

一种对空间散乱点集进行简化的新方法

权 勇,李文辉,卢奕南,张振花

(吉林大学 计算机科学与技术学院, 长春 130012)

摘要:提出了一种剔除空间散乱点集中冗余点的新方法。利用称为“球片点”的新的点基元近似表示空间的离散点及其附近的区域。由于曲率的特性,所提出的方法可用较少的离散点获得与其他相关方法相似的绘制效果。利用全局光照模型实现对三维形体的绘制,并利用大规模点模型对所提出的简化方法进行了试验,试验结果证明了方法的有效性。在保证绘制质量近似相同的情况下,该方法能将原始点集的点数减少约40%。

关键词: 计算机应用; 三维数据处理; 球片点; 形体绘制; 简化

中图分类号: TP39 **文献标识码:** A **文章编号:** 1671-5497(2008)Sup. -0156-07

New method for simplification of 3D scattered points

Quan Yong, Li Wen-hui, Lu Yi-nan, Zhang Zhen-hua

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: A new method for removing redundant points from 3D scattered point sets is presented. A novel point primitive, referred as Spherical Patch Point, is employed to approximate a 3D point and its vicinity. Due to the property of curvature, this new method can deliver similar visual quality with much less primitives in comparison with other point-based methods. Global illumination is used to render 3D models. The proposed simplification method is tested using large-scale point data, and the results verify its effectiveness. Using the new method can reduce 40% of the original points, meanwhile, ensures the similar visual appearances.

Key words: computer application; 3D data processing; spherical patch point; model drawing; simplification

近年来,随着计算机系统和三维激光测量技术的快速发展,逆向工程技术迅速发展并广泛应用于机械制造、精密加工、医学、动漫娱乐等领域。如何将采集到的三维点数据进行简化处理,使其方便用于CAD/CAM及后期快速高效的生产制造已成为重点关注的工程实际问题。基于点的技术最初由Levoy和Whitted在1985年提出^[1]。

Grossman和Dally^[2]利用等边三角形网格对需要绘制的物体进行重采样得到其基于点的表示,然后采用基于点的绘制技术对物体进行绘制。Alexa等人^[3]利用移动最小二乘法在每个离散点处得到一个多项式曲面。该曲面的绘制需以一种与视点相关的方式在曲面上产生附加的点。Schauffler和Wann Jensen提出了一种直接对点

收稿日期: 2007-11-01.

基金项目: 国家自然科学基金项目(60573182);高等学校博士学科点专项科研基金项目(20060183042);吉林省科技发展计划项目(20070703,20060527);教育部留学回国人员科研启动基金项目。

作者简介: 权勇(1977-),男,讲师,博士研究生。研究方向:计算机图形学,虚拟现实技术。

E-mail: quanyong_jlu@yahoo.com.cn

集的光线跟踪算法^[4]。Adamson 和 Alexa^[5]研究了光线跟踪点集曲面的技术。由于投影操作过程耗时太多,导致光线跟踪点集曲面绘制速度较慢。文献[6]提出了一种基于迭代求交算法的曲面近似技术。这种经过改进的方法在速度上与 Schauffler 和 Jensen 的方法^[4]相当。Pauly 等^[7]通过比较和实现几种不同的点模型简化算法得到了与原模型相近的简化点模型。同年 Pauly 等设计了一个多分辨率点模型造型框架^[8]。Kalaiah 和 Varshney^[9]通过微分点集来表现曲面并利用微分点实现对基于点的模型的简化。但微分点只适用于局部光照模型。

作者利用微分几何学的知识提出使用“球片点”这种新的点基元对三维点数据进行简化,即在保留特征点的前提下剔除尽可能多的冗余点。所提出的“球片点”适合于全局光照模型并易于简化操作。

