сжатия, также могут содержать дополнительные данные краткого описания изображения для предварительного просмотра.

Достоинства и недостатки растровой графики

Достоинства:

Растровая графика эффективно представляет реальные образы. Реальный мир состоит из миллиардов мельчайших объектов и человеческий глаз как раз приспособлен для восприятия огромного набора дискретных элементов, образующих предметы. На своем высшем уровне качества – изображение выглядит вполне реально подобно тому, как выглядят фотографии в сравнении с рисунками. Это верно только для очень детализированных изображений, обычно получаемых сканированием фотографий. Помимо естественного вида растровые изображения имеют другие преимущества. Устройства вывода, такие как лазерные принтеры, для создания изображений используют наборы точек. Растровые изображения могут быть очень легко распечатаны на таких принтерах, потому что компьютерам легко управлять устройством вывода для представления отдельных пикселей с помощью точек.

Недостатки:

Растровые изображения занимают большое количество памяти. Существует так же проблема редактирования растровых изображений, так как большие растровые изображения занимают значительные массивы памяти, то для обеспечения работы функций редактирования таких изображений потребуются так же значительные массивы памяти и другие ресурсы компьютера.

О сжатии растровой графики

Иногда характеристики растрового изображения записывают в такой форме: 1024x768x24. Это означает, что ширина изображения равна 1024 пикселям, высота – 768 и глубина цвета равна 24. 1024x768 – рабочее разрешение для 15 – 17 дюймовых мониторов. Несложно догадаться, что размер несжатого изображения с такими параметрами будет равен $1024 \times 768 \times 24 = 18874368$ байт. Это более 18 мегабайт – слишком много для одной картинке, особенно если требуется хранить несколько тысяч таких картинок – это не так уж много по компьютерным меркам. Вот почему компьютерную графику используют почти всегда в сжатом виде.

RLE (Run Length Encoding) – метод сжатия, заключающийся в поиске последовательностей одинаковых пикселей в точках растрового изображения («красный, красный, ..., красный» записывается как «N красных»).

LZW (Lempel–Ziv–Welch) – более сложный метод, ищет повторяющиеся фразы – одинаковые последовательности пикселей разного цвета. Каждой фразе ставится в соответствие некоторый код, при расшифровке файла код замещается исходной фразой.

При сжатии файлов формата JPEG (с потерей качества) изображение разбивается на участки 8x8 пикселей, и в каждом участке их значение усредняется. Усреднённое значение располагается в левом верхнем углу блока, остальное место занимает меньшими по яркости пикселями. Затем большинство пикселей обнуляются. При расшифровке нулевые пиксели получают одинаковый цвет. Затем к изображению применяется алгоритм Хаффмана.

Алгоритм Хаффмана основан на теории вероятности. Сначала элементы изображения (пиксели) сортируются по частоте встречаемости. Затем из них строится кодовое дерево Хаффмана. Каждому элементу сопоставляется кодовое слово. При стремлении размера изображения к бесконечности достигается максимальность сжатия. Этот алгоритм также используется в архиваторах.

Сжатие применяется и для векторной графики, но здесь уже нет таких простых закономерностей, так как форматы векторных файлов достаточно сильно различаются по содержанию.

Геометрические характеристики растра

Для растровых изображений, состоящих из точек, особую важность имеет понятие *разрешения*, выражающее количество точек, приходящихся на единицу длины. При этом следует различать:

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

Разрешение оригинала. Разрешение оригинала измеряется в *точках на дюйм (dots per inch – dpi)* и зависит от требований к качеству изображения и размеру файла, способу оцифровки и создания исходной иллюстрации, избранному формату файла и другим параметрам. В общем случае действует правило: чем выше требование к качеству, тем выше должно быть разрешение оригинала.

Разрешение экранного изображения. Для экранных копий изображения элементарную точку растра принято называть *пикселом*. Размер пиксела варьируется в зависимости от выбранного *экранного разрешения* (из диапазона стандартных значений), *разрешения оригинала* и масштаб отображения.

Мониторы для обработки изображений с диагональю 20–21 дюйм (профессионального класса), как правило, обеспечивают стандартные экранные разрешения 640x480, 800x600, 1024x768, 1280x1024, 1600x1200, 1600x1280, 1920x1200, 1920x1600 точек. Расстояние между соседними точками люминофора у качественного монитора составляет 0,22–0,25 мм.

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

Разрешение печатного изображения и понятие линиатуры. Размер точки растрового изображения как на твердой копии (бумага, пленка и т. д.), так и на экране зависит от примененного метода и параметров *растрирования* оригинала. При растрировании на оригинал как бы накладывается сетка линий, ячейки которой образуют *элемент растра*. Частота сетки растра измеряется числом *линий на дюйм (lines per inch – lpi)* и называется *линиатурой*.

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

Таким образом, разрешающая способность характеризует расстояние между соседними пикселями (рис. 1.). Разрешающую способность измеряют количеством пикселей на единицу длины. Наиболее популярно

лярной единицей измерения является **dpi** (dots per inch) – количество пикселей в одном дюйме длины (2.54 см). Не следует отождествлять шаг с размерами пикселей – размер пикселей может быть равен шагу, а может быть как меньше, так и больше, чем шаг.

Размер растра обычно измеряется количеством пикселей по горизонтали и вертикали. Можно сказать, что для компьютерной графики зачастую наиболее удобен растр с одинаковым шагом для обеих осей, то есть $\text{dpiX} = \text{dpiY}$. Это удобно для многих алгоритмов вывода графических объектов. Иначе – проблемы. Например, при рисовании окружности на экране дисплея EGA (устаревшая модель компьютерной видеосистемы, ее растр – прямоугольный, пиксели растянуты по высоте, поэтому для изображения окружности необходимо генерировать эллипс).

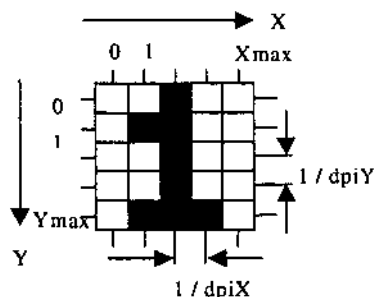


Рис. 1. Растр.

Форма пикселей растра определяется особенностями устройства графического вывода (рис. 1.2). Например, пиксели могут иметь форму прямоугольника или квадрата, которые по размерам равны шагу растра (дисплей на жидких кристаллах); пиксели круглой формы, которые по размерам могут и не равняться шагу растра (принтеры).

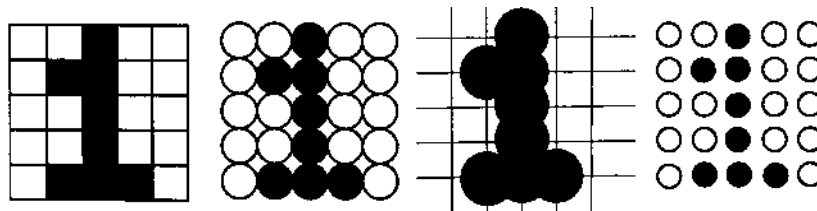


Рис. 2. примеры показа одного и того же изображения на различных растрах

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

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

Динамический диапазон. Качество воспроизведения тоновых изображений принято оценивать *динамическим диапазоном (D)*. Это *оптическая плотность*, численно равная десятичному логарифму величины, обратной *коэффициенту пропускания* (для оригиналов, рассматриваемых “на просвет”, например слайдов) или *коэффициенту отражения* (для прочих оригиналов, например полиграфических отпечатков).

Для оптических сред, пропускающих свет, динамический диапазон лежит в пределах от 0 до 4. Для поверхностей, отражающих свет, значение динамического диапазона составляет от 0 до 2. Чем выше динамический диапазон, тем большее число полутонов присутствует в изображении и тем лучше качество его восприятия.

В цифровом мире компьютерных изображений термином пиксел обозначают несколько разных понятий. Это может быть отдельная точка экрана компьютера, отдельная точка напечатанная на лазерном принтере или отдельный элемент растрового изображения. Эти понятия не одно и то же, поэтому чтобы избежать путаницы следует называть их следующим образом: видео пиксел при ссылке на изображение экрана компьютера; точка при ссылке на отдельную точку, создаваемую лазерным принтером. Существует ко-

эefficient прямоуольности изображения, который введен специально для изображения количества пикселей матрицы рисунка по горизонтали и по вертикали.

Возвращаясь к аналогии с листом бумаги можно заметить, что любой растровый рисунок имеет определенное количество пикселей в горизонтальных и вертикальных рядах. Существуют следующие коэффициенты прямоуольности для экранов: 320x200, 320x240, 600x400, 640x480, 800x600 и др. Этот коэффициент часто называют размером изображения. Произведение этих двух чисел дает общее количество пикселей изображения.

Существует также такое понятие как коэффициент прямоуольности пикселей. В отличие от коэффициента прямоуольности изображения он относится к реальным размерам видео пиксела и является отношением реальной ширины к реальной высоте. Данный коэффициент зависит от размера дисплея и текущего разрешения, и поэтому на разных компьютерных системах принимает различные значения. Цвет любого пиксела растрового изображения запоминается в компьютере с помощью комбинации битов. Чем больше битов для этого используется, тем больше оттенков цветов можно получить. Число битов, используемых компьютером для любого пиксела, называется битовой глубиной пиксела. Наиболее простое растровое изображение состоит из пикселей имеющих только два возможных цвета черный и белый, и поэтому изображения, состоящие из пикселей этого вида, называются однобитовыми изображениями. Число доступных цветов или градаций серого цвета равно 2 в степени равной количеству битов в пикселе.

Цвета, описываемые 24 битами, обеспечивают более 16 миллионов доступных цветов и их часто называют естественными цветами. Растровые изображения обладают множеством характеристик, которые должны быть организованы и фиксированы компьютером.

Размеры изображения и расположение пикселей в нем это две основных характеристики, которые файл растровых изображений должен сохранить, чтобы создать картинку. Даже если испорчена информация о цвете любого пиксела и любых других характеристиках компьютер все равно сможет воссоздать версию рисунка, если будет знать, как расположены все его пиксели. Пиксел сам по себе не обладает никаким размером, он всего лишь область памяти компьютера, хранящая информацию о цвете, поэтому коэффициент прямоуольности изображения не соответствует никакой реальной размерности. Зная только коэффициент прямоуольности изображения с некоторой разрешающей способностью можно определить настоящие размеры рисунка. Поскольку размеры изображения хранятся отдельно, пиксели запоминаются один за другим, как обычный блок данных. Компьютеру не приходится сохранять отдельные позиции, он всего лишь создает сетку по размерам заданным коэффициентом прямоуольности изображения, а затем заполняет ее пиксел за пикселем.

Количество цветов растрового изображения

Количество цветов (глубина цвета) – также одна из важнейших характеристик растра. Количество цветов является важной характеристикой для любого изображения, а не только растрового.

Классифицируем изображения следующим образом:

- Двухцветные (бинарные) – 1 бит на пиксел. Среди двухцветных чаще всего встречаются черно-белые изображения.
- Полутоновые – градации серого или иного цвета. Например, 256 градаций (1 байт на пиксел).
- Цветные изображения. От 2 бит на пиксел и выше. Глубина цвета 16 бит на пиксел (65 536 цветов) получила название **High Color**, 24 бит на пиксел (16,7 млн цветов) – **True Color**. В компьютерных графических системах используют и большую глубину цвета – 32, 48 и более бит на пиксел.

Форматы растровых графических файлов

GIF – формат, использующий алгоритм сжатия без потерь информации LZW. Максимальная глубина цвета – 8 бит (256 цветов). В нём также есть возможность записи анимации. Поддерживает прозрачность пикселей (двухуровневая – полная прозрачность, либо полная непрозрачность). Данный формат широко применяется при создании Web-страниц. GIF-формат позволяет записывать изображение «через строчку», благодаря чему, имея только часть файла, можно увидеть изображение целиком, но с меньшим разрешением. Его выгодно применять для изображений с малым количеством цветов и резкими границами (например, для текстовых изображений).

JPEG (JPG) – формат, использующий алгоритм сжатия с потерями информации, который позволяет уменьшить размер файла в сотни раз. Глубина цвета – 24 бит. Не поддерживается прозрачность пикселей. При сильном сжатии в области резких границ появляются дефекты. Формат JPEG хорошо применять для

сжатия полноцветных фотографий. Учитывая то, что при повторном сжатии происходит дальнейшее ухудшение качества, рекомендуется сохранять в JPEG только конечный результат работы. JPEG широко применяется при создании Web-страниц, а также для хранения больших коллекций фотографий.

Сравнение GIF и JPEG

- GIF – формат удобен при работе с рисованными картинками;
- JPEG – формат лучше использовать для хранения фотографий и изображений с большим количеством цветов;
- для создания анимации и изображений с прозрачным фоном применяется GIF-формат.

BMP – это формат графического редактора Paint. В нём не применяется сжатие. Он хорошо подходит для хранения очень маленьких изображений – таких как иконки на рабочем столе. Большие же файлы в этом формате занимают слишком много места.

PNG – разработан с целью заменить формат GIF. Использует алгоритм сжатия Deflate без потерь информации (усовершенствованный LZW). Максимальная глубина цвета – 48 бит. Поддерживает каналы градиентных масок прозрачности (256 уровней прозрачности). PNG – относительно новый формат, и поэтому ещё не очень распространён. В основном используется в Web-дизайне. К сожалению, даже в некоторых современных браузерах (таких, как Internet Explorer 6) отсутствует поддержка прозрачности PNG и поэтому не рекомендуется использовать прозрачные PNG изображения на Web-страницах.

TIFF – формат, специально разработанный для сканированных изображений. Может использовать алгоритм сжатия без потерь информации LZW. Позволяет сохранять информацию о слоях, цветовых профилях (ICC-профилях) и каналах масок. Поддерживает все цветовые модели. Аппаратно независим. Используется в издательских системах, а также для переноса графической информации между различными платформами.

