

参数平面二叉树剖分网格简化

关东东¹⁾ 关华勇¹⁾ 汪嘉业²⁾

¹⁾(山东大学数学与系统科学学院, 济南 250100) ²⁾(山东大学计算机科学与技术学院, 济南 250100)

摘要 为了快速地对 3 维网格模型进行简化, 提出了一种曲率自适应的 3 维网格简化算法, 该算法首先将原始网格投影至参数平面上, 并构造反映原始网格曲率分布的平面曲率灰度分布, 用以表征简化过程中对网格各部分不同的采样密度要求; 然后根据等曲率灰度分割的原则来对参数平面进行二叉树剖分, 以构造反映其不均匀分布的非均衡二叉树结构, 并依此选取简化后的网格顶点集合, 以构造简化的三角网格。该算法的优点是执行速度快, 同时在简化过程中仍能充分保持原始网格的细节。

关键词 网格简化 参数化 曲率

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2006)01-0089-07

Mesh Simplification Using Non-balance Binary Tree on Parameterization Plane

GUAN Dong-dong¹⁾, GUAN Hua-yong¹⁾, WANG Jia-ye²⁾

¹⁾(School of Mathematics and System Science, ShanDong University, Jinan 250100)

²⁾(School of Computer Science and Technology, ShanDong University, Jinan 250100)

Abstract A curvature adaptive algorithm is presented to simplify 3d meshes rapidly based on splitting of the parameterization plane. The original 3d mesh is mapped onto the parameterization plane. A planar gray-level distribution is constructed based on the curvature values of the vertices of the original mesh to proclaim the various sampling density requirements of the original mesh. An iteration algorithm is used to select the curvature adaptive resample vertices on the parameterization plane to construct the simplified mesh. The parameterization plane is iteratively split into two parts with the same summation of the gray-level value in each step. A non-balance binary tree was constructed during the procedure of splitting. The resample vertices set is obtained by allocating one vertex in each leaf node, and those resampled vertices are triangulated to construct the simplified mesh. This algorithm is very fast and can preserve the detail very well.

Keywords mesh simplification, parameterization, curvature

1 引言

随着 3 维扫描设备的广泛应用, 产生了顶点数目多达几十万乃至几百万的复杂 3 维网格模型。这种高密度的均匀采样网格模型, 一方面虽然保证了曲面的足够精度, 但另一方面却为此付出了顶点分布过于冗余的代价, 由于其对人机交互、计算机辅助设计、动画制作等应用形成了瓶颈, 因此必须对高密

度均匀采样的 3 维网格进行合理的简化。

众所周知, 3 维网格模型是对连续曲面的分段线性逼近, 其网格顶点是分布在连续表面上的离散采样点, 而 3 维扫描设备输出的高密度 3 维网格则是对连续曲面的均匀采样, 虽然其扫描精度足以保证曲面的细节部分有足够多的采样顶点, 但同时均匀采样的特性却致使曲面平坦部分出现过度采样, 造成大量冗余。网格简化的目的就是从高密度网格顶点集合中抽取出一个曲率自适应的非均匀采样的

收稿日期: 2005-04-25; 改回日期: 2005-09-16

第一作者简介: 关东东(1975 -), 男, 2000 年获山东大学硕士学位, 现为山东大学博士研究生。主要研究方向为计算机图形学、3 维网格处理、3 维动画。E-mail: ddguan2005@yahoo.com.cn

顶点子集,并赋予适当的三角化连接关系,以构成简化网格。与原始网格相比,简化网格的顶点集合在曲率大的细节部分可保持采样密度不变或略有降低,而在曲率小的平坦部分采样密度却大幅降低,其在采样顶点数目大为减少的同时仍能较好地反映其连续曲面的几何信息。

本文提出的 3 维网格简化算法的思路是,首先将原始网格投影至参数平面,然后根据顶点的曲率属性进行等权值剖分,以形成一种非平衡二叉树结构,用于表征网格各部分对采样顶点密度分布的不同要求,并据此来获取简化的采样顶点集合,其主要优点是执行速度快,并具有良好的曲率自适应特性,其在去除大量冗余采样顶点的同时,仍能较好地保持原始网格的细节特性。

