

保留细节特征的轮廓线远程绘制算法

刘瑞芳 梁晓辉 谢可 沈旭昆

(北京航空航天大学计算机学院虚拟现实技术与系统国家重点实验室, 北京 100191)

摘要 目前,智能掌上电脑、个人数字助理和智能手机等移动终端因为其便携性和可移动性,越来越多地被应用于分布式虚拟环境中,但是因为移动终端自身的局限性,在移动终端上的 3 维模型实时绘制问题仍未得到很好的解决。结合多分辨率网格模型和基于轮廓线的远程绘制的优点,提出了一种保留细节特征的轮廓线远程绘制算法。该算法预处理阶段,在原始精细网格模型上提取细节特征线,并进行多分辨率网格模型的预处理;运行时阶段,根据视点、帧率等交互信息使用选取策略对细节特征线进行选取,然后构建合适的多分辨率网格模型,并在该模型上提取出外围轮廓线,最后发送这两种轮廓线到移动终端进行绘制。实验结果表明,该算法在基于轮廓线的远程绘制中较好地保留了模型的细节特征,并保证了在移动终端的实时绘制。

关键词 移动终端 细节特征保留 轮廓线 多分辨率网格

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2009)02-0346-07

Feature-preserving Contour-based Remote Rendering Algorithm

LIU Rui-fang, LIANG Xiao-hui, XIE Ke, SHEN Xu-kun

(State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191)

Abstract Nowadays, mobile devices such as Pocket PC, PDA and Smartphone are used in Distributed Virtual Environment because of their portability and mobility. But mobile devices have their own limitations. For instance real-time rendering of 3D models on the mobile devices have not yet been satisfactorily resolved. In this paper, a new contour-based remote rendering algorithm for mobile device applications is proposed, which integrates the advantages of multi-resolution mesh and contour-based remote rendering, and can preserve the feature details of the models. At the preprocessing stage, the algorithm extracts the feature lines from the original mesh and then preprocess for multi-resolution mesh construction; At the real-time stage, the algorithm selects the final feature lines using a selection strategy and then constructs the appropriate multi-resolution mesh according to the interaction information, such as the viewpoint and frame rate, then extracts the contour lines from this mesh. At last, it sends the selected feature lines and the contour lines to the mobile device for rendering. Experiments show that the algorithm can preserve the details well in contour-based remote rendering, and can assure the rendering to be real-time on mobile device.

Keywords mobile device, feature preserving, contour line, multi-resolution mesh

1 引言

近年来,智能手机(Smartphone)、个人数字助理

(PDA)等移动终端持续快速增长,由于其具备良好的便携和移动特性,因此,被越来越多地应用于虚拟现实系统中。然而,移动终端自身存在的特点也导

基金项目:国家自然科学基金项目(6087,3159,6053,3070);教育部新世纪优秀人才支持计划项目(NCET-07-0039);国家高技术研究发展计划(863)项目(2006AA01Z333)

收稿日期:2008-08-10;改回日期:2008-11-06

第一作者简介:刘瑞芳(1983~),女。北京航空航天大学计算机学院硕士研究生。主要研究方向为计算机图形学与分布式虚拟现实。

E-mail:liurf@vrlab.buaa.edu.cn

致了其应用的局限性,主要表现在:(1)移动终端使用无线网络,带宽有限,对于数据量较大的3维模型数据来说容易造成网络拥塞或延迟;(2)移动终端的存储资源、计算资源有限,屏幕分辨率低;(3)移动终端的绘制能力有限。因此,如何在满足实时的前提下实现尽可能逼真的绘制成为研究的关键问题。

目前移动终端上的绘制方法主要分为本地绘制和远程绘制两种。本地绘制需要把大量的模型数据存储在移动终端上,这不仅占用移动终端的存储空间,而且由于其计算能力和绘制能力的限制,绘制效果和帧率都不能满足实际应用的需求。

