计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(12):168−176 [doi: 10.15888/j.cnki.csa.010038] [CSTR: 32024.14.csa.010038]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于八叉树分块和顶点划分策略的加密 3D 网格模型可逆数据隐藏①

季云飞[1], 丁　睿[2]

[1](南京信息工程大学 软件学院, 南京 210044)
[2](南京信息工程大学 计算机学院、网络空间安全学院, 南京 210044)
通信作者: 季云飞, E-mail: 202212210033@nuist.edu.cn

摘　要: 加密域可逆数据隐藏 (reversible data hiding in encrypted domain, RDHED) 技术可在保护载体隐私的同时嵌入秘密信息, 但当前针对 3D 网格模型的 RDHED 方法普遍面临嵌入容量低的难题. 针对这一问题, 提出了一种基于八叉树分块和顶点划分策略的加密 3D 网格模型可逆数据隐藏方法. 首先, 采用八叉树结构将模型自适应地划分为不重叠子块, 保留块内空间相关性; 其次, 设计基于顶点熵的划分策略, 精确选取参考顶点以提升预测精度; 最后, 采用自适应 MSB (most significant bit) 预测方法, 最大化每个顶点的可嵌入空间, 从而显著提升嵌入容量. 实验结果表明, 该方法在提高 3D 网格模型嵌入容量的同时, 确保了数据的可逆性与可分离性, 为 3D 模型的可逆数据隐藏提供了一种有效的解决方案.

关键词: 可逆数据隐藏; 加密域; 3D 网格模型; 八叉树; 自适应预测

引用格式: 季云飞,丁睿.基于八叉树分块和顶点划分策略的加密 3D 网格模型可逆数据隐藏.计算机系统应用,2025,34(12):168–176. http://www.c-s-a.org.cn/1003-3254/10038.html

Reversible Data Hiding in Encrypted 3D Mesh Model Based on Octree Partitioning and Vertex Segmentation Strategy

JI Yun-Fei[1], DING Rui[2]

[1](School of Software, Nanjing University of Information Science & Technology, Nanjing 210044, China)
[2](School of Computer Science & School of Cyber Science and Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract: Reversible data hiding in the encrypted domain (RDHED) enables the embedding of secret data without revealing the privacy of the carrier. However, in view of the generally low embedding capacity of existing RDHED methods for 3D mesh models, this study proposes a reversible data hiding method for encrypted 3D mesh models based on octree partitioning and a vertex segmentation strategy. Firstly, the model is adaptively divided into non-overlapping blocks, with intra-block spatial correlation preserved. Then, a partitioning strategy based on vortex entropy is designed to accurately select reference points for prediction accuracy enhancement. Finally, an adaptive most significant bit (MSB) prediction method is employed to maximize the embeddable space of each vortex and thus notably improve the embedding capacity. Experimental results demonstrate that the proposed method improves the embedding capacity of 3D mesh models while ensuring data reversibility and separability, thus offering an effective solution for reversible data hiding in 3D models.

Key words: reversible data hiding (RDH); encrypted domain; 3D mesh model; octree; adaptive prediction

① 基金项目: 江苏省研究生科研与实践创新计划 (SJCX25_0523)
收稿时间: 2025-05-26; 修改时间: 2025-06-24, 2025-07-21; 采用时间: 2025-07-29; csa 在线出版时间: 2025-11-04
CNKI 网络首发时间: 2025-11-05

1 引言

随着大数据、云计算和人工智能技术的快速发展, 数据已成为数字经济中的关键资产. 然而, 数据规模的快速扩大也伴随着越来越严峻的网络安全风险, 尤其是个人信息泄露和非法数据交易的隐患日益凸显. 因此, 确保数据在传输、存储和处理过程中的安全性、完整性和隐私保护, 成为全球范围内的重要课题.

针对这一挑战, 可逆数据隐藏 (reversible data hiding, RDH)[1–3]技术应运而生. RDH 能够在数字载体中嵌入额外信息, 并在信息提取后实现载体的无损恢复, 因而被广泛应用于医学成像、军事通信等对数据准确性和完整性要求极高的领域. 早期 RDH 技术主要针对图像数据展开研究, 逐步形成了无损压缩[4,5]、直方图平移[6]、预测误差扩展[7–9]和多直方图修改[10–12]等方法. 然而, 传统 RDH 技术通常假设载体图像为明文状态, 这无法满足军事、医疗和云存储等隐私敏感领域对数据隐私保护的需求. 因此, 加密域可逆数据隐藏技术成为研究热点.

RDHED 结合数据隐藏与加密技术, 通过在加密载体中嵌入秘密数据, 确保了在数据传输和存储阶段的隐私保护与信息安全. 近年来, 图像领域的 RDHED 方法已取得显著进展, 并在嵌入容量、可逆性和视觉质量方面接近理论最优性能[13–20]. 其中, 加密后腾出空间 (vacating room after encryption, VRAE) 方案和加密前预留空间 (reserving room before encryption, RRBE) 方案成为主流研究方向. VRAE 方案直接对图像进行加密后再嵌入信息, 但因加密操作破坏了图像空间相关性, 其嵌入容量通常受限[13–15]. RRBE 方案则在图像加密前利用空间相关性进行预处理, 预留出足够的空间用于数据嵌入, 显著提高了容量[16–20].

尽管图像 RDHED 方法已较为成熟, 但这些方法难以直接迁移到其他媒体, 如音频、视频, 以及近年来应用迅速增长的 3D 网格模型中. 与规则的二维图像不同, 3D 网格模型具有不规则的点面拓扑结构, 且顶点空间分布不均匀, 这使得已有 RDHED 方法在 3D 模型中嵌入容量有限、预测精度不足的问题尤为突出[21–27]. 例如, Jiang 等[21]首先尝试将图像 RDHED 技术扩展到 3D 网格模型中, 但其方案的嵌入性能远低于当前图像领域的先进技术. 此后, Tsai 等[22]、Xu 等[23]及 Yin 等[24]分别提出了空间编码、多 MSB 预测误差嵌入等方法, 进一步提升了 3D 模型 RDHED 的性能, 但顶点预测精度不足、顶点利用率低等问题仍未得到有效解决[25–27].

