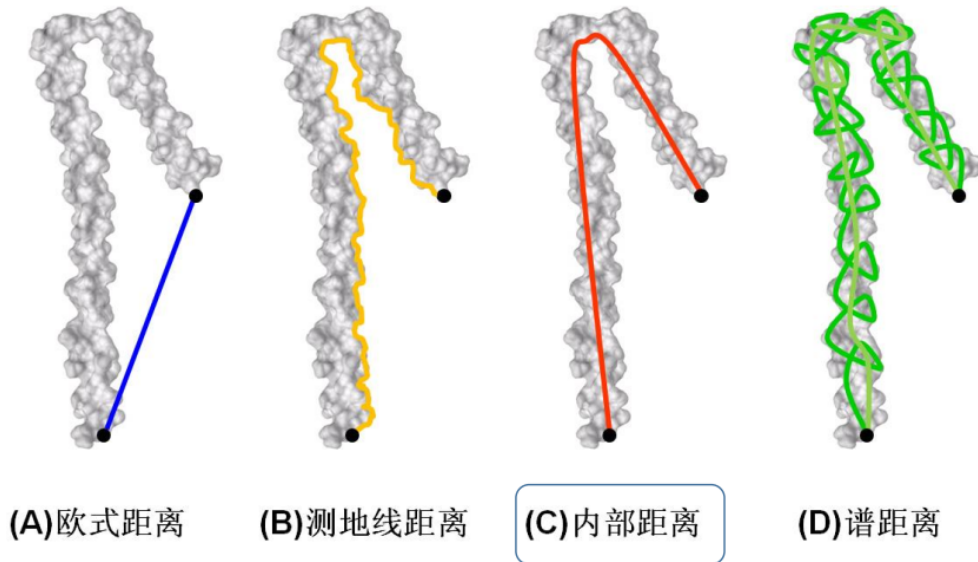


1.特征

设计特征的关键：易于计算，能够很好区分不同物体，对平移、旋转等刚性变换/非刚性形变有不变性。

1.1 距离分布

物体内部两点之间的各种距离的分布，有刚性变换不变性，测地线距离（Geodesic Distance）、内部距离和谱距离对非刚性变换也有不变性。



1.2 球谐函数描述 Spherical Harmonic Descriptor (SHD)

利用傅里叶展开实现旋转不变性

1.3 光场描述 Light Field Descriptor (LFD)

用一个物体在足够多的视角的投影图描述物体，有刚性变换不变性。

物体匹配：对视角进行匹配，度量相似性

1.4 Spin image

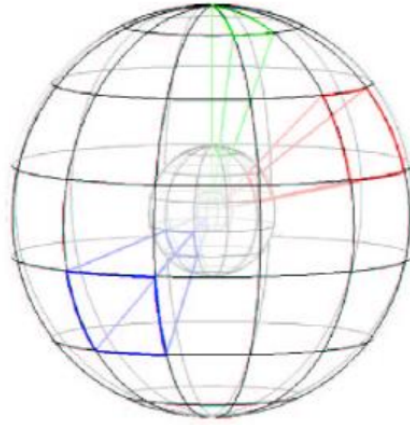
物体在圆柱坐标系下的投影，有刚性变换不变性

1.5 形状上下文

将所有采样点到锚点的偏置在极对数坐标系（log-polar coordinates）下构建直方图，一种统计信息。

• Extension to 3D

- Pick random points on surface
- For each point, compute and histogram offsets
- For each point, compute offsets
- Histogram the offset vectors



1.6 曲率 (局部信息)