1 球片点

球片点的属性包括位置向量、主曲率、主方向和邻域半径。曲面上某一点的单位法向量和经过这一点的切平面可以由该点的主方向确定。

假定 P 是三维空间曲面 S 上的一点, λ_u 和 λ_v 是 P 的主曲率, u 和 v 是 P 的主方向。 P 的单位法向量 n 和过 P 的切平面 T 可以由 u 和 v 确定。设 Q 为 P 的切平面上邻近 P 的任意一点, 则向量 PQ 确定了切线方向 t 。通过 t 和 n 做平面 N 。设 N 与 S 的交线为 l , 则由微分几何知识可知, l 即是曲面 S 在 P 点沿切线 t 的法截线。如图 1 所示。若用 $(0 \leq \theta < \pi)$ 表示 t 和 u 之间的夹角, 用 λ_t 来表示曲面 S 在 P 点沿切线 t 方向的法曲率, 则由微分几何中的欧拉公式可得

$$\lambda_t = \lambda_u \cos^2 \theta + \lambda_v \sin^2 \theta, \quad 0 \leq \theta < \pi$$

而法曲率有这样的性质: 曲面在一点沿一条切线的法曲率等于沿同一切线的法截线的曲率。

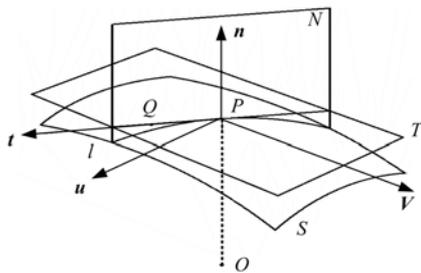


图 1 球片点的局部几何结构
Fig. 1 Local geometry of a SPP

据此可知, l 在 P 点的曲率等于 λ_t 。相应地, λ_t 的倒数 ρ 为 l 在 P 点的曲率半径。当 $\rho > 0$ 时, 法截线 l 在 P 点向 n 正向弯曲; 当 $\rho < 0$ 时, l 向负向弯曲。 p 表示 P 点的位置向量。球心在 $p + \rho n$, 半径为 ρ 的球面是法截线 l 在 P 点的一个密切球。作者正是利用这个球面的邻近 P 点的部分来近似表示 P 点沿其切线 t 方向附近的曲面。为了限定该球面邻近 P 点的部分, 要用到 P 点的邻域半径 r 。设入射光线与切平面 T 的交点为 Q' , 仅当 $|PQ'| \leq r$ 时, 将此入射光线与相应的球面求交。称切平面上以 P 点为圆心, 以 r 为半径的圆盘为对应于 P 点的邻域圆盘。

假定 L 表示曲面 S 在 P 点沿所有切线方向的法截线的集合, D 表示对应于 P 点的邻域圆盘。则称 P 点连同集合 L 及邻域圆盘 D 为一个球片点。

2 点数据的处理

2.1 计算点的属性

因为点数据能通过三维扫描仪直接得到, 所以可利用文献[10]中的技术对点数据进行去噪操作并获得每一点处的微分几何信息。

邻域半径是球片点的一个重要属性。本文中, 一个球片点的邻域半径的初始值被取作该球片点与其邻近点之间的最大距离。设当前点 P 的邻近点集为 $N(P)$ 。可取与点 P 距离最近的 8 个点组成 $N(P)$ 。查找与点 P 距离最近的点的操作可通过空间数据结构——八叉树^[11]来加速实现。

2.2 简化

用于描述三维形体的采样点通常会有冗余。简化过程就是删除三维形体上冗余程度较高的那些点的过程。本方法中一个点能否被删除取决于冗余程度的高低和其邻域圆盘能否被其邻近点的邻域圆盘共同替代。

2.2.1 相对位置偏差和相对法向偏差

本文用相对位置偏差 γ 和相对法向偏差 η 来描述一个点的冗余度, 所讨论的点都是三维空间曲面 S 上的点, 且其主曲率和主方向已知。仍假定 P 是当前点。 P 的位置向量用 p 表示。 P 的单位法向量 n 可以由 P 的主方向确定。令 P 的邻近点为 Q_1, \dots, Q_m 。取其中任一点 $Q_i (i = 1, \dots, m)$ 。过 Q_i 的切平面 T_i 可以由 Q_i 的主方向确定。设 P 点沿法向量 n 在平面 T_i 上的投影为

R。据欧拉公式可得曲面 S 在 Q_i 点沿切线 Q_iR 的法曲率。由此可确定曲面 S 沿切线 Q_iR 的法截线在 Q_i 点的密切球 C。假定此密切球的球心为 O。称直线 PR 与此密切球的交点中距离 P 点较近的那个点为 P 点关联于 Q_i 点的参照点。设该点为 P' 。如图 2 所示。分别用 p' 和 n' 来表示

