Оглавление

[*1.* *Опишите направления овбработки информации, связанной с изображениями.* 2](#_Toc124109345)

[*2. Дайте определение основным терминам машинной графики (альфа-канал, метафайл, цветовая модель, пиксел, растр, графический объект, примитив).* 2](#_Toc124109346)

[*3. Дайте характеристику основным стандартам компьютерной графики.* 2](#_Toc124109347)

[*4. Приведите хронологию разработок в области графических систем.* 2](#_Toc124109348)

[*5. Опишите основные элементы векторной иллюстрации.* 2](#_Toc124109349)

[*6. Опишите свойства векторных объектов.* 2](#_Toc124109350)

[*7. Дайте характеристику приемам и программным средствам точного позиционирования (на примере AutoCAD).* 2](#_Toc124109351)

[*8. Опишите способы и форматы ввода координат (на примере AutoCAD).* 2](#_Toc124109352)

[*9. Опишите физические характеристики светового потока.* 2](#_Toc124109353)

[*10. Дайте характеристику аддитивным цветовым моделям.* 2](#_Toc124109354)

[*11. Дайте характеристику субтрактивным цветовым моделям.* 2](#_Toc124109355)

[*12. Какие модели опираются на интуитивные понятия тона, насыщенности и яркости?* 2](#_Toc124109356)

[*13. Дайте характеристику форматам растровой графики.* 2](#_Toc124109357)

[*14. Дайте характеристику форматам векторной графики.* 2](#_Toc124109358)

[*15. Опишите устройства ввода графической информации.* 2](#_Toc124109359)

[*15 Новый.* *Дайте характеристику комплексным форматам* 2](#_Toc124109360)

[*16. Опишите устройства вывода графической информации.* 2](#_Toc124109361)

[*17. Опишите способ сжатия графической информации без потерь.* 2](#_Toc124109362)

[*18. Опишите способ сжатия графической информации с потерями.* 2](#_Toc124109363)

[*19. Аффинная геометрия. Свойства аффинных преобразований. Перемещение.* 2](#_Toc124109364)

[*20. Аффинная геометрия. Свойства аффинных преобразований. Масштабирование.* 2](#_Toc124109365)

[*21. Аффинная геометрия. Свойства аффинных преобразований. Поворот.* 2](#_Toc124109366)

[*22. Основные матрицы 3D- геометрических преобразований.* 2](#_Toc124109367)

[*23. Опишите алгоритм Брезенхама для вычерчивания отрезков прямых линий.* 2](#_Toc124109368)

[*24. Опишите алгоритм Брезенхама для вычерчивания окружностей.* 2](#_Toc124109369)

[*25. Опишите алгоритмы устранения лестничного эффекта.* 3](#_Toc124109370)

[*26. Опишите алгоритмы заполнения области.* 3](#_Toc124109371)

[*27. Дайте характеристику основным этапам отображения 3-х мерной информации.* 3](#_Toc124109372)

[*28. Методы объемного представления. Воксельное представление.* 3](#_Toc124109373)

[*29. Методы объемного представления. Восьмеричное дерево.* 3](#_Toc124109374)

[*30. Опишите основные операции, применяемые для создания моделей в Компас 3D.* 3](#_Toc124109375)

[*31. Какие типы моделей можно создавать в системе КОМПАС 3D? Из каких объектов состоит трехмерная модель в КОМПАС 3D?* 3](#_Toc124109376)

[*32. Метод выдавливания. Какие требования предъявляются к эскизу элемента выдавливания?* 3](#_Toc124109377)

[*33. Из чего состоят геометрические объекты модели в КОМПАС 3D?* 3](#_Toc124109378)

[*34. Что такое эскиз? Опишите принципы работы с эскизами.* 3](#_Toc124109379)

[*35. Метод вращения. Какие требования предъявляются к эскизу элемента вращения?* 4](#_Toc124109380)

[*36. Дайте характеристику понятиям «базовая точка», «дерево построения», «компоненты сборки».* 4](#_Toc124109381)

[*37. Опишите особенности построения сборок в Компас 3D. Что такое сопряжение?* 5](#_Toc124109382)

[*38. Дайте характеристику понятиям «ассоциативный чертеж», «спецификация».* 5](#_Toc124109383)

# *Опишите направления овбработки информации, связанной с изображениями.*

Самая важная функция компьютера — обработка информации. Особо можно выделить обработку информации, связанную с изображениями. Она разделяется на три основных направления: *компьютерная графика (КГ), обработка изображений и распознавание изображений.*

Задача **компьютерной графики**— визуализация (создание изображения). Визуализация выполняется исходя из того, что нужно отображать. Существует много методов и алгоритмов визуализации, которые различаются между собою в зависимости от того, что и как отображать.

**Обработка** изображений — это преобразование изображений. То есть входными данными является изображение, и результат - изображение. Примером обработки изображений могут служить: повышение контраста, четкости, коррекция цветов, редукция цветов, сглаживание, уменьшение шумов и так далее.

Задачей обработки изображений может быть как улучшение в зависимости от определенного критерия (реставрация, восстановление), так и специальное преобразование, кардинально изменяющее изображения. В последнем случае обработка изображений может быть промежуточным этапом для дальнейшего распознавания изображения.

Для **распознавания изображений** основная задача - получение описания изображенных объектов. Методы и алгоритмы распознавания разрабатывались, прежде всего, для обеспечения зрения роботов и для систем специального назначения. Но в последнее время компьютерные системы распознавания изображений все чаще появляются в повседневной практике многих людей — например, офисные системы распознавания текстов или программы векторизации, создание трехмерных моделей человека.

Цель распознавания может формулироваться по-разному - выделение отдельных элементов; классификация изображения в целом. Методы классификации и выделения отдельных элементов могут быть взаимосвязаны. Так, классификация может быть выполнена на основе структурного анализа отдельных элементов объекта. Или для выделения отдельных элементов можно использовать методы классификации. Задача распознавания является обратной по отношению к визуализации.

# *2. Дайте определение основным терминам машинной графики (альфа-канал, метафайл, цветовая модель, пиксел, растр, графический объект, примитив).*

**Альфа-канал** (alphachannel) — дополнительный канал растровых данных, используемый для хранения сведений о прозрачности изображения (попиксельной, поблочной или для всего изображения). Степень прозрачности пикселя, заданная восьмибитовым альфа-значением, находится в интервале от 0 (пиксель полностью невидим — прозрачен) до 255 (пиксель полностью виден — непрозрачен).

**Метафайл** (metafile) — файл, который содержит элементы изображения, выполненный с использованием как растровой, так и векторной графики. Термин введен в употребление Комитетом по графическим стандартам и планированию SGGRAPH в 1997 году, соответствующие форматы записи называют «метафайловыми». Их отличает облегченный перенос с одной системы на другую; меньший размер по отношению к растровой версии того же изображения; обычно метафайлы хорошо сжимаются. Однако работа с метафайлами сложна и требует специальной подготовки.

**Пиксел** (pixel, pictureelement) — минимальный адресуемый элемент двумерного растрового изображения.

**Растр** (raster) — дискретное изображение, представленное в виде матрицы «точечных» элементов — пикселей.

**Примитив** (графический примитив, primitive) — элементарный графический объект (линия, прямоугольник, треугольник, окружность, конус, тор, куб), используемый в графической системе в качестве шаблона для построения более сложных графических объектов. Примитив является составной частью набора графических примитивов, предназначенного для реализации эффективного построения изображений объектов.

**Графический объект** (graphicsobject) — совокупность графических примитивов, соответствующих одному объекту отображаемого пространства или сцены.

**Цветовая модель** (colormodel) — способ распределения и задания цвета в конкретной программе или системе. Все используемые в компьютерной графике цветовые модели можно условно подразделить на монохромные (в том числе двухградационные или графические и полутоновые) и цветные (в том числе индексные и полноцветные).

# *3. Дайте характеристику основным стандартам компьютерной графики.*

Сегодня стандартизацией в машинной графике занимается 24-й подкомитет (ISO/IEC JTC1/SC24). Основными стандартами являются:

1.GKS (Graphical Kernel System) - набор базовых функций для 2D аппаратно-независимой машинной графики.

2.GKS-3D (Graphical Kernel System for 3 Dimensions) - расширение GKS для поддержки базовых функций в 3D.

3.PHIGS (Programmer's Hierarchical Interactive Graphics System) - набор базовых функций 3D графики аналогичный GKS-3D, но в отличие от GKS-3D, ориентированной на непосредственный вывод графических примитивов, группируемых в сегменты, графическая информация накапливается в иерархической структуре данных. В целом PHIGS ориентирован на приложения, требующие быстрой модификации графических данных, описывающих геометрию объектов.

4. Языковые интерфейсы (Languagebindings) - представление функций и типов данных функциональных графических стандартов в стандартизованных языках программирования.

5. CGM (Computer Graphics Metafile) - аппаратно-независимый формат обмена графической информацией. Используется для передачи и запоминания информации, описывающей изображения.

6.CGI (Computer Graphics Interface) - набор базовых элементов для управления и обмена данными между аппаратно-независимым и аппаратно-зависимым уровнями графической системы.

7. CGRM (Computer Graphics Reference Model) - модель стандартов в машинной графике, которая определяет концепции и взаимоотношения применительно к будущим стандартам в машинной графике.

8. Регистрация - механизм регистрации стандартизуемых аспектов примитивов вывода, обобщенных примитивов, escape-функций (для доступа к аппаратным возможностям устройств) и других графических элементов.

9.Тестирование реализаций на соответствие графическим стандартам - основные цели этого проекта: специфицирование характеристик стандартизованных тестов, используемых для определения соответствия реализаций графическим стандартам, и выработка предписаний разработчикам функциональных стандартов относительно правил соответствия.

# *4. Приведите хронологию разработок в области графических систем.*

Первой официально признанной попыткой использования дисплея для вывода изображения из ЭВМ явилось создание в Массачусетском технологическом университете машины Whirlwind-I в 1950 г. Таким образом, *возникновение компьютерной графики* можно отнести к 1950-м годам.

Первое реальное применение компьютерной графики связывают с именем Дж. Уитни. Он занимался кинопроизводством в 50-60-х годах и впервые использовал компьютер для создания титров к кинофильму.

Следующим шагом в своем развитии компьютерная графика обязана Айвэну Сазерленду, который в 1961 г., еще будучи студентом, создал программу рисования, названную им Sketchpad. Программа использовала световое перо для рисования простейших фигур на экране. Полученные картинки можно было сохранять и восстанавливать. В этой программе был расширен круг основных графических примитивов, в частности, помимо линий и точек был введен прямоугольник, который задавался размерами и расположением.

В 1961 г. студент Стив Рассел создал первую компьютерную видеоигру Spacewar, а научный сотрудник Bell Labs Эдвард Зэджек создал анимацию.

Университет штата Юта становится центром исследований в области компьютерной графики благодаря Д. Эвансу и А. Сазерленду, которые в это время были самыми заметными фигурами в этой области. Позднее их круг стал быстро расширяться. Учеником Сазерленда стал Э. Кэтмул, будущий создатель алгоритма удаления невидимых поверхностей с использованием Z-буфера (1978). Здесь же работали Дж. Варнок, автор алгоритма удаления невидимых граней на основе разбиения области (1969) и основатель Adobe System (1982), Дж. Кларк, будущий основатель компании Silicon Graphics (1982). Все эти исследователи очень сильно продвинули алгоритмическую сторону компьютерной графики.

В 1971 г. Гольдштейн и Нагель впервые реализовали метод трассировки лучей с использованием логических операций для формирования трехмерных изображений.

В 1970-е годы произошел резкий скачок в развитии вычислительной техники благодаря изобретению микропроцессора, в результате чего началась миниатюризация компьютеров и быстрый рост их производительности. И в это же время начинает интенсивно развиваться индустрия компьютерных игр. Одновременно компьютерная графика начинает широко использоваться на телевидении и в киноиндустрии. Дж. Лукас создает отделение компьютерной графики на Lucasfilm.

В середине 1970-х годов графика продолжает развиваться в сторону все большей реалистичности изображений. Э. Кэтмул в 1974 г. создает первые алгоритмы текстурирования криволинейных поверхностей. В 1975 г. появляется метод закрашивания Фонга. В 1977 г. появляется новый журнал "Computer Graphics World". В 1977 г. Дж. Блинн предлагает алгоритмы реалистического изображения шероховатых поверхностей; Ф. Кроу разрабатывает методы устранения ступенчатого эффекта при изображении контуров.

Дж.Брезенхем создает эффективные алгоритмы построения растровых образов отрезков, окружностей и эллипсов. Уровень развития вычислительной техники к этому времени уже позволил использовать "жадные" алгоритмы, требующие больших объемов памяти, и в 1978 г. Кэтмул предлагает метод Z-буфера, в котором используется область памяти для хранения информации о "глубине" каждого пикселя экранного изображения. В этом же году Сайрус и Бэк развивают алгоритмы клиппирования (отсечения) линий. А в 1979 г. Кэй и Гринберг впервые реализуют изображение полупрозрачной поверхности.

В 1980 г. Т.Уиттед разрабатывает общие принципы трассировки лучей, включающие отражение, преломление, затенение и методы антиэлайзинга.

В 1984 г. группой исследователей (Горэл, Торрэнс, Гринберг и др.) была предложена модель излучательности, одновременно развиваются методы прямоугольного клиппирования областей.

В 1980-е годы появляется целый ряд компаний, занимающихся прикладными разработками в области компьютерной графики. В 1982 г. Дж.Кларк создает Silicon Graphics, тогда же возникает Ray Tracing Corporation, Adobe System, в 1986 г. компания Pixar отпочковывается от Lucasfilm.

Таким образом, в процессе развития компьютерной графики можно выделить несколько этапов.

В 1960-1970-е годы она формировалась как **научная дисциплина**. В это время разрабатывались основные методы и алгоритмы: отсечение, растровая развертка графических примитивов, закраска узорами, реалистическое изображение пространственных сцен (удаление невидимых линий и граней, трассировка лучей, излучающие поверхности), моделирование освещенности.