PSD – формат графического редактора Adobe Photoshop. Использует алгоритм сжатия без потерь информации RLE. Позволяет сохранять всю информацию, создаваемую в этой программе. Кроме этого, в связи с популярностью Photoshop, данный формат поддерживается практически всеми современными редакторами компьютерной графики. Его удобно использовать для сохранения промежуточного результата при работе в Photoshop и других растровых редакторах.

RIFF – формат графического редактора Corel Painter. Позволяет сохранять всю информацию, создаваемую в этой программе. Его следует использовать для сохранения промежуточного результата при работе в Painter.

Формат	Макс. число бит/пиксел	Макс. число цветов	Макс. размер изображения, пиксел	Методы сжатия	Кодирование нескольких изображений
BMP	24	16 777 216	65535 x 65535	RLE	–
GIF	8	256	65535 x 65535	LZW	+
JPEG	24	16 777 216	65535 x 65535	JPEG	–
PCX	24	16 777 216	65535 x 65535	RLE	–
PNG	48	281 474 976 710 656	2 147 483 647 x 2 147 483 647	Deflation (вариант LZ77)	–
TIFF	24	16 777 216	всего 4 294 967 295	LZW, RLE и другие	+

Средства для работы с растровой графикой

В обширном классе программ для обработки растровой графики особое место занимает пакет Photoshop компании Adobe. Сегодня он является стандартом в компьютерной графике, и все другие программы неизменно сравнивают именно с ним.

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

- **Палитра Кисти** управляет настройкой параметров инструментов редактирования. В режим редактирования кисти входят после двойного щелчка на ее изображении в палитре. Щелчок при нажатой клавише CTRL уничтожает кисть. Двойным щелчком на свободном поле палитры открывают диалоговое окно формирования новой кисти, которая автоматически добавляется в палитру.

– **Палитра Параметры** служит для редактирования свойств текущего инструмента. Открыть ее можно не только из строки меню, но и двойным щелчком на значке инструмента в панели инструментов. Состав элементов управления палитры зависит от выбранного инструмента.

– **Палитра Инфо** обеспечивает информационную поддержку средств отображения. На ней представлены: текущие координаты указателя мыши, размер текущей выделенной области, цветовые параметры элемента изображения и другие данные.

– **Палитра Навигатор** позволяет просмотреть различные фрагменты изображения и изменить масштаб просмотра. В окне палитры помещена миниатюра изображения с выделенной областью просмотра.

– **Палитра Синтез** отображает цветовые значения текущих цветов переднего плана и фона. Ползунки на цветовой линейке соответствующей цветовой системы позволяют редактировать эти параметры.

– **Палитра Каталог** содержит набор доступных цветов. Такой набор можно загрузить и отредактировать, добавляя и удаляя цвета. Цветовой тон переднего плана и фона выбирают из состава набора. В стандартном комплекте поставки программы предусмотрено несколько цветовых наборов, в основном компании Pantone.

– **Палитра Слои** служит для управления отображением всех слоев изображения, начиная с самого верхнего. Возможно определение параметров слоев, изменение их порядка, операции со слоями с применением разных методов.

– **Палитру Каналы** используют для выделения, создания, дублирования и удаления каналов, определения их параметров, изменения порядка, преобразования каналов в самостоятельные объекты и формирования совмещенных изображений из нескольких каналов.

– **Палитра Контуры** содержит список всех созданных контуров. При преобразовании контура в выделенную область его используют для формирования обтравочного контура.

– **Палитра Операции** позволяет создавать макрокоманды – заданную последовательность операций с изображением. Макрокоманды можно записывать, выполнять, редактировать, удалять, сохранять в виде файлов.

Особую группу программных средств обработки изображений представляют **Фильтры**. Это подключаемые к программе модули, часто третьих фирм, позволяющие обрабатывать изображение по заданному алгоритму. Иногда такие алгоритмы бывают очень сложными, а окно фильтра может иметь множество настраиваемых параметров. Из групп фильтров популярны продукты серий Kai's Power Tools, Alien Skin, Andromeda и другие.

ТЕМА № 6. ВЕКТОРНАЯ ГРАФИКА

Векторная графика. Объекты и их атрибуты. Структура векторной иллюстрации. Достоинства и недостатки векторной графики. Пиксель. Битовая глубина, определение числа доступных цветов в компьютерной графике. Элементы (объекты) векторной графики. Средства для создания векторных изображений.

Векторная графика, общие сведения

Векторная графика описывает изображения с использованием прямых и изогнутых линий, называемых **векторами**, а также параметров, описывающих цвета и расположение. Например, изображение древесного листа (см. рис. 1.) описывается точками, через которые проходит линия, создавая тем самым контур листа. Цвет листа задается цветом контура и области внутри этого контура.



Рис. 1. Пример векторной графики

В отличие от растровой графики в векторной графике изображение строится с помощью математических описаний объектов, окружностей и линий. Хотя на первый взгляд это может показаться сложнее, чем использование растровых массивов, но для некоторых видов изображений использование математических описаний является более простым способом.

Ключевым моментом векторной графики является то, что она использует комбинацию компьютерных команд и математических формул для объекта. Это позволяет компьютерным устройствам вычислять и помещать в нужном месте реальные точки при рисовании этих объектов. Такая особенность векторной графики дает ей ряд преимуществ перед растровой графикой, но в тоже время является причиной ее недостатков.

Векторную графику часто называют **объектно-ориентированной графикой** или **чертежной графикой**. Простые объекты, такие как окружности, линии, сферы, кубы и тому подобное называется примитивами, и используются при создании более сложных объектов. В векторной графике объекты создаются путем комбинации различных объектов.

Для создания векторных рисунков необходимо использовать один из многочисленных иллюстрационных пакетов. Достоинство векторной графики в том, что описание является простым и занимает мало памяти компьютера. Однако недостатком является то, что детальный векторный объект может оказаться слишком сложным, он может напечататься не в том виде, в каком ожидает пользователь или не напечатается вообще, если принтер неправильно интерпретирует или не понимает векторные команды.

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

Векторное представление заключается в описании элементов изображения математическими кривыми с указанием их цветов и заливочности.

Еще одно преимущество – качественное масштабирование в любую сторону. Увеличение или уменьшение объектов производится увеличением или уменьшением соответствующих коэффициентов в математических формулах. К сожалению, векторный формат становится невыгодным при передаче изображений с большим количеством оттенков или мелких деталей (например, фотографий). Ведь каждый мельчайший блик в этом случае будет представляться не совокупностью одноцветных точек, а сложнейшей математической формулой или совокупностью графических примитивов, каждый из которых, является формулой. Это приводит к утяжелению файла. Кроме того, перевод изображения из растрового в векторный формат (например, программой Adobe Strime Line или Corel OCR-TRACE) приводит к наследованию последним невозможности корректного масштабирования в большую сторону. От увеличения линейных размеров количество деталей или оттенков на единицу площади больше не становится. Это ограничение накладывается разрешением вводных устройств (сканеров, цифровых фотокамер и др.).

Элементы (объекты) векторной графики. Объекты и их атрибуты

Основным логическим элементом векторной графики является геометрический объект. В качестве объекта принимаются простые геометрические фигуры (так называемые примитивы – прямоугольник, окружность, эллипс, линия), составные фигуры или фигуры, построенные из примитивов, цветовые заливки, в том числе градиенты.

Важным объектом векторной графики является **сплайн**. Сплайн – это кривая, посредством которой

описывается та или иная геометрическая фигура. На сплайнах построены современные шрифты TrueType и PostScript.

Объекты векторной графики легко трансформируются и модифицируются, что не оказывает практически никакого влияния на качество изображения. Масштабирование, поворот, искривление могут быть сведены к паре–тройке элементарных преобразований над векторами.

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

Линия – элементарный объект векторной графики. Как и любой объект, линия обладает свойствами: формой (прямая, кривая), толщиной, цветом, начертанием (сплошная, пунктирная). Замкнутые линии приобретают свойство заполнения. Охватываемое ими пространство может быть заполнено другими объектами (текстуры, карты) или выбранным цветом. Простейшая незамкнутая линия ограничена двумя точками, именуемыми узлами. Узлы также имеют свойства, параметры которых влияют на форму конца линии и характер сопряжения с другими объектами. Все прочие объекты векторной графики состояются из линий. Например, куб можно составить из шести связанных прямоугольников, каждый из которых, в свою очередь, образован четырьмя связанными линиями.

Цвет в векторной графике

Различные векторные форматы обладают различными цветовыми возможностями. Простейшие форматы, которые могут не содержать вообще никакой информации о цвете, используют цвет по умолчанию тех устройств, на которые они выводятся, другие форматы способны сохранять данные о полном тридцати двух битном цвете. Какую бы цветовую модель не применял бы векторный формат, на размер файла он не влияет, кроме тех случаев, когда файл содержит растровые образы. В обычных векторных объектах значение цвета относится ко всему объекту в целом. Цвет объекта хранится в виде части его векторного описания. Некоторые векторные файлы могут создать растровый эскиз изображений хранящихся в них. Эти растровые картинки, иногда называемые краткими описаниями изображений, обычно представляют собой эскизы векторных рисунков в целом. Краткое описание изображения, особенно полезно в ситуациях, когда вы не хотите открывать весь файл, чтобы посмотреть, что в нем хранится или когда вы не можете видеть векторный рисунок во время его использования.

Первая ситуация возникает, когда вам необходимо найти файл с помощью одной из многих специально разработанных для этого программ. Для облегчения поиска нужного векторного файла такие программы могут считывать растровый эскиз изображения и другие характеристики, например, векторный формат, время создания, битовую глубину изображения и так далее.

Вторая ситуация возникает, когда в каком-либо издательском пакете помещается на страницу векторный рисунок. Изображение, которое вы увидите, будет растровым эскизом настоящего векторного рисунка, у которого нельзя изменить размер, обрезать или как-то иначе обработать изображение. За эскизы изображения приходится расплачиваться памятью, т.к. эскизы – это растровая версия рисунков, а растровые данные используют много памяти компьютера.

Структура векторной иллюстрации.

Структуру любой векторной иллюстрации можно представить в виде иерархического дерева. В такой иерархии сама иллюстрация занимает верхний уровень, а ее составные части – более низкие уровни иерархии.

1. Самый верхний иерархический уровень занимает сама картинка, которая объединяет в своем составе объекты + узлы + линии + заливки.
2. Следующий уровень иерархии – объекты, которые представляют собой разнообразные векторные формы.
3. Объекты иллюстрации состоят из одного или нескольких контуров: замкнутых и открытых. **Контуром** называется любая геометрическая фигура, созданная с помощью рисующих инструментов векторной программы и представляющая собой очертания того или иного графического объекта (окружность, прямоугольник и т.п.). **Замкнутый контур** – это замкнутая кривая, у которой начальная и конечная точки совпа-

дают (окружность). **Открытый контур** имеет четко обозначенные концевые точки (синусоидальная линия).

4. Следующий уровень иерархии составляют сегменты, которые выполняют функции кирпичиков, используемых для построения контуров. Каждый контур может состоять из одного или нескольких сегментов. Начало и конец каждого сегмента называются узлами, или опорными точками, поскольку они фиксируют положение сегмента, «привязывая» его к определенной позиции в контуре. Перемещение узловых точек приводит к модификации сегментов контура и к изменению его формы. Замкнутые контуры (формы) имеют свойство заполнения цветом, текстурой или растровым изображением (картой). Заливка – это цвет или узор, выводимый в замкнутой области, ограниченной кривой.

5. На самом нижнем уровне иерархии расположены узлы и отрезки линий, соединяющих между собой соседние узлы. Линии наряду с узлами выполняют функции основных элементов векторного изображения.

Достоинства векторной графики

Самая сильная сторона векторной графики в том, что она использует все преимущества разрешающей способности любого устройства вывода. Это позволяет изменять размеры векторного рисунка без потери его качества. Векторные команды просто сообщают устройству вывода, что необходимо нарисовать объект заданного размера, используя столько точек сколько возможно. Другими словами, чем больше точек сможет использовать устройство вывода для создания объекта, тем лучше он будет выглядеть. Растровый формат файла точно определяет, сколько необходимо создать пикселей и это количество изменяется вместе с разрешающей способностью устройства вывода. Вместо этого происходит одно из двух либо при увеличении разрешающей способности, размер растровой окружности уменьшается, так как уменьшается размер точки составляющих пиксел; либо размер окружности остается одинаковым, но принтеры с высокой разрешающей способностью используют больше точек для любого пиксела. Векторная графика обладает еще одним важным преимуществом, здесь можно редактировать отдельные части рисунка не оказывая влияния на остальные, например, если нужно сделать больше или меньше только один объект на некотором изображении, необходимо просто выбрать его и осуществить задуманное. Объекты на рисунке могут перекрываться без всякого воздействия друг на друга. Векторное изображение, не содержащее растровых объектов, занимает относительно не большое место в памяти компьютера. Даже очень детализированные векторные рисунки, состоящие из 1000 объектов, редко превышают несколько сотен килобайт.

Недостатки векторной графики

Природа избегает прямых линий. К сожалению, они являются основными компонентами векторных рисунков. До недавнего времени это означало, что уделом векторной графики были изображения, которые никогда не старались выглядеть естественно, например, двухмерные чертежи и круговые диаграммы, созданные специальными программами САПР, двух и трех мерные технические иллюстрации, стилизованные рисунки и значки, состоящие из прямых линий и областей, закрашенных однотонным цветом. Векторные рисунки состоят из различных команд посылаемых от компьютера к устройствам вывода (принтеру). Принтеры содержат свои собственные микропроцессоры, которые интерпретируют эти команды и пытаются их перевести в точки на листе бумаги. Иногда из-за проблем связи между двумя процессорами принтер не может распечатать отдельные детали рисунков. В зависимости от типов принтера случаются проблемы, и у вас может оказаться чистый лист бумаги, частично напечатанный рисунок или сообщение об ошибке.