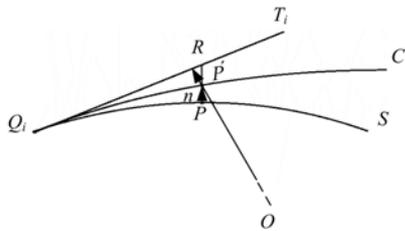


图 2 球片点关联于其某个邻近点的参照点

Fig. 2 Reference point of a SPP related to a certain neighboring point

P' 点的位置向量和向量 OP' 的单位法向量。于是,可定义 P 点关联于 Q_i 点的位置偏差 γ' 和 P 点关联于 Q_i 点的法向偏差 η' 如下

$$\gamma' = |p - p'|$$

$$\eta' = 1 - n \cdot n'$$

式中:“| |”表示向量的取模运算;“ \cdot ”表示向量的点乘运算。若 P 点关联于其邻近点 $Q_i (i=1, \dots, m)$ 的位置偏差分别为 $\gamma_i (i=1, \dots, m)$, 则可进一步定义 P 点关联于其全部邻近点的位置偏差为

$$\gamma = \frac{1}{m} \sum_{i=1}^m \gamma_i$$

类似地,若 P 点关联于其邻近点 $Q_i (i=1, \dots, m)$ 的法向偏差分别为 $\eta_i (i=1, \dots, m)$, 定义 P 点关联于其全部邻近点的法向偏差为

$$\eta = \frac{1}{m} \sum_{i=1}^m \eta_i$$

为简便起见,称 P 点关联于其全部邻近点的位置偏差为 P 点的相对位置偏差,称 P 点关联于其全部邻近点的法向偏差为 P 点的相对法向偏差。只有当一个点的相对位置偏差小于某一阈值 δ 时,该点才被确定为备选可删点。在简化过程中,会对每个点进行检查,以确定其是否为备选可删点。阈值 δ 可由用户根据需要来设定。如果用户对绘制精度要求不是很高,则可选取较大的 δ 值,这样能在图像质量没有明显下降的前提下大大提高绘制速度。

2.2.2 简化过程

借助最小堆数据结构对相对法向偏差进行比

较。首先定义一个空的最小堆。然后,针对每个备选可删之点,以其相对法向偏差的值作为关键码将其插入到该最小堆中。堆顶元素应具有堆中元素最小的关键码。创建最小堆的算法如算法 1 所示。

算法 1:

- S 为一个点的集合。 δ 为用户根据需要设定的阈值。
- 1 $S \leftarrow$ 描述三维形体的全部采样点
- 2 定义一个空的最小堆 H
- 3 for S 中的每个点 P
- 4 获得 P 点的邻近点集 $N(P)$
- 5 计算 P 点的相对位置偏差 $\gamma(P)$
- 6 if $\gamma(P) < \delta$ then
- 7 计算 P 点的相对法向偏差 $\eta(P)$
- 8 以 P 点的相对法向偏差的值作为关键码将其插入到最小堆 H 中

当所有的备选可删点都被插入到最小堆中后,就开始从堆中取走堆顶元素并获得其所对应的备选可删点,以检查所得到的点的邻域圆盘能否被该点在曲面上的邻近点的邻域圆盘共同替代。如果能,则从描述三维形体的点集中删除所得到的点。这会导致被删点的邻近点的邻近点集发生变化,被删点的邻近点的相对位置偏差和相对法向偏差也会发生变化。从而可能导致堆中剩余的元素或它们之间的大小关系发生变化。如果所得到的点的邻域圆盘不能被该点的邻近点的邻域圆盘共同替代,所得到的点为可显示的点。接下来,要调整变化后的堆中剩余的元素为一个新的最小堆。然后,重复上述取出堆顶元素并对其进行检查的操作。这是一个反复的过程,直至这个堆最终成为一个空堆。

以下讨论如何判断一个点的邻域圆盘能否被该点在曲面上的邻近点的邻域圆盘共同替代。这要间接地用到作者设计的一种平面上的圆对该平面上的特殊直线段的分割算法(算法 2)。平面上的特殊直线段是指该直线段的两个端点分别在同一平面上的某个圆的内部和外部。算法 2 的输入为直线段和圆,输出为直线段上的分割点。

算法 2:

