Теоретический минимум к курсу “Компьютерная графика” (2017)

{Презентация №1}

1. Дайте определение цифровому цветному изображению.

Цифровое цветное изображение $X \in R^{n \times m \times c}$ — трёхмерный массив, в котором для каждого пиксела с координатами (x, y) хранится описание цвета, где:

- n, m — высота и ширина изображения
- c — число каналов (длина вектора, описывающего цвет пиксела)
- Обычно каждая компонента x_{ijk} дискретизируется до $[0,255]$ в 8 бит

2. Дайте определение понятию цвет.

Цвет — психологическое свойство нашего зрения, возникающее при наблюдении объектов и света (а не физические свойства объектов и света).

Цвет — это результат взаимодействия света, сцены и нашей зрительной системы.
Видимый цвет — это результат взаимодействия спектра излучаемого света и поверхности.

3. Сформулируйте закон аддитивности Грассмана.

Эмпирический закон о линейности человеческого зрения. Верно:

1. Если наблюдатель задаст цвет лучей 1 и 2 как (R_1, G_1, B_1) и (R_2, G_2, B_2), тогда, если мы сложим источники 1 и 2, то мы можем воспроизвести их как $(R_1 + R_2, G_1 + G_2, B_1 + B_2)$
2. Соответствие цветов выполняется на всех уровнях яркости. Т. е., если $C_1 = (R_1, G_1, B_1)$, то $kC_1 = (kR_1, kG_1, kB_1)$

{Презентация №2}

4. Опишите принцип работы алгоритма “серый мир”.

- Средний уровень («серый») по каждому каналу должен быть одинаков для всех каналов
- Если цветовой баланс нарушен, тогда «серый» в этом канале больше «серого» других каналов
- Вычислим коэффициенты усиления так, чтобы среднее в каждом канале стало одинаковым:

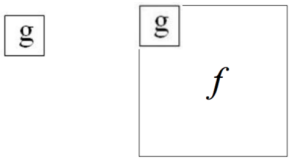
$$\overline{R} = \frac{1}{N}\sum R(x,y);\ \overline{G} = \frac{1}{N}\sum G(x,y);\ \overline{B} = \frac{1}{N}\sum B(x,y);\ Avg = \frac{\overline{R} + \overline{G} + \overline{B}}{3};$$

$$R' = R \cdot \frac{Avg}{\overline{R}};\ G' = G \cdot \frac{Avg}{\overline{G}};\ B' = B \cdot \frac{Avg}{\overline{B}};$$

5. Опишите принцип работы операции свёртки.

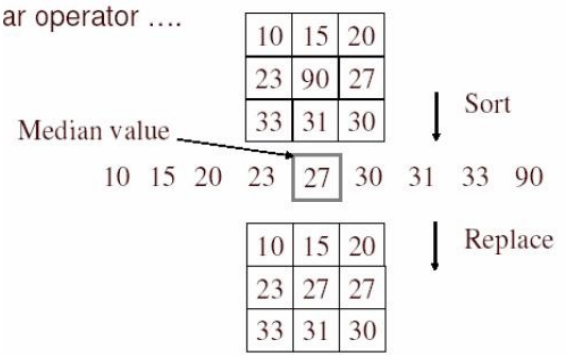
Пусть f — изображение, g — ядро. Свёртка изображения f с помощью g обозначается как $f * g$ и определяется как:

$$(f * g)[m,n] = \sum_{k,l} f\left[m-k, n-l\right] g\left[k,l\right]$$



6. Опишите принцип работы медианного фильтра.

Выбор медианы из выборки пикселей по окрестности данного.



7. Перечислите основные этапы алгоритма поиска границ Canny

1. Свёртка изображения с ядром — производной от фильтра Гаусса
2. Поиск силы и направления градиента
3. Выделение локальных максимумов (non-maximum suppression) — утончение полос в несколько пикселей до одного пикселя
4. Двойной порог (гистерезис)
   - Выше верхнего порога — «сильные края»
   - Ниже нижнего порога — шум (отсекаем)
   - Посередине — потенциальные края
5. Связывание краёв

8. Сформулируйте и опишите формулами метод минимизации эмпирического риска

- Дано: $X^m = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ — обучающая выборка;
- Эмпирический риск (ошибка обучения):

$$R_{emp}\left(f, X^m\right) = \frac{1}{m}\sum_{i=1}^{m} L\left(f\left(x_i\right), y_i\right)$$

- Метод минимизации эмпирического риска:

$$f = \arg\min_{f \in F} R_{emp}\left(f, X^m\right)$$

- Смысл: подбираем параметры t решающего правила f таким образом, чтобы эмпирический риск R_{emp} был минимален.

9. Сформулируйте алгоритм кластеризации k-средних.