为了进一步突破 3D 网格模型加密域可逆数据隐藏嵌入容量低、预测精度不足的瓶颈, 本文提出一种基于八叉树分块和顶点划分策略的加密 3D 网格模型可逆数据隐藏方法. 具体而言, 本研究首先通过八叉树结构将 3D 模型自适应划分为空间相关性强的子块; 其次, 提出一种基于顶点熵值的划分策略, 精准识别适用于高精度预测的参考顶点; 最后, 采用自适应的 MSB 预测方案以最大化嵌入容量. 实验表明, 本方法显著提升了加密 3D 网格模型中的嵌入容量, 并保证了数据提取和模型恢复的完全可逆性与可分离性. 本文的主要贡献如下.

1) 提出一种基于八叉树的分块策略, 将模型自适应地划分为不重叠的子块. 此策略可以自动将相邻顶点收集到一个块中, 更好地利用了相邻顶点的空间相关性.

2) 提出了一种顶点划分策略, 通过计算顶点熵值, 为每一个子块搜寻最适合预测的参考顶点, 大幅提升了预测精确度.

3) 提出的方法可以实现加密 3D 模型的可分离和高嵌入性能. 与同类型的方法相比, 所提出的方法可以实现无损模型恢复、无误差数据提取以及最优的嵌入性能.

2 方案设计

2.1 方案总体框架

本节介绍了所提出方案的总体框架, 并详细描述实现细节.

所提出方法的框架如图 1 所示. 它涉及 3 个角色: 内容拥有者、数据隐藏者和接收者. 整个过程分为 5 个阶段: 预处理、空间预留、加密、数据隐藏、数据提取和图像重构.

在预处理阶段, 模型拥有者先平移原始模型, 使其处于第 I 卦限内, 然后对模型的所有顶点做整数变换. 接着, 通过八叉树分块策略把模型分为多个子块. 八叉树结构可以自适应地对三维空间进行递归划分, 使得空间中顶点密集的区域被更细致地划分, 从而最大限度地保留块内顶点的局部空间相关性. 与传统的固定网格划分相比, 八叉树分块方式能更有效地支持高精度的预测, 从而提升 MSB 预测准确度与嵌入容量. 在

每个子块中, 顶点的空间分布和几何变化复杂程度各异. 为了选择更稳定、可预测的参考点, 本文基于顶点的局部熵值进行划分. 低熵值表示顶点周围变化较小, 具有较强的结构可预测性, 因此被优先选作预测参考点, 从而提升整体嵌入过程的准确性和稳定性. 完成对原始模型的预处理后, 通过自适应 MSB 预测计算出每个嵌入顶点的标签. 根据所有标签的出现频率, 使用哈夫曼编码来压缩标签长度得到预留空间的 3D 网格模型. 为了保护原始 3D 网格模型在数据隐藏前的隐私性, 本文采用基于流密码的加密机制对模型进行处理. 具体而言, 模型拥有者首先使用加密密钥初始化一个伪随机数生成器, 生成与模型顶点坐标数目相同长度的伪随机比特流. 随后, 将该比特流与 3D 模型中所有顶点的坐标数据逐位进行异或运算, 实现顶点级加密. 得到加密后的 3D 网格模型, 并发送给数据隐藏者.

数据隐藏者在接收到加密模型后, 通过提取辅助信息中的预留空间坐标来确认该信息预留空间的起始位置. 在读取定位信息后, 数据隐藏过程基于预测误差与 MSB 相似度策略进行. 通过计算嵌入顶点与其参考顶点间的预测误差, 判断可嵌入的 MSB 位数, 数据隐藏者将加密的秘密数据嵌入到保留的空间中, 从而生成标记的加密图像, 并将标记的加密模型发送给接收者.

在恢复阶段, 授权接收者能够使用相应的密钥单独执行数据提取或模型重构.

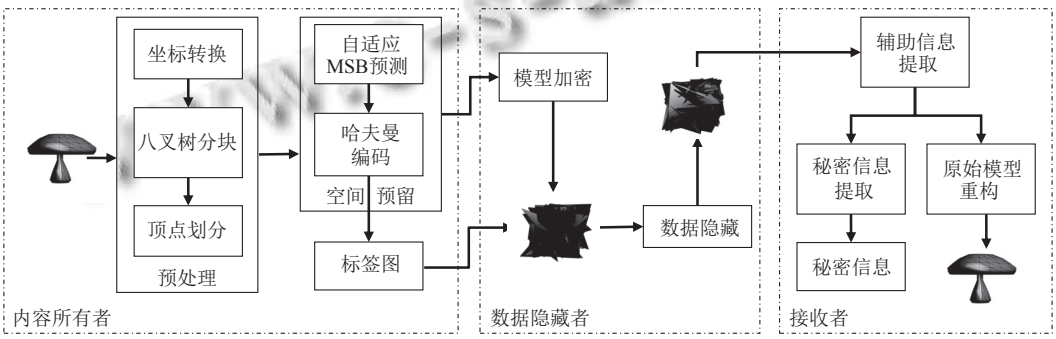


图 1 方案总体框架图

2.2 预处理

2.2.1 坐标转换

为了方便八叉树分块和空间预留, 使用坐标变换将 3D 模型 M 中顶点的十进制坐标值转换为正整数. 首先, 使用式 (1) 计算所有顶点中每个轴坐标的最小值.

