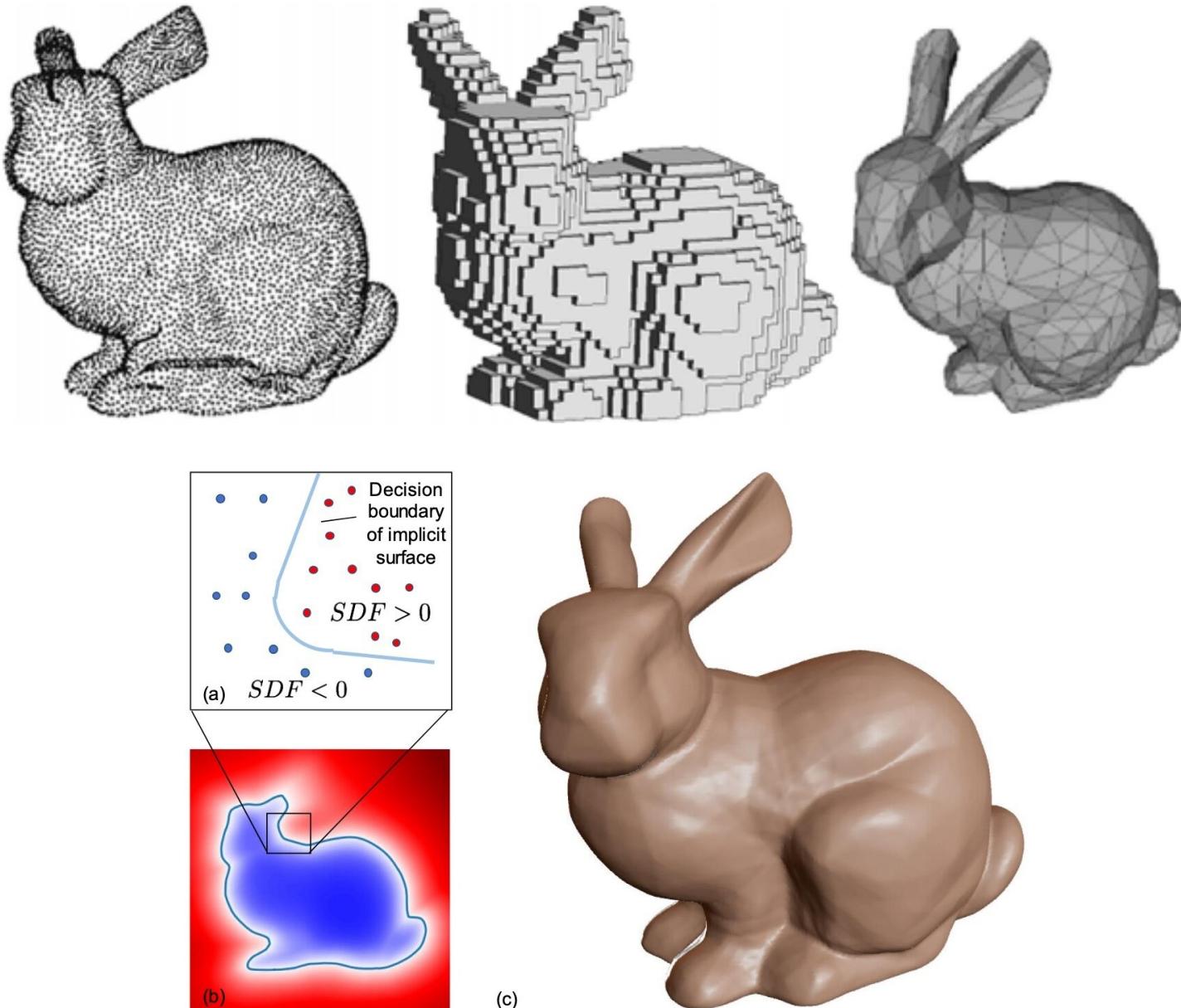


Глубокое обучение для обработки изображений

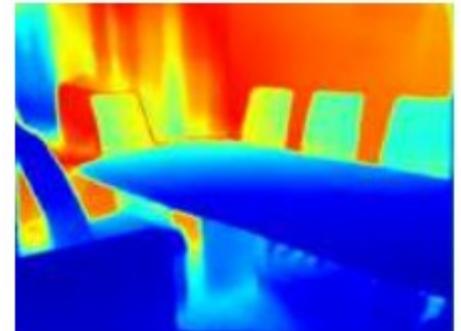
Лекция 7

Представление 3D объектов

- Воксели (Voxel)
- Облако точек (Point Cloud)
- Сетки (Meshes)
- Implicit surface
- Depth map
- Multi-view