远程绘制包括服务器端和客户端两部分,通过把大量的计算工作和模型数据转移到服务器端,来有效地减轻作为客户端的移动终端的负担,提高绘制帧率。目前研究主要集中在基于图像和基于轮廓线两类方法上。基于图像的方法是在服务器端根据用户视点等信息进行绘制,将绘制结果以图像的方式发送给客户端进行显示。如 Bao 等人提出在服务器上绘制3维模型^[1],然后把包括像素颜色和深度信息的初始深度图像传给客户端,当视点发生变化时使用 IBR (image based rendering) 算法,每次只传输“剩余图像”给客户端;Lamberti 等人提出使用一组具有硬件加速的服务器 cluster^[2],在这些服务器中根据 Chromium 体系结构^[3]对 OpenGL 应用的计算任务进行划分,然后分配给各个 GPU (graphics process unit) 进行绘制,绘制的结果以一个静态的图像流发送到移动终端;Lamberti 等人对服务器绘制的结果图像进行 MPEG 编码^[4],在客户端进行相应的解码操作。但基于图像的方法在移动终端上只保留图像信息,而不是模型信息,因此,只适用于场景漫游等应用,无法满足虚拟现实应用中交互性的需求,而且图像的数据量比较大。

基于轮廓线的方法,即在服务器端根据模型的几何信息提取轮廓线,把轮廓线传输给移动终端进行绘制。针对现有模型高度复杂和细节化的特点,为达到在移动终端的实时绘制,目前基于轮廓线方法的研究主要集中在传输策略和利用多分辨率模型两方面,如 Quillet 等人尝试把使用轮廓线的非真实感绘制应用于3维城市的远程可视化及漫游^[5],根据拓扑结构把场景划分为几个单元,进行轮廓线的预提取;国内黄月珠等人提出设置质量水平参数^[6],该参数有两个取值,为0时只传输 silhouette 和边界线,为1时传输包括 silhouette、边界线、脊线

和谷线4种类型的所有轮廓线;黄月珠等人还提出由用户根据各类线条对画面的贡献大小^[7],指定各类线条的权重,根据权重的大小来进行轮廓线的传输,以减少用户等待时间。但是传输策略的改进只是减少了网络延迟,并没有考虑到在移动终端绘制所能达到的帧率。在利用多分辨率模型方面,Hekmatzada 等人及黄月珠等人采用渐近网格得到不同分辨率的细节层次 (level of detail, LOD) 模型^[7-8],在运行时刻选择合适的 LOD 模型提取轮廓线并传输,但是简化模型必然导致细节特征的损失,而现有方法并没有考虑到对模型细节特征信息的保留。

针对上述问题,为了提高绘制帧率,实现尽可能逼真的实时绘制,本文结合多分辨率网格模型和基于轮廓线的远程绘制的优点,提出了一种面向移动终端保留细节特征的轮廓线远程绘制算法。实验结果表明,该算法在简化模型,减少轮廓线数据量的同时能有效地保留模型的细节特征,达到在移动终端绘制的帧率和逼真度的均衡。

2 保留细节特征的轮廓线远程绘制算法

2.1 基本概念

轮廓线是3维模型的一种特征值,根据轮廓线就可以识别出模型的3维结构和细节几何信息。本文讨论的轮廓线包括外围轮廓线和细节特征线两类。

(1) 外围轮廓线 (E_{contour})

对于边 e ,它的两个相邻面的法向量分别为 n_1 , n_2 , 视线向量为 n_e 。

如果 $(n_1 \cdot n_e) * (n_2 \cdot n_e) < 0$, 则 $e \in E_{\text{contour}}$

如图1所示,外围轮廓线与视点相关,需要每帧都进行重新计算。

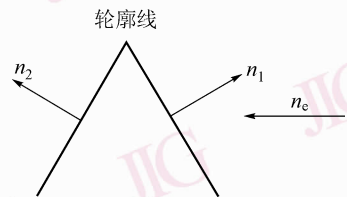


图1 外围轮廓线

Fig. 1 Contour

(2) 细节特征线 (E_{feature})

