



特征表示与3D生成大模型

孙铭阳

2025.02.27

尝试回答一个问题：

3D生成，如何选择**特征表示方式**与**模型架构**？



3D物体的表示方式

表示方式在生成式模型中的作用：① 作为输入，告诉模型这个物体有什么特点。② 作为输出，将生成的结果转化为可用的物体。

Level	Content	Encoding	
topology	triangle, quad, hex,	<ul style="list-style-type: none"> Quad: cross field, position & orientation field 	
per element	For vertex/face intrinsic: {curvature, HKS (Heat Kernel Signature), low-frequency Laplacian eigenvectors} extrinsic: {(barycenter) coordinates, normal, dihedral angles}		
3D物体的表示方式	point cloud (samples)	<ul style="list-style-type: none"> point cloud encoder 	
	patch	poly soup/group 3DGS	
	mesh	explicit (vertex, face), graph (node, edge) implicit SDF, NeRF, (continues Function or discrete Grid)	<ul style="list-style-type: none"> mesh convolution: decimation graph convolution: (feature Aggregation, message passing) voxelization and 3D convolution tokenization



3D生成大模型调研



先说结论:

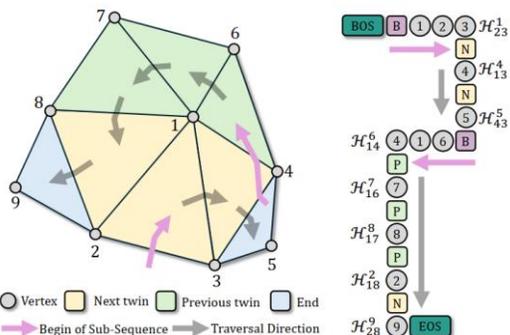
- 拓扑结构、sharp特征: 选AR架构, 用tokenization
- 重视渲染质量: 选LRM, 用latent code



