

点绘制技术的研究

张 武, 黄争舸, 张桢夏

(浙江大学工程与科学计算研究中心, 杭州 310027)

摘 要: 对于形状复杂且不规则的物体, 点绘制能实现高的绘制效率, 还能获得高质量的绘制效果。该文讨论了点绘制方法的基本原理和基本绘制管道以及点绘制方法的硬件实现, 并对今后研究和发展的方向进行了展望。

关键词: 计算机图形学; 点绘制; 绘制技术

Research of Point-based Rendering Technique

ZHANG Wu, HUANG Zhengge, ZHANG Zhenxia

(Center of Engineering and Scientific Computation, Zhejiang University, Hangzhou 310027)

【Abstract】 For irregular objects with complex geometry, point-based rendering brings both high efficiency and high quality. This paper discusses the basic idea and basic pipeline of point-based rendering, introduces its hardware implementation, and then gives a prospect.

【Key words】 Computer graphics; Point-based rendering; Rendering technique

1 概述

近年来, 计算机图形硬件的处理能力取得巨大的进展, 绘制具有几百万个三角面片的模型已经变得很平常。但是由于显示分辨率并没有获得相应的提高, 绘制到屏幕空间每个三角面片的投影区域往往只占几个像素, 甚至在一个像素之内, 在这样的情况下, 光栅化操作不再具有原来的高效。因此, 需要一种比三角片更加简单有效的绘制单元。另一方面, 随着三维扫描技术的进步, 利用采样点来表征复杂物体表面并进行交互绘制已成为计算机图形学发展的一个重要前沿。然而实际采样点是不规则分布的且数目往往是 10^6 以上数量级, 如斯坦福大学计算机图形实验室的笑面佛模型(1 060 220 个点)、龙模型(1 279 481 个点)、胜利女神模型(10 072 906 个点)。如果采用传统的基于三角片的绘制方法, 必需先根据采样点生成三角片, 这是件费时且容易出错的事。如果有一种方法能够直接从获得的采样点上绘制出视觉等价的物体表面, 这将具有极大的实际意义。

点绘制方法正是在这样的应用背景下发展起来的。点绘制方法采用离散化的且与视点无关的点集来表征物体表面, 这些点不仅记录着物体表面的几何信息(位置、法向量), 而且还记录着如颜色、材质等其它信息。与传统的三角片绘制方法相比, 点绘制方法不包含点与点之间的连接关系, 绘制时不需要维护模型的拓扑一致性, 便于重取样, 可以建立灵活的层次结构来实现多分辨率(multiresolution)绘制。另外点绘制方法更适合绘制羽毛、头发、树这些具有细小的特征和非常复杂的轮廓线的有机模型。

点绘制方法最早由Levoy和Whitte^[1]在 1985 提出, 但由于硬件技术和应用环境的限制, 这一技术并没有得到太多的关注。直到 1998 年, Grossman和Dally^[2]在欧洲图形学大会上提出了一种点集的生成和绘制算法, 点绘制技术才真正被关注并得以深入的研究。Rusinkiewicz^[3], Matusik^[4]等把点作为物体表面表征和绘制的基本单元来直接绘制由三维扫描系统获得的数据。Chen^[5]等提出了一套点与三角片相结合的混合

绘制系统。Wand^[6]等用点来绘制大规模动画场景。Deussen^[7]等用点来处理生态系统中草、树木等三角片绘制技术难以很好绘制的物体。Zwicker^[8]等设计了一套软件系统 Pointshop3D, 实现了对点取样物体表面的建模、编辑和修改。

2 点绘制的基本原理和基本绘制管道

点绘制方法采用不规则分布的点云来表征物体表面, 因此点绘制的最基本任务就是在这些不规则分布的点云上重建出视觉等价的光滑曲面。我们可以把点绘制看作是点云的重取样操作, 输入数据是物体空间上不规则分布的点云, 输出数据是在图像空间上重取样的分布规则的点集, 其绘制管道如图 1 所示。

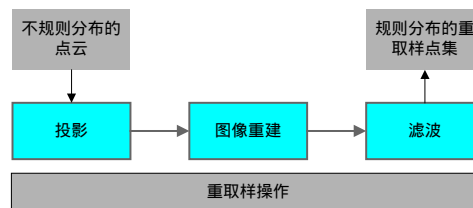


图 1 点绘制基本绘制管道

(1) 投影, 为点建立一种模型使投影到图像空间的点之间无“空洞”。

(2) 图像重建, 主要是可见性判断和像素归一化, 使最后显示的图像光滑无瑕疵(如亮斑)。

(3) 滤波, 滤掉某些频率变化过高的点, 达到反走样效果。

2.1 投影

点绘制最简单的方式是把点直接投影到图像空间并设置相对应的像素的属性。这么做最大的优点是硬件支持, 绘制速度快, 但由于点云的不规则分布, 常常会产生“空洞”。

基金项目: 国家杰出青年基金资助项目(60225509)

