

基于混合式绘制技术的分布式虚拟环境系统*

郑文庭 鲍虎军 彭群生

(浙江大学 CAD&CG 国家重点实验室, 杭州 310027)

Hanqiu Sun

(Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, China)

摘要 给出了一个基于分布式计算技术的虚拟环境原型系统。系统根据当前摄像机的运动趋势,首先利用预测技术实时地确定下帧图象摄像机的方位,然后分布式地采用基于几何的快速绘制算法来生成预测图象,最后采用图象 warping 算法来实时生成前后预测图象间的画面过渡。由于引入了几何信息,因而本系统允许用户对虚拟景物进行直接交互操作,从而克服了传统基于图象的图形绘制技术只能处理静态场景的缺陷。实验结果表明,本系统对场景复杂性的依赖程度低,能在中等计算能力的工作站上实现复杂场景的实时交互。

关键词 虚拟现实 纹理映射 分布计算 并行计算

漫游(walkthrough)是虚拟环境系统应具备的最基本功能。该功能要求系统根据摄像机方位的改变实时地刷新画面,因而,虚拟现实系统的最基本要求是画面的实时生成与显示。

经过几年的发展,真实感图形的实时绘制技术得到了极大的发展,出现了一系列极具潜力的实时图形绘制算法^[1~3],其中有些算法已成功地应用于各类虚拟现实系统中。从技术路线来看,实时图形绘制算法可分为两类:一类是基于几何的;另一类则基于图象。前者以多边形的绘制作作为基本操作,结合硬件 Z-buffer 和一些专门设计的场景简化技术来实现实时绘制,而后者则利用图象作为基本元素,结合硬件纹理映射和重投影技术来实现画面绘制的实时性^[4~7]。这两种技术各有优缺点,基于几何的图形绘制技术容易实现景物的交互操作,所生成的图形质量较好,但它涉及到巨大的消隐计算,难以在中低档硬件平台上实现画面的实时绘制。而基于图象的图形绘制技术(IBR)既可以采用实拍图象,也可以采用计算机生成图象作为画面绘制的基本元素。该技术由于充分利用前后帧画面的连贯性,且避免了复杂的消隐问题,因而能在中低档硬件平台上实现实时绘制,但因该技术的画面重采样特性,导致生成的图形产生走样,中间帧画面上往往存在难以填补的空洞区域。值得指出的是,由于基于图象的图形绘制技术采用预处理技术来生成并存储采样图象,因而该技术仅适合于静态场景的漫游,无法对场景中所含景物进行交互操作。苹果公司的 Quick Time VR 系统^[8]已证明了这一点。为将基于图象的图形绘制技术应用于动态场景,Raskar 等设计了一种虚拟办公室系统^[9],该系统利用

1998-09-21 收稿,1999-01-25 收修改稿

* 霍英东青年教师基金和国家自然科学基金(批准号:69673027)资助项目

计算机视觉理论来实时恢复摄像机拍摄得到图象的深度信息,然后利用多幅图象的重投影技术来重建所需画面.这一系统由于采用了人眼不能觉察的结构光(imperceptible structured light)技术来辅助深度信息的快速恢复,因而,它只能应用于办公室等小范围的场景,对大范围场景则无能为力.

基于作者给出的混合式图形绘制技术^[10],我们设计并实现了一个交互式桌面型虚拟环境系统.系统充分利用了混合式绘制算法的特点,设计了并行、分布式的绘制模式.本文将详细介绍系统的结构、性能及其运动控制原理.

1 系统结构

图象warping技术能高质量、实时地实现两帧视点相邻的深度图象之间的自然过渡^[10],因此,假如我们能交互地生成漫游路径上(或其邻近)的一些具有一定间距的采样深度图象,则可在以任何速度沿该路径漫游场景.为方便起见,我们称这些采样深度图象为预测图象.图1给出了这一过程的说明.

最近发展起来的基于图象的图形绘制技术由于无法对场景中的景物进行交互操作,它的应用受到了极大的限制.而这对基于几何的图形绘制算法来说,却是非常容易实现的.因此我们采用基于几何的图形绘制算法来生成预测图象,而图象warping算法则用来生成中间过渡画面.由于warping算法对场景复杂性依赖程度较低,这一混合绘制模式将极大地提高图形的绘制速度,而且几何信息的引入,使得用户可对景物进行直接交互操作.

