

基于 Bayes-MeTiS 网格划分的 3D 几何重构

张小华 黄 波

(四川大学计算机学院 成都 611844)

摘 要 为提升 3D 模型几何重构过程的压缩效率,提出一种基于 MeTiS 网格划分的贝叶斯 3D 模型几何重构算法。首先,在编码端采用 MeTiS 方法对原始 3D 网格进行子网划分,采用随机线性矩阵对子网几何形状进行编码,并对边界节点的邻居节点使用伪随机数生成器进行数据序列构建;然后,利用贝叶斯算法进行几何模型重构算法的设计,在理论上给出了均值、方差矩阵以及模型参数学习规则,实现了 3D 模型的几何重构;最后,将其与图傅里叶光谱压缩(GFT)、最小二乘压缩(LMS)和基于压缩感知的图傅里叶光谱压缩(CSGFT)等算法进行仿真对比。结果表明,所提方法具有较高的比特率压缩指标以及较低的重构误差,计算效率明显提高。

关键词 3D 模型,几何重构,MeTiS 网格划分,贝叶斯,邻居节点

中图分类号 TP391.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.047

3D Geometric Reconstruction Based on Bayes-MeTiS Mesh Partition

ZHANG Xiao-hua HUANG Bo

(College of Computer Science, Sichuan University, Chengdu 611844, China)

Abstract In order to improve the compression efficiency of the geometric reconstruction process of 3D model, this paper proposed a bayesian geometric reconstruction algorithm based on MeTiS mesh partition for 3D model. At the encoding part, the MeTiS method is used to realize the subnetting for original 3D grid, the random linear matrix is used to encode the geometry of the subnet, and the pseudo random number generator is used for data sequence construction by considering the neighbor nodes of the boundary nodes. Then, the Bayesian algorithm is used to design the geometric model reconstruction algorithm, and the mean, variance matrix and the model parameters are given in theory to realize the geometric reconstruction of the 3D model. Finally, by comparing with graph Fourier transform spectral compression (GFT), least square compression (LMS) and compressed sensing based graph Fourier transform spectral compression algorithms (CSGFT), the simulation results show that the proposed method has relatively high bit rate compression index and low reconstruction error.

Keywords 3D model, Geometric reconstruction, MeTiS mesh partition, Bias, Neighbor node

1 引言

三维模型重建隶属于人工智能研究领域,具有广阔的应用前景和需求,该问题的研究成果可以直接应用于机器导航、精密工业测量、物体识别、医学仪器、虚拟现实以及军事等领域。近年来,搜索、系统设计以及应用开发等领域对三维模型的获取、处理、传输和存储的研究兴趣日益增长,如移动云游戏和 3D 远程设备等。在许多应用中,重构算法不一定完全等于输入数据,允许一些不容易区分的精度损失,这为高密度三维模型的高效传输带来了便利^[1-3]。

在应用程序中,3D 模型表示的最常用的方法是多边形建模,即使用 3D 网格对表面进行近似。3D 网格一般是由提供几何信息的顶点和确定多边形基本面的顶点的连线构成^[4]。在这方面的研究中,文献[5]重点介绍了不同的网格压缩方法的最新发展情况,主要集中在几何信息连接编码的忽略上。文献[6]的研究表明,现有连接编码效率已经接近最优水平,

其结果促使我们专注于几何压缩算法的研究。在几何压缩算法的研究中,文献[7]建议采用三维空间坐标进行直接量化。这种形式的量化误差引入了高频分量,极大地改变了模型的外观,使其成为了具有块状结构的三维模型,但其重构误差明显。文献[8]提出了一种 Laplace 高通编码方式,它建立在坐标系微分量化的基础上。这种方法能成功地捕捉顶点的局部关系,通常优于文献[7]的直接量化方法。类似地,文献[10]建议采用笛卡尔坐标投影,通过发送少量的“低频”组件实现图傅里叶域的压缩估计,但增加了拉普拉斯矩阵的特征值分解以及顶点坐标投影的计算,因此增加了计算的复杂度。

为了克服上述限制,压缩感知(Compressed Sensing, CS)算法以其低复杂度的特性得到了广泛研究。该方案的优点在于:1)编码复杂度低,因为不需要在编码器上计算稀疏信号;2)具有普遍性,因为传感器具有盲源分布特性;3)具有隐私保护性,因为它本质上涉及一个随机化过程。然而,随着三维模型的顶点数量的增长,基于普拉斯压缩/重建的方法是不可行

到稿日期:2017-04-20 返修日期:2017-07-18 本文受四川省教育厅科研项目(17ZB0007)资助。

张小华(1975-),男,博士,讲师,主要研究领域为机器视觉、智能算法和虚拟现实技术,E-mail:137136612@qq.com(通信作者);黄波(1976-),男,硕士,讲师,主要研究领域为机器视觉、智能算法和虚拟现实技术。

的,因为它们需要进行矩阵的特征值分解求逆。为解决这一问题,需要对三维对象进行子网划分和处理(例如文献[10]中的方法),但其须在解码器端知道拉普拉斯特征向量的先验知识。