Дано:
- Набор векторов $x_i, i = \overline{1,p}$;
- k — число кластеров, на которые нужно разбить набор x_i;

Найти:
- k средних векторов $m_j, j = \overline{1,k}$ (центров кластеров)
- Отнести каждый из векторов x_i к одному из k кластеров;

Алгоритм:
1. Случайным образом выбрать k средних $m_j, j = \overline{1,k}$;
2. Для каждого $x_i, i = \overline{1,p}$ подсчитать расстояние до каждого из $m_j, j = \overline{1,k}$, отнести (приписать) x_i к кластеру j', расстояние до центра которого m_j минимально;
3. Пересчитать средние $m_j, j = \overline{1,k}$ по всем кластерам;
4. Повторять шаги 2, 3 пока кластеры не перестанут изменяться

10. Укажите основные этапы вычисления признаков для классификации изображений

Пример для гистограммы градиентов:

1. Разбить изображение на области сеткой
2. В каждой сетке посчитать гистограмму распределения пикселов по ориентации градиентов
3. Вычислить градиент в каждой точке

{Презентация №4}

11. Сформулируйте метод скользящего окна, критерий обнаружения объектов IoU, верных и ложноположительных обнаружений по этому критерию.

Метод скользящего окна — просмотр изображения по фрагментам определённого размера и применение к ним классификатора

Критерий обнаружения объектов IoU (Intersection over Union) —

$$\text{Score} = \frac{\text{Area of overlap}}{\text{Area of union}}$$

- Обнаружение, если $\text{IoU} > p$ (например, $p = 0.5$)

Верные и ложноположительные обнаружения — все обнаружения оцениваются через IoU:

- $\text{IoU} > p \Rightarrow$ True positive
- $\text{IoU} < p \Rightarrow$ False positive

12. Дайте определение понятиям точности, полноты, кривой точности-полноты и F-меры.

- Точность (Precision) — доля истинных объектов основного класса среди всех классифицированных, как основной класс.

- Полнота (Recall) — доля правильно распознанных объектов основного класса среди всех объектов основного класса из тестовой выборки

$$\text{Precision} = \frac{\text{Number of true detections}}{\text{Number of detections}};\ \text{Recall} = \frac{\text{Number of true detections}}{\text{Number of objects}}$$

- Кривая точности-полноты — зависимость между точностью и полнотой в зависимости от параметров

- F-мера / Интегральный показатель F / F-measure:

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

13. Перечислите основные идеи алгоритма Виола-Джонса

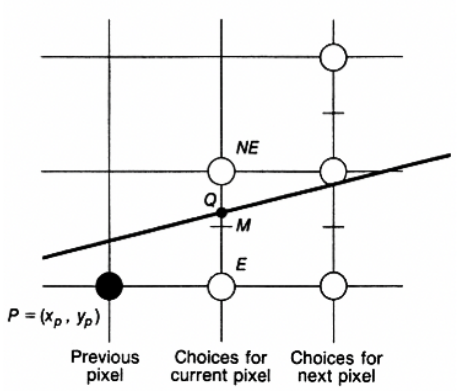
- Основополагающий метод для поиска объектов на изображении в реальном времени

- Обучение очень медленное, но поиск очень быстр

- Основные идеи:
  - *Признаки Хоара [прямоугольные фильтры]* в качестве *слабых классификаторов*
  - *Интегральные изображения* для быстрого вычисления признаков
    - Значение каждого пиксела равно сумме значений всех пикселов левее и выше данного включительно
    - Рассчитывается за один проход
  - *Бустинг* для выбора признаков
  - *Каскад (Attentional cascade)* для быстрой отбраковки окон без лица
    - Несколько классификаторов один за другим

14. Сформулируйте алгоритм Брезенхема для прямых.

- Задача: построить отрезок $P\left(x_1, y_1\right), Q\left(x_2, y_2\right)$
- Уравнение прямой: $y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}\left(x - x_1\right), x \in \left[x_1, x_2\right]$
- Рассматриваем направление «вправо-вверх», угол меньше 45° $\Rightarrow 0 \leqslant y_2 - y_1 \leqslant x_2 - x_1$
- Идея алгоритма: инкрементный выбор y-координаты следующей точки:
  - Знаем положение пикселя $P\left(x_p, y_p\right)$
  - Для пикселя $x_p + 1$ есть варианты выбора y: $E\left(y_p\right)$ либо $NE\left(y_p + 1\right)$
  - Выбираем нужный пиксель в зависимости от положения точки M относительно Q
  - Повторяем до конца отрезка