В 1980-е графика развивается более как **прикладная дисциплина**. Разрабатываются методы ее применения в самых различных областях человеческой деятельности.

В 1990-е годы методы компьютерной графики становятся основным **средством организации диалога "человек-компьютер"** и остаются таковыми по настоящее время.

# *5. Опишите основные элементы векторной иллюстрации.*

Основные элементы векторной графики: точка, прямая линия, отрезки прямой, кривая второго порядка, кривая третьего порядка, кривая Безье.

В векторной графике *точке* соответствует узел. На плоскости этот объект представляется двумя числами (X, Y), задающими его положение относительно начала координат.

Для описания *прямой линии* используется уравнение Y = aX + b. Поэтому для построения данного объекта требуется задание всего двух параметров: а и b. Результатом будет построение бесконечной прямой в декартовых координатах.

*Отрезок прямой* требует для своего описания двух дополнительных параметров, соответствующих началу и концу отрезка (например, X1 и Х2).

К классу *кривых второго порядка* относятся параболы, гиперболы, эллипсы и окружности, то есть все линии, уравнения которых содержат переменные в степени не выше второй. В векторной графике эти кривые используется для построения базовых форм (примитивов) в виде эллипсов и окружностей. Кривые второго порядка не имеют точек перегиба. Используемое для описания этих кривых каноническое уравнение требует для своего задания пяти параметров.

*Кривые третьего порядка* могут иметь точку перегиба. Например, график функции Y = Х3 имеет точку перегиба в начале координат (0, 0). Именно эта особенность данного класса функций позволяет использовать их в качестве основных кривых для моделирования различных природных объектов в векторной графике. Следует отметить, что упомянутые раньше прямые и кривые второго порядка являются частным случаем кривых третьего порядка. Каноническое уравнение, используемое для описания уравнения третьего порядка требует для своего задания девяти параметров.

*Кривые Безье* — это частный вид кривых третьего порядка, требующий для своего описания меньшего количества параметров — восьми вместо одиннадцати. В основе построения кривых Безье лежит использование двух касательных, проведенных к крайним точкам отрезка линии. На кривизну (форму) линии влияет угол наклона и длина отрезка касательной, значениями, которых можно управлять в интерактивном режиме путем перетаскивания их концевых точек. Таким образом, касательные выполняют функции виртуальных рычагов, позволяющих управлять формой кривой.

# *6. Опишите свойства векторных объектов.*

Каждый объект располагает своими *собственными свойствами*: слой, цвет, тип и ширина линий. Свойства используют для того, чтобы связать характеристики объектов, которые вы чертите, с характеристиками этих же объектов в реальном мире. Кроме этого, в САПР свойства используют в процессе систематизации объектов для их дальнейшего редактирования и печати.

**Слой** (layer) — кроме всего прочего, это еще одно из свойств каждого объекта. Слои — независимые пространства для создания изображений, которые, как прозрачные пленки, расположены одна над другой, в результате чего и получается полное изображение чертежа.

**Цвет** (Color) – это цвет объекта на экране монитора и на печатной версии чертежа. Для печатной версии чертежа цвет объекта может означать ширину его линий.

**Тип линии** – это отображаемый на экране монитора и на распечатке образец, описывающий размещение в линии штрихов, точек и пустых областей.

**Ширина линии** (Lineweight) — это ширина линии на экране монитора и на распечатке.

**Стиль печати** (Plot Style) – это свойства печати.

Контролировать свойства объектов в программе AutoCAD можно двумя различными способами: на уровне слоя и на уровне объекта. У каждого слоя есть собственные, установленные по умолчанию значения свойств — определенный цвет, тип линии, ширина линии и стиль печати. До тех пор, пока вы не укажете программе, что нужно поступить другим способом, все объекты унаследуют свойства того слоя, на котором они созданы. В AutoCAD такая методика называется контролем свойств на уровне слоя (by layer). Но вы можете заменить установленные таким образом значения, присваивая свойствам объекта специфический цвет, тип линии, ширину линии и стиль печати, которые отличаются от свойств слоя. В AutoCAD такая методика называется контролем свойств на уровне объекта (by object).

# *7. Дайте характеристику приемам и программным средствам точного позиционирования (на примере AutoCAD).*

Первый способ задания первой точки отрезка — указать ее с помощью мыши на видимой части графического экрана, а затем зафиксировать нажатием левой кнопки мыши. При выборе точки можно ориентироваться на счетчик координат.

Второй способ — ввод координат точки с клавиатуры, например: 65,113.24. В данном примере введена точка с двумя координатами: X = 65 мм, Y= 113,24 мм. При вводе координат с клавиатуры запятая является разделителем между абсциссой и ординатой, а точка используется как разделитель между целой и дробной частью числа. Вводимые координаты могут быть целыми или вещественными. При вводе координат следует учитывать, где вы выбрали точку с координатами 0.0. Чаше всего это точка левого угла графического экрана (хотя в процессе работы вы перемещаетесь по рисунку, и точка 0,0 может оказаться в любом месте, даже уйти в невидимую часть чертежа).

Третий способ ввода точек — это относительный ввод в декартовых координатах с клавиатуры, например: @50,25 Данная запись означает, что новая точка задается относительно предыдущей (что определяет символ "@") со сдвигом по оси X на +50 мм (т. е. вправо на 50 мм) и сдвигом по оси Y на +25 мм (т. е. вверх на 25 мм). Здесь запятая также является разделителем координат. Вводимые числа могут быть целыми и вещественными, положительными, нулевыми и отрицательными.

Четвертый способ ввода точек — это относительный ввод в полярных координатах с клавиатуры, например: @33.5

# *8. Опишите способы и форматы ввода координат (на примере AutoCAD).*

**Самый простой способ** задания первой точки отрезка — указать ее с помощью мыши на видимой части графического экрана, а затем зафиксировать нажатием левой кнопки мыши. При выборе точки можно ориентироваться на счетчик координат.

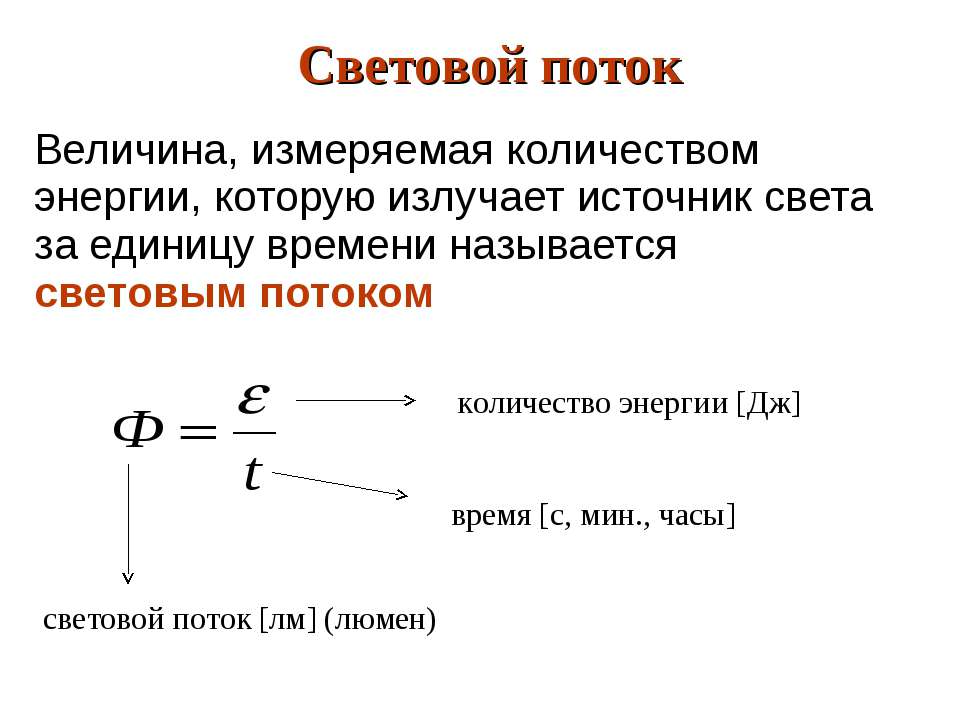
**Второй способ** — ввод координат точки с клавиатуры, например: 65,113.24 В данном примере введена точка с двумя координатами: X = 65 мм, Y= 113,24 мм. При вводе координат с клавиатуры запятая является разделителем между абсциссой и ординатой, а точка используется как разделитель между целой и дробной частью числа. Вводимые координаты могут быть целыми или вещественными. При вводе координат следует учитывать, где вы выбрали точку с координатами 0.0. Чаше всего это точка левого утла графического экрана (хотя в процессе работы вы перемещаетесь по рисунку, и точка 0,0 может оказаться в любом месте, даже уйти в невидимую часть чертежа).

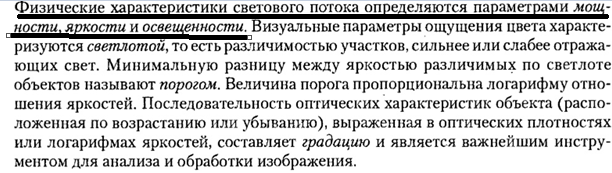
**Третий способ** ввода точек — это относительный ввод в декартовых координатах с клавиатуры, например: @50,25 Данная запись означает, что новая точка задается относительно предыдущей (что определяет символ "@") со сдвигом по оси X на +50 мм (т. е. вправо на 50 мм) и сдвигом по оси Y на +25 мм (т. е. вверх на 25 мм). Здесь запятая также является разделителем координат. Вводимые числа могут быть целыми и вещественными, положительными, нулевыми и отрицательными.

**Четвертый способ** ввода точек — это относительный ввод в полярных координатах с клавиатуры, например: @33.5

Имеется возможность вводить координаты в следующих **форматах**: научном, десятичном, инженерном, архитектурном и дробном. Углы могут вводиться в градах, радианах, топографических единицах (в виде румбов), а также в градусах, минутах и секундах. Управление форматом единиц осуществляется командой ЕДИНИЦЫ.

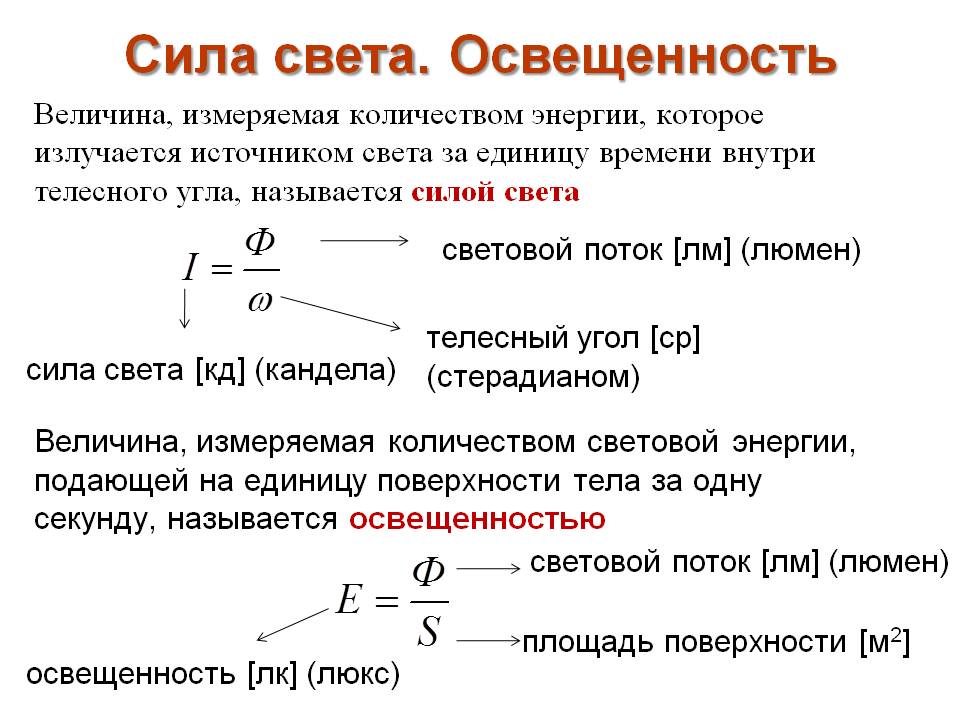
# *9. Опишите физические характеристики светового потока.*



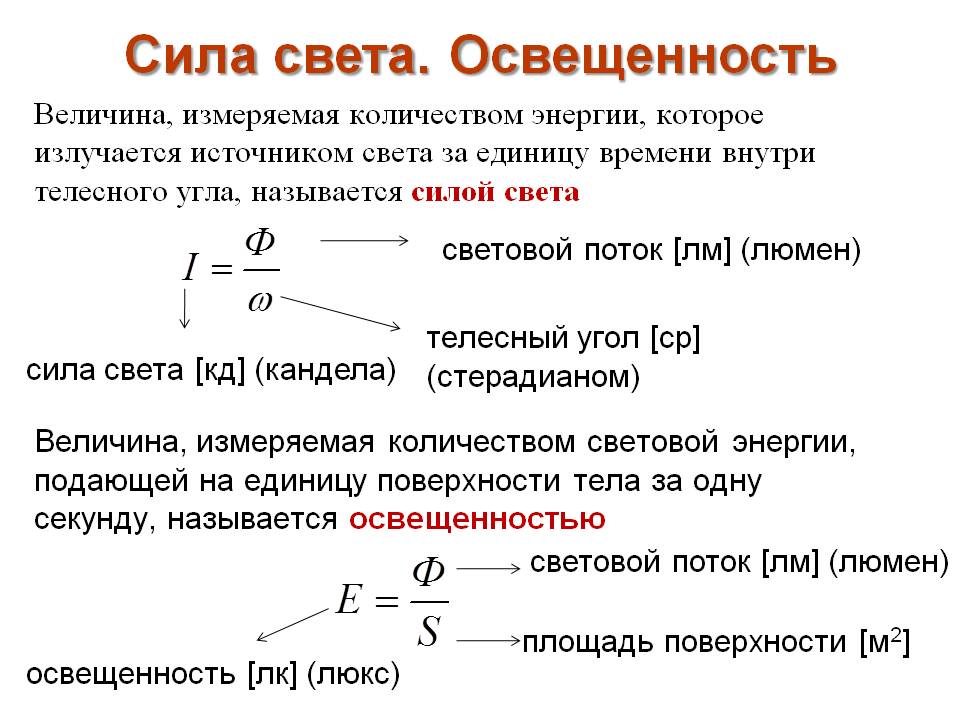


Под световой *мощностью* понимается световая энергия, переносимая излучением через некоторую поверхность за единицу времени.