- XY 为平面上将被分割的直线段,U 为平面上的某个圆。直线段的端点 X 在 U 的内部,直线段的端点 Y 在 U 的外部,U 的圆心为 E 点,U 的半径为 r。 $|AB|$ 表示 A 点和 B 点之间的距离。CurrentLine 表示当前直线段。M 表示当前直线段的中点,是一个小正数,令 $\epsilon = |XY|/30$ 。

- 1 CurrentLine \leftarrow 直线段 XY
- 2 M \leftarrow CurrentLine 的中点

```

3 while |EM| < r - or |EM| > r
4 if |EM| < r then
5 CurrentLine←直线段 MY
6 else
7 CurrentLine←直线段 XM
8 M←CurrentLine 的中点
9 return M
    
```

由算法 2 可得到平面上的圆对该平面上的等边三角形的操作算法,称之为算法 3。该算法的输入为三角形和圆,它的输出是一个三角形的集合。算法 3 分 6 种情况来描述平面上的圆对该平面上的等边三角形的操作。(a) $\triangle ABC$ 的 3 个顶点在圆 U 上或圆 U 内。(b) $\triangle ABC$ 的 3 个顶点及其重心在圆 U 上或圆 U 外。(c) $\triangle ABC$ 的 3 个顶点在圆 U 外且其重心在圆 U 上或圆 U 内。(d) $\triangle ABC$ 的顶点 A 在圆 U 内,顶点 B 在圆 U 上,顶点 C 在圆 U 外。(e) $\triangle ABC$ 的顶点 A 和 B 在圆 U 内,顶点 C 在圆 U 外。(f) $\triangle ABC$ 的顶点 A 在圆 U 内,顶点 B 和 C 在圆 U 外。以上 6 种情况如图 3 所示。

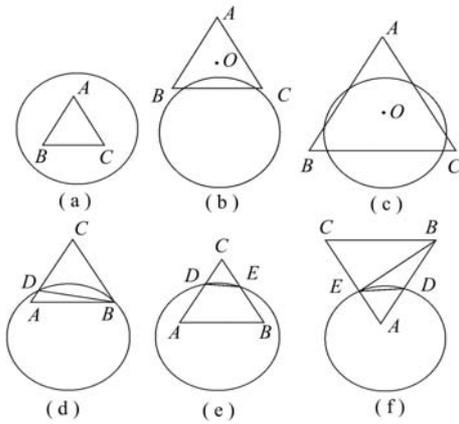


图 3 平面上的圆对平面上的等边三角形的分割
Fig. 3 Division of an equilateral triangle by a circle
算法 3:

$\triangle ABC$ 为平面上将被分割的等边三角形, U 为平面上的某个圆。 $\triangle ABC$ 的 3 个顶点 A, B 和 C 的位置可以任意互换。

```

1 if  $\triangle ABC$  的三个顶点在圆  $U$  上或圆  $U$  内 then
2 return NIL
3 else if  $\triangle ABC$  的 3 个顶点及其重心在圆  $U$  上或圆  $U$  外 then
4 return  $\triangle ABC$ 
5 else if  $\triangle ABC$  的 3 个顶点在圆  $U$  外且其重心在圆  $U$  上或圆  $U$  内 then
6 将  $\triangle ABC$  等分为 4 个小等边三角形
    
```

```

7 return 这 4 个等边三角形
8 else if  $\triangle ABC$  的顶点  $A$  在圆  $U$  内, 顶点  $B$  在圆  $U$  上, 顶点  $C$  在圆  $U$  外 then
9 利用算法 2 对直线段  $AC$  进行分割得到分割点  $D$ 
10 return  $\triangle DBC$ 
11 else if  $\triangle ABC$  的顶点  $A$  和  $B$  在圆  $U$  内, 顶点  $C$  在圆  $U$  外 then
12 利用算法 2 对直线段  $AC$  和  $BC$  分别进行分割得到分割点  $D$  和  $E$ 
13 return  $\triangle DEC$ 
14 else if  $\triangle ABC$  的顶点  $A$  在圆  $U$  内, 顶点  $B$  和  $C$  在圆  $U$  外 then
15 利用算法 2 对直线段  $AB$  和  $AC$  分别进行分割得到分割点  $D$  和  $E$ 
16 return  $\triangle DBE$  和  $\triangle EBC$ 
    
```