1.7 交界相关

Visible Internal Volume (VIV) : 从某点可见的形状范围

VIV在交界处变化巨大

1.8 Bilateral Maps

以两点来定义感兴趣的区域ROI, 常用球谐函数/测地线距离等

2.常见问题

2.1 曲面参数化

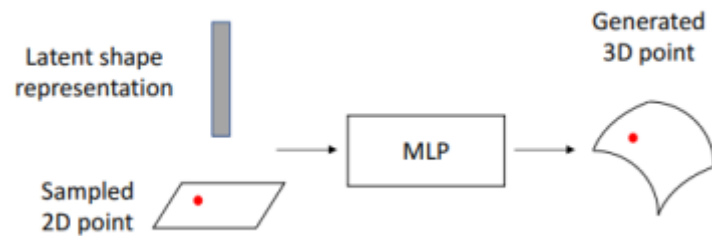
核心：找到3D曲面和2D平面的映射关系（双射）

应用：纹理映射，曲面重参数化（Remeshing），曲面滤波（和2D图片滤波类似，需要定义对应的像素、邻域，利用法向进行更新）

方法：传统方法包括PCA、奇异值分解等降维方法

重要结论：3D曲面位于2D流形上，所以可以在2D平面上sample点，构建和3D曲面的双射。

AtlasNet：利用提取的特征，将2D平面上的采样点映射到3D表面上的采样点---一个feature可以构建一个物体的局部（patch）---拼接实现物体三维重建

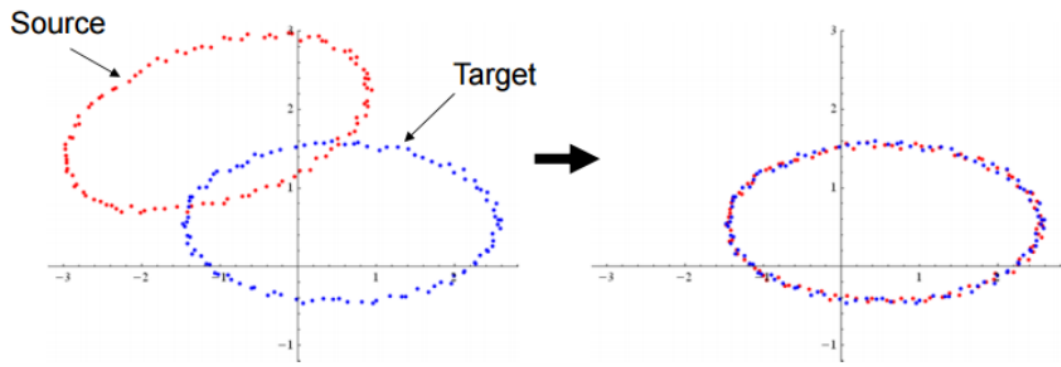


FoldingNet: 利用提取的特征，做2次折叠（一次2D-3D，一次3D-结果），将2D grid变成三维物体

Input	2D grid	1st folding	2nd folding

2.2 3D对齐、配准

给定两个物体P和Q，找到一个刚体变换（旋转+平移），让两者能够尽可能接近



12

基本流程：提取特征---配对---计算变换

提取特征：有神经网络之后，可以用神经网络方便提取

配对：

- 利用几何一致性约束（对应点对的距离应该是接近的，RMSE loss）
- Attention和Transformer（神经网络学习对应关系）
- 在数据集上采样一些点对进行匹配训练

计算变换：

- PCA
- SVD（更鲁棒）
- ICP（迭代算法，综合PCA和SVD的好处）
- RANSAC等

2.3 3D查询

3D Shape Retrieval：需要定义距离--EMD（Earth Mover's Distance）、Chamfer Distance（CD）。

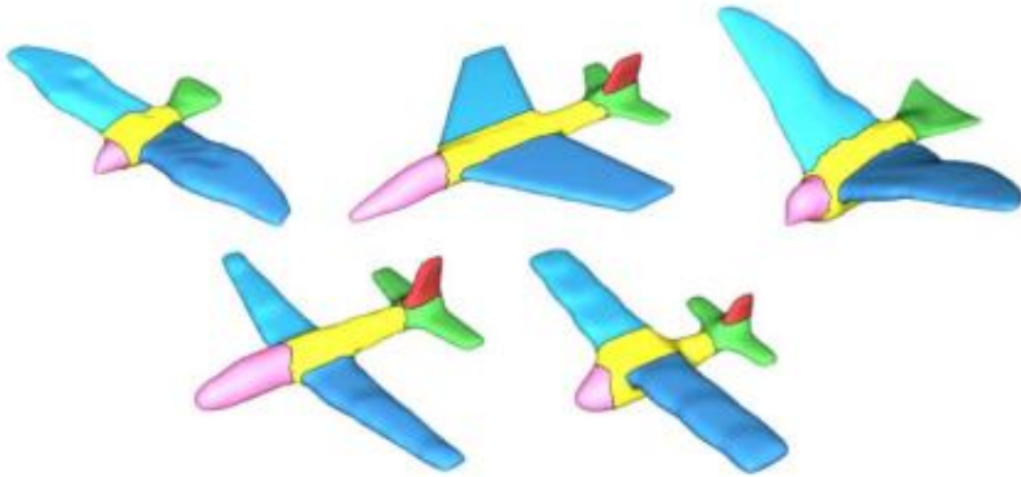
评价指标：Precision, Recall。

方法：特征提取+embedding（无监督学习）+retrieval。

度量指标：Cumulative Gain（CG）---Normalized Discounted Cumulative Gain（NDCG）。CG度量整体搜索质量，DCG对每一条增加对数权重，NDCG进行归一化。