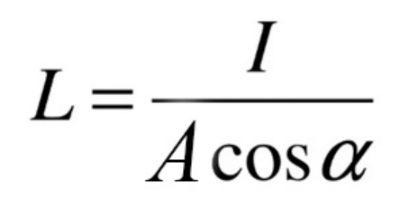
**Мощность светового потока** измеряется в люменах (лм). 1 лм равен потоку света, излучаемому в пределах телесного угла, точечным источником с силой в 1 кд. (кандела).



**Освещенность** (E) — это отношение светового потока, подающего на элемент поверхности, к площади этого элемента. Е=Ф/S, где, S -площадь. Единица освещенности — **люкс** (лк).



**Яркость (L).** Яркость (плотность света) — это отношение светового потока, переносимого в элементарном пучке лучей и распространяющемся в телесном угле, к площади сечения данного пучка. Единица измерения яркости — кд/м2.



Яркость связана с уровнем зрительного ощущения; распространение яркости в поле зрения (в помещении/интерьере) характеризует качество (зрительный комфорт) освещения.

# *10. Дайте характеристику аддитивным цветовым моделям.*

Аддитивные цвета – это цвета от источников освещения, в том числе от солнца.

Первичными аддитивными (прямыми) цветами являются красный, зеленый и синий. Комбинация этих трех цветов создает все остальные оттенки.

В аддитивной цветовой модели мы образуем белый цвет, заполняя черное пространство разными смешанными цветами т.е. идем от чёрного к белому. За основу здесь берется полное отсутствие света (темнота, черный монитор компьютера, экран телевизора и т.д)

Монитор компьютера создает цвет непосредственно излучением света и использует, таким образом, систему цветов RGB.

Аддитивное смешение цветов – метод синтеза цвета, основанный на сложении аддитивных цветов, то есть цветов непосредственно излучающих объектов. Смешивая три основных цвета, можно воспроизвести большинство воспринимаемых человеком цветов.

Современным стандартом для аддитивного смешения цветов является модель цветового пространства RGB, где основными цветами являются красный (**R**ed), зелёный (**G**reen) и синий (**B**lue).

***Цветовая модель RGB***

◼ Используется в цветных мониторах, которые излучают свет

◼ При сложении (смешении) лучей основных цветов результат светлее составляющих.

◼ Цвета этого типа называются *аддитивными*

Аддитивные цвета нашли широкое применение в системах освещения, видеосисте­мах, устройствах записи на фотопленку, мониторах, сканерах и цифровых камерах.

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.

# *11. Дайте характеристику субтрактивным цветовым моделям.*

В отличие от экрана монитора, воспроизведение цветов которого основано на излучении света, печатная страница может только отражать цвет. Поэтому, в данном случае RGB - модель неприемлема. Вместо нее для описания печатных цветов используется модель CMY, базирующаяся на субтрактивных цветах.

Модель CMY описывает цвета, полученные в результате отражения света объектами. Субтрактивные цвета в отличие от аддитивных цветов получаются вычитанием вторичных цветов (голубой, пурпурный и желтый) из общего луча света. **Желтый, пурпурный и голубой** являются базовымидля этой цветовой модели.

Цветовая модель CMY является обратной модели RGB, поэтому белый цвет – это полное отсутствие краски – белый лист бумаги (уровни всех трех составляющих равны 0), присутствие всех цветов дает черный цвет (уровни всех трех составляющих имеют максимальные значения).

На практике при помощи трех базовых красок (Cyan, Magenta, Yellow) нельзя получить весь цветовой диапазон, а при смешивании всех трех составляющих цвет получается не чисто черным, а грязно-коричневым.

Для устранения данного недостатка к трем краскам добавили четвертую – черную (Black), и цветовая модель получила название CMYK - Cyan, Magenta, Yellow, Black. В слове Black используется не первая буква, а последняя, чтобы избежать путаницы с цветом Blue модели RGB.

*Цвета в модели CMYK образуются путем вычитания из черного цветов желтого (Yellow), пурпурного (Magenta) и голубого (Cyan).* Поэтому модель CMYK называется **субтрактивной (вычитающей).**

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

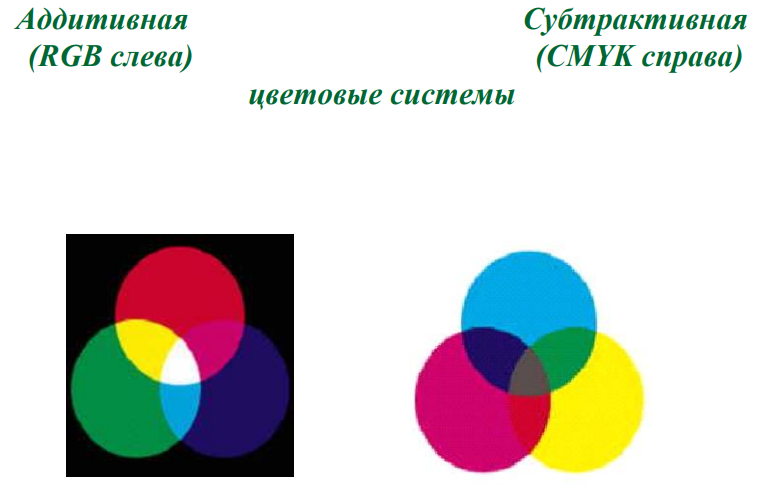
***Цветовая модель CMYK (информация из лекции)***

•Используется при отражении объектов, которые отражают свет

•Смешение составляющих затемняет результирующий цвет (объект поглощает больше цвета).

•Цвета этого типа называются субтрактивными (разностными)





◼ Главная трудность при переходе из системы RGB в CMYK заключается в том, что на бумаге (в системе CMYK) не могут быть представлены некоторые цвета, которые с легкостью можно представить на экране (невозможно точно подобрать полиграфические красители). Поэтому часто то, что на экране выглядит ошеломляюще, на бумаге выглядит блекло и некрасиво.

◼ Это приводит к тому, что смешение трех основных красок, которое должно дать черный цвет, дает неопределенный ("грязный") темный цвет

◼ Для компенсации этого недостатка в число основных полиграфических красок была внесена черная краска. Именно она добавила последнюю букву в название модели (К - blacK -последняя буква (В - занята под обозначение Blue) или, (другая версия) главная - ключевая Key).

# *12. Какие модели опираются на интуитивные понятия тона, насыщенности и яркости?*

Модель HSV *(Hue, Saturation, Value)* построена на основе субъективного восприятия цвета человеком (художником). В цветовом пространстве модели HSV используется цилиндрическая система координат, а множество допустимых цветов представляет шестигранную пирамиду, поставленную на вершину

◼ **Hue** - цветовой тон

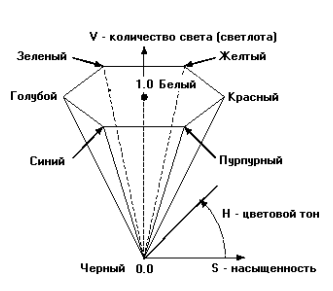
◼ **Saturation** - насыщенность

◼ **Value** - количество света или светлота.

Цветовой тон (Н) измеряется углом, отсчитываемым вокруг вертикальной оси OV. При этом красному цвету соответствует угол 00, зеленому - угол 1200 и т. д. Цвета, взаимно дополняющие друг друга до белого, находятся напротив один другого, т. е. их тона отличаются на 1800. Насыщенность S изменяется от 0 на оси OV до 1 на гранях пирамиды.

Пирамида имеет единичную высоту (V = 1) и основание, рас положенное в начале координат. В основании пирамиды величины Н и S смысла не имеют. Белому цвету соответствует пара S = 1, V = 1. Ось OV(S = 0) соответствует ахроматическим цветам (серым тонам).

Процесс добавления белого цвета к заданному можно представить как уменьшение насыщенности S, а процесс добавления черного цвета -как уменьшение яркости V. Основанию шестигранной пирамиды соответствует проекция RGB куба вдоль его главной диагонали.



По вертикальной оси конуса задается *V - светлота*, меняющаяся от 0 до 1. Значению V = 0 соответствует вершина конуса, значению V = 1 - основание конуса; цвета при этом наиболее интенсивны

*Цветовой тон H* задается углом, отсчитываемым вокруг вертикальной оси. В частности, 0° - красный, 60° - желтый, 120° - зеленый, 180° - голубой, 240° - синий, 300° - пурпурный, т.е. дополнительные цвета расположены друг против друга (отличаются на 180°)

*Насыщенность S* определяет, насколько близок цвет к "чистому" пигменту и меняется от 0 на вертикальной оси V до 1 на боковых гранях шестигранного конуса.

**Модель HLS**

Еще одним примером системы, построенной на интуитивных понятиях тона насыщенности и яркости, является система HLS *{Hue, Lightness, Saturation).* Здесь множество всех цветов представляет собой двойную шестигранную пирамиду, являющуюся расширением одиночной пирамиды HSV (рисунок 3.6).

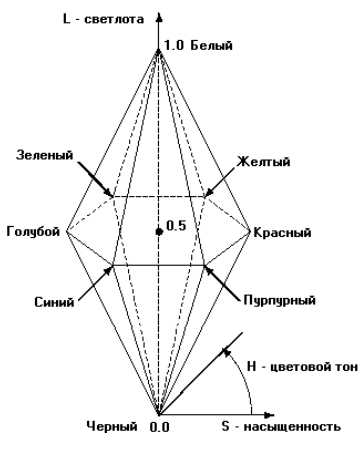
RGB-модель может применяться для задания цвета, однако эксперименты показывают, что модели HSV и HLS позволяют выбирать цвет аккуратнее. Процедура перевода координат из системы RGB в системы HSV и HLS довольно проста. Достаточно спроектировать куб на плоскость вдоль главной диагонали куба.

◼ **Hue** - цветовой тон. Цветовой тон позволяет различать цвета, такие как красный, зеленый, желтый и т.д.

◼ **Lightness** - светлота. Светлота отражает полутоновое представление об интенсивности, как о факторе, не зависящем от цветового тона и насыщенности.

◼ **Saturation** - насыщенность. Насыщенность характеризует чистоту, т.е. степень ослабления (разбавления) данного цвета белым, и позволяет отличать розовый от красного, небесно-голубой от ярко-синего и т.п.

HLS можно представить, как полученную из HSV "вытягиванием" точки V=1, S=0, задающей белый цвет, вверх для образования верхнего конуса



# *13. Дайте характеристику форматам растровой графики.*

***TIFF***

Предложенный компанией Aldus формат TIFF ближе всех к статусу стандартного. Формат TIFF позволяет сохранять растровые изображения с компрессией без потери качества. Помимо традиционных цветов CMY формат поддерживает цветоделение с большим числом красок. Принцип хранения данных основан на использовании специальных маркеров (тэгов) в сочетании с битовыми последовательностями кусков растра.

***GIF***

Первая версия формата GIF была разработана в 1987 г. специалистами компьютерной сети CompuServe. Как и у программ-архиваторов, степень сжатия графической информации в GIF сильно зависит от уровня ее повторяемости и предсказуемости, а иногда еще и от ориентации картинки. GIF может иметь любое количество цветов от двух до 256-ти.

Изменив порядок следования данных в файле, создатели GIFa заставили картинку становиться все четче и детальнее по мере подхода из сети новых данных. Для этого файл с изображением тасуется при записи так, чтобы сначала шли все строки пикселов с номерами, кратными восьми, затем четырем, потом двум и, наконец, последний проход – все оставшиеся строки с нечетными номерами. Во время приема и декодирования такого файла каждый следующий проход заполняет пустоты в предыдущих, постепенно приближая изображение к исходному состоянию.

Другой полезной возможностью формата является использование прозрачности. Формат может быть использован для создания анимационных изображений.

***BMP***

Формат BMP был создан компанией Microsoft и широко используется в операционных системах семейства Windows для растровой графики. В файлах BMP информация о цвете каждого пиксела кодируется 1, 4, 8, 16, 24, 32, 48 бит (бит/пиксел).

В формате BMP изображения могут храниться как есть или же с применением некоторых распространённых алгоритмов сжатия. В частности, формат BMP поддерживает сжатие без потери качества, а современные операционные системы и программное обеспечение позволяют использовать JPEG и PNG (эти форматы встраиваются в BMP как в контейнер).

Данные изображения – это последовательность пикселей, записанных в том или ином виде. Пиксели хранятся построчно, снизу вверх. Каждая строка изображения дополняется нулями до длины, кратной четырём байтам.

***JPEG***

JPEG — один из популярных графических форматов, применяемый для хранения фотоизображений и подобных им изображений.

Алгоритм JPEG в наибольшей степени пригоден для сжатия фотографий и картин, содержащих реалистичные сцены с плавными переходами яркости и цвета. Наибольшее распространение JPEG получил в цифровой фотографии и для хранения и передачи изображений с использованием сети Интернет.

С другой стороны, JPEG малопригоден для сжатия чертежей, текстовой и знаковой графики, где резкий контраст между соседними пикселами приводит к появлению заметных артефактов.

# *14. Дайте характеристику форматам векторной графики.*

**2D:**

* Scalable Vector Graphics (SVG и SVGZ);
* Encapsulated PostScript (EPS);
* Метафайлы Windows: WMF, EMF;
* Файлы CorelDraw: →CDR, CMX;
* Adobe Illustrator (AI).