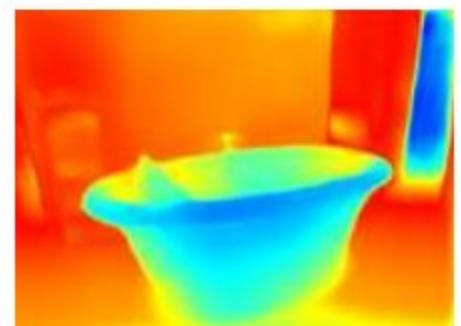
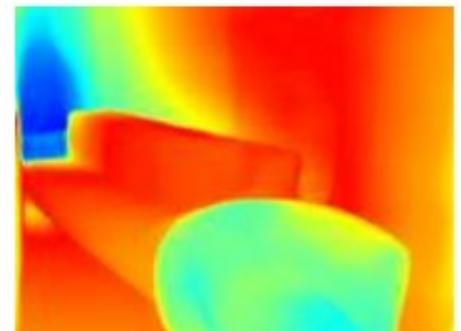


Depth map (карта глубины)

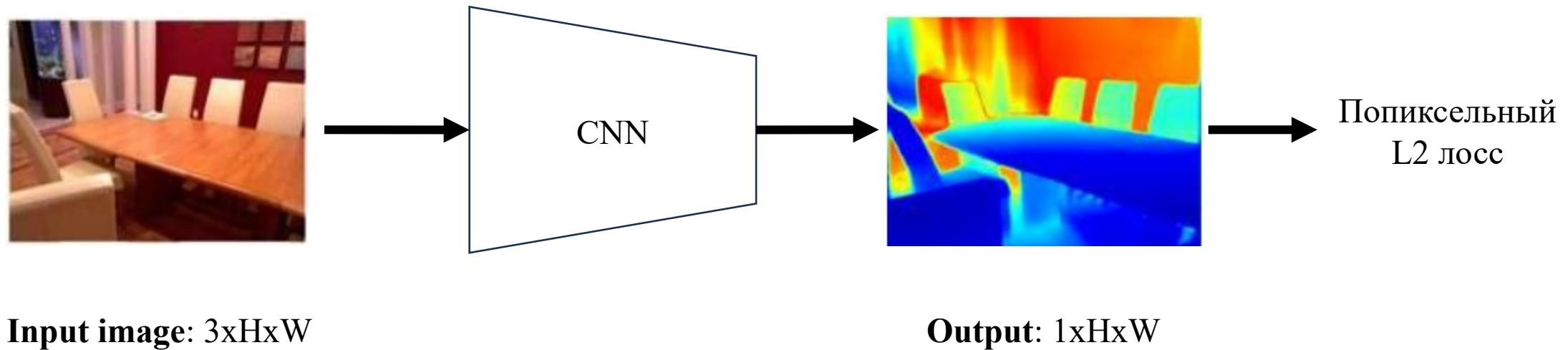


RGB: $3 \times H \times W$
RGB-D = RGB + Depth Map ($H \times W$)

Под глубиной может пониматься
расстояние от камеры до объекта
(Kinect, например, возвращает depth map)



Depth map (карта глубины)