- Функция F — индюкатор положения точки прямой относительно середины
  - Определим функцию F:
    - $F\left(x, y\right) = \left(x - x_1\right) dy - \left(y - y_1\right) dx$, где $dx = x_2 - x_1$, $dy = y_2 - y_1$
    - Свойства F: $\begin{cases} F\left(x, y\right) = 0 \Rightarrow \text{Точка на отрезке} \\ F\left(x, y\right) < 0 \Rightarrow \text{Точка выше отрезка} \\ F\left(x, y\right) > 0 \Rightarrow \text{Точка ниже отрезка} \end{cases}$
- Вычисляем значение d: $M = \left(x_p + 1,\ y_p + \frac{1}{2}\right);\ d = F\left(M\right)$
- Если $d < 0 \Rightarrow$ Выбираем E. Тогда $d_{new} = d + \Delta E$, где $\Delta E = dy$
- Если $d \geqslant 0 \Rightarrow$ Выбираем NE. Тогда $d_{new} = d + \Delta NE$, где $\Delta NE = dy - dx$
- $d_{start} = F\left(x_1 + 1, y_1 + \frac{1}{2}\right) = \ldots = dy - \frac{dx}{2}$

15. Дайте определение сплайну Безье и перечислите его свойства.

Сплайн, задаваемый следующим образом:
- Степень кривой равна $N - 1$
- Базис — полиномы Бернштейна:
$$b_{v,n}(x) = \binom{n}{v} x^v (1-x)^{n-v},\ v = \overline{0,n},\ {}^nC_i = \binom{n}{i} = \frac{n!}{i!(n-i)!}$$
- Позволяет аппроксимировать функции

Свойства:
- Проходит через первую и последнюю точку
- Касательные в первой и последней точке проходит вдоль линии, соединяющей крайнюю и ближайшую контрольную точку
- Кривая полностью лежит внутри выпуклой оболочки контрольных точек

16. Сформулируйте алгоритм Де Кастельжо для визуализации сплайнов Безье.

- Задана кривая Безье с опорными точками $P_0, \ldots, P_n$. Соединив их последовательно, получаем ломаную линию;
- Разделяем каждый полученный отрезок этой ломаной в соотношении $t : (1 - t)$ и соединяем полученные точки. В результате получаем ломаную с количеством отрезков, меньшим на один, чем исходная ломаная линия.
- Повторяем процесс до тех пор, пока не получим единственную точку. Эта точка и будет являться точкой на заданной кривой Безье с параметром t.

{Где-то здесь материал перестаёт наличествовать в презентациях и начинает браться из других шпор, википедии, левой пятки автора и т. д.}

17. Перечислите основные этапы графического конвейера.

1. Получение данных
2. Обработка вершин (вершинный шейдер)
3. Обработка примитивов
4. Растеризация
5. Обработка пикселей (пиксельный шейдер)
6. Слияние (merger)
7. Вывод результата

18. Опишите отличия модельной, мировой, видовой системы координат.

- Модельная система координат — СК, определённая таким образом, чтобы было удобно определять и ориентировать один объект
- Мировая (правосторонняя) система координат — СК, в которой располагаются все модели
- Видовая (левосторонняя) система координат — система с камерой в центре, смотрящей по оси Z

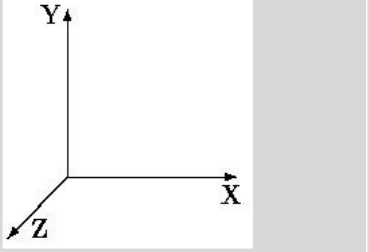


Рис. 5.1. Правосторонняя система координат (мировая)

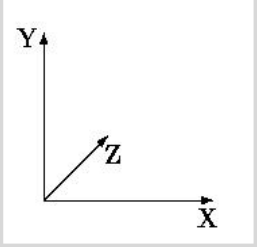


Рис. 5.2. Левосторонняя система координат (видовая)

19. Дайте определение октодерева и опишите алгоритм его построения из воксельной модели.

Октодерево — иерархический вариант воксельной модели, рекурсивное разбиение пространства на восемь октант.

Обозначения ветви дерева:
- B(lack) — заполнено
- W(hite) — пусто
- G(græy) — частично (рекурсивно разбито на 8 подоктант)

Построение:
- Делим модель на 8 октант, смотрим каждую из них
  - Если октанта полностью заполнена, то узел B
  - Если полностью пустая, то W
  - Иначе G и рекурсивно повторить алгоритм для этой октанты

20. Дайте определение понятию конструктивной геометрии.

- Технология моделирования, использующая как основу набор базовых примитивов (сфера, куб, цилиндр и т. д.) и операций по их комбинированию (объединение, пересечение, разность)

21. Дайте определение индексированному по вершинам представлению.

- Способ граничного представления со следующими свойствами:

  - Хранит индексы вершин
  - Выделение координат вершин в отдельную структуру
  - С гранями ассоциируются не координаты вершин, а индексы в массиве координат

22. Опишите различие между первичными и вторичными источниками света.

