

3D компьютерное зрение и графика

Лекция 1. Введение. 3D-представления

Жемчужников Дмитрий Сергеевич

Команда курса

Лекции 1–5

Введение и 3D-представления

Геометрия: LA → SO(3)

Камера и калибровка

Многовидовая геометрия

Matching и SfM

Лекции 6–10

Растеризация, графический конвейер

Трассировка лучей

Освещение

Дифференцируемый рендеринг I–II

Лекции 11–15

Оценка глубины

Многовидовая реконструкция

Семантическая реконструкция

Advanced camera

SLAM


Дмитрий Жемчужников


Сослан Кабисов


Олег Павлов


Максим Елистратов


Владимир Фролов


Альберт Гарифуллин


Максим Колодяжный


Никита Дроздов

Факультет искусственного интеллекта
МГУ имени М.В. Ломоносова

Команда курса



Дмитрий Жемчужников

PhD университета Гренобль-Альпы, научный сотрудник Центра ИИ МГУ и AIRI, руководитель команды по CAD реконструкции. 1-5 лекции



Олег Павлов

Аспирант Физического Факультета МГУ (практически к.н.). Раньше занимался ИИ для биофизики. Сейчас разрабатывает сети для облаков точек. 2 семинар.



Сослан Кабисов

Магистр МФТИ, долго работал в продовом CV, сотрудник Сбера (по факту AIRI), главный разработчик методов image2cad. 4, 5 семинары



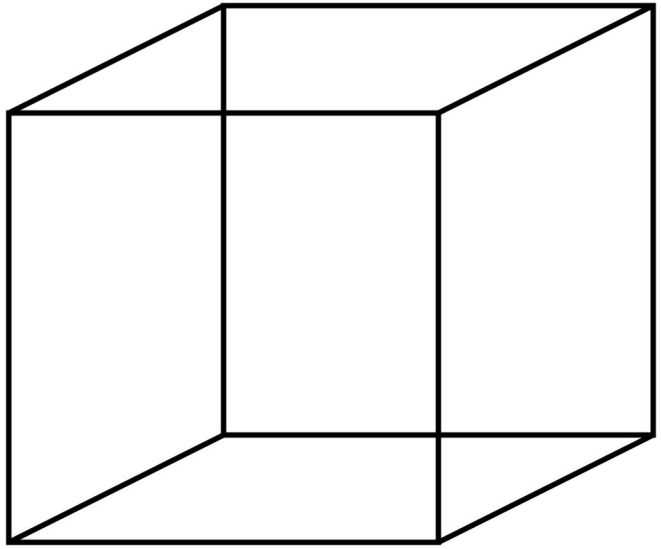
Максим Елистратов

Аспирант Факультета ИИ МГУ (1 год). Имеет опыт работы инженера в авиастроении. Автор эволюционного пайплайна для генерации данных. 1, 3 семинары.

Как устроен курс

- 15 лекций и 15 семинаров, по одному семинару к каждой лекции
- По 2 домашних задания на каждый блок, всего 6 дз
- Устный экзамен

Как мы воспринимаем 3Д пространство



BenFrantzDale, CC BY-SA 3.0, Wikimedia Commons

Одна картинка — два прочтения





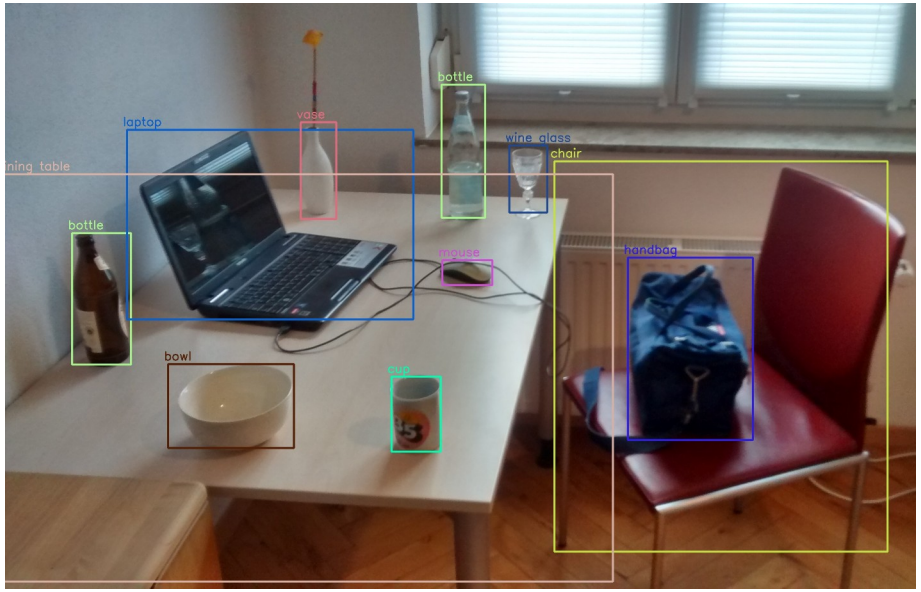
Skagedal, public domain, Wikimedia Commons · маска Бьорна Борга, опыты в Уппсальском университете

Маска вогнутая — но мозг видит выпуклое лицо

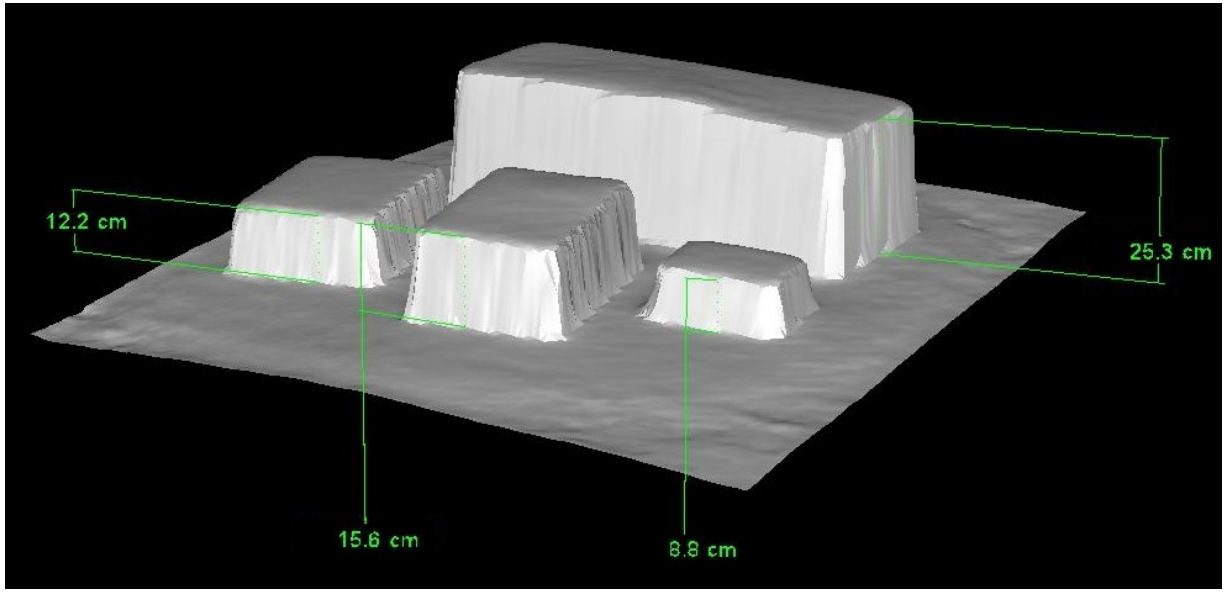
- На сетчатке — двумерная проекция; трёхмерная сцена это реконструкция, а не измерение
- Мозг доопределяет её приорами: два глаза, движение, затенение, перспектива, перекрытия

Где приор не выполняется — там иллюзия: и у мозга, и у алгоритма

Вопросы компьютерного зрения



MTheiler, CC BY-SA 4.0, Wikimedia Commons · детекция YOLOv3 (COCO)



Captaindistance, CC BY 3.0, Wikimedia Commons · времяпролётная камера

«Что это?» «Где на картинке?»