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} \min_{i\in\{1,2,\cdots,p\}}(v_{i,x}) \\ \min_{i\in\{1,2,\cdots,p\}}(v_{i,y}) \\ \min_{i\in\{1,2,\cdots,p\}}(v_{i,z}) \end{bmatrix} \tag{1}$$

然后, 为了将每个顶点的坐标值落在 0–1 的范围内, 坐标变换过程由式 (2) 给出:

$$\bar{v}_i = \begin{bmatrix} \bar{v}_{i,x} \\ \bar{v}_{i,y} \\ \bar{v}_{i,z} \end{bmatrix} = \begin{bmatrix} (v_{i,x}-x_m)/10^{\alpha} \\ (v_{i,y}-y_m)/10^{\alpha} \\ (v_{i,z}-z_m)/10^{\alpha} \end{bmatrix} \tag{2}$$

其中, α 表示所有位移中移位的顶点坐标值中最长的整数. 于是所有顶点坐标都被平移至第 I 卦限. 为了方便后续的处理, 接下来使用式 (3) 将浮点型坐标转化为整数型.

$$\tilde{v}_i = [\bar{v}_i \times 10^{\beta}] \tag{3}$$

其中, β 表示截断精度. 最后, 在恢复阶段, 接收者可以通过式 (4) 来恢复平移后的坐标$\tilde{v}'_i$:

$$\tilde{v}'_i = \tilde{v}_i/10^{\beta} \tag{4}$$

2.2.2 八叉树分块

完成对所有顶点坐标转换后, 接着使用八叉树分块策略递归地把原始模型分为多个子块. 用八叉树的叶子节点来存储划分的子块, 分块步骤如算法 1 所示.

算法 1. 八叉树划分

输入: 顶点集为 *vertex*, 面集为 *face* 的原始模型 M, 中止参数 $D_{\max}$ 和 $S_{\max}$.
输出: 八叉树 *octree*.

1. 初始化八叉树节点 *octree*, *children*←null, 存储 *vertexIndex*;
2. **if** *depth*<$D_{\max}$ 且 *size*(*vertex*)>$S_{\max}$
3. 计算当前边界 *box*;
4. 生成 8 个子边界 *child_box*;
5. 初始化 *i*=1;
6. **while** *i*<9 **do**
7. *cb*←*child_box*{*i*};
8. 搜索子块内顶点;
9. 记录原始索引 *subVertexIndex*=*vertexIndex*(*idx*);

```
10.       if subvertex!=null
11.           递归生成子节点;
12.       end if
13.   end while
14. end if
15. Return octree
```

步骤 1: 初始化八叉树. 首先读取顶点$\tilde{v}_i$、面信息, 记录当前递归深度 D, 并初始化叶子节点列表.

步骤 2: 设置递归终止条件. 最大树深度 $D_{\max}$ 和最大子块顶点数 $S_{\max}$. 由于 D 决定了八叉树的时间复杂度和空间复杂度, 每增加一层深度, 存储需求都会呈指数增长. 因此, 需要设置 $D_{\max}$ 来控制最大深度. 同样的, 如果块内顶点数量较多, 说明此时仍有继续划分的必要. 于是, 使用 $S_{\max}$ 来控制块内顶点数量.

步骤 3: 检查递归条件. 如果当前区域的顶点数超过 $S_{\max}$, 且深度未超过 $D_{\max}$, 那么继续划分. 否则终止递归, 该节点成为叶子节点.

2.2.3 顶点划分

受图像信息熵的启发, 由于平滑区域的像素值变化较小, 相邻像素的颜色或灰度值接近, 进而熵值通常会更小. 在平滑区域, 像素值可以被较少的信息精确描述 (例如, 平均值或简单插值即可重构), 信息冗余较高. 因此, 本文设计了一种基于熵值计算的顶点划分策略. 顶点$\tilde{v}_i$的熵值计算过程为: 设顶点$\tilde{v}_i=(\tilde{v}_{i,x},\tilde{v}_{i,y},\tilde{v}_{i,z})$以及与$\tilde{v}_i$欧氏距离最小的 k 个邻居构成的邻居集合$\tilde{v}_i_neighbour=\{\tilde{v}_i_n_1,\tilde{v}_i_n_2,\cdots,\tilde{v}_i_n_k\}$. $\tilde{v}_i$的熵$H_{\tilde{v}_i}$为: 其邻居集$\tilde{v}_i_neighbour$在 x、y、z 这 3 个方向的分布复杂度. 式 (5) 给出了$H_{\tilde{v}_i}$的计算方法.

$$\begin{cases} H_{\tilde{v}_i}=\dfrac{H_x+H_y+H_z}{3} \\ H(x)=-\sum\limits_i p(x_i)\log_2 p(x_i) \\ H(y)=-\sum\limits_i p(y_i)\log_2 p(y_i) \\ H(z)=-\sum\limits_i p(z_i)\log_2 p(z_i) \end{cases} \tag{5}$$

最终, 为块内所有顶点计算熵值, 并得到熵值最小的顶点, 记为$\tilde{v}_i_hmin$. 根据熵值的定义, 此时顶点$\tilde{v}_i_hmin$为块内坐标变化程度最小的顶点, $\tilde{v}_i_hmin$即为当前子块的参考顶点, 其余顶点皆为嵌入顶点. 针对其他子块, 用相同的方法划分出嵌入顶点和参考顶点.

另外, 为了在恢复阶段可以正确地提取数据和模型重构, 需要区分参考顶点和嵌入顶点. 根据原始 OFF 文件中的顶点索引, 分别用 1 和 0 来标记参考顶点和嵌入顶点, 并将所有标签依次记录在 location map 中. 显然, 每个块只有一个参考顶点, 因此 location map 中存在大量连续的 0 和少量的 1. Location map 具有较强的稀疏性, 可以用算术编码来压缩.