Применение векторной графики

Успехи компьютерных технологий, достигнутые в последние годы, не оставляют места сомнениям при выборе способов получения, хранения и переработки данных о сложных комплексных трехмерных объектах, таких, например, как памятники архитектуры и археологии, объекты спелеологии и т. д. Несомненно, что применение компьютеризации для этих целей – дело не далекого будущего, а уже настоящего времени. Последнее, конечно, в большой мере зависит от количества денежных средств, вкладываемых с этой целью.

Наука и инженерия

Системы CAD/CAM используются сегодня в различных областях инженерной конструкторской деятельности от проектирования микросхем до создания самолетов. Ведущие инженерные и производственные компании, такие как Boeing, в конечном счете двигаются к полностью цифровому представлению конструкции самолетов.

Архитектура является другой важной областью применения для CAD/CAM и совсем недавно созданных систем класса walkthrough (прогулки вокруг проектируемого объекта с целью его изучения и оценки). Такие фирмы, как McDonald's, уже с 1987 года используют машинную графику для архитектурного дизайна, размещения посадочных мест, планирования помещений и проектирования кухонного оборудования. Есть ряд эффектных применений векторной графики в области проектирования стадионов и дизайна спортивного инвентаря, новый парк в Балтиморе (Baltimore Orioles' Camden Yards Park).

Медицина стала весьма привлекательной сферой применения компьютерной графики, например: автоматизированное проектирование имплантатов, особенно для костей и суставов, позволяет минимизировать необходимость внесения изменений в течение операции, что сокращает время пребывания на операционном столе (очень желательный результат как для пациента, так и врача). Анатомические векторные модели также используются в медицинских исследованиях и в хирургической практике.

Научные лаборатории продолжают генерировать новые идеи в области визуализации. Задача сообщества компьютерной графики состоит в создании удобных инструментов и эффективных технологий, позволяющих пользователям продолжать научные изыскания за границей возможного и безопасного эксперимента. Например, проект виртуального туннеля NASA Ames Research Center переносит аэродинамические данные в мир виртуальной реальности, интерес к которой значительно вырос в девяностые годы. NASA Ames было одним из пионеров в использовании и развитии технологий погружения людей в мнимую реальность. Специалисты NASA занимались разработкой специальных шлемов и дисплеев, трехмерных аудиоустройств, уникальных устройств ввода для оператора и созданием соответствующего программного обеспечения. Возник ряд компаний, занимающихся виртуальной реальностью, например: Fakespace, Cristal River Engineering и Telepresence Research.

Все эти инженерные и научные применения убеждают, что индустрия машинной графики начала обеспечивать пользователей новой технологией, при которой они действительно уже не заботятся о том, как формируется изображение – им важен результат.

Искусство, развлечения и бизнес

Согласно проведенным мною исследованиям, вплоть до начала девяностых годов доходы от использования векторной графики в научно-инженерных приложениях были значительно выше, чем доходы в области бизнеса и других областях, непосредственно не связанных с наукой. Однако в 1991 году доходы были поделены в равной степени, а баланс теперь устойчиво сдвигается в сторону нетехнических приложений. Я считаю, что к 1998 году около двух третей всех доходов от компьютерной графики поступит именно из нетехнических областей применения. Некоторые из этих применений получили настолько широкое распространение, что возникли споры, насколько они действительно являются машинной графикой. Например, мультимедиа воспринимают отдельно от машинной графики, что, однако, не так, вследствие явного доминирования графических изображений.

"Классическая" векторная графика до сих пор используется в различных приложениях бизнеса, включая разработку концепции, тестирование и создание новых продуктов, но бизнес также стал лидирующим потребителем систем мультимедиа, например, в обучении или маркетинговых презентациях. Графика все шире проникает в бизнес – сегодня фактически нет документов, созданных без использования какого-либо графического элемента. Соответствующее программное обеспечение специально разработано, чтобы позволить пользователям сконцентрироваться больше на содержании, а не на графическом исполнении.

Грядет всплеск использования графики в анимации, особенно в области индустрии развлечений. Кинофильм Стивена Спилберга "Парк Юрского периода" установил в 1993 году новый стандарт фотореализма в графике. Этот фильм не единичный случай применения 3D графики в кино, и Голливуд расширяет сферу использования специальных эффектов машинной графики, только в 1994 году выпустив несколько высокохудожественных фильмов: "The Lion King", "The Mask", "True Lies" и "Forrest Gump".

Виртуальная реальность находит свою нишу в индустрии развлечений и видеоиграх. Число виртуальных галерей и развлекательных парков быстро растет. По моим оценкам 30% (то есть 144 млрд. долл.) всего дохода от использования систем виртуальной реальности было получено в прошлом году именно от разного рода игр, и доходы от этих применений будут расти.

Лаборатория Media Lab МТИ является уникальным исследовательским центром разработки совершенных систем взаимодействия "человек-компьютер". Например, система News в проекте Future использует последние достижения в области графики, реконструкции звука и изображений, а также моделировании различных объектов для представления новых результатов исследований и их презентации в виде соответствующих текстов, графики, аудио и видео.

Векторная графика в Интернете

Ни для кого не секрет – сегодня, чтобы не затеряться на просторах Internet и привлечь к себе внимание пользователей, никак нельзя обойтись без графического оформления Web–страниц и узлов. Однако здесь на пути разработчиков возникает проблема: графические технологии для Web не поспевают в своем развитии за другими технологиями, и возможности в данной области остаются весьма ограниченными.

В самом деле, два наиболее популярных в настоящее время графических формата Internet – GIF и JPEG – являются уже довольно старыми. Конечно, неудачными назвать их никак нельзя, ведь сам факт столь длительного их существования – свидетельство этому. Но, с другой стороны, вряд ли можно поспорить с тем, что возможности данных форматов не отвечают современным требованиям в области графики. Так, формат GIF поддерживает только 256–битовый цвет, а в случае применения формата JPEG при большой степени сжатия существенно снижается качество изображения. Кроме того, еще в 1995 г. возможность свободного использования GIF оказалась под вопросом, когда компании Unisys, которой принадлежит реализованный в этом формате алгоритм сжатия LZW, и CompuServe, разработавшей сам формат, собрались взимать лицензионные отчисления с каждой программы, использующей его.

В сложившейся ситуации группа независимых разработчиков Internet приняла решение о разработке формата, который соответствовал бы или даже превосходил по своим возможностям GIF, но был при этом простым в создании и полностью мобильным. Новый формат получил название Portable Network Graphics (PNG) и был одобрен консорциумом W3C в 1996 г.

Формат PNG поддерживает 48–битовые цветные и 16–битовые черно–белые изображения и обеспечивает более быструю их загрузку, чем формат GIF. Он также включает в себя немало дополнительных возможностей, например альфа–каналы (alpha channel), позволяющие устанавливать уровень прозрачности для каждого пиксела, и гамма–коррекцию. Механизм сжатия изображения в PNG реализован на базе фильтров, позволяющих оптимизировать данные перед сжатием, и алгоритма LZ77, применяемого в ZIP–архиваторах.

Однако, несмотря на ряд преимуществ PNG пока не удалось стать реальной альтернативой GIF и JPEG. Виной тому было отсутствие поддержки со стороны разработчиков браузеров. Правда, к сегодняшнему дню в данном направлении произошли существенные сдвиги: начиная с Internet Explorer 4.0 и Netscape Navigator 4.04 поддержка PNG реализована непосредственно в браузерах; до этого она обеспечивалась за счет встраиваемых компонентов. По мнению ряда специалистов, вскоре можно ожидать широкого распространения нового формата (после массового перехода пользователей на последние версии популярных браузеров).

Следующим по популярности растровым форматом для Web можно назвать FlashPix, разработанный группой компаний: Kodak, Hewlett–Packard, Microsoft и Live Picture. Он базируется на принципах JPEG–компрессии, но содержит ряд усовершенствований, которые позволяют уменьшить степень искажения изображений. Основное преимущество данного формата – многоуровневая организация файла. В начале загружается изображение с самым низким разрешением и впоследствии, по мере надобности, подкачивается более качественная версия. Microsoft избрала модификацию этого формата в качестве основы для своего растрового редактора PhotoDraw 2000, так что в недалеком будущем следует ожидать поддержки его браузером Internet Explorer.

Интересной разработкой обладает компания Iterated Systems, которая создала свой формат на основе фрактальной компрессии (Fractal Image Format, FIF), а также выпустила программу преобразования основных форматов в FIF и плагины для просмотра сжатых по фрактальному алгоритму изображений в основных браузерах.

К сожалению, фрактальная компрессия, как и JPEG, имеет существенный недостаток: согласно этим алгоритмам, для анализа изображение перед сжатием разбивается на отдельные блоки, что затрудняет его постепенную прорисовку при загрузке с Web–сайта.

Наиболее перспективные – растровые форматы, основанные на алгоритмах wavelet–сжатия. В этой области ведут разработки практически все компании, которые занимаются созданием графических форматов. Самым многообещаемым является, безусловно, JPEG 2000. Работа над ним еще не завершена, но заявленные параметры впечатляют: 256 каналов цвета, что позволит формату работать с любым цветовым пространством и поддерживать множество альфа–каналов; встраивание ICC–профилей; неограниченное поле для метаданных. Но главное преимущество wavelet–технологии – потоковость. Wavelet–поток можно прерывать в любое время, при этом изображение все равно воспроизводится, только качество его будет зависеть

от количества загруженных данных.

Компания AT&T разработала и собственный формат на основе wavelet-компрессии – DjVu. Его главная особенность – распознавание текста при компрессии содержащих его изображений и сжатие отдельно графического и текстового слоя. По утверждению компании, основным предназначением этого формата и является публикация в Web сканированных документов. На сайте AT&T по адресу djvu.research.att.com можно получить бесплатный плагин для просмотра DjVu –файлов, а также целую библиотеку, опубликованную в этом формате.

Как бы ни были хороши вышеперечисленные форматы, всех их объединяет один недостаток – растр. Например, реализованные с их помощью изображения довольно сложно модифицировать и даже масштабировать. Кроме того, несмотря на использование различных методов сжатия, они все-таки имеют немалый размер, а следовательно, и относительно большое время загрузки, что для Web-графики является особенно критичным.

Векторная графика основана не на хранении информации о каждом пикселе, а на командах рисования линий и заполнения форм. Используется она уже довольно давно, но в отличие от традиционных замкнутых форматов векторные форматы для Web построены на базе открытых стандартов, главным образом языков маркировки, в которых для определения тегов и других элементов применяется обычный текст, что значительно упрощает манипулирование свойствами изображений. Преимуществами векторной графики на основе языков маркировки являются также возможности выбора, индексирования и поиска элементов изображения и привязки ее к другим элементам.

Однако говорить о массовом внедрении векторной графики в Web пока еще рано, в первую очередь из-за отсутствия единого формата.

Наиболее распространенным в данный момент является формат, разработанный компанией Macromedia, – Flash. Благодаря своим уникальным возможностям его последняя (третья) версия очень быстро завоевала популярность. Flash 3 поддерживает анимацию по кейфреймам, морфинг, прозрачные объекты, гиперссылки, встраивание звуковых и видеофайлов. Средства для его создания достаточно просты в использовании, хорошо документированы, плагины для просмотра распространяются бесплатно, а размер выходных файлов крайне мал.

Но все его преимущества, к сожалению, блекнут перед одним единственным недостатком, который заставил Macromedia отказаться от дальнейшей разработки формата. Этот недостаток – закрытость, ведь файл Flash – двоичный. Таким образом, его можно редактировать только в специальной программе. Поэтому в последнее время различными компаниями и организациями предложен целый ряд языковых форматов, и каждый из них претендует на роль единого стандарта. В число таких форматов входят Web Schematics, DrawML, PGML и VML.

Web Schematics представляет собой язык гипертекстовой маркировки для создания чертежей и диаграмм. Его разработчики попытались создать аналог функций рисования, используемых в базовых графических средствах систем воспроизведения документов, таких, как Adobe FrameMaker и Microsoft Word. Данный формат использует модели рендеринга и представлений HTML и CSS1.

Редактируемая двумерная графика для Интернета

Microsoft, Hewlett-Packard и три фирмы, специализирующиеся в области ПО, представили на рассмотрение консорциума World Wide Web стандарт, предназначенный для реализации высококачественной редактируемой двумерной векторной графики в Интернете.

С помощью языка векторной разметки (Vector Markup Language – VML) Web-дизайнеры смогут без труда редактировать, вырезать и вставлять векторные изображения в прикладные программы. Такая возможность, по мнению Стива Склеповича, менеджера по продуктам компании Microsoft, чрезвычайно нужна пользователям.

В отличие от растровой графики, представляющей собой изображения, образованные матрицами пикселей, векторные изображения состоят из линий, квадратов и других геометрических объектов.

До сих пор желающим дополнить свои Web-страницы векторной графикой приходилось пользоваться форматами растровых изображений, таких, как GIF, JPEG и PNG. Однако уменьшить или увеличить их размеры, даже расширяя или сжимая окно браузера, не удавалось.

Благодаря VML дизайнеры Web-узлов смогут изменять масштаб векторных изображений на Web-страницах и загружать графику значительно быстрее, чем изображения в растровом формате.

«Все отчаянно нуждаются в векторном графическом формате, – сказал Склепович. – Он действительно необходим. Программисты, работающие с векторной графикой, придумывали хитрые приемы и сохраняли изображения в растровом формате, чтобы обрабатывать их должным образом».

Формат VML появился на базе языка Extensible Markup Language (XML – расширяемый язык разметки), дополняющего HTML. Web-дизайнеры получили в свое распоряжение более гибкий инструмент, который позволяет создавать собственные теги, сообщают представители компаний.

Вместе с Microsoft и HP в консорциум W3C с предложением принять стандарт VML обратились компании AutoDesk, Macromedia и Visio Corp.

Они планируют в будущем выпускать продукты, использующие VML.

Например, Microsoft намерена предусмотреть возможность работы с VML в браузере Microsoft Explorer, операционной системе Windows и очередной версии Microsoft Office, заявил Склепович.

Это обеспечит взаимодействие офисного ПО и программ для проектирования и черчения.