对于边 e ,它的两个相邻面的法向量分别为 n_1 , n_2 。

如果 $\cos^{-1}\left(\frac{\mathbf{n}_1 \cdot \mathbf{n}_2}{|\mathbf{n}_1| * |\mathbf{n}_2|}\right) < \theta$, 则 $e \in E_{\text{feature}}$ (θ 为一个阈值, 一般设为 60)。

细节特征线不随视点的变化而变化。

2.2 保留细节特征的轮廓线远程绘制算法

基于轮廓线的远程绘制研究的重点在于如何在减少传输数据量(即轮廓线数目)的前提下尽量保

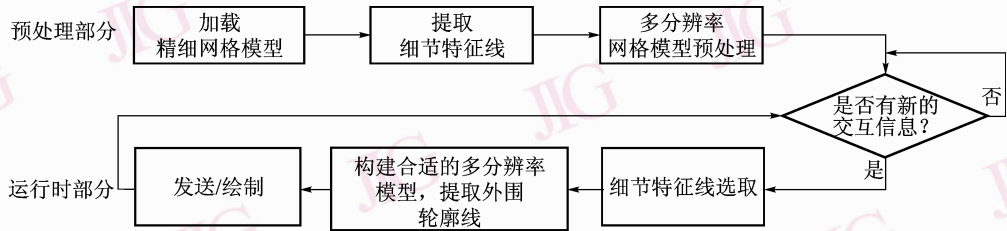


图 2 算法流程图

Fig. 2 Algorithm flow chart

(1) 预处理部分 设定阈值 θ , 根据基本概念(2)从原始精细网格模型中提取细节特征线 E_{feature} 。采用 Garland 等人的基于二次误差度量的网格简化算法对原始精细网格模型进行多分辨率网格模型预处理^[9]。

(2) 运行时部分 当有新的交互信息(如视点信息、绘制帧率等)时, ①服务器端根据用户视点信息, 使用选取策略对细节特征线 E_{feature} 进行选取, 得到 \hat{E}_{feature} ; ②根据用户希望达到的在移动终端的绘制帧率 f , 由映射函数 $g: f \rightarrow L$ 得到需要传输的轮廓线数目 L (这里的映射关系为由实际在移动终端上的绘制实验测试得到的经验公式); ③计算 L 与 \hat{E}_{feature} 中细节特征线的数目 L_1 的差值得到需要提取的外围轮廓线数目 L_2 , 然后由边与面片对应的经验公式得到多分辨率网格模型的面片数 F , 并以此为标准构建合适的多分辨率网格模型; ④在该多分辨率网格模型上依据简化后模型外围轮廓线的提取方法提取出外围轮廓线 E_{contour} ; ⑤把 \hat{E}_{feature} 和 E_{contour} 的集合发送给客户端进行绘制。这部分是本文的重点, 接下来对其中的选取策略和外围轮廓线的提取分别予以介绍。

2.2.1 E_{feature} 选取策略

传统算法中, 对于提取出的细节特征线不管它们距离视点远近, 都传输给客户端进行绘制, 但如果这样的话, 从远处看, 场景中充满了线条, 影响图像的效果, 因此, 本文采用 LOD 的方法来解决这个问题。依据视点距离被观察场景的远近, 选择提取

证绘制逼真度。本文提出的细节特征保留轮廓线远程绘制算法处理对象为三角网格模型。通过在原始精细网格模型上提取细节特征线来实现对模型细节特征的保留, 运行时在选取的多分辨率网格模型上提取外围轮廓线, 然后对两类轮廓线融合以尽可能逼真的表现模型。其具体过程如图 2 所示。