反观现有的利用平面参数化来进行 3 维网格处理的算法,基本上都是通过对参数平面的均匀采样将 3 维数据转换为“2.5 维”数据,以便能借用数字图像处理现成算法,其中的关键是保持 3 维网格在参数平面投影分布的密度与对应的数字图像的均匀网格疏密相一致,但在参数化投影过程中出现的不可避免的几何形变将对数字图像的均匀网格的密度提出极高的要求,以至难以实现,为此,该类算法不得不对原始网格进行复杂的切割,并采用搜索式优化投影算法,最后还要进行繁琐的排料与切缝缝合过程,才能完成网格的处理过程,这些复杂的预处理与后处理过程使整个算法的时效性与稳定性都大为降低,必须付出高昂的代价才能完成对 3 维网格的处理。

而本文提出的参数平面二叉树剖分算法,由于摆脱了参数平面的均匀采样的要求和借用图像处理现成算法的束缚,因而并不在乎投影形变的程度,只需对原始网格进行简单切割,并可采用最为简单的参数化投影算法,而且完全无排料与缝合问题的困扰,并能在巨大投影形变存在时,仍保持对细节的描绘能力,这就大大提高了简化算法的时效性与稳定性。

2 相关工作

2.1 三角网格模型与网格参数化

三角网格可以形式上表示为 $M(K, P)$, 其中, 子集族 K 是一个单纯复合形, 用其即可描述三角网格顶点、边及三角面的连接关系, 并可确定三角网格

的拓扑连接; 顶点位置集合 $P = \{p_1, p_2, \dots, p_N\}$, $p_i \in \mathbb{R}^m$, 记录了网格顶点的坐标位置, 用其可确定三角网格的几何形状, 对于空间 3 维网格 $m=3$, 对于平面 2 维网格 $m=2$ 。单纯复合形 K 由顶点集合 V 、边集合 E 及三角面集合 F 3 个集合组成, 即 $K = \{V, E, F\}$ 。 $\{i, j\} \in E$, 表明顶点 i 与 j 连成一条边; $\{i, j, k\} \in F$, 表明顶点 i, j 与 k 连成一个三角面。并称与顶点 i 连成一条边的顶点为顶点 i 的邻接顶点, 而由顶点 i 的所有邻接顶点组成的集合则称为顶点 i 的邻接集合, 记为 $N(i) = \{j | (i, j) \in E\}$ 。

空间网格参数化是指将空间 3 维三角网格 $M(K, P)$ 映射为一个与之同构的 2 维平面三角网格 $\hat{M}(K, Q)$, 其中 2 维平面三角网格的顶点位置集合 $Q = \{q_1, q_2, \dots, q_N\}$, $q_i = (u_i, v_i) \in \mathbb{R}^2$ 。空间 3 维三角网格 M 与 2 维平面三角网格 \hat{M} 具有相同的拓扑连接 K , 二者的顶点位置集合 P 与 Q 之间存在一一映射 Φ , 映射 Φ 满足 $\Phi(p_i) = q_i$, 称之为参数化映射, $q_i = (u_i, v_i)$ 称之为空间 3 维网格顶点 i 的参数化点。通过参数化映射, 3 维网格的边界可被映射为平面网格的边界, 为处理方便, 平面参数化网格的边界一般取为单位正方形, 其参数化效果见图 1。除了简单的可展曲面以外, 由于一般 3 维网格的参数化都会引起距离或角度的形变, 因此空间网格参数化的关键是找出适当的参数化映射 Φ 。