В настоящее время компании используют для работы с векторной графикой различные стандарты.

Например, фирма Macromedia использует в своем инструментальном пакете векторной графики и анимации Flash собственный формат векторной графики под названием SWF. Компания Autodesk в пакете AutoCAD использует формат DWF.

Кроме того, благодаря VML пользователи смогут открывать и редактировать изображения, используя для этого пакет Office или в виде HTML-файла без потери качества, заявил Склепович.

Средства для создания векторных изображений

В настоящее время создано множество пакетов иллюстративной графики, которые содержат простые в применении, развитые и мощные инструментальные средства векторной графики, предназначенной как для подготовки материалов к печати, так и для создания страниц в Интернете.

Пакеты векторной или иллюстративной графики всегда основывались на объектно-ориентированном подходе, позволяющем рисовать контуры объектов, а затем закрашивать их или заполнять узорами.

В Microsoft Office предусмотрена линейка заданий, содержащая множество вариантов выбора для создания рисунков, логотипов и текста при работе с разными приложениями, а галерея WordArt предоставляет интересные и цветные стили текста, которыми можно пользоваться для заголовков или ярлыков.

Редакторы векторной графики:

Corel Draw

Пакет CorelDraw всегда производит сильное впечатление. В комплект фирма Corel включила множество программ, в том числе Corel Photo-Paint. Новый пакет располагает бесспорно самым мощным инструментарием среди всех программ обзора, а при этом по сравнению с предыдущей версией интерфейс стал проще, а инструментальные средства рисования и редактирования узлов – более гибкими. Однако что касается новых функций, в частности подготовки публикаций для Web, то здесь CorelDraw уступает CorelXara.

Adobe Illustrator

Corel Xara

CorelXara – служит в первую очередь для создания графического изображения на странице за один раз и формирования блока текста за один раз. Программа позволяет выполнять с рисунками, градиентным заполнением, изображениями и диапозитивами такие действия, о которых вы могли только мечтать. Хотя Corel рекламирует CorelXara как дополнение к CorelDraw 7 для создания графики Web, по существу благодаря высокой производительности, средствам для работы с Web и специализированному инструментарию CorelXara превосходит CorelDraw во многих отношениях. Благодаря возможностям масштабирования векторной графики и текстурам растровых изображений двумерные объекты начинают все более напоминать трехмерные.

Canvas

Пытаясь объединить возможности рисования, создания растровых изображений, редактирования и верстки страниц в одной программе, фирма Deneba Systems разработала пакет Canvas, который, реализуя многочисленные функции, ни одну из них не может выполнить безупречно.

Canvas, при всех честолюбивых замыслах его разработчиков, не может считаться полноценным пакетом иллюстративной графики. И хотя утверждается, что Canvas объединяет в себе множество возможностей, на самом деле вам уже в самом начале потребуется выбрать тип документа, который вы собираетесь создавать.

Важным преимуществом программ векторной графики является развитые средства интеграции изображений и текста, единый подход к ним. Поэтому программы векторной графики незаменимы в области дизайна, технического рисования, для чертежно-графических и оформительских работ.

Растровая и векторная графика

Таким образом, выбор растрового или векторного формата зависит от целей и задач работы с изображением. Если нужна фотографическая точность цветопередачи, то предпочтительнее растр. Логотипы, схемы, элементы оформления удобнее представлять в векторном формате. Понятно, что и в растровом и в векторном представлении графика (как и текст) выводятся на экран монитора или печатное устройство в виде совокупности точек. В Интернете графика представляется в одном из растровых форматов, понимаемых браузерами без установки дополнительных модулей – GIF, JPG, PNG.

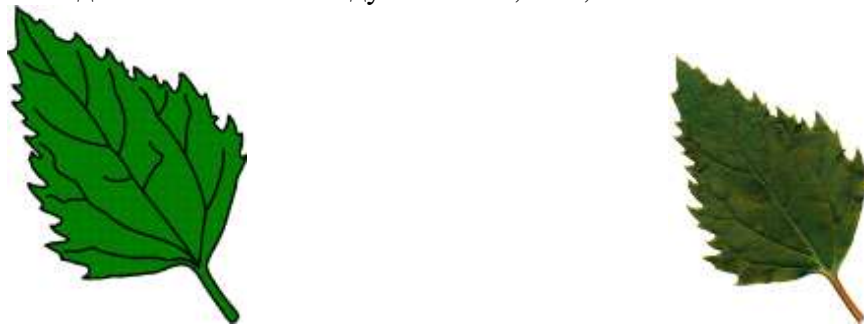


Рис. 1. Пример векторного и растрового изображения

Без дополнительных плагинов (дополнений) наиболее распространенные браузеры понимают только растровые форматы – .gif, .jpg и .png (последний пока мало распространен). На первый взгляд, использование векторных редакторов становится неактуальным. Однако большинство таких редакторов обеспечивают экспорт в .gif или .jpg с выбираемым Вами разрешением. А рисовать начинающим художникам проще именно в векторных средах – если рука дрогнула и линия пошла не туда, получившийся элемент легко редактируется. При рисовании в растровом режиме Вы рискуете непоправимо испортить фон.

Из-за описанных выше особенностей представления изображения, для каждого типа приходится использовать отдельный графический редактор – растровый или векторный. Разумеется, у них есть общие черты – возможность открывать и сохранять файлы в различных форматах, использование инструментов с одинаковыми названиями (карандаш, перо и т.д.) или функциями (выделение, перемещение, масштабирование и т.д.), выбирать нужный цвет или оттенок... Однако принципы реализации процессов рисования и редактирования различны и обусловлены природой соответствующего формата. Так, если в растровых редакторах говорят о выделении объекта, то имеют в виду совокупность точек в виде области сложной формы. Процесс выделения очень часто является трудоемкой и кропотливой работой. При перемещении такого выделения появляется «дырка». В векторном же редакторе объект представляет совокупность графических примитивов и для его выделения достаточно выбрать мышкой каждый из них. А если эти примитивы были сгруппированы соответствующей командой, то достаточно «щелкнуть» один раз в любой из точек сгруппированного объекта. Перемещение выделенного объекта обнажает нижележащие элементы.

ТЕМА № 7. ТРЕХМЕРНАЯ ГРАФИКА

Основные понятия трехмерной графики. Области применения трехмерной графики. Программные средства обработки трехмерной графики.

Основные понятия трехмерной графики

Трехмерная графика нашла широкое применение в таких областях, как научные расчеты, инженерное проектирование, компьютерное моделирование физических объектов.

Для создания реалистичной модели объекта используются геометрические примитивы (куб, шар, конус и пр.) и гладкие, так называемые **сплайновые поверхности**. Вид поверхности определяется расположенной в пространстве сеткой опорных точек. Каждой точке присваивается коэффициент, величина которо-

го определяет степень ее влияния на часть поверхности, проходящей вблизи точки. От взаимного расположения точек и величины коэффициентов зависит форма и гладкость поверхности в целом.

Деформация объекта обеспечивается перемещением контрольных точек, расположенных вблизи. Каждая контрольная точка связана с ближайшими опорными точками, степень ее влияния на них определяется удаленностью. Другой метод называют сеткой деформации. Вокруг объекта или его части размещается трехмерная сетка, перемещение любой точки которой вызывает упругую деформацию как самой сетки, так и окруженного объекта.

Еще одним способом построения объектов из примитивов служит твердотельное моделирование. Объекты представлены твердыми телами, которые при взаимодействии с другими телами различными способами (объединение, вычитание, слияние и др.) претерпевают необходимую трансформацию.

Все многообразие свойств в компьютерном моделировании сводится к визуализации поверхности, то есть к расчету коэффициента прозрачности поверхности и угла преломления лучей света на границе материала и окружающего пространства. Свойства поверхности описываются в создаваемых массивах текстур, в которых содержатся данные о степени прозрачности материала, коэффициенте преломления, цвете в каждой точке, цвете блика, его ширине и резкости и др.

После завершения конструирования и визуализации объекта приступают его "оживлению", т.е. заданию параметров движения. Компьютерная анимация базируется на ключевых кадрах.

Применение сложных математических моделей позволяет имитировать различные физические эффекты: взрывы, дождь, снег, огонь, дым, туман и др.

Основную долю рынка программных средств обработки трехмерной графики занимают три пакета: 3D Studio Max фирмы Kinetix; Softimage 3D компании Microsoft; Maya, разработанная консорциумом известных компаний (Alias, Wavefront, TDI). На сегодняшний день Maya является наиболее передовым пакетом в классе средств создания и обработки трехмерной графики для персональных компьютеров.

Трехмерная графика нашла широкое применение в таких областях, как научные расчеты, инженерное проектирование, компьютерное моделирование физических объектов (рис. 3). В качестве примера рассмотрим наиболее сложный вариант трехмерного моделирования – создание подвижного изображения реального физического тела.

В упрощенном виде для пространственного моделирования объекта требуется:

- спроектировать и создать виртуальный каркас (“скелет”) объекта, наиболее полно соответствующий его реальной форме;
- спроектировать и создать виртуальные материалы, по физическим свойствам визуализации похожие на реальные;
- присвоить материалы различным частям поверхности объекта (на профессиональном жаргоне – “спроектировать текстуры на объект”);
- настроить физические параметры пространства, в котором будет действовать объект, – задать освещение, гравитацию, свойства атмосферы, свойства взаимодействующих объектов и поверхностей;
- задать траектории движения объектов;
- рассчитать результирующую последовательность кадров;
- наложить поверхностные эффекты на итоговый анимационный ролик.

Для создания реалистичной модели объекта используют геометрические примитивы (прямоугольник, куб, шар, конус и прочие) и гладкие, так называемые *сплайновые поверхности*. В последнем случае применяют чаще всего метод *бикубических рациональных B-сплайнов на неравномерной сетке (NURBS)*. Вид поверхности при этом определяется расположенной в пространстве сеткой опорных точек. Каждой точке присваивается коэффициент, величина которого определяет степень ее влияния на часть поверхности, проходящей вблизи точки. От взаимного расположения точек и величины коэффициентов зависит форма и “гладкость” поверхности в целом.

После формирования “скелета” объекта необходимо покрыть его поверхность материалами. Все многообразие свойств в компьютерном моделировании сводится к визуализации поверхности, то есть к расчету коэффициента прозрачности поверхности и угла преломления лучей света на границе материала и окружающего пространства.

Закраска поверхностей осуществляется методами Гуро (*Gouraud*) или Фонга (*Phong*). В первом случае цвет примитива рассчитывается лишь в его вершинах, а затем линейно интерполируется по поверхности. Во втором случае строится нормаль к объекту в целом, ее вектор интерполируется по поверхности составляющих примитивов и освещение рассчитывается для каждой точки.

Свет, уходящий с поверхности в конкретной точке в сторону наблюдателя, представляет собой сумму компонентов, умноженных на коэффициент, связанный с материалом и цветом поверхности в данной точке. К таковым компонентам относятся:

- свет, пришедший с обратной стороны поверхности, то есть преломленный свет (Refracted);
- свет, равномерно рассеиваемый поверхностью (Diffuse);
- зеркально отраженный свет (Reflected);
- блики, то есть отраженный свет источников (Specular);
- собственное свечение поверхности (Self Illumination).

Следующим этапом является наложение (“проектирование”) текстур на определенные участки каркаса объекта. При этом необходимо учитывать их взаимное влияние на границах примитивов. Проектирование материалов на объект – задача трудно формализуемая, она сродни художественному процессу и требует от исполнителя хотя бы минимальных творческих способностей.

После завершения конструирования и визуализации объекта приступают к его “оживлению”, то есть заданию параметров движения. Компьютерная анимация базируется на ключевых кадрах. В первом кадре объект выставляется в исходное положение. Через определенный промежуток (например, в восьмом кадре) задается новое положение объекта и так далее до конечного положения. Промежуточные значения вычисляет программа по специальному алгоритму. При этом происходит не просто линейная аппроксимация, а плавное изменение положения опорных точек объекта в соответствии с заданными условиями.

Эти условия определяются иерархией объектов (то есть законами их взаимодействия между собой), разрешенными плоскостями движения, предельными углами поворотов, величинами ускорений и скоростей. Такой подход называют методом *инверсной кинематики движения*. Он хорошо работает при моделировании механических устройств. В случае с имитацией живых объектов используют так называемые *скелетные модели*. То есть, создается некий каркас, подвижный в точках, характерных для моделируемого объекта. Движения точек просчитываются предыдущим методом. Затем на каркас накладывается оболочка, состоящая из смоделированных поверхностей, для которых каркас является набором контрольных точек, то есть создается *каркасная модель*. Каркасная модель визуализуется наложением поверхностных текстур с учетом условий освещения. В ходе перемещения объекта получается весьма правдоподобная имитация движений живых существ.

Наиболее совершенный метод анимации заключается в фиксации реальных движений физического объекта. Например, на человеке закрепляют в контрольных точках яркие источники света и снимают заданное движение на видео- или кинопленку. Затем координаты точек по кадрам переводят с пленки в компьютер и присваивают соответствующим опорным точкам каркасной модели. В результате движения имитируемого объекта практически неотличимы от живого прототипа.

Процесс расчета реалистичных изображений называют *рендерингом* (визуализацией). Большинство современных программ рендеринга основаны на *методе обратной трассировки лучей (Backway Ray Tracing)*. Применение сложных математических моделей позволяет имитировать такие физические эффекты, как взрывы, дождь, огонь, дым, туман. По завершении рендеринга компьютерную трехмерную анимацию используют либо как самостоятельный продукт, либо в качестве отдельных частей или кадров готового продукта.

Особую область трёхмерного моделирования в режиме реального времени составляют тренажеры технических средств – автомобилей, судов, летательных и космических аппаратов. В них необходимо очень точно реализовывать технические параметры объектов и свойства окружающей физической среды. В более простых вариантах, например при обучении вождению наземных транспортных средств, тренажеры реализуют на персональных компьютерах.