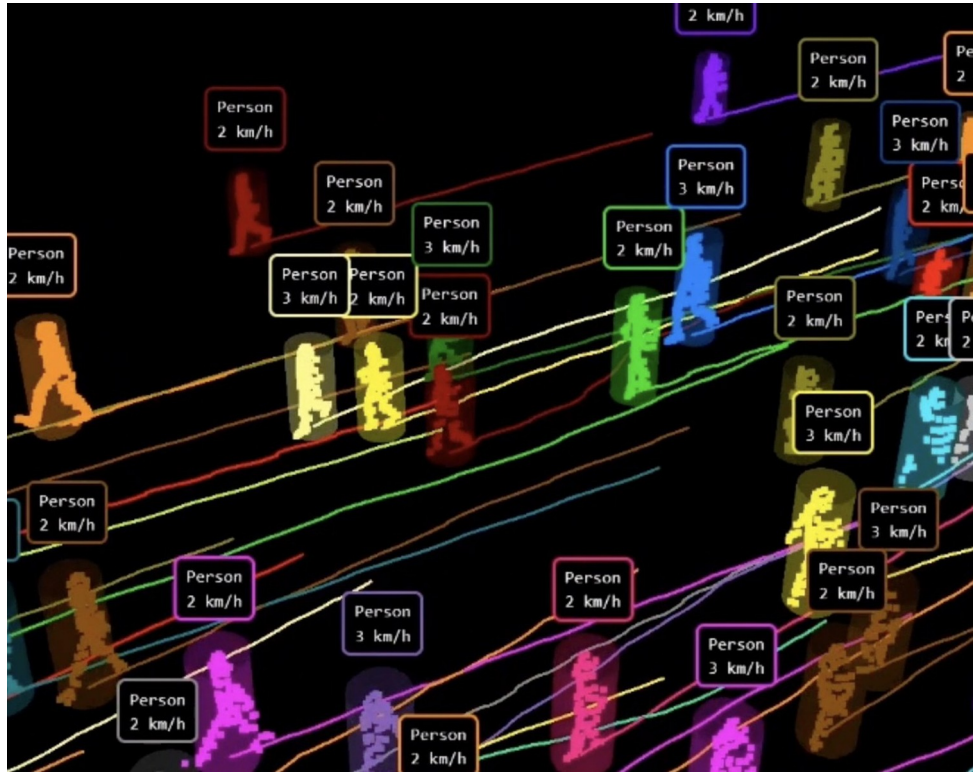
- класс: cup, laptop, bowl, chair
- рамка на картинке — ответ в пикселях
- это параллельный курс CV

«Где в пространстве?» «Какого размера?»

- положение и расстояние — ответ в метрах
- размеры 12,2 и 25,3 см — сверяет рулетка
- это наш курс

«Где» спрашивают оба курса — разница в том, в пикселях ответ или в метрах

Почему одного семейства мало



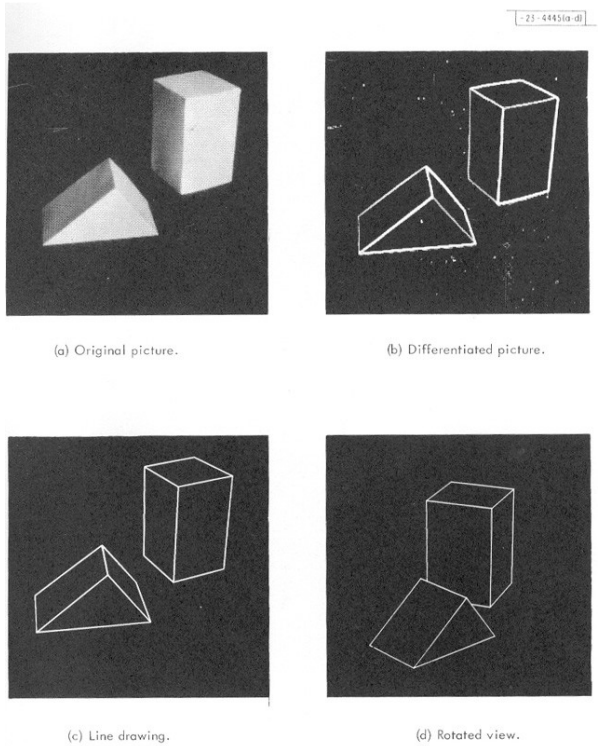
Saxotango, CC BY-SA 4.0, Wikimedia Commons · лидарный мониторинг людей

- Роботу метка «кружка» бесполезна: нужно, где именно и какого она размера
- Автомобилю мало «впереди пешеход»: нужны расстояние, скорость, габариты
- Шлему AR нужна геометрия комнаты — иначе виртуальный объект не перекроется мебелью
- На картинке у каждого объекта есть и класс, и скорость, и траектория — это и есть объединение двух семейств

3D-зрение = метрическое зрение + семантическое зрение

Не «здесь машина», а «машина длиной 4,5 м в семи метрах впереди, движется под углом 15°»

Зрение начиналось с трёх измерений



L. G. Roberts. Machine Perception of Three-Dimensional Solids. PhD thesis, MIT, 1963

Три уровня Марра, 1982

1. Первичный эскиз

края, штрихи, пятна на изображении

2. 2½-мерный эскиз

глубина и ориентация видимых поверхностей в системе координат наблюдателя

3. Трёхмерная модель

форма объекта в его собственных координатах

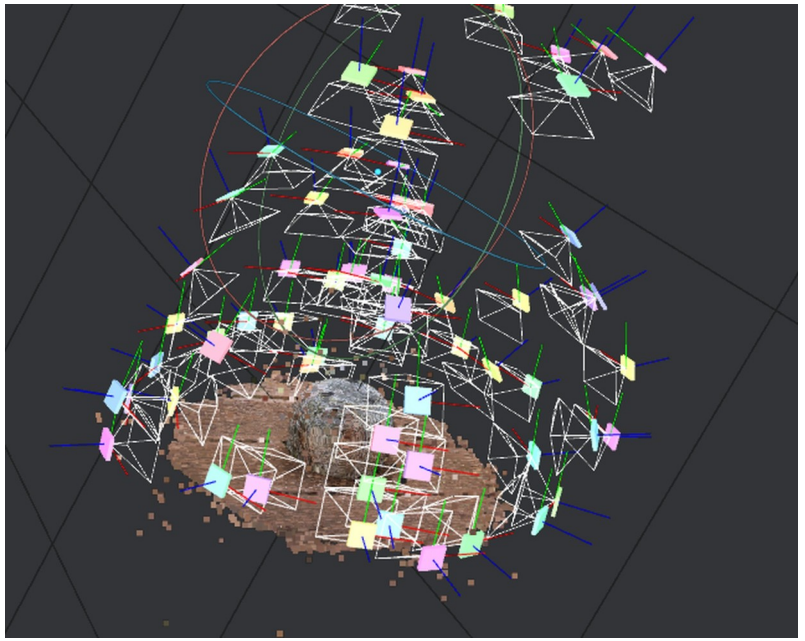
По фотографии — рёбра, по рёбрам — 3D-модель

Плоское зрение — более поздняя специализация; к 2020-м маятник качнулся обратно

Прямая и обратная задача



KaiaVintr, CC BY-SA 4.0, Wikimedia Commons · Blender Cycles, path tracing



Colin Kranz, CC BY-SA 4.0, Wikimedia Commons · фотограмметрия в Meshroom

Прямая задача: сцена → изображение

- задали форму, материалы, свет и камеру
- физика известна — картинка получается одна
- это графика: лекции 6–8

Обратная задача: изображения → сцена

- по снимкам восстановлены и форма, и позы камер
- решений бесконечно много — нужен дополнительный источник знания
- это наш курс: лекции 4–5, 12

Прямая и обратная задача: схема

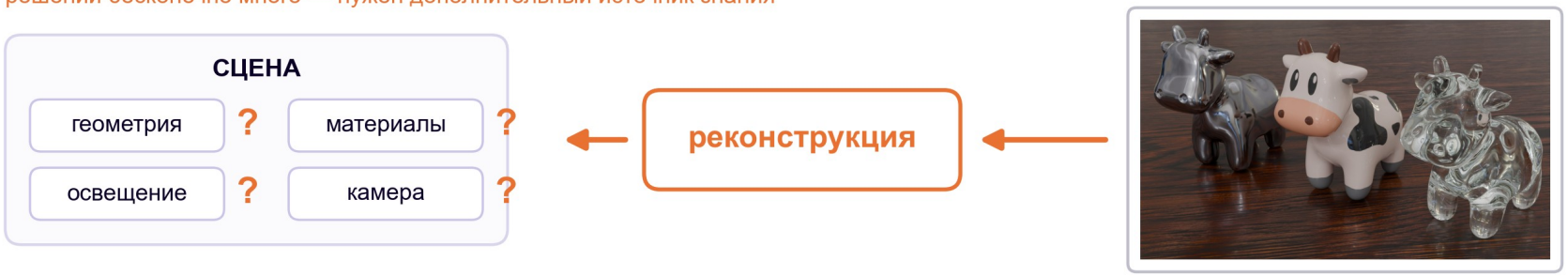
Прямая задача: сцена → изображение

физика известна — ответ единственный



Обратная задача: изображение → сцена

решений бесконечно много — нужен дополнительный источник знания



рендер: KaiaVintr, CC BY-SA 4.0, Wikimedia Commons

Неизвестны все четыре группы сразу: уравнений меньше, чем неизвестных

задача не «трудная», а некорректно поставленная: выбрать один ответ можно только по знанию, которого в изображении нет