**3D:**

* COLLADA — формат, разработанный для обмена между 3D приложениями;
* SKP;
* STL — A stereolithography format;
* U3D — Universal 3D file format;
* VRML — Virtual Reality Modeling Language;
* .3ds.

***SVG*** — язык разметки масштабируемой векторной графики, входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML. Поддерживает как неподвижную, так и анимированную интерактивную. Не поддерживает описание трёхмерных объектов. Это открытый стандарт, который является рекомендацией консорциума W3C. Разрабатывается с 1999 года.

***EPS*** — формат файлов, базирующийся на подмножестве языка PostScript и предназначенный для обмена графическими данными между различными приложениями. Формат EPS был создан компанией Adobe и послужил базой для создания ранних версий формата Adobe Illustrator. Данные в файле EPS должны записываться в соответствии со стандартом DSC, файл не может содержать представление более чем одной страницы.

В своей минимальной конфигурации EPS-файл имеет так называемый BoundingBox DSC comment — информацию, описывающую размер изображения. Формат используется в профессиональной полиграфии и может содержать растровые изображения, векторные изображения, а также их комбинации. Изображение, записанное в формате EPS, может быть сохранено в разных цветовых пространствах: Grayscale, RGB, CMYK, Lab, Multi-channel.

***CDR*** — файл проекта, созданный в программе Corel DRAW, который содержит векторное изображение или растровый рисунок. Данный формат файла разработан компанией Corel для использования в собственных программных продуктах.

Так как формат CDR проприетарный, он не поддерживается многими сторонними программами, предназначенными для редактирования изображений. Однако, с помощью свободного конвертора uniconvertor возможна конвертация в свободные форматы, например в SVG.

Полигональные сетки могут храниться в множестве файловых форматов:

***FBX*** - технология и формат файлов (.fbx) разрабатываемый компанией Autodesk. Используется для обеспечения совместимости различных программ трехмерной графики;

***COLLADA*** — это формат, разработанный для обмена между 3D приложениями. COLLADA использует открытый стандарт XML для обмена форматами, которые в противном случае были бы несовместимы. COLLADA был задуман как промежуточный формат для переноса файлов.

***DXF*** — открытый формат файлов для обмена графической информацией между приложениями САПР. Был создан фирмой Autodesk для системы AutoCAD. Поддерживается практически всеми CAD-системами на платформе PC.

***OBJ***— это формат файлов описания геометрии. Формат файла является открытым и был принят другими разработчиками приложений 3D графики. По большей части это общепринятый формат.

Формат файлов OBJ — это простой формат данных, который содержит только 3D геометрию, а именно, позицию каждой вершины, связь координат текстуры с вершиной, нормаль для каждой вершины, а также параметры, которые создают полигоны.

***STL*** — формат файла, широко используемый для хранения трехмерных моделей объектов для использования в технологиях быстрого прототипирования, обычно, методом стереолитографии. Информация об объекте хранится как список треугольных граней, которые описывают его поверхность, и их нормалей. STL-файл может быть текстовым (ASCII) или двоичным.

***VRML*** — это текстовый формат файлов, где, например, вершины и грани многогранников могут указываться вместе с цветом поверхности, текстурами, блеском, прозрачностью и так далее. Движение, звуки, освещение и другие аспекты виртуального мира могут появляться как реакция на действия пользователя или же на другие внешние события, например таймеры. Особый компонент Script Node позволяет добавлять программный код к VRML-файлу.

VRML-файлы обычно называются мирами и имеют расширение .wrl. Хотя VRML-миры используют текстовый формат, они часто могут быть сжаты с использованием алгоритма компрессии gzip для того, чтобы их можно было передавать по сети за меньшее время. Большинство программ трёхмерного моделирования могут сохранять объекты и сцены в формате VRML.

# *15. Опишите устройства ввода графической информации.*

Существуют различные технические средства, осуществляющие процесс преобразования изображений в цифровую форму, например: *сканеры, дигитайзеры (графические планшеты), цифровые фото- и видеокамеры*.

***Сканер***

Сканером называется устройство, позволяющее вводить в компьютер образы изображений, представленных в виде текста, рисунков, слайдов, фотографий или другой графической информации.

* Ручные сканеры
* Листопротяжные сканеры
* Барабанные сканеры
* Планшетные сканеры

*Основные характеристики сканеров*

Оптическое разрешение - измеряется в точках на дюйм. Характеристика, показывающая, чем больше разрешение, тем больше информации об оригинале может быть введено в компьютер и подвергнуто дальнейшей обработке.

Глубина цвета – это характеристика, обозначающая количество цветов, которое способен распознать сканер.

Оптическая плотность есть характеристика оригинала, равная десятичному логарифму отношения света, падающего на оригинал, к свету отраженному. Минимально возможное значение 0.0 D - идеально белый оригинал. Значение 4.0 D – абсолютно черный оригинал.

Наиболее популярные модели сканеров производят Hewlett-Packard, Agfa, Canon, Mustek.

***Дигитайзер***

Дигитайзер (графический планшет) — это устройство, предназначенное для оцифровки изображений, применяемое для создания на компьютере рисунков и набросков. Как правило, планшет используют профессиональные художники для более точной обработки (создания) изображений. Дигитайзер, или планшет, как его часто называют, состоит из двух основных элементов: основания и курсора, двигающегося по его поверхности. Принцип действия дигитайзера основан на фиксации местоположения курсора с помощью встроенной в планшет сетки. При нажатии на кнопку курсора его местоположение на поверхности планшета фиксируется, а его координаты передаются в компьютер.

*Основные характеристики дигитайзера*

Механизм регистрации позволяет получить шаг считывания информации намного меньше шага сетки (до 100 линий на мм). Шаг считывания информации называется разрешением дигитайзера.

Шаг считывания регистрирующей сетки является физическим пределом разрешения дигитайзера.

Важной характеристикой дигитайзера является регистрируемое число степеней нажатия электронного пера. Программа-обработчик использует эту величину, устанавливая в зависимости от нее, например, толщину проводимой линии.

Наиболее популярны модели следующих фирм: CalComp, NUMONICS, WACOM.

***Цифровая камера***

Цифровая камера получает изображения, обрабатывает их и хранит в цифровом формате. Вместо пленки она использует встроенную или сменную полупроводниковую память, чтобы хранить снимки. Она обладает теми же основными свойствами, что и нормальная фотокамера, и, помимо этого, может соединяться с компьютером, телевизором или принтером.

В настоящее время на рынке работают десятки известнейших фирм-производителей как традиционного фотооборудования и материалов (Kodak, Konica, Nikon, Fuji, Agfa, Olympus и др.), так и компьютерной периферии и прочего электронного оборудования (Hewlett-Packard, SeikoEpson, Sony, Mustek, UMAX, LGElectronics, Minolta)

# *15 Новый.* *Дайте характеристику комплексным форматам*

Существуют также комплексные форматы, которые могут хранить как векторную, так и растровую информацию. Это форматы **DjVu**, **CGM**, **AI** (формат программы Adobe Illustrator), **EPS** (Encapsulated PostScript – профессиональный универсальный векторно-растровый формат, используемый всеми профессиональными графическими программами) и **PDF** (Portable Document Format – формат программы Adobe Acrobat, который может содержать растровую и векторную графику, а также текстовую информацию).

**DjVu** стал основой для нескольких библиотек научных книг. Огромное количество книг в этом формате доступно в файлообменных сетях.

Однако, при этом не стоит забывать, что в DjVu используется сжатие данных с потерями. Для особо важных документов, возможно, будет разумнее использовать более «надёжные» форматы: PNG, JPEG 2000, TIFF и т. п. В общей сложности выигрыш объёма в этом случае составляет 4--10 раз.

**Portable Document Format (PDF)** — межплатформенный открытый формат электронных документов, изначально разработанный фирмой Adobe Systems в 1992 году с использованием ряда возможностей языка PostScript. В первую очередь предназначен для представления полиграфической продукции в электронном виде. Для просмотра существует множество программ, а также официальная бесплатная программа Adobe Reader. Значительное количество современного профессионального печатного оборудования имеет аппаратную поддержку формата PDF, что позволяет производить печать документов в данном формате без использования какого-либо программного обеспечения. Традиционным способом создания PDF-документов является виртуальный принтер, то есть документ как таковой готовится в своей специализированной программе — графической программе или текстовом редакторе, САПР и т. д., а затем экспортируется в формат PDF для распространения в электронном виде, передачи в типографию и т. п.

# *16. Опишите устройства вывода графической информации.*

**Устройства для вывода визуальной информации**

* Монитор (дисплей)
* Принтер
* Графопостроитель

**Процесс преобразования информации при выполнении вывода**

Процесс преобразования при выполнении вывода визуальной информации может быть представлен состоящим из следующих этапов:

1. Модельные преобразования. Проблемно-ориентированный уровень из геометрических моделей отдельных объектов, задаваемых в собственных локальных системах координат, формирует описание совокупного объекта в некоторой единой (мировой) системе координат. Описание совокупного объекта подается в графическую систему.

2. Нормализующие преобразования. Графическая система переводит описание из мировой, вообще говоря, произвольной, системы координат в нормализованные координаты устройства, имеющие фиксированные пределы изменения координат, например, от 0.0 до 1.0

3. Преобразования сегментов. Если графическая система предоставляет средства манипулирования отдельными подкартинами изображения (часто именуемыми сегментами), например, для независимого размещения отдельных самостоятельных частей изображения, то могут потребоваться такие преобразования.

4. Видовые преобразования. В случае 3D описания изображения и 2D устройства вывода необходимо выполнить проецирование изображения на заданную картинную плоскость. Наоборот, при 2D сцене и 3D устройстве вывода необходимо выполнить преобразование, связанное с размещением изображения. При выполнении этих преобразований, естественно, может потребоваться выполнение отсечения частей изображения. После этого этапа по сути дела готово описание изображения в некоторой аппаратно-независимой форме, пригодной для вывода на любое устройство.

5. Преобразование рабочей станции. Для выполнения вывода на конкретное устройство необходимо преобразование данных из аппаратно-независимой формы в координаты устройства. Процесс преобразования координатной информации при вводе от координатных устройств является обратным преобразованию при выводе.

**Монитор**

Монитор — конструктивно законченное устройство, предназначенное для визуального отображения информации.

Современный монитор состоит из экрана (дисплея), блока питания, плат управления и корпуса. Информация для отображения на мониторе поступает с электронного устройства, формирующего видеосигнал (в компьютере — видеокарта). В некоторых случаях в качестве монитора может применяться и телевизор.

*Классификация мониторов*

По виду выводимой информации

1. алфавитно-цифровые

2. графические, для вывода текстовой и графической (в том числе видео) информации

* векторные (vector-scandisplay)
* растровые (raster-scandisplay)

По размерности отображения

* двумерный (2D) — одно изображение для обоих глаз;
* трёхмерный (3D) — для каждого глаза формируется отдельное изображение для получения эффекта объёма.

**Графопостроитель** (плоттер) устройство для вывода данных в графической форме на бумагу, пластик, фоточувствительный материал или иной носитель путем черчения.

**Проектор** – это устройство вывода, которое берет изображение на экране компьютера и проецирует его в большом размере на плоскую поверхность. Проекторы часто используются на совещаниях и при показе презентаций, чтобы все присутствующие в комнате могли с удобством просматривать слайды.

**Принтер** (матричный принтер, струйный принтер и лазерный принтер) то устройство для вывода цифровой информации на бумагу.  
Существуют три основных типа принтеров: матричные, струйные и лазерные.

1. Матричные принтеры формируют изображение построчно с помощью печатающей головки, которая ударяет по бумаге через красящую ленту. Головка содержит ряд иголок (pin), от количества которых зависит качество изображения.
2. Струйные принтеры печатают путем набрызгивания чернил на бумагу через мелкие сопла в печатающей головке.
3. В лазерных принтерах изображение создается путем переноса на бумагу специального порошка (тонера). Источник света (лазер) освещает предварительно заряженную поверхность фотобарабана. На тех местах, куда попал свет, меняется заряд, и к ним притягивается тонер. Затем тонер за счет электростатики переносится на бумагу, после чего попадает в печку, где и закрепляется под действием высокой температуры. Качество такого изображения очень высокое.

# *17. Опишите способ сжатия графической информации без потерь.*

Сжатие "без потерь» — это отсутствие информационных потерь, а именно: такие алгоритмы гарантируют, что после декомпрессии информация совпадет "бит в бит" с исходной.

В качестве примеров таких алгоритмов сжатия без потерь можно рассмотреть следующие (как я поняла, нужно расписать подробно только один):

* кодирование длин серий;
* метод Хаффмана;
* алгоритм LZW;
* арифметическое сжатие.

***1 Кодирование длин серий***

Один из самых простых способов сжатия файлов - кодирование длин серий. Серии повторяющихся величин заменяются единственной величиной и количеством. Например:

abbbbbbccdddttddd => 1a6b2c3d2t3d.

Этот алгоритм просто реализуется и хорошо работает с длинными сериями повторяющихся величин. Например: для изображений с большими областями постоянной яркости или цвета.

Одна из популярных реализаций группового сжатия - PackBits, используемая для bitmap данных на AppleMacintosh. Предполагая 8-битные данные, PackBits кодирует регулярные величины двумя битами.

Групповое кодирование используется в таких форматах, как MacPaint, TIFF, GEM, PCX, FLI.

***2 Кодирование методом Хаффмана***

Данные заменяются более эффективными кодами. Этот подход, созданный в 1952 г. для текстовых файлов, породил множество вариантов.

Основная схема - присвоение двоичного кода каждой уникальной величине, причем длина этих кодов различна. Более короткие коды не пользуются для более часто появляющихся величин. Значение кодов хранятся в таблице перекодировки.

Например: abbbcccddeeeeeeeeef.

Частота, с которой повторяются уникальные величины a:1 b:3 c:3 d:2 e:9 f:1.

Для формирования минимального кода используется двоичное дерево. Алгоритм объединяет вместе элементы, появляющиеся наименее часто; затем пара рассматривается как один элемент и их частоты объединяются. Это повторяется до тех пор, пока все элементы не объединяются в пары.