Самые совершенные на сегодняшний день устройства созданы для обучения пилотированию космических кораблей и военных летательных аппаратов. Моделированием и визуализацией объектов в таких тренажерах заняты несколько специализированных графических станций, построенных на мощных RISC-процессорах и скоростных видеоадаптерах с аппаратными ускорителями трехмерной графики. Общее управление системой и просчет сценариев взаимодействия возложены на суперкомпьютер, состоящий из десятков и сотен процессоров. Стоимость таких комплексов выражается девятизначными цифрами, но их применение окупается достаточно быстро, так как обучение на реальных аппаратах в десятки раз дороже.

Программные средства обработки трехмерной графики

На персональных компьютерах основную долю рынка программных средств обработки трехмерной

графики занимают три пакета. Эффективней всего они работают на самых мощных машинах (в двух- или четырехпроцессорных конфигурациях Pentium II/III, Xeon) под управлением операционной системы Windows NT.

Программа создания и обработки трехмерной графики 3D Studio Max фирмы Kinetix изначально создавалась для платформы Windows. Этот пакет считается «полупрофессиональным». Однако его средств вполне хватает для разработки качественных трехмерных изображений объектов неживой природы. Отличительными особенностями пакета являются поддержка большого числа аппаратных ускорителей трехмерной графики, мощные световые эффекты, большое число дополнений, созданных сторонними фирмами. Сравнительная нетребовательность к аппаратным ресурсам позволяет работать даже на компьютерах среднего уровня. Вместе с тем по средствам моделирования и анимации пакет 3D Studio Max уступает более развитым программным средствам.

Программа Softimage 3D компании Microsoft изначально создавалась для рабочих станций SGI и лишь сравнительно недавно была конвертирована под операционную систему Windows NT. Программу отличают богатые возможности моделирования, наличие большого числа регулируемых физических и кинематографических параметров. Для рендеринга применяется качественный и достаточно быстрый модуль Mental Ray. Существует множество дополнений, выпущенных «третьими» фирмами, значительно расширяющих функции пакета. Эта программа считается стандартом «де-факто» в мире специализированных графических станций SGI, а на платформе IBM PC выглядит несколько тяжеловато и требует мощных аппаратных ресурсов.

Наиболее революционной с точки зрения интерфейса и возможностей является программа Maya, разработанная консорциумом известных компаний (Alias, Wavefront, TDI). Пакет существует в вариантах для разных операционных систем, в том числе и Windows NT. Инструментарий Maya сведен в четыре группы: Animation (анимация), Modeling (моделирование), Dynamic (физическое моделирование), Rendering (визуализация). Удобный настраиваемый интерфейс выполнен в соответствии с современными требованиями. На сегодняшний день Maya является наиболее передовым пакетом в классе средств создания и обработки трехмерной графики для персональных компьютеров.

ТЕМА № 8. БАЗОВЫЕ РАСТРОВЫЕ АЛГОРИТМЫ

Алгоритм вывода прямой линии. Алгоритм вывода окружности. Алгоритм вывода эллипса. Алгоритмы вывода фигур. Алгоритмы закрашивания (простейший алгоритм закрашивания, волновой алгоритм, алгоритм закрашивания линиями). Заполнение прямоугольников. Заполнение круга. Алгоритм вывода толстой и пунктирной линии.

Алгоритм вывода прямой линии

Поскольку экран растрового дисплея с электронно-лучевой трубкой (ЭЛТ) можно рассматривать как матрицу дискретных элементов (пикселей), каждый из которых может быть подсвечен, нельзя непосредственно провести отрезок из одной точки в другую. Процесс определения пикселей, наилучшим образом аппроксимирующих заданный отрезок, называется разложением в растр. В сочетании с процессом построчной визуализации изображения он 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, что показано на рис.1.

Общие требования к алгоритмам вычерчивания отрезков следующие: Отрезки должны выглядеть прямыми, начинаться и заканчиваться в заданных точках, яркость вдоль отрезка должна быть постоянной и не зависеть от длины и наклона, рисовать нужно быстро.

Постоянная вдоль всего отрезка яркость достигается лишь при проведении горизонтальных, вертикальных и наклоненных под углом 45° прямых. Для всех других ориентаций разложение в растр приведет к неравномерности яркости, как это показано на рис. 1.

В большинстве алгоритмов вычерчивания отрезков для упрощения вычислений используется пошаговый алгоритм. Приведем пример подобного алгоритма:

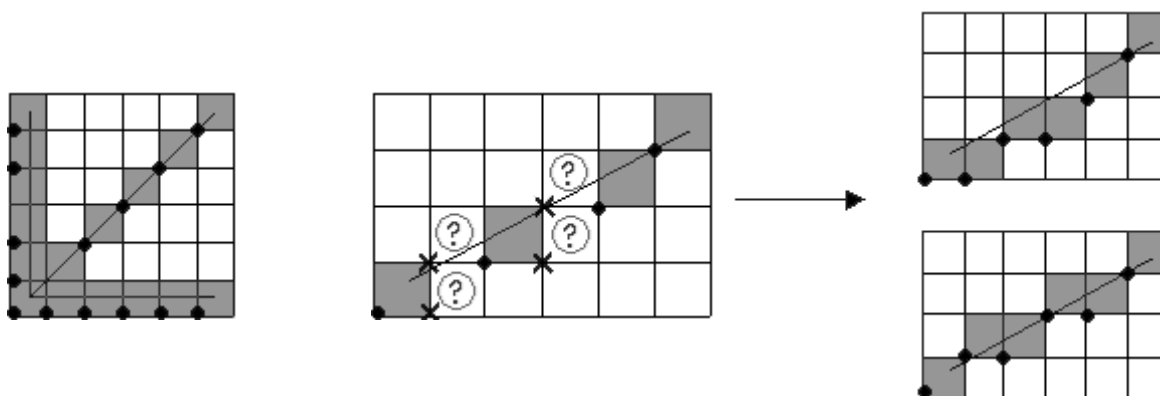


Рис.1.1. Разложение в растр отрезков прямых.

Простой пошаговый алгоритм

```

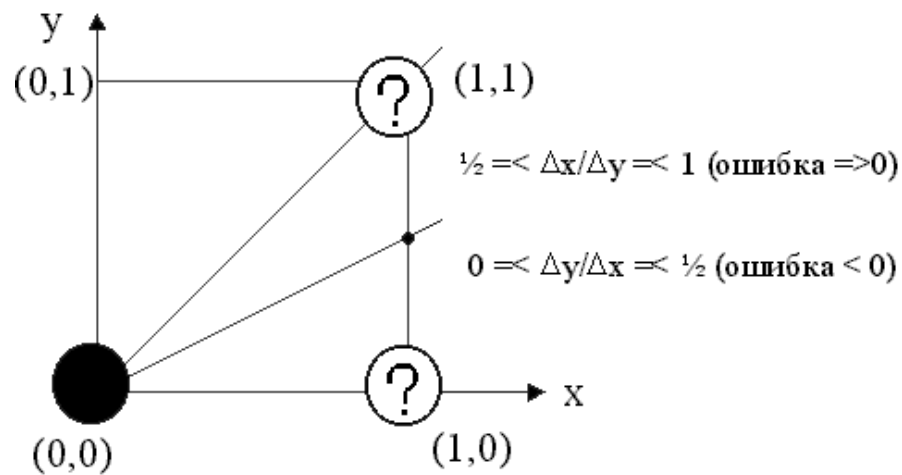
позиция = начало
шаг = приращение
1. if позиция - конец < точность then 4
if позиция > конец then 2
if позиция < конец then 3
2. позиция = позиция - шаг
go to 1
3. позиция = позиция + шаг
go to 1
4. finish

```

Алгоритм Брезенхема.

Хотя алгоритм Брезенхема был первоначально разработан для цифровых графопостроителей, однако он в равной степени подходит для использования растровыми устройствами с ЭЛТ. Алгоритм выбирает оптимальные растровые координаты для представления отрезка. В процессе работы одна из координат - либо x , либо y (в зависимости от углового коэффициента) - изменяется на единицу. Изменение другой координаты (на 0 или 1) зависит от расстояния между действительным положением отрезка и ближайшими координатами сетки. Такое расстояние мы назовем ошибкой.

Алгоритм построен так, что требуется проверить лишь знак этой ошибки. На рис.3.1 это иллюстрируется для отрезка в первом октанте, т.е. для отрезка с угловым коэффициентом, лежащим в диапазоне от 0 до 1. Из рисунка можно заметить, что если угловой коэффициент отрезка из точки $(0,0)$ больше, чем $1/2$, то пересечение с прямой $x = 1$ будет расположено ближе к прямой $y = 1$, чем к прямой $y = 0$. Следовательно, точка раstra $(1,1)$ лучше аппроксимирует ход отрезка, чем точка $(1,0)$. Если угловой коэффициент меньше $1/2$, то верно обратное. для углового коэффициента, равного $1/2$, нет какого либо предпочтительного выбора. В данном случае алгоритм выбирает точку $(1,1)$.



Инициализировать ошибку в $-1/4$
 $\text{ошибка} = \text{ошибка} + \Delta y / \Delta x$

Рис. 3.1. Основная идея алгоритма Брезенхема.

Не все отрезки проходят через точки раstra. Подобная ситуация иллюстрируется рис.3.2, где отрезок с тангенсом угла наклона $3/8$ сначала подходит через точку раstra $(0,0)$ и последовательно пересекает три пиксела. Также иллюстрируется вычисление ошибки при представлении отрезка дискретными пикселями.

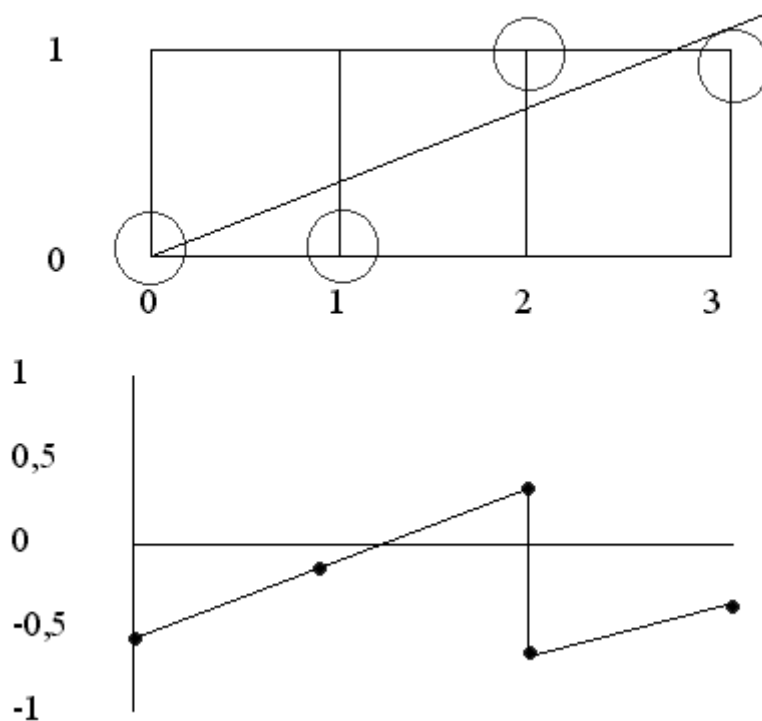


Рис.3.2. График ошибки в алгоритме Брезенхема.

Так как желательно проверять только знак ошибки, то она первоначально устанавливается равной $-1/2$. Таким образом, если угловой коэффициент отрезка больше или равен $1/2$, то величина ошибки в следующей точке раstra с координатами $(1,0)$ может быть вычислена как

$$e = e + m$$

где m - угловой коэффициент. В нашем случае при начальном значении ошибки $-1/2$

$$e = 1/2 + 3/8 = -1/8$$

Так как e отрицательно, отрезок пройдет ниже середины пиксела. Следовательно, пиксел на том же самом горизонтальном уровне лучше аппроксимирует положение отрезка, поэтому y не увеличивается. Аналогично вычисляем ошибку

$$e = -1/8 + 3/8 = 1/4$$

в следующей точке раstra $(2,0)$. Теперь e положительно, значит отрезок пройдет выше средней точ-

ки. Растровый элемент (2,1) со следующей по величине координатой y лучше аппроксимирует положение отрезка. Следовательно y увеличивается на 1. Прежде чем рассматривать следующий пиксел, необходимо откорректировать ошибку вычитанием из нее 1. Имеем

$$e = 1/4 - 1 = -3/4$$

Заметим, что пересечение вертикальной прямой $x = 2$ с заданным отрезком лежит на $1/4$ ниже прямой $y = 1$. Если же перенести отрезок $1/2$ вниз, мы получим как раз величину $-3/4$. Продолжение вычислений для следующего пиксела дает

$$e = -3/4 + 3/8 = -3/8$$

Так как e отрицательно, то y не увеличивается. Из всего сказанного следует, что ошибка - это интервал, отсекаемый по оси y рассматриваемым отрезком в каждом растровом элементе (относительно $-1/2$).

Приведем алгоритм Брезенхема для первого октанта, т.е. для случая $0 \leq x \leq x_2$.

Алгоритм Брезенхема разложения в растр отрезка для первого октанта

предполагается, что концы отрезка (x_1, y_1) и (x_2, y_2) не совпадают

Integer - функция преобразования в целое

$x, y, \Delta x, \Delta y$ - целые

e - вещественное

инициализация переменных

$x = x_1$

$y = y_1$

$\Delta x = x_2 - x_1$

$\Delta y = y_2 - y_1$

Инициализация с поправкой на половину пиксела

$e = \Delta y / \Delta x - 1/2$

начало основного цикла

for $i = 1$ to Δx

plot (x, y)

while $(e \geq 0)$

$y = y + 1$

$e = e - 1$

end while

$x = x + 1$

$e = e + \Delta y / \Delta x$

next i

finish

Блок-схема алгоритма приводится на рис.3.3. Пример приведен ниже.