Корректность по Адамару



настоящая трещина в леднике



краска на ровном асфальте, снято с нужной точки

Существует — да

объяснений не ноль, а слишком много

Единственно — нет

одному изображению отвечает бесконечно много сцен

Устойчиво — не всегда

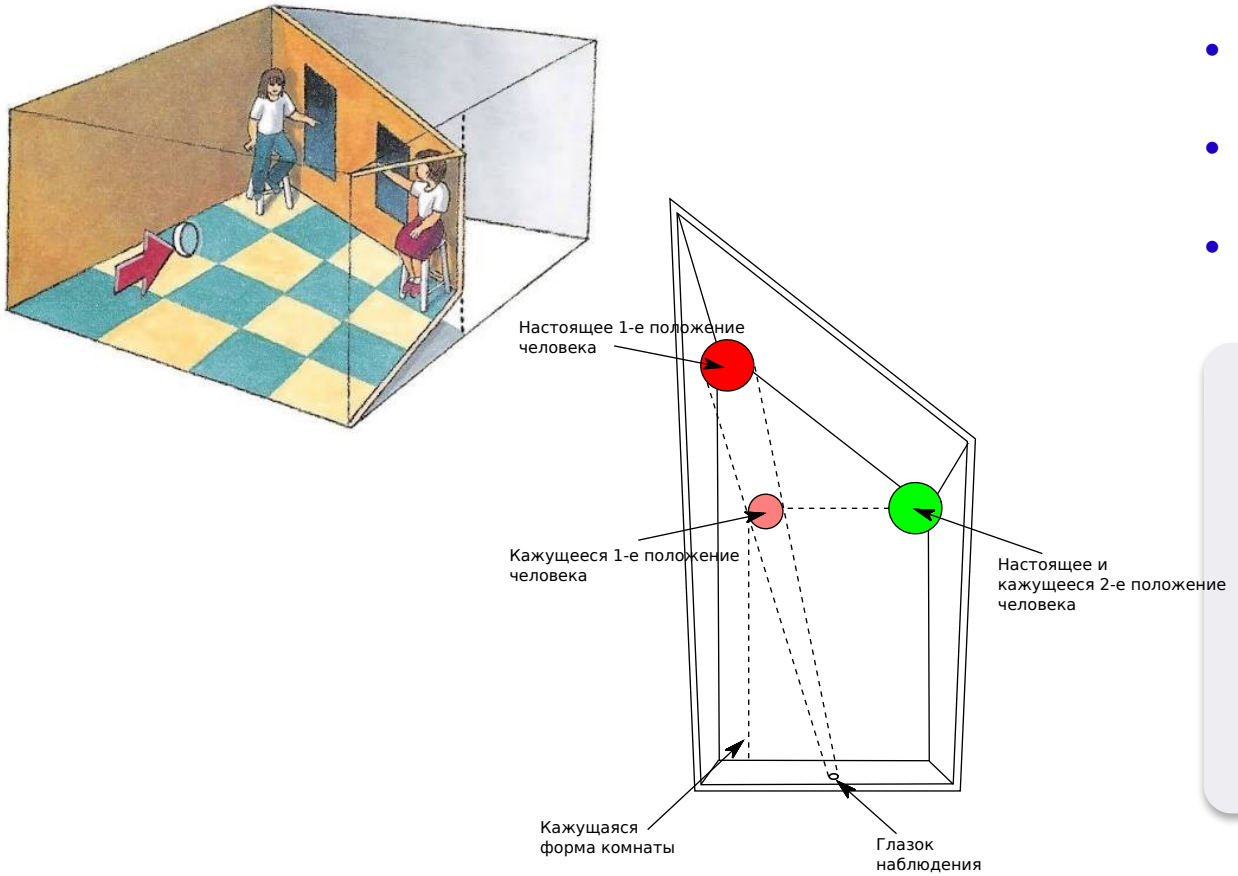
малый шум в пикселях двигает оценку

Это свойство самой задачи, а не нехватка вычислений

Вопрос при чтении статьи: чем именно этот метод доопределяет задачу?

трещина: Leonhard Lenz, CC0 · роспись на асфальте: Metanamorph, CC BY-SA 4.0

Неоднозначность 1: глубина вдоль луча



Kirill Borisenko, CC BY-SA 4.0, Wikimedia Commons · схема комнаты Эймса

- Пиксель фиксирует направление, но не расстояние
- Все точки одного луча дают один и тот же пиксель
- Маленькая кружка близко и большая далеко — одинаковая проекция

Комната Эймса

Проецируется так же, как обычная прямоугольная

Дальний угол вдвое дальше — но кажется, что на том же расстоянии

Мозг выбирает решение «комната прямоугольная» — и человек «меняет» рост

Снимается вторым видом, активным сенсором или приором

Неоднозначность 2: масштаб



Buiobuione, CC BY-SA 4.0, Wikimedia Commons · Тадж-Махал

- Увеличим сцену и базу в s раз — картинки не изменятся
- Важно отношение размера к расстоянию, а оно то же

Семейство решений — группа подобия

3 поворота + 3 сдвига + 1 масштаб = 7 степеней свободы

В лекции 5 это будет называться gauge freedom

Чем снимается

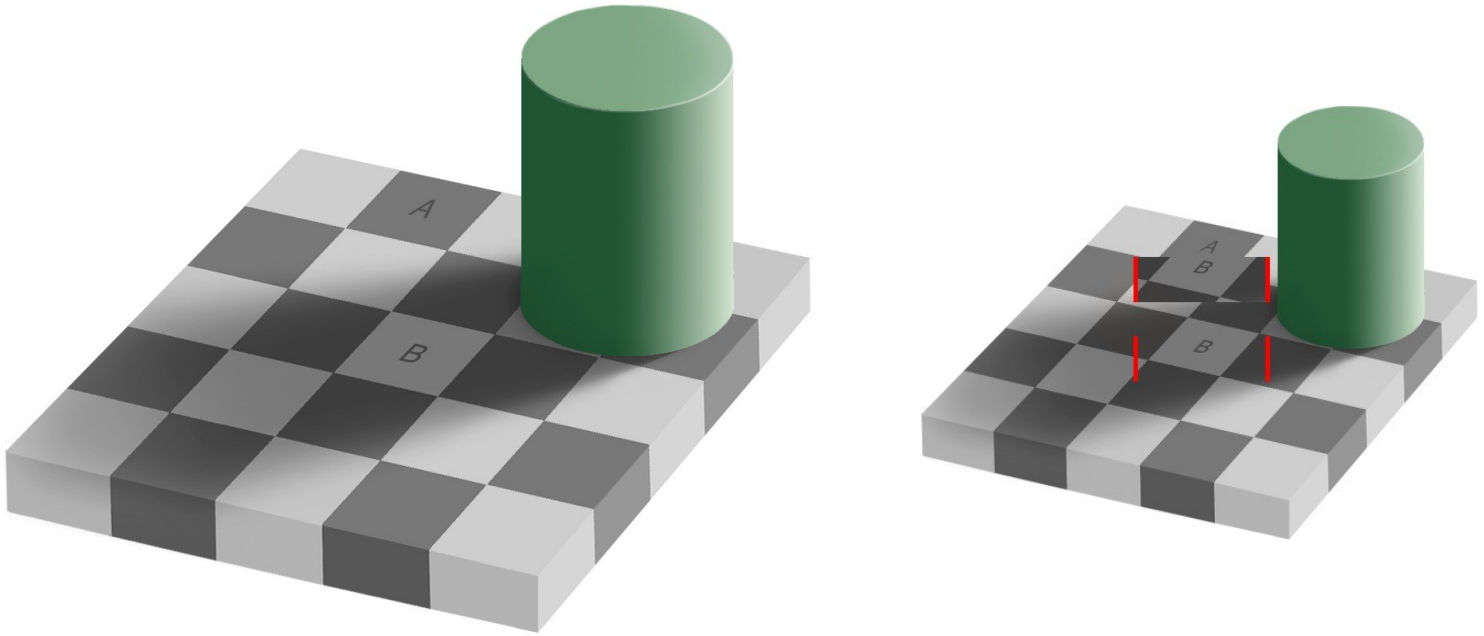
известный размер объекта в кадре

база стереопары

инерциальный датчик

Масштаб приходит только извне — из самих снимков его не достать

Неоднозначность 3: материал против освещения



оригинал Edward H. Adelson, файл Gustavb, Wikimedia Commons · клетки A и B одного оттенка

яркость = альбедо × освещённость

- одно число — два неизвестных
- тёмная поверхность при ярком свете и светлая при тусклом дают одно значение
- разделение — inverse rendering, лекция 8

Бас-рельефная неоднозначность · Belhumeur, Kriegman, Yuille, 1999

При ламбертовой модели и ортографической проекции сжатие рельефа вдоль оси взгляда вместе с преобразованием освещения даёт неотличимые изображения

По затенению форма восстанавливается лишь с точностью до этого семейства

А разве нейросети не восстанавливают по одному кадру?

