Обоснование

В процессе чтения данной лекции, студент узнает что такое графика, об оцифровке, о растровой и векторной графике. Также, о программных средствах для работы с тем или иным файловым форматом. Будет подробно рассказано о файловых форматах и истории их создания.

Данная лекция формирует такую компетенцию, как ОПК-2. Из рабочей программы дисциплины и данной лекции, мы можем вынести то, что студент будет знать современные технологии и программные средства для работы с различным типом файлов. Уметь выбирать нужную программу для работы с тем или иным типом файлов и владеть такой программой.

Лекция. Введение в компьютерную графику.

В этой главе вы найдете ответы на следующие вопросы:

* Что такое оцифровка сигналов?
* В чем отличие векторной графики от растровой?
* Какие существуют графические файловые форматы?

Термином **графика**обозначаются программные и аппаратные технологии, которые применяются в компьютерных системах для создания, видоизменения и отображения неподвижных картинок, записанных в цифровом виде. В этом смысле графика имеет фундаментальное значение для мультимедиа не только по тому, что она позволяет создавать и показывать неподвижные изображения, но и потому, что она лежит в основе воспроизведения движущихся изображений и текста. Чтобы понять, каким образом графика играет эту роль, нужно рассмотреть процесс создания и отображения неподвижных картинок.

1. Оцифровка

В мультимедийных средствах мы встречаемся со значениями, которые изменяются непрерывно. Например, амплитуда звуковой волны непрерывно изменяется во времени, как и амплитуда электрического сигнала, вырабатываемого микрофоном в результате воздействия звуковой волны. Яркость в любой точке черно-белой фотографии, также, может иметь любое значение.

Когда у нас есть непрерывно изменяющийся сигнал, значение, которое мы измеряем, и интервал, на котором мы можем его измерять, могут изменяться на бесконечно малую величину. Если бы нужно было преобразовать его в цифровой сигнал, то пришлось бы ограничиться набором дискретных значений. Иначе говоря, оцифровка состоит из двух этапов:

* дискретизации, когда мы измеряем величину сигнала на определенных дискретных промежутках, и
* квантования, когда эти значения ограничиваются определенным набором уровней.

Одно из преимуществ, которые имеет цифровое представление перед аналоговым, это то, что корректными являются только определенные значения сигнала — те, которые попадают на уровни квантования. Если сигнал передается по проводам или записывается на физический носитель, например на магнитную ленту, на него неизбежно накладываются случайные шумы. Эти шумы приводят к изменению значения сигнала, которое невозможно обнаружить. Но если сигнал квантованный, то любые незначительные изменения, вызванные шумом, переводят допустимые значения в разряд недопустимых, которые находятся между уровнями квантования. После этого очень просто восстановить исходный сигнал, проквантовав его еще раз. Ошибки при передаче могут возникать только тогда, когда шумы достаточно большие и приводят к такому изменению сигнала, что его значение попадает на другой уровень. Следовательно, цифровые сигналы намного устойчивее, чем аналоговые, и не искажаются при копировании или передаче через зашумленную среду.

Однако, в процессе оцифровки часть информации теряется. Как же тогда можно заявлять о том, что результат оцифровки является точным описанием исходного аналогового сигнала? Мерой точности должно быть то, насколько близко к оригиналу может быть восстановлен сигнал. Для того чтобы восстановить аналоговый сигнал по набору дискретных значений, все, что нужно сделать, — это решить, чем заполнить промежутки между этими точками.

Один из возможных способов — использование метода выборки-хранения, который заключающегося в следующем: значение в точке выборки остается постоянным на всем интервале между этой точкой и точкой, следующей за ней. Это дает сигнал с резкими переходами, что является не очень хорошей аппроксимацией оригинала. Однако когда сигнал подается на экран электронно-лучевой трубки или громкоговоритель, задержки и изъяны, которые свойственны любым физическим приборам, сглаживают резкие переходы, и в результате получается достаточно хорошая, с теоретической точки зрения, аппроксимация.

Если дискретные значения исходного сигнала находятся слишком далеко одно от другого, любое восстановление будет неадекватным, поскольку аналоговый сигнал может содержать такие детали, которые упускаются в процессе дискретизации из-за больших расстояний между элементами выборки. Значения, взятые в соседних точках, одинаковы, и нельзя сделать вывод о присутствии пика между этими двумя точками. Влияние такой дискретизации с недостаточной частотой проявляется в виде искажений и артефактов, которые всегда нежелательны.

