

*Курс «Трёхмерное компьютерное зрение»*

**Лекция №1**

**«Введение. 3D модели»**

Антон Конушин

2025 год

# Команда курса

---



**Антон Конушин**

К.ф.-м.н. ,с.н.с. Института AIRI, руководитель группы  
«Пространственный Интеллект»

Доцент ВМК МГУ и ФКН НИУ ВШЭ

Программный директор фонда «Интеллект»



**Дмитрий Сенюшкин**

Научный сотрудник Института AIRI, руководитель группы  
«Автономное зрение»

Главный ассистент – Тимур Мамедов

# Краткая история курса

---

- Курс ШАД и ВМК «Компьютерное зрение» с 2009 года, в начале семестровый, со второго раза годовой
- Тема 3D зрения, в основном 3Д реконструкция во втором семестре, во второй части курса
- Ближе к 2020 курс сократили до 1 семестра, в конце оставили только 3 лекции по камере и 3Д реконструкции
- В 2021 решили в основном курсе «Компьютерное зрение» оставить только 2Д зрение, а по теме 3Д зрения сделать отдельный мини-курс
- В 2022 первый раз его читали, и с тех пор постепенно расширяем и дополняем

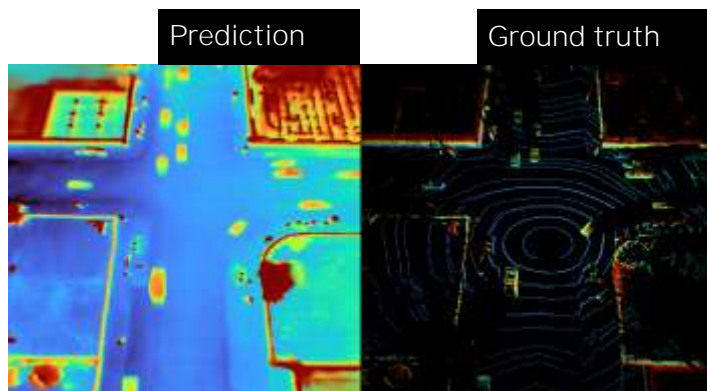
# Почему 3D зрение?



Рис. 2 Общая схема работы системы построения 3-х мерных моделей по набору изображений

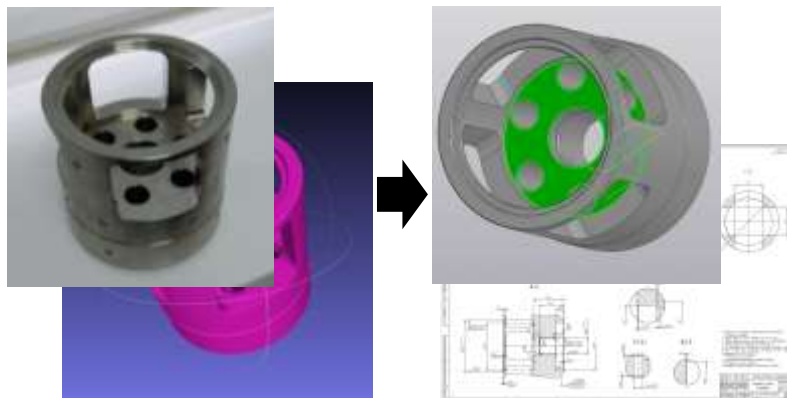
# Текущие задачи в Институте AIRI

## Perception Foundation Model



- Цель - Self-supervised multi-modal 4D perception-forecasting model
- Приложения – беспилотные автомобили

## Reverse Engineering

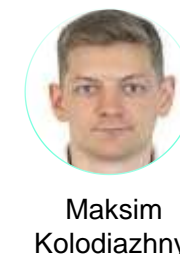
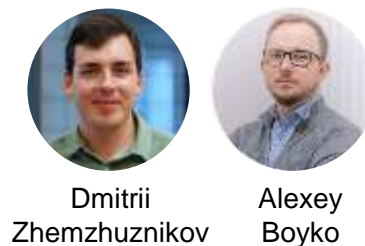
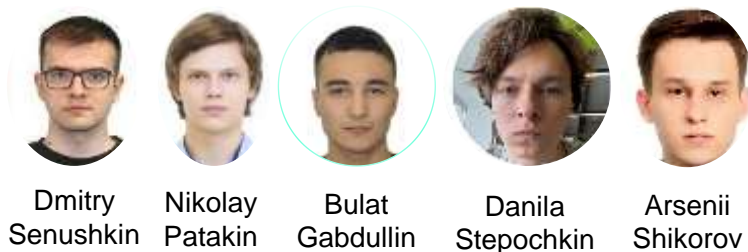


- Цель – построение CAD моделей по сканам и фото механических деталей
- Приложение – производство деталей для ремонта сложного оборудования

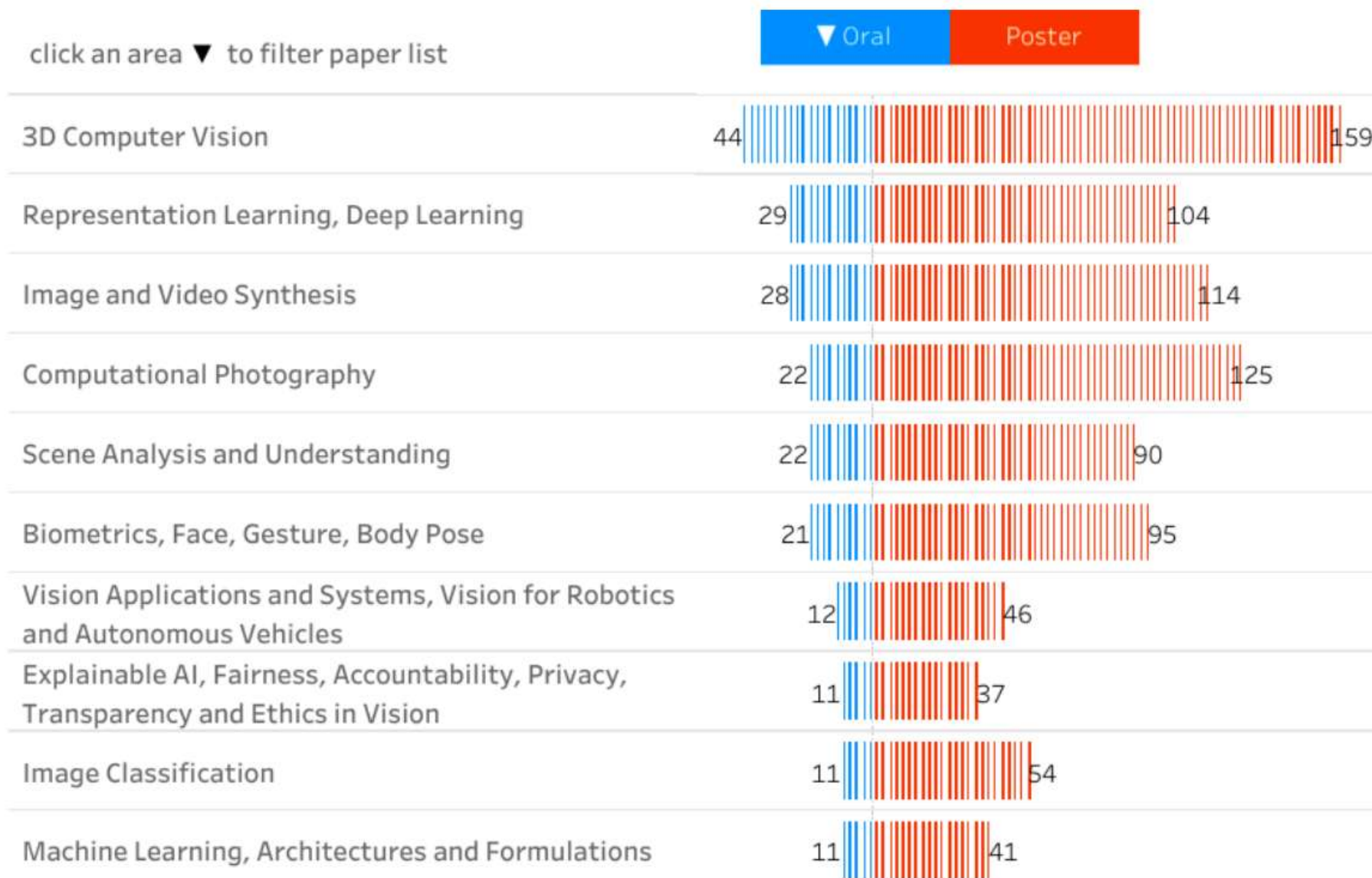
## Indoor2CAD



- Цель – построение семантических 3D моделей и CAD моделей помещений
- Приложение – построение цифровых моделей для планирования ремонтных работ и робототехники



# 3D Зрение – «горячая область»



3-х мерное зрение (3D vision)

# Что такое компьютерное зрение?

---



**Задача зрения:** понять, что находится на изображении

**Компьютерное зрение:** построение компьютерной модели системы зрения

**Компьютерное зрение** – часть области искусственного интеллекта (AI)

**Тест Тьюринга для компьютерного зрения:**  
Ответить на любой вопрос про изображении, на который может ответить человек.

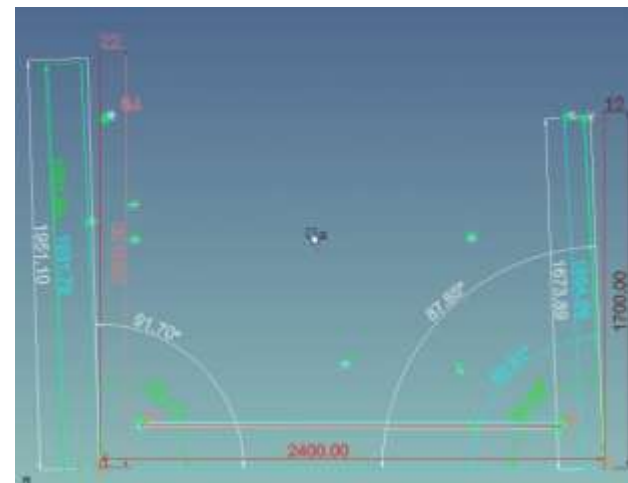


# Что и где? (Семантическое зрение)



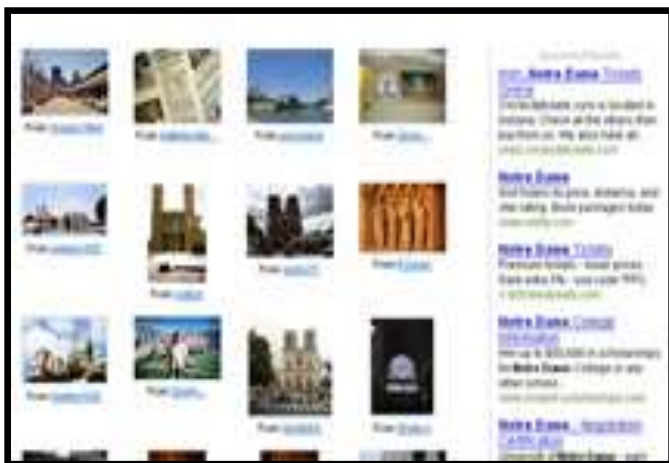
Необходимо определить, есть ли на изображении объекты заданного типа и если да, то определить их положение

# Какой формы? (Метрическое зрение)



Фотограмметрия

Source: Pollefeys et al.

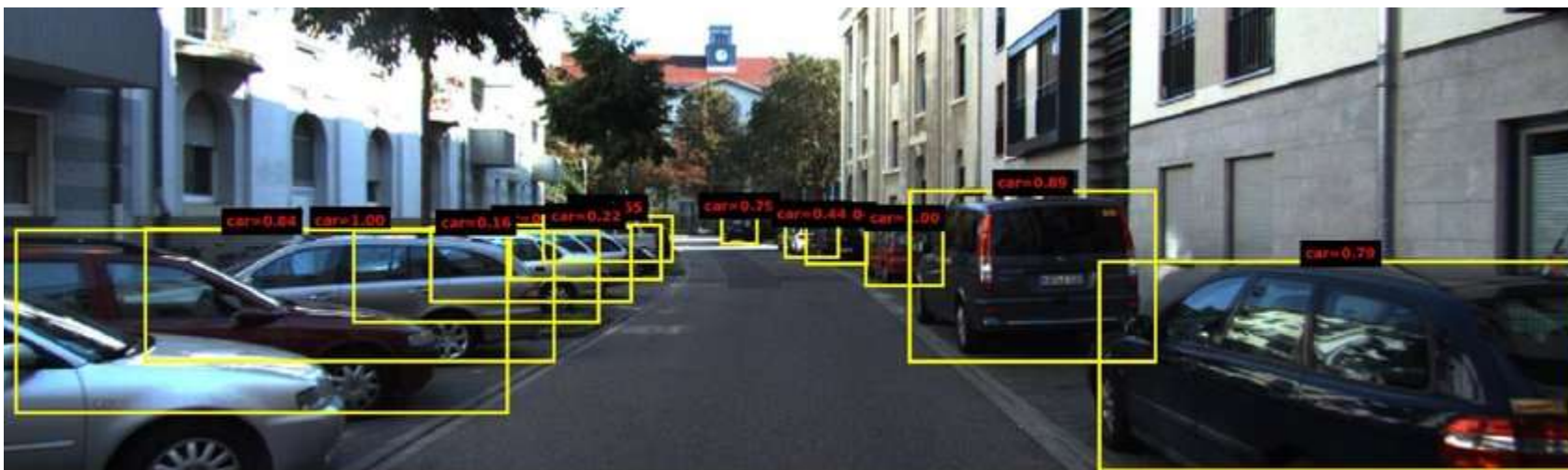


3-х мерная реконструкция по изображениям

# 3D зрение

---

3D зрение = Метрическое зрение + Семантическое зрение + 3D



2D детекция

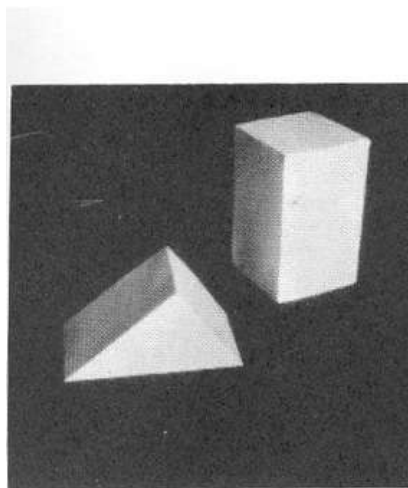


3D детекция

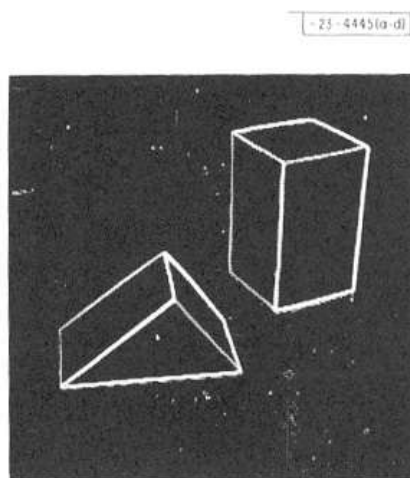


# Компьютерное зрение изначально в 3D!

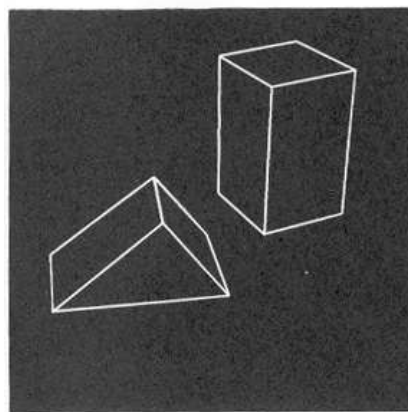
---



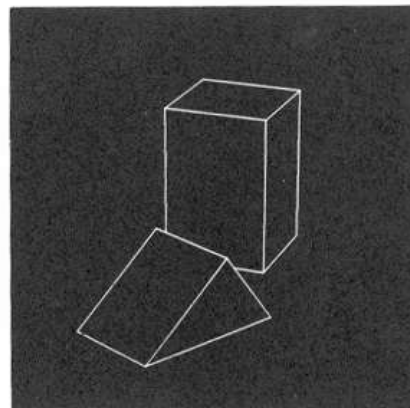
(a) Original picture.