Да, выдают. Но знание приходит из весов, а не из изображения





тот же механизм отказа, что у мозга: маска вогнутая, а приор «лица выпуклые» побеждает измерение

Где отлично работает

визуализация, черновая модель, ассет — когда нужен правдоподобный результат

Где ломается

обратная сторона выдумана — свидетелей нет

масштаб угадан: сменится объектив — поедет оценка

вне распределения ошибается уверенно

Правдоподобный ответ ≠ измеренный ответ

нужны метры с известной точностью — нужны второй вид, известная база или калиброванный сенсор

Три источника доопределения



1. Больше наблюдений

второй кадр даёт второй луч —
пересечение определяет точку

плата: нужны позы камер и соответствия

лекции 4, 5, 12



2. Физические приоры

гладкость, ламбертовость,
манхэттенский мир

плата: мир может быть не таким

лекция 3



3. Выученные приоры

статистика мира,
извлечённая из данных

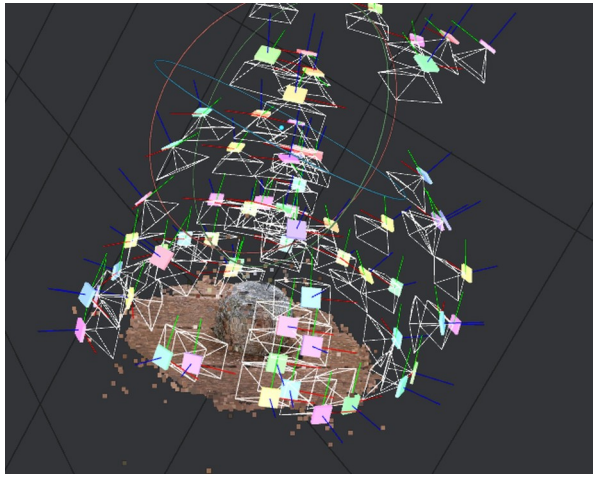
плата: зависит от обучающей выборки

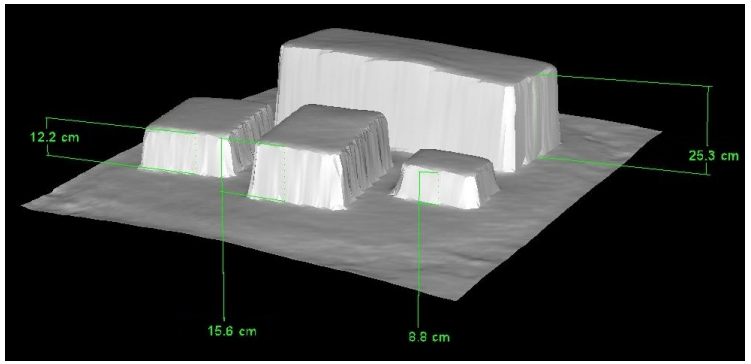
лекции 9–11, 14

Ошибки у трёх источников разные — и это подсказывает, кто подвёл

эпиполярные линии: .gs8, CC BY-SA 3.0 · Киото: Basile Morin, CC BY-SA 4.0 · монокулярная глубина: Kritik Soman, CC BY-SA 4.0

На чём стоит выученный приор







Многовидовая реконструкция
SfM и плотное стерео по обычным фотографиям

Активный сенсор
RGB-D, лидар: расстояние измерено

Синтетика
сцену задали числами — геометрия известна точно

DUSt3R — 8 наборов, ≈8,5 млн пар · Depth Anything — 1,5 млн размеченных и 62 млн с псевдоразметкой

Выученный приор — это упакованная статистика ранее выполненных измерений

сеть выучивает те же признаки глубины, что и классическое зрение

Современная форма: инверсия рендерера



NeRF: сцена — поле плотности и цвета



гауссовы сплаты: сцена — миллионы гауссиан

1. Рендерим гипотезу

прямая задача: по параметрам сцены получаем изображение

2. Сравниваем со снимками

невязка по всем кадрам сразу

3. Двигаем параметры

градиент невязки по параметрам сцены, и так до совпадения

Одна схема — разные представления сцены и разные рендереры

Прямую задачу разбираем в лекциях 6–8, обращаем её — в лекциях 9–12

NeRF: Jurdein, CC BY-SA 4.0 · гауссовы сплаты: Óscar Mirás, CC BY-SA 4.0

Что именно оптимизируем

$$\theta^* = \arg\min_{\theta} \sum_{k} \left\| \mathcal{R}(\theta, \pi_k) - I_k \right\|^2 + \lambda \mathcal{P}(\theta)$$

$\mathcal{R}$ дифференцируемый рендерер — прямая задача, лекции 6–8

I_k наблюдаемое изображение — то, что сняла k-я камера

$\mathcal{P}$ регуляризатор — тот самый приор

θ — параметры сцены, π — параметры камеры

Представление сцены обязано быть дифференцируемым

- иначе оно просто не встраивается в эту схему
- дифференцируемость — одна из осей, по которым мы сравниваем представления

Генерация: наблюдений нет вообще



ни одной из этих машин не существует: проход по латентному пространству генеративной модели формы

Реконструкция

$$\left\| \mathcal{R}(\theta, \pi_k) - I_k \right\|^2$$

невязка между рендером и реальным снимком

Генерация

$$D_\varphi(\mathcal{R}(\theta, \pi_k))$$

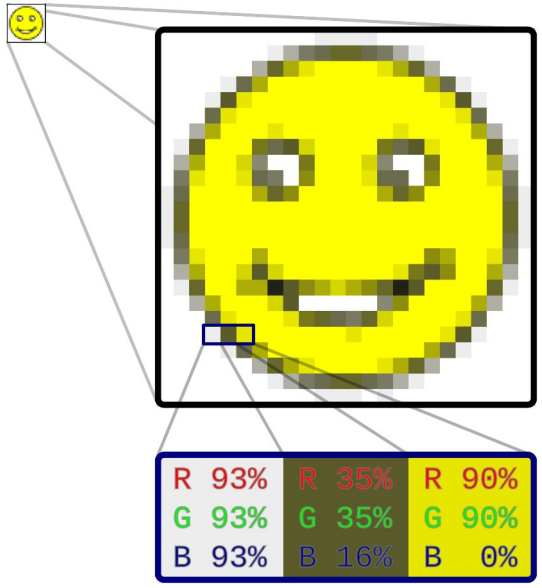
снимков нет: рендер оценивает обученная 2D-модель (DreamFusion, 2022 — score distillation)

Наблюдений ноль — всю работу делает приор: это третий источник в чистом виде

без дифференцируемости некуда девать градиент, без свободы топологии не отрастить ручку

морфинг форм: Joël Jean Henry, CC BY-SA 4.0 · DreamFusion — Poole et al., 2022 (схема — лекции 9–10)

Почему представлений много



растр W×H×C: один формат
на сенсор, алгоритм и дисплей

В 2D повезло: представление одно

свёртка определена на решётке — отсюда все сети для изображений

В 3D такого представления нет — и, похоже, не будет

- точка внутри или снаружи? — одна структура
- где луч встречает поверхность? — другая
- как плавно деформировать? — третья
- как учить нейросеть? — четвёртая

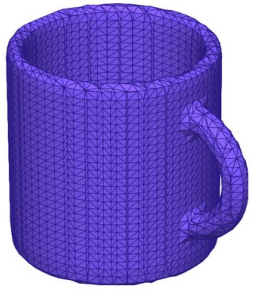
Представление — это структура данных плюс операции, которые она делает дешёвыми