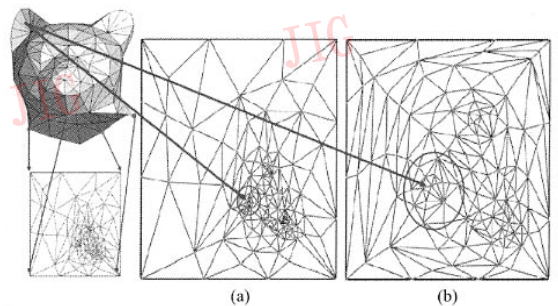


图 1 3 维网格在单位正方形上的参数化效果

Fig. 1 The Parameterization result of a 3d mesh on unit square

文献[1]中给出的 Shape-Preserve 参数化算法是将求解参数化映射的问题转化为线性代数方程组的求解问题, 但处理复杂网格时, 可能会出现极大的形变。为减少参数化形变, 文献[2]提出了基于试探性搜索的 L^2 -stretch 优化参数化算法, 虽然该算法能使全局形变指标达到某种最优, 但实现复杂, 速度

慢,且存在不收敛的可能。

2.2 网格切割

拓扑理论指出,任意封闭的3维网格都可以通过沿特定三角边链剪切的方法将其展开成为与圆盘同胚的网格,而用来剪切的三角边链则作为3维网格的边界投影至参数平面的单位正方形上。

2.3 几何图(geometry images)算法

文献[3]提出了几何图算法,该算法首先对3维网格的平面参数化网格进行均匀采样,然后将均匀采样点的 x, y, z 坐标视作 r, g, b 色彩分量,以构成利用彩色数字图像形式来表示3维信息的几何图(geometry images),再借用图像处理算法来进行3维动态显示加速和3维数据无损压缩以及纹理映射等处理。

文献[3]指出,均匀采样过程对由参数化投影引起的几何形变非常敏感,由于在形变大的细节部

分会出现欠采样与特征混叠的严重问题,从而使均匀采样结果无法逼近原始网格。为解决此问题,文献[3]采用了文献[2]中提出的试探性搜索优化参数化算法,以减少投影形变,并对封闭网格的切割路径提出了苛刻的定义,要求其通过网格中各个大曲率细节部分,以尽量延展投影边界和减少形变,但这两种预处理过程都是试探性搜索过程,不仅实现复杂,而且执行速度慢。

文献[4]提出的多片几何图算法对几何图算法进行了改进,即利用流形分析中的地图集(Atlas)理论,首先将原始网格切割成为多个子片(charts),然后分别对各子片进行平面投影,以保证各子片都近似实现保距投影(isometric),从而最大限度地减少了形变,但这又带来了各子片的不规则投影结构在参数平面上的排料问题与切割边界的缝合问题(如图2所示),而且二者都是难以解决的优化问题。