(b) Differentiated picture.



(c) Line drawing.



(d) Rotated view.

L. G. Roberts, *Machine Perception of Three Dimensional Solids*,  
Ph.D. thesis, MIT Department of  
Electrical Engineering, 1960

# Пространственный интеллект (Spatial AI)

---

- Spatial AI – основа мозгов роботов
- 3D зрение – это важный компонент Spatial AI
- Также 3D зрение важно для AR/VR

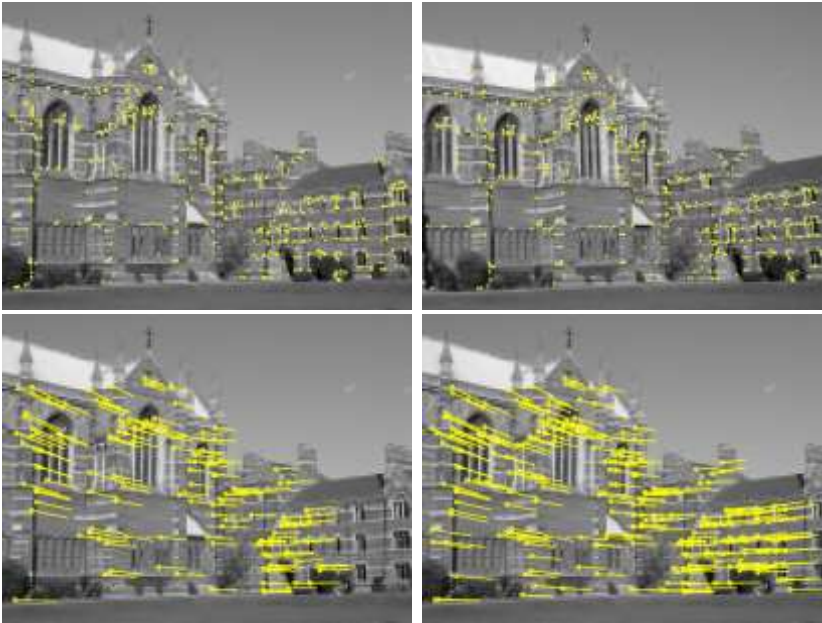


Какие задачи можно отнести к 3D зрению?

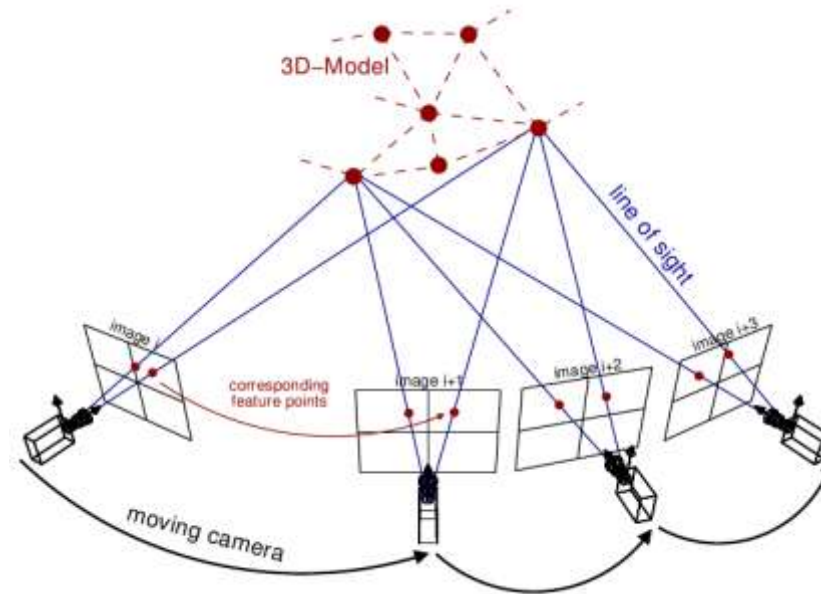
Photos by [FDATA ROBOT](#) on [Unsplash](#)

All trademarks are the property of their respective owners

# Оценка положения камеры в 3D



Point features and matching



Camera localization and structure estimation

- Есть набор кадров / видео
- Нужно определить положение камеры в пространстве или движение камеры в пространстве относительно первого кадра
- Какие области применения?

# 3D реконструкция по изображениям

---



Набор изображений



Трёхмерная (3D) модель

Пример задачи: построение 3D модели реального объекта по набору фотографий из интернета



# Оценка глубины

---

Source



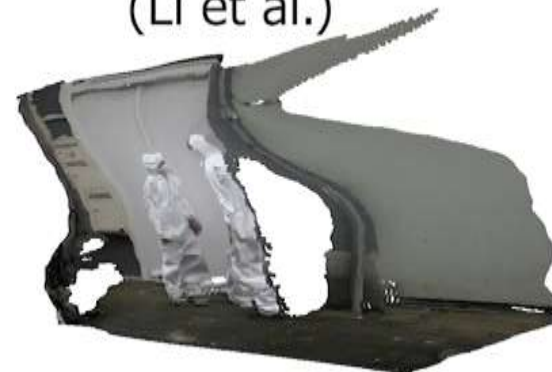
Ours, B5-LRN4



Megadepth



Mannequin Challenge  
(Li et al.)



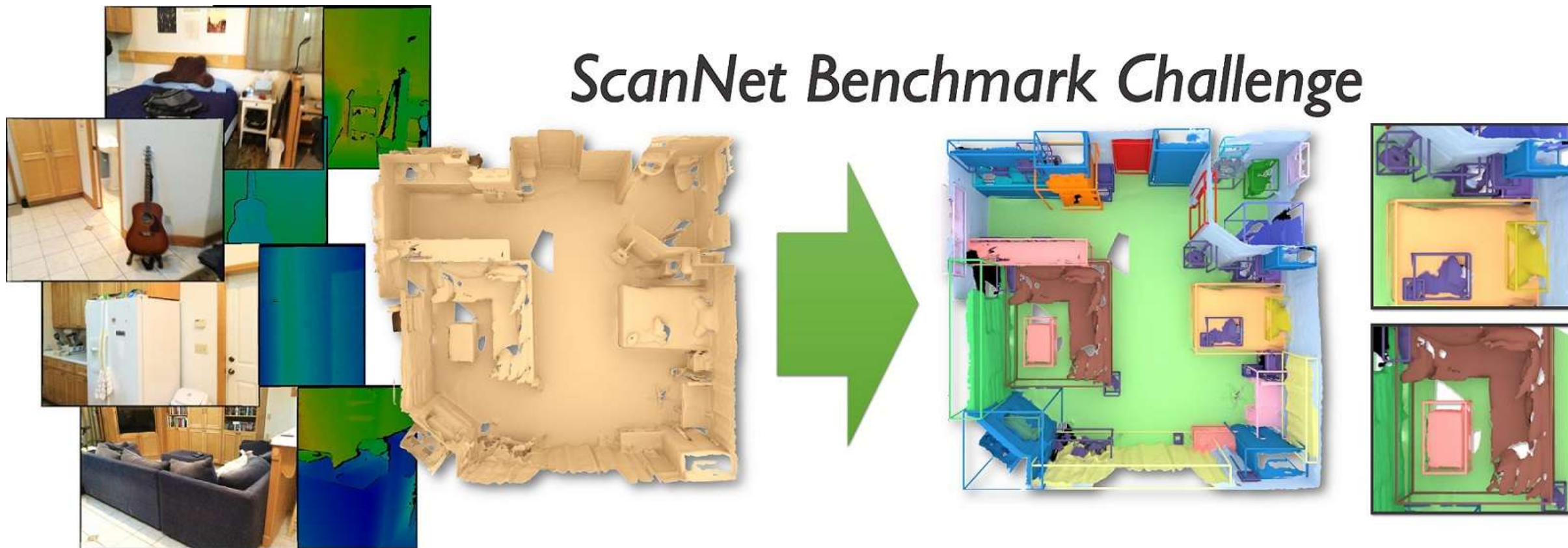
MIDAS  
(w/o alignment)





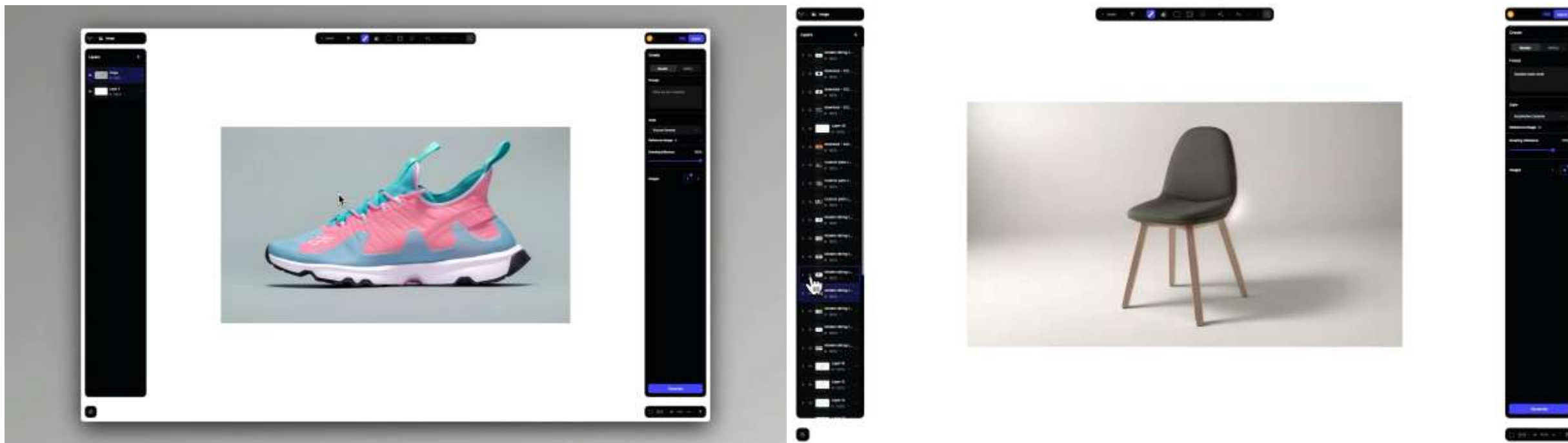
# 3D детекция и сегментация

---



# Генерация 3D объектов

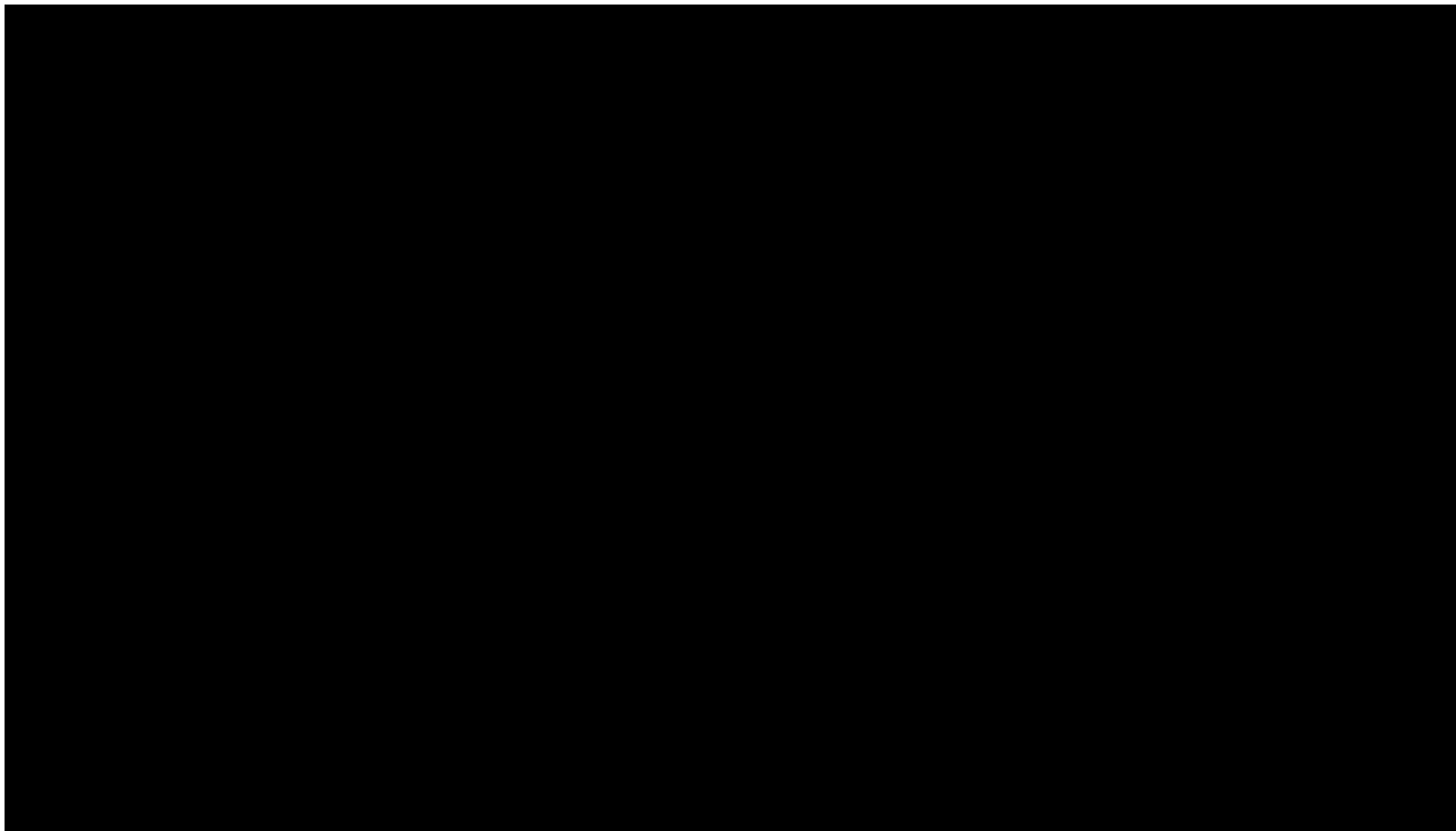
---



Пример – продукты компании Vizcom.AI, основанной в 2021. Цель - расширить существующий пайплайн на базе 2D рисунков, уже существующий у concept designers, за счёт генеративных моделей.