Выбор представления — это выбор того, что будет дёшево, а что дорого

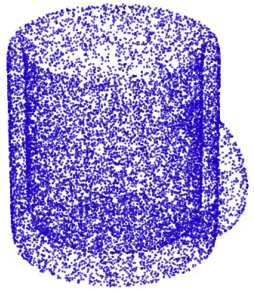
растр: Gringer, CC0

Факультет искусственного интеллекта
МГУ имени М.В. Ломоносова

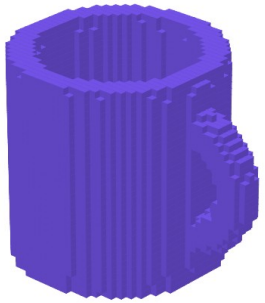
Одна кружка — пять представлений



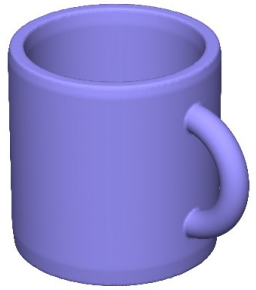
полигональная сетка
вершины и грани



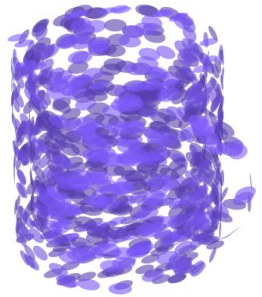
облако точек
множество без связей



воксели
регулярная 3D-сетка



неявная функция
поверхность = { f = 0 }



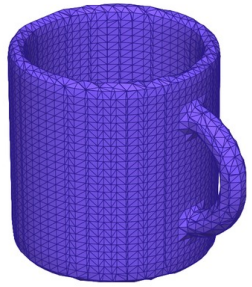
гауссовы сплаты
примитивы с объёмом

Разбираем каждое: что оно делает дешёвым и во что это обходится

и дифференцируемо ли оно — ось, которая появилась из схемы с рендерером

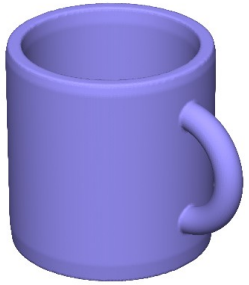
Две главные дихотомии

- поверхностные или объёмные — только граница тела или всё пространство
- явные или неявные — перечисляем точки или задаём функцию, чей нулевой уровень даёт поверхность



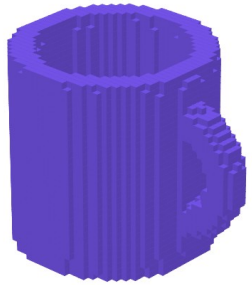
явно-поверхностные

облака точек, полигональные сетки, параметрические поверхности



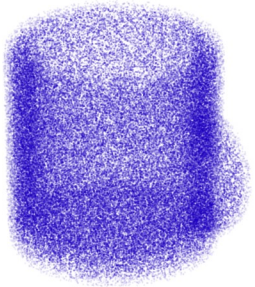
неявно-поверхностные

SDF, occupancy — поверхность как нулевой уровень



объёмные дискретные

воксели, октодеревья, TSDF, MPI



объёмные непрерывные

поля плотности, нейронные поля

Объёмное описывает поверхность, обратное — нет: поэтому идём от поверхности к объёму

оси классификации: Tewari et al., STAR 2022, §3.1

Сигнал и структура данных — разные вещи

$$\Phi(\mathbf{x};\Theta)$$

поле, определённое во всех точках пространства, плюс параметры, которые где-то хранятся

Сигналы: что определено в точке

SDF — расстояние со знаком

occupancy — внутри или снаружи

плотность, радиантность

глубина

Структуры: где живут параметры

воксельная сетка

октодерево, хэш-таблица

веса MLP

набор примитивов

Одна и та же occupancy:

в вокселях — запрос O(1), память $O(n^3)$; в MLP — память равна числу весов, запрос — полный проход сети

«Нейронное поле против вокселей» — это сравнение структур, а не сигналов

спрашивайте отдельно: какой сигнал и в чём он хранится

рамка «поле + параметры»: Xie et al., 2022

Остальные оси: чек-лист

ось	что спрашиваем
лагранжево или эйлерово	информация едет вместе с точками или лежит в узлах сетки → динамические сцены
зависимость от ракурса	карта глубины видит одну сторону; сетка и неявная функция — объект целиком
память по разрешению	поверхность растёт как n^2, объём — как n^3
топологическая свобода	может ли меняться число ручек в ходе оптимизации: у сеток нет, у неявных да
стоимость операций	внутри/снаружи · пересечение с лучом · ближайшая точка · сэмплирование · деформация · нормаль
дифференцируемость	есть ли градиент потерь по параметрам — без этого не встроить в схему с рендерером
пригодность для сетей	свёртки на регулярных, инвариантность к перестановкам на множествах, графы на сетках
редактируемость	возьмёт ли это в руки художник или инженер

Универсального представления нет — есть удачный выбор под задачу

оси из Botsch et al., Polygon Mesh Processing, гл. 1 и Tewari et al., STAR 2022

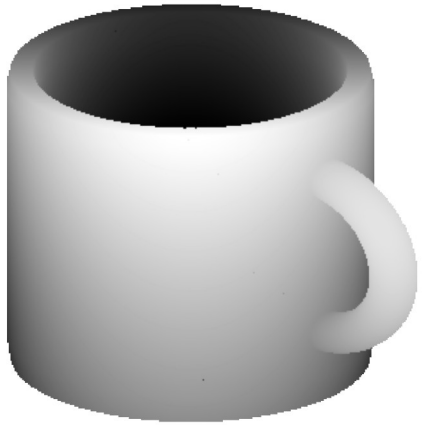
Карта глубины: вместо цвета — расстояние



объект

как его видит камера

карта глубины



светлее — ближе к камере

карта нормалей



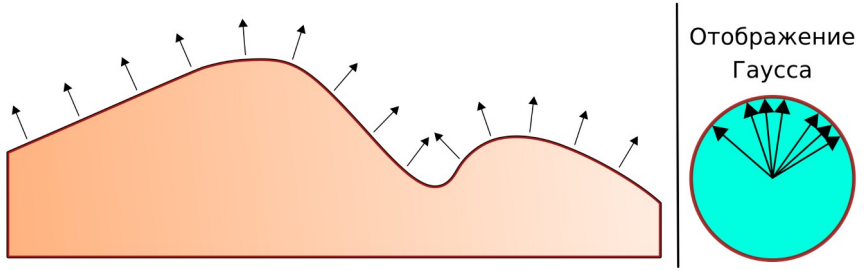
RGB = (n + 1) / 2 в координатах камеры

- вместе с цветом — RGB-D
- работают все инструменты обработки изображений, включая свёртки (лекция 11)

Нормаль — точка на сфере S^2, а не три числа

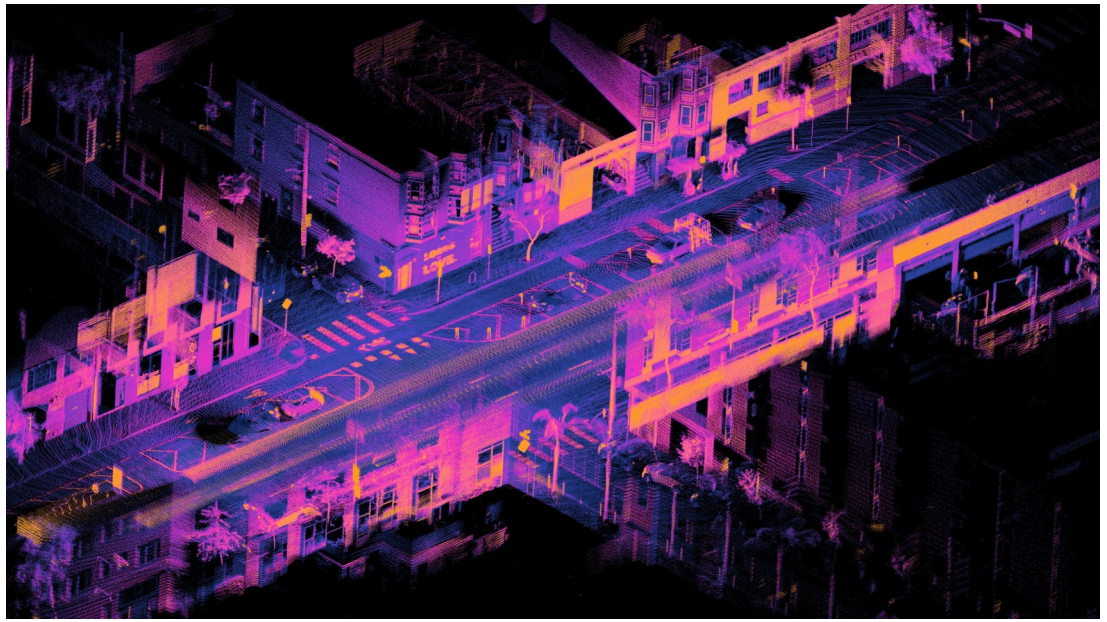
все нормали собираются на единичной сфере — отображение Гаусса

у сети единичность надо требовать явно; в лекции 2 то же с вращениями



отображение Гаусса: Erik Streb, CC BY-SA 3.0

Облако точек: то, что выдают сенсоры



лидар, времяпролётные камеры, структурированный свет,
а также стерео и SfM после обратной проекции