作者简介: 张 武(1981 -), 男, 硕士生, 主研方向: 计算机图形技术, 大规模可视化; 黄争舸, 副研究员; 张桢夏, 硕士生

收稿日期: 2006-04-07 E-mail: zhw_cqu@hotmail.com

目前最通用的做法是用面元 (Surfel^[9]) 来表示一个点。面元是点切平面上的一个圆盘, 包含位置、法向量、半径、颜色、材质等信息, 各个面元相互重叠形成紧密的物体表面, 如图 2 所示。面元的大小与局部采样密度相关, 一般来说物体表面曲率大的地方采样比较密集, 面元较小; 曲率小的地方采样比较稀疏, 面元较大。

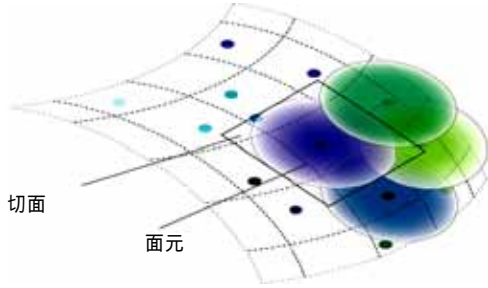


图 2 物体空间中的面元

一个面元加上一个 alpha 掩码 (alpha mask) 构成一个重建核(reconstruction kernel), alpha 掩码通常由面元上的一个截断高斯分布函数计算得到, 主要用于计算重建核在像素上的权重。重建核到图像空间的投影实际上是一个二维空间到另一二维空间的映射, 即把重建核在切平面上的局部信息转换成图像空间中的全局信息。这一过程分两步走, 先把重建核在切平面上的局部信息转换成三维物体空间中的全局信息, 再通过模型视图矩阵投影到图像空间, 如图 3 所示。

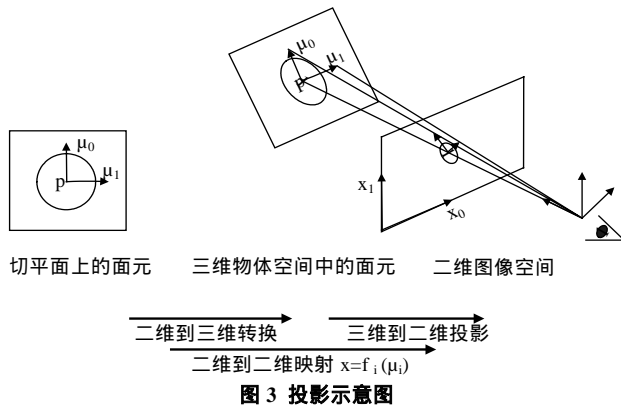


图 3 投影示意图

2.2 图像重建

重建核在图像空间上的投影称为 Splat, 这种将整个重建核投影到图像空间的方法称为 Splatting。图像空间上的单个像素常常被多个 Splat 覆盖, 我们需要判定哪些 Splat 是可见的, 需要被融合; 哪些 Splat 是被遮挡的, 需要被剔除。通常采用所谓的 epsilon-z-buffering 算法来实现, 它是 z-buffering 算法的扩展。其主要思想是对于每个像素, 所有与离视点最近的 Splat 的距离差距小于给定阈值 epsilon 的 Splat 都是可见的, 应该要融合; 反之, 都是不可见的, 应该被剔除。

理想上, 背景剔除和 Splat 融合后, 各个像素的权值和都应该为 1。但由于采样点的不规则分布, 各像素的权值和往往不是 1。如果不进行归一化处理, 将会导致显而易见的瑕疵, 如图 4 所示。

Zwicker^[10]等用软件方法实现了上述算法思想, 算法描述如下:

```
Splatting{
  for each sample point {
    shade surface sample;
```

```
splat = projected reconstruction kernel;
TestDepth(x,y,splat);
}
for each output pixel {
  Normalize(x,y);
}
DepthTest(x,y,splat) {
  if (abs(splat z - z(x,y)) < threshold) {
    c(x,y) = c(x,y) + splat color;
    w(x,y) = w(x,y) + splat w(x,y);
  } else if (splat z < z(x,y)) {
    z(x,y) = splat z;
    c(x,y) = splat color;
    w(x,y) = splat w(x,y);
  }
}
Normalize(x,y){
  c(x,y) = c(x,y)/w(x,y);
}
```