# Аватары человека

---



# План курса

# Общий план

---

1. Введение. 3D модели (Антон Конушин)
2. Модели камеры (Дмитрий Сенюшкин)
3. Перспективная проекция (Дмитрий Сенюшкин)
4. Многовидовая геометрия (Антон Конушин)
5. SLAM и SFM (Антон Конушин)
6. Дифрендеринг 1 (Дмитрий Сенюшкин)
7. Дифрендеринг 2 (Дмитрий Сенюшкин)
8. Оценка глубины (Антон Конушин)
9. 3D реконструкция (Антон Конушин)
10. 3D генерация (Антон Конушин)
11. *Аватары человека (Ренат Баширов)*
12. *CAD (Антон Конушин)*

# О курсе

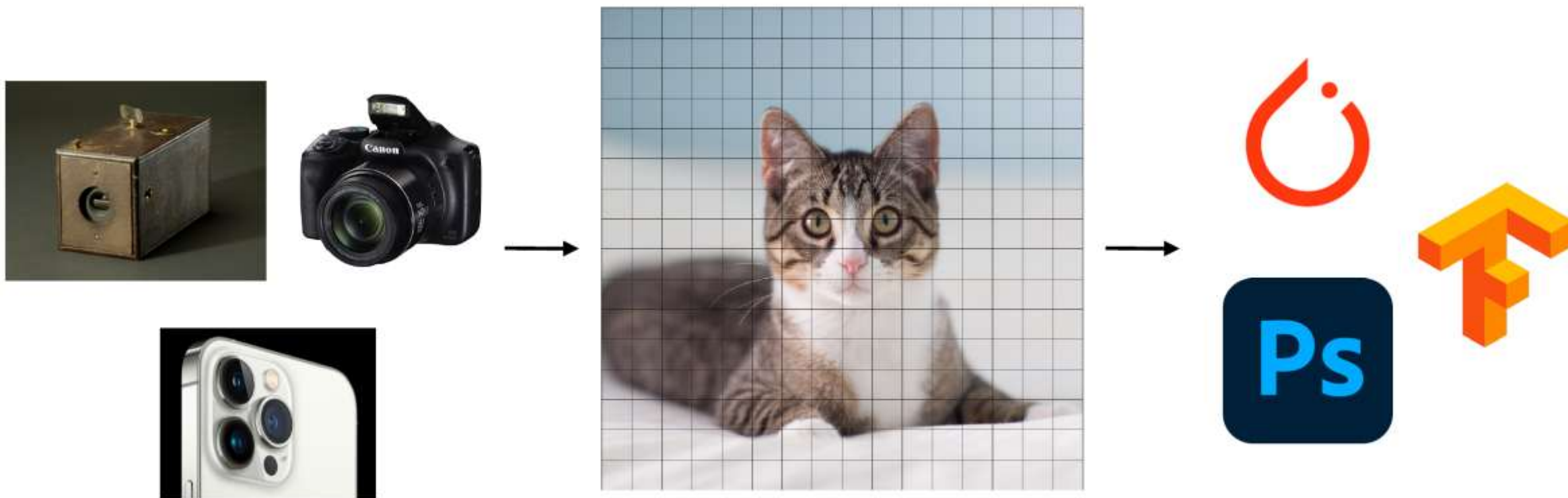
---

- Страница курса
  - <https://code.mipt.ru/courses-public/cv/3d/public>
  - Слайды, видеозаписи
- Система для сдачи и проверки заданий
  - <https://fall.cv-gml.ru/>
- Практическая часть
  - 4 домашних задания
  - М.б. успеем разработать новые задания (выдадим как бонусные)
- Теоретическая часть
  - Лекции
  - Итоговая письменная контрольная работа (экзамен)

3D модели

# 2D представление

---

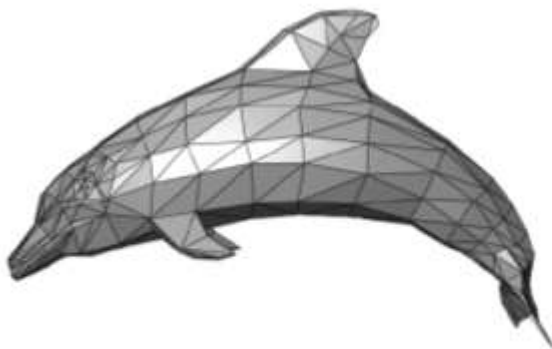


Универсальное представление – 2D изображение (матрица  $\text{width} \times \text{height} \times \text{channel}$ ) между сенсорами, методами обработки и визуализации



# Тем временем в 3D мире...

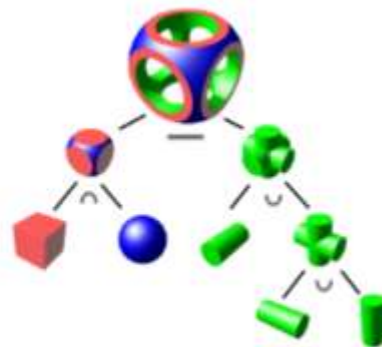
---



$$\{\mathbf{p} \mid f(\mathbf{p}) = 0\}$$



$$f(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^3$$



and many more ...

# Карты глубины (Depth Maps)

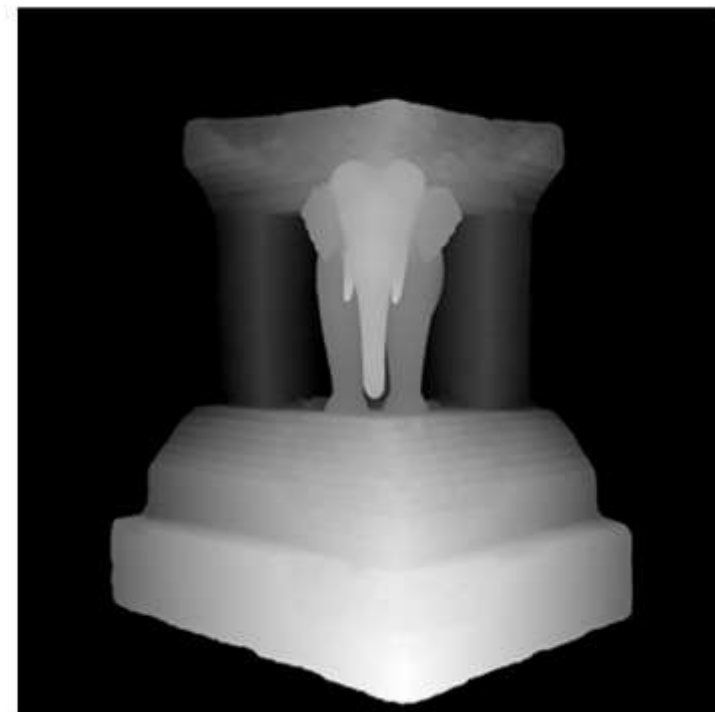
---



Сенсоры глубины



Изображение

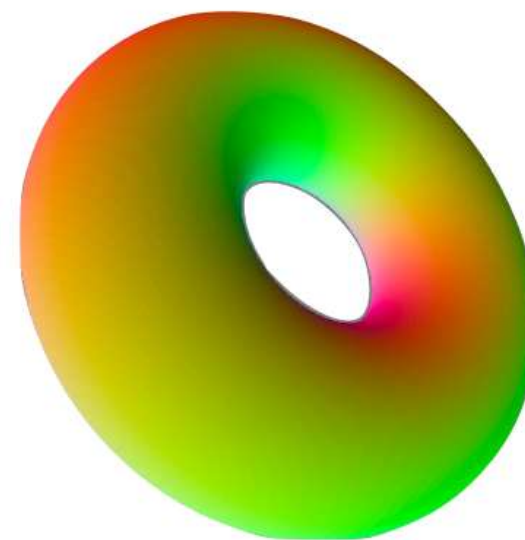
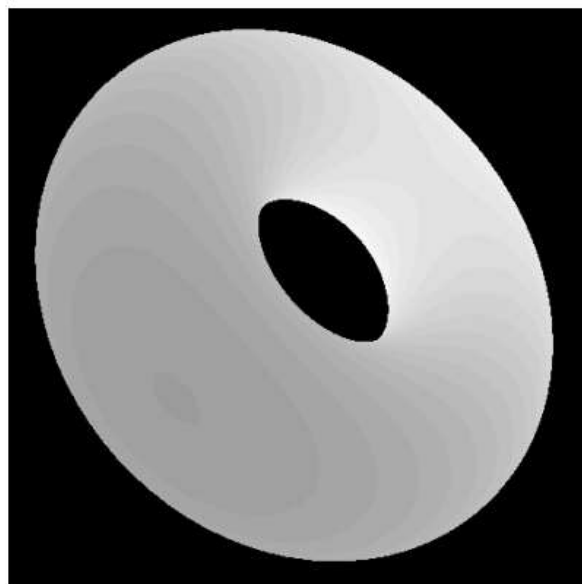
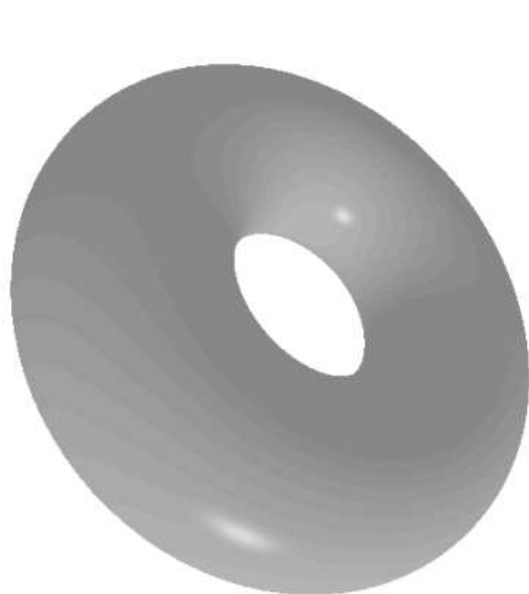


Карта глубины

- Карта глубины – изображение с 1м каналом, в котором записаны расстояния до ближайшего объекта сцены
- Часто объединяют с RGB изображением, получается RGBD изображение
- Можем использовать обычные методы обработки изображений

# Карты глубины и карты нормалей

---

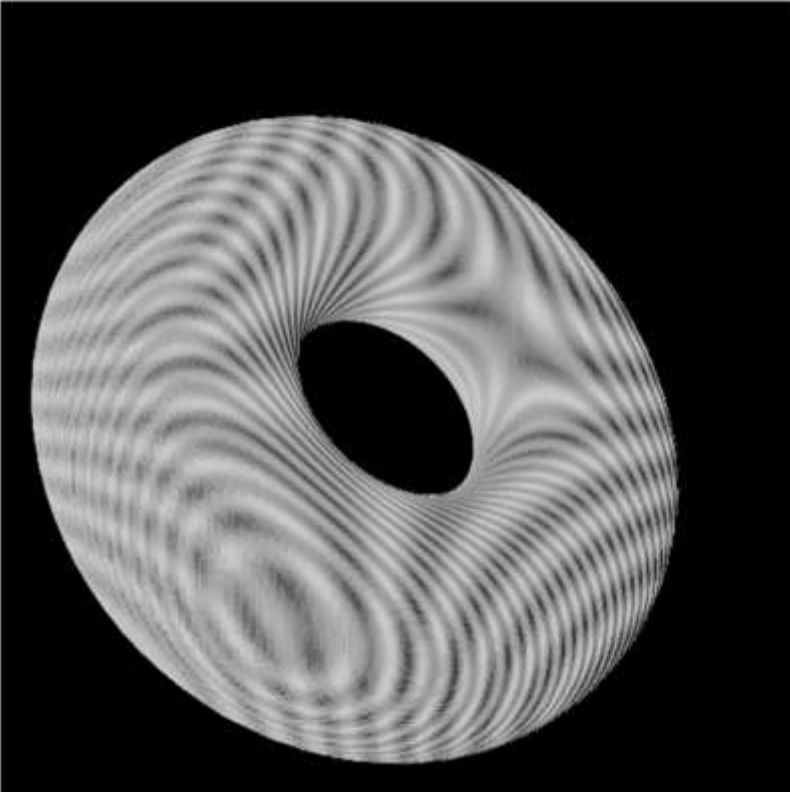


$$N[\mathbf{p}] \in \mathbb{S}^2$$

- Представление 3Д информации через изображения можно расширить и на другие геометрические свойства, например, нормали
- Получим карту нормалей, часто в дополнение к карте глубины