Если выполнять дискретизацию сигнала с недостаточной частотой, то в процессе восстановления сигнала возникнут искажения, которые будут проявляться по-разному. При воспроизведении звука, оцифрованного с недостаточной частотой дискретизации, будут слышны искажения. На изображениях недостаточная частота дискретизации проявится в виде зубчатых краев или в виде муаровых картин. В случае движущихся картин, недостаточная частота дискретизации по времени приводит к прерывистым движениям или к неадекватному представлению вращательного движения, например в старых фильмах, часто видно как будто колеса вращаются в обратном направлении.

Влияние недостаточного количества уровней квантования проиллюстрируем на следующем примере. Если мы можем задавать только ограниченное число различных значений, мы не сможем различать те значения, которые находятся между ними. Этот эффект хорошо виден на черно-белых изображениях, на котором показаны градиенты с уровнями яркости 256, 128, 64, 32, 16, 8, 4 и 2. Первый градиент демонстрирует постепенный переход от чистого белого цвета к чистому черному, а затем видно, как при уменьшении количества уровней яркости соседние значения сливаются, поскольку их квантование становится все более и более грубым.

Самой распространенной причиной ограничения количества уровней квантования является уменьшение объема памяти, необходимого для записи цифровой информации, путем ограничения количества битов, которые используются для записи каждого значения. В нашем примере мы использовали от 8 до 1 бит. Как видно, хотя сам эффект заметен уже при 128 уровнях яркости, он становится ощутимым только тогда, когда количество уровней яркости снижается до 32, после чего качество начинает быстро ухудшаться.

1. Векторная и растровая графика

На мониторах картинки отображаются как прямоугольные массивы пикселей – небольших, как правило, прямоугольных цветных точек, которые оптически сливаются одна с другой.

Процесс воссоздания узора пикселей по модели называется визуализацией.

Принято считать, что существует два различных подхода к графическому моделированию:

* растровая графика
* векторная графика.

В случае растровой графики изображение моделируется с помощью массива значений пикселей.

В случае векторной графики изображение записывается в виде математического описания набора отдельных прямых, кривых и геометрических фигур, составляющих изображение. Для вывода векторного рисунка на экран необходимо выполнить вычислительные операции, в ходе которых и создается массив пикселей, которые следует отображать. Например, прямая линия в модели задается через координаты ее концов. Когда модель визуализируется, должны вычисляться координаты всех пикселей, которые лежат на прямой линии, соединяющей эти концы.

Векторная и растровая графика кардинально отличаются одна от другой. Требования, предъявляемые ими к компьютерной системе, будут разными: при создании растрового изображения нужно записывать значение каждого пикселя, тогда как описание векторного изображения может быть намного компактнее.

Размер квадрата 1 на 1 см. Если его записывать с помощью 100 логических пикселей на 1 см , то его битовый массив будет состоять из 1002 = 10 000 пикселей. Если предположить, что рисунок предназначен для отображения на мониторе, способном одновременно передавать миллионы цветов, то в таком случае, чтобы различать возможные цвета, понадобится 3 байта памяти на каждый пиксель. Это значит, что все изображение будет занимать около 30 Кбайт. С другой стороны, тот же рисунок можно записать в виде описания размеров квадрата, а также цветов его внутренней части и контуров. Одним из возможных форматов может быть короткий документ на специальном графическом языке и занимающий в общей сложности менее 300 байт.

Для растрового изображения всегда записываются значения каждого логического пикселя, поэтому размер изображения и разрешение — полностью определяют занимаемый объем памяти. При векторном представлении записываются описания всех объектов, составляющих изображение; чем сложнее рисунок, тем больше будет объектов и тем большим будет описание, но, поскольку значения самих пикселей в действительности не записываются, размер рисунка не будет зависеть от его разрешения.

Традиционно различают программы для *рисования*, работающие с битовыми массивами, и программы для *черчения*, работающие с векторными представлениями. Программные пакеты, например, Illustrator и Corel Draw, прежде всего, предназначены для создания и редактирования изображений векторной графики и предлагают только ограниченные возможности для работы с битовыми массивами, тогда как, например, PhotoShop и Painter, поддерживающие операции с битовыми массивами, предоставляют либо незначительную поддержку для векторной графики, либо слабенькую встроенную подсистему для черчения.



Рисунок 1 – Иллюстрации

На рисунке 1 показаны два цветка. Первый из них — это векторное изображение, полученное с помощью программы Illustrator, а второй — растровое изображение. Битовый массив передает текстуру и обеспечивает хорошее воспроизведение непрерывных тонов. Векторный рисунок носит совсем другой характер: его элементы с четкими границами, состоящие из набора гладких кривых линий, заполнены одним цветом.



Рисунок 2 – Иллюстрации

На рисунке 2 разница между этими двумя форматами показана иным способом. В случае векторного представления можно легко выбирать отдельные элементы — лепестки, стебель, тычинки — и независимо перемещать их, разворачивать или выполнять другие преобразования. Каждый элемент рисунка остается самим собой, и его можно редактировать, как отдельный объект. Для растрового изображения, которое является всего лишь массивом пикселей это невозможно.