3D分类：使用分类器进行有监督学习

2.4 3D物体分割



方法:

- K-means聚类
- RANSAC算法/Region Growing算法
- 根据几何信息设计loss, 然后用图割算法求解

Data term: The cost of assigning a label to f . **Smooth term:** Make the border smooth.

$$E(\bar{x}) = \sum_{f \in F} e_1(f, x_f) + \lambda \sum_{\{f, g\} \in N} e_2(x_f, x_g)$$

$$e_1(f, x_p) = -\log(P(f|x_f) + \varepsilon)$$

$$e_2(x_f, x_g) = \begin{cases} -\log(\theta(f, g)/\pi) & x_f \neq x_g \\ 0 & x_f = x_g \end{cases}$$

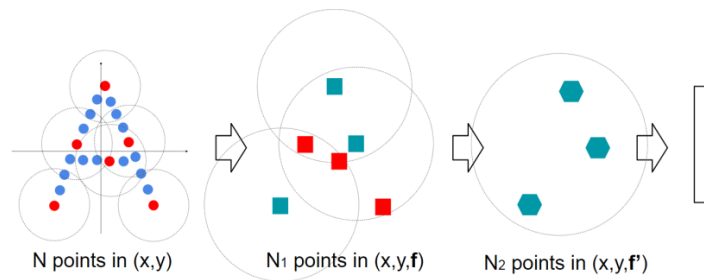
3.3D深度学习

3.1 基本框架

体素: 3D CNN卷积进行特征学习。可以做物体分类, 目标检测等。要精确的话开销太大

点云: PointNet---MLP网络, 利用正交损失函数保证点云旋转不变性。PointNet++---在PointNet基础上改进, 引入FPS (Farthest Point Sampling, 最远点采样) 和分组机制, 可以层次化学习特征 (在点云密度较高的地方看得更精细, 在点云密度较低的地方感受野要大一点)。可以做物体分类, 语义分割等任务。

PointNet++



1. Larger receptive field in higher layers ✓
2. Less points in higher layers (more scalable) ✓
3. Weight sharing ✓
4. Translation invariance (local coordinates in local regions) ✓

曲面：用人工特征作为目标特征，然后用auto encoder学习

多视角：多图单独用CNN提取特征，再聚合得到结果。可以利用多视角信息做retrieval

Comparison

- Which representation is the best?

	Advantage	Disadvantage
Voxel-based	Can represent both the surface and inner information	Aligned before voxelization; Low resolution
PointCloud-based	Avoid the combinatorial irregularities and complexities of meshes; easier to learn from	Invariant to permutations of points and rigid motion
Surface-based	Robust to isometric deformation of 3D shapes	Constrained to smooth manifold meshes
View-based	Can directly exploit the image-based CNNs; can handle high resolution inputs	Unclear how to determine the number