Наиболее редко используемые значения a и f, так что они становятся первой парой; a - присваивается нулевая ветвь, а f - первая. Это означает, что 0 и 1 будут младшими битами кодов для а и f. Старшие биты будут получены из дерева по мере того, как оно будет построено.

Эффективность алгоритма Хаффмана изменяется с точностью алгоритма и типом изображения. Алгоритм достигает сжатия 8:1.

Кодирование по Хаффману нуждается в точной статистике, поэтому для обеспечения правильной работы, алгоритм часто реализуется в два прохода. В первый проход создается статистическая модель; во второй - кодируются данные. В результате, и благодаря тому, что коды переменной длины требуют больше времени для обработки компрессии и декомпрессии по Хаффману – это сравнительно медленные процессы.

***3 Схема сжатия LZW (1984)***

LZW - более современный алгоритм сжатия. В отличие от схемы Хаффмана, перед кодированием не требуется создавать таблицу кодов. Начиная с простой таблицы кодов алгоритм формирует более эффективную таблицу по мере продвижения. Этот алгоритм является адаптивным.

LZW начинается с таблицы с одним элементом кода для каждой возможной величины (для восьми бит - 256 элементов). Затем она добавляет в таблицу данные для каждого уникального имени, которое обнаруживает. Необходимо формировать максимальную длину таблицы потому, какой длины код может быть представлен. (Некоторые реализации LZW допускают переменную длину).

***ПРИМЕР*.** Ababaaacaaaad

Каждая буква - два бита (всего четыре буквы). Начальная кодовая таблица: а:00, b:01, c:10, d:11.

LZW ищет самую длинную последовательность. Он находит **а** и распознает, затем проверяет **ab** и не распознает. Тогда он заводит код для этой величины а(000) и делает новый элемент для величин, которые не распознает. ab таблица: a:000, b 001, c:010, d:011, ab:100. Шифратор затем берет последнюю величину b и проверяет последовательность со следующей величиной ba - не распознается => код для **ba**. Шифратор и дешифратор могут определить следующий элемент ab. Трехбитный код заменяет два двухбитных.

Типичные коэффициенты сжатия при LZW от 1:1 до 3:1, хотя некоторые изображения сжимаются 10:1. Изображение с “шумом” плохо сжимаются. LZW используется в gif и tiff.

***4 Арифметическое сжатие***

Арифметическое сжатие, подобно кодированию по Хаффману, использует более короткие коды для часто появляющихся участков и более длинные для редко появляющихся. Это более эффективный алгоритм, хотя он (подобно LZW) сжимает последовательность величин, а не сами величины.

Так же, как и код Хаффмана, арифметическое кодирование эффективно представляет изображения с более часто появляющимися последовательностями пикселей (более вероятные величины) с помощью меньшего количества бит.

Арифметическое сжатие может эффективно уменьшать размер файла в зависимости от распределения интенсивности и от точности используемой статистической модели. Иногда сжатие бывает 100:1.

# *18. Опишите способ сжатия графической информации с потерями.*

Исследователями визуального восприятия человека отмечено, что далеко не вся информация требуется для того, чтобы адекватно воспринимать цветное изображение. Для реализации этого закона были разработаны алгоритмы с потерей информации, которые обеспечивают выбор уровня компрессии с уровнем качества изображения. Тем самым достигается компромисс между размером и качеством изображений.

Наиболее известным методом компрессии с потерями является JPEG-компрессия. Метод компрессии основан на особенности человеческого восприятия: глаз достаточно четко различает яркость объекта и цветовые контрасты, а плавные изменения в светах и тенях значительно меньше. При записи такой изобразительной информации часть цветовых данных может быть опущена, как предполагается, без заметного ущерба для восприятия.

Для этого обработка изображения происходит в несколько этапов.

1 этап. Изображение конвертируется в особое цветовое пространство, напоминающее цветовую модель **CIE Lab**, в котором один канал сохраняет яркостные характеристики, а в остальных двух цветовых каналах уменьшается разрешение (по методу мозаики).RGB-изображение конвертируется в пространство **YUV** (иногда называемое также **YcrCb**), основанное на характеристиках яркости (составляющая Y) и цветности (составляющие U и V).

2 этап. Изображение разбивается на фрагменты квадратной формы со стороной в 8 пикселов. Каждый фрагмент подвергается достаточно сложным математическим преобразованиям.

Название этого преобразования – дискретное косинус-преобразование (**ДКП**). Одновременно каждый блок разлагается на составляющие цвета и производится подсчет частоты встречаемости каждого цвета. Информация о частоте позволяет исключить небольшую часть яркостной характеристики и довольно значительную цветовой. Уровень исключения информации как раз и определяется установкой требуемого качества.

3 этап. Затем информация о яркости и цвете кодируется таким образом, что остаются только различия между соседними блоками. Результатом всего процесса обработки являются последовательности простых чисел, которые в свою очередь легко сжимать каким-либо алгоритмом сжатия без потерь из уже упомянутых, например алгоритмом Хаффмана.

Алгоритмы сжатия с потерями, в частности алгоритм **JPEG**, не позволяют полностью восстановить изображение до его исходного состояния, а следовательно, не рекомендуется сжимать изображения несколько раз.

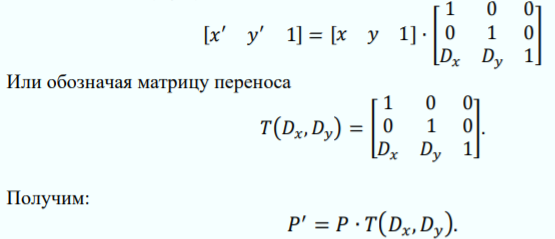
# *19. Аффинная геометрия. Свойства аффинных преобразований. Перемещение.*

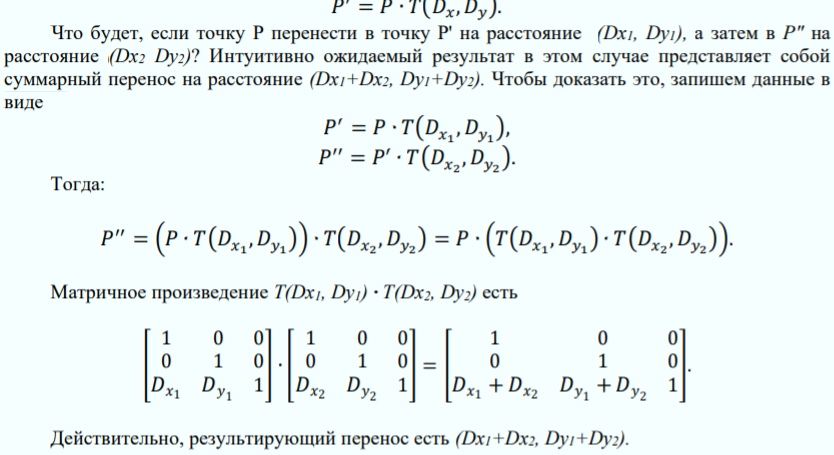
**Аффинная геометрия** – раздел геометрии, в котором изучаются свойства фигур, сохраняющиеся при любых аффинных преобразованиях, или, что равносильно, при параллельной проекции.

**Свойства аффинных преобразований**

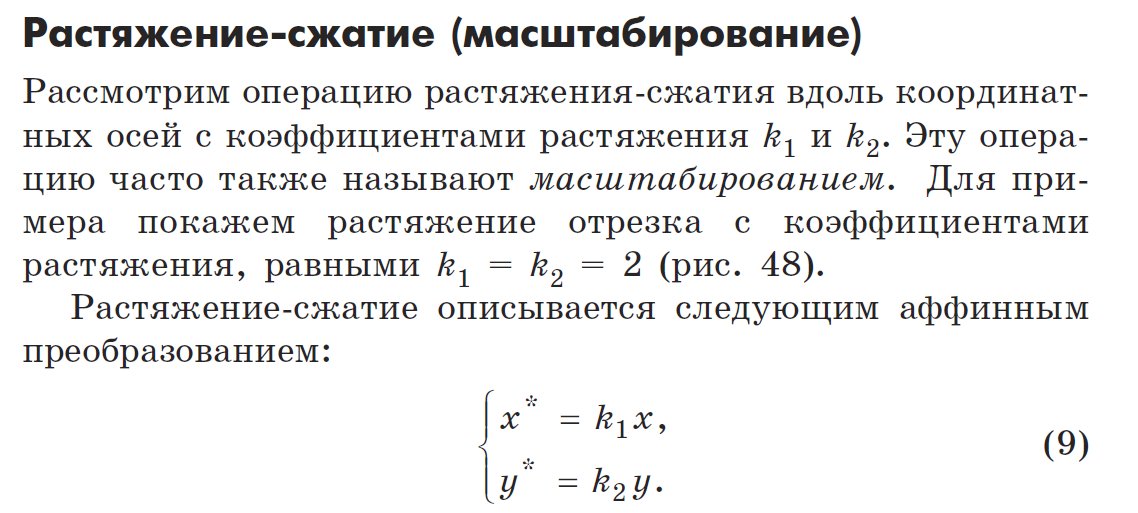
1. Все точки плоскости преобразуются взаимно однозначно. (Аффинное преобразование , переводя любую прямую a в некоторую прямую f(a) отображает множество точек прямой f(a).)
2. Каждая прямая переходит в прямую. (Если прямые a и b параллельны a || b то и их образы при аффинном преобразовании f f(a) и f(b) также параллельны.), аналогично для плоскостей.
3. Сохраняет пропорции параллельных объектов – длин отрезков на параллельных прямых и площадей на параллельных плоскостях.

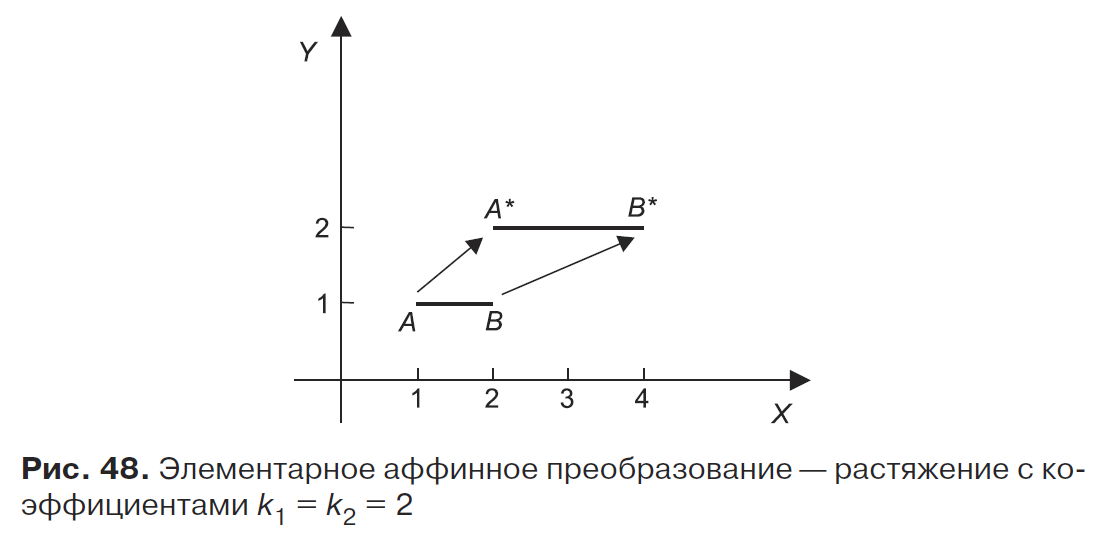
Перенос точки производится добавлением к каждой координате точки положительной или отрицательной константы. Если обозначить константы переноса через Dх (перенос в направлении оси х) и Dу (перенос в направлении оси у), получим новые координаты P' (x',y',1) перенесенной точки Р(x,y,1):

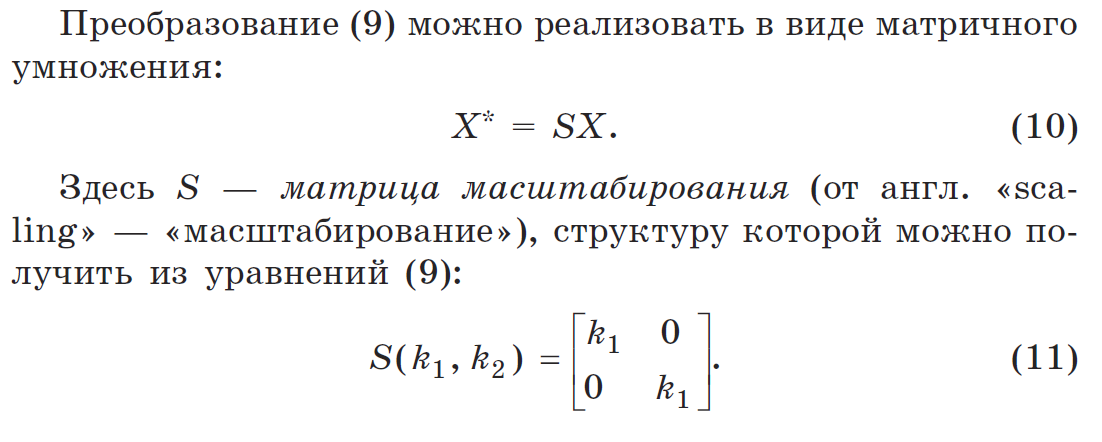




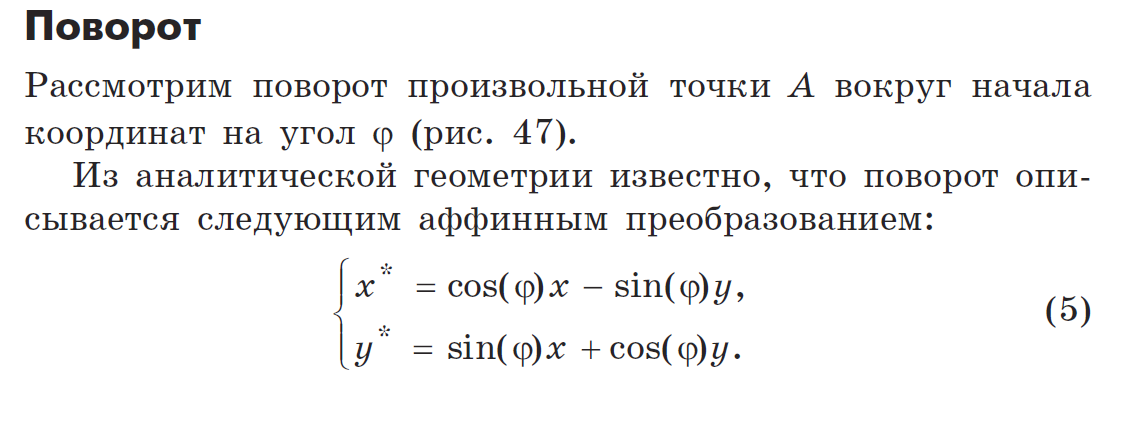
# *20. Аффинная геометрия. Свойства аффинных преобразований. Масштабирование.*