3D生成大模型调研

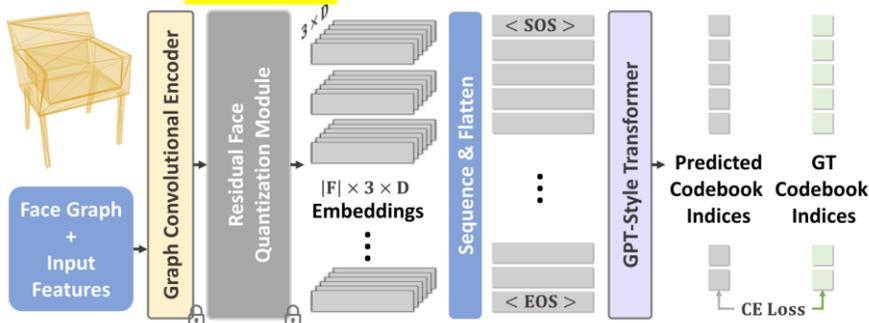
EdgeRunner

AR架构

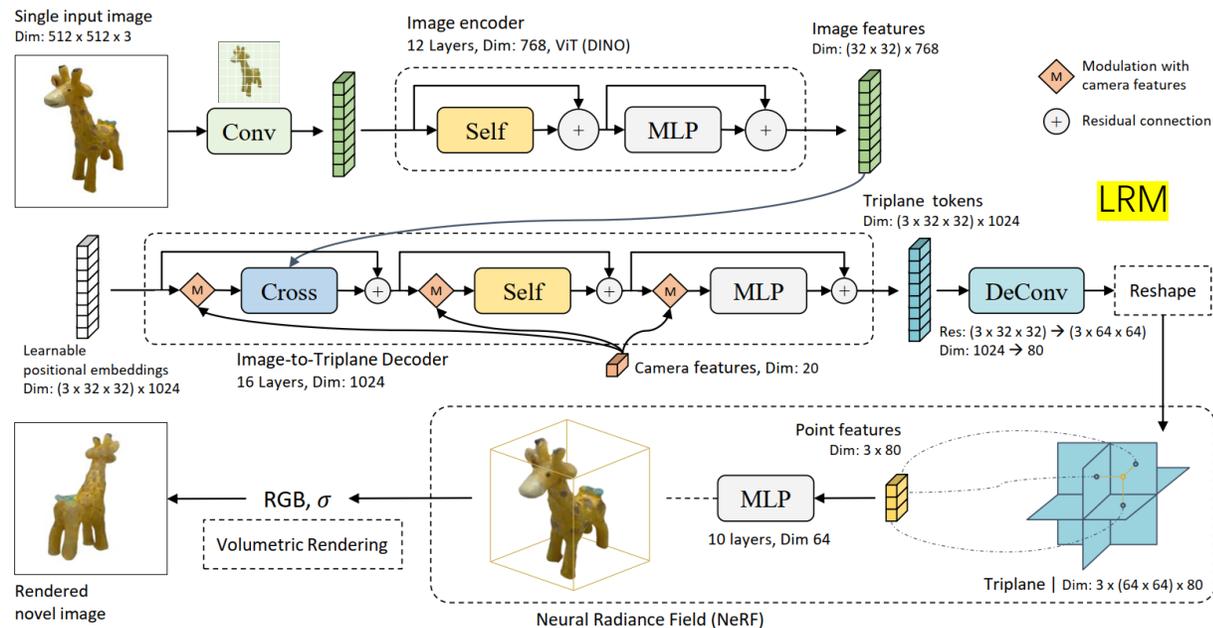


Tokenize by Face Traversal

MeshGPT

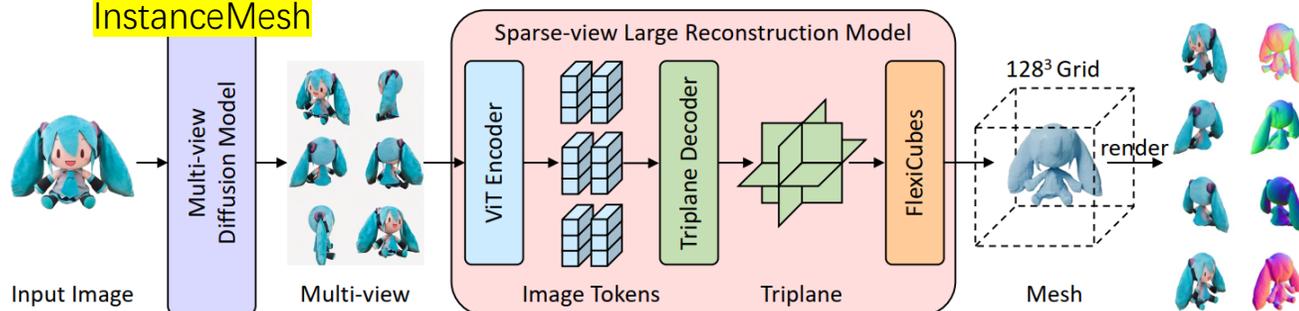


LRM架构



LRM

InstanceMesh



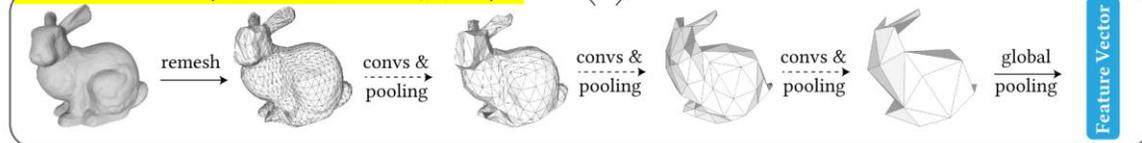


3D物体的特征表示

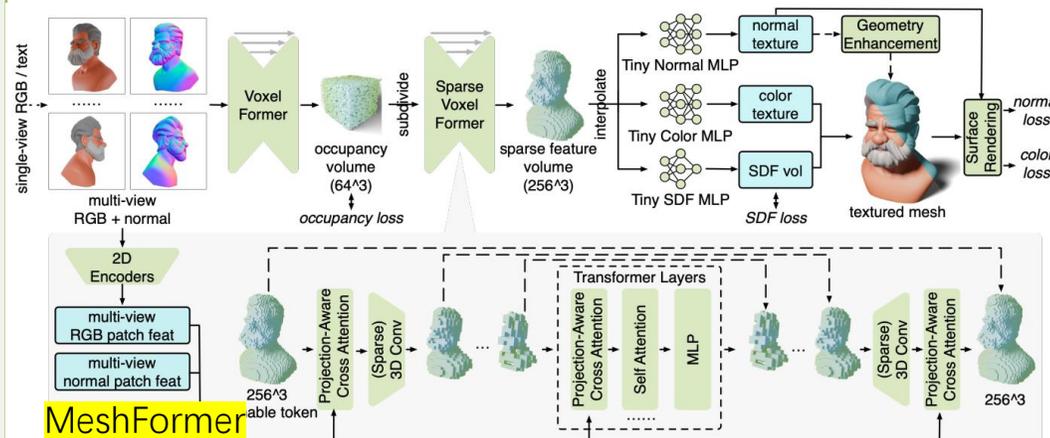
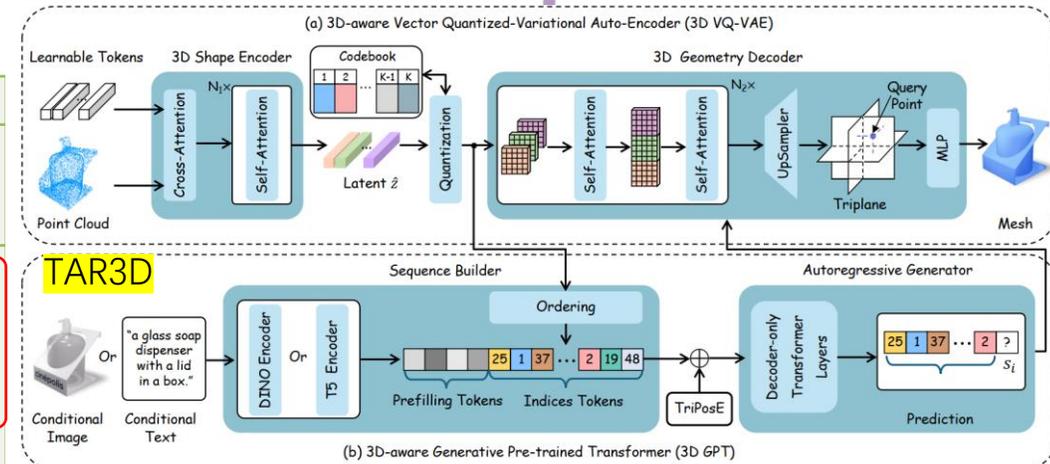
图像的特征编码：分辨率统一，卷就完了

问题：如何将拥有不同顶点数、面数的mesh编码为统一长度的feature vector?

SubdivNet (非3D生成大模型)



item	2D image	3D mesh
primitive	pixel, 每个pixel都有信息	对于vertex: 如果参考pixel, 将mesh的vertex的坐标离散到voxel grid中, 很多voxel里面是空的。 对于face/edge: vertex之间的连接不规则
Quantification	image-wise: 直接用CNN就能提取feature map, 进而作为codebook的query, 自然实现量化	<ul style="list-style-type: none"> LRM: latent code: <ul style="list-style-type: none"> mesh-wise: <ul style="list-style-type: none"> 点云采样: 需要统一mesh的顶点/面数, 用MeshCNN或点云CNN编码为feature vector; 体素化: 然后3D CNN (MeshFormer)。 LRM: rendering image: <ul style="list-style-type: none"> 大重建模型, 输入image, 输出3D的, 一般是将image的feature reshape成triplane, 然后decode为Occ、NeRF、SDF、3DGS等 <ul style="list-style-type: none"> 对于Mesh本身而言, 丢失了原始的结构信息, 无法保持sharp特征、拓扑(布线) auto-regressive (AR) model: tokenization: <ul style="list-style-type: none"> element-wise: VQ-VAE量化。 <ul style="list-style-type: none"> MeshGPT将连续的vertex坐标值量化到voxel grid中, 并根据坐标对vertex排序(重索引), 然后重新得到face, 以face为基本单位送入VQVAE的encoder, 特征提取; decoder输出1D tokens, 其几何意义就是face的3个vertex的坐标。 mesh-wise: 直接对顶点-face进行排序。针对长度不同的问题, 用padding补零。