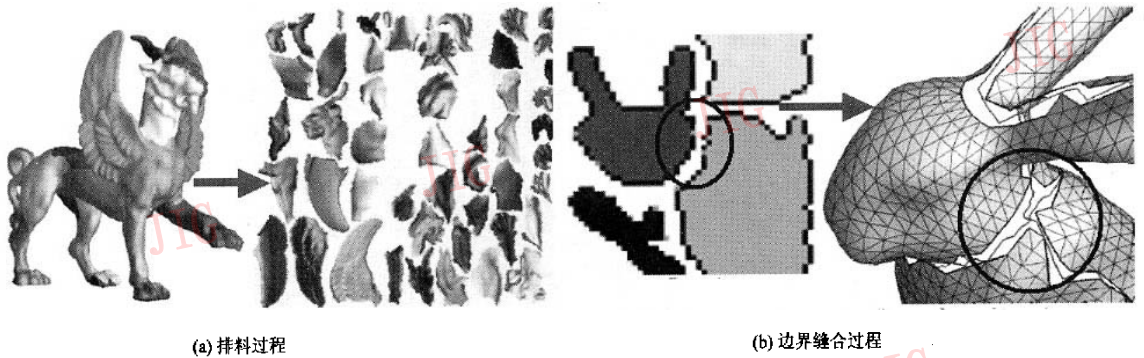


图2 多片几何图在算法中的复杂处理过程

Fig. 2 The complex process in Multi-Chart geometry images

3 参数平面二叉树剖分网格简化

本文提出的曲率自适应的网格简化算法分为以下两步:(1)利用网格参数化算法,首先将原始3维网格投影至参数平面上的单位正方形中,再将各顶点的曲率属性规一化为 $0 \sim 255$,并将其作为灰度属性赋给参数平面上对应的投影点,形成散乱平面曲率灰度分布,然后根据平面曲率灰度分布计算出参数平面上某一区域中曲率灰度总和,以体现网格中对应部分的采样密度要求,即曲率灰度和越大,表明该区域对应于网格中曲率大的细节部分,应在简化过程中应赋予较高的采样密度;曲率灰度和越小,表明该区域对应于网格中曲率小的平坦部分,在简化

过程中只需赋予较低的采样密度,最后将网格顶点的曲率属性映射为参数平面上的散乱灰度分布;(2)在平面曲率灰度分布的约束下,通过二叉树剖分,在均匀采样的原始网格顶点集合中获取一个体现曲率自适应特性的非均匀分布的子集来构造出简化网格。

为实现第2步,本文利用“分而治之”的思想,即根据平面曲率灰度分布,将参数平面一分为二,并使每一部分中的曲率灰度总和相等,即每一部分应具有大致相同的采样密度要求,虽然两部分各自所对应的原始网格顶点数目不同,但应分别从中抽出数目相等的顶点以构成简化网格。按照上述等灰度和剖分的原则来对参数平面不断进行一分为二剖分,直至每一剖分子块中的曲率灰度总和小于预先

设定的阈值为止,在此剖分过程中就构成了一个非均衡二叉树结构,树中的各叶节点对应剖分结束后形成的各剖分子块。由于平面灰度分布反映了原始网格对应部分的曲率分布,因此二叉树中各叶节点的深度也各不相同,深度大的叶节点对应曲率较大的细节部分,而深度小的叶节点则对应曲率较小的平坦部分。这样如果在与非均衡二叉树的每一个叶节点对应的剖分子块中选取一个新采样顶点,则其组成的采样顶点集合的分布就与平面曲率的灰度分布相匹配,也即与原始网格顶点的曲率分布相适应,若为其赋予合适的三角化连接关系,即可构成曲率自适应的简化网格。

下面详细叙述算法过程,为方便叙述,本文只讨论最为常见的与球同胚的、亏格为 0 (genus 0) 的 3 维网格的简化。

3.1 网格剪切

为将亏格为 0 (genus-0) 的封闭网格映射至参数平面上,应将其沿三角边链剪切,并展开成为与圆盘同胚的网格模型,本文采用一种简单而直接的剪切方式,即在原始网格的平坦区域中删去两个共边的三角面,先剪切出一个小四边形空洞,以使原始网格变为与圆盘同胚,同时将四边形空洞的 4 个顶点按顺序映射至参数平面上单位正方形的 4 个顶点,并将其作为参数化映射的边界条件;然后再将其余网格顶点映射至单位正方形内部,虽然整个复杂网格要向一个微小的四边形空洞中映射,会造成极大的形变,但由于后续的参数平面二叉树剖分过程并不在乎形变的程度,因此这种简单的剪切方式是可行的,而无需采用文献[3]中的复杂的剪切方式来减少形变;最后在构造简化网格时,可手工地填补因剪切而成的小四边形空洞,因其对简化结果影响甚微。

3.2 参数化映射

对于存在封闭边界的网格模型,可将边界顶点按顺序投影到参数平面单位正方形的边界上,以作为参数化映射的边界条件;对于封闭网格,则可直接将剪切出的四边形空洞的 4 个顶点映射至单位正方形的 4 个顶点(图 3)。参数化映射的边界条件确定之后,本文采用文献[1]中的 Shape-Preserve 参数化算法来计算出其余内部网格顶点在参数平面单位正方形中的第 i 个参数化点 $q_i = (u_i, v_i)$ 。不妨设三角网格的 N 个顶点中,前 n 个顶点为内部顶点,其参数化点 q_1, q_2, \dots, q_n 未知待求,而后 $N-n$ 个顶点为切割边界顶点,其参数化点 q_{n+1}, \dots, q_N 已被映射至