本文用一种保守的方法来判断一个点的邻域圆盘能否被该点的邻近点的邻域圆盘共同替代。已知 P 是三维形体表面上的一点。容易得到 P 点的邻域圆盘的外接正六边形 $A_1 A_2 A_3 A_4 A_5 A_6$ 。把此正六边形等分为 6 个正三角形。如图 4 所示。这样, 可将判断 P 点的邻域圆盘能否被 P 点的邻近点的邻域圆盘共同替代的问题转变为判断这 6 个三角形能否被 P 点的邻近点的邻域圆盘共同覆盖的问题。具体算法如算法 4。

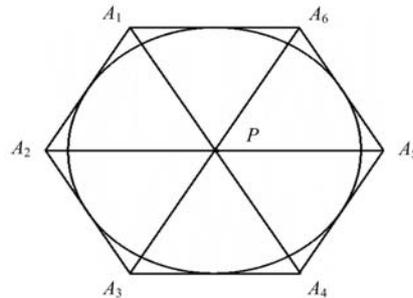


图 4 等分球片点的邻域圆盘的外接正六边形
Fig. 4 Equipartition of circumscribing regular hexagon of a SPPs neighborhood disk

算法 4:

P 点为三维形体表面上的一点。 Q 点为 P 点的邻近点集中的任意一点。我们要判断 P 点的邻域圆盘能否被其邻近点的邻域圆盘共同替代。 P 点的法向量为 n 。 S 为三角形的集合。

```

1 S ← 分割  $P$  点的邻域圆盘的外接正六边形所得到的 6 个正三角形
2 for  $P$  点的每个邻近点  $Q$ 
3 for  $S$  中的每个三角形  $T$ 
4 将三角形  $T$  从  $S$  中删除
5 将三角形  $T$  沿  $n$  的方向投影到  $Q$  点的切平面上得
    
```

到三角形 T
 6 根据算法 3 用 Q 点的邻域圆盘对三角形 T 进行分割
 7 将算法 3 输出的经分割得到的三角形沿 n 的反方向投影回 P 点的切平面
 8 将投影回 P 点的切平面得到的三角形加入 S
 9 if S 为空 then
 10 return TRUE
 11 return FALSE

由前述可知,从描述三维形体的点集中删除点会导致被删之点的邻近点的邻近点集发生变化。使用如下方法对被删点的邻近点的邻近点集进行更新。设 P, Q, T 是三维形体表面上的点,且有 $Q \in N(P), T \in N(P), T \notin N(Q)$ 。值得注意的是, P 点不一定在 $N(Q)$ 中。假设 P 点从描述三维形体的点集中被删除。下面以 Q 点的邻近点集 $N(Q)$ 为例说明如何对 P 点的邻近点的邻近点集进行更新。如果 P 点原来在 $N(Q)$ 中,需首先将 P 点从 $N(Q)$ 中删除。这里,更关心的是如何将 $N(P)$ 中异于 Q 点且不在 $N(Q)$ 中的点按是否符合一定的条件加入 $N(Q)$ 。本文针对 T 点进行说明。假定 T 点与 Q 点之间的距离为 d , T 点的邻域半径为 r_1 , Q 点的邻域半径为 r_2 。如果 $r_1 + r_2 > d$,则将 T 加入 $N(Q)$ 。

本文中,一个点的邻域半径的初始值被取作该点与其邻近点之间的最大距离。但是,当一个点的邻近点被删除时,该点的邻域半径的大小并不改变。事实上,本文的邻域半径的初始值的选取方法有利于对冗余程度高的点的删除,但会导致某些点所对应的邻域圆盘的面积偏大。因此需要在从描述三维形体的点集中删除点的过程结束后对某些点的邻域半径进行一定的调整。这就需要在删除点的过程中对相关的点作标记。已知 P, Q 是三维形体表面上的点,且 $Q \in N(P)$ 。假设 P 点从描述三维形体的点集中被删除。则应对 Q 点作标记以表示 Q 点的邻域半径不可改变。但 Q 点仍是可能被删除的点。在从描述三维形体的点集中删除点的过程结束后,点集中存在某些被作了标记的点,将点集中的那些未作标记的点的邻域半径变为其初始邻域半径的 $\sqrt{2}/2$ 倍。