在漫游过程中,摄像机的运动由用户交互确定,由于它没有固定的运动路径,因而,我们不能预先生成好各预测采样点处的图象.也就是说,系统必须根据用户的交互输入信息动态确定下一预测采样点的位置,并实时生成该位置的预测图象.图2给出了系统的绘制流程.该流程的顺序不能换位,否则将得到错误的结果.

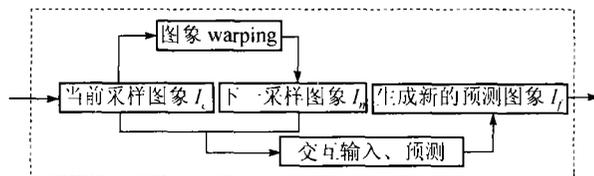


图2 系统的绘制流程

系统进入下一循环,即执行当前采样图象到下一预测图象的warping过渡.这一绘制模式非常适合于并行或分布式执行.

与传统分布、并行算法不同^[11,12],本系统的绘制流程有严格的顺序关系,算法以生成预测图象的进程为父进程,warping过渡进程为子进程.为满足所需的执行顺序,父进程必须等待

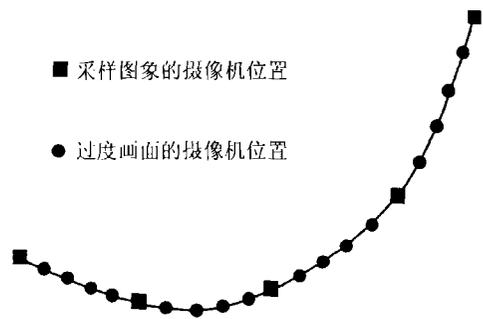


图1 采样图象与过渡画面的关系

2 分布式画面实时绘制策略

由流程图2知,下一帧预测图象的生成与前一帧预测图象向当前帧预测图象的warping过程是同步进行的,而且只有当前者先于后者完成才能保证系统在漫游时没有延迟,不会产生不自然的停顿.进而,系

子进程结束后,才能启动下一轮 warping 过渡和预测循环. 我们采用 UNIX 系统中的 fork() 函数来生成子进程. 父、子进程的执行完全在自己独立的空间中进行,它们之间的通信则通过共享内存技术来完成. 鉴于硬件条件的限制,本文仅考虑系统的分布式实现. 并行策略亦可类似地实现.

分布式绘制策略利用两台以上计算机来协同完成上述绘制过程^[12]. 由于本系统的绘制模式分解为基于几何和基于图象的两个独立绘制过程,而且两者有着严格的执行顺序,即几何绘制在前,图象 warping 过渡在后,因而,我们将交互界面设置在主机上,主机的主要功能是完成预测图象的接收及实现采样图象的 warping 过渡,并向各个从机发送下一帧预测图象的绘制信息及有关交互操作信息. 而从机的主要功能是根据接收到的绘制及交互操作信息,绘制预测图象并往主机发送. 从机可以是多台,这样我们实际上生成了多帧相邻视点的预测采样图象. 若主机的计算能力允许的话,我们可采用多帧预测图象来重建中间过渡画面,从而得到更为精确的绘制结果.

图 3 给出了分布式绘制策略的算法流程. 为清晰起见,我们仅给出一台从机的情形. 值得注意的是,图中所谓的绘制信息是指以几何方式绘制一帧画面所需的所有必需信息,它包括有关摄像机参数及景物几何的变动信息等.

3 运动控制

我们通过控制预测图象的生成来表现对摄像机的运动控制,预测图象间的过渡画面的帧数在系统初始化中被设置成恒定的值. 这样,前、后两相邻帧预测图象的视点间距越大,摄像机的运动就越快. 类似地景物的操作亦通过控制下一预测图象的绘制来实现.

3.1 摄像机的运动控制技术

我们用 3 个鼠标键来控制摄像机的视点及视线方向的运动,即摄像机坐标系的变化. 摄像机坐标系可由一二元组 (E, D) 来定义,其中 $E(x, y, z)$ 为视点坐标, $D(\phi, r, h)$ 定义了该摄像机坐标系关于世界坐标系的朝向. 若将摄像机坐标系初始化为与世界坐标系重合,则由 (E, D) 定义的摄像机坐标系可确定如下:依次将初始摄像机坐标系绕世界坐标系的 z 轴旋转 h 角、绕 x 轴旋转 p 角、绕 y 轴旋转 r 角,然后将它