的细节特征线集的一个子集来发送和绘制, 视点越远, 这个子集越小, 这样可以更进一步地减少数据量, 并在一定程度上提高绘制帧率。

边的长度和边距离视点的远近对于它在最后生成的绘制图像的贡献来说都是非常重要的因素, 如果对于一个距离视点很远的边, 它的长度很短的话, 对于最后的图像, 它可能只是显示为一个点, 对于这类型的边, 可以把它们删除不进行发送和绘制, 因此, 本文以边的长度和边距离视点的距离为基准, 定义一个选择因子 m , 并根据实际应用人工确定一个阈值 T 来对预处理部分提取的细节特征线 E_{feature} 进行选择, 最后得出需要发送/绘制的细节特征线的集合 \hat{E}_{feature} 。选择策略具体的步骤如下:

遍历细节特征线集 E_{feature} 中的每条边 $e(v_1, v_2)$, v_1, v_2 为边 e 的两个顶点, v_{eye} 为视点位置向量。

(1) 计算该边的长度 l , 距离 v_{eye} 的距离 d 。

$$l = \sqrt{(v_1 \cdot x - v_2 \cdot x)^2 + (v_1 \cdot y - v_2 \cdot y)^2 + (v_1 \cdot z - v_2 \cdot z)^2} \quad (1)$$

$$d = \left[(v_{\text{eye}} \cdot x - v_{\text{mid}} \cdot x)^2 + (v_{\text{eye}} \cdot y - v_{\text{mid}} \cdot y)^2 + (v_{\text{eye}} \cdot z - v_{\text{mid}} \cdot z)^2 \right]^{1/2} \quad (2)$$

式中, $v_{\text{mid}} = (v_1 + v_2) / 2$ 。

(2) 计算选择因子 m , $m = l/d$ 。

(3) 进行选择, if $m > T$, 则 $e \in \hat{E}_{\text{feature}}$ 。

其中, 式(1)和式(2)可以不进行开方计算。

2.2.2 满足绘制帧率的外围轮廓线提取

按照本文算法流程, 根据交互信息中的绘制帧

率构建合适的多分辨率网格模型后,就需要从该模型中提取与视点相关的外围轮廓线。对于原始精细网格来说,提取外围轮廓线,只需遍历网格模型的每条边,判断它相邻两个面的法向量与视线向量的点积的乘积,如果为负,则该边为外围轮廓边,但是,对于简化后的多分辨率网格模型,采用这种方法提取到的轮廓线不连续,因此,本文在外围轮廓线提取时,提出使用线性插值方法以有效地得到简化后模型的外围轮廓线。具体步骤如下:

遍历简化网格模型的每个三角面片 $\Delta(a, b, c)$, a, b, c 为该三角面片的 3 个顶点,其顶点位置为 v_a, v_b 和 v_c , 顶点法向量分别为 n_a, n_b 和 n_c, v_{eye} 为视点位置向量。

(1) 计算顶点 i 的单位点积式 d_i 和符号值 S_i 。

$$d_i = \frac{n_i \cdot (v_i - v_{eye})}{|n_i| |v_i - v_{eye}|} \quad (i = a, b, c) \quad (3)$$

$$S_i = \begin{cases} + & d_i \geq 0 \\ - & d_i < 0 \end{cases} \quad (i = a, b, c) \quad (4)$$

(2) 对于面片的每条边,如果该边两个顶点的符号值为一正一负,则说明这条边上有一个 d 值为 0 的点,这里称这个点为轮廓点,并通过线性插值来计算这个点的位置

$$V' = \frac{|d_i|}{|d_i| + |d_j|} V_j + \frac{|d_j|}{|d_i| + |d_j|} V_i \quad (5)$$

式中, $i = a, b, c; j = a, b, c; i \neq j$ 且 $S_i \cdot S_j < 0$ 。

(3) 连接同一面片中的两个轮廓点得到边 e , 则 $e \in E_{contour}$ 。

使用这种方法提取的外围轮廓线已经不是真实的物体轮廓线,而且对于十分简化的图形来说,可能导致轮廓线失真。

运行时,使用上述两种方法按照本文算法流程求得最后需要发送/绘制的两类轮廓线 $\hat{E}_{feature}$ 与 $E_{contour}$, 把它们的集合发送给客户端进行绘制。