С другой стороны, к растровым изображениям можно применять такие эффекты, как искажение и размывание.

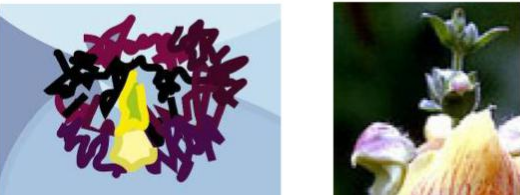


Рисунок 3 – Иллюстрации

Поскольку векторное изображение состоит из описаний геометрических фигур, составляющих рисунок, а не из значений пикселей, масштабирование можно легко осуществить с помощью несложных математических операций. В результате этого кривые будут оставаться гладкими, независимо от того, насколько увеличивается векторное изображение, тогда как при изменении масштаба растрового изображения они будут зубчатыми или размытыми. Этот эффект показан на рисунке 3, на которых элементы двух изображений различных цветков увеличены в 8 раз.

1. Файловые форматы

Чтобы сохранять изображения и перемещать их из одной программы в другую, необходимо иметь возможность записи информации об изображении в файл.

Для растровых изображений одним из основных отличий между файловыми форматами является способ сжатия информации об изображении. Растровые изображения могут состоять из большого количества пикселей, поэтому для них нужны большие файлы, часто занимающие несколько мегабайтов памяти для каждого изображения. Чтобы уменьшить необходимый для таких изображений объем памяти, к ним часто применяют технологию сжатия данных. Алгоритмы сжатия без потерь позволяют точно восстановить исходные данные по сжатой версии файла. При выполнении алгоритмов сжатия с потерями часть информации отбрасывается для достижения большей степени сжатия.

Полное описание цвета мы отложим на потом. Здесь же отметим, что один из способов уменьшения размера растрового изображения состоит в ограничении количества цветов, которые могут содержаться в изображении. Если для воспроизведения изображения необходимо максимум 256 цветов, то для записи каждого пикселя нужен всего один байт. Если же нужно открыть для работы полный диапазон, состоящий из миллионов цветов, что способны передавать очень многие мониторы, то для описания каждого пикселя понадобится по три байта.

Первым появился формат **GIF**. В файлах GIF используется технология сжатия без потерь, а цветовая палитра ограничивается 256 цветами. Самая полезная особенность данного формата заключается в том, что один цвет можно обозначить как прозрачный, т.е. если изображение GIF выводить на экран на цветном фоне или на фоне другого изображения, то сквозь прозрачные области будет виден этот фон.

Для отсканированных изображений или фотографий более предпочтителен формат **JPEG**. Строго говоря, JPEG — это технология сжатия, а изображения, которые были сжаты с ее помощью, можно записывать в любом файловом формате, данные JPEG могут входить в состав других файлов, в том числе формата TIFF.

Третий, самый новый, файловый формат, широко поддерживаемый сетью Интернет, — это **PNG**, который был создан вместо GIF. С форматом GIF связан следующий неприятный момент: используемые в нем алгоритмы сжатия запатентованы компанией Unisys, и эта компания требует оплаты права использования любой программы, в которой реализуются процессы сжатия или распаковки файлов GIF. С другой стороны, в формате PNG применяются различные методики сжатия без потерь, для доступа к которым подобных препятствий не существует, и поэтому ими свободно могут пользоваться все. Кроме того, формат PNG не ограничен 256 цветами, и в нем предлагается более сложная форма прозрачности, чем в формате GIF. Формат PNG был разработан в 1996 году, но поддержка PNG разрабатывается очень медленно, поэтому более популярными остаются форматы GIF и JPEG.

К числу других наиболее распространенных файловых форматов для растровой графики относятся TIFF и ВМР. **ТIFF**представляет собой детально разработанный открытый файловый формат, в котором можно записывать полноцветные растровые изображения, используя для этого несколько разных схем сжатия, в том числе JPEG. Формат **ВМР**правильнее было бы назвать растровым форматом Микрософт. ВМР поддерживает только простую форму сжатия без потерь, и файлы ВМР, как правило, записываются в несжатом виде. Ситуация с векторной графикой несколько иная.

**Формат EPS**— расширенный PostScript – это один из самых распространенных форматов векторной графики. Формат EPS был создан компанией Adobe на основе языка PostScript и послужил базой для создания ранних версий формата Adobe Illustrator. Формат используется в профессиональной полиграфии, и может содержать растровые изображения, векторные изображения, а также их комбинации. Программы для работы с EPS QuarkXPress и Adobe InDesign.