Порядок точек не несёт смысла

облако — множество, а не массив: ответ не должен меняться при перестановке точек

PointNet (Qi et al., 2017): симметричная агрегация — max-pooling по точкам

Связности нет

какие точки соседние — неизвестно; kd-дерево или радиусный поиск строятся заново на каждом шаге

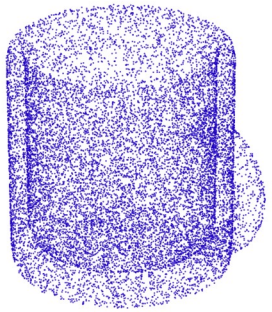
дёшево получить и хранить: это сырой выход сенсора, без обработки

дорого соседство: kd-дерево или радиусный поиск строятся заново

Отсортировать точки не выход: малое смещение меняет порядок скачком

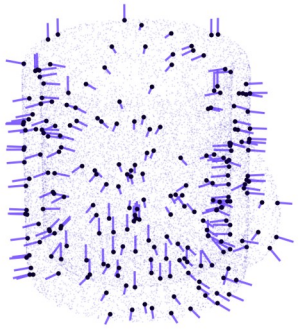
лидарное облако: Daniel L. Lu, CC BY 4.0 (Ouster OS1-64, Сан-Франциско)

Ориентированное облако: точка плюс нормаль



облако точек $\{p_i\}$

16 000 точек; ни порядка, ни связей



ориентированное облако $\{(p_i, n_i)\}$

у каждой точки — единичная нормаль

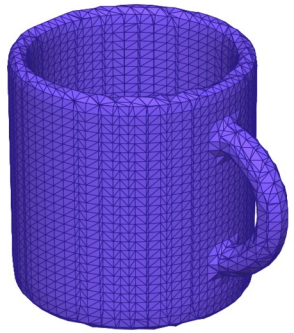
- нормаль превращает точку в плоский диск — хватает, чтобы рендерить и реконструировать
- нормаль — из PCA окрестности: собственный вектор при наименьшем собственном значении

дёшево рендерить сплаттингом и реконструировать поверхность

дорого согласовать знак: локально «наружу» не определено

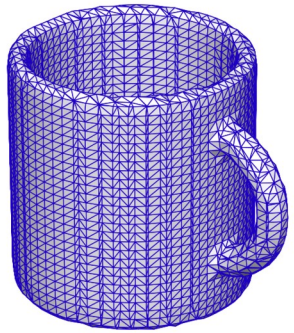
На семинаре 1: без согласования примерно половина нормалей смотрит внутрь

Полигональные сетки: вершины и грани



грани

10744 треугольников



вершины и рёбра

5372 вершин

- V — вершины, F — грани: упорядоченные наборы индексов
- почти всегда треугольники: три точки всегда лежат в одной плоскости, четыре — уже нет
- кусочно-линейная аппроксимация; точность растёт с числом вершин

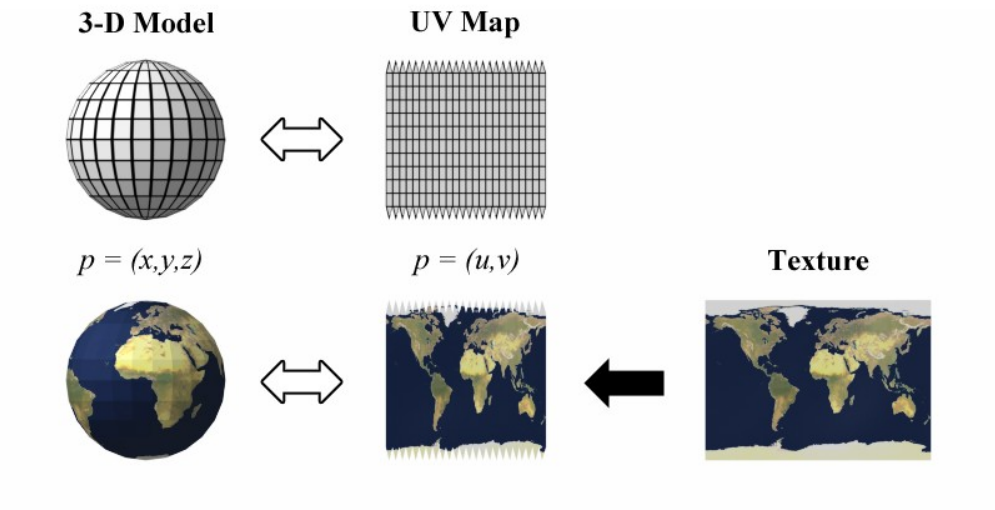
Порядок обхода вершин задаёт нормаль

отсюда отсечение задних граней в растеризации — и вывернутые модели при ошибках экспорта

Стандарт индустрии: в реальном времени всё рисуется треугольниками (лекция 6)

«наружу» у сетки не вычисляют, а читают: нормаль задана порядком перечисления вершин грани

Хорошая сетка и цена операций



форма — в сетке, цвет — в текстуре, связь — через UV

Многообразность (manifold)

каждое ребро принадлежит ровно двум граням, окрестность вершины устроена как диск

нужна большинству алгоритмов обработки; сканеры и marching cubes её дают, игровые движки — не всегда

дёшево двигать вершины — на этом стоят модели тела и лица; сглаживать, упрощать, уплотнять

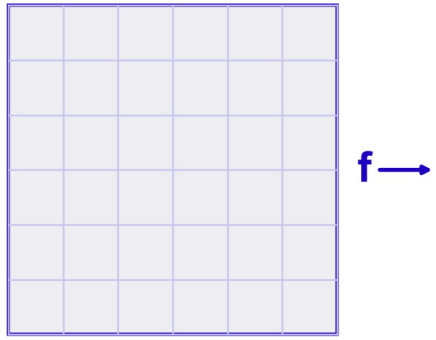
дорого менять связность — неустойчиво

невозможно изменить топологию деформацией — число ручек не меняется

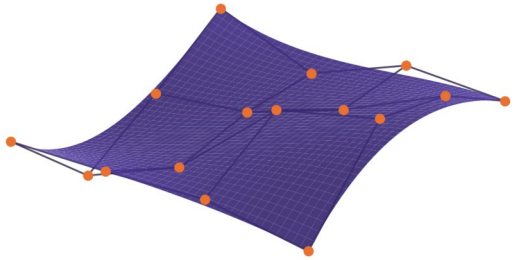
Топологическому ограничению посвящён отдельный раздел лекции

UV-развёртка: Tschmits, CC BY-SA 3.0

Параметрические поверхности



область параметров $\Omega \ni (u, v)$



поверхность Безье и её 4×4 контрольные точки

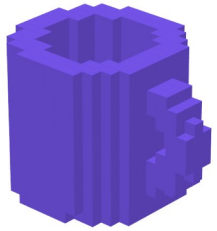
- поверхность — образ отображения $f: \Omega \subset R^2 \to R^3$; топология наследуется от области
- Безье: сетка контрольных точек и полиномы Бернштейна; NURBS — язык промышленного CAD
- в CAD поверхность точная, а не аппроксимированная — треугольники только для экрана (тесселяция)

дёшево сэмплировать точки на поверхности: берём случайные (u, v)

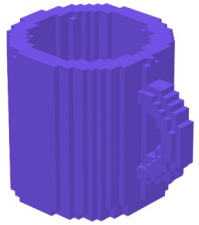
дорого тест «внутри/снаружи»: параметризация задаёт оболочку, а не тело

ту же идею реализуют нейросетью: AtlasNet, Groueix et al., 2018

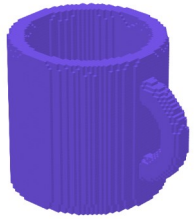
Воксели и октодеревья



16^3
занято 1185 из 4096 ячеек
плотный float32: 0.02 МБ



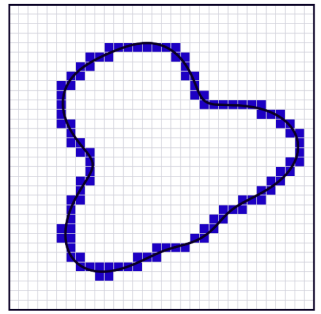
32^3
занято 5155 из 32768 ячеек
плотный float32: 0.12 МБ



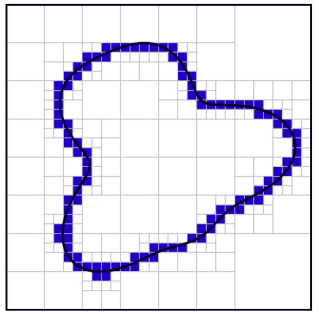
64^3
занято 20719 из 262144 ячеек
плотный float32: 1.00 МБ



исходная сетка
90924 вершин,
181848 граней



плотная сетка: память $O(n^3)$, занято 10%
срез 32×32: хранятся все 1024 ячейки, у поверхности лишь 103