本文引入了一个新的几何压缩/重建算法,以提高拉普拉斯处理性能。该算法在部分密集的3D模型处理和重建过程中,根据其局部特点(如光谱值的潜在相关性)获得图傅里叶域不同部分的几何信息。在大型3D模型上的性能评估显示,所提算法具有更高的计算精度和效率。

2 问题描述

规定矩阵 A 的第 i 行和第 j 列可表示为 $A_{(i,j)}$,同时矩阵 A 的第 i 行和第 j 列可表示为 $A_{(i, \cdot)}$ 和 $A_{(\cdot, j)}$; $(\cdot)^T$ 表示矩阵的转置; $E(\cdot)$ 表示期望算子;矩阵 A 的迹为 $Tr(A)$ 。

本文主要考虑三角网格,其是最常见的多边形模型。假设每个具有 n 个顶点的三角形网格(\mathcal{M})都可表示为两个不同的集合 $\mathcal{M}=(V, F)$,其中, V 为对应顶点, F 为索引的面。边集可直接在顶点 V 和索引的面 F 中获得。大部分的网格几何压缩工作的实现基础是光滑的几何形状应该由低频分量占主导地位的光谱产生。因此在傅里叶基图中采用笛卡儿坐标系 $x, y, z \in \mathbb{R}^{n \times 1}$,其特征向量为 u_i , Laplace 操作算子为 $L^{[11-12]}$:

$$L = I_n - D^{-1}C \quad (1)$$

其中, I_n 为 $n \times n$ 的单位矩阵, $C \in \mathbb{R}^{n \times n}$ 是元素网格的连通矩阵:

$$C_{(i,j)} = \begin{cases} 1, & (i,j) \in (E) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

其中, D 为对角矩阵, $D_{(i,i)} = |N(i)|$ 。 $N(i) = \{j | (i,j) \in (E)\}$ 为节点 i 的直接邻居集。假设 L 的特征值分解为:

$$L = U\Lambda U^T \quad (3)$$

其中, Λ 是一个包含 L 特征值的对角矩阵, $U = [u_1, u_2, \dots, u_n]$ 是一个包含特征向量 $u_i (u_i \in \mathbb{R}^{n \times 1})$ 的矩阵。那么上述的压缩方案可利用拉普拉斯算子 L 和特征向量 U 的欧氏投影坐标 (v) 的稀疏表示,其紧凑表示形式为:

$$v \approx \sum_{i=1}^k (u_i^T v) u_i, k \leq n \quad (4)$$

3 基于贝叶斯的3D模型重构

3.1 算法体系结构

首先对提出的稠密三维模型的压缩/重建方案进行描述。所提出的体系结构如图1所示。

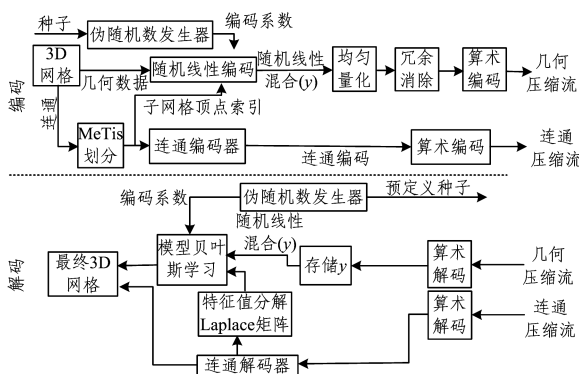


图1 算法体系结构

Fig. 1 Algorithm architecture

根据所提出的体系结构,单独处理网格的几何形状和连通性。使用目前最先进的连接编码器进行链接,因此本文专注于提供一种新的网格几何形状的压缩/重建方法。该算法表现出与光谱压缩方法相似的性能,同时能尽量减少编码器的处理过程,因为信号是稀疏的,其对源信号分布也是未知的。

3.2 编码器的子网划分和处理

在编码端,采用 MeTiS 方法对原始 3D 网格进行 L 子网划分。MeTiS 算法提供了解决大型图划分问题的可行方案,例如处理超过 100000 个顶点的图,其通过线性时间优化来实现。MeTiS 通过计算 L -way 分区最小化,有效解决了一般的分区问题,可跨越不同的分区边数。每个子网格 l 包含 n_l 个节点,其中 $\sum_{l=1}^L n_l = n$ 。第 l 个子网的 n_l 个节点可以表示为一个 $n_l \times 3$ 的矩阵, $v_l = [x_l, y_l, z_l]$,其中 $x_l, y_l, z_l \in \mathbb{R}^{n_l \times 1}$ 。图2给出了采用 MeTiS 算法获得的斯坦福暴龙模型、斯坦福狢狢模型与 Kinect 传感器扫描的椅子模型的分区结构。子网颜色通过随机方式选择,而子网划分的顶点采用白色。



图2 MeTiS 分区

Fig. 2 MeTiS partition

权衡点的选取是将三维物体表示为单一几何网格,重建质量退化的子网端点,其被称为边界节点。产生退化的原因是忽视了子网处理过程中的边界节点的邻居节点。为了解决这个问题,本文在重叠的子网处理过程中,通过相邻子网的边界节点进行子网扩展。首先如果 $I_l = \{i_1, i_2, \dots, i_{n_l}\}$ 表示包含在子网 l 内的 3D 网格节点索引集,则可得:

$$v_{(I_l, \cdot)} = [v_{i_1}^T, \dots, v_{i_{n_l}}^T]^T, v_i = [x_i, y_i, z_i] \quad (5)$$