由以上讨论可得出如下简化算法。

简化算法:

S 为一个点的集合。 δ 为用户根据需要设定的阈值。 rds 表示点的邻域半径。

1 $S \leftarrow$ 描述三维形体的全部采样点

2 利用算法 1 创建最小堆 H
 3 while H 非空
 4 取出堆顶元素并获得其所对应的备选可删之点 Q
 5 if Q 点的邻域圆盘能被 Q 点的邻近点的邻域圆盘共同替代 then
 6 for Q 点的邻近点集 $N(Q)$ 中的每一点 L
 7 对 L 点做标记
 8 if Q 点在 L 点的邻近点集 $N(L)$ 中 then
 9 将 Q 点从 $N(L)$ 中删除
 10 for $N(Q)$ 中异于 L 点且不在 $N(L)$ 中的每个点 V
 11 计算 V 点与 L 点之间的距离 d
 12 if $V.rds + L.rds > d$ then
 13 将 V 加入 $N(L)$
 14 重新计算 L 点的相对位置偏差 $\gamma(L)$ 和相对法向偏差 $\eta(L)$
 15 if $\gamma(L) \geq \delta$ and L 点在 H 中 then
 16 将 L 点从 H 中移出
 17 else if $\gamma(L) < \delta$ and L 点不在 H 中 then
 18 将 L 点移入 H
 19 从 S 中删除 Q 点
 20 重新调整 H 中的元素为一个新的最小堆
 21 将集合 S 中的那些未作标记的点的邻域半径变为其初始邻域半径的 $\sqrt{2}/2$ 倍。
 22 输出集合 S

3 绘制

本文通过光线跟踪技术实现对球片点的绘制。针对球片点的特点,设计了一种光线与球片点结构求交的新算法,见算法 5。

算法 5:

当前球片点为 P 。 p 表示 P 点的位置向量, rds 表示 P 点的邻域半径, u 和 v 表示在 P 点的主方向, λu 和 λv 分别表示在 P 点沿 u 方向和 v 方向上的主曲率。当前光线为 r 。

1 由 u 和 v 确定过 P 点的切平面 T
 2 计算光线 r 与切平面 T 的交点 Q
 3 计算 P 点与 Q 点之间的距离 d
 4 if $d \leq rds$ then
 5 由 u 和 v 确定 P 点的单位法向量 n
 6 计算起点为 P 点、终点为 Q 点的向量与向量 u 之间的夹角 ($0 \leq \theta < \pi$)
 7 计算曲率半径 $\rho = (\lambda u \cos^2 \theta + \lambda v \sin^2 \theta)^{-1}$
 8 计算光线 r 与球心在 $p + \rho n$, 半径为 ρ 的球面的第一个交点 I
 9 输出 I 点
 10 else
 11 判定光线 r 与球片点 P 不相交

4 实验结果

本文所有的示例都是在一台 PC 机上实现的。该 PC 机配有频率为 2 GHz 的 Pentium 4 处理器、256 MB 的内存和 Radeon 9000 图形加速卡。在 5 个模型上对所提出的技术进行试验。这 5 个模型分别为:奶牛模型,Buddha 模型, Isis 模型,马模型和兔子模型。

图 5 显示了当 δ 取不同值时处于不同的细节层次的 Buddha 模型的绘制效果。从左至右, Buddha 模型由 543k 个点简化为 37k 个点。

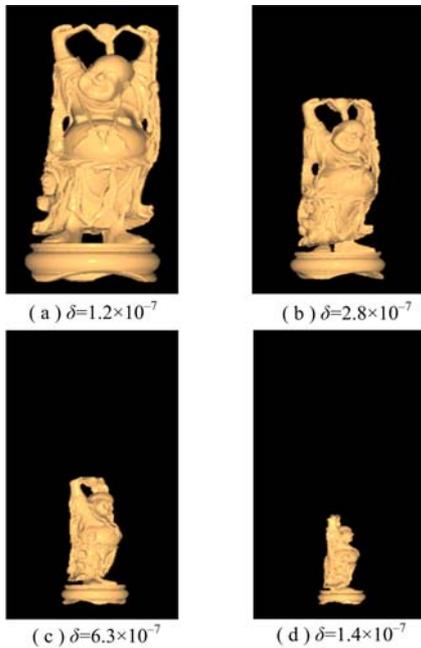


图 5 不同细节层次的 Buddha 模型

Fig. 5 Buddha at different levels of detail

图 6 表明本文能通过相对少的多的离散点获得与文献[4]和文献[5]中的方法相似的视觉效果。使用 2.2 节中提出的方法对模型进行简化。图 6(a)为使用本文方法得到的 Stanford 兔子模型,该模型仅包含 28 556 个离散点。在图 6(b)和(c)中,Stanford 兔子模型分别是采用文献[4]和文献[5]中的方法绘制得到,这两个模型各包含 34 834 个点。