## 2.5D представления

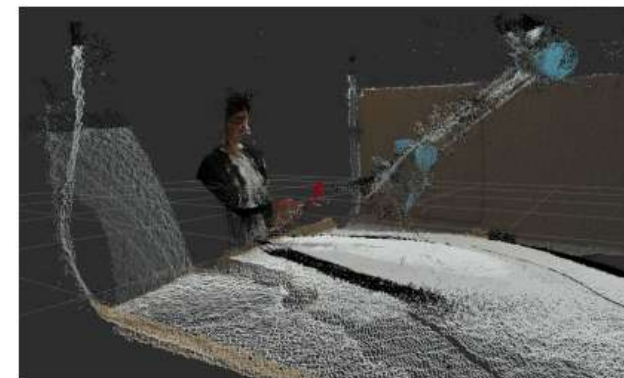
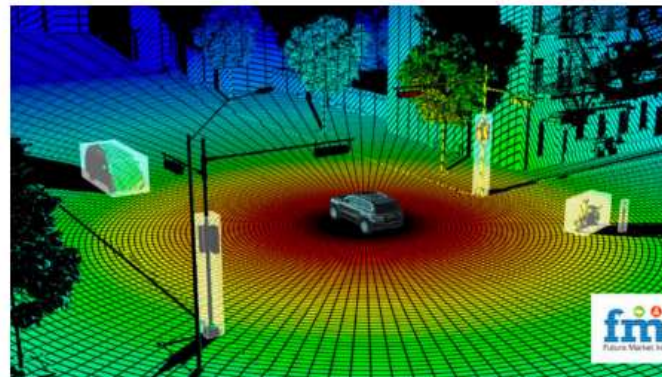
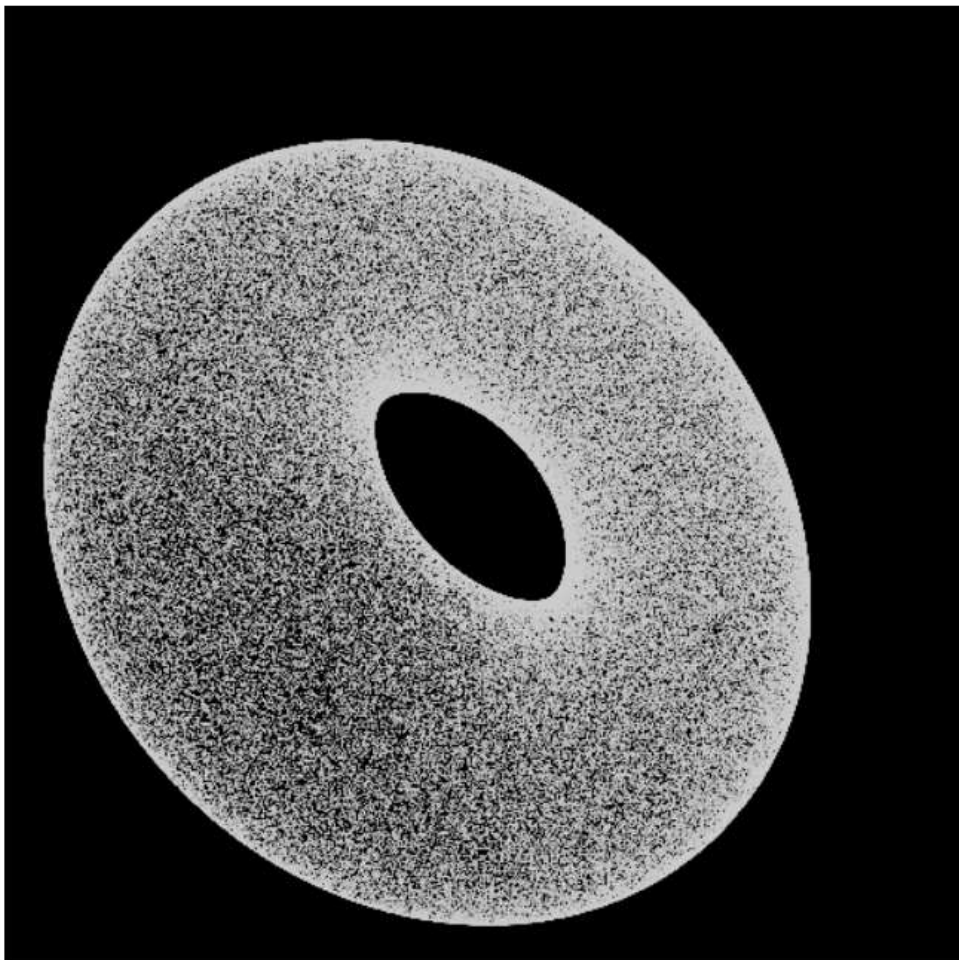
---



- Карты глубины и карты нормалей описывают только видимую часть сцены
- Это 2.5D представление, 3D информация ассоциирована с пикселями изображения
- Не полноценные 3D модели
- Как мы можем «ослабить» связь 3D информации с изображением?

# Облако точек

---

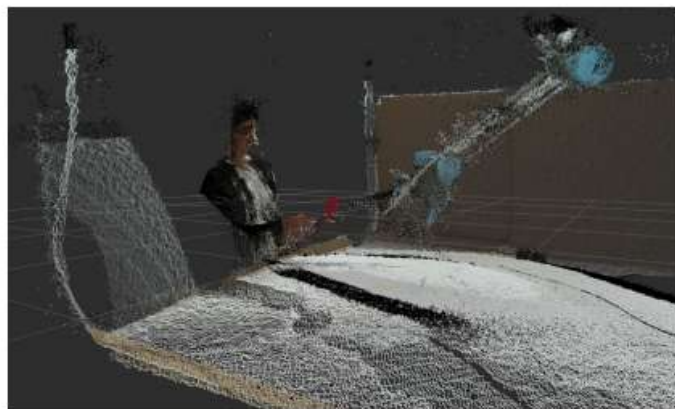
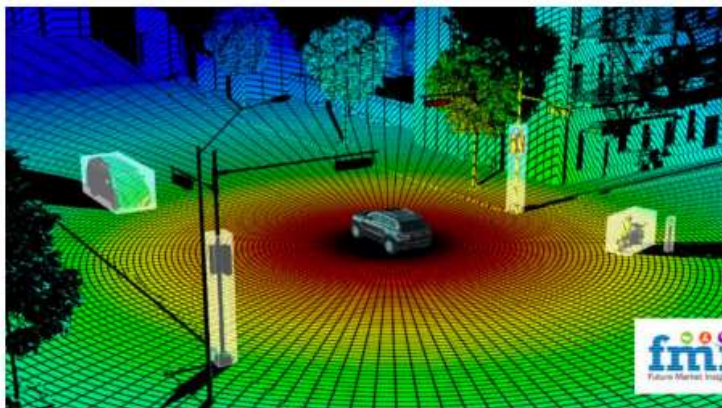


- Каждому пикселю изображения с глубиной мы можем сопоставить 3Д точку
- Множество пикселей даст облако точек



# Облака точек

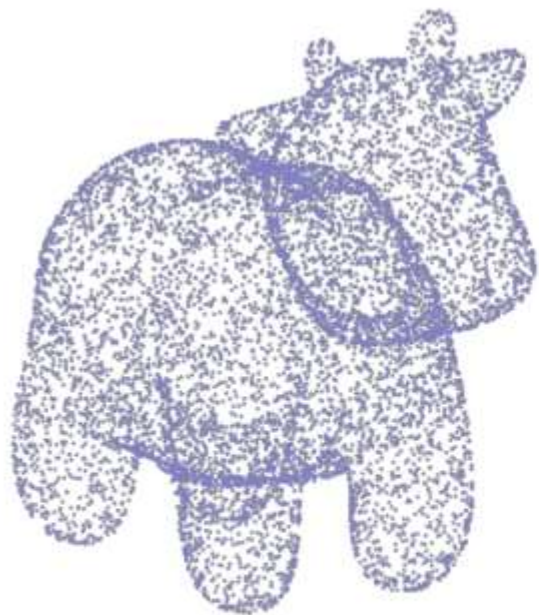
---



- Облака точек обычно получаются (“acquired”) процессом, подобным получению изображений
- Например, lidar строит «панораму» окружающей сцены
- Но бывает удобно «забыть» про исходную природу 3д точек
- Мы можем объединить информацию с разных ракурсов в одно общее облако точек
- Это удобно, поскольку позволяет исключить информацию о типе и природе сенсоров, с которых облако точек получено и работать чисто с облаком точек

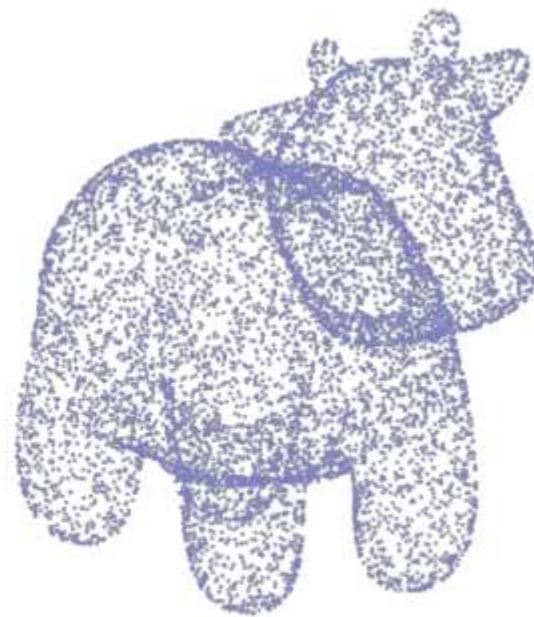
# Облако точек (point cloud)

---



$$\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$$

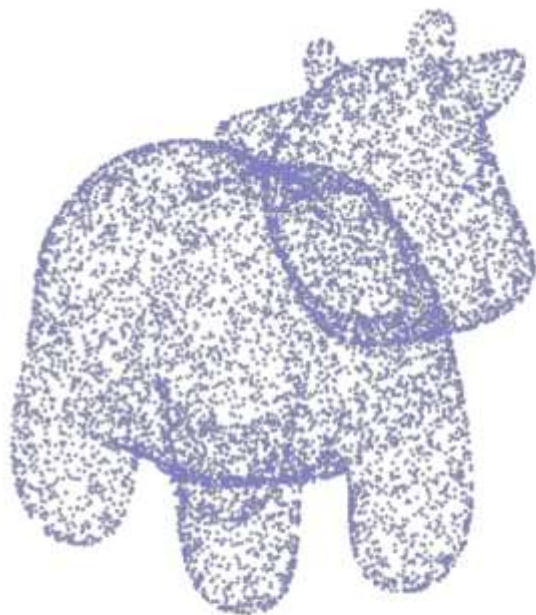
Permutation  $\sigma$   
 $\longrightarrow$



$$\{\mathbf{p}_{\sigma(1)}, \mathbf{p}_{\sigma(2)}, \dots, \mathbf{p}_{\sigma(N)}\}$$

# Облако точек

---



$$\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$$

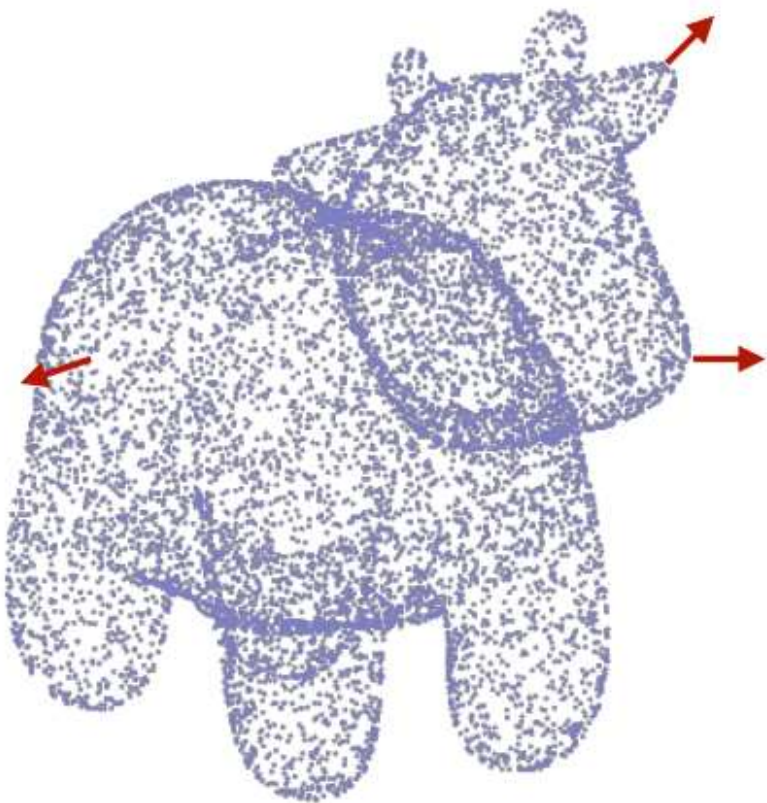
x1,y1,z1
.
.
.
.
.
.
.
.
.
.
.
.
.
.

- Храним 3Д точки в массиве
- В отличие от изображений, порядок точек не важен
- Для обработки облаков точек нужны методы, которые инвариантны к перестановкам точек
- Какие нейросети подходят и не подходят для обработки точек из-за этого?