3 实验及结果

3.1 单机实验及结果

本文所采用的单机实验的平台为 Intel Core 2 2.2 GHz, 主内存为 2 GB, 显卡为 256 MB 显存的 NVIDIA GeForce 8600。所有实验均在 Windows XP 下实现。实验中, 阈值 θ 的取值为 60, T 的取值为 0.005。实验主要对本文算法与传统的轮廓线远程绘制算法的性能和绘制效果进行比较, 结果如表 1 和图 3 所示。

表 1 单机实验中本文算法与传统算法的数据对比

Tab. 1 Comparative data of our algorithm and traditional algorithm in stand-alone experiment

| 模型 | 传统算法 | | | | 本文算法 | | | |
|-------|--------|--------|--------|---------|-------|--------|--------|---------|
| | 面片数 | 细节特征线数 | 外围轮廓线数 | 执行时间(s) | 面片数 | 细节特征线数 | 外围轮廓线数 | 执行时间(s) |
| cow | 5 804 | 230 | 706 | 0.15 | 2 060 | 189 | 306 | 0.03 |
| bunny | 69 451 | 80 | 2 497 | 1.75 | 3 330 | 33 | 349 | 0.05 |
| duck | 41 264 | 389 | 2 214 | 1.12 | 2 783 | 248 | 400 | 0.12 |

其中,每一帧的执行时间包括计算新的视点信息从模型中提取轮廓线并进行绘制的时间(其中并不包括预处理的时间)。从表 1 可以看出,使用本文的算法,模型的数据量及每帧算法的执行时间得到了显著的提高。以 bunny 模型为例,使用本文的算法后,传输的两类轮廓线的数由 2 500 条左右减少为 400 条左右,数据量明显减少,同时,每帧的执行时间由 1.75 s 减少为 0.05 s,执行效率得到了显著的提高。

由图 3 中绘制效果对比图可以看出,在使用本文的算法有效地减少了数据量的同时,模型的细节

特征也得到很好地保留。特别是图 3 中的 bunny 模型及 duck 模型,在数据量减少 50% 以上的基础上,模型的外围轮廓及细节特征仍然得到很好的表现。

图 4 为使用保留细节特征的本文算法和使用未保留细节特征的其他已有算法在不同的视点下绘制的 cow 模型对比,从图中可以看出,在不同的视点下,使用本文的算法,模型的眼睛、耳朵、嘴巴以及腿部关节等细节特征都得到了很好的保留,但是相比较基于精细网格模型的轮廓线绘制来说,细节特征还是有一些丢失。