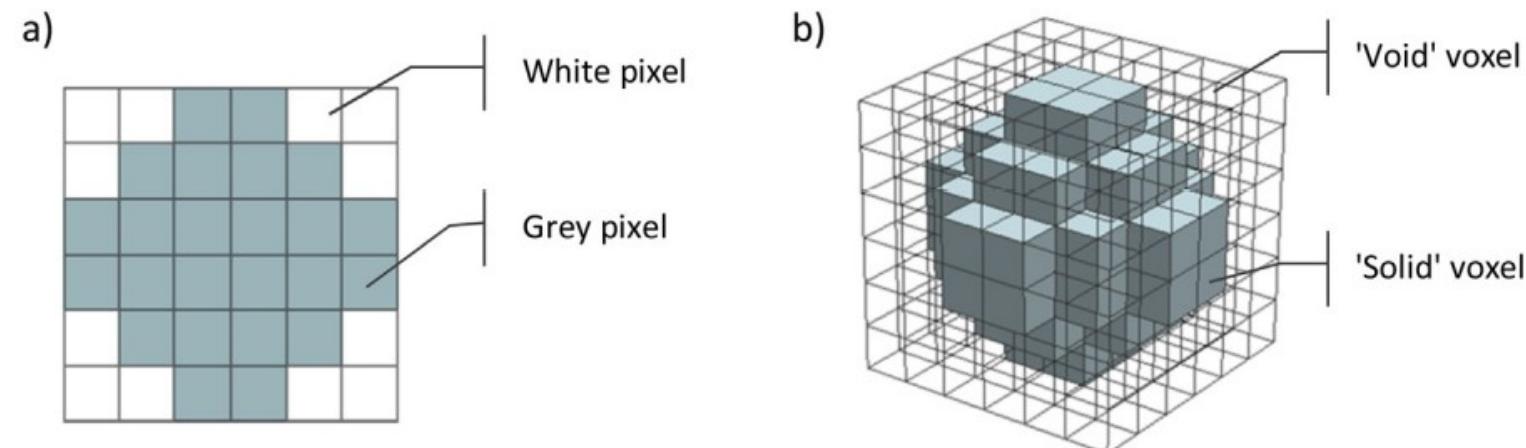
Воксели (Voxels)

Плюсы:

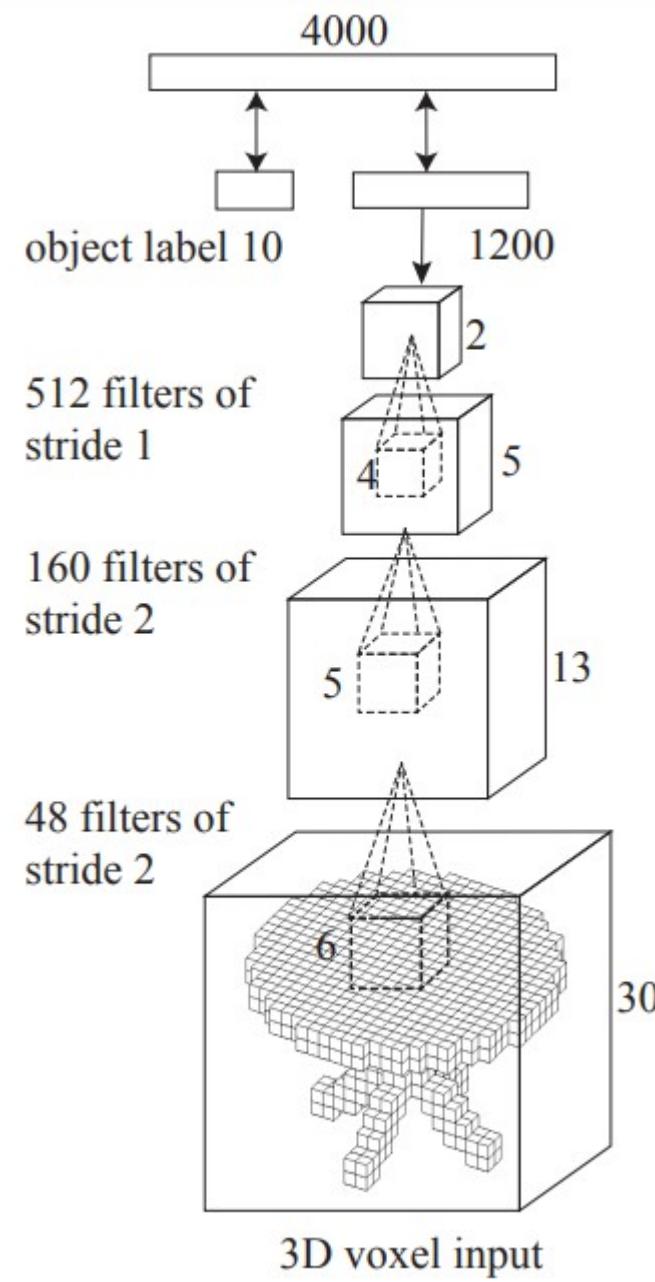
- Простая идея: переходим от двумерной сетки в трехмерной