адаптивно: октодерево / hash-grid
295 ячеек вместо 1024: мелкие только у поверхности

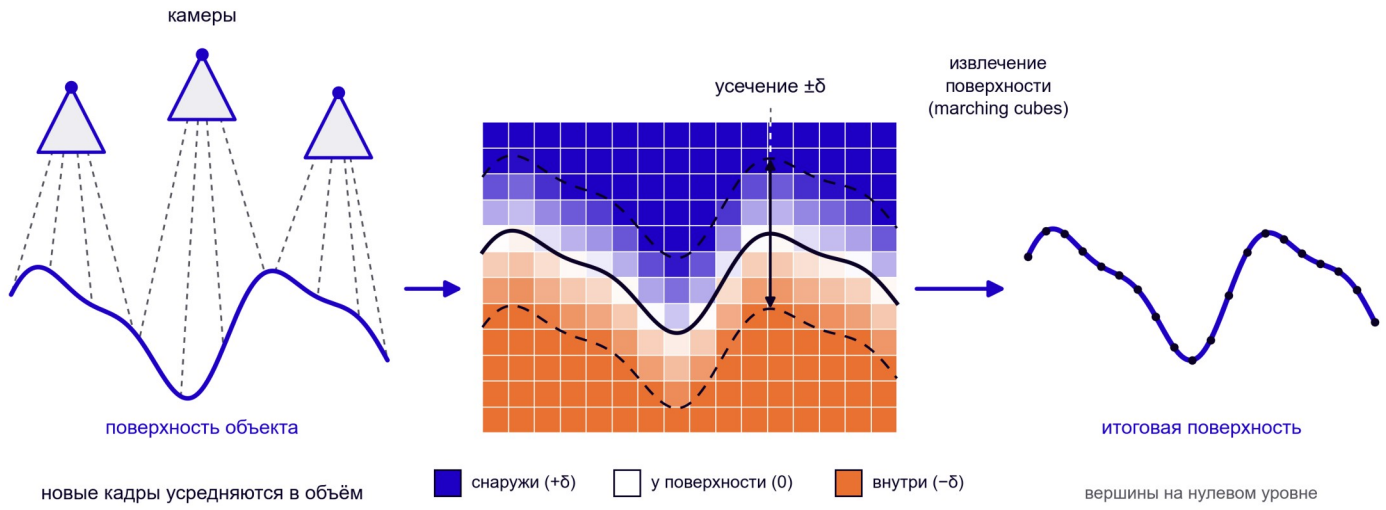
- прямое обобщение растра: регулярная 3D-решётка, работают трёхмерные свёртки
- память $O(n^3)$, а поверхность двумерна: занято 29 % ячеек при 16^3, 8 % при 64^3, около 1 % при 512^3 — и это полгигабайта на объект во float32
- октодерево делит куб на восемь, пока ячейка пересекает поверхность: память падает до $O(n^2 \log n)$

дёшево тест «внутри/снаружи», трассировка луча, свёртки прямо по сетке

дорого память по разрешению и любая деформация — значения живут в узлах

разреженные свёртки и хэш-сетки признаков: Instant-NGP, Müller et al., 2022 (лекция 10)

TSDF: как накапливают глубину



$$D \leftarrow \frac{WD + wd}{W + w}, \quad W \leftarrow W + w$$

в ячейке — не занятость, а усечённое знаковое расстояние и вес уверенности

усреднение подавляет шум сенсора

усечение ±δ не даёт далёким поверхностям мешать друг другу

структура та же, что у вокселей, — изменился только сигнал

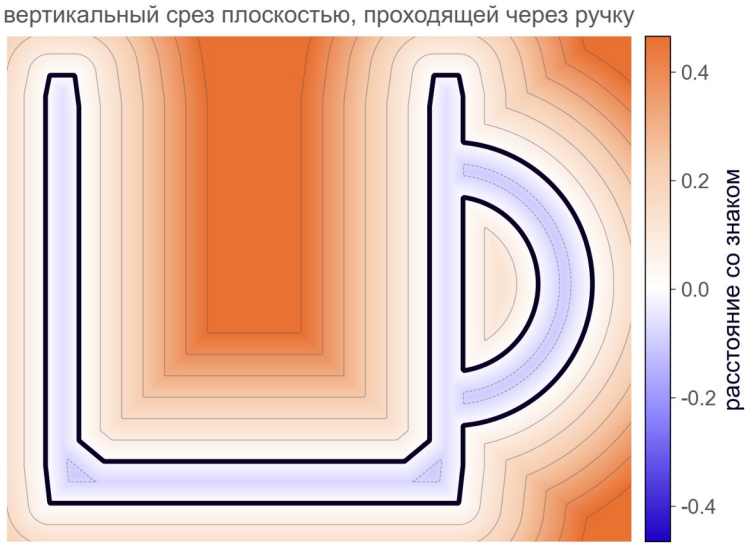
дёшево влить новый кадр: одно взвешенное усреднение по ячейкам

дорого показать результат: нужно извлечение поверхности (marching cubes)

На TSDF построен KinectFusion и почти вся плотная реконструкция в реальном времени

Curless & Levoy, 1996; KinectFusion — Newcombe et al., 2011 (лекции 12 и 15)

Неявные функции: SDF и occupancy



$$\mathcal{S} = \{\mathbf{p} \in \mathbb{R}^3 : f(\mathbf{p}) = 0\}$$

$$\|\nabla f\| = 1$$

уравнение эйконала: длина градиента всюду единица, поэтому нормаль равна градиенту

- знак f — готовый тест «внутри/снаружи»
- градиент — нормаль, без всякой оценки по соседям
- топология меняется непрерывно, без перестройки связности
- разрешение не фиксировано: опрашиваем в любой точке

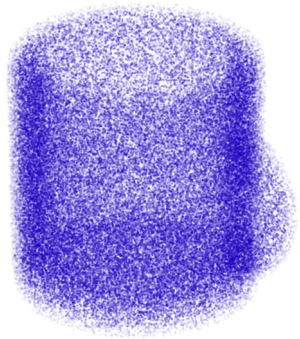
Occupancy — это знак SDF, и ничего больше

классификация вместо регрессии: учить проще, но вдали от поверхности градиент не говорит, где она

дёшево	внутри/снаружи, нормали, смена топологии, любое разрешение
дорого	показать: нужно извлечение — marching cubes или трассировка лучей

DeepSDF — Park et al., 2019; Occupancy Networks — Mescheder et al., 2019; эйкональная регуляризация — IGR, Gropp et al., 2020

Поля плотности и излучения: NeRF



поле плотности σ(p)

нет поверхности — есть «сколько вещества» в каждой точке

$$C(\mathbf{r}) = \sum_i T_i \left(1 - e^{-\sigma_i \delta_i}\right) \mathbf{c}_i$$

- для дыма и тумана поверхности просто нет — есть плотность σ(p) в каждой точке
- NeRF: (точка, направление) → (плотность, цвет); цвет зависит от взгляда — отсюда блики
- формула дифференцируема по всем σ и c, а значит и по весам сети — ради этого представление и выбрано

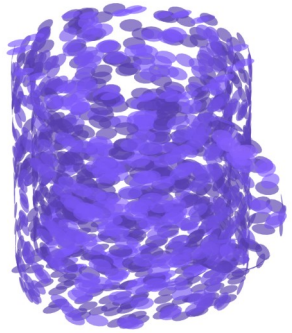
дёшево дифференцируемость всюду и обучение только по 2D-изображениям

дорого поверхности нет: её извлечение — отдельная задача, геометрия шумная

NeRF существует ровно потому, что рендеринг сделали дифференцируемым

Mildenhall et al., 2020; детали объёмного рендеринга — лекция 9

Гауссовы сплаты



гауссовы сплаты

центр, ковариация, непрозрачность, цвет — у каждого

$$\Sigma_k = R_k S_k S_k^\top R_k^\top$$

- сцена — набор гауссиан: центр, ковариация, непрозрачность, цвет (сферические гармоники)
- разложение на вращение и масштаб держит ковариацию положительно определённой при оптимизации
- рендеринг — растеризация и альфа-композитинг: структура как у NeRF, но на порядок быстрее
- число сплатов не фиксировано: гауссианы клонируются, делятся и удаляются

дёшево рендеринг и деформация: сплаты лагранжевы — переносят себя

дорого явной поверхности нет — её извлечение отдельная задача

Ориентированное облако точек, у которого диски получили толщину и прозрачность

Kerbl et al., 2023 (лекция 10)

Инженерный мир: B-rep и CSG



CSG: тело как дерево булевых операций над примитивами

- B-rep — точные поверхности (плоскости, цилиндры, NURBS) плюс топология: как грани, рёбра и вершины связаны
- CSG — дерево булевых операций над примитивами: объединение, пересечение, вычитание
- CAD хранит ещё и историю построения: эскиз, вытягивание, скругление с параметрами