参数平面单位正方形的边界上,作为已知的边界条件。

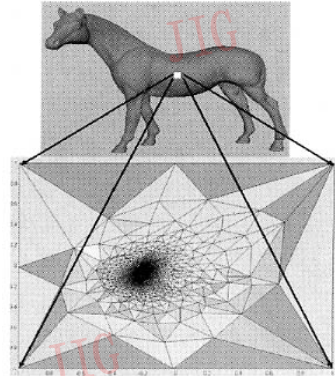


图 3 封闭的马网格模型在单位正方形上的参数化

Fig. 3 The parameterization mesh of the horse mesh on unit square

文献[1]是将网格顶点 i 的参数化点 $q_i = (u_i, v_i)$ 视为其各邻接顶点的参数化点的凸组合,即

$$q_i = \sum_{j=1}^N \lambda_{i,j} q_j, \quad i = 1, 2, \dots, n \quad (1)$$

其中, $\lambda_{i,j}$ 的取值需满足以下条件:

由于当 $j \in N_i$ 时, $\lambda_{i,j} > 0$; $j \notin N_i$ 时, $\lambda_{i,j} = 0$,

并且 $\sum_{j=1}^N \lambda_{i,j} = 1$,

故式(1)可改写为

$$q_i - \sum_{j=1}^n \lambda_{i,j} q_j = \sum_{j=n+1}^N \lambda_{i,j} q_j, \quad i = 1, \dots, n \quad (2)$$

根据式(2)可得到下列两个与未知参数化点 q_1, q_2, \dots, q_n 的坐标值分别对应的非齐次线性方程组:

$$AU = b_1 \quad \text{及} \quad AV = b_2 \quad (3)$$

其中, $U = (u_1, u_2, \dots, u_n)^T$, $V = (v_1, v_2, \dots, v_n)^T$, $n \times n$ 系数矩阵 A 的各元素取值应满足 $a_{i,i} = 1$ 及 $a_{i,j} = -\lambda_{i,j}$, ($i \neq j$), 且为稀疏矩阵,而方程组中的非齐次项则分别为

$$b_1 = \left(\sum_{j=n+1}^N \lambda_{1,j} u_j, \dots, \sum_{j=n+1}^N \lambda_{n,j} u_j \right)^T$$

$$b_2 = \left(\sum_{j=n+1}^N \lambda_{1,j} v_j, \dots, \sum_{j=n+1}^N \lambda_{n,j} v_j \right)^T$$

文献[1]是根据顶点 i 的局部几何信息来设计对应的凸组合系数 $\lambda_{i,j}$, 并证明了由此生成的系数矩阵 A 是非奇异矩阵,故式(3)中的两个方程组存在唯一解,并可分别解出各未知参数化点 q_1, q_2, \dots ,

q_n 的 2 维坐标值。利用该算法计算封闭的马网格模型在单位正方形上的参数化结果如图 3 所示。

3.3 顶点曲率计算

为生成平面曲率灰度分布,应将网格顶点的曲率属性量化为灰度值并赋予在参数平面上的对应投影点,本文采用文献[5]中的微分几何算子先计算出网格顶点的平均曲率与高斯曲率,并由此解出网格顶点的两个主曲率 k_1 与 k_2 ,再计算主曲率的组合:

$$c = \frac{1}{2} - \frac{\arctg((k_1 + k_2)/(k_1 - k_2))}{\pi} \in [0, 1] \quad (4)$$

将式(4)的计算结果作为顶点处的曲率属性,可较好地反映顶点处的弯曲程度,而将各顶点曲率属性 c 规一化为 0 ~ 255,作为灰度值赋予在参数平面上的对应投影点,则可形成平面曲率灰度分布。图 4 给出了几种网格模型顶点的曲率灰度分布。