2.3 空间预留

2.3.1 自适应 MSB 预测

假设在第 i 个子块中, 参考顶点和嵌入顶点分别记为$v_{i,r}$和$v_{i,e}$. 式 (6) 为自适应 MSB 预测的过程, 从 MSB 到 LSB 顺序比较$v_{i,j}^{t\text{MSB}}$和$v_{i,e}^{t\text{MSB}}$的每个比特, 直到某个比特不同.

$$\begin{cases} \arg\max\limits_t v_{i,j}^{t\text{MSB}}=v_{i,e}^{t\text{MSB}}, \quad t=1,2,\cdots,l \\ \text{s.t. } v_{i,r}^{t\text{MSB}}=\left\lfloor v_{i,r}/2^{l-t}\right\rfloor \bmod 2 \end{cases} \tag{6}$$

其中, l 表示二进制坐标的长度. 为了缩短标签长度, 取子块内所有的嵌入顶点的 t 值中的最小值$t_{\min}$作为子块的可嵌入长度. 同样地, 用上述的方法, 可以计算出所有子块的可嵌入长度. 图 2 是 x 轴上块内的自适应 MSB 预测误差标签示例. 以索引为 44 的顶点为参考顶点, 块内剩余顶点为嵌入顶点. 首先, 将浮点型顶点坐标转化为整数型. 以 x 轴坐标为例, 0.89 被转换为 8 900, 用二进制表示为: 0010001011100100. 类似地, 将所有嵌入顶点放大后表示为二进制的形式. 接下来, 用式 (4)–式 (6) 来预测该块的 MSB 值. 显然, 嵌入顶点从第 6 位 MSB 位开始与参考顶点不同, 于是该字块的 MSB 标签映射为 6. 在这种情况下, 该子块在 x 轴方向上每个顶点可以预留 6 比特的空间, 即t_x的值为 6. 同样地, 可以依次计算出t_y和t_z的值. 为了进一步缩短编码长度, 将t_x、t_y和t_z中的最小值记为$t_{\min}$, 作为该子块的嵌入长度.

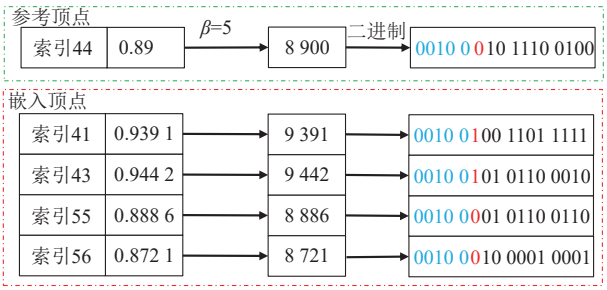


图 2 x 轴上块内的自适应 MSB 预测误差标签示例

2.3.2 自适应哈夫曼编码

根据第 2.3.1 节, 自适应 MSB 预测长度可以作为

标签. 然后, 标签可以被转换为二进制序列作为辅助信息, 并嵌入到加密的 3D 模型中. 标签映射的目的是保证原始 3D 模型可以无损恢复. 对于各种标签, 可以考虑用哈夫曼编码来记录标签图, 这可以大大压缩标签图的大小. 哈夫曼编码的核心思想是: 使用较短的码字来表示频率更高的标签. 以 mushroom 模型为例, 使用哈夫曼编码对 MSB 标签进行编码, 如表 1 所示, 其中 10 和 01 分别表示出现频率最高的 18 和 20.

表 1 Mushroom 的标签分布和哈夫曼编码

标签	17	18	19	20	21	22
频率	3	14	5	14	1	1
哈夫曼编码	000	10	110	01	1110	1111

2.4 模型加密与数据隐藏

预留空间后, 通过加密保护原始模型的隐私. 本方案使用最常用的加密方法, 即流密码加密. 首先, 通过式 (7) 将所有坐标转换为二进制比特流.

$$b_{i,j,k} = \left\lfloor \bar{v}_{i,j} / 2^k \right\rfloor \bmod 2,\ k = 0,1,\cdots,l-1 \tag{7}$$

其中, $b_{i,j,k}$ 表示转换后的比特流. 然后, 使用加密密钥 K_e 生成与原始图像大小相同的伪随机序列 P. 接下来, 使用式 (7) 将伪随机序列转换为二进制序列. 如式 (8) 所示, 逐比特 XOR 加密.

$$e_{i,j,k} = b_{i,j,k} \oplus P_{i,j,k},\ k = 0,1,\cdots,l-1 \tag{8}$$

其中, $e_{i,j,k}$ 是生成的加密序列, 并且$\oplus$是 XOR 运算. 最终, 加密的 3D 网格模型坐标计算如下:

$$v''_{i,j} = \sum_{k=1}^{l-1} e_{i,j,k} \cdot 2^k \tag{9}$$

在图像加密之后, 接收到的数据为加密的载体模型组成. 在数据隐藏阶段, 数据隐藏者首先提取哈夫曼映射图, 然后通过解码映射编码, 获得每个子块的嵌入空间. 接着, 通过像素替换将 K_d 加密过的秘密数据嵌入到顶点中. 最后, 生成具有秘密数据的加密模型.

2.5 模型重构与信息提取

接收者在接收到标记的加密模型后, 根据其所持有的密钥进行数据提取或模型重构. 因此, 以下讨论 3 种可能的情况.