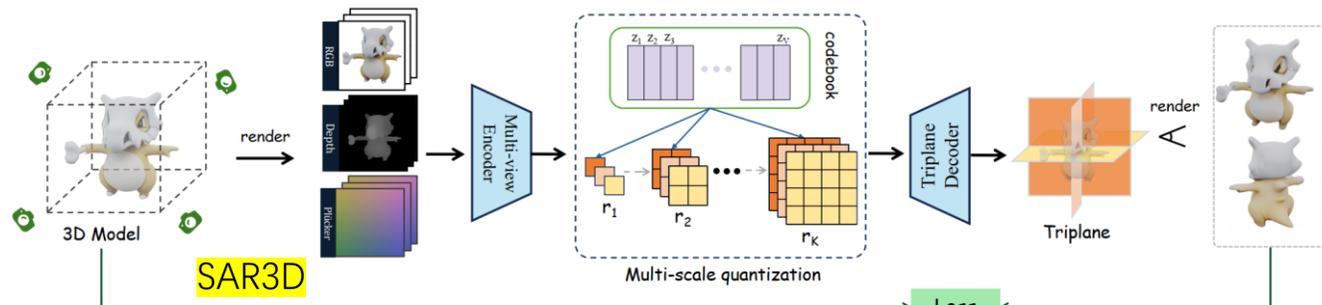




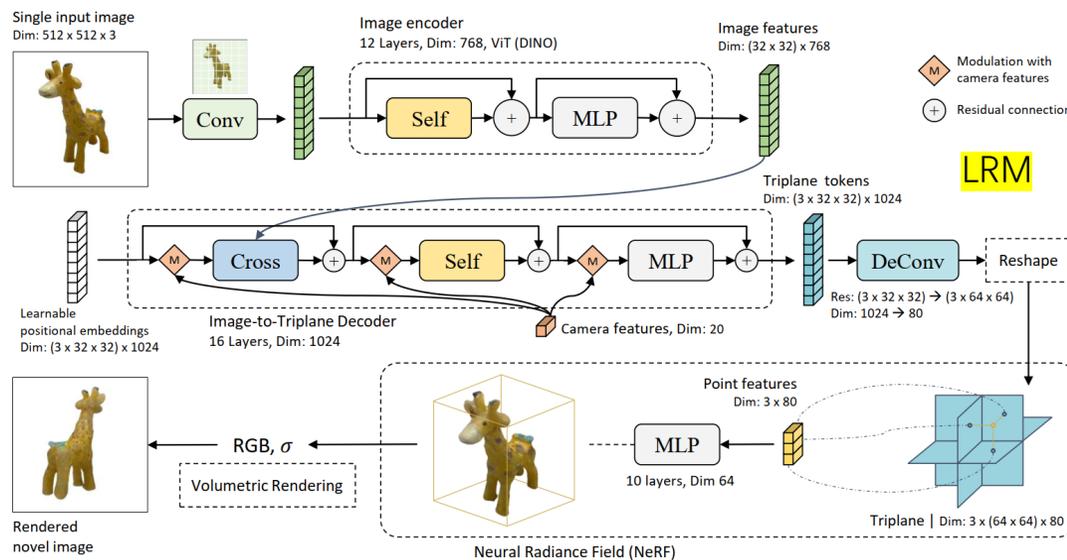
3D物体的特征表示

图像的特征编码：分辨率统一，卷就完了

问题：如何将拥有不同顶点数、面数的mesh编码为统一长度的feature vector?



item	2D image	3D mesh
primitive	pixel, 每个pixel都有信息	对于vertex: 如果参考pixel, 将mesh的vertex的坐标离散到voxel grid中, 很多voxel里面是空的。 对于face/edge: vertex之间的连接不规则
Quantification	image-wise: 直接用CNN就能提取feature map, 进而作为codebook的query, 自然实现量化	<ul style="list-style-type: none"> LRM: latent code: <ul style="list-style-type: none"> mesh-wise: <ul style="list-style-type: none"> 点云采样: 需要统一mesh的顶点/面数, 用MeshCNN或点云CNN编码为feature vector; 体素化: 然后3D CNN (MeshFormer)。 LRM: rendering image: <ul style="list-style-type: none"> 大重建模型, 输入image, 输出3D的, 一般是将image的feature reshape成triplane, 然后decode为Occ、NeRF、SDF、3DGS等 <ul style="list-style-type: none"> 对于Mesh本身而言, 丢失了原始的结构信息, 无法保持sharp特征、拓扑 (布线) auto-regressive (AR) model: tokenization: <ul style="list-style-type: none"> element-wise: VQ-VAE量化。 <ul style="list-style-type: none"> MeshGPT将连续的vertex坐标值量化到voxel grid中, 并根据坐标对vertex排序 (重索引), 然后重新得到face, 以face为基本单位送入VQVAE的encoder, 特征提取; decoder输出1D tokens, 其几何意义就是face的3个vertex的坐标。 mesh-wise: 直接对顶点-face进行排序。针对长度不同的问题, 用padding补零。