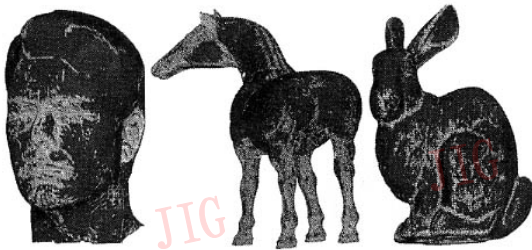


图 4 3 维网格顶点曲率属性

Fig. 4 The curvature attributes of mesh vertices

3.4 参数平面二叉树剖分

采用递归算法来实现对参数平面的等曲率灰度二叉树剖分,其每次剖分都是先将参数平面上单位正方形中的一个矩形块剖分为两个矩形子块,若输入矩形块的长度大于宽度,则竖向剖分,否则横向剖分,并使剖分后产生的两个子矩形块中投影点的曲率灰度总和相等;然后再对两个矩形子块进行同样的递归操作,直到所有的矩形块中的曲率灰度和小于预先设定的阈值 T 为止。在递归调用开始时,以参数平面上的单位正方形作为初始输入矩形块,并以单位正方形内曲率灰度总和除以一个常数作为阈值 T 的估计,而阈值 T 的大小则决定了二叉树的深度,也对应决定了最终简化网格中的顶点数目。对平面曲率灰度分布进行二叉树剖分的递归调用算法如下:

```
Function Split( $E_{up}, E_{down}, E_{left}, E_{right}, T$ )
// $E_{up}, E_{down}, E_{left}, E_{right}$ : 输入矩形块的上、下、左、右边界
```

```
// $T$ : 预先设定的结束递归调用的曲率灰度阈值
if (当前矩形块中的曲率灰度总和 < 曲率灰度阈值  $T$ )
{
    记录当前矩形块边界位置,生成二叉树中的叶节点。
    Return; //结束递归调用
}
 $H = E_{up} - E_{down}; W = E_{left} - E_{right};$ 
if ( $W > H$ ) //如果当前矩形块宽度  $W$  大于长度  $H$ 
{
    将当前矩形块纵向剖分为左右两个子块,剖分位置为  $X_{mid}$ ,并使左右两子块中的曲率灰度总和相等。
    //递归调用,对右矩形子块继续进行剖分
    Split( $E_{up}, E_{down}, X_{mid}, E_{right}, T$ );
    //递归调用,对左矩形子块继续进行剖分
    Split( $E_{up}, E_{down}, E_{left}, X_{mid}, T$ );
}
else //如果当前矩形块长度大于宽度
{
    将当前矩形块横向剖分为上下两个子块,剖分位置为  $Y_{mid}$ ,并使上下两子块中的曲率灰度总和相等。
    //递归调用,对上矩形子块继续进行剖分
    Split( $E_{up}, Y_{mid}, E_{left}, E_{right}, T$ );
    //递归调用,对下矩形子块继续进行剖分
    Split( $Y_{mid}, E_{down}, E_{left}, E_{right}, T$ );
}
}
```

网格模型平面参数化引起的形变并不影响参数平面的二叉树剖分过程(如图 5 所示),例如尽管兔网格模型的平面参数化引起了强烈的几何形变,其耳朵部分也被映射至参数平面中一个很小的区域中,但在二叉树剖分过程中,仍能够充分保留兔耳部分的几何信息。

3.5 构造简化网格

二叉树剖分结束后,再按照以下要求在每个叶节点对应的剖分子块中选取一个投影点:如果剖分子块中只含有一个投影点,则选取该投影点;如果剖分子块中含有多个投影点,则计算它们的重心位置,并选取距重心最近的投影点;然后再根据参数化映射关系,通过确定选取的投影点所对应的原始网格顶点来得到原始网格顶点集合的一个子集,以构成简化网格的顶点集合。