其中, $I_b (I_b = \{i_1, i_2, \dots, i_{n_b}\} \subset I_l)$ 是子网 l 的边界节点索引集;然后用属于集合 $I_{nb} = \bigcup_{j=1}^b N(i_{i_j}) \setminus I_l$ 的边界节点的邻居节点的笛卡儿坐标系进行矩阵 v_l 的拓展,形式为:

$$v_l = [v_{(I_l, \cdot)}, v_{(I_b, \cdot)}]^T \quad (6)$$

其中,顶点数 $|I_l \cup I_{nb}| = n_{l_c} > n_l$ 。对于每个子网,数据源通过使用维度为 $M \times n_{l_c}$ 的随机线性矩阵 A 产生 $M \times 3$ 的随机线性组合,随机线性组合选取 ± 1 值的概率为 0.5 (Rademacher 分布)。

$$y = Av_{l_c} \quad (7)$$

对于每个子网几何形状,采用 $M \times n_{l_c} (M \leq n_{l_c})$ 的随机线性矩阵 A 进行编码,在编码器和解码器中使用伪随机数生成器 (Pseudo Random Number Generator, PRNG) 进行构建,产生近似随机数性质的数列。生成的序列完全由随机种子决定,由唯一信息确定,其将被发送到接收机。为减少更多的通信需求,使用同样的种子对所有的子网组成的原始网格进行编码。因此,即使考虑到传输种子的开销,解码器的计算压缩率也不会受到影响,因为使用相同的种子对总数为 $n = \sum_{l=1}^L n_l >$

10^5 的节点进行编码,每个顶点利用 3 个对应于欧氏坐标 x, y, z 的 32 位数字表示。

在编码阶段,去除一个子网内所有剩余的冗余相邻顶点坐标。该模块的输出是 n_{l_e} 维向量 \mathbf{z}_q 的均匀量化。然后,利用无损赫夫曼编码获得编码向量 \mathbf{c} 。在解码器端,通过对坐标间的差异进行解码获得随机线性组合修复,随机线性编码向量 \mathbf{y}_q 为:

$$\mathbf{y}_q = \mathbf{Q}(\mathbf{A}\mathbf{v}_{l_e}) = \mathbf{A}\mathbf{v}_{l_e} + \mathbf{w}_q = \mathbf{A}_{u_i}\mathbf{s}_{l_e} + \mathbf{w}_q \quad (8)$$

其中, $\mathbf{A}_{u_i} = \mathbf{A}\mathbf{U}_{l_e}$, \mathbf{U}_{l_e} 是拓展子网 l 的 Laplacian 算子 \mathbf{L}_{l_e} 的特征向量, $\mathbf{s}_{l_e} = [s_{x_{l_e}}, s_{y_{l_e}}, s_{z_{l_e}}] \in \mathcal{R}^{n_{l_e} \times 3}$ 是在相应图傅里叶基上的笛卡尔坐标投影, $\mathbf{w}_q = [w_{q_x}, w_{q_y}, w_{q_z}]$ 表示量化误差。

根据向量 \mathbf{y}_q 对点 \mathbf{v}_{l_e} 进行重构的过程中,从解码的连通性角度直接建立 Laplacian 矩阵 \mathbf{L}_{l_e} , 并通过执行奇异值分解过程,对每个子网格进行矩阵 \mathbf{U}_{l_e} 评价。然后,通过解码器生成二进制矩阵 \mathbf{A} , 并使用预定义随机种子对每个子网格的顶点进行编码。最后,利用二进制矩阵 \mathbf{A} 、Laplace 特征向量 \mathbf{U}_{l_e} 和线性组合编码向量 \mathbf{y}_q 对每个子网格的原始几何形状信息进行解码,具体描述如下。

3.3 贝叶斯模型重构的理论分析

算法设计的基础是:1)GFT 的行为与 DCT 过程非常相似,因为它对蕴含在数据中的能量进行重新分配,使得大部分的能量包含在少量的组件中;2)自然图像和音频信号的离散余弦变换(DCT)系数的值通常被描述为多变量的高斯分布。将每个扩展子网 l 在 GFT 域内表示为直角投影坐标,例如 $\mathbf{s}_{l_e} = \mathbf{U}_{l_e}^T \mathbf{v}_{l_e}$, 可利用由 k 个非零元素和 $n_{l_e} - k$ 个零元素组成的稀疏向量表示:

$$\mathbf{s}_{l_e} = [s_{i_1}, \mathbf{0}_{n_{l_e}-k}], \mathbf{s}_{i_1} = [s_{i_1}, s_{i_2}, \dots, s_{i_k}] \quad (9)$$

其中, $k < n_{l_e}$, i 对应于 x, y, z 的坐标, \mathbf{s}_{i_1} 表示大小为 k 的非零块,可被建模为一个参数化的多元高斯分布:

$$\begin{cases} p(\mathbf{s}_{i_1}) \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_i) \\ \mathbf{C}_i = \gamma_i \Sigma_i \\ i = \{x, y, z\} \end{cases} \quad (10)$$