Минусы:

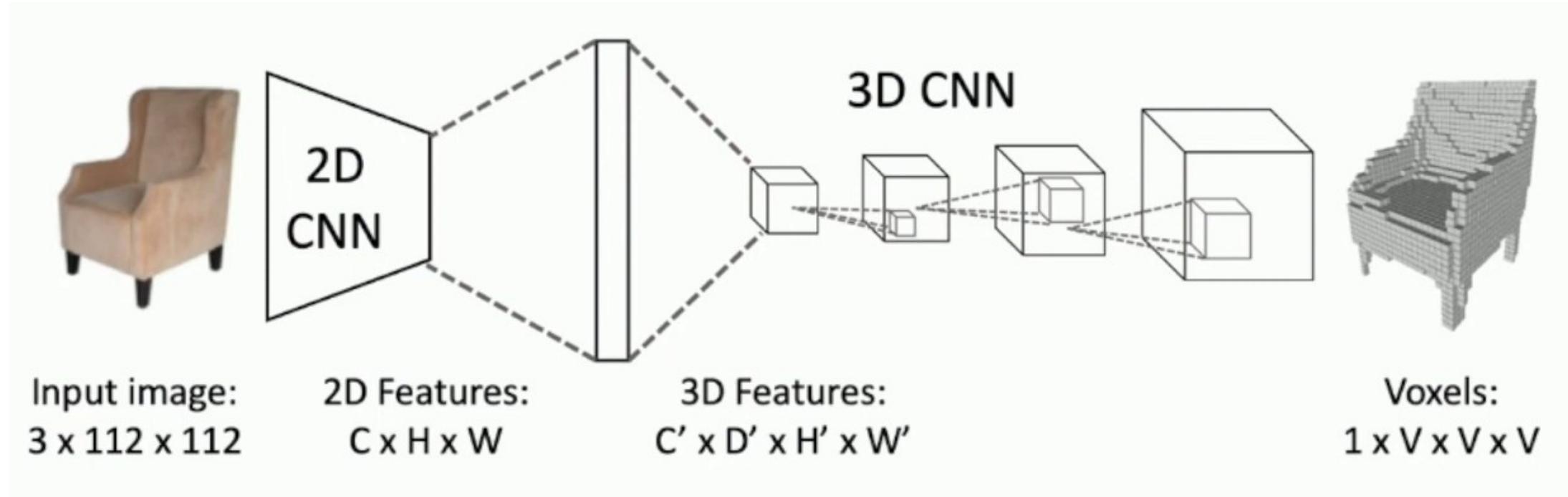
- Нужно большое разрешение, чтобы отобразить мелкие детали
- Много памяти требуется для хранения объекта



Воксели (Voxels)



Воксели (Voxels)



Conv3d

```
class torch.nn.Conv3d(in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1,  
groups=1, bias=True, padding_mode='zeros', device=None, dtype=None) \[source\]
```