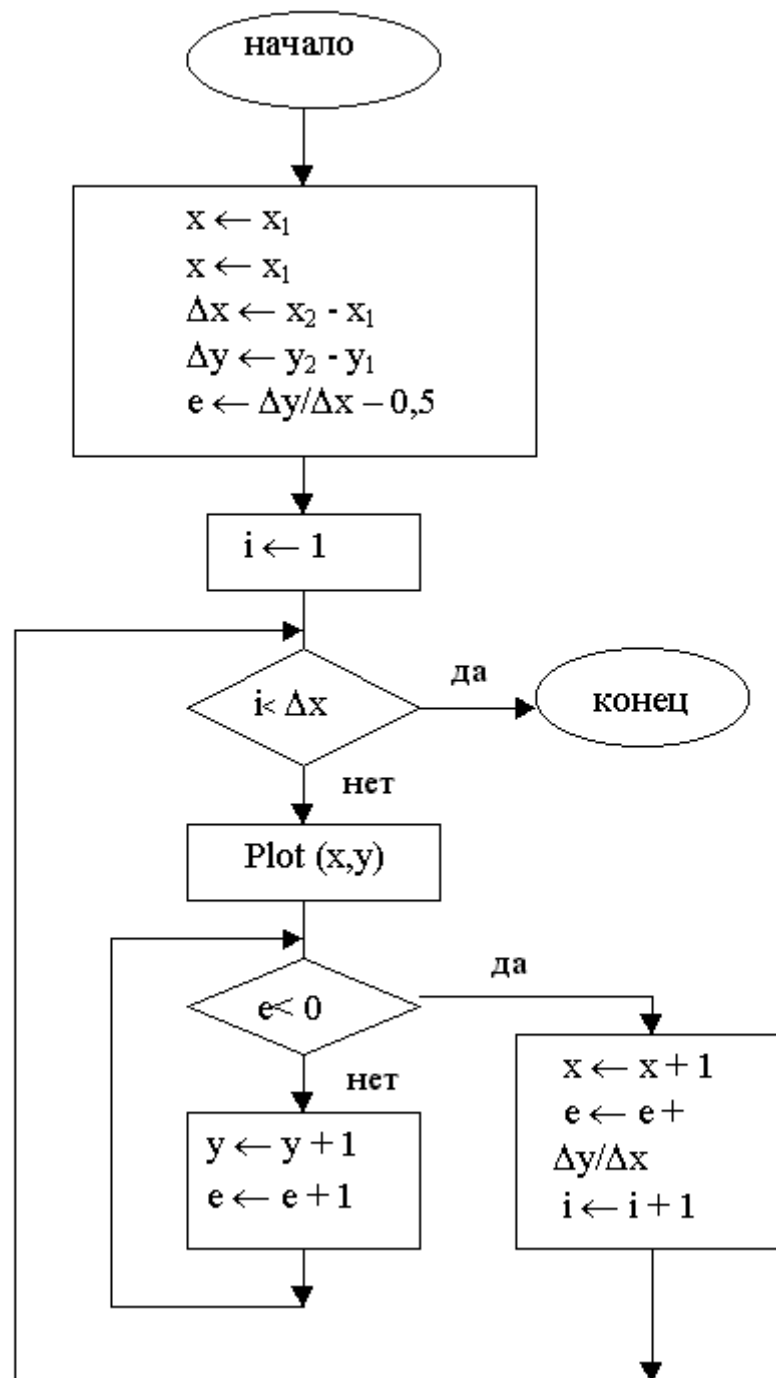


Рис. 3.3. Блок-схема алгоритма Брезенхема.

Пример 3.1. Алгоритм Брезенхема.

Рассмотрим отрезок проведенный из точки (0,0) в точку (5,5). Разложение отрезка в растр по алгоритму Брезенхема приводит к такому результату:

начальные установки

$x = 0$

$y = 0$

$\square x = 5$

$\square y = 5$

$e = 1 - 1/2 = 1/2$

результаты работы пошагового цикла

	Plot	e		
		1/2		
	(0,0)			
		-1/2		

		1/2		
	(1,1)			
		-1/2		
		1/2		
	(2,2)			
		-1/2		
		1/2		
	(3,3)			
		-1/2		
		1/2		
	(4,4)			
		-1/2		
		1/2		

Результат показан на рис.3.4 и совпадает с ожидаемым. Заметим, что точка растра с координатами (5,5) не активирована. Эту точку можно активировать путем изменения цикла for-next на 0 to $\square x$. Активацию точки (0,0) можно устранить, если поставить оператор Plot непосредственно перед строкой next i.

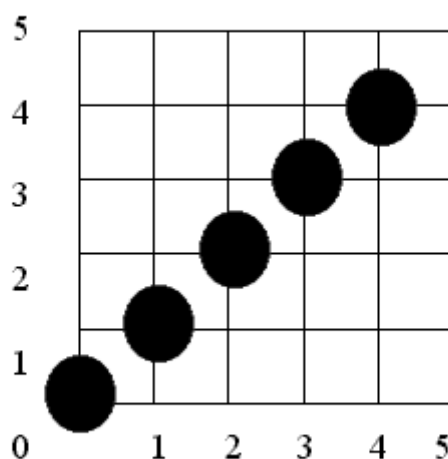


Рис. 3.4. Результат работы алгоритма Брезенхема в первом октанте.

В следующем разделе описан общий алгоритм Брезенхема.

4. Общий алгоритм Брезенхема.



Чтобы реализация алгоритма Брезенхема была полной необходимо обрабатывать отрезки во всех октантах. Модификацию легко сделать, учитывая в алгоритме номер квадранта, в котором лежит отрезок и его угловой коэффициент. Когда абсолютная величина углового коэффициента больше 1, y постоянно изменяется на единицу, а критерий ошибки Брезенхема используется для принятия решения об изменении величины x . Выбор постоянно изменяющейся (на +1 или -1) координаты зависит от квадранта (рис.4.1.). Общий алгоритм может быть оформлен в следующем виде:

Обобщенный целочисленный алгоритм Брезенхема квадрантов

предполагается, что концы отрезка (x_1, y_1) и (x_2, y_2) не совпадают
все переменные считаются целыми

Sign - функция, возвращающая -1, 0, 1 для отрицательного, нулевого и положительного аргумента соответственно

инициализация переменных

$x = x_1$

$y = y_1$

$\Delta x = \text{abs}(x_2 - x_1)$

$\Delta y = \text{abs}(y_2 - y_1)$

$s1 = \text{Sign}(x_2 - x_1)$

```

s2 = Sign(y2 - y1)
обмен значений  $\Delta x$  и  $\Delta y$  в зависимости от углового коэффициента наклона отрезка
if  $\Delta y < \Delta x$  then
  Врем =  $\Delta x$ 
   $\Delta x = \Delta y$ 
   $\Delta y = \text{Врем}$ 
  Обмен = 1
else
  Обмен = 0
end if
инициализация  $\square$  с поправкой на половину пиксела
e =  $2 * \Delta y - \Delta x$ 
основной цикл
for i = 1 to  $\Delta x$ 
  Plot(x,y)
  while(e  $\geq$  0)
    if Обмен = 1 then
      x = x + s1
    else
      y = y + s2
    end if
    e = e -  $2 * \Delta x$ 
  end while
  if Обмен = 1 then
    y = y + s2
  else
    x = x + s1
  end if
  e = e +  $2 * \Delta y$ 
next i
finish

```

Пример 4.1. обобщенный алгоритм Брезенхема.

Для иллюстрации рассмотрим отрезок из точки (0,0) в точку (-8, -4).

начальные установки

```

x = 0
y = 0
 $\Delta x = 8$ 
 $\Delta y = 4$ 
s1 = -1
s2 = -1
Обмен = 0
e = 0

```

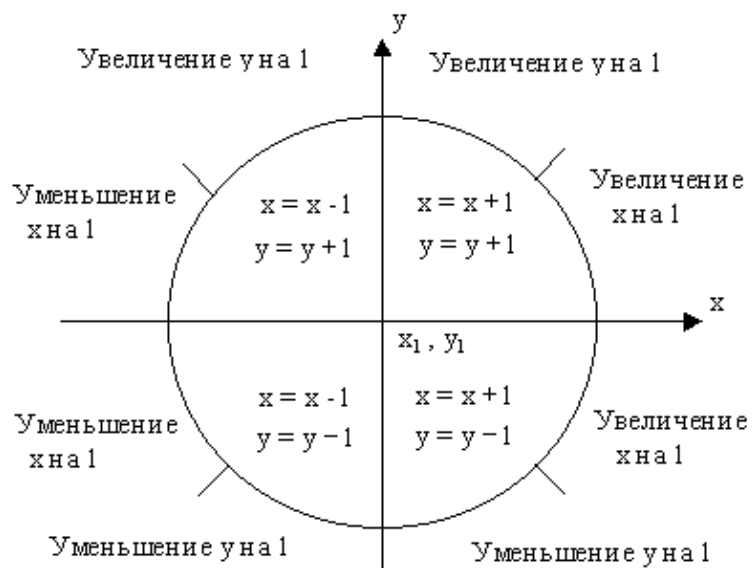


Рис.4.1. Разбор случаев для обобщенного алгоритма Брезенхема.
результаты работы пошагового цикла

i	Plot	e	x	y
		0	0	0
1	(0,0)			
		-16	0	-1
		-8	-1	-1
2	(-1,-1)			
		0	-2	-1
3	(-2,-1)			
		-16	-2	-2
		-8	-3	-2
4	(-3,2)			
		0	-4	-2
5	(-4,2)			
		-16	-4	-3
		-8	-5	-3
6	(-5,-3)			
		0	-6	-3
7	(-6,-3)			
		-16	-6	-4
		-8	-7	-4
8	(-7,-4)			
		0	-8	-4

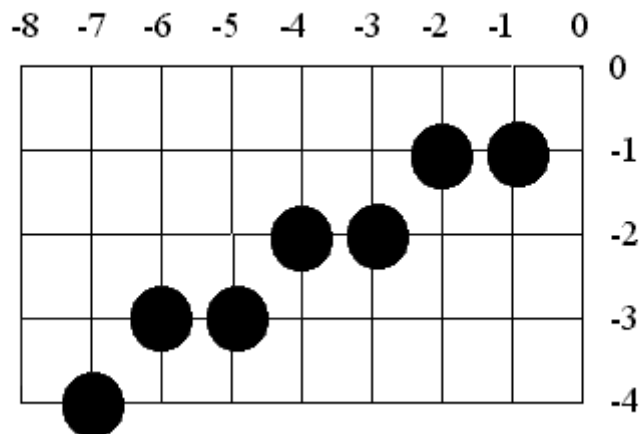


Рис.4.2. Результат работы обобщенного алгоритма Брезенхема в третьем квадранте.

На рис. 4.2 продемонстрирован результат. Сравнение с рис. 2.2 показывает, что результаты работы двух алгоритмов отличаются.

В [следующем разделе](#) рассматривается алгоритм Брезенхема для генерации окружности.

Алгоритм Брезенхема для генерации окружности.

В растр нужно разлагать не только линейные, но и другие, более сложные функции. Разложению конических сечений, т. е. окружностей, эллипсов, парабол, гипербол, было посвящено значительное число работ. Наибольшее внимание, разумеется, уделено окружности. Один из наиболее эффективных и простых для понимания алгоритмов генерации окружности принадлежит Брезенхему. Для начала заметим, что необходимо сгенерировать только одну восьмую часть окружности. Остальные ее части могут быть получены последовательными отражениями, как это показано на рис. 5.1. Если сгенерирован первый октант (от 0 до 45° против часовой стрелки), то второй октант можно получить зеркальным отражением относительно прямой $y = x$, что дает в совокупности первый квадрант. Первый квадрант отражается относительно прямой $x = 0$ для получения соответствующей части окружности во втором квадранте. Верхняя полуокружность отражается относительно прямой $y = 0$ для завершения построения. На рис. 5.1 приведены двумерные матрицы соответствующих преобразований.

Для вывода алгоритма рассмотрим первую четверть окружности с центром в начале координат. Заметим, что если работа алгоритма начинается в точке $x = 0, y = R$, то при генерации окружности по часовой стрелке в первом квадранте y является монотонно убывающей функцией аргумента (рис. 5.2). Аналогично, если исходной точкой является $y = 0, x = R$, то при генерации окружности против часовой стрелки x будет монотонно убывающей функцией аргумента y . В нашем случае выбирается генерация по часовой стрелке с началом в точке $x = 0, y = R$. Предполагается, что центр окружности и начальная точка находятся точно в точках раstra.

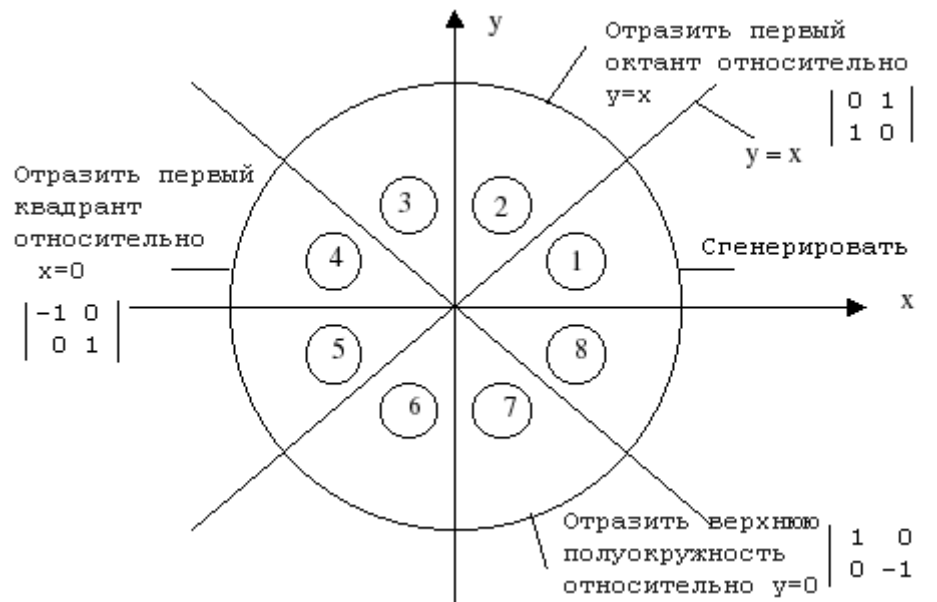


Рис. 5.1. Генерация полной окружности из дуги в первом октанте.