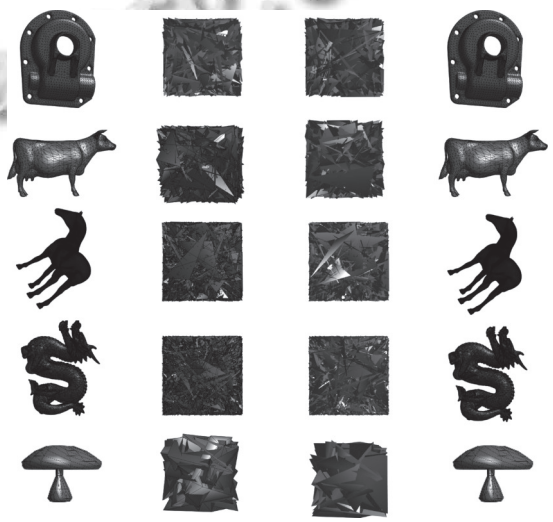
情况 1: 接收者持有加密密钥 K_e. 接收者首先根据加密密钥 K_e 解密出嵌入秘密数据的比特流. 然后, 通过 location map 定位出所有参考顶点. 接着, 通过解码哈夫曼标签图, 得到每个子块的可嵌入长度 t, 并把嵌入像素的前 t 位 MSB 替换为参考顶点的前 t 位. 最终, 获得了恢复的模型 M.

情况 2: 接收者持有数据隐藏密钥 K_d. 如果接收方只拥有数据隐藏密钥 K_d, 则可以首先通过 location map 定位出所有参考顶点. 然后, 提取哈夫曼编码, 然后恢复标签映射, 以获得子块的嵌入长度. 接着, 按照索引顺序连接所有嵌入顶点的可嵌入部分, 获得加密后的秘密信息. 最后, 通过持有的 K_d, 解密出原始信息.

情况 3: 接收方同时拥有数据隐藏密钥 K_d 和加密密钥 K_e. 接收方可以获得恢复的 3D 模型和秘密信息. 信息提取和模型恢复可以按任意顺序进行. 通过这种方式, 本方案是可分离的.

3 实验结果与分析

本方法在 Windows 11 操作系统下用 Matlab R2016a 实现. 最大树深 $D_{\max}$ 和最大子块顶点数 $S_{\max}$ 分别设置为 6 和 40. 图 3 中的原始模型为本方案中的实验中使用的 3D 网格模型. 其中 dragon 模型来自 Stanford 3D Scanning Repository, 其余模型均为公开可获取或文献常用标准模型. 另外, 除了原始模型, 加密模型、标记加密模型和重构模型的视觉效果如图 3 所示. 首先, 介绍在 RDHED 中, 最主要的几个评价指标. 接着, 分析了本方法的可逆性. 最后, 将本方法与同类型的方法进行了比较, 以充分证明其优越性.



(a) 原始模型 (b) 加密模型 (c) 标记加密模型 (d) 恢复模型

图 3 5 个测试模型上每个阶段的仿真结果

3.1 评价指标

以 3D 网格模型为载体的 RDHED 方法主要关注

数据嵌入性能和重构模型质量. 衡量方法的嵌入性能由逐顶点比特数 (bit per vertex, bpv) 描述. 豪斯多夫距离 (Hausdorff distance, *HD*) 和信噪比 (signal to noise ratio, *SNR*) 用于评估重构模型的质量. *HD* 是描述两模型之间相似性的指标, *HD* 越小, 两组顶点越相似, 其定义由定义 1 给出.

定义 1. 设两个模型 A、B 的顶点集合表示为$A=\{a_1,a_2,\cdots,a_n\}$, $B=\{b_1,b_2,\cdots,b_n\}$. n 表示 3D 网格模型的顶点数量. A 与 B 的$HD(A,B)$定义为:

$$HD(A,B)=\max\left\{\max_{a\in A,b\in B}d(a,b),\max_{b\in B,a\in A}d(b,a)\right\} \tag{10}$$

其中, d 表示欧几里得距离. *SNR* 用来评估模型的几何失真, 较高的 *SNR* 值表示 3D 模型的几何变形较小. 式 (11) 给出了 *SNR* 的计算方法.

$$SNR=10\times\lg\frac{\sum_{i=1}^{p}[(v_{i,x}-\bar{v}_x)^2+(v_{i,y}-\bar{v}_y)^2+(v_{i,z}-\bar{v}_z)^2]}{\sum_{i=1}^{p}[(v'_{i,x}-v_{i,x})^2+(v'_{i,y}-v_{i,y})^2+(v'_{i,z}-v_{i,z})^2]} \tag{11}$$

其中, $\bar{v}_x$、$\bar{v}_y$、$\bar{v}_z$表示原始坐标的平均值, $v'_{i,x}$、$v'_{i,y}$、$v'_{i,z}$表示重构模型的坐标.

3.2 参数取值分析

本节主要分析截断系数 β 的取值. 在预处理阶段, 为了方便八叉树分块和空间预留, 针对浮点型坐标, 使用式 (3) 将其转化为整数型. 根据第 3.1 节的分析, 坐标被放大了 10^{β} 倍. 在恢复阶段, 部分小数被省略, 这就导致了恢复模型与原始模型的差异. 在坐标转换的过程中, β 取值越大, 保留的小数部分就越多, 恢复模型与原始模型的差距就越小; β 取值越小, 保留的小数部分就越少, 恢复模型与原始模型的差距就越大. 因此 β 的取值不仅影响嵌入容量, 还影响最终恢复模型的失真程度. 为了在嵌入性能和模型失真中取得平衡, 本节首先在 5 个不同的模型上分析 β 的取值对嵌入容量的影响. 图 4 是 5 个不同模型的 β 从 2 增加到 9 的嵌入率的变化趋势. 从图中可知, 当 β 从 2 增加至 3 时, 嵌入率有明显的增加. 当 β 增加至 4 时, 嵌入率急剧下降. 当 β=5 时, 此时嵌入率达到峰值. 当 β>5 并增加至 9 时, 此时嵌入率相应的下降. 由此可知, 当 β=5 时, 此时模型嵌入率最大.