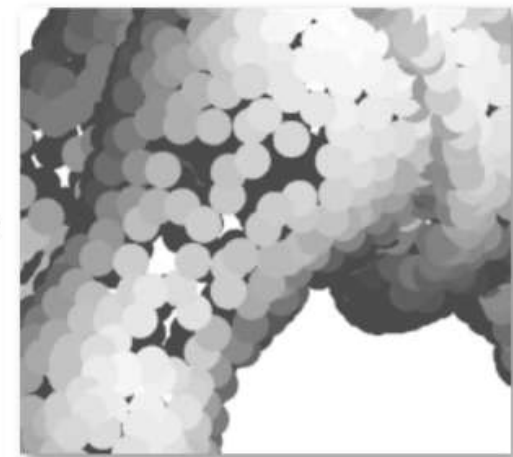
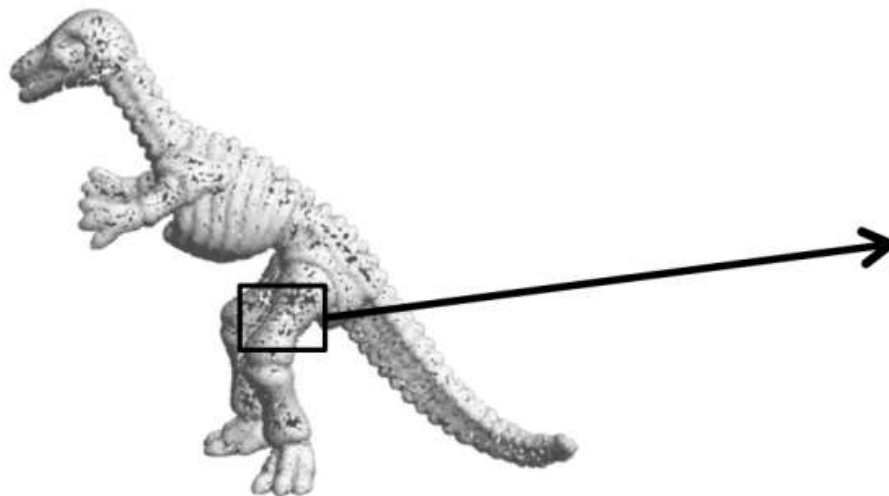
# Ориентированное облако точек

---



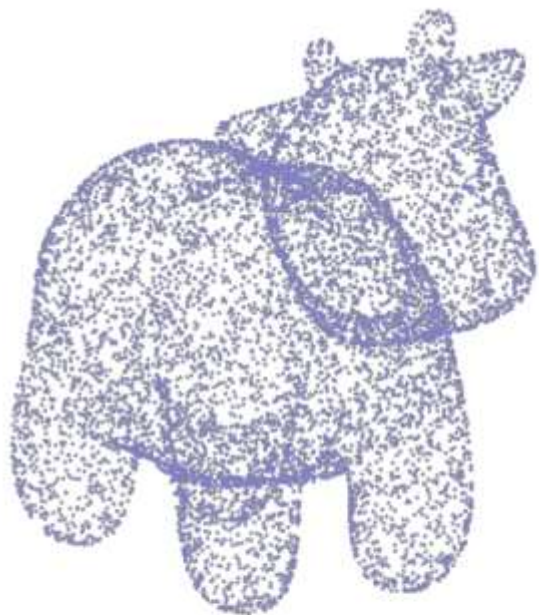
$$\{(\mathbf{p}_1, \mathbf{n}_1), (\mathbf{p}_2, \mathbf{n}_2), \dots, (\mathbf{p}_N, \mathbf{n}_N)\}$$

- Если есть нормаль, тогда можем приближенно восстановить поверхность в точке
- Например, плоским ориентированным диском
- Полезно для рендеринга

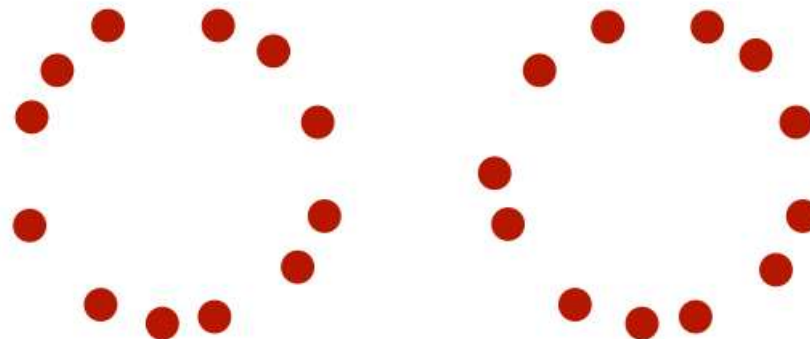


# Облако точек

---

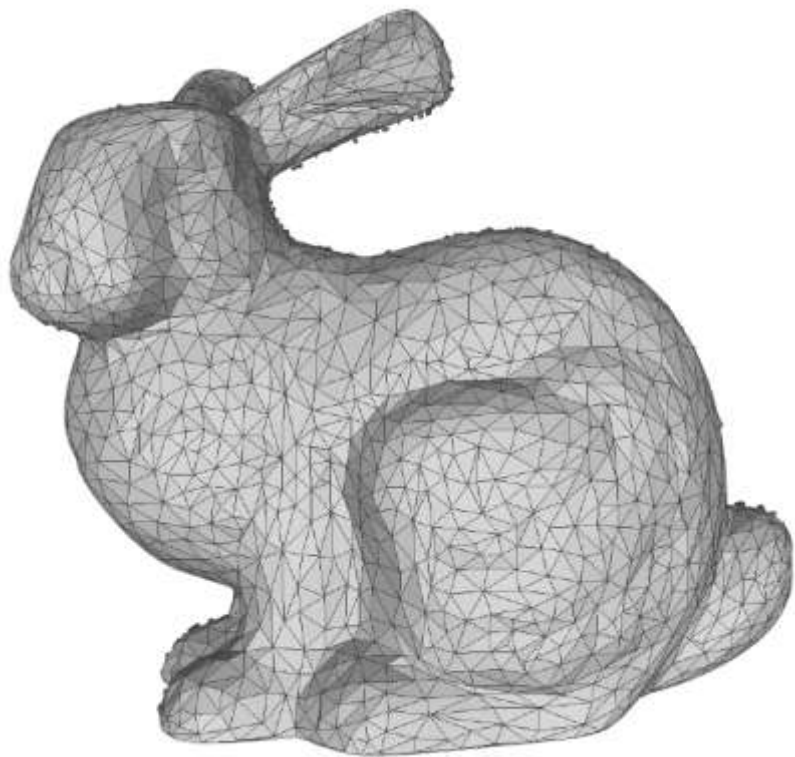


$$\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$$



- Нет информации о связанности точек
- Для обработки чаще всего информация о точках в окрестности нужна

# Полигональная сетка (mesh)



## Vertices

[illegible]

# Faces

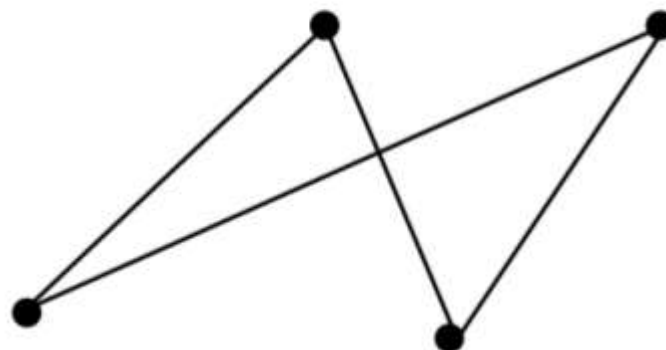
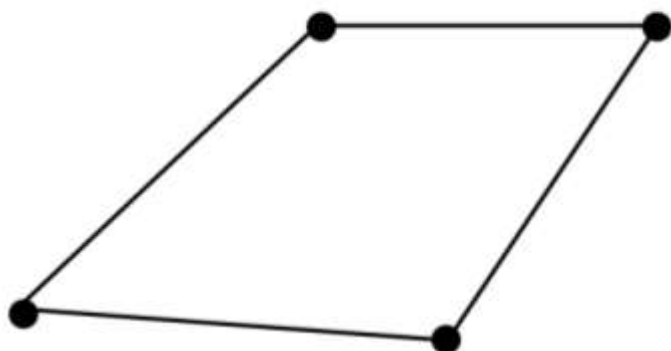
[illegible]

- Вершины (vertices) и грани (faces)
- Кусочно-линейное представление поверхности объекта
- Чем больше вершин, тем точнее представление поверхности

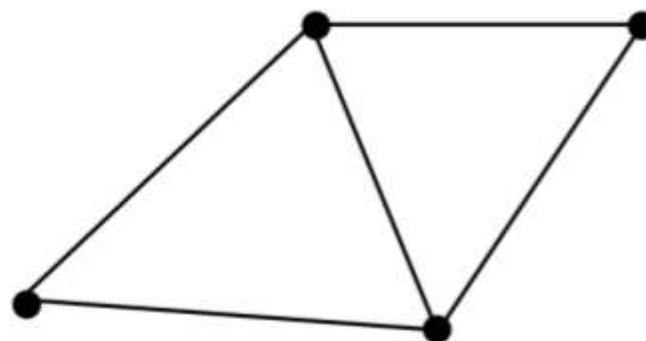
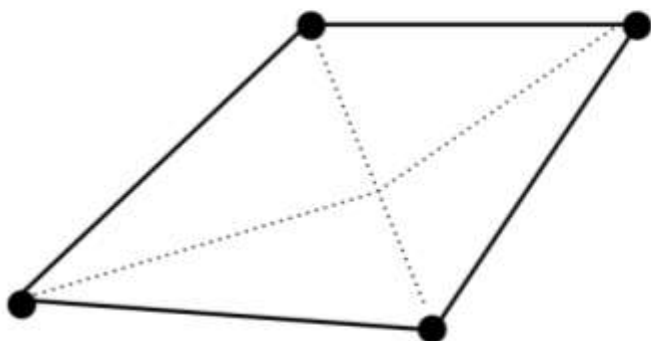
# Порядок вершин важен!

---

Ordering Matters in Face Indices



Arbitrary Polygons can be non-planar



restrict to *Triangular* Meshes

# Текстурирование полигональных сеток

---



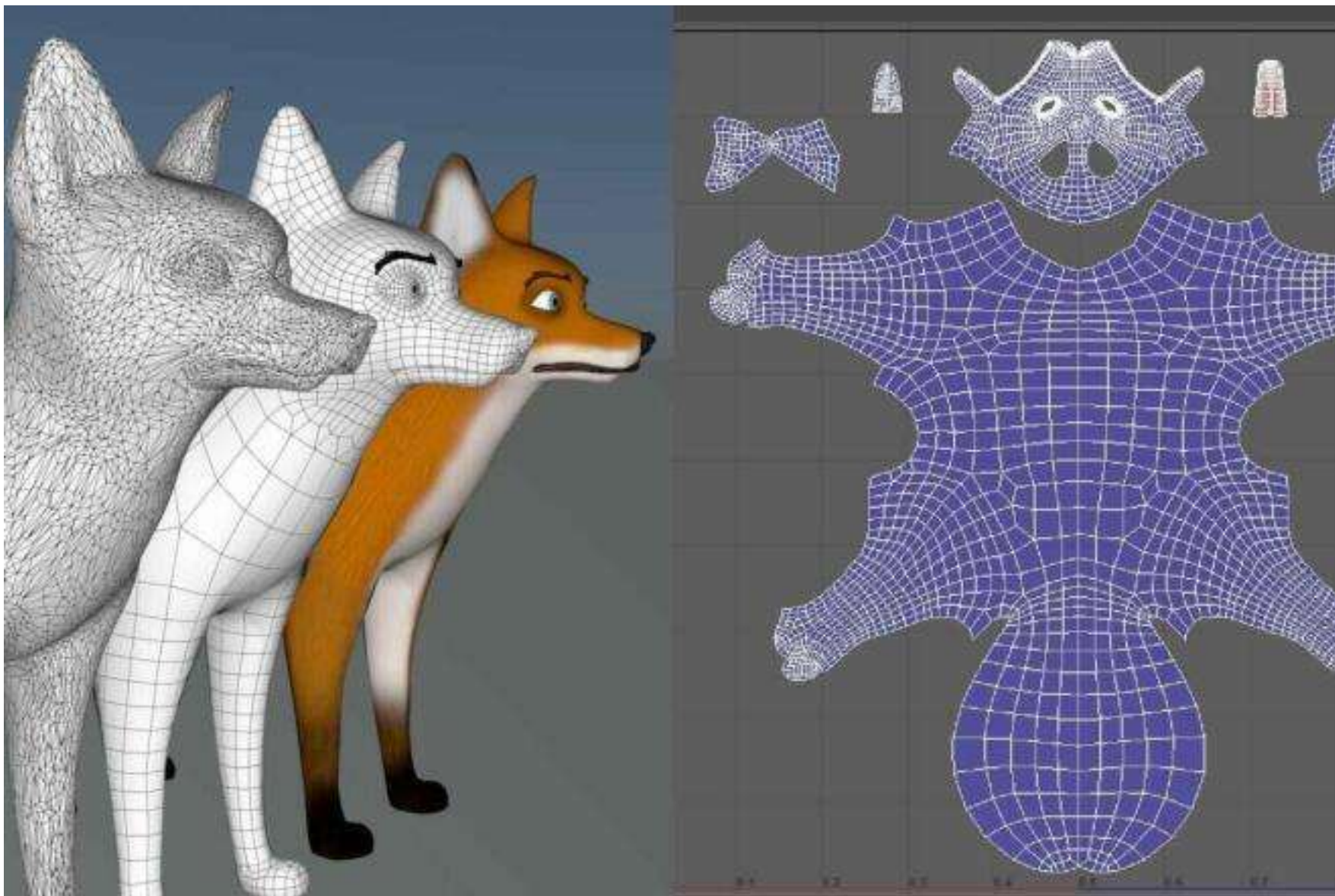
Efficient to compute ray-  
triangle intersections



Easy to texture and render  
(common representation  
across graphics)