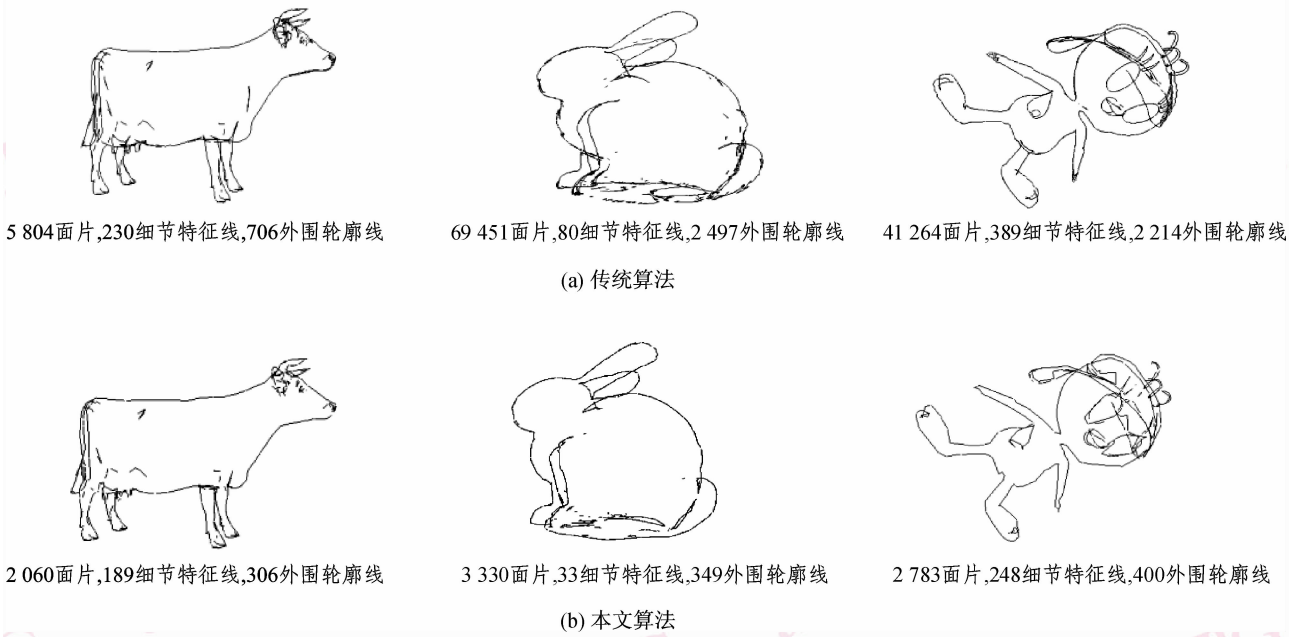


图 3 绘制效果对比图

Fig. 3 Rendering comparison

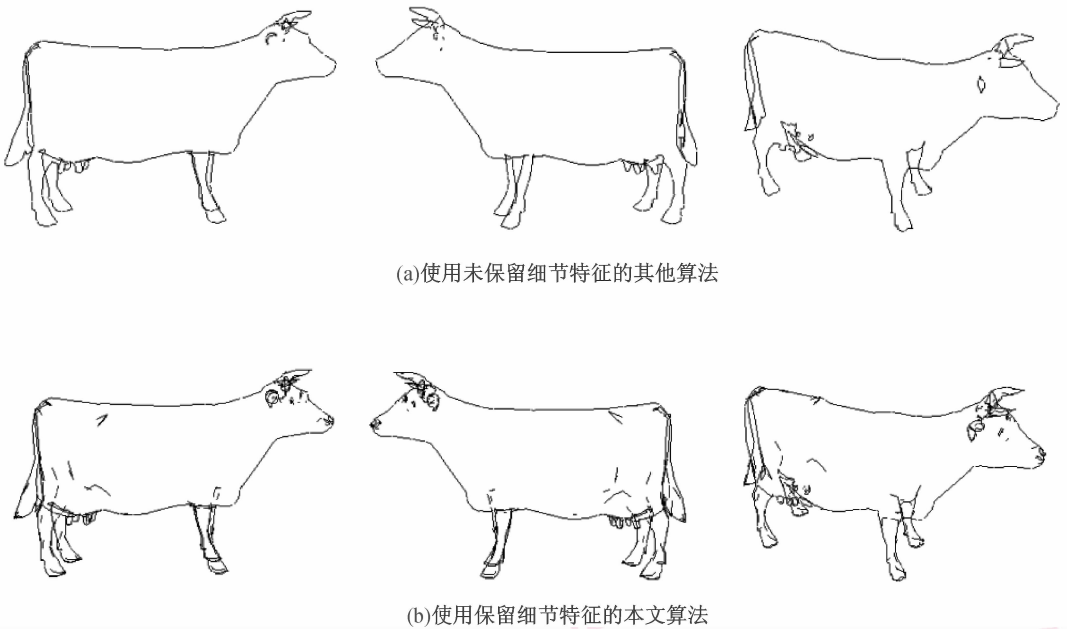


图 4 在不同视点下的 cow 模型

Fig. 4 Cow model on different viewpoint

3.2 远程绘制实验及结果

上面的实验有效地证明了本文的算法在简化模型,减少轮廓线数据量的同时能有效地保留模型的细节特征。下面将主要验证本文算法应用于移动终端时的效果。实验中,远程绘制系统采用服务器/客