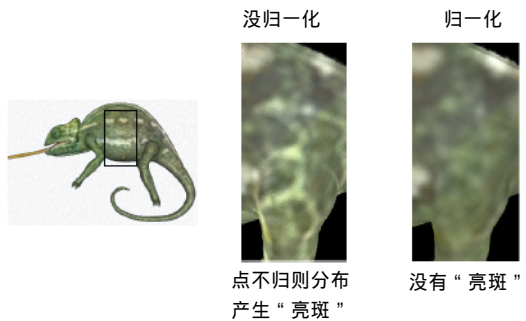


图 4 归一化和未归一化效果对比图

2.3 滤波

由融合和归一化重建得到的连续表面并不能消除走样现象, 特别是模型离视点非常近(低频信号)或非常远(高频信号)时。根据信号处理理论, 走样现象是由信号的带宽不符合 Nyquist 频率造成的。解决的方法是在重取样之前对其进行滤波处理, 即输出信号与一个低通滤波因子(low-pass filter)卷积(convolution), 消除其中某些频率变化过大的点, 使最终的输出信号的带宽符合 Nyquist 频率。

卷积是一个线性操作, 而且输出信号是各个像素上所有可见的 Splat 加权平均得到, 因此可以把滤波这一步提到图像重建前执行, 即先滤波再重建, 我们把滤波后的 Splat 称为重取样核(resampling kernel), 如图 5 所示。

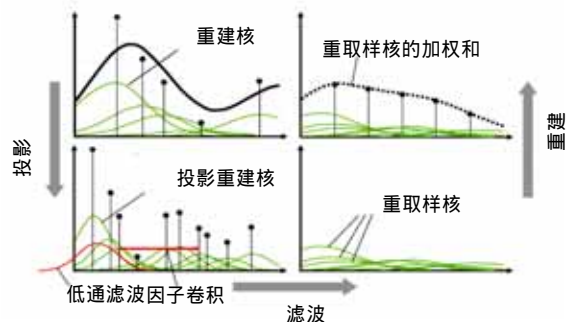


图 5 先滤波后重建示意图

Zwicker^[11]等把变形到图像空间的椭圆高斯函数与低通滤波因子卷积所得的高斯重取样核称作椭圆加权平均(elliptical weighted average, EWA)重取样核。Ren^[12]等在此基

基础上推导出了物体空间的椭圆加权平均重取样核，即由高斯函数与变形到物体空间的低通滤波因子卷积所构成，可获得最新图形的部分支持。

3 点绘制的硬件实现

基于软件实现的采用高斯过滤的 Splat 作为绘制单元的点绘制方法虽然能获得很高的显示质量，但绘制速度受到限制，其主要缺点在于 CPU 负担太重，投影、融合等操作都需要 CPU 来处理。另一方面，现代计算机图形处理单元 (Graphic processing units, GPUs) 的处理能力和可编程性能都有了很大提高，一些由 CPU 执行的操作完全可以交由 GPUs 来完成。

基于硬件实现时，必须解决 2 个问题：(1)现今的图形处理单元并不直接支持 epsilon-z-buffering 算法；(2)归一化必须在所有 Splat 都处理完后才能进行，需要在帧缓存上额外增加一步操作。因此，基于硬件实现的方法分 3 步走。(1)只将 Splat 绘制到 z-buffer，在绘制的时候将每个像素的深度 z 加上一个 epsilon；(2)仍然进行深度测试，但不进行深度更新，深度值大于 z-buffer 中对应值的 Splat 将被剔除，所有通过深度测试的 Splat 进行加权平均，其中权重和则通过帧缓存的 alpha 通道计算得到；(3)把帧缓存当作一个纹理，这样每个纹元与屏幕的像素一一对应，可利用一个简单的 GPU 片元程序 (fragment program) 实现像素归一化。

4 结语

实践表明，用点绘制技术来表达形状复杂且不规则的物体时，不仅能实现很高的绘制效率，而且能获得高质量的绘制效果。但作为一种新发展的技术，在很多地方有待完善和发展。比如，如何动态确定采样密度又确保预处理时间不会成倍增加；传统真实感图形学所研究的如光线跟踪、阴影生成等在点绘制中的相应问题；发展支持点绘制的图形硬件；点绘制的并行实现，都是点绘制未来的研究方向。

参考文献

1 Levoy M, Whitted T. The Use of Points as Display Primitives[R]. Technical Report, CS Department, University of North Carolina at Chapel Hill, 1985.