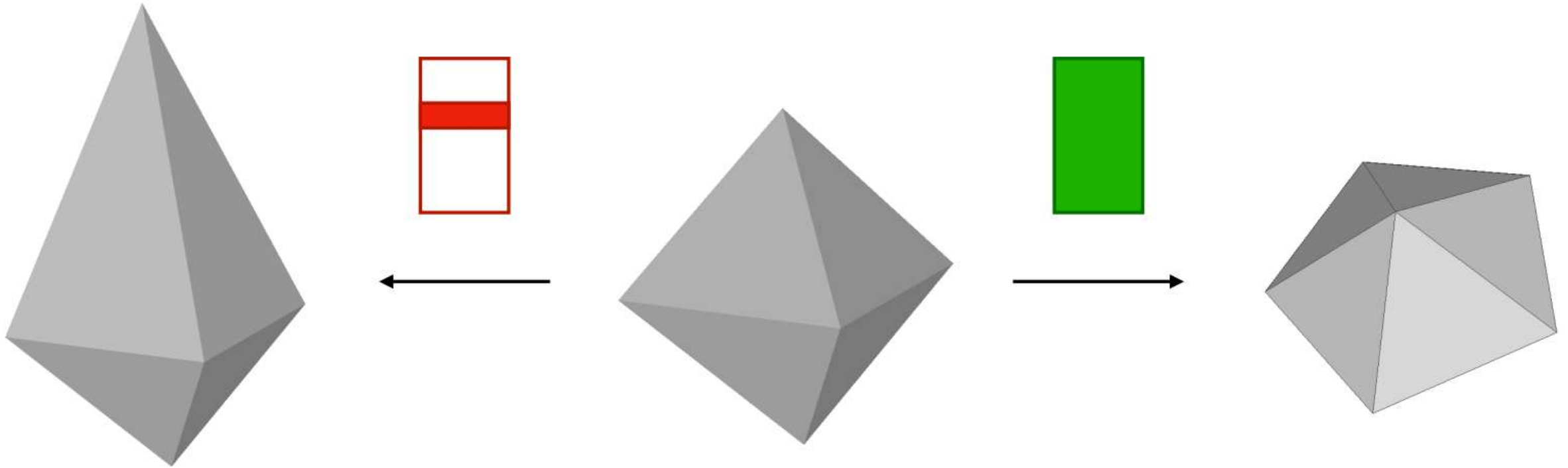


# Развёртка текстуры



# Полигональные сетки

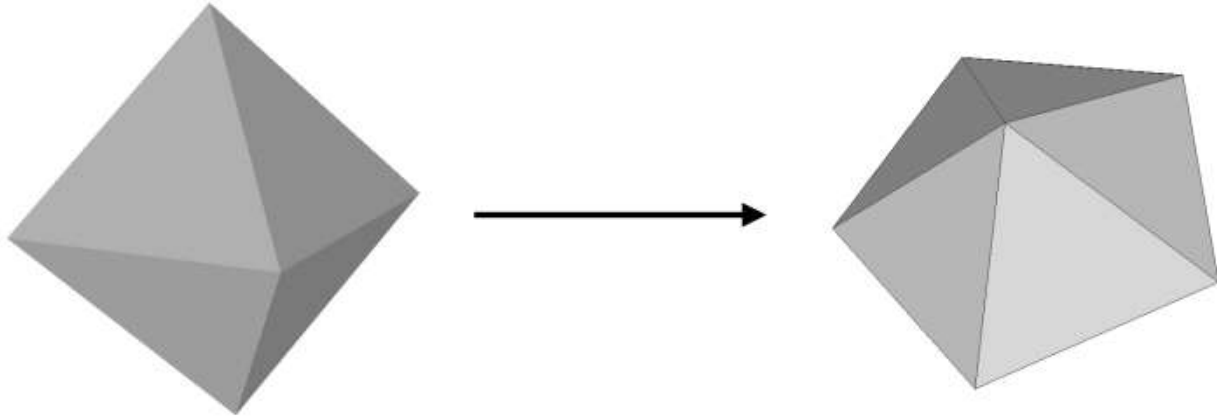
---



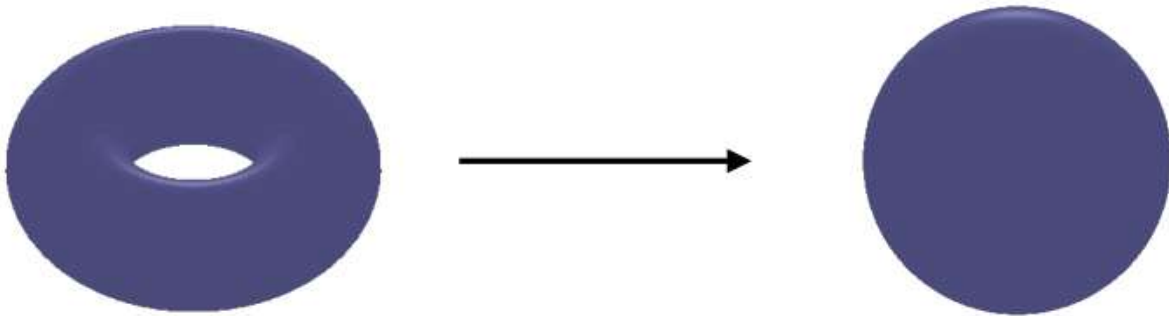
- Можем легко менять положение вершин и за счёт этого форму объекта
- Но другие преобразования уже затруднены

# Трансформации сеток

---



Сложно задавать преобразования даже для простых объектов.



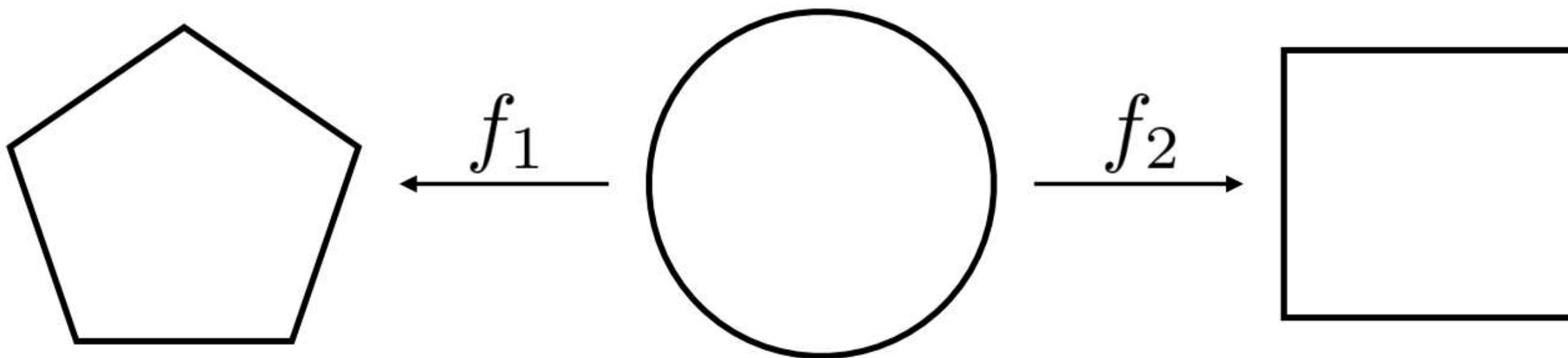
Менять топологию особенно сложно.

В общем случае сложно сделать методы, которые будут предсказывать объекты произвольной связанности и топологии (например, в задачах построения 3Д модели по изображению)



# Параметрические модели

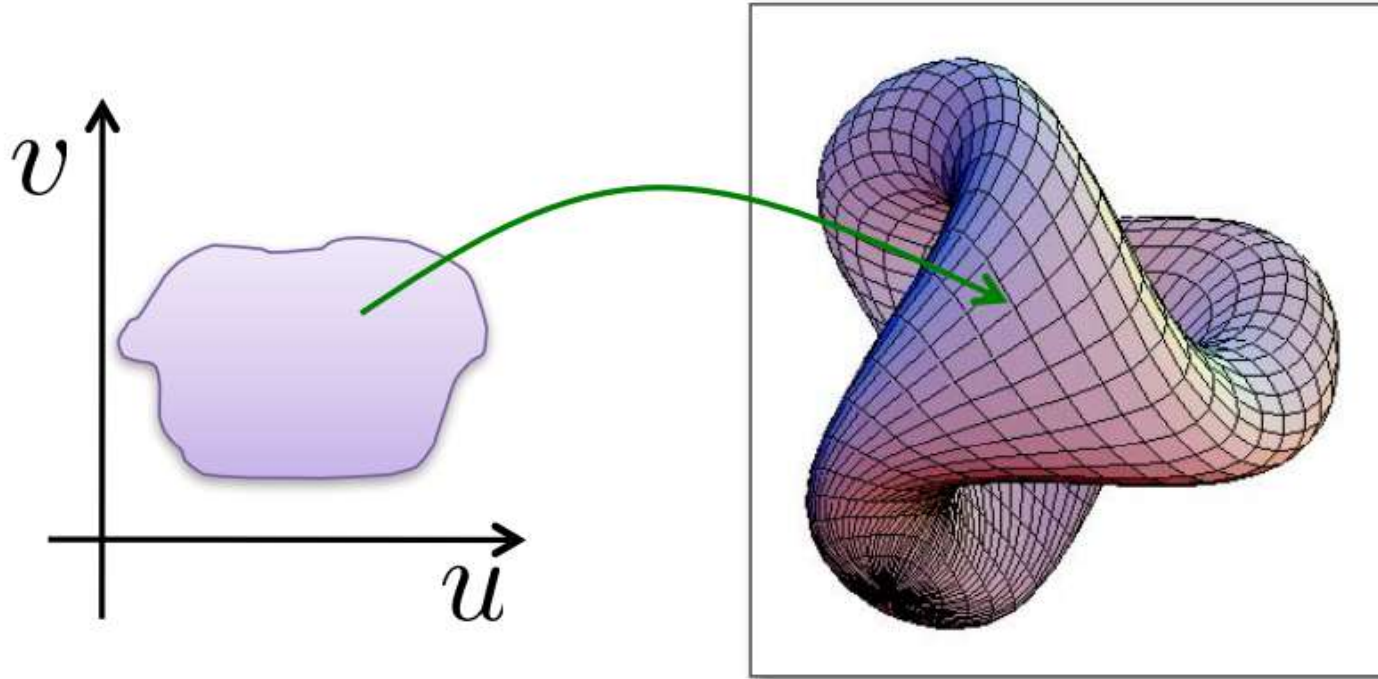
---



$$f(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^2; \quad \mathbf{u} \in \mathbb{S}^1$$

# Параметрические поверхности

---



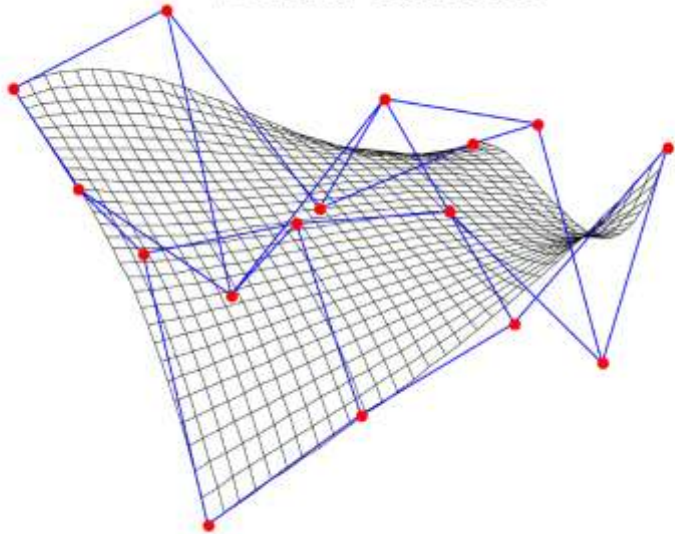
$$f(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^3; \mathbf{u} \in \mathcal{M}$$

- Непрерывная функция  $f$  по 2D многообразию  $\mathcal{M}$  задаёт поверхность
- Связанность / топология поверхности такая же, как и у  $\mathcal{M}$
- Мы отделяем дискретизацию поверхности (вершины/рёбра) от представления поверхности
- Функция  $f$  может быть конкретной аналитической функцией, семейством параметризованных функций, или даже нейросетью!

# Параметрические поверхности

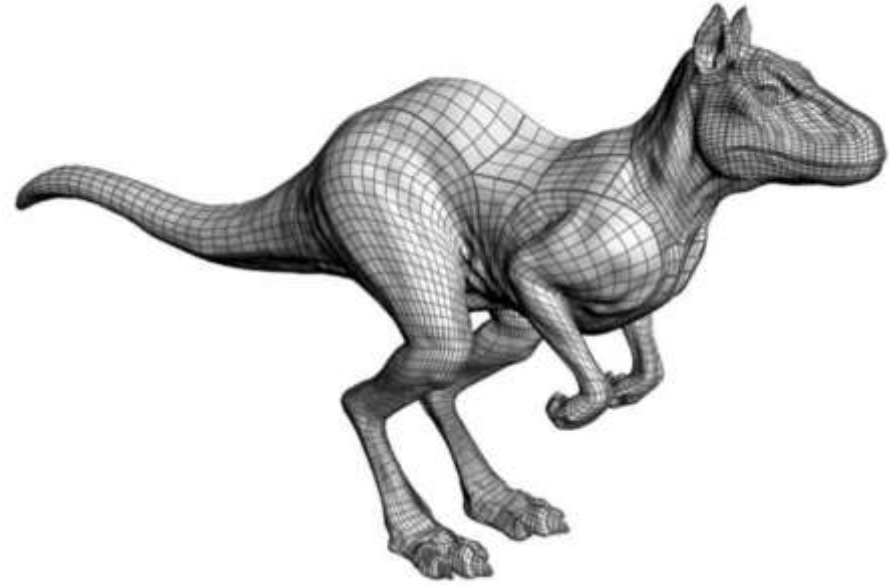
---

Bezier Surface



$$f((u, v)) = \sum_i^N \sum_j^M B_i^N(u) B_j^M(v) k_{i,j}$$

$$B_i^N(u) = \binom{N}{i} u^i (1-u)^{N-i}$$



- MxN контрольных точек задают поверхность
- Для визуализации обычно приходится применять тесселяцию, преобразование в сетку
- Часто используется для моделирования объектов

# Параметрические поверхности

---

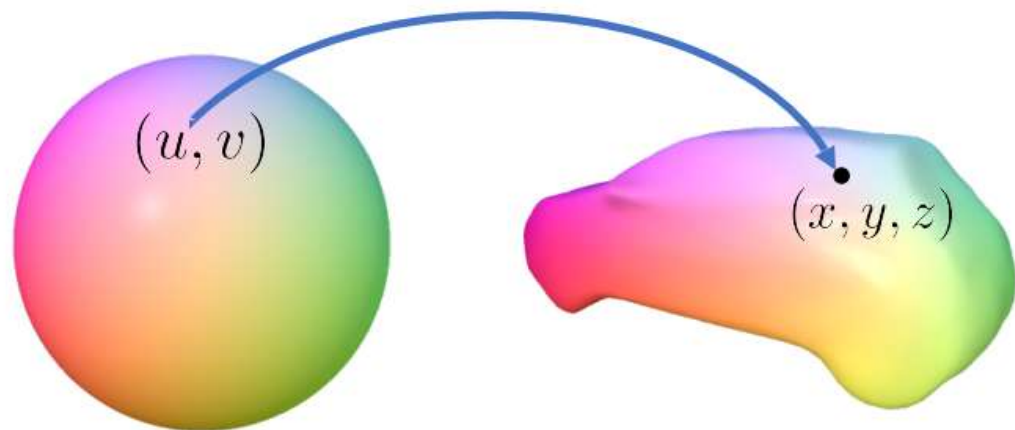


$$f_{\theta}(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^3; \quad \mathbf{u} \in \mathbb{S}^2$$