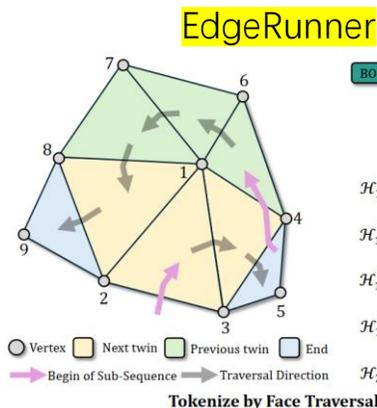
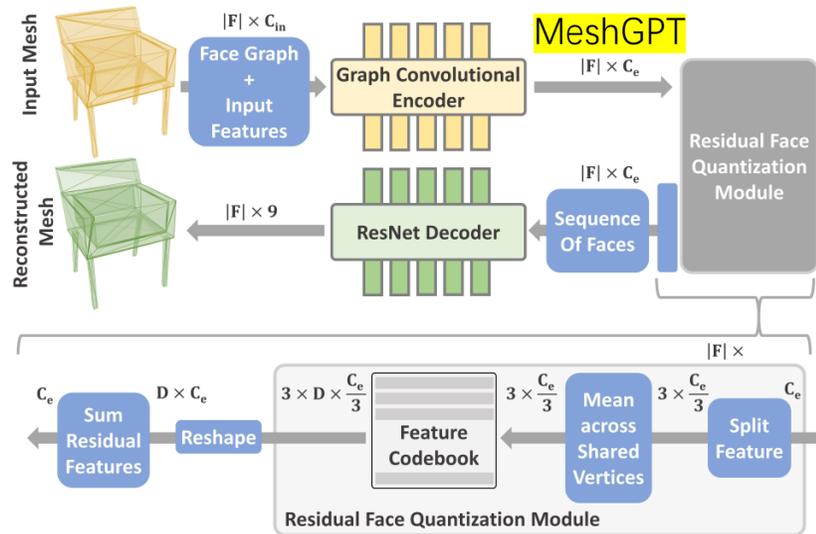
3D物体的特征表示

图像的特征编码：分辨率统一，卷就完了

问题：如何将拥有不同顶点数、面数的mesh编码为统一长度的feature vector？

item	2D image	3D mesh
primitive	pixel, 每个pixel都有信息	对于vertex: 如果参考pixel, 将mesh的vertex的坐标离散到voxel grid中, 很多voxel里面是空的。 对于face/edge: vertex之间的连接不规则
Quantification	image-wise: 直接用CNN就能提取feature map, 进而作为codebook的query, 自然实现量化	<ul style="list-style-type: none"> LRM: latent code: <ul style="list-style-type: none"> mesh-wise: <ul style="list-style-type: none"> 点云采样: 需要统一mesh的顶点/面数, 用MeshCNN或点云CNN编码为feature vector; 体素化: 然后3D CNN (MeshFormer)。 LRM: rendering image: <ul style="list-style-type: none"> 大重建模型, 输入image, 输出3D的, 一般是将image的feature reshape成triplane, 然后decode为Occ、NeRF、SDF、3DGS等 <ul style="list-style-type: none"> 对于Mesh本身而言, 丢失了原始的结构信息, 无法保持sharp特征、拓扑 (布线) auto-regressive (AR) model: tokenization: <ul style="list-style-type: none"> element-wise: VQ-VAE量化。 <ul style="list-style-type: none"> MeshGPT将连续的vertex坐标值量化到voxel grid中, 并根据坐标对vertex排序 (重索引), 然后重新得到face, 以face为基本单位送入VQVAE的encoder, 特征提取; decoder输出1D tokens, 其几何意义就是face的3个vertex的坐标。 mesh-wise: 直接对顶点-face进行排序。针对长度不同的问题, 用padding补零。

token代表特征向量的index, 去codebook中拿到特征向量, 随后被解码为element的几何信息



token代表element的几何信息