其中, γ_i 是标量参数, Σ_i ($\Sigma_i \in \mathcal{R}^{k \times k}$) 是正定矩阵。通过 Bayes 规则,并假定式(8)中的噪声向量 \mathbf{w}_q 可视为 $M \times 1$ 维的高斯独立同分布的随机变量 $\mathbf{w}_q \sim \mathcal{N}(\mathbf{0}, \sigma_{w_i} \mathbf{I}_M)$, 可以得到以下评估后验概率编码系数的命题。

命题 1 \mathbf{s}_{i_1} 的后验密度同样也是高斯矩阵 $p(\mathbf{s}_{i_1} | \mathbf{y}_{q_i}; \sigma_{w_i}, \gamma_i \Sigma_i) \sim \mathcal{N}(\mu_{s_{i_1}}, \mathbf{C}_i)$, 其均值和方差矩阵分别为:

$$\begin{cases} \mu_{s_{i_1}} = \mathbf{C}_i \mathbf{A}_{u_i}^T (\mathbf{A}_{u_i} \mathbf{C}_i \mathbf{A}_{u_i}^T + \sigma_{w_i} \mathbf{I}_M)^{-1} \mathbf{y}_{q_i} \\ \mathbf{C}_i = \mathbf{C}_i - \mathbf{C}_i \mathbf{A}_{u_i}^T (\mathbf{A}_{u_i} \mathbf{C}_i \mathbf{A}_{u_i}^T + \sigma_{w_i} \mathbf{I}_M)^{-1} \mathbf{A}_{u_i} \mathbf{C}_i \end{cases} \quad (11)$$

其中, \mathbf{A}_{u_i} ($\mathbf{A}_{u_i} = \mathbf{A}_{u_i(\cdot, i)}$) 为 $M \times k$ 矩阵,其包含 \mathbf{A}_{u_i} 的前 k 列。

证明:假定式(8)的线性模型观测 \mathbf{A}_{u_i} 和 \mathbf{s}_{i_1} 的概率密度可基于噪声独立性假设获得:

$$\begin{aligned} p(\mathbf{y}_q | \mathbf{A}_{u_i}, \mathbf{s}_{i_1}) &= \prod_{n=1}^M p(y_{q_n} | (\mathbf{A}_{u_i})_{(\cdot, n)}, \mathbf{s}_{i_1}) = \\ & \exp\left(-\frac{1}{2\sigma_{w_i}} \|\mathbf{y}_q - \mathbf{A}_{u_i} \mathbf{s}_{i_1}\|_2^2\right) \\ & \sim \mathcal{N}(\mathbf{A}_{u_i} \mathbf{s}_{i_1}, \sigma_{w_i} \mathbf{I}_M) \end{aligned} \quad (12)$$

后验分布可以根据贝叶斯规则计算:

$$p(\mathbf{s}_{i_1} | \mathbf{y}_q, \mathbf{A}_{u_i}) = \frac{p(\mathbf{y}_q | \mathbf{A}_{u_i}, \mathbf{s}_{i_1}) p(\mathbf{s}_{i_1})}{p(\mathbf{y}_q | \mathbf{A}_{u_i})} \quad (13)$$

其中的归一化常数也称边缘似然,是独立的图形傅里叶系数,其计算式为:

$$p(\mathbf{y}_q | \mathbf{A}_{u_i}) = \int p(\mathbf{y}_q | \mathbf{A}_{u_i}, \mathbf{s}_{i_1}) p(\mathbf{s}_{i_1}) d\mathbf{s}_{i_1} \quad (14)$$

替换式(13)中的后验概率,可得如下形式:

$$\begin{aligned} p(\mathbf{s}_{i_1} | \mathbf{y}_q, \mathbf{A}_{u_i}) &\propto \exp \\ & \left(-\frac{(\mathbf{s}_{i_1} - \mu_{s_{i_1}})^T (\sigma_{w_i}^{-1} \mathbf{A}_{u_i}^T \mathbf{A}_{u_i} + \mathbf{C}_i^{-1}) (\mathbf{s}_{i_1} - \mu_{s_{i_1}})}{2}\right) \end{aligned} \quad (15)$$

$$\mu_{s_{i_1}} = \sigma_{w_i}^{-1} (\sigma_{w_i}^{-1} \mathbf{A}_{u_i}^T \mathbf{A}_{u_i} + \mathbf{C}_i^{-1})^{-1} \mathbf{A}_{u_i} \mathbf{y}_q \quad (16)$$

因此,将后验分布识别为高斯形式:

$$p(\mathbf{s}_{i_1} | \mathbf{y}_q, \mathbf{A}_{u_i}) \sim \mathcal{N}\left(\frac{1}{\sigma_{w_i}} \mathbf{C}_i \mathbf{A}_{u_i} \mathbf{y}_q, \mathbf{C}_i\right) \quad (17)$$