户端框架,之间使用无线网络来进行通信和数据传输。3 维模型存储在服务器端,服务器端完成模型的读取以及对客户端绘制所需的数据进行提取和处理,然后传输到客户端,客户端对收到的数据进行绘制,当用户产生交互时,根据交互类型,选择

在客户端进行适当的计算响应交互或者把交互信息传给服务器端,由服务器完成对该交互操作的响应。

本文系统的服务器配置与上节中的PC机相同,使用VC++来实现服务器各模块的功能。客户端使用vs2005实现基于Pocket PC2003和Windows Mobile4.2的客户端程序开发,使用OpenGL ES提供的接口来进行客户端的绘制,服务器与客户端之间使用ACE来进行数据传输和通信。系统在PPC(pocket PC)模拟器和HP iPAQ212的真实环境下进行下面的实验。实验结果如表2和图5所示。

表2 远程绘制实验中本文算法与传统算法数据对比

Tab.2 Comparative data of our algorithm and traditional algorithm in remote rendering experiment

| 模型 | 传统算法 | | | 本文算法 | | |
|-------|------------|--------------|---------------|------------|--------------|---------------|
| | RTT (s) | Data (KB) | fps in PDA | RTT (s) | Data (KB) | fps in PDA |
| cow | 0.282 9 | 21.9 | 10 | 0.108 9 | 11.6 | 27 |
| bunny | 2.219 11 | 60.4 | 3 | 0.111 | 8.95 | 31 |
| duck | 1.89 | 61 | 3.7 | 0.215 93 | 15.19 | 17.8 |

在上表中,响应时间RTT(round trip time)表示从移动终端PDA计算新的视点并发出交互信息到接收从服务器端传过来的数据完成绘制为止之间的时间间隔,单位为秒;Data表示在服务器和移动终端

之间传输的数据量的大小,单位为字节。fps in PDA为在PDA端的绘制帧率。两种方法所用的模型的几何数据与上节实验相同。从表2可以看出,使用本文的方法后,移动终端交互的响应时间得到了明显的提高,PDA端的绘制帧率也基本达到了实时,满足应用的需求。

RTT时间包含了以下几个阶段的耗时:(1)视点计算并发送,(2)服务器接收交互信息,提取轮廓线并发送给客户端,(3)客户端接收服务器发送过来的轮廓线数据,(4)绘制,(5)网络上花费的时间。表3是对RTT各阶段的时间分析。

表3 RTT各阶段时间分析

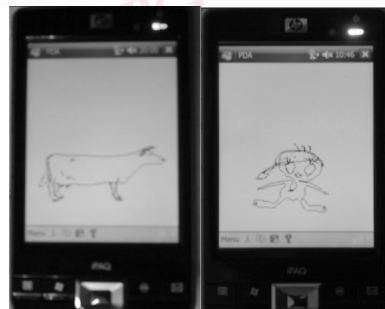
Tab.3 Time spent analysis

| 阶段 | 时间(s) | | |
|------------|---------|-------|---------|
| | cow | bunny | duck |
| 计算新的视点并发送 | 0.001 | 0.001 | 0.001 |
| 提取并发送两类轮廓线 | 0.025 | 0.036 | 0.093 |
| 接收轮廓线数据 | 0.012 | 0.009 | 0.014 |
| 绘制 | 0.031 | 0.032 | 0.059 |
| 网络传输 | 0.04 | 0.033 | 0.04 |
| 总计 | 0.108 9 | 0.111 | 0.215 9 |