Для любой заданной точки на окружности при генерации по часовой стрелке существует только три возможности выбрать следующий пиксел, наилучшим образом приближающий окружность: горизонтально вправо, по диагонали вниз и вправо, вертикально вниз. На рис. 5.3 эти направления обозначены соответственно m_H , m_D , m_V . Алгоритм выбирает пиксел, для которого минимален квадрат расстояния между одним из этих пикселов и окружностью, т. е. минимум из

$$m_H = |(x_i + 1)^2 + (y_i)^2 - R^2|$$

$$m_D = |(x_i + 1)^2 + (y_i - 1)^2 - R^2|$$

$$m_V = |(x_i)^2 + (y_i - 1)^2 - R^2|$$

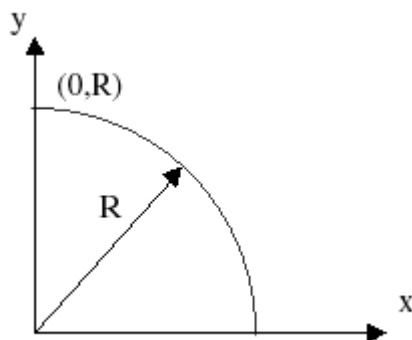


Рис. 5.2. Окружность в первом квадранте.

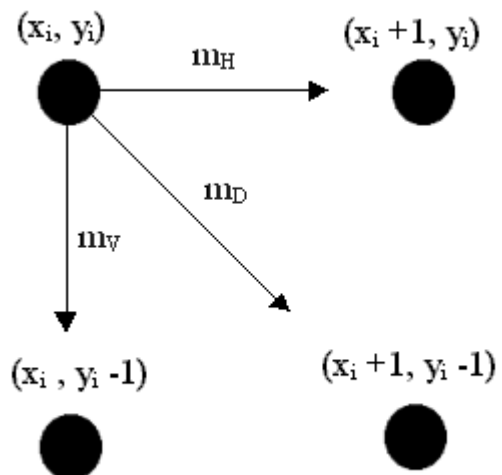


Рис.5.3. Выбор пикселов в первом квадранте.

Вычисления можно упростить, если заметить, что в окрестности точки (x_i, y_i) возможны только пять типов пересечений окружности и сетки растра, приведенных на рис. 5.4.

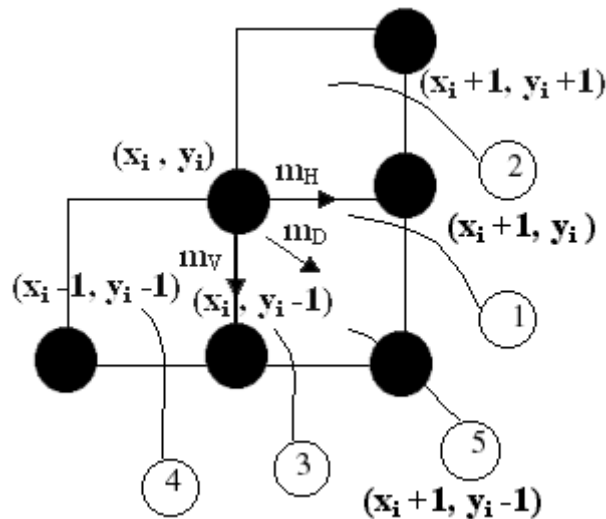


Рис. 5.4. Пересечение окружности и сетки раstra.

Разность между квадратами расстояний от центра окружности до диагонального пиксела $(x_i + 1, y_i - 1)$ и от центра до точки на окружности R^2 равна

$$\Delta_i = (x_i + 1)^2 + (y_i - 1)^2 - R^2$$

Как и в алгоритме Брезенхема для отрезка, для выбора соответствующего пиксела желательно использовать только знак ошибки, а не ее величину.

При $\Delta_i < 0$ диагональная точка $(x_i + 1, y_i - 1)$ находится внутри реальной окружности, т. е. это случаи 1 или 2 на рис. 5.4. Ясно, что в этой ситуации следует выбрать либо пиксел $(x_i + 1, y_i)$, т. е. m_H , либо пиксел $(x_i + 1, y_i - 1)$, т. е. m_D . Для этого сначала рассмотрим случай 1 и проверим разность квадратов расстояний от окружности до пикселов в горизонтальном и диагональном направлениях:

$$E = |(x_i + 1)^2 + (y_i)^2 - R^2| - |(x_i + 1)^2 + (y_i - 1)^2 - R^2|$$

При $E < 0$ расстояние от окружности до диагонального пиксела больше, чем до горизонтального. Напротив, если $E > 0$, расстояние до горизонтального пиксела больше. Таким образом,

при $E \leq 0$ выбираем m_H в $(x_i + 1, y_i)$

при $E > 0$ выбираем m_D в $(x_i + 1, y_i - 1)$

При $E = 0$, когда расстояние от окружности до обоих пикселов одинаковы, выбираем горизонтальный шаг.

Количество вычислений, необходимых для оценки величины E , можно сократить, если заметить, что в случае 1

$$(x_i + 1)^2 + (y_i)^2 - R^2 \geq 0$$

$$(x_i + 1)^2 + (y_i - 1)^2 - R^2 < 0$$

так как диагональный пиксел $(x_i + 1, y_i - 1)$ всегда лежит внутри окружности, а горизонтальный $(x_i + 1, y_i)$ - вне ее. Таким образом, E можно вычислить по формуле

$$E = (x_i + 1)^2 + (y_i)^2 - R^2 + (x_i + 1)^2 + (y_i - 1)^2 - R^2$$

Дополнение до полного квадрата члена $(y_i)^2$ с помощью добавления и вычитания $-2y_i + 1$ дает

$$E = 2[(x_i + 1)^2 + (y_i - 1)^2 - R^2] + 2y_i - 1$$

В квадратных скобках стоит по определению Δ_i и его подстановка

$$E = 2(\Delta_i + y_i) - 1$$

существенно упрощает выражение.

Рассмотрим случай 2 на рис. 5.4 и заметим, что здесь должен быть выбран горизонтальный пиксел $(x_i + 1, y_i)$, так как y является монотонно убывающей функцией. Проверка компонент E показывает, что

$$(x_i + 1)^2 + (y_i)^2 - R^2 < 0$$

$$(x_i + 1)^2 + (y_i - 1)^2 - R^2 < 0$$

поскольку в случае 2 горизонтальный $(x_i + 1, y_i)$ и диагональный $(x_i + 1, y_i - 1)$ пикселы лежат внутри окружности. Следовательно, $E < 0$, и при использовании того же самого критерия, что и в случае 1, выбирается пиксел $(x_i + 1, y_i)$.

Если $\Delta_i > 0$, то диагональная точка $(x_i + 1, y_i - 1)$ находится вне окружности, т. е. это случаи 3 и 4 на рис. 5.4. В данной ситуации ясно, что должен быть выбран либо пиксел $(x_i + 1, y_i - 1)$, либо $(x_i, y_i - 1)$. Аналогично разбору предыдущего случая критерий выбора можно получить, рассматривая сначала случай 3 и проверяя разность между квадратами расстояний от окружности до диагонального m_D и вертикального m_V

пикселей,

$$\text{т. е. } E' = |(x_i + 1)^2 + (y_i - 1)^2 - R^2| - |(x_i)^2 + (y_i - 1)^2 - R^2|$$

При $E' < 0$ расстояние от окружности до вертикального пиксела $(x_i, y_i - 1)$ больше и следует выбрать диагональный шаг к пикселу $(x_i + 1, y_i - 1)$. Напротив, в случае $E' > 0$ расстояние от окружности до диагонального пиксела больше и следует выбрать вертикальное движение к пикселу $(x_i, y_i - 1)$. Таким образом,

при $E' \leq 0$ выбираем m_D в $(x_i + 1, y_i - 1)$

при $E' > 0$ выбираем m_V в $(x_i, y_i - 1)$

Здесь в случае $E' = 0$, т. е. когда расстояния равны, выбран диагональный шаг.

Проверка компонент E' показывает, что

$$(x_i)^2 + (y_i - 1)^2 - R^2 \geq 0$$

$$(x_i + 1)^2 + (y_i - 1)^2 - R^2 < 0$$

поскольку для случая 3 диагональный пиксел $(x_i + 1, y_i - 1)$ находится вне окружности, тогда как вертикальный пиксел $(x_i, y_i - 1)$ лежит внутри ее. Это позволяет записать E' в виде

$$E' = (x_i + 1)^2 + (y_i - 1)^2 - R^2 + (x_i)^2 + (y_i - 1)^2 - R^2$$

Дополнение до полного квадрата члена $(x_i)^2$ с помощью добавления и вычитания $2x_i + 1$ дает

$$E' = 2[(x_i + 1)^2 + (y_i - 1)^2 - R^2] - 2x_i - 1$$

Использование определения Δ_i приводит выражение к виду

$$E' = 2(\Delta_i - x_i) - 1$$

Теперь, рассматривая случай 4, снова заметим, что следует выбрать вертикальный пиксел $(x_i, y_i - 1)$, так как y является монотонно убывающей функцией при возрастании x .

Проверка компонент E' для случая 4 показывает, что

$$(x_i + 1)^2 + (y_i - 1)^2 - R^2 > 0$$

$$(x_i)^2 + (y_i - 1)^2 - R^2 > 0$$

поскольку оба пиксела находятся вне окружности. Следовательно, $E' > 0$ и при использовании критерия, разработанного для случая 3, происходит верный выбор m_V .

Осталось проверить только случай 5 на рис. 5.4, который встречается, когда диагональный пиксел $(x_i, y_i - 1)$ лежит на окружности, т. е. $\Delta_i = 0$. Проверка компонент E показывает, что

$$(x_i + 1)^2 + (y_i)^2 - R^2 > 0$$

$$(x_i + 1)^2 + (y_i - 1)^2 - R^2 = 0$$

Следовательно, $E > 0$ и выбирается диагональный пиксел $(x_i + 1, y_i - 1)$. Аналогичным образом оцениваем компоненты E' :

$$(x_i + 1)^2 + (y_i - 1)^2 - R^2 = 0$$

$$(x_i + 1)^2 + (y_i - 1)^2 - R^2 < 0$$

и $E' < 0$, что является условием выбора правильного диагонального шага к $(x_i + 1, y_i - 1)$. Таким образом, случай $\Delta_i = 0$ подчиняется тому же критерию, что и случай $\Delta_i < 0$ или $\Delta_i > 0$. Подведем итог полученных результатов:

$$\Delta_i < 0$$

$E \leq 0$ выбираем пиксел $(x_i + 1, y_i)$ - m_H

$E > 0$ выбираем пиксел $(x_i + 1, y_i - 1)$ - m_D

$$\Delta_i > 0$$

$E' \leq 0$ выбираем пиксел $(x_i + 1, y_i - 1)$ - m_D

$E' > 0$ выбираем пиксел $(x_i, y_i - 1)$ - m_V

$\Delta_i = 0$ выбираем пиксел $(x_i + 1, y_i - 1)$ - m_D

Легко разработать простые рекуррентные соотношения для реализации пошагового алгоритма. Сначала рассмотрим горизонтальный шаг m_H к пикселу $(x_i + 1, y_i)$. Обозначим это новое положение пиксела как $(i + 1)$. Тогда координаты нового пиксела и значение Δ_i равны

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i$$

$$\Delta_{i+1} = (x_{i+1} + 1)^2 + (y_{i+1} - 1)^2 - R^2 \square \square \Delta_i + 2x_{i+1} + 1$$

Аналогично координаты нового пиксела и значение Δ_i для шага m_D к пикселу $(x_i + 1, y_i - 1)$ таковы:

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i - 1$$

$$\Delta_{i+1} = \Delta_i + 2x_{i+1} - 2y_{i+1} + 2$$

То же самое для шага m_V к $(x_i, y_i - 1)$

Plot	Δi	E	E'	x	y
	-14			0	8
(0,8)					
	-11	-13		0	8
(1,8)					
	-6	-7		2	8
(2,8)					
	-12	3		3	7
(3,7)					
	-3	11		4	7
(4,7)					
	-3	7		5	6
(5,6)					
	1	5		6	5
(6,5)					
	9		-11	7	4
(7,4)					
	4		3	7	3
(7,3)					
	18		-7	8	2
(8,2)					
	17		19	8	1
(8,1)					
	18		17	8	0
(8,0)					

алгоритм завершен.

Результаты показаны на рис.5.6. вместе с реальной окружностью. Алгоритм легко обобщается для других квадрантов или дуг окружностей.

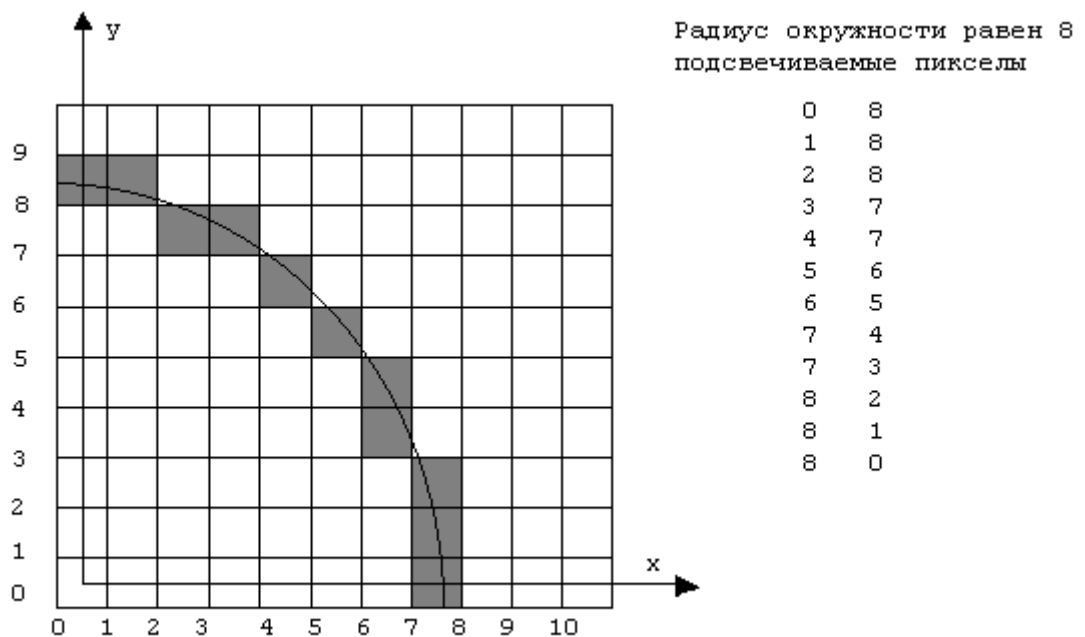


Рис. 5.6. Результаты работы пошагового алгоритма Брезенхема генерации окружности.