在分析完 β 对嵌入容量的影响后, 接下来分析不同截断精度 β 值下 3D 模型的恢复质量. 图 5 为 5 个不同模型的 β 从 2 增加到 9 时, *HD* 和 *SNR* 的变化趋势. 从图中可知, 随着 β 的增加, 所有 5 个测试模型的信噪比逐渐向无穷大上升, *HD* 相应地向 0 下降. 这表明, 当 β 的值足够大以包含模型顶点坐标值的所有重要部分时, *SNR* 将达到无穷大, *HD* 将接近 0, 从而允许在本方法中对 3D 模型进行完整无损的可逆恢复.

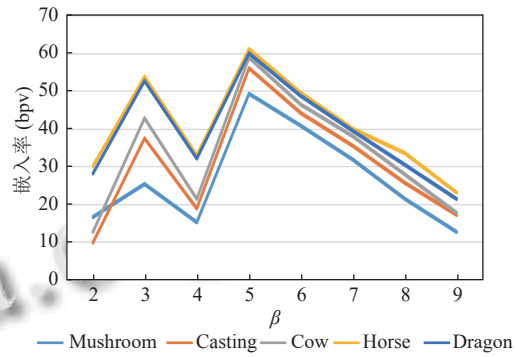
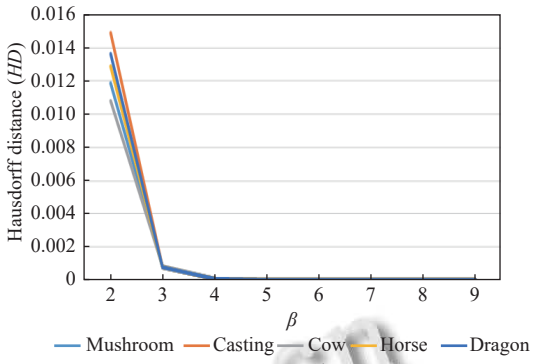
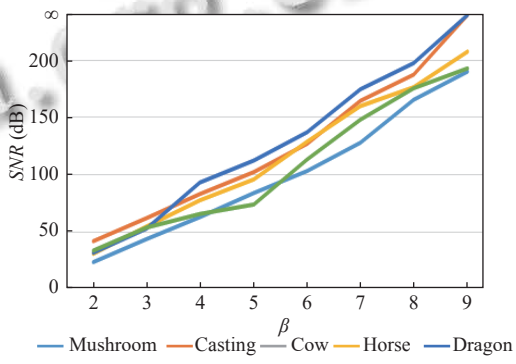


图 4 5 个测试模型上不同 β 对嵌入率的影响



(a) *HD*结果



(b) *SNR*结果

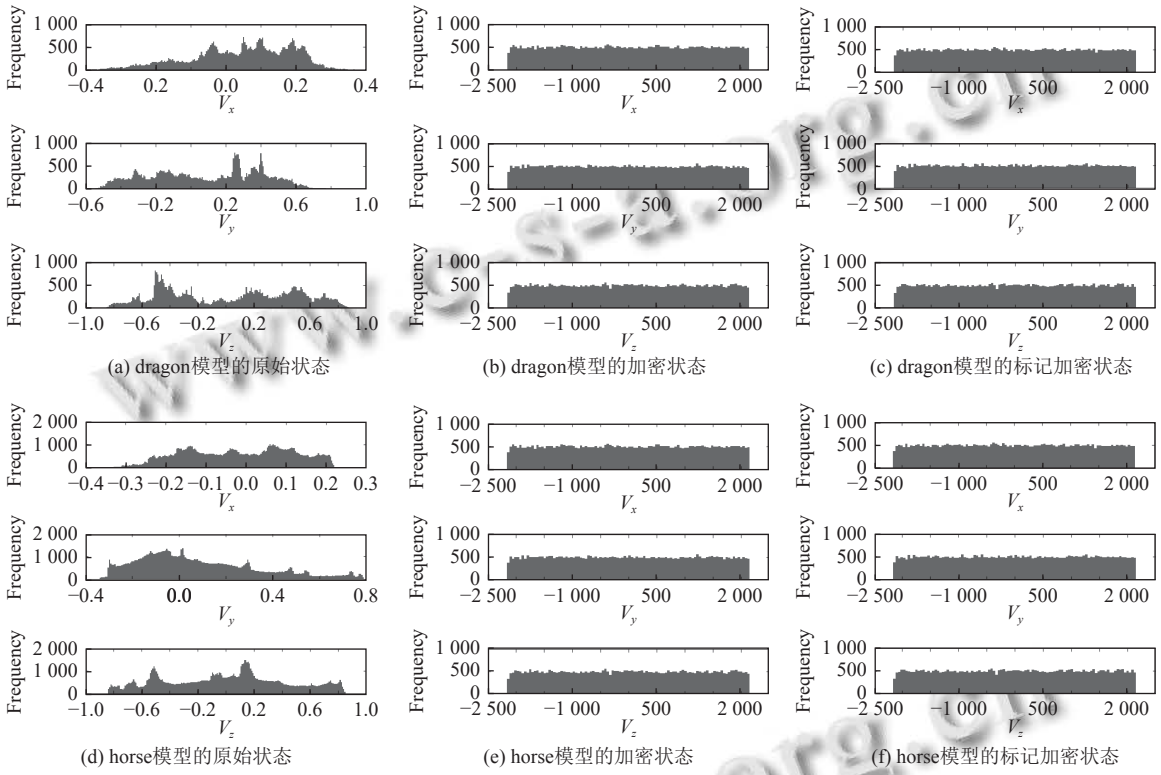
图 5 5 个测试模型上不同 β 的结果

通过对嵌入率、*HD*、*SNR* 的分析, 在 β=5 时, 平衡了嵌入容量和模型重构视觉质量这两个重要的评价指标, 因此 β 的最优取值为 5.