3.2 语义分割，实例分割

语义分割：给每个点云一个label，softmax分类问题

实例分割/目标检测：获取每个实例的位置，需要基于RPN/包围盒方法

二者互相促进：相同语义的临近点应该在一个实例内，相同实例的点语义也应该一样

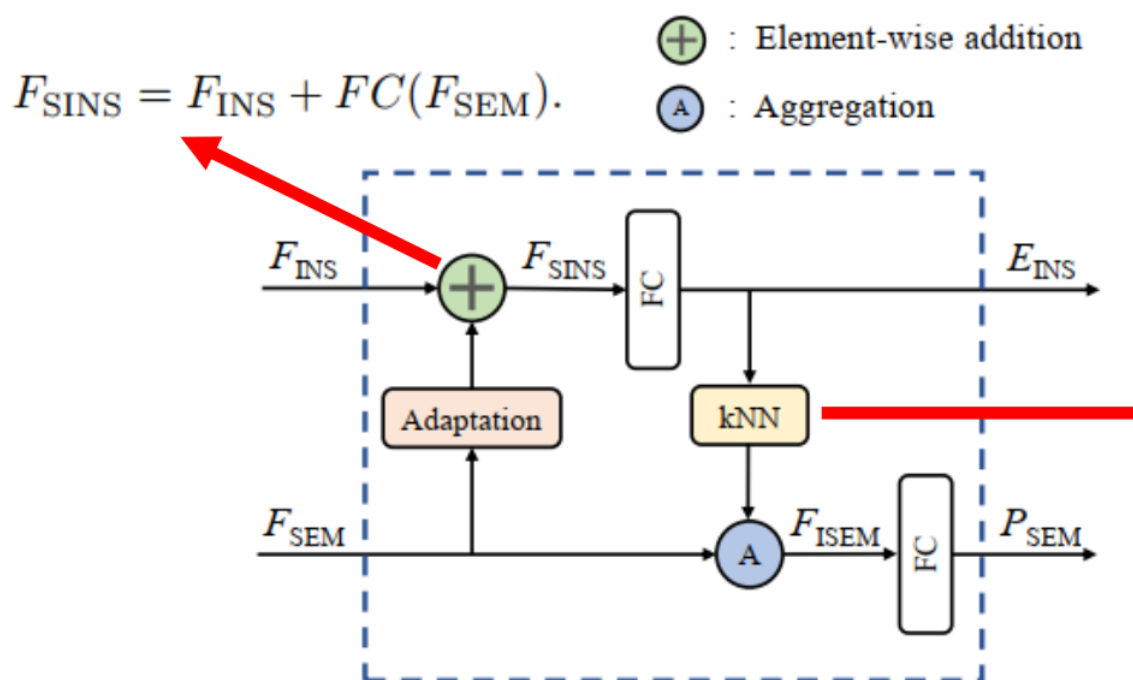
方法：ASIS:

分为两个分支：语义分割和实例分割

语义分割得到的信息通过全连接层融合进入实例分割分支

实例分割得到的结果做KNN，保证每个点的k近邻都在同一个实例内，然后以这k个近邻的语义信息的众数作为这个点的语义，融合进入语义分割分支

• ASIS - module



3.4 点云补全和三维重建

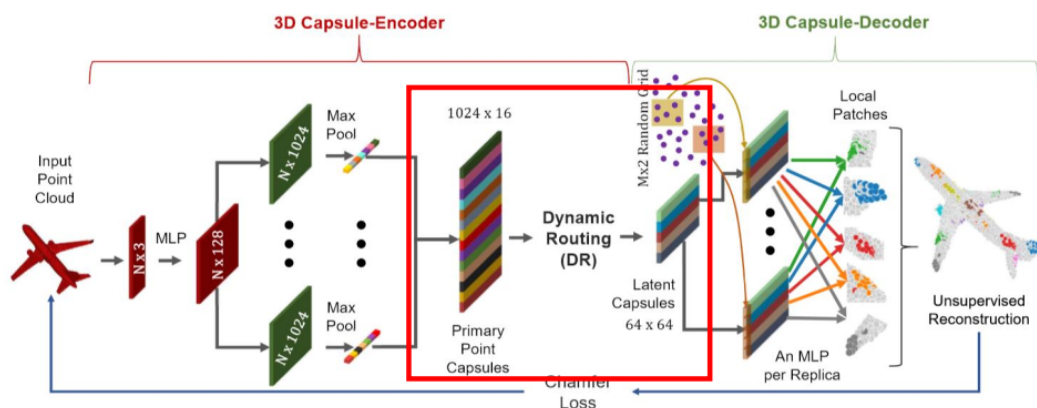
三维重建包括给定稀疏、残缺的点云等，或者是特征，生成完整的三维模型。

有几种常见的范式：

1. 将点云转化成体素，基于体素信息，使用3D CNN进行重建，代表是GRNet

2. 从点云中提取特征，然后基于特征去生成对应的三维模型。生成这些三维模型有以下思路：

- 利用三维物体都是二维流形这一假设，通过折叠的方法生成三维模型，代表是FoldingNet, SA-Net
- 可以利用**由粗到细**的思路。可以首先得到较为粗糙的三维点云，然后利用折叠（同FoldingNet）、反卷积（常见的上采样方法）、Self Attention（首先复制点云、然后利用Self Attention来让新的点云分布较为均匀）等方法进一步完善点云信息。
- 可以利用**层次化，局部化**的思路。层次化思路参考TopNet，逐层在不同的特征上精细化点云，最后合并在一起。局部化的思路参考3D Point Capsule Networks，在提取特征阶段将点云分成若干patch分别提取特征，然后在重建阶段每个patch分别重建最后结合。