2 Grossman J P, Dally W J. Point Sample Rendering[C]. Proc. of Eurographics Workshop on Rendering, Vienna, Austria, 1998: 181-192.

3 Rusinkiewicz S, Levoy M. QSplat: A Multiresolution Point Rendering System for Large Meshes[C]//Proceedings of SIGGRAPH, New Orleans, Louisiana, USA. 2000: 343-352.

4 Matusik W, Buehler C, Raskar R, et al. Image-based Visual Hulls[C]//Proceedings of SIGGRAPH, New Orleans, Louisiana, USA. 2000: 369-374.

5 Chen B, Nguyen M X. Pop: A Hybrid Point and Polygon Rendering System for Large Data[C]//Proc. of IEEE Visualization, San Diego California, USA. 2001: 45-52.

6 Wand M, Fischer M, Peter I. The Randomized Z-buffer Algorithm: Interactive Rendering of Highly Complex Scenes[C]//Proceedings of SIGGRAPH, Los Angeles, California, USA. 2001: 361-370.

7 Deussen O, Colditz C, Stamminger M, et al. Interactive Visualization of Complex Plant Ecosystems[C]//Proc. of IEEE Visualization, Boston, Massachusetts, USA. 2002: 265-271.

8 Zwicker M, Pauly M, Knoll O, et al. PointShop3D: An Interactive System for Point-based Surface Editing[C]//Proceedings of SIGGRAPH, San Antonio, Texas, USA. 2002: 322-329.

9 Pfister H, Zwicker M, Baar J, et al. Surfels: Surface Elements as Rendering Primitives[C]//Proceedings of SIGGRAPH, New Orleans, Louisiana, USA. 2000: 335-342.

10 Zwicker M, Pfister H, Baar J, et al. Surface Splatting[C]//Proceedings of SIGGRAPH, Los Angeles, California, USA. 2001: 371-378.

11 Zwicker M, Pfister H, Baar J, et al. EWA Splatting[J]. IEEE Transactions on Visualization and Computer Graphics, 2002, 8(3): 223-228.

12 Ren L, Pfister H, Zwicker M. Object Space EWA Surface Splatting: A Hardware Accelerated Approach to High Quality Point Rendering [C]//Proceedings of Eurographics, Saarbrücken, Germany. 2002: 461-470.

(上接第 153 页)

太多的约束时，在 3 步协议之外的搜索没有带来额外的增益，这是因为当有更少的约束时在先前的搜索中已经找到一个非常好的解，而当太多的约束时，它已经很难找到一个更好的解作为额外搜索的一个结果 (图 3 中，解质量是与最好解相比较的相对质量，数字 100 意味着最好解)。

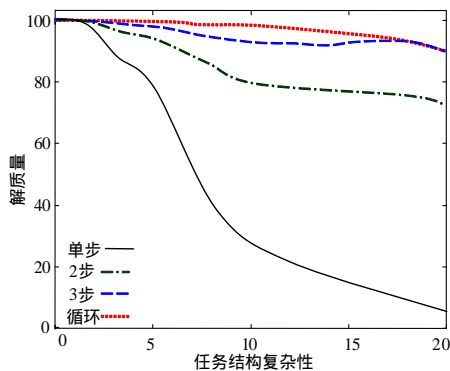


图 3 协议执行的解质量随任务复杂性的变化曲线

4 结论

防空监视网络中的每个传感器代理有关于动态和不确定环境的不完全信息，基于传感器代理协商实现传感器资源的动态分配是一个有效的方法。我们设计一个多维、多步的协商策略，发起协商的代理与响应代理之间就传感器的完成协作任务的资源约束和时间约束进行协商，以期任务能在最合适的时间以最大的合成效用得以执行。各传感器代理基于局部的环境感知，协商实现传感器资源的动态分配，充分利用了传感器代理的自治性和协作性，能够满足实现防空监视网络传感器资源的动态分配的实时性和准确性要求。

参考文献

1 黄树彩, 李为民, 李威. 基于多代理技术的防空监视网络传感器协作管理方法[J]. 传感器技术, 2005, 24(3): 38-40.

2 Lopez J M M, Corredera J R C. Cooperative Management of Netted Surveillance Sensors[EB/OL]. 2003-08. <http://www.go.com/>.

3 Zhang Xiaojin, Lesser V, Podorozhny R. New Result on Cooperative, MultiStep Negotiation over a Multi-dimensional Utility Function [EB/OL]. 2001-02. <http://www.go.com/>.