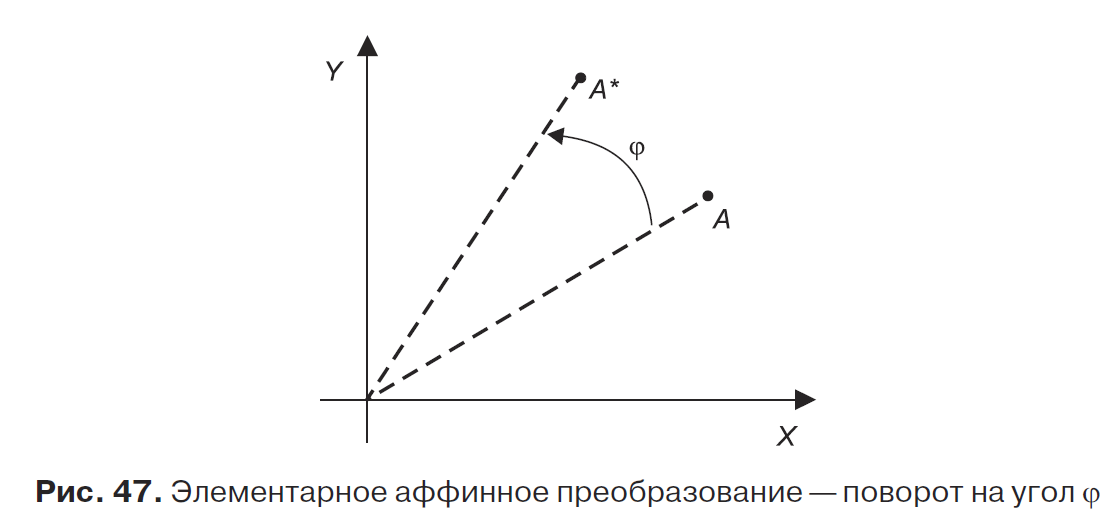


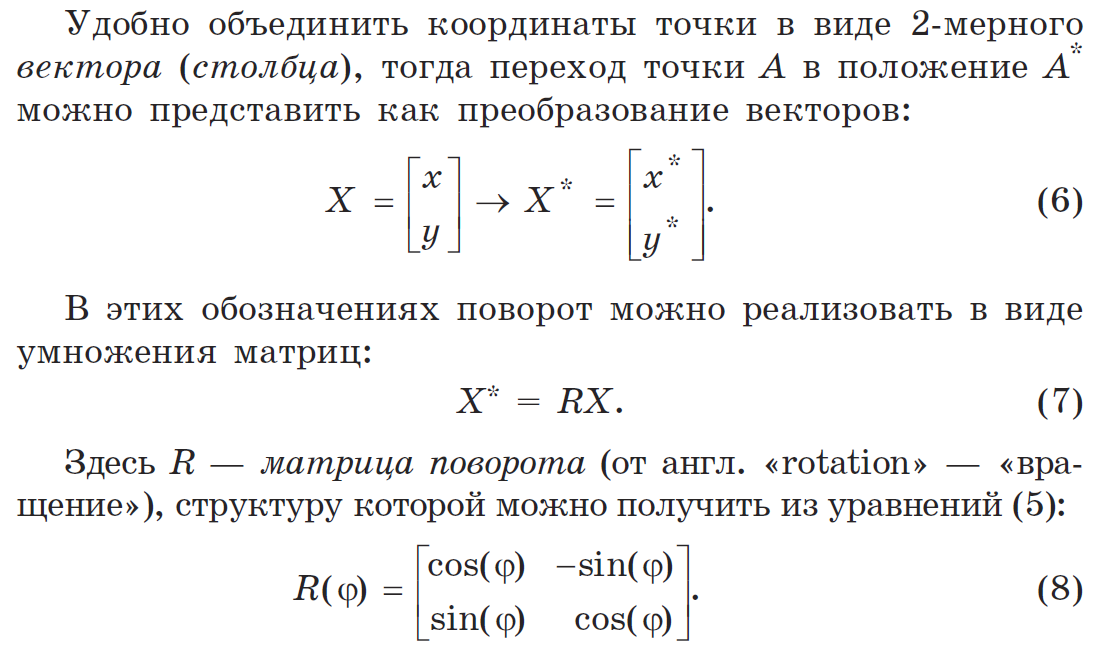




# *21. Аффинная геометрия. Свойства аффинных преобразований. Поворот.*







# *22. Основные матрицы 3D- геометрических преобразований.*

Приведем основные матрицы 3D- геометрических преобразований.

**1. Матрица поворота вокруг оси Х на угол g**

Исходная задача сводится к задаче поворота в двухмерном пространстве. Если точка (x,y,z) преобразуется в точку (x’,y’,z’), то

x’= x\*1 + y\*0 + z\*0

y’= x\*0 + y\*cos γ + z\*sin γ

z’= x\*0 – y\*sin γ + z\*cosγ.





**2. Матрица поворота вокруг оси Y на угол b**

Задача также сводится к задаче поворота в двухмерном пространстве. Если точка (x,y,z) преобразуется в точку (x’,y’,z’), то

x’=x\*cos β + y\*0 - z\*sin β

y’=x\*0 + y\*1 + z\*0

z’= x\*sinβ + y\*0 + z\*cosβ



**3. Матрица поворота вокруг оси Z на угол a**

Задача сводится к задаче поворота в двухмерном пространстве. Если точка (x,y,z) преобразуется в точку (x’,y’,z’), то

x’= x\*Сos α + y\*Sin α + z\*0

y’= - x\*Sin α + y\*cosα + z\*0

z’= x\*0 + y\*0 + z\*1.

  
**4. Матрица масштабирования по осям**



Размерность всех матриц на единицу больше числа измерений. Последовательное применение двух преобразований А и В может быть заменено использованием одного преобразования С = А·В. Необходимо соблюдать последовательность преобразований, так как А·В ≠ В·А. Каждому преобразованию А соответствует обратное А-1.

В трехмерной графике добавляется еще одно (четвертое) геометрическое преобразование – проецирование.

# *23. Опишите алгоритм Брезенхама для вычерчивания отрезков прямых линий.*

Алгоритм Брезенхема — это алгоритм, определяющий, какие точки двумерного растра нужно закрасить, чтобы получить близкое приближение прямой линии между двумя заданными точками.

Отрезок проводится между двумя точками — x0,y0 {\displaystyle (x\_{0},y\_{0})} и x1,y1 {\displaystyle (x\_{1},y\_{1})}, где в этих парах указаны столбец и строка соответственно, номера которых растут вправо и вниз. Сначала мы будем предполагать, что наша линия идёт вправо и вниз, причём горизонтальное расстояние x1 – x0  
превосходит вертикальное y1 – y0, то есть наклон линии от горизонтали — менее 45°. Наша цель состоит в том, чтобы для каждого столбца x между x0 и x1 определить , какая строка *y* ближе всего к линии, и нарисовать точку x,y

Общая формула линии между двумя точками:



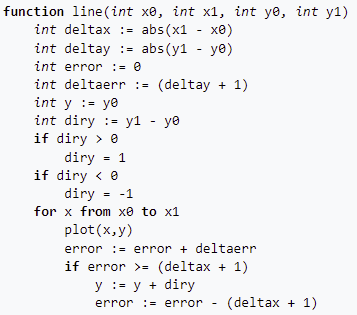
Поскольку мы знаем колонку x, то строка y получается округлением к целому следующего значения:



Однако, вычислять точное значение этого выражения нет необходимости. Достаточно заметить, что y уменьшается от y0 и за каждый шаг мы добавляем к x единицу и добавляем к y значение наклона (в нашем случае значение наклона будет отрицательным числом):



Что из этих двух выбрать — можно решить, отслеживая значение ошибки, которое означает — вертикальное расстояние между текущим значением y и точным значением y для текущего x. Всякий раз, когда мы увеличиваем x, мы увеличиваем значение ошибки на величину наклона s, приведённую выше. Если ошибка превысила 1.0, линия стала ближе к следующему y, поэтому мы увеличиваем y на 1.0, одновременно уменьшая значение ошибки на 1.0. В реализации алгоритма, приведённой ниже, plot(x,y) рисует точку, а abs возвращает модуль числа, для того чтобы избавится от вещественных величин домножаем все вещественные числа на deltax + 1.



Необходимость прибавлять единицу к deltax и deltay вызвана тем, что функция должна строить линию от точки (x0, y0) до точки (x1, y1) включительно! Теперь мы можем быстро рисовать линии, направленные вправо-вниз с величиной наклона меньше 1. Осталось распространить алгоритм на рисование во всех направлениях. Это достигается за счёт зеркальных отражений, то есть заменой знака (шаг в 1 заменяется на −1), обменом переменных *x* и *y*, обменом координат начала отрезка с координатами конца.

# *24. Опишите алгоритм Брезенхама для вычерчивания окружностей.*

Алгоритм Брезенхема по методу построения он похож на рисование линии. В этом алгоритме строится дуга окружности для первого квадранта, а координаты точек окружности для остальных квадрантов получаются симметрично. На каждом шаге алгоритма рассматриваются три пикселя, и из них выбирается наиболее подходящий путём сравнения расстояний от центра до выбранного пикселя с радиусом окружности.

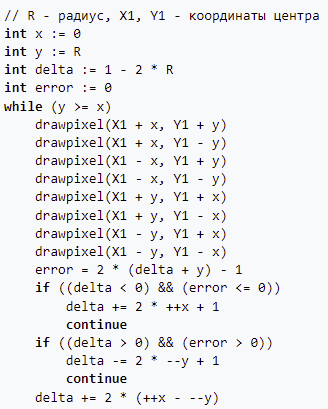
Алгоритм основан на приращении координат х и у на величину погрешности между ними. Значение погрешности присваивается переменной δ. Полученная функция выполняет запись точек по окружности, используя свойство симметрии окружности, что позволяет вычислять точки только на дуге 45°. При этом х меняется от х = 0 до х = у = r / √2 (тогда y = r … r / √2), где r – радиус.

Итак, алгоритм:

- вычисляем δ1 = (3 - 2r) и засвечиваем начальный пиксель (х = 0, y = r);

- вычисляем δi+1 по значению δi и в зависимости от него засвечиваем (xi+1, yi+1); по одной (каждой) вычисленной точке засвечиваем сразу 8 симметричных пикселей;

- завершаем алгоритм по достижении xi = r/√2.



# *25. Опишите алгоритмы устранения лестничного эффекта.*

Приведенные выше алгоритмы в чистом виде не используются из-за лестничного эффекта (aliasing *Алиасинг*): при выводе наклонных линий пиксели образуют ступеньки.

Поэтому в алгоритмы Брезенхама вводятся уточняющие методы, позволяющие сгладить лестничный эффект.



Такие методы можно разделить на две группы: 1) алгоритмы генерации сглаженных изображений и 2) алгоритмы сглаживания уже сформированных изображений.

Устранение лестничного эффекта (antialiasing) с использованием алгоритмов первой группы связано с изменением интенсивности граничных пикселей.

В обычной ситуации условия корректного вывода линий и других графических объектов можно сформулировать так: если в контур объекта попадает больше половины площади ячейки сетки растра, то соответствующий пиксель закрашивается цветом объекта (например, черным с интенсивностью Iо=Iч), иначе – пиксель сохраняет цвет фона (например, белый с интенсивностью Iф=Iб).



На рисунке на растровое изображение толстой прямой линии наложен идеальный контур исходной линии.

Чтобы растровое изображение линии выглядело более гладким, можно интенсивность (яркость) каждой цветовой компоненты граничных пикселей заменить на некоторую интенсивность, промежуточную между интенсивностью объекта и фона. Наиболее корректно вычислять интенсивность пропорционально площади ячейки растра, покрываемой идеальным контуром объекта.

Если площадь всей ячейки обозначить как S, а часть площади, покрываемой контуром - Sx, то искомая интенсивность равна:

Ix = Iф + (Io - Iф) ·Sx/S,

или с учетом того, что площадь ячейки S=1 (ее длина и высота равны 1 пикселю),

Ix = Iф + (Io - Iф) ·Sx.



Площадь части ячейки растра, лежащей ниже пересекающей линии равна Sx = Площадь прямоугольника + Площадь треугольника, то есть Sx = 1·m + 1·k/2=m + tg(α)/2 = d1 + Δy/(2·Δx),

так как tg(α) = Δy/Δx (треугольник подобен большому треугольнику со сторонами Δx, Δy – см. алгоритм Брезенхама). Поэтому окончательно: Ix = Iф + (Io - Iф) · [d1 + Δy/(2·Δx)].



На рисунке показано как будет скорректировано растровое изображение с помощью приведенного выше способа

Другую группу методов устранения лестничного эффекта составляют алгоритмы сглаживания уже сформированного изображения, обычно методом цифровой фильтрации. Локальная цифровая фильтрация выполняется методом взвешенного суммирования яркостей пикселей, расположенных в некоторой окрестности текущего обрабатываемого пикселя.