其中,

$$\begin{aligned} \mathbf{C}_i &= (\sigma_{w_i}^{-1} \mathbf{A}_{u_i}^T \mathbf{A}_{u_i} + \mathbf{C}_i^{-1})^{-1} \\ &= \mathbf{C}_i - \mathbf{C}_i \mathbf{A}_{u_i}^T (\sigma_{w_i} \mathbf{I}_M + \mathbf{A}_{u_i} \mathbf{C}_i \mathbf{A}_{u_i}^T)^{-1} \mathbf{A}_{u_i} \mathbf{C}_i \end{aligned} \quad (18)$$

因此,对于给定的参数 $\sigma_{w_i}, \mathbf{C}_i$ 的 MAP 估计是后验分布 $p(\mathbf{s}_{i_1} | \mathbf{y}_q, \mathbf{A}_{u_i})$ 的平均值,则式(16)可改写为:

$$\begin{aligned} \hat{\mu}_{s_{i_1}} &= (\sigma_{w_i} \mathbf{C}_i^{-1} + \mathbf{A}_{u_i}^T \mathbf{A}_{u_i})^{-1} \mathbf{A}_{u_i}^T \mathbf{y}_q \\ &= \mathbf{C}_i \mathbf{A}_{u_i}^T (\mathbf{A}_{u_i} \mathbf{C}_i \mathbf{A}_{u_i}^T + \sigma_{w_i} \mathbf{I}_M)^{-1} \mathbf{y}_q \end{aligned} \quad (19)$$

证毕。

对于给定的参数 $\sigma_{w_i}, \gamma_i, \Sigma_i$ ($i = \{x, y, z\}$), 扩展子网 L 的笛卡尔坐标最大后验概率(MAP)估计为:

$$\mathbf{v}_{l_e} = \mathbf{U}_{l_e} \begin{bmatrix} \mu_{s_{x_i}} & \mu_{s_{y_i}} & \mu_{s_{z_i}} \\ \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k} \end{bmatrix} \quad (20)$$

然后,对于给定的参数 $\sigma_{w_i}, \gamma_i, \Sigma_i$, 采用期望最大化(EM)对每个坐标算法求取最大化 $p(\mathbf{y}_{q_i}; (\mathbf{s}_{i_1} | \mathbf{y}_{q_i}; \sigma_{w_i}, \gamma_i, \Sigma_i))$, 其中 i 可为 x, y 或 z 。这相当于最大限度地减少 $-\log(p(\mathbf{y}_{q_i}; \theta))$, 可得到观察到的数据的边缘似然为:

$$\mathcal{Q}(\theta; \mathbf{y}_{q_i}) = (\mathbf{y}_{q_i})^T \sum_{i_1}^{-1} \mathbf{y}_{q_i} + \log |\Sigma_{y_i}| \quad (21)$$

其中, $\Sigma_{y_i} = \sigma_{w_i} \mathbf{I}_M + \mathbf{A}_{u_i} \mathbf{C}_i \mathbf{A}_{u_i}^T$ 。以下命题给出了用于估计模型参数 $\sigma_{w_i}, \gamma_i, \Sigma_i$ 的迭代方案。

命题 2 对于每个坐标 $i = \{x, y, z\}$, 应用 EM 算法可得到模型参数 $\sigma_{w_i}, \gamma_i, \Sigma_i$ 的学习规则:

$$\sigma_{w_i} = \frac{\|\mathbf{y}_{q_i} - \mathbf{A}_{u_i} \hat{\mathbf{s}}_{i_1}\|_2^2 + \sigma_{w_i} [k - \text{Tr}(\mathbf{C}_i \mathbf{C}_i^{-1})]}{M} \quad (22)$$

$$\gamma_i = \frac{\text{Tr}(\sum_i^{-1} (\mathbf{C}_i + \mu_{s_{i_1}} (\mu_{s_{i_1}})^T))}{k} \quad (23)$$

$$\Sigma_i = \frac{\mathbf{C}_i + \mu_{s_{i_1}} (\mu_{s_{i_1}})^T}{\gamma_i} \quad (24)$$

证明:EM 算法试图通过以下两个步骤来寻找边缘似然最大似然估计: E 过程和 M 过程。

1) E 过程, 计算似然函数期望值:

$$\begin{aligned} Q(\theta | \theta^{(t)}) &= E_{\mathbf{s}_{i_1} | \mathbf{y}_{q_i}; \theta^{(t)}} [\log p(\mathbf{y}_q, \mathbf{s}_{i_1}; \theta^{(t)})] + \\ & E_{\mathbf{s}_{i_1} | \mathbf{y}_{q_i}; \theta^{(t)}} [\log p(\mathbf{s}_{i_1}; \gamma_i, \Sigma_i)] \end{aligned} \quad (25)$$

2) M 过程, 寻找参数 $\theta = (\sigma_{w_i}, \gamma_i, \Sigma_i)$, 式(26)最大化:

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta | \theta^{(t)}) \quad (26)$$

通过简化式(24)中的参数 σ_{w_i} , 得到:

$$Q(\sigma_{w_i} | \theta^{(t)}) \propto E_{\mathbf{s}_{i_1} | \mathbf{y}_{q_i}; \theta^{(t)}} [\log p(\mathbf{y}_q, \mathbf{s}_{i_1}; \sigma_{w_i})] \quad (27)$$