$f$  – нейросеть с параметрами  $\theta$

# Свойства параметрических поверхностей

---

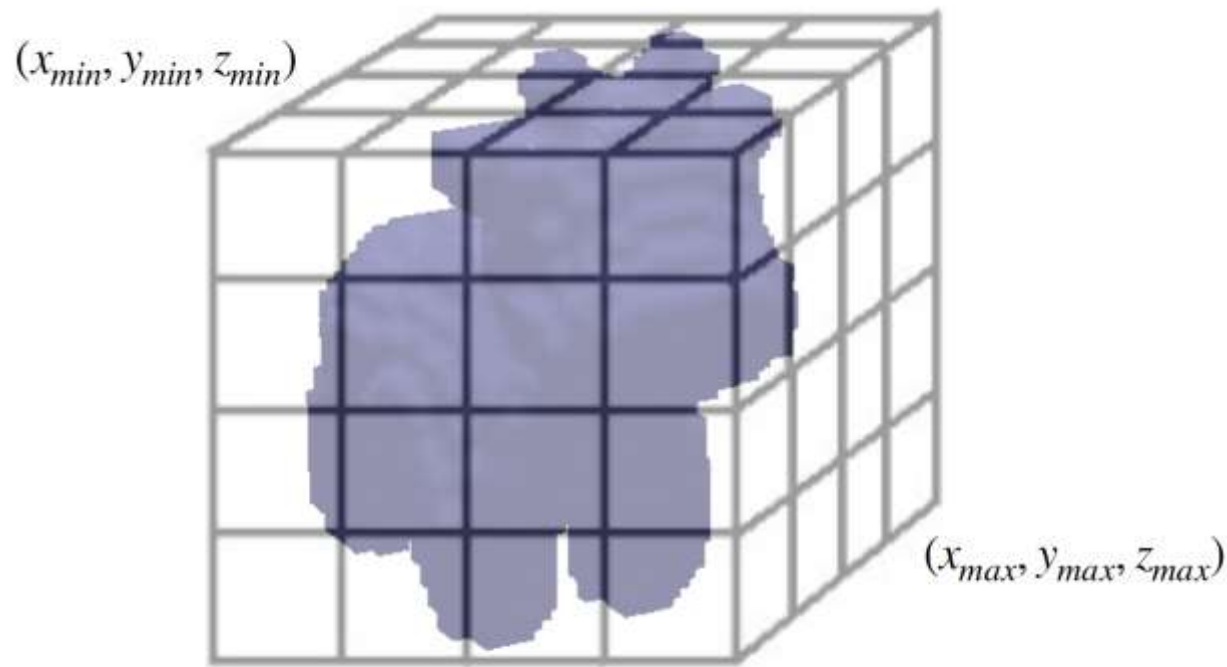


$$f(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^3; \mathbf{u} \in \mathcal{M}$$

- Удобно сэмплировать точки на поверхности
- Сложно судить о глобальной форме объекта
- Пр.: точка  $q$  вне или внутри объекта?
- Сложно визуализировать. Т.к. сложно посчитать пересечение луча из камеры с поверхностью объекта

# Воксельное (voxel) представление

---



$$V[x, y, z] \in [0, 1]$$

- Дискретизированное представление ограниченного кубиком пространство
- Матрица  $W \times H \times D$ , содержащая occupancy (или probability) каждой ячейки
- Матричная форма позволяет применять операции свёртки и адаптировать 2D NN модели к 3D объёму
- В чем будет существенная разница 2D и 3D данных и моделей?

# Воксельное представление

---



$32^3$



$64^3$



$128^3$



$256^3$



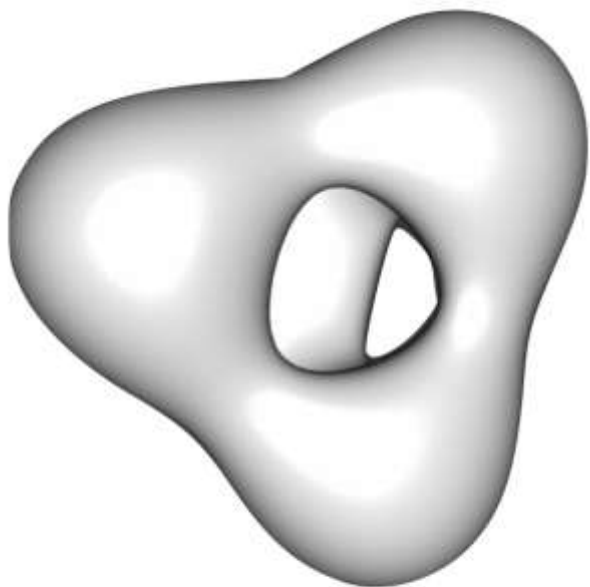
mesh

Вычислительно очень накладно масштабировать модель



# Неявные представления (Implicit Representations)

---

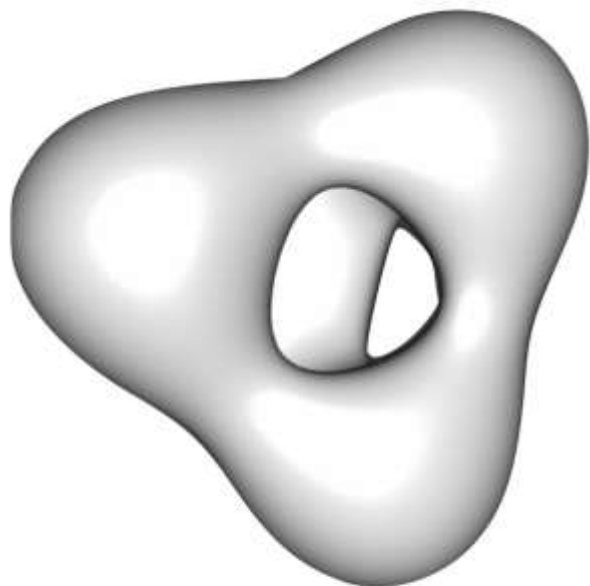


$$\{\mathbf{p} \mid f(\mathbf{p}) = 0\}$$

- Непрерывная функция  $f(\mathbf{p})$
- Переход через 0 соответствует поверхности
- Функция может задавать как угодно, хоть нейросетью, и дискретизироваться тоже!
- Легко отвечать на вопрос, лежит ли  $\mathbf{q}$  внутри или вне объекта
- Сложно судить о поверхности, её потребуется ещё «извлечь»

# Signed Distance Function (SDF)

---



$$\{\mathbf{p} \mid f(\mathbf{p}) = 0\}$$

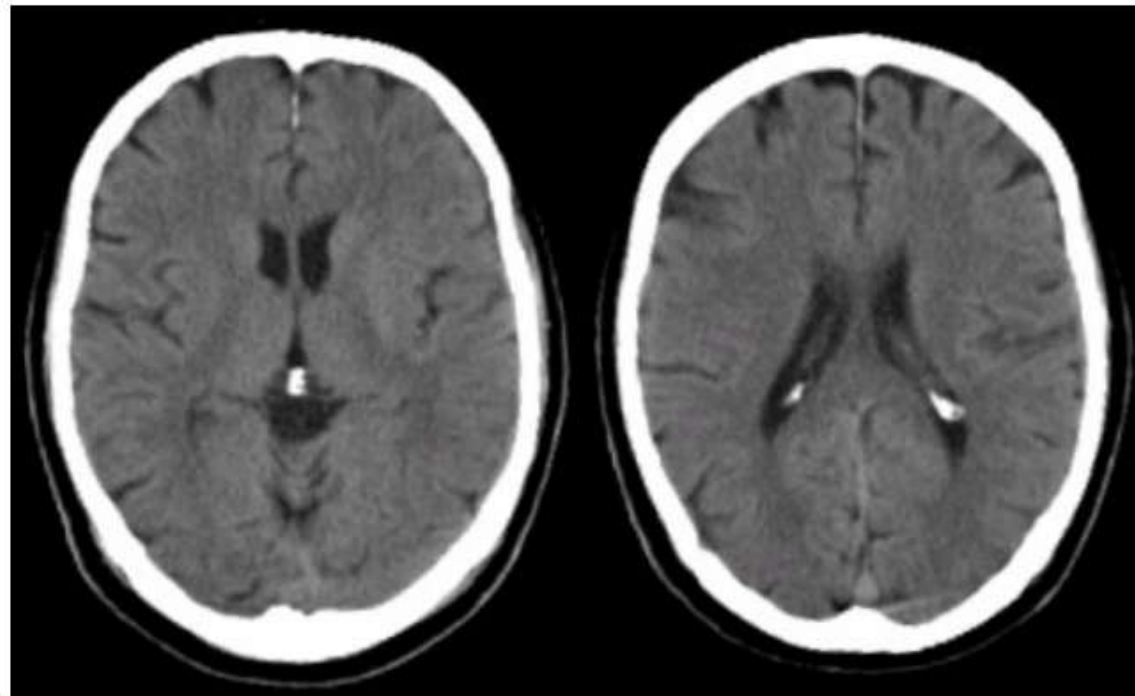
- $f(\mathbf{p})$  – задаёт расстояние до поверхности со знаком
- При её дискретизации можно точнее находить положение поверхности

# Поле плотности (density field)

---



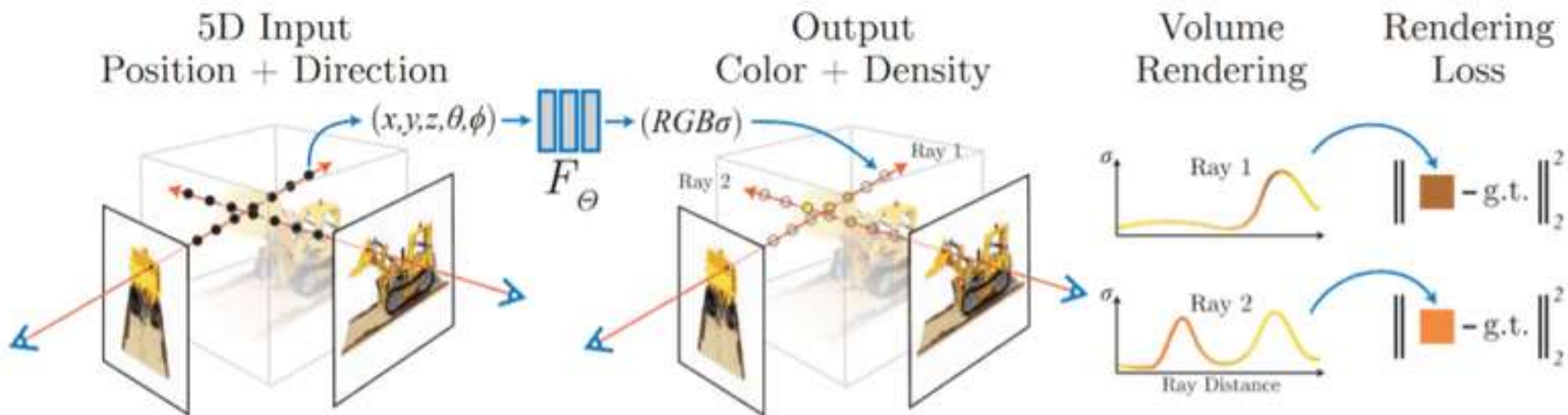
$$V[x, y, z] \in \mathbf{R}^+$$



$$f(\mathbf{p}) \in \mathbb{R}^+$$

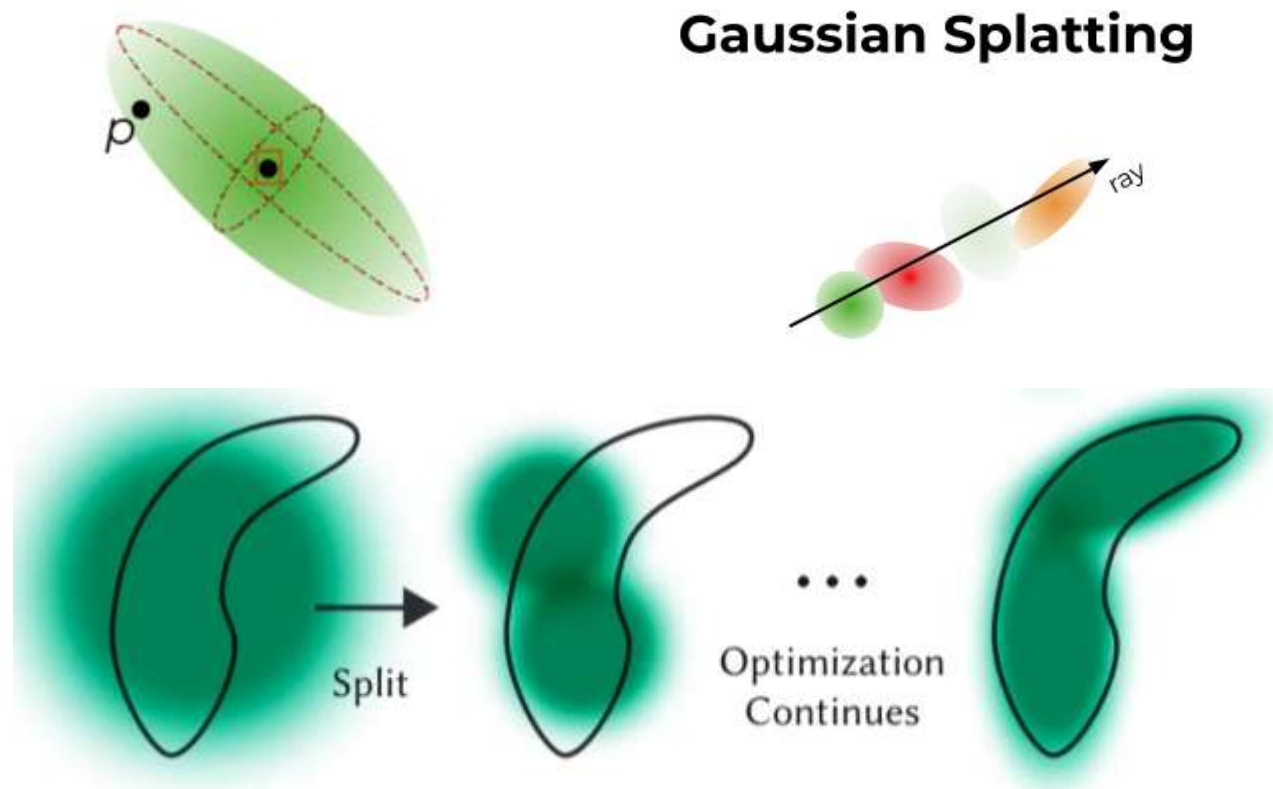
→ Нет в явном виде информации о поверхности, только об объёме

# NeRF (Neural Radiance Field)



- Для каждой точки пространства запишем RGB + density
- Закодируем это в виде нейросети
- Можем выучить такое представление по набору изображений