Applies a 3D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, D, H, W) and output $(N, C_{out}, D_{out}, H_{out}, W_{out})$ can be precisely described as:

$$out(N_i, C_{out_j}) = bias(C_{out_j}) + \sum_{k=0}^{C_{in}-1} weight(C_{out_j}, k) \star input(N_i, k)$$

where \star is the valid 3D [cross-correlation](#) operator

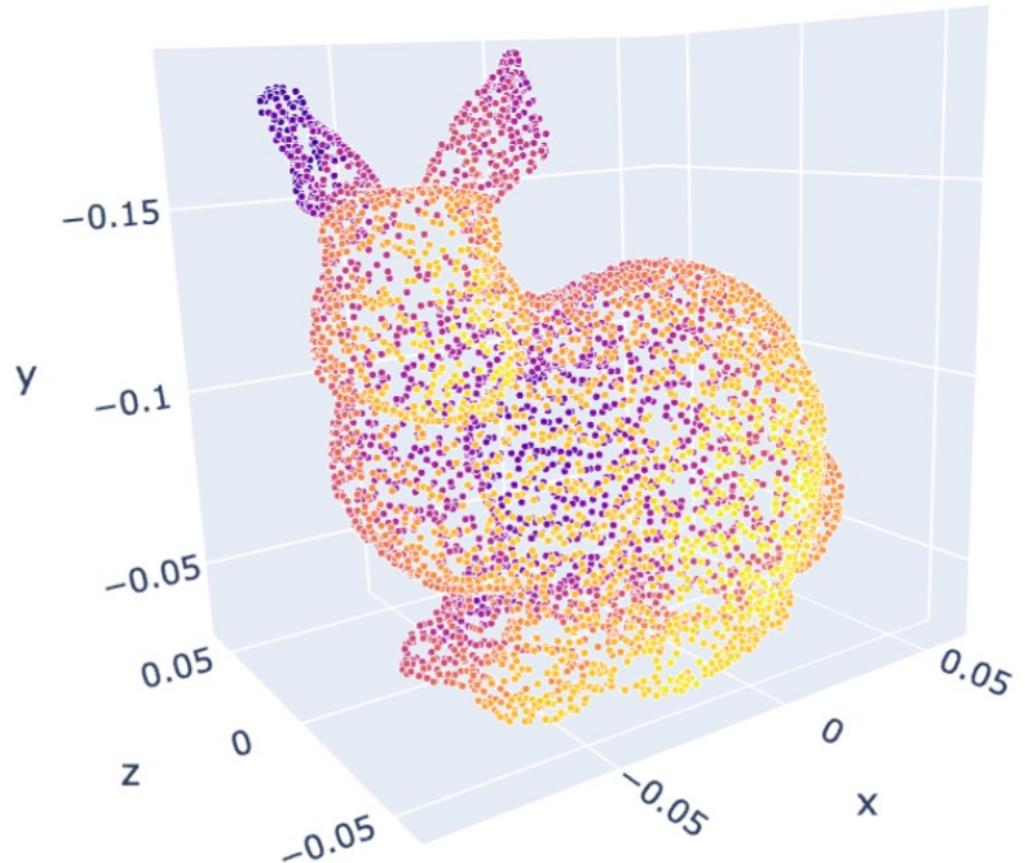
Облако точек (Point Cloud)

Плюсы:

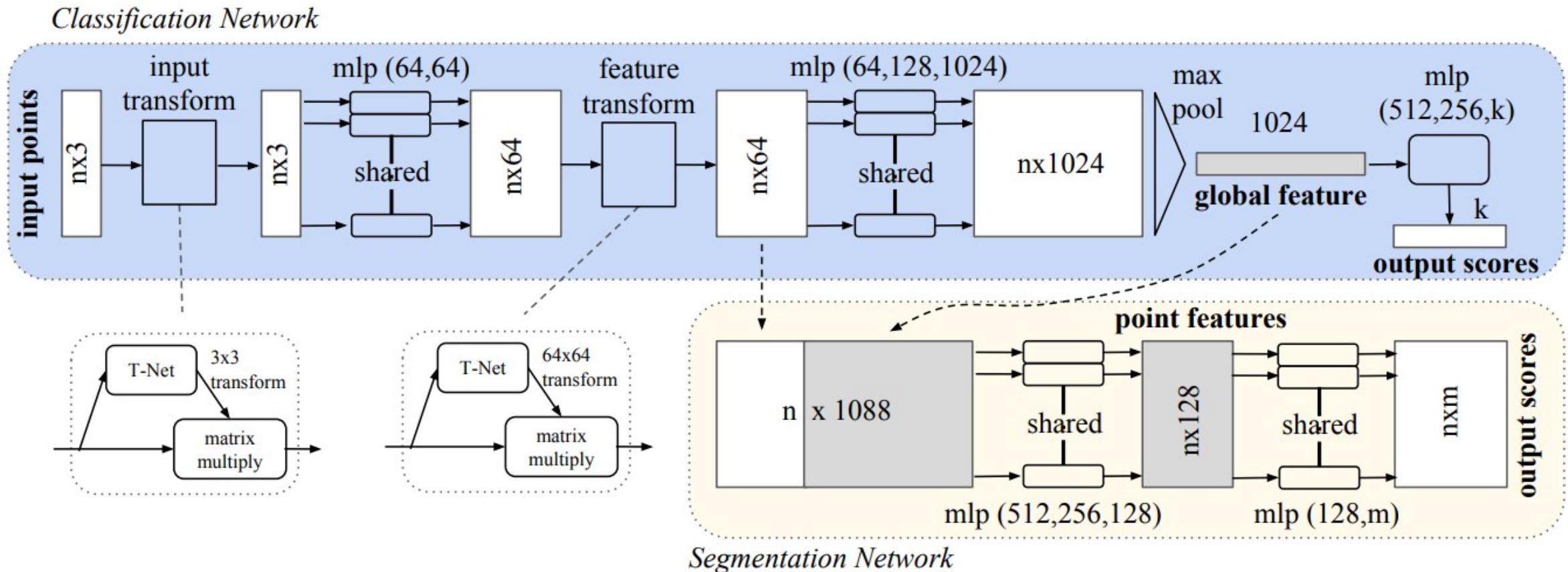
- Удобно изображать сложные структуры, формы
- Можно получать точки прямо с сенсоров (LiDAR, камеры и т.д.)

Минусы:

- Получаем просто набор точек (без сетки, структуры), нужна пост-обработка



Облако точек (Point Cloud)



Облако точек (Point Cloud)



Input

Reconstructed 3D point cloud

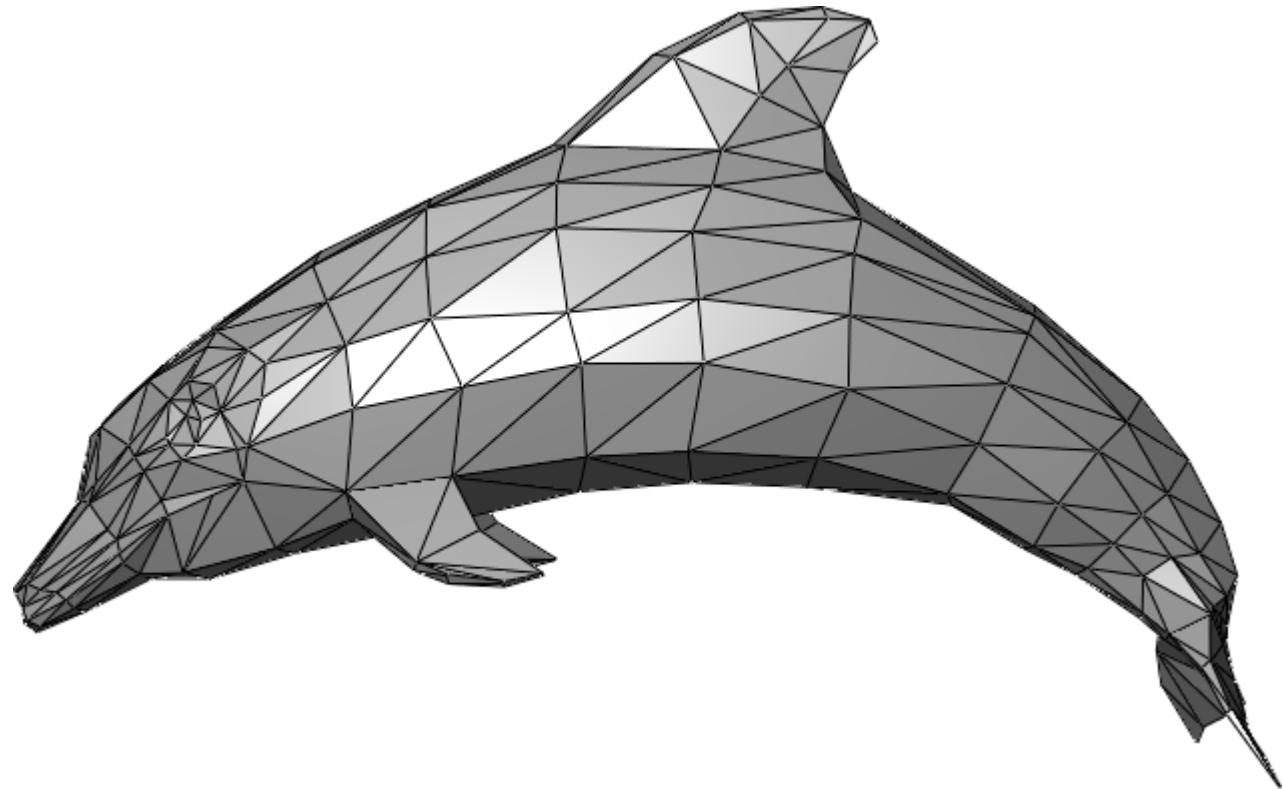
Полигональная сетка (polygon mesh)