得到简化网格顶点集合之后,应为其赋予合适的三角化连接关系,以构成简化网格。由于上述算法所得到的简化网格顶点集合是原始网格顶点集合

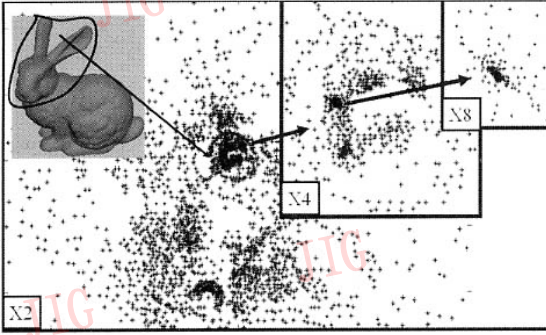


图 5 Bunny 兔网格的曲率灰度分布

Fig.5 The curvature gray-level distribution of the Bunny mesh

的一个子集,故可继承原始三角网格的连接关系,即由原始三角网格开始,逐次抛掉不属于简化网格顶点集合的顶点,而对遗留下来的空洞,则采用文献[6]中的算法来将空洞的边界顶点投影至平面上,即先构成一个平面多边形,再进行限制 Delaunay 三角划分 (constrained Delaunay triangulation),并根据所得三角面连接关系来填补 3 维网格中的空洞,以保证每一步都能得到一个三角网格,直到只剩余简化的网格顶点集合为止,此时所得到的三角网格即为简化的三角网格。

4 讨论与实验结果

本文提出的算法将 3 维网格简化问题转化为参数平面上的 2 维散乱点集的二叉树剖分问题,其主要优点是执行速度快。实验表明,与常用的基于二次型指标的网格简化算法 (Qslim) 及自适应空间聚类网格简化算法 (R-simp) 相比,本文提出的网格简化算法执行速度更快,表 1 给出了在奔腾 4 平台上分别采用 Qslim 算法、R-simp 算法及本文算法来对马网格模型、Bunny 兔网格模型及人脸网格模型进行简化的执行时间。

本文提出的算法是通过网格顶点的曲率灰度背景图进行二叉树剖分来实现网格简化,由于其输出的简化网格在曲率较低平坦部分保留的顶点较少,而在曲率较高的细节部分则保留了较多的顶点,从而使网格顶点数目在大幅减少的同时,仍能够较好地保持原始网格的细节信息。图 6 ~ 图 8 是利用本文算法分别对马、Bunny 兔型及人脸网格模型进行简化的结果,图 6 ~ 图 8 中的左图为原始网格,中

表 1 本文算法与 Qslim 算法及 R-simp 算法执行时间比较
Tab.1 Running time of our algorithm, Qslim and R-simp

网格模型	输入顶点	输出顶点	算法执行时间 (s)		
			Qslim	R-simp	本文算法
马	48 485	2 048	4.6	1.3	0.24
Bunny 兔	34 835	2 048	3.4	1.05	0.21
人脸	38 775	2 048	3.8	1.1	0.22

间图与右图分别是选择阈值 T 为平面曲率灰度总和的 $1/2 048$ 及 $1/1 024$ 时得到的简化结果,对应的网格顶点数目分别为 2 048 及 1 024。由图 6 ~ 图 8 可见,简化网格在较平坦部分的顶点数目大幅减少,而在马头、腿处,人脸的眼、鼻、耳处,及兔耳处等细节部分却仍分布有较多的顶点,这也体现了本文算法保持原始网格细节特征的特性。