图5为本实验在PPC模拟器及PDA实体机上的绘制效果图,可以看出,使用本文算法提取的轮廓线可以很好的表现模型的外围轮廓和细节特征,基本可以实现在PDA端对模型的近似逼真绘制。



(a)在PPC模拟器绘制效果图



(b)在PDA实体机绘制效果图

图5 PPC模拟器及PDA实体机绘制效果图

Fig.5 Rendering on PPC Simulator and PDA

4 结论

本文结合多分辨率网格模型和基于轮廓线的远程绘制的优点,提出了一种面向移动终端保留细节

特征的轮廓线远程绘制算法。通过在原始精细网格上提取细节特征线来进行细节特征保留,然后再根据交互信息构建的多分辨率模型上提取外围轮廓线,在减少数据量的同时有效地保留模型细节特征。本文算法较之以往的算法主要有以下优点:第一,有

效的减少了远程绘制中需要传输的数据量;第二,通过在原始精细网格上提取细节特征线来对细节特征进行保留,达到在移动终端绘制的帧率和逼真度的均衡,满足虚拟现实系统中对于实时性的需求。实验证明,本文算法对于较大的模型仍然能达到实时的绘制要求,并较大程度地保证了绘制的逼真性。但是现有的研究对于虚拟现实系统中的交互性仍然不能满足需求,接下来还需不断增强对于远程绘制系统中交互性和可操作性的研究,以满足更广泛的应用需求。

致 谢 感谢北京航空航天大学研究生创新实践基金对本论文研究的资助。

参考文献 (References)

- 1 Bao P, Gourley D. Low bandwidth remote rendering using 3D image warping [A]. In: Proceedings of International Conference on Visual Information Engineering [C], Guildford, Surrey, UK, 2003:61-64.
- 2 Lamberti F, Zunino C, Sanna A, *et al.* An accelerated remote graphics architecture for PDAs [A]. In: Proceedings of the Eighth International Conference on 3D Web Technology [C], Saint Malo, France, 2003:55-62.
- 3 Humphreys G, Houston M, Ng R, *et al.* Chromium: a stream-processing framework for interactive rendering on clusters [J]. ACM Transactions on Graphics, 2002, **21**(3): 693-702.
- 4 Lamberti F, Sanna A. A streaming-based solution for remote visualization of 3D graphics on mobile devices [J]. IEEE Transactions on Visualization and Computer Graphics, 2007, **13**(2): 247-260.
- 5 Quillet J C, Thomas G, Granier X, *et al.* Using expressive rendering for remote visualization of large city models [A]. In: Proceedings of the Eleventh International Conference on 3D Web Technology [C], Columbia, Maryland, USA, 2006: 27-35.
- 6 Huang Y Z, Yang C L, Meng X X, *et al.* Remote non-photorealistic rendering of 3D models on mobile devices [A]. In: Proceedings of the First International Symposium on Pervasive Computing and Applications [C], Urumchi, China, 2006:364-369.
- 7 Huang Yue-zhu, Yang Cheng-le, Meng Xiang-xu, *et al.* Stylistic line drawings of 3D models on mobile devices [J]. Journal of Computer-Aided Design & Computer Graphics, 2007, **19**(8): 1075-1079. [黄月珠, 杨承磊, 孟祥旭等. 移动设备上基于三维模型的线条画绘制 [J]. 计算机辅助设计与图形学学报, 2007, **19**(8): 1075-1079.]
- 8 Hekmatzade D, Meseth J, Klein R. Non-photorealistic rendering of complex 3D models on mobile devices [A]. In: Proceedings of the 8th Annual Conference of the International Association for Mathematical Geology [C], Berlin, Germany, 2002:143-148.
- 9 Garland M, Heckbert P S. Surface simplification using quadric error metrics [A]. In: Proceedings of ACM SIGGRAPH '97 [C], Los Angeles, CA, USA, 1997: 209-216.