对 $Q(\sigma_{w_i} | \theta^{(t)})$ 求导, 并设定其为 0, 得到:

$$\sigma_{w_i} = \frac{\| \mathbf{y}_{q_i} - \mathbf{A}_{u_i} \mathbf{s}_{i_k} \|_2^2 + \sigma_{w_i} [k - \text{Tr}(\mathbf{C}_i \mathbf{C}_i^{-1})]}{M} \quad (28)$$

为顾及参数 γ_i, Σ_i , 将式(24)简化为:

$$Q(\gamma_i, \Sigma_i | \theta^{(l)}) \propto E_{s_{i_k} | \gamma_i, \Sigma_i} [\log p(s_{i_k}; \gamma_i, \Sigma_i)] \quad (29)$$

同时, 因为满足:

$$\log p(s_{i_k}; \gamma_i, \Sigma_i) \propto -\frac{k}{2} \log |\gamma_i, \Sigma_i| - \frac{\mathbf{s}_{i_k}^T (\gamma_i \Sigma_i)^{-1} \mathbf{s}_{i_k}}{2} \quad (30)$$

所以可得式(26)等价于:

$$Q(\gamma_i, \Sigma_i | \theta^{(l)}) \propto -\frac{k}{2} \log |\gamma_i| - \frac{k}{2} \log |\Sigma_i| - \frac{\text{Tr}((\gamma_i \Sigma_i)^{-1} (\mathbf{C}_i + \mu_{s_{i_k}} \mu_{s_{i_k}}^T))}{2} \quad (31)$$

其中, $\mu_{s_{i_k}}$ 和 \mathbf{C}_i 可根据式(11)进行估计, 利用式(30)对 γ_i 求偏导, 则得到 γ_i 的学习规则:

$$\gamma_i = \frac{\text{Tr}(\sum_i (\mathbf{C}_i + \mu_{s_{i_k}} \mu_{s_{i_k}}^T))}{k} \quad (32)$$

类似地, 得到 $\Sigma_i = \frac{\mathbf{C}_i + \mu_{s_{i_k}} (\mu_{s_{i_k}})^T}{\gamma_i}$ 成立。证毕。

3.4 几何重构算法步骤

为提高算法的性能, 可将矩阵 Σ_i 限制为 Toeplitz 对称结构, 其元素为 $(\Sigma_i)(m, l) = r_i^{|m-l|}, \forall m, l \in [1, 2, \dots, k]$ 。该形式相当于将非零块中的元素建模为一阶自回归过程。 r_i 的估计值为:

$$r_i = \text{sign}(\frac{m_1}{m_0}) \min\{|\frac{m_1}{m_0}|, 0.99\} \quad (33)$$

其中, m_0 是沿主对角线元素的平均值, m_1 是沿 Σ_i 主子对角线的元素的平均值。所提几何重构算法如算法 1 所示。

算法 1 几何重构算法

输入: 编码规则, \mathbf{A}, \mathbf{U}_l , 编码样本 \mathbf{y}_{q_i} , 非零块长度 k

输出: 估计参数 $x, y, z \in \mathcal{R}^{n_c \times 1}$

For each Submesh $l=1, 2, \dots, L$

For each iteration $m=1, 2, \dots, K$

计算非零值 $\mu_{s_{i_1}}, \mu_{s_{i_2}}$ 和 $\mu_{s_{i_3}}$:

$$\mu_{s_{i_k}} = \mathbf{C}_i \mathbf{A}_{u_{s_{i_k}}}^T (\mathbf{A}_{u_{i_k}} \mathbf{C}_i \mathbf{A}_{u_{i_k}}^T + \sigma_{w_i} \mathbf{I}_M)^{-1} \mathbf{y}_{q_i}$$

计算对应方差 $\mathbf{C}_x, \mathbf{C}_y$ 和 \mathbf{C}_z :

$$\mathbf{C}_i = \mathbf{C}_i - \mathbf{C}_i \mathbf{A}_{u_{i_k}}^T (\mathbf{A}_{u_{i_k}} \mathbf{C}_i \mathbf{A}_{u_{i_k}}^T + \sigma_{w_i} \mathbf{I}_M)^{-1} \mathbf{A}_{u_{i_k}} \mathbf{C}_i$$

更新参数 $\sigma_{w_i}, \gamma_i, \Sigma_i$:

$$\sigma_{w_i} = \frac{\| \mathbf{y}_{q_i} - \mathbf{A}_{u_{i_k}} \mathbf{s}_{i_k} \|_2^2 + \sigma_{w_i} [k - \text{Tr}(\mathbf{C}_i \mathbf{C}_i^{-1})]}{M}$$

$$\gamma_i = \frac{\text{Tr}(\sum_i (\mathbf{C}_i + \mu_{s_{i_k}} (\mu_{s_{i_k}})^T))}{k}$$

$$\Sigma_i = \frac{\mathbf{C}_i + \mu_{s_{i_k}} (\mu_{s_{i_k}})^T}{\gamma_i}$$

重新对 γ_i 进行估计:

$$r_i = \text{sign}(m_1/m_0) \min\{|m_1/m_0|, 0.99\}$$

重新对 Σ_i 进行估计:

$$\Sigma_i = \text{Toeplitz}\{[1, r, \dots, r^{k-1}]\}$$

EndFor

计算 l 子网格的笛卡尔坐标:

$$\mathbf{v}_l = \mathbf{U}_l \begin{bmatrix} \mu_{s_{i_1}} & \mu_{s_{i_2}} & \mu_{s_{i_3}} \\ 0_{n_c-k} & 0_{n_c-k} & 0_{n_c-k} \end{bmatrix}$$