Плюсы:

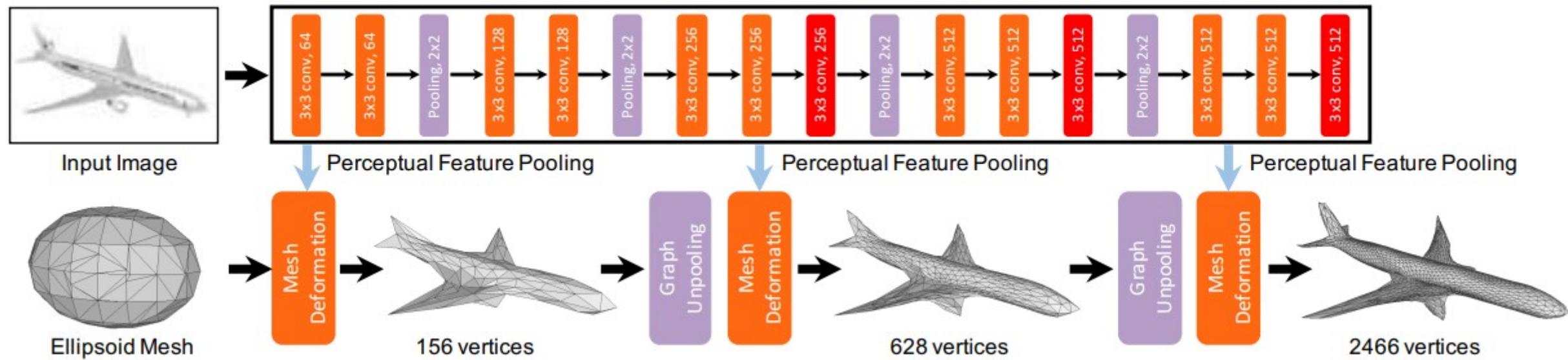
- Широко используется в графике
- Может адаптироваться под форму
(можно сгенерировать больше элементов
для мелких деталей)
- Может содержать данные в узлах (цвет,
координаты и т.д.)

Минусы:

- Необходимо придумывать новые
подходы к обработке с помощью
нейронных сетей



Полигональная сетка (polygon mesh)



Идея алгоритма:

- Начинаем с исходной сетки (эллипсоид)
- Сеть предсказывает, на сколько нужно сместить узлы сетки
- Повторить

Полигональная сетка: графовые нейронные сети

Новое значение в узле (feature) может зависеть от значений соседних узлов.

Подобные структуры удобно описывать с помощью **графов** и затем обрабатывать с помощью **графовых нейронных сетей** (Graph Neural Network, GNN).