Алгоритмы растровой графики

Растром называется прямоугольная сетка точек, формирующих изображение на экране компьютера. Каждая точка растра характеризуется двумя параметрами: своим положением на экране и своим цветом, ес-

ли монитор цветной, или степенью яркости, если монитор черно-белый. Поскольку растровые изображения состоят из множества дискретных точек, то для работы с ними необходимы специальные алгоритмы. Рисование отрезка прямой линии - одна из простейших задач растровой графики. Смысл ее заключается в вычислении координат пикселей, находящихся вблизи непрерывных отрезков, лежащих на двумерной растровой сетке.

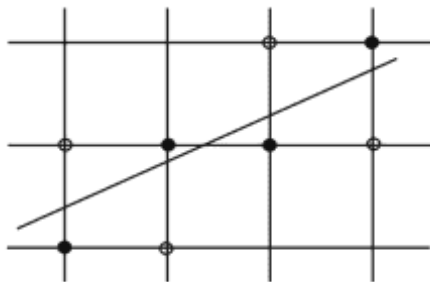


Рис. 28. Растеризация отрезка прямой линии.

Термин “пиксел” образован от английского pixel (picture element - элемент изображения) - то есть точка на экране. Будем считать, что пикселы имеют целочисленные координаты. На первый взгляд кажется, что эта задача имеет простое решение. Пусть конечные точки отрезка имеют целочисленные координаты, и уравнение прямой, содержащей отрезок: $y=kx+b$. Не нарушая общности, будем также считать, что тангенс угла наклона прямой лежит в пределах от 0 до 1. Тогда для изображения отрезка на растре достаточно для всех целых x , принадлежащих отрезку, выводить на экран точки с координатами $(x, \text{Round}(kx+b))$. Однако в этом методе присутствует операция умножения kx . Хотелось бы иметь алгоритм без частого использования операции умножения вещественных чисел. Избавиться от операции умножения можно следующим образом. Поскольку $k = \frac{\Delta y}{\Delta x}$ то один шаг по целочисленной сетке на оси x будет соответствовать $\Delta x = 1$. Отсюда получаем, что y будет увеличиваться на величину k . Итерационная последовательность выглядит следующим образом:

$$x_{i+1} = x_i + 1, \quad y_{i+1} = y_i + k$$

Когда $k > 1$, то шаг по x будет приводить к шагу по $y > 1$, поэтому x и y следует поменять ролями, придавая y единичное приращение, а x будет увеличиваться на $\Delta x = \frac{\Delta y}{k} = \frac{1}{k}$ единиц. Этот алгоритм все же не свободен от операций с вещественными числами. Наиболее изящное решение задачи растровой развертки отрезков прямых было найдено Брезенхемом. В его алгоритме вообще не используются операции с вещественными числами, в том числе операции умножения и деления.

Для вывода формул алгоритма Брезенхема рассмотрим рис. 29.

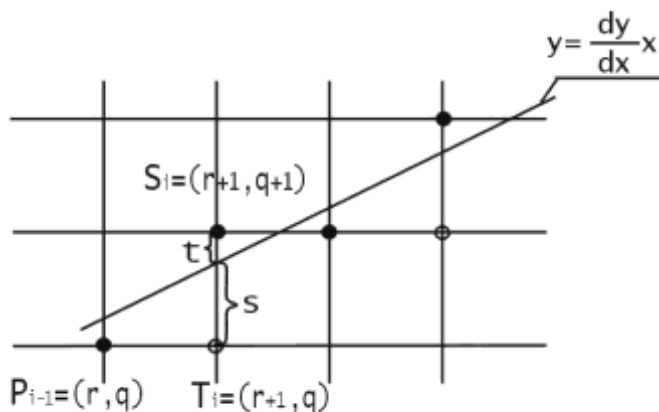


Рис. 29. Рисование отрезков прямых по методу Брезенхема.

Пусть начало отрезка имеет координаты (x_1, y_1) , а конец (x_2, y_2) . Обозначим $dx = (x_2 - x_1)$,

$dy(2-d)$. Не нарушая общности, будем считать, что начало отрезка совпадает с началом координат, и прямая имеет вид $y = \frac{dx}{dy}x$, где $\frac{dx}{dy} \in [0,1]$. Считаем что начальная точка находится слева. Пусть на $(i-1)$ -м шаге текущей точкой отрезка является $P_{i-1} = (r, q)$. Выбор следующей точки S_i или T_i зависит от знака разности $(s-t)$. Если $(s-t) < 0$, то $P_i = (r+1, q)$ и тогда $x_{i+1} = x_i + 1$, $y_{i+1} = y_i$, если же $(s-t) \geq 0$, то $P_i = (r+1, q+1)$ и тогда $x_{i+1} = x_i + 1$, $y_{i+1} = y_i + 1$.

$$s = \frac{dy}{dx}(r+1) - \epsilon, t = q + 1 - \frac{dy}{dx}(r+1) \Rightarrow$$

$$s - t = 2 \frac{dy}{dx} - 2 \Rightarrow$$

$$d_i = 2dy - dx$$

Поскольку знак $dx(s-t)$ совпадает со знаком разности $(s-t)$, то будем проверять знак выражения $d_i = d(s-t)$. Так как $r = x_{i-1}$ и $q = y_{i-1}$, то $d_i = d(x_i - y_{i-1})$.

Пусть на предыдущем шаге $d_i < 0$, тогда $(y_i - y_{i-1}) = 0$ и $d_{i+1} = d_i + 2d$. Если же на предыдущем шаге $d_i \geq 0$, то $(y_i - y_{i-1}) = 1$ и $d_{i+1} = d_i + d$.

Осталось узнать как вычислить d_1 . Так как при $i = 1$:

$$(x_0, y_0) = (0, 0), d_1 = d(2dy - dx).$$

Далее приводится листинг процедуры на языке Паскаль, реализующей алгоритм Брезенхема.

```

Procedure Bresenham(x1,y1,x2,y2,Color: integer);
var
dx,dy,incr1,incr2,d,x,y,xend: integer;
begin
  dx:= ABS(x2-x1);
  dy:= Abs(y2-y1);
  d:=2*dy-dx;      {начальное значение для d}
  incr1:=2*dy;      {приращение для d<0}
  incr2:=2*(dy-dx); {приращение для d>=0}
  if x1>x2 then     {начинаем с точки с меньшим знач. x}
  begin
    x:=x2;
    y:=y2;
    xend:=x1;
  end
  else
  begin
    x:=x1;
    y:=y1;
    xend:=x2;
  end;
  PutPixel(x,y,Color); {первая точка отрезка}
  While x<xend do
  begin
    x:=x+1;
    if d<0 then
      d:=d+incr1      {выбираем нижнюю точку}
    else
      begin
        y:=y+1;
        d:=d+incr2;    {выбираем верхнюю точку, y-возрастает}
      end;
  end;
  PutPixel(x,y,Color);

```

```
end;{while}
end;{procedure}
```

Перед тем, как исследовать методы получения изображений более сложных, чем отрезки прямых, рассмотрим проблему, незримо присутствующую в большинстве задач компьютерной графики. Эта проблема отсечения изображения по некоторой границе, например, по границе экрана, или, в общем случае, некоторого прямоугольного окна. Рассмотрим эту задачу применительно к отрезкам прямых. Некоторые из них полностью лежат внутри области экрана, другие целиком вне ее, а некоторые пересекают границу экрана. Правильное отображение отрезков означает нахождение точек пересечения их с границей экрана и рисование только тех их частей, которые попадают на экран. Один из очевидных способов отсечения отрезков состоит в определении точек пересечения прямой, содержащей отрезок, с каждой из четырех прямых, на которых лежат границы окна и проверки не лежит ли хотя бы одна точка пересечения на границе. В этом случае для каждой пары сторона-отрезок необходимо решать систему из двух уравнений, используя операции умножения и деления. При этом удобно параметрическое задание прямых:

$$\begin{aligned} x &= x_1 + t(x_2 - x_1) \\ y &= y_1 + t(y_2 - y_1) \end{aligned}$$

Для $t \in [0,1]$ эти уравнения определяют точки, находящиеся между (x_1, y_1) и (x_2, y_2) . Специальной проверки требует случай, когда отрезок параллелен стороне окна. Пусть координата x точки пересечения найдена, тогда

$$t = \frac{x - x_1}{x_2 - x_1} \Rightarrow y = y_1 + \frac{x - x_1}{x_2 - x_1} (y_2 - y_1)$$

Рассмотрим алгоритм Козна-Сазерленда для отсечения отрезков прямых. Этот алгоритм позволяет легко определять нахождение отрезка полностью внутри или полностью снаружи окна, и если так, то его можно рисовать или не рисовать, не заботясь об отсечении по границе окна.

Для работы алгоритма вся плоскость в которой лежит окно разбивается на девять подобластей или квадрантов, как показано на рис. 31.

1001	1000	1010
0001	0000	0010
0101	0100	0110

Рис. 31. Разбиение на подобласти в методе Козна-Сазерленда.

Окну соответствует область обозначенная кодом 0000. Конечным точкам отрезка приписывается 4-битный код “вне/внутри” в зависимости от нахождения отрезка в соответствующей подобласти. Каждому биту присваивается значение 1 в соответствии со следующим правилом.

- Бит 1 - точка находится выше окна;
- Бит 2 – точка находится ниже окна;
- Бит 3 - точка находится справа от окна;
- Бит 4 - точка находится слева от окна;

Иначе биту присваивается нулевое значение. Значения этих битов для конечных точек отрезков легко определить по знакам соответствующих разностей: $(y_{\max} - y)$ - для 1-го бита, $(y - y_{\min})$ - для 2-го бита,

$(x_{\max} - x)$ - для 3-го бита и $(x - x_{\min})$ - для 4-го бита. Отрезок рисуется без отсечения, то есть принимается целиком, если оба кода равны 0000, или $[кодP1]ИЛИ[кодP2] = 0000$, где ИЛИ – бинарная операция. Отрезок отбрасывается без вычислений если оба его конца находятся выше, ниже, правее или левее окна. В этих случаях соответствующие биты в обоих кодах равны 1 и это легко определить, умножив эти коды по бинарной операции И. Если результат операции И равен 0000, то отрезок нельзя ни принять ни отбросить, так как он может пересекаться с окном. В этом случае применяется последовательное разделение отрезка, так что на каждом шаге конечная точка отрезка с ненулевым кодом вне/внутри заменяется на точку, лежащую на стороне окна или на прямой содержащей сторону. При этом порядок перебора сторон окна не имеет значения.

Далее приводится текст процедуры на языке Паскаль, с довольно изящной реализацией этого метода. Отрезок задан граничными точками $P1 = (x1, y1)$, $P2 = (x2, y2)$, границы окна: $xmin, xmax, ymin, ymax$. Используются вызовы процедур: Accept_Check – выполняет проверку на полное принятие отрезка; Reject_Check – на полный отказ от рисования отрезка; Outcodes – вычисляет 4-х битовый код “вне/внутри”; SWAP – меняет местами координаты двух точек.

```

Procedure CLIP(x1,x2,y1,y2,xmin,xmax,ymin,ymax: real);
type
  outcode = array[1..4] of boolean;
var
  accept,reject,done: boolean;
  outcode1,outcode2,
  outcode3,outcode4:outcode; {коды вне/внутри}
begin
  accept:= false;
  reject:= false;
  done:= false;
repeat
  Outcodes(x1,y1,outcode1);
  Outcodes(x2,y2,outcode2);
  {проверка на отбрасывание}
  reject:=Reject_Check(outcode1,outcode2);
  if reject then done:= true
  else
    begin {возможно принятие целиком}
      accept:=Accept_Check(outcode1,outcode2);
      if accept then done:=true
      else
begin {разделить отрезок}
{если P1 внутри, то с помощью SWAP сделать снаружи}
if not((outcode1[1])or(outcode1[2])or
      (outcode1[3])or(outcode1[4])) then SWAP;
{теперь P1 перемещается в точку пересечения}
  if outcode1[1] then
    begin {отбросить верхнюю часть}
      x1:=x1+(x2-x1)*(ymax-y1)/(y2-y1);
      y1:=ymax;
    end
  else if outcode1[2] then
  if outcode1[1] then
    begin {отбросить нижнюю часть}
      x1:=x1+(x2-x1)*(ymin-y1)/(y2-y1);
      y1:=ymin;
    end
end

```

```

else if outcode1[3] then
  begin {отбросить правую часть}
     $y1 := x1 + (y2 - y1) * (y_{\max} - x1) / (x2 - x1);$ 
     $x1 := x_{\max};$ 
  end
else if outcode1[4] then
  begin {отбросить левую часть}
     $y1 := x1 + (y2 - y1) * (y_{\min} - x1) / (x2 - x1);$ 
     $x1 := x_{\min};$ 
  end;
end;
until done;
if accept then
  Line(x1,y1,x2,y2); {нарисовать отрезок}
end; {procedure}

```