3.3 安全性分析

在分析嵌入取值后, 我们进一步探讨模型传输过程中的安全性. 从图 3 中的加密模型和标记加密模型

可知, 在视觉层面上, 原始模型经过加密和数据嵌入后, 已经转换成类噪声模型. 这意味着非授权用户无法得知与原始模型的任何信息. 接下来, 通过分析原始模型、加密模型和标记加密模型的顶点坐标分布来说明本方案的安全性. 由于篇幅限制, 图 6 选取了两个经典模型的顶点分布直方图. 图 6(a)–(f) 分别为模型 dragon 和 horse 在原始状态、加密状态与标记加密状态下的 $x/y/z$ 坐标分布直方图对比. 从图 6(a)、(d) 可以看出原始模型的直方图分布不均匀, 从模型中可以清晰地获得特征信息. 加密后的模型的直方图分布是均匀的, 这意味着加密模型不会提供任何有效信息. 在嵌入秘密数据后, 标记加密模型的直方图也是均匀的, 且明显不同于原始模型和加密模型. 实验结果从统计信息方面验证了本方法的安全性.



(a) dragon模型的原始状态 (b) dragon模型的加密状态 (c) dragon模型的标记加密状态

(d) horse模型的原始状态 (e) horse模型的加密状态 (f) horse模型的标记加密状态

图 6 测试模型的顶点分布直方图

3.4 嵌入性能分析

嵌入能力是 RDHED 方法的核心指标. 接下来将所提出的方法的性能与几项同类型的方法[23–27]进行了比较, 实验将截断系数 β 统一设置为 5. Xu 等[23]利用坐标 MSB 的强相关性来预留空间, 这种策略在一定程度上提高 ER. 然而, 他们的方法只使用最高位 MSB 来嵌入额外的数据, 同时需要大量的参考顶点来预测嵌入顶点. 每个顶点可嵌入的数据小于 3 比特. Yin 等[24]设计了基于多 MSB 相关性的方法. 将嵌入平面从最高位 MSB 扩展到多 MSB, 此方法结果在 16 bpv 左右. Lyu 等[25]改进了文献[24]的方法, 将奇数或偶数顶点分配为可嵌入的顶点, 从而实现了 50% 的顶点利用率. Tsai 等[26]的方法采用阈值随机选择可嵌入顶点周围的预测顶点, 顶点利用率和 ER 都随着阈值的变化而变化, 其平均嵌入率为 37.49 bpv. 本文所提出的方案在嵌入率方面相较于嵌入性能最优的方案[27]有明显的提升, 平均嵌入率提升 2.98 bpv. 图 7 使用条形图比较并显示了上述先进方法和同类型方法在 5 个测试模型上的 ER. 从图中可以明显看出, 在所有测试模型中, 所提出的方法在 ER 方面都优于其他方法, 另外对于具有大量顶点的模型 dragon 和 horse, 所提出的方案比性能最佳的方案高出 3.1 bpv 以上. 为进一步验证所提方法在公开标准数据集上的通用性与有效性, 另外在 PSB (princeton segmentation benchmark)[28]上进行了补充实验. PSB 包含大量拓扑结构各异、几何复杂性不同的 3D 网格模型, 被广泛应用于 3D 图形处理与模型分析研究中. 如图 7 所示, 在 PSB 数据库上, 本文方法的平均嵌入容量依然保持显著领先.

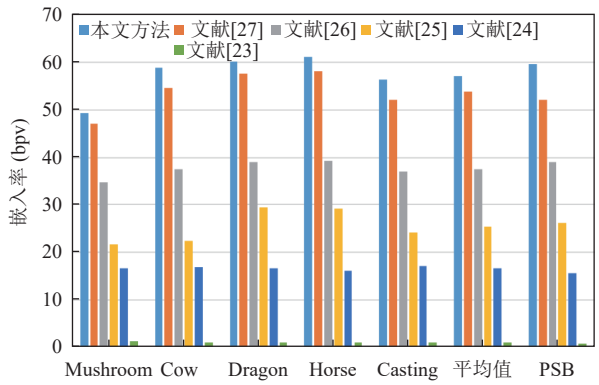


图 7 相似方法之间的嵌入性能比较

4 结束语

本文提出了一种基于八叉树分块和顶点划分策略的加密 3D 网格模型可逆数据隐藏方法. 通过模型分块, 充分利用了相邻顶点的空间相关性. 同时, 根据所提出的顶点划分策略, 为每一个子块划分出最适合用来预测的参考顶点, 大大提高了预测精度. 实验结果和分析表明, 本方法在保证可逆性与可分离性的同时, 提升了嵌入容量. 在未来的工作中, 针对子块顶点较少的情况, 我们将继续探索更高效的编码方式 (例如混合编码、自适应编码), 以进一步提升 3D 网格模型的数据嵌入能力.

参考文献

1 Ni ZC, Shi YQ, Ansari N, *et al*. Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology, 2006, 16(3): 354–362. [doi: 10.1109/TCSVT.2006.869964]

2 Shi YQ, Li XL, Zhang XP, *et al*. Reversible data hiding: Advances in the past two decades. IEEE Access, 2016, 4: 3210–3237. [doi: 10.1109/ACCESS.2016.2573308]

3 Shi YQ. Reversible data hiding. Proceedings of the 3rd International Conference on Digital Watermarking. Seoul: Springer, 2004. 1–12.