从文献[6]中可知,在绘制质量近似相同的情况下,文献[4]中的方法比对点集曲面的光线跟踪方法[5]大约快两个数量级,因此作者将本文提出的方法与文献[4]中的方法在绘制时间上进行了比较,如表 1 所示。以奶牛模型、马模型和 Isis 模型为例进行说明。由于速度与时间成反比,从表 1 中可以得出:本文方法在速度上大约比文献[4]



(a) 本文方法



(b) 文献[4]的方法



(c) 文献[5]的方法

图 6 不同方法绘制同一模型结果比较

Fig. 6 Result comparison of drawing method same model with different method

中的方法快 1.5 倍。为减少求交的次数,本文采用八叉树结构^[11]来组织点数据。此外,从表中亦可得出,本文的方法能将原始点集的点数减少约 40%。

表 1 本文方法与文献[4]中的技术在绘制时间上的比较
Table 1 Comparison with approach in^[5] in rendering time

		奶牛-a	马-b	Isis-c
文献[4]中的方法	点的数量	11996	48485	187644
	绘制时间/s	8.7	25.1	33.9
本文的方法	点的数量	6173	26353	51628
	绘制时间/s	5.6	17.5	23.1

5 结束语

提出了一种称为“球片点”的新的点基元。用这种点基元近似表示空间的离散点及其附近的区域,并针对这种点基元,设计了一种对基于点的模型进行简化的新方法。该方法能有效地删除基于点的模型上冗余程度较高的离散点,从而能在绘

制质量近似的前提下通过较少的离散点来表现客体模型。该方法利用全局光照模型并通过光线跟踪技术实现对球片点的绘制,在保证绘制质量近似相同的情况下,能将原始点集的点数减少约 40%。

参考文献:

- [1] Levoy M, Whitted T. The use of points as a display primitive[R]. USA: CS Department, University of North Carolina at Chapel Hill, 1985.
- [2] Grossman J P, Dally W J. Point sample rendering [C]//Proceedings of Rendering Techniques '98, EG Workshop on Rendering. Germany: Springer-Verlag, 1998: 181-192.
- [3] Alexa M, Behr J, Cohen-Or D, et al. Point set surfaces[C]//Proceedings of IEEE Visualization 2001. USA: IEEE Computer Society, 2001: 21-28.
- [4] Schaufler G, Wann Jensen H. Ray tracing point sampled geometry[C]//Proceedings of Rendering Techniques 2000, EG Workshop on Rendering. Germany: Springer-Verlag, 2000: 319-328.
- [5] Adamson A, Alexa M. Ray tracing point set surfaces[C]//Proceedings of Shape Modeling International 2003. USA: IEEE Computer Society, 2003: 272-279.
- [6] Adamson A, Alexa M. Approximating and intersecting surfaces from points[C]//Proceedings of Eurographics Symposium on Geometry Processing. Eurographics, 2003: 230-239.
- [7] Pauly M, Gross M, Kobbelt L P. Efficient simplification of point-sampled surfaces[C]//Proceedings of IEEE Visualization 2002. USA: IEEE Computer Society, 2002: 163-170.
- [8] Pauly M, Gross M, Kobbelt L P. Multiresolution modeling of point-sampled geometry[R]. Zurich: Computer Science Department, ETH Zurich, 2002.
- [9] Kalaiah A, Varshney A. Modeling and rendering of points with local geometry[J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9 (1): 30-42.
- [10] Lange C, Polthier K. Anisotropic smoothing of point set[J]. Computer Aided Geometric Design, 2005, 22 (7): 680-692.
- [11] Ahuja N, Veenstra J. Octree generation and display [R]. Urbana: Coordinated Science Laboratory, University of Illinois, 1986.