3.作为点云补全问题，可以转化成形变问题：将输入点云的每个点进行一定的位移，让位移总距离最小，得到最终点云。PMP-Net引入了RNN网络来计算和存储每个点的移动（每个点的移动是时间序列），在损失函数里限制总位移距离。

4.隐式表达

和点云，体素，面片不同，类似一个模型数据库，很难直接感知全局信息，但是可以通过运算/进入神经网络来查询局部信息。

4.1 Occupancy

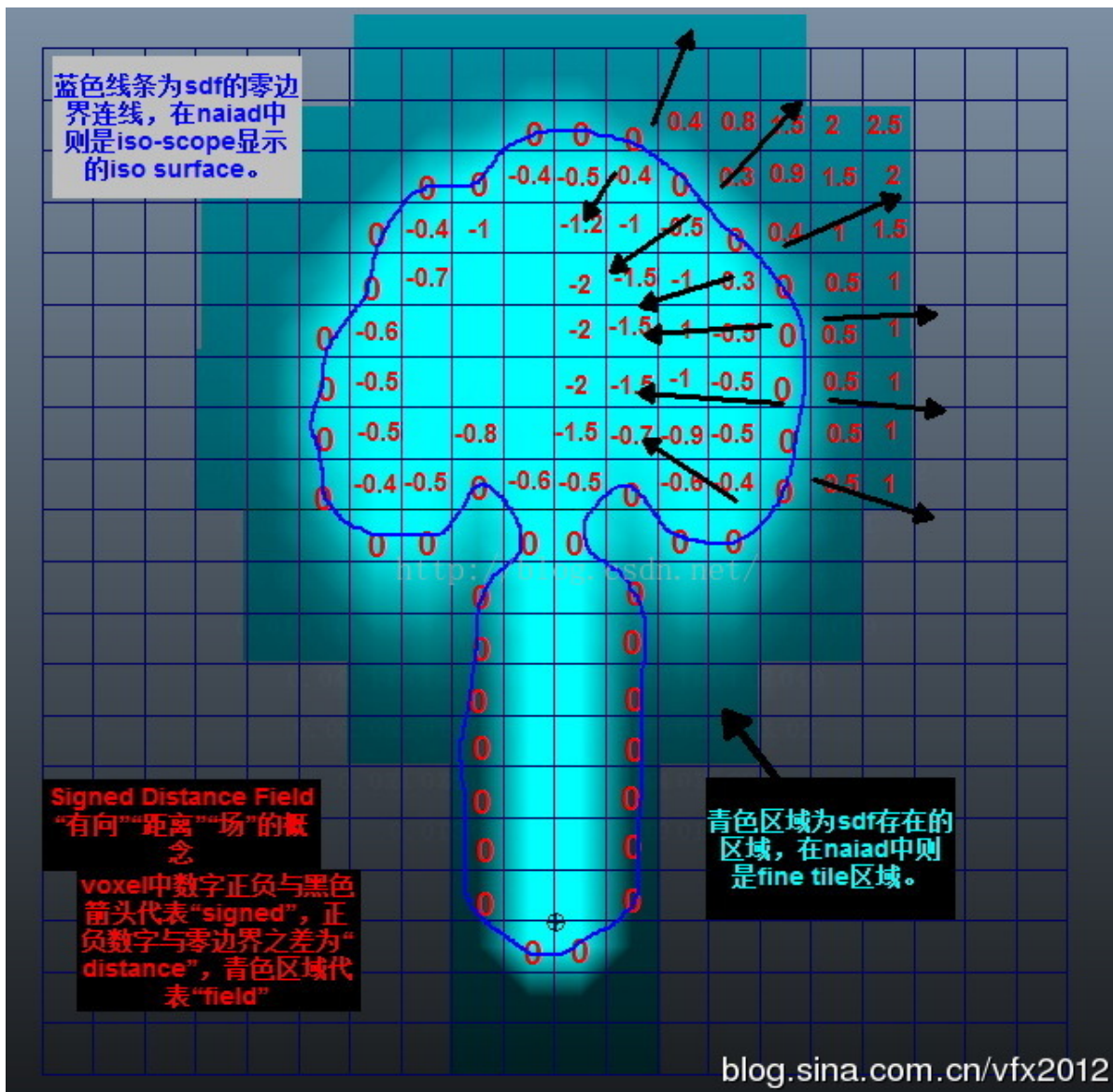
Network: a function that takes an **observation** $\mathbf{x} \in \mathcal{X}$ as input and has a function from $\mathbf{p} \in \mathbb{R}^3$ to \mathbb{R} as output can be equivalently described by a function that takes a pair $(\mathbf{p}, \mathbf{x}) \in \mathbb{R}^3 \times \mathcal{X}$ as input and **outputs a real number indicating whether the point is outside or inside the shape.**