4 Celik MU, Sharma G, Tekalp AM. Lossless watermarking for image authentication: A new framework and an implementation. IEEE Transactions on Image Processing, 2006, 15(4): 1042–1049. [doi: 10.1109/TIP.2005.863053]

5 Zhang WM, Hu XC, Li XL, *et al*. Recursive histogram modification: Establishing equivalency between reversible data hiding and lossless data compression. IEEE Transactions on Image Processing, 2013, 22(7): 2775–2785. [doi: 10.1109/TIP.2013.2257814]

6 Coatrieux G, Pan W, Cuppens-Boulahia N, *et al*. Reversible watermarking based on invariant image classification and dynamic histogram shifting. IEEE Transactions on Information Forensics and Security, 2013, 8(1): 111–120. [doi: 10.1109/TIFS.2012.2224108]

7 Ou B, Li XL, Zhao Y, *et al*. Pairwise prediction-error expansion for efficient reversible data hiding. IEEE Transactions on Image Processing, 2013, 22(12): 5010–5021. [doi: 10.1109/TIP.2013.2281422]

8 He WG, Cai ZC. Reversible data hiding based on dual pairwise prediction-error expansion. IEEE Transactions on Image Processing, 2021, 30: 5045–5055. [doi: 10.1109/TIP.2021.3078088]

9 Bai YQ, Jiang GY, Zhu ZJ, *et al*. Reversible data hiding scheme for high dynamic range images based on multiple prediction error expansion. Signal Processing: Image Communication, 2021, 91: 116084. [doi: 10.1016/j.image.2020.116084]

10 Zhang TC, Hou TS, Weng SW, *et al*. Adaptive reversible data hiding with contrast enhancement based on multi-histogram modification. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(8): 5041–5054. [doi: 10.1109/TCSVT.2022.3146159]

11 Zhang C, Ou B. Reversible data hiding based on multiple adaptive two-dimensional prediction-error histograms modification. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(7): 4174–4187. [doi: 10.1109/TCSVT.2021.3125711]

12 Li X, Xiao MY, Li XL, *et al*. Matrix embedding based multiple histograms modification for efficient reversible data hiding. IEEE Signal Processing Letters, 2024, 31: 2555–2559. [doi: 10.1109/LSP.2024.3455995]

13 Zhang XP. Separable reversible data hiding in encrypted image. IEEE Transactions on Information Forensics and Security, 2012, 7(2): 826–832. [doi: 10.1109/TIFS.2011.2176120]

14 Zhang XP. Reversible data hiding in encrypted image. IEEE Signal Processing Letters, 2011, 18(4): 255–258. [doi: 10.1109/LSP.2011.2114651]

15 Hong W, Chen TS, Wu HY. An improved reversible data hiding in encrypted images using side match. IEEE Signal Processing Letters, 2012, 19(4): 199–202. [doi: 10.1109/LSP.2012.2187334]

16 Ma KD, Zhang WM, Zhao XF, *et al*. Reversible data hiding in encrypted images by reserving room before encryption. IEEE Transactions on Information Forensics and Security,

2013, 8(3): 553–562. [doi: 10.1109/TIFS.2013.2248725]

17 Puteaux P, Puech W. An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images. IEEE Transactions on Information Forensics and Security, 2018, 13(7): 1670–1681. [doi: 10.1109/TIFS.2018.2799381]

18 张敏情, 姜超, 狄富强, 等. 基于密码反馈秘密共享的大容量密文域可逆隐藏. 通信学报, 2023, 44(9): 48–57. [doi: 10.11959/j.issn.1000-436x.2023170]

19 Yao Y, Wang K, Chang Q, *et al*. Reversible data hiding in encrypted images using global compression of zero-valued high bit-planes and block rearrangement. IEEE Transactions on Multimedia, 2023, 26: 3701–3714.

20 Zhang XQ, He FY, Yu CQ, *et al*. Reversible data hiding in encrypted images with asymmetric coding and bit-plane block compression. IEEE Transactions on Multimedia, 2024, 26: 10174–10188. [doi: 10.1109/TMM.2024.3405717]

21 Jiang RQ, Zhou H, Zhang WM, *et al*. Reversible data hiding in encrypted three-dimensional mesh models. IEEE Transactions on Multimedia, 2018, 20(1): 55–67. [doi: 10.1109/TMM.2017.2723244]

22 Tsai YY. Separable reversible data hiding for encrypted three-dimensional models based on spatial subdivision and space encoding. IEEE Transactions on Multimedia, 2020, 23: 2286–2296.

23 Xu N, Tang J, Luo B, *et al*. Separable reversible data hiding based on integer mapping and MSB prediction for encrypted 3D mesh models. Cognitive Computation, 2022, 14(3): 1172–1181. [doi: 10.1007/s12559-021-09919-5]

24 Yin ZX, Xu N, Wang F, *et al*. Separable reversible data hiding based on integer mapping and multi-MSB prediction for encrypted 3D mesh models. Proceedings of the 4th Chinese Conference on Pattern Recognition and Computer Vision. Beijing: Springer, 2021. 336–348.

25 Lyu WL, Cheng LL, Yin ZX. High-capacity reversible data hiding in encrypted 3D mesh models based on multi-MSB prediction. Signal Processing, 2022, 201: 108686. [doi: 10.1016/j.sigpro.2022.108686]

26 Tsai YY, Liu HL. Integrating coordinate transformation and random sampling into high-capacity reversible data hiding in encrypted polygonal models. IEEE Transactions on Dependable and Secure Computing, 2023, 20(4): 3508–3519. [doi: 10.1109/TDSC.2022.3204291]

27 Hou GY, Ou B, Long M, *et al*. Separable reversible data hiding for encrypted 3D mesh models based on octree subdivision and multi-MSB prediction. IEEE Transactions on Multimedia, 2024, 26: 2395–2407. [doi: 10.1109/TMM.2023.3295578]

28 Chen X, Golovinskiy A, Funkhouser T. A benchmark for 3D mesh segmentation. ACM Transactions on Graphics, 2009, 28(3): 1–12. [doi: 10.1145/1531326.1531379]

(校对责编: 张重毅)