# 3D Gaussian Splats

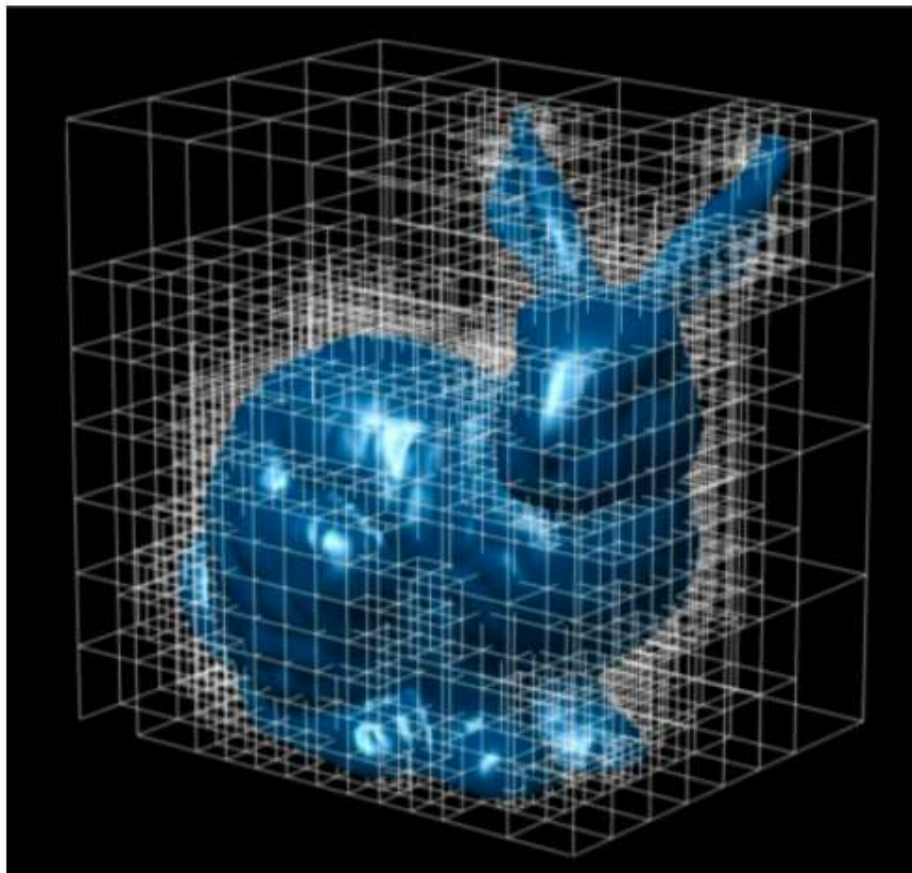


- Опишем сцену набором 3D gaussian splats
- Комбинация NeRF и Oriented Point Cloud



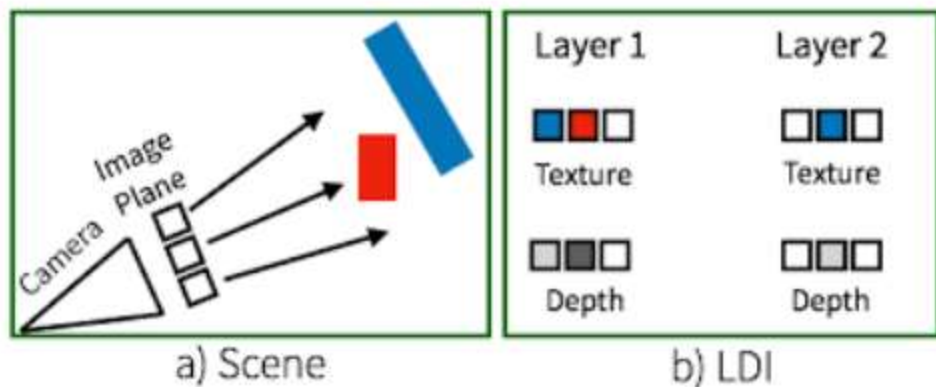
# Октодеревья (Octo-trees)

---



- Воксельная сетка с адаптивным разрешением
- Для задач рендеринга и хранения столь же эффективны, как явное представление поверхности
- Также эффективны, как объёмные представления, для запросов о принадлежности точки
- Но сложно модифицировать, сложно предсказывать разреженность и структуру

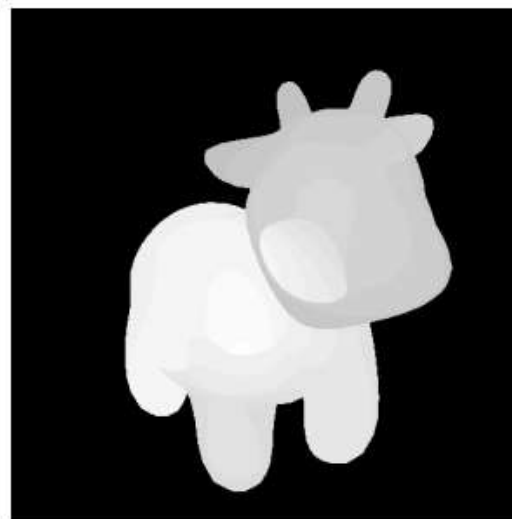
# Layered Depth Images



Surface



Layer 1



Layer 2



Layer 3

- Layer 1 = Видимая поверхность
- Layer N+1 = Глубина и цвет «следующей» точки поверхности вдоль луча
- Это расширение карт глубины для описания полной 3Д модели

# Multi Plane Image

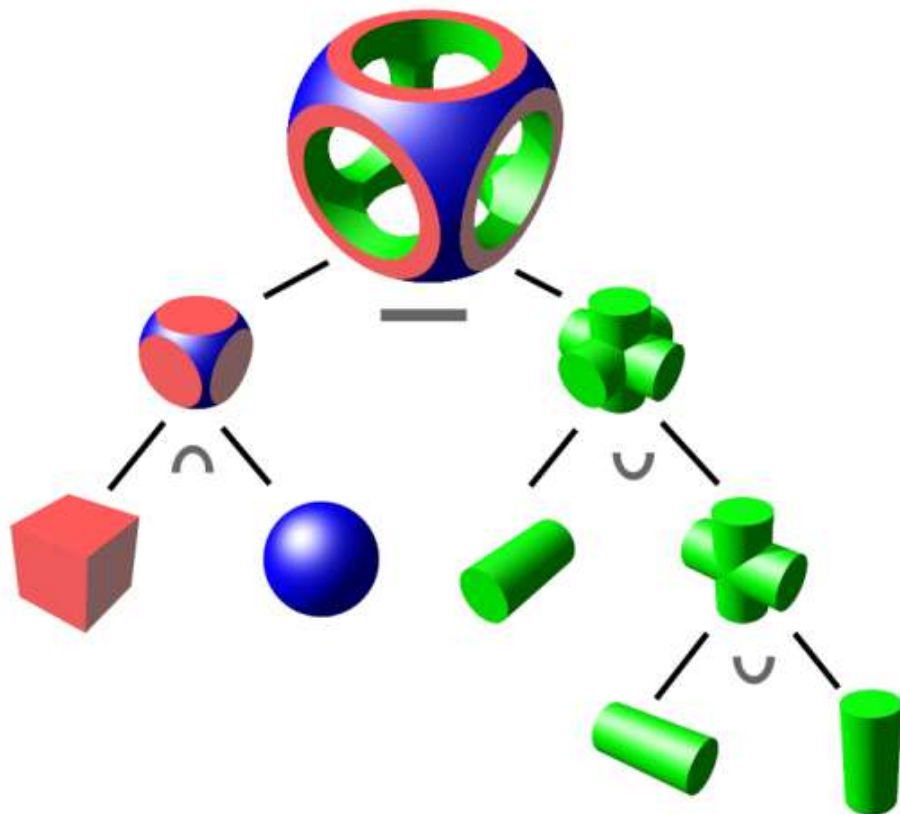
---



- Ещё одно «послойное» представление сцены
- Каждое изображение – точки на фиксированной глубине
- Очень эффективно для визуализации
- Ограниченное разрешение для «дальних» объектов
- Визуализация получается качественной для малых углов отклонения от исходного

# CSG (Constructive Solid Geometry)

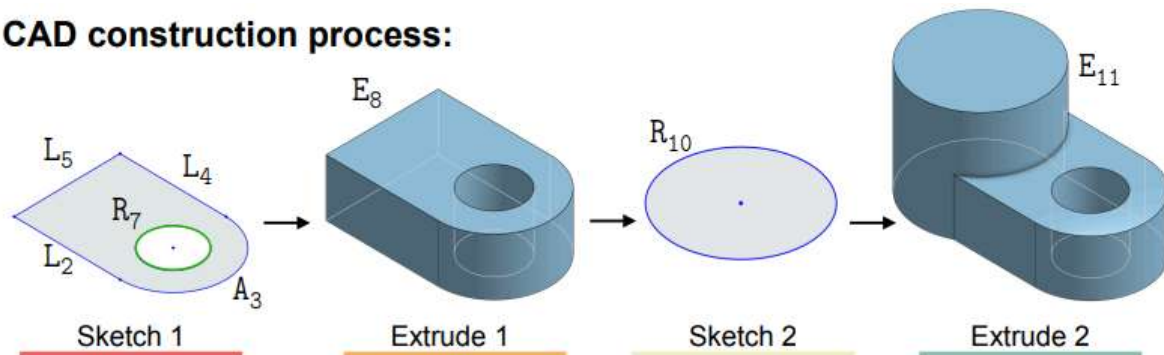
---



- Иерархическое представление
- Начинаем с простых 3Д объектов («примитивов»)
- Строим простыми операциями типа «объединение», «пересечение», «вычитание»

# CAD-модели

CAD construction process:



Parametrized command sequence:

$\langle \text{SOL} \rangle_1 : \emptyset$	$E_8 : (0, 0, 0, -2, -1, 0, 3,$
$L_2 : (2, 0)$	$1, 0, \text{New body, One-sided})$
$A_3 : (2, 2, \pi, 1)$	$\langle \text{SOL} \rangle_9 : \emptyset$
$L_4 : (0, 2)$	$R_{10} : (0, 0, 1.125)$
$L_5 : (0, 0)$	$E_{11} : (0, 0, 0, -2, 0, 0, 2.25,$
$\langle \text{SOL} \rangle_6 : \emptyset$	$2, 0, \text{Join, One-sided})$
$R_7 : (2, 1, 0.5)$	$\langle \text{EOS} \rangle_{12} : \emptyset$

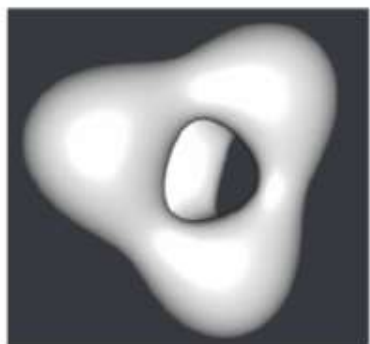
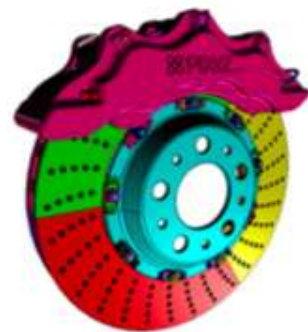
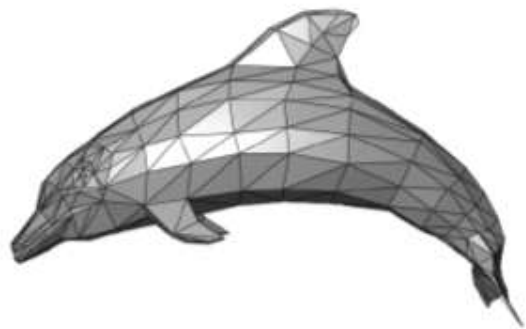
Commands	Parameters
$\langle \text{SOL} \rangle$	$\emptyset$
L (Line)	$x, y$ : line end-point
A (Arc)	$x, y$ : arc end-point $\alpha$ : sweep angle $f$ : counter-clockwise flag
R (Circle)	$x, y$ : center $r$ : radius
E (Extrude)	$\theta, \phi, \gamma$ : sketch plane orientation $p_x, p_y, p_z$ : sketch plane origin $s$ : scale of associated sketch profile $e_1, e_2$ : extrude distances toward both sides $b$ : boolean type, $u$ : extrude type
$\langle \text{EOS} \rangle$	$\emptyset$

- Loop - одна замкнутая кривая
- Sketch profile – набор loop в одной плоскости
- Extrude - "вытягивание" в 3D с каким-то взаимодействием с предыдущими 3D формами



# Преобразование между моделями

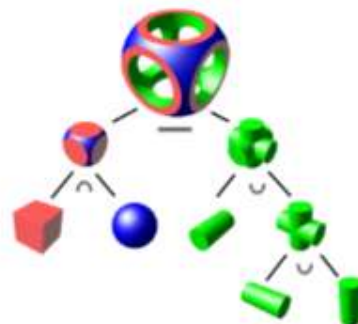
---



$$\{\mathbf{p} \mid f(\mathbf{p}) = 0\}$$



$$f(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^3$$



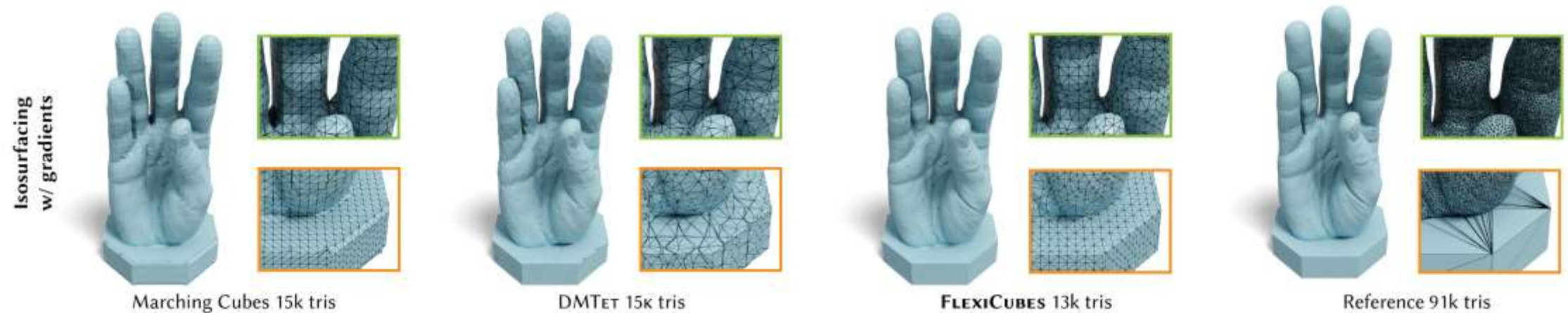
and many more ...

- Вопрос не простой, требуются отдельные методы для каждой пары моделей

# Конвертация между представлениями

---

- Иногда проще прямое преобразование представить как композицию ИЗВЕСТНЫХ



<https://research.nvidia.com/labs/toronto-ai/flexicubes/>

# Резюме лекции

---

- 3D зрение – это компьютерное зрение сразу в 3D пространстве
- Сейчас в него включают и генерацию 3D объектов и сцен, и визуализацию через нейросети, и обработку инженерных данных
- В отличие от 2D зрения, 3D моделей – огромное множество
- Они сильно отличаются по свойствам, и это существенно расширяет пространство методов и решений
- Мы рассмотрели основные типы 3D моделей