$$f_{\theta} : \mathbb{R}^3 \times \mathcal{X} \rightarrow [0, 1]$$

输入：观测 \mathbf{x} ，点 \mathbf{p} ；输出：点 \mathbf{p} 位于物体中的概率（0到1之间的实数）

4.2 有向距离场（SDF）

到物体面的最近距离，变种是TSDF，做个Truncate。可以用Marching Cube进行重建。

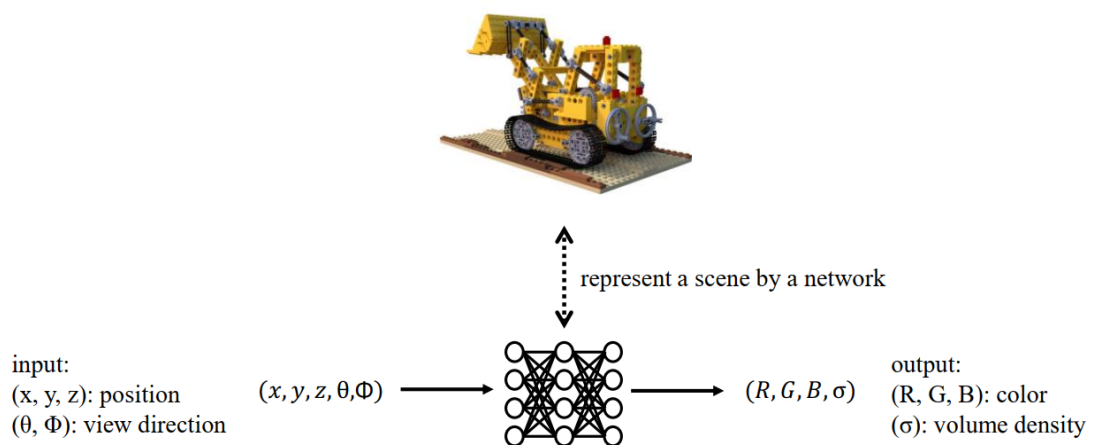


4.3 NERF

用一个全连接网络来表示一个场景。

输入：点和视角

输出：RGB值



渲染方法：利用Volume Rendering的方法，每条光线的RGB=路径物体颜色的加权积分

- use classical volume rendering techniques to accumulate those colors and densities into a 2D image
- for each camera ray $r(t) = o + td$, with near and far bounds t_n and t_f
the expected color $C(r) = \int_{t_n}^{t_f} T(t) \sigma(r(t)) c(r(t), d) dt$, where $T(t) = \exp(-\int_{t_n}^t \sigma(r(s)) ds)$
density acts like a differential opacity controlling how much radiance is accumulated
the $T(t)$ denotes the accumulated transmittance along the ray from t_n to t

5.数据集、指标等

物体方面：ModelNet、**ShapeNet**、ObjectNet3D、Thing10K、A Large Dataset of Object Scans

场景方面：NYU Depth Dataset（室内场景，只有RGBD）、ScanNet（简单室内场景，什么都有）、SUNCG（室内场景，什么都有）、Facebook House3D（室内场景，什么都有）、Matterport3D（复杂室内场景，什么都有）

指标方面：Chamfer distance（CD）（点云）、Earth mover's distance（EMD）（点云）、Intersection Over Union（IOU）（mesh）、F-score（和二分类的F-score一样，需要定义正确/错误）