对下降点执行操作 $\bigcup_{j=1}^b N(i_l) \setminus I_l$, 其中 i_l 为子网 l 的边界节点。

EndFor

上述几何重构算法存在两处循环计算, 分别是“For each Submesh”和“For each iteration”, 这两个循环呈现嵌套形式, 前者的计算量为 $1:L$, 后者的计算量为 $1:K$, 因此“For each iteration”循环内的代码计算复杂度为 $O(4L \times K)$, “For each Submesh”循环内、“For each iteration”循环外代码的计算复杂度为 $O(L)$, 因此上述的总计算复杂度为 $O(4L \times K + L) = O(n^2)$ 。而经典的图傅里叶变换光谱压缩算法 GFT, LSM 和 CSGFT 的计算复杂度均为 $O(n^3)$ 。

4 实验分析

实验的硬件配置如下: 处理器为 Intel(R) i5-2440k, 2.20 GHz, 系统内存为 4GB, 频率为 DDR3L-1600 RAM, 操作系统为 Win7 旗舰版。仿真实验软件为 Matlab 2013a。图 3 为本实验中采用的 3D 网格模型及其参数设置。对比指标选取比特率和均方根误差 (RMSE)。



图 3 选取的 3D 模型样本

Fig. 3 Selected 3D model samples

对比算法为 GFT, LSM 和 CSGFT, 3 种算法的具体描述如表 1 所列。

表 1 对比算法

Table 1 Contrast algorithms

| 算法 | 简称 |
|-----------------------------------|-------|
| 图傅里叶光谱压缩算法 ^[13] | GFT |
| 最小二乘压缩算法 ^[14] | LSM |
| 基于压缩感知的图傅里叶光谱压缩算法 ^[15] | CSGFT |

在仿真实验的对比过程中, 对位于 x, y 和 z 坐标上的每个顶点进行 12bits 的预先量化处理, 则 4 种算法的均方根误差指标的对比如图 4 所示。图 4 给出了在网格模型上 LSM 算法、GFT 算法、CSGFT 算法及本文算法的均方根误差指标随比特率指标增加而变化的情况。由图 4 可知, 除模型 1 外, 本文算法在比特率较小时, RMSE 指标表现一般, 处于中等水平, 但随着比特率的增大, 本文算法的 RMSE 指标过渡到最优位置, 这显示出了本文算法在大比特流时的 3D 模型重构性能的优势。LSM 算法的重构性能表现最差。在比特流较小时, CSGFT 算法略优于本文算法, 但在大比特流时, CSGFT 算法的重构性能显著下降。比特流较大时, 其他几种对比算法的性能也弱于本文算法。

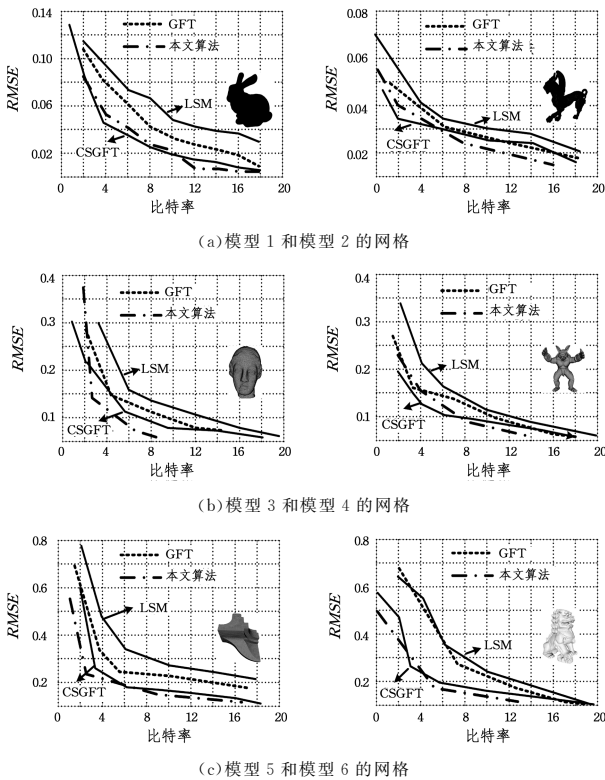


图 4 指标结果对比

Fig. 4 Comparison of index results

不同模型下对比算法的比特率对比数据如表 2 所列。

表 2 比特率指标对比

Table 2 Comparison of bit rate index

| 模型 | GFT | LSM | CSGFT | 本文算法 |
|------|------|------|-------|------|
| 模型 1 | 19.9 | 16.8 | 19.7 | 15.3 |
| 模型 2 | 15.7 | 14.5 | 16.9 | 13.1 |
| 模型 3 | 14.6 | 15.2 | 19.3 | 8.3 |
| 模型 4 | 16.8 | 15.7 | 18.1 | 13.1 |
| 模型 5 | 12.3 | 10.5 | 18.4 | 9.2 |
| 模型 6 | 18.2 | 14.8 | 21.7 | 12.5 |