- Первичный источник света — тело, испускающее свет самостоятельно
- Вторичный источник света — тело, отражающее падающий на него свет

23. Дайте определение ДФО (двулучевой функции отражения)?

ДФО — отношение яркости поверхности к её освещённости;
  - Яркость поверхности — количество света, излучаемого ей
  - Освещённости поверхности — количество света, падающего на неё

24. Напишите формулу модели освещения Фонга, поясните обозначенные переменные.

$L_0 = k_a L_a + L_i \left(k_d \left(\vec{s} \cdot \overrightarrow{n} \right) + k_s \left(\vec{r} \cdot \overrightarrow{v} \right)^{k_e} \right)$, где

- L_0 — отражение данной поверхности
- L_i — излучение i-го источника света
- k_a, k_s, k_d — коэффициенты фонового, зеркального и диффузного освещения соответственно;
- L_a — фоновое излучение
- $\overrightarrow{n}$ — вектор нормали к поверхности в точке
- $\vec{s}$ — направление на источник света
- $\overrightarrow{v}$ — направление на наблюдателя
- $\vec{r}$ — направление отражения луча от поверхности
- k_e — коэффициент блеска (свойство материала)

25. Дайте определение понятия шейдер.

Шейдер — компьютерная программа, предназначенная для исполнения процессорами видеокарты.

26. Опишите отличие VBO от VAO.

VBO (Vertex Buffer Objects) — технология хранения координат вершин с их атрибутами в видеопамяти
VAO (Vertex Arrays Object) — технология OpenGL, определяющая, какую часть VBO надо использовать в последующих командах и как эти данные интерпретировать.

27. Укажите, в какой момент и кем/чем компилируются шейдеры.

В OpenGL шейдеры компилируются во время исполнения программы.

28. Дайте определение обратной трассировки лучей, перечислите её плюсы и минусы.

Обратная трассировка лучей — метод построения изображения, основанный на отслеживании пути каждого луча, попавшего в глаз наблюдателя или объектив камеры в направлении, противоположном распространению света

29. Перечислите основные шаги рекурсивной трассировки лучей.

1. Создать трассируемый луч
2. Найти пересечение луча с поверхностью
3. Вычислить цвет пикселя (возможно, используя соседние лучи)

30. Опишите идею регулярной сетки как ускоряющей структуры в трассировке лучей.

Начинаем с описывающего параллелепипеда, треугольники разбиваются по вокселям. При трассировке лучей луч проходит только через часть вокселей. В данных вокселях проверяется пересечение луча с объектами, которые расположены в них (Алгоритм Брезенхема в 3D, останов — найдено пересечение в текущем вокселе). Простой алгоритм, быстро строится, выгоден для небольших сцен.

31. Опишите механизм работы буфера глубины (Z буфер).

Буфер глубины — двумерный массив, каждый элемент которого соответствует пикселю на экране и равен расстоянию вдоль направления проецирования от картинной плоскости до соответствующей точки пространства.

32. Дайте определение альфа-канала и поясните его назначение.

Альфа-канал — дополнительный канал, который может быть добавлен к пикселю. Содержит значение от 0 до 1 и кодирует информацию о прозрачности пикселя (0 — полностью прозрачный, 1 — полностью непрозрачный).

33. Опишите схему работы метода Монте-Карло для вычисления объема (или площади в 2D) сферы.

Пусть у нас есть какая-нибудь фигура на плоскости, площадь которой нам необходимо найти. Ограничим её другой фигурой, площадь которой мы можем легко вычислить. Например, прямоугольником со сторонами, параллельными координатным осям. И пусть про любую точку прямоугольника мы можем быстро узнать, попадает эта точка внутрь фигуры, площадь которой мы ищем, или нет. Тогда будем бросать на плоскость точки случайным образом в пределах ограничивающей фигуры, причем координаты x и y должны быть распределены равномерно по соответствующим отрезкам. Будем считать количество точек, попавших в фигуру, и общее количество брошенных точек. Поделив количество точек, попавших в фигуру, на общее количество, мы можем найти, какую часть прямоугольника занимает фигура. Количество брошенных точек зависит от точности, которую мы хотим получить (чем больше точек, тем точнее).

- $f(x)$ — функция, описывающая фигуру, площадь которой необходимо вычислить
- u — случайная величина, равномерно распределённая на $[a, b]$
- $p(u)$ — плотность вероятности u
- Тогда $f(u)$ — тоже случайная величина:
  - $\mathbb{E}f(u) = \int_a^b f(x)\, p(x)\, dx;$
  - $p(x) = \frac{1}{b-a}$
  - $\int_a^b f(x)\, dx = (b-a)\, \mathbb{E}f(u) \approx \frac{b-a}{N} \sum f(u_i)$