дёшево точность, компактность и правка детали изменением параметра операции

дорого дифференцируемость и вход для нейросети — почти никак

Обратная разработка восстанавливает не форму, а замысел конструктора

Сводная таблица представлений

представление	память	внутри / снаружи	луч	деформация	смена топологии	вход для сети
карта глубины	$O(n^2)$	нет	тривиально	—	—	отлично (растр)
облако точек	$O(n^2)$	нет	сложно	легко	легко	инвариантность к перестановкам
полигональная сетка	$O(n^2)$	если замкнута	быстро	легко	невозможно	операции на графах
параметрическая	мало	трудно	численно	легко	от области	да
воксели	$O(n^3)$	тривиально	просто	тяжело	легко	отлично (3D-свёртки)
октодерево / хэш	$\approx O(n^2)$	тривиально	просто	тяжело	легко	хорошо
SDF / occupancy	любая	по знаку	sphere tracing	легко	бесплатно	отлично
поле плотности	любая	нет понятия	интеграл	легко	бесплатно	отлично
гауссовы сплаты	$O(n^2)$	нет	растеризация	легко	легко	хорошо
B-rep / CAD	мало	точно	аналитически	параметром	операцией	плохо

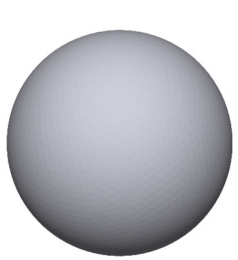
- столкновения в физике — нужны «внутри/снаружи» и замкнутость: воксели, SDF, замкнутая сетка
- генератор мебели — нужна свобода топологии и дифференцируемость: неявные функции
- скан здания для замеров — нужны точность и компактность: облако точек или сетка

Таблицу читают не по строкам, а по задачам: ответ определяется задачей

Эйлерова характеристика и род

$$\chi = V - E + F = 2 - 2g$$

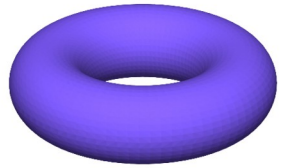
не зависит от того, как поверхность триангулирована



сфера
род 0



кружка
род 1



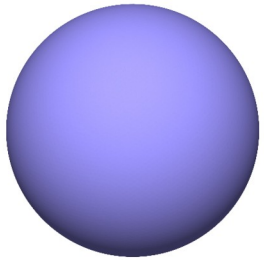
тор
род 1

у кружки и тора одна дырка — непрерывной деформацией одно переходит в другое; сферу так не получить

- род g — число ручек: сфера 0, тор 1, двойной тор 2
- рёбра считают как уникальные пары вершин: наивная оценка 3F вдвое больше (проверите на семинаре 1)

Дырка — это сквозное отверстие, а не полость: полость чашки — вмятина на сфере

Род не меняется деформацией



шаблон: сфера

род 0



предел деформации: отверстие запаяно

род 0



настоящая кружка

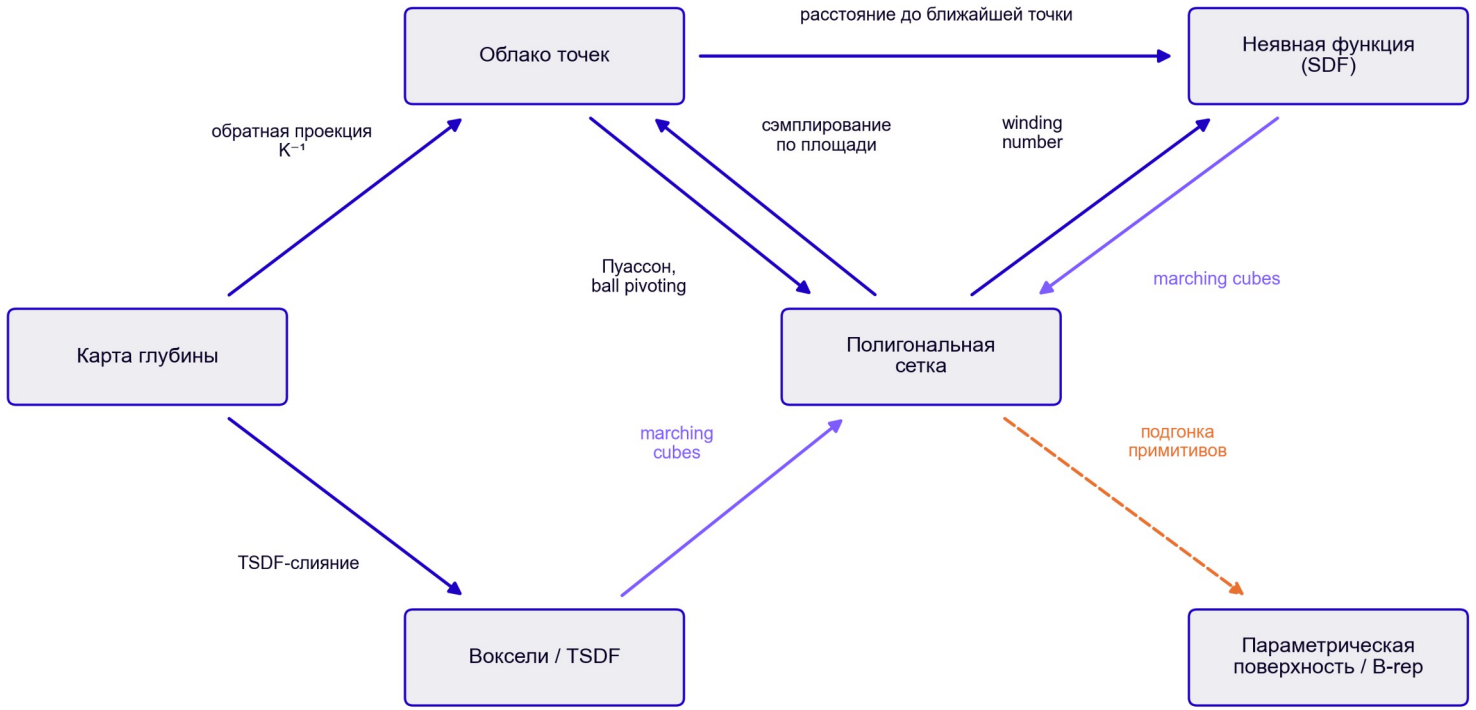
род 1

- методы «деформируй шаблон» стартуют со сферы: Pixel2Mesh, AtlasNet с одной заплатой
- кружку с ручкой такой метод не построит никогда — не «плохо», а принципиально не сможет
- обходные пути — заплаты или шаблон нужного рода — требуют знать топологию заранее

У неявных функций род выпадает из значений f и меняется непрерывно

поэтому генеративные методы формы построены на них, а не на сетках

Карта конвертаций



Что нужно

обратная проекция
калибровка K

TSDF-слияние
позы камер

Пуассон, ball pivoting
ориентированные нормали

подгонка примитивов
работает плохо

Для каждой пары — свой метод; общего решения не существует

но основных переходов немного, и все они хорошо изучены

Карта курса: куда это всё пойдёт

тема этой лекции	где развивается
проекция и потеря размерности	лекция 3 — модель камеры, калибровка
нормали и вращения как точки на многообразии	лекция 2 — SO(3), кватернионы, оптимизация
доопределение вторым видом	лекции 4–5 — эпиполярная геометрия, SfM
рендеринг сеток и лучей	лекции 6–8 — растеризация, трассировка, освещение
инверсия рендерера, NeRF и сплаты	лекции 9–10 — дифференцируемый рендеринг
карты глубины и 2.5D	лекция 11 — оценка глубины
TSDF, marching cubes, реконструкция по облаку	лекция 12 — многовидовая реконструкция
point map	лекция 14 и заключение — feed-forward реконструкция
октодеревья и карты	лекция 15 — SLAM

Каждая следующая лекция — выбор представления плюс способ доопределить задачу

представление берём из сводной таблицы, способ доопределить — из трёх источников: больше наблюдений, физические приоры, выученные приоры

Резюме лекции

- обратная задача зрения некорректна: одному изображению отвечает бесконечно много сцен
- это свойство самой задачи, а не нехватка вычислений — быстрый компьютер единственности не вернёт
- доопределяют тремя способами: больше наблюдений, физические приоры, выученные приоры
- современная форма решения — инверсия дифференцируемого рендерера, отсюда требование дифференцируемости
- универсального представления нет: это структура данных плюс операции, которые она делает дешёвыми
- род поверхности не меняется деформацией, и метрика об этом не предупредит
- переход между представлениями — свой метод на каждую пару, и у каждого своё входное требование