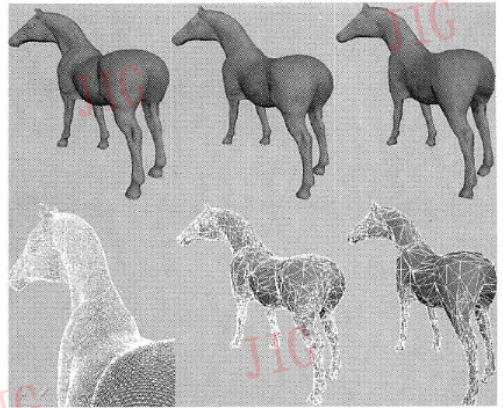


图 6 马网格模型简化结果

Fig.6 The Simplification results of the horse mesh

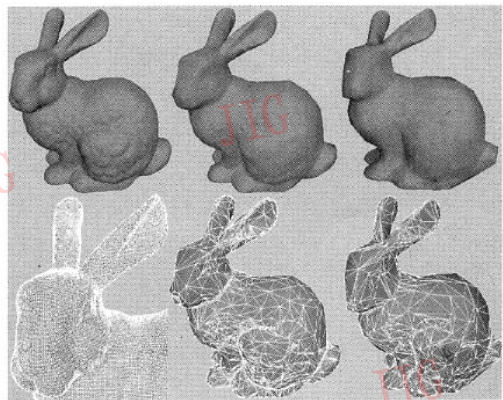


图 7 Bunny 兔网格模型简化结果

Fig.7 The Simplification results of the Bunny mesh

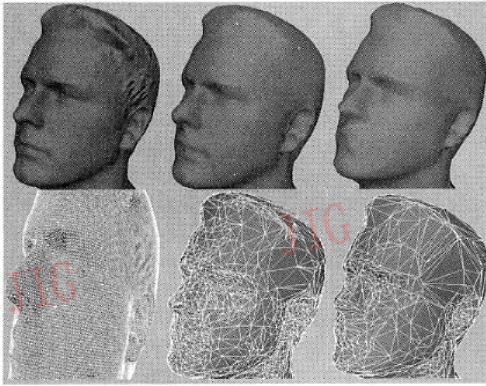


图 8 人脸网格模型简化结果

Fig. 8 The Simplification results of the face mesh

5 结 论

虽然本文算法可快速地进行 3 维网格简化,并可充分保持原始网格的细节特征,但仍有以下问题需进一步解决:

(1) 探讨对同类型的 3 维网格(例如人脸 3 维网格)平面参数化的一致性问题,以保证几何结构相近的 3 维网格,其平面参数化结构也相近,并可使文中算法能应用于 3 维网格模型的检索与比对;

(2) 研究新的参数化点集合总体配置最优化的

问题,例如在满足各个新的参数化点间最小距离最大化的约束下,如何获得各个端点在矩形区内对新参数化点的合理配置;

(3) 解决如何补偿非平衡二叉树各端点属性和的非均匀性问题,以便使简化网格顶点集合的分布更为合理;

(4) 研究如何利用非平衡二叉树对 3 维网格进行分解的思想来构造出 3 维网格多分辨率分解的新途径。

参考文献 (References)

- 1 Floater M S. Parameterization and smooth approximation of surface triangulations [J]. *Computer Aided Geometric Design*, 1997, 14(3): 231 ~ 250.
- 2 Hormann K. MIPS: An efficient global parameterization method [M]. St. Malo, France: Vanderbilt University Press, 1999: 153 ~ 162.
- 3 GU Xian-feng, Gortler J, Hoppe H. Geometry images [A]. In: *Proceedings of ACM SIGGRAPH* [C]. Antonio, Texas, USA, 2002: 355 ~ 361.
- 4 Sander P, Wood Z, Gortler J, *et al.* Multi-chart geometry images [A]. In: *Proceedings of the 1st Symposium on Geometry Processing* [C], Aachen, Germany, 2003: 138 ~ 145.
- 5 Meyer Mark, Desbrun Mathieu, Schroder Peter. Discrete differential-geometry operators for 2-manifold [EB/OL]. <http://Multires.caltech.edu/pubs/diffGeoOps.pdf>, 2000.
- 6 Lee Aaron, Sweldens Wim, Schorder Peter. MAPS: Multiresolution Adaptive Parameterization of Surfaces [A]. In: *Proceedings of ACM SIGGRAPH* [C]. Orlando, Florida, USA, 1998: 95 ~ 104.