**Формат PDF -**кроссплатформенный формат электронных документов, созданный фирмой Adobe Systems с использованием ряда возможностей языка PostScript. В первую очередь предназначен для представления в электронном виде полиграфической продукции, — значительное количество современного профессионального печатного оборудования может обрабатывать PDF непосредственно. Для просмотра можно использовать официальную бесплатную программу Adobe Reader, а также программы сторонних разработчиков. Традиционным способом создания PDF-документов является виртуальный принтер, то есть документ, как таковой, готовится в своей специализированной программе — графическом или текстовом редакторе, САПР и т. д., а затем экспортируется в формат PDF для распространения в электронном виде, передачи в типографию и т. п. PDF с 1 июля 2008 года является открытым стандартом ISO 32000. Формат PDF позволяет внедрять необходимые шрифты (построчный текст), векторные и растровые изображения, формы и мультимедиа-вставки.

**Postscript**был разработан Джоном Уорноком и Чаком Гешке из Adobe Systems в начале 80-х гг. Исходно Postscript использовался как ядро механизма печати компьютеров Apple, но вскоре стал широко распространенным стандартом для большинства компьютерных систем. Каждый документ Postscript включает в себя программу, которая печатает на принтере (или отображает на экране монитора) следующие друг за другом страницы.

PostScript соединил в себе лучшие возможности принтеров и плоттеров. Подобно плоттерам, PostScript предоставляет возможность вывода высококачественной векторной графики и единый язык управления, который может быть использован любым производителем принтеров. Подобно матричным принтерам, PostScript предлагает удобные возможности по печати растровой графики и текста. В отличие от тех и других, PostScript может совмещать все эти типы вывода на одной странице, давая намного больше гибкости, чем до этого имел любой принтер или плоттер.

PostScript — больше, чем типичный язык управления принтером, он является полнофункциональным языком программирования. Многие прикладные программы могут преобразовать документ в PostScript-программу, при выполнении которой будет получен начальный документ. Эта программа может быть послана непосредственно на принтер с поддержкой PostScript или преобразована интерпретатором PostScript в другой формат (для принтеров без поддержки PostScript), или результат её выполнения интерпретатором может быть показан на экране. Так как исходная PostScript программа одна и та же, PostScript называется независимым от устройства.

Большинство высокопроизводительных принтеров и плоттеров имеют встроенный интерпретатор языка PostScript. В то же время, простые принтеры домашнего класса поддерживают только элементарные графические операции, поэтому задача создания растрового изображения возлагается на центральный процессор. Существуют интерпретаторы языка PostScript для различных операционных систем, наиболее известный из них — свободная программа Ghostscript.

**Формат SVG**язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двухмерной векторной и смешанной векторно/растровой графики в формате XML. Поддерживает как неподвижную, так анимированную и интерактивную графику.

SVG - текстовый формат — файлы SVG можно читать и редактировать при помощи обычных текстовых редакторов. При просмотре документов, содержащих SVG графику, имеется доступ к просмотру кода просматриваемого файла и возможность сохранения всего документа. Кроме того, SVG файлы обычно получаются меньше по размеру, чем сравнимые по качеству изображения в форматах JPEG или GIF, а также хорошо поддаются сжатию.

Широко доступно использование растровой графики в SVG документах. Имеется возможность вставлять элементы с изображениями в форматах PNG, GIF или JPG.

Редакторы SVG:

* Inkscape — свободный векторный графический пакет для работы с SVG.
* OpenOffice.org Draw умеет экспортировать графику в файлы формата SVG.
* Adobe Illustrator версии от CS2.
* Adobe GoLive версии от CS2.
* CorelDRAW векторный графический редактор.

**Формат WMF**— универсальный формат векторных графических файлов для Windows приложений. Используется для хранения коллекции графических изображений Microsoft Clip Gallery. Формат разработан Microsoft и является неотъемлемой частью Windows. Очень часто WMF неявно используется для сохранения образа окна вывода программы и его последующего восстановления, а также при переносе информации через буфер обмена (clipboard).

**Формат SWF**– Adobe Flash (ранее известный как Macromedia Flash), или просто Flash— мультимедийная платформа, используемая для создания векторной анимации и интерактивных приложений (в том числе, игр), а также для интеграции видеороликов в веб-страницы.

Flash-контент воспроизводится с помощью целого ряда программных средств, но доминирующее положение на рынке занимает официальный Adobe Flash Player, распространяемый в качестве бесплатного плагина для большинства современных браузеров. Также SWF-файлы можно просматривать с помощью различных свободных плееров.

Также, как и файловые форматы общего назначения, широко используются определенные запатентованные форматы, связанные с популярными программами. В частности, файлы PhotoShop и CorelDraw традиционно используются в качестве форматов для обмена информацией с растровой и векторной графикой соответственно в допечатных и издательских процессах.