根据表 2 中的对比数据可知,本文算法优于选取的 3 种对比方法。例如,对于模型 3,GFT 算法的比特率指标是 14.6,LSM 算法的比特率指标是 15.2,CSGFT 算法的比特率指标是 19.3,相比之下,本文算法分别降低了 42.2%,44.3%和 56.4%。以上结果表明,所提算法具备更好的 3D 模型压缩性能。

表 3 列出了几种对比方法的重构误差,这里的重构误差采用的是重构错误像素占比。

表 3 重构误差对比

Table 3 Contrast of reconfiguration error

(单位:%)

| 模型 | GFT | LSM | CSGFT | 本文算法 |
|------|------|------|-------|------|
| 模型 1 | 16.3 | 18.2 | 14.1 | 10.4 |
| 模型 2 | 18.9 | 20.5 | 17.6 | 15.7 |
| 模型 3 | 16.8 | 19.3 | 15.2 | 13.6 |
| 模型 4 | 19.6 | 22.3 | 17.5 | 14.8 |
| 模型 5 | 21.6 | 24.8 | 19.7 | 17.2 |
| 模型 6 | 16.7 | 17.8 | 14.2 | 13.5 |

根据表 3 中的对比数据可知,本文算法优于选取的 3 种对比方法。例如,对于模型 4,GFT 算法的重构误差指标是 19.6%,LSM 算法的重构误差指标是 22.3%,CSGFT 算法的重构误差指标是 17.5%,相比之下,本文算法分别降低了

24.5%,33.6%和 15.4%。以上结果表明,所提算法具备更好的 3D 模型重构性能。

表 4 列出了几种对比方法的计算时间。

表 4 计算时间

Table 4 Computing time

(单位:s)

| 模型 | GFT | LSM | CSGFT | 本文算法 |
|------|------|------|-------|------|
| 模型 1 | 25.4 | 34.6 | 41.7 | 24.8 |
| 模型 2 | 33.8 | 38.6 | 48.1 | 27.5 |
| 模型 3 | 19.5 | 23.6 | 31.7 | 20.4 |
| 模型 4 | 29.1 | 34.2 | 53.5 | 25.6 |
| 模型 5 | 9.2 | 12.7 | 18.2 | 8.1 |
| 模型 6 | 5.6 | 8.1 | 9.3 | 5.2 |

由表 4 可知,本文算法的计算速度显著快于 GFT 算法、LSM 算法以及 CSGFT 算法。根据前面的实验结果,CSGFT 算法的压缩精度优于 GFT 算法和 LSM 算法,但其比特率指标结果最大,这表示其压缩比例较低,因此执行速度较慢,而本文算法同时兼顾了算法精度和压缩比例,具有相对更高的重构精度和计算效率。

结束语 本文贡献可概括如下:1)提出了一种快速且高效的基于 CS 的 3D 模型的压缩方法,可获得较高的压缩比,并且不会导致显著的视觉质量损失。2)提出了基于模型贝叶斯学习的三维网格压缩应用。利用幂律衰减结构的图傅里叶变换(GFT),在 GFT 域拟合欧氏投影坐标数值的指数衰减系数模型。3)通过最大限度地减少内向的正常值,对各子网局部结构进行平滑。仿真结果显示,所提方法具有相对较高的比特率压缩指标以及较低的重构误差,计算效率明显提高。下一步研究方向为:1)对贝叶斯算法进行改进,以提高重构效果;2)对算法进行优化,进一步提升计算效率;3)进行测试系统开发,验证真实环境下的算法性能,并为后续算法设计搭建实验平台。

参考文献

[1] LI Z Z,WANG Z H,YAO L,et al. Parallel Multigrid Solver for Unstructured Grid[J]. Journal of Software, 2013, 24(2): 391-404. (in Chinese)
李宗哲,王正华,姚路,等. 非结构网格的并行多重网格解算器[J]. 软件学报,2013,24(2):391-404.

[2] LIU Q,YANG K,FU Y C,et al. An Algorithm for Hole Recovery in Wireless Sensor Networks Based on Triangle Net[J]. Acta Electronica Sinica, 2013, 41(2): 209-213. (in Chinese)
刘全,杨凯,伏玉琛,等. 一种三角形网格空洞修复算法[J]. 电子学报,2013,41(2):209-213.

[3] ZHANG C,GU Y Z,HU K L. Face recognition using SIFT features under 3D meshes[J]. Journal of Central South University, 2015, 22(5): 1817-1825.

[4] BAKHVALOV P A,KOZUBSKAYA T K. Cell-centered quassione-dimensional reconstruction scheme on 3D hybrid meshes [J]. Mathematical Models and Computer Simulations, 2016, 8(6): 625-637.

[5] LAVOU A M,GUILLAUME F D,HUDELOT C. 3D Mesh Compression: Survey, Comparisons, and Emerging Trends[J]. Acm Computing Surveys, 2015, 47(3): 1-41.

[6] ALLIEZ P,DESBRUN M. Valence-driven connectivity encoding for 3D Meshes[J]. Computer Graphics Forum, 2001, 20(3): 480-489.