В ходе действия алгоритма обработки, по растру скользит квадратное окно (размерностью, как правило, 3х3 пикселя, в его центре – текущий пиксель), которое объединяет информацию о пикселях, окружающих текущий пиксель, и на основе этой информации меняется яркость текущего пикселя.

Тогда новая интенсивность текущего пикселя, расположенного на i-й строке и j-ом столбце, будет такой (для матрицы 3х3)



где K – нормирующий коэффициент, M – маска фильтра (матрица коэффициентов 3х3), Imn – интенсивности пикселей.

# *26. Опишите алгоритмы заполнения области.*

Задача заключается в закрашивании произвольного контура, нарисованного в растре. Задается область, ограниченная набором пикселей заданного цвета, и точка (x,y), лежащая внутри этой области.

***Простейший алгоритм заполнения*** заключается в следующем. Цвет текущего пикселя изменяется на нужный цвет заполнения. Затем проводится анализ цветов всех соседних пикселей. Если цвет некоторого соседнего пикселя не равен цвету границы контура или цвету заполнения, то цвет этого пикселя изменяется на цвет заполнения.

Затем анализируется цвет пикселей, соседних с предыдущими. И так далее, до тех пор, пока внутри контура все пиксели не перекрасятся в цвет заполнения. Соседними могут считаться только четыре пикселя (четырехсвязность) или все восемь пикселей (восьмисвязность).

Алгоритм абсолютно корректен, но неэффективен, так как рекурсивная функция вызывается для каждого пикселя (и не обязательно только один раз), что медленно и требует большого размера стека. Более предпочтительными являются алгоритмы, обрабатывающие сразу целые группы пикселей.

**Алгоритм закрашивания линиями** отличается тем, что на каждом шаге закрашивания рисуется горизонтальная линия, которая размещается между пикселями контура. По сравнению с предыдущим рекурсивным алгоритмом здесь количество вложенных вызовов уменьшается пропорционально длине линии.

**Волновой алгоритм** закрашивания основан на поиске трассы на графе. Здесь для начальной точки (вершины на графе) находятся соседние точки (другие вершины), которые отвечают двум условиям: эти вершины связаны с начальной, и они рассматриваются впервые.

Соседние вершины текущей итерации становятся текущими точками для поиска новых соседних вершин в следующей итерации. По достижении конечной точки (границы области) совершается обратный цикл к начальной точке, в ходе которого находятся все кратчайшие пути между двумя заданными точками на графе и определяется порядок закраски пикселей.

# *27. Дайте характеристику основным этапам отображения 3-х мерной информации.*

Процесс отображения 3-х мерной информации имеет свои, присущие лишь ему этапы:

1.Определение координат наблюдателя (по отношению к объекту).

2.Определение зоны визуализации (куда попадают объекты).

3.Построение изображения в оперативной памяти (формирование базы данных математической модели сцены).

4. Проецирование (переход 3D -> 2D).

5.Удаление невидимых линий и поверхностей.

6.Затенение и закраска (переход от каркасного к твердотельному изображению).

7.Визуализация (вывод объекта на экран).

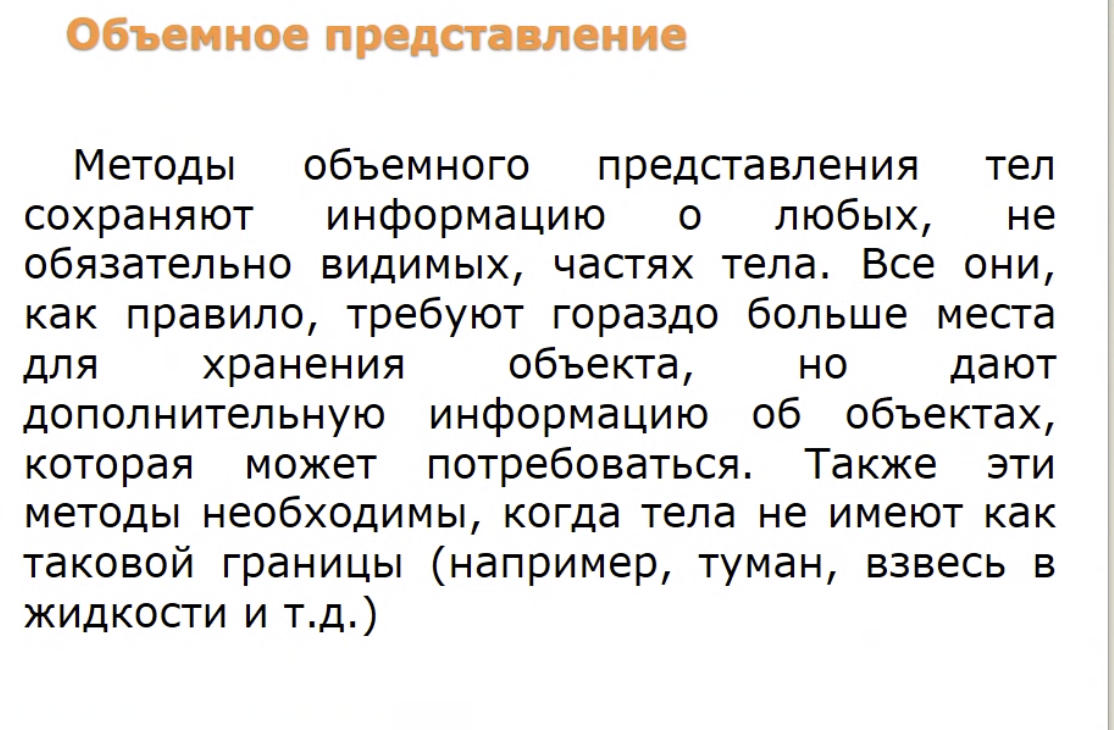
Отдельные этапы процесса могут следовать в ином порядке. Многие системы отображения информации используют синтез изображения в реальном времени, где один из основных показателей – производительность, то есть количество графических примитивов, обрабатываемых за время формирования кадра.

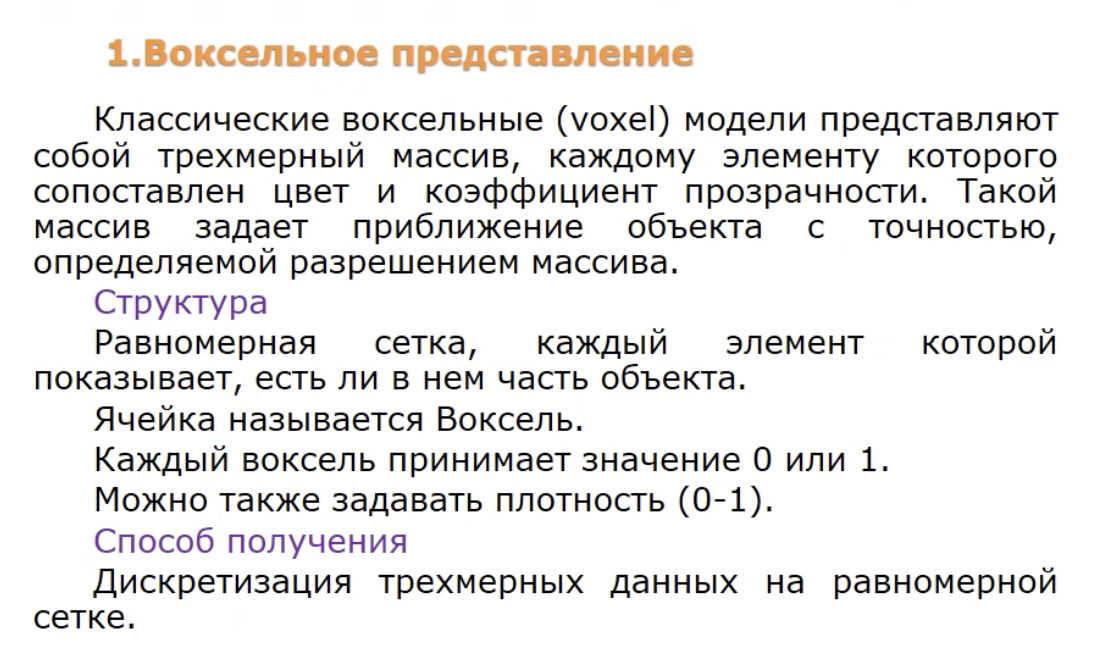
Непосредственному синтезу изображения во многих системах КГ (компьютерной графики) предшествует разработка базы данных модели сцены.

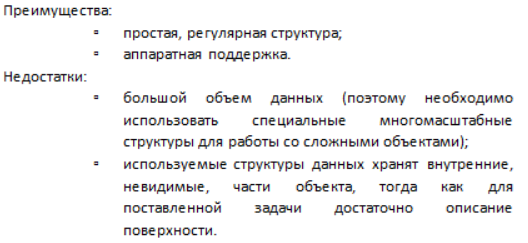
Наиболее сложные этапы при синтезе изображения – удаление невидимых линий, закраска и затенение с учетом освещенности сцены. Например, для создания реалистических изображений при определении освещенности следует учитывать, что данный участок поверхности освещается светом, отраженным от соседних участков.

# *28. Методы объемного представления. Воксельное представление.*

**Из лекции:**







**Из интернета:**

**Объемное представление**

Методы объемного представления тел сохраняют информацию о любых, не обязательно видимых, частях тела. Все они, как правило, требуют гораздо больше места для хранения объекта, но дают дополнительную информацию об объектах, которая может потребоваться. Также эти методы необходимы, когда тела не имеют как таковой границы (например, туман, взвесь в жидкости и т.д.).

**Воксельное представление**

Воксел – это трехмерный аналог пиксела, от слов pixel («пиксель») и volume («объем»).

Воксельная модель — это представление объектов в виде трехмерного массива объемных (кубических) элементов. Каждый воксел может иметь некоторое числовое значение, являющееся атрибутом (цвет, прозрачность и т. п.) соответствующей точки в пространстве.

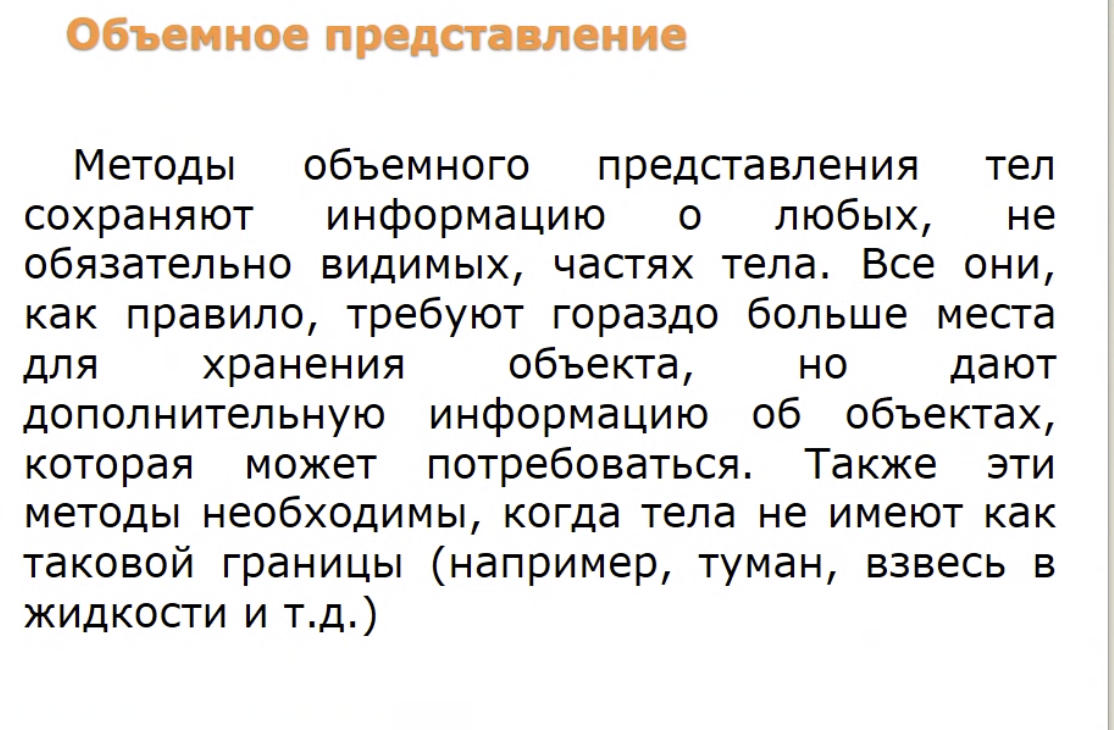
Воксельное представление позволяет приблизительно описать объемное тело совершенно произвольной формы.

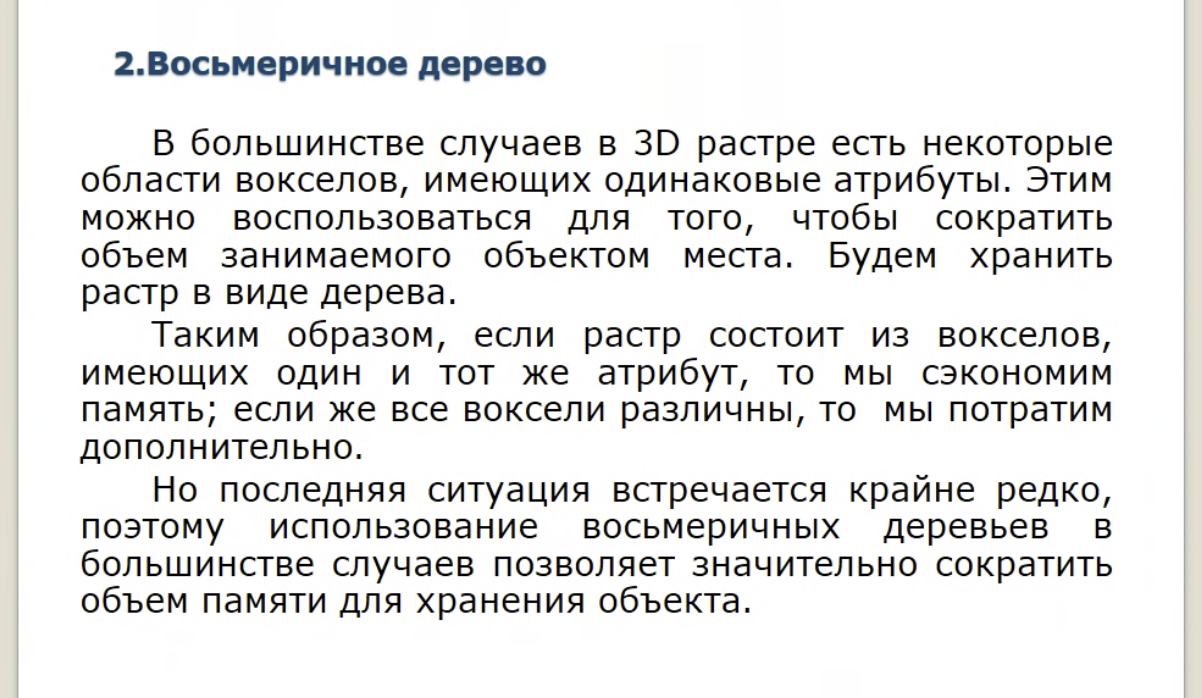
Хотя воксельное представление предназначено для описания  
объемного тела в пространстве, оно автоматически описывает и  
пространство вне тела. Поэтому оно удобно для расчета объема полых  
структур.

Данный метод, несмотря на свою простоту, имеет серьезный недостаток: для хранения даже небольшого объекта в приемлемом качестве требуется очень много места. Объем памяти, требуемый для хранения воксельного представления тела, резко возрастает с уменьшением размеров вокселов. Размер вокселов определяет точность приближения исходного тела**.**

# *29. Методы объемного представления. Восьмеричное дерево.*

**Из лекции:**





**Из интернета:**

**Объемное представление**

Методы объемного представления тел сохраняют информацию о любых, не обязательно видимых, частях тела. Все они, как правило, требуют гораздо больше места для хранения объекта, но дают дополнительную информацию об объектах, которая может потребоваться. Также эти методы необходимы, когда тела не имеют как таковой границы (например, туман, взвесь в жидкости и т.д.).

**Восьмеричное дерево**

Одним из способов улучшения воксельного представления является представление воксела в памяти в виде октодерева (восьм. дерева).

Пространство разбито на восемь октант, которые представляются деревом (упорядоченным). Начало октодерева располагается вокруг начала его локальной системы координат. Октанты первого уровня совпадают с октантами системы координат. Каждый узел октантного дерева соответствует некоторому кубу в трехмерном пространстве, который является либо:

- полностью (с заданной точностью) принадлежащим описываемому телу;

- полностью непринадлежащим описываемому телу;

- частично пересекающимся с описываемым телом.

Первые два типа узлов - терминальные (листья в октантном дереве), а каждый узел третьего типа обязательно имеет 8 дочерних узлов, соответствующих геометрическому разбиению его куба на 8 частей (октантов)

# *30. Опишите основные операции, применяемые для создания моделей в Компас 3D.*

Новое тело в Компас можно создать с помощью одной из следующих операций:

* Выдавливание. Образует тело путем перемещения сечения вдоль прямолинейной траектории на заданное расстояние.
* Вращение. Образует тело путем поворота сечения вокруг оси на заданный угол.
* По сечениям. Образует тело путем соединения нескольких сечений.
* Кинематическая. Образует тело путем перемещения сечения вдоль произвольной траектории.
* Деталь-заготовка. Вставляет в модель тело, существующее в другой модели.
* Придание толщины. Образует тело путем добавления слоя материала на указанную поверхность.
* Листовое тело. Образует тело путем выдавливания эскиза в направлении, перпендикулярном его плоскости.
* Сшивка поверхностей. Образует тело, ограниченное указанными поверхностями.

После создания нового тела производится «приклеивание» к нему или «вырезание» из него дополнительных объемов, т.е. добавление или удаление материала тела.

# *31. Какие типы моделей можно создавать в системе КОМПАС 3D? Из каких объектов состоит трехмерная модель в КОМПАС 3D?*

**Типы моделей** в системе КОМПАС 3D: каркасные (проволочные), поверхностные и твердотельные.

* Конструктивными элементами **каркасной** модели являются ребра и вершины. Основное преимущество каркасных моделей — простота. При использовании таких моделей возможны различные интерпретации одной модели, поскольку известны только ребра и вершины. В современных системах геометрического моделирования каркасные модели используются при отображении конструируемых объектов как один из методов визуализации.
* **Поверхностная** модель кроме вершин и ребер содержит грани (прямоугольные или треугольные), необходимые для аппроксимации поверхностей. Возможны различные виды задания поверхностей (плоскости, поверхности вращения, линейчатые поверхности). Используются различные математические модели аппроксимации поверхностей (методы Кунса, Безье, В-сплайны).
* **Твердотельные** модели состоят из последовательного выполнения операций объединения, вычитания и пересечения над простыми объемными элементами (призмы, цилиндры, пирамиды и т.д.), из которых и состоит большинство механических деталей. Многократно выполняя эти простые операции над различными объемными элементами, можно построить сложную модель.

**Модель в КОМПАС-3D состоит из геометрических объектов** — эскизов, пространственных кривых, поверхностей, тел.

* **Эскиз** может располагаться в одной из ортогональных плоскостей координат, на плоской грани существующего тела или во вспомогательной плоскости.
* **Пространственные кривые** − это те линии, у которых точки не лежат в одной плоскости.
* **Поверхность** – это объект, который представляется значениями высоты Z, распределенными по области, определенной координатами X и Y.
* **Тело** – область трехмерного пространства, состоящая из однородного материала и ограниченная замкнутой поверхностью, которая сформирована из одной или нескольких стыкующихся граней.

# *32. Метод выдавливания. Какие требования предъявляются к эскизу элемента выдавливания?*

Элемент выдавливания образуется путем перемещения сечения по прямолинейной направляющей в одну или в обе стороны на заданное расстояние. Например, элемент, показанный на рисунке, образован выдавливанием эскиза в направлении, перпендикулярном его плоскости.

Элемент выдавливания может быть самостоятельным телом, а может быть приклеен к телу или вырезан из него.

Для создания нового тела выдавливания или приклеивания элемента выдавливания к имеющемуся телу (т.е. для добавления материала) служит операция **Выдавливание**, а для вырезания элемента выдавливания из тела (т.е. для удаления материала) — операция **Вырезать выдавливанием**.

В качестве сечения элемента выдавливания может использоваться грань, эскиз, ребро или пространственная кривая (в том числе контур, построенный по линиям эскиза, ребрам грани или произвольный).

При выдавливании грани, замкнутого эскиза, замкнутого Контура на грани или Контура эскиза возможен выбор между сплошным и тонкостенным элементом. При разомкнутом сечении возможно построение только тонкостенного элемента.

Если сечение представляет собой плоскую грань, эскиз, контур, построенный по линиям эскиза или плоской грани, и выдавливается в направлении, перпендикулярном себе, то возможен уклон боковых граней элемента.

**Требования к эскизу элемента выдавливания**

1. Объекты эскиза могут составлять один или несколько контуров.
2. Если контуров несколько, они должны быть либо все замкнуты, либо все разомкнуты.
3. Если контуры замкнуты, они могут быть вложенными друг в друга. Уровень вложенности не ограничивается.
4. Если сечением элемента является эскиз или плоская грань, то этот объект автоматически выбирается в качестве направляющего.
5. Если сечением является контур, построенный по эскизу или по ребрам плоской грани, то в качестве направляющего объекта автоматически выбирается соответствующий эскиз или грань.

При необходимости направляющий объект можно сменить. Направляющими объектами для выдавливания могут быть:

* Существующий плоский или прямолинейный объект.
* Прямолинейный объект задает направление, параллельное себе.
* Плоский объект задает направление, перпендикулярное своей плоскости.
* Вектор.

# *33. Из чего состоят геометрические объекты модели в КОМПАС 3D?*

Геометрические объекты состоят из примитивов — вершин, ребер, граней.

**Грань** – гладкая (не обязательно плоская) часть поверхности детали, ограниченная замкнутым контуром из ребер. Частный случай – шарообразные твердые тела и тела вращения с гладким профилем, состоящие из единой грани, которая, соответственно, не имеет ребер.

**Ребро** – пространственная кривая произвольной конфигурации, полученная на пересечении двух граней.

**Вершина** – точка в трехмерном пространстве. Для твердого тела это может быть одна из точек на конце ребра.

# *34. Что такое эскиз? Опишите принципы работы с эскизами.*

Эскиз — объект трехмерного моделирования, созданный средствами чертежно-графического редактора. Эскизы используются для разных целей, например:

• задание формы сечения тела или поверхности,

• задание траектории перемещения сечения,

• задание положения экземпляров массива.

Эскиз может располагаться на координатной или вспомогательной плоскости, а также на плоской грани.

Эскиз может содержать одну или несколько цепочек объектов — контуров эскиза. Контуры могут быть замкнуты или разомкнуты. Они могут пересекаться друг с другом. Если при этом образуются замкнутые области, то эти области можно указывать в качестве исходных элементов при различных построениях. Самопересечение контуров не допускается.

Один и тот же эскиз может использоваться в нескольких различных операциях. Для выполнения некоторых операций можно указать как эскиз целиком, так и замкнутую область в эскизе, ограниченную контуром или несколькими пересекающимися контурами.

Работа с эскизом ведется в специальном режиме работы с моделью — режиме эскиза.

По умолчанию каждый новый эскиз создается параметрическим (в системе включен па­раметрический режим).

При работе в параметрическом режиме на объекты эскиза автоматически накладывают­ся параметрические связи и ограничения.

Вы можете построить непараметрический эскиз, отключив параметрический режим. Однако такой стиль работы не рекомендуется.

Если в модели уже имеются объекты (грани, ребра, вершины и др.), то их можно исполь­зовать при построении эскиза. Вы можете спроецировать объекты в эскиз или выпол­нить привязку к объектам. В последнем случае проекции объектов в эскизе создаются автоматически.

# *35. Метод вращения. Какие требования предъявляются к эскизу элемента вращения?*

Любая формообразующая операция базируется на эскизе. К эскизу есть определенные требования, которые зависят от типа операции. Для операции вращения справедливо:

* объекты, которые должны участвовать в операции, должны быть выполнены основной и осевой линиями. Контур выполняется основной линией, осевая линия обозначает ось вращения. Все вспомогательные линии выполняются любым другим стилем, в том числе и утолщенной линией;
* если эскиз содержит ось, то она не должна пересекать контур. Крайние точки контура могут лежать на оси, либо её продолжении;
* один замкнутый контур может быть вложен в другой замкнутый контур, степень вложенности любая.

## *36. Дайте характеристику понятиям «базовая точка», «дерево построения», «компоненты сборки».*

**Базовая точка** - это точка, по положениям которой до и после операции сдвига, копирования, поворота или деформации определяется перемещение объектов, участвующих в операции или точка, от которой отсчитываются углы и расстояния для определения нового положения объектов при копировании вдоль кривой, по сетке и по концентрической сетке.

**Дерево построения** - при работе с любой деталью или сборкой на экране может отображаться окно, содержащее Дерево построения модели.

Дерево построения — это представленная в графическом виде последовательность объектов, составляющих модель.

В Дереве построения отображаются:

* обозначение начала координат
* плоскости
* оси
* пространственные кривые
* поверхности
* условные обозначения
* эскизы
* операции

Компонентами называются входящие в сборку детали, подсборки и стандартные изделия.

Компоненты записаны в отдельных файлах на диске. В сборке хранятся ссылки на эти компоненты.

Пользователь может указать взаимное положение компонентов сборки, задав параметрические связи между их гранями, ребрами и вершинами (например, совпадение граней двух деталей или соосность втулки и отверстия). Эти параметрические связи называются сопряжениями.

В сборке можно выполнить формообразующие операции, имитирующие обработку изделия в сборе (например, создать отверстие, проходящее через все компоненты сборки и отсечь часть сборки плоскостью).

## *37. Опишите особенности построения сборок в Компас 3D. Что такое сопряжение?*

После того, как в сборке будут созданы компоненты, можно приступать к созданию па­раметрических связей между ними.

Сопряжение – это параметрическая связь между гранями, ребрами или вершинами раз­ных компонентов сборки.

В KOMПAC-3D можно задать сопряжения следующих типов:

* Совпадение
* Касание
* Соосность
* Параллельность
* Перпендикулярность
* Расположение элементов на заданном расстоянии
* Расположение элементов под заданным углом

## *38. Дайте характеристику понятиям «ассоциативный чертеж», «спецификация».*

**Ассоциативный чертеж** – это вид чертежа, ассоциативно связанный с определенной 3D-моделью. При изменении формы или размеров модели автоматически изменяется изображение на всех связанных с ней ассоциативных видах.

**Спецификация** — документ, содержащий текстовую информацию о составе сборки, представленную в виде таблицы. Спецификация оформляется рамкой и основной надписью. Файл спецификации имеет расширение sрw.

В ручном режиме все графы спецификации заполняются с клавиатуры.

В полуавтоматическом режиме ряд граф заполняется благодаря указанию источника данных — чертежа или модели.

Спецификация может быть составлена на основе готового сборочного чертежа. При вставке в чертеж или модель стандартных изделий (болтов, шайб, гаек т.д.) из конструкторской библиотеки, их обозначение формируется и вносится в спецификацию автоматически.

Осуществляется двунаправленная ассоциативная связь между спецификацией и соответствующими ей документами. Благодаря наличию этой связи изменения в сборочном чертеже автоматически отражаются в спецификации. Изменения в спецификации могут автоматически передаваться в сборочный чертеж.

Объект спецификации — строка или несколько следующих друг за другом строк спецификации, относящихся к одному материальному объекту.

Объекты спецификации бывают базовые и вспомогательные.

Для базовых объектов предусмотрена возможность автоматического заполнения колонок, сортировки внутри раздела, подключения графических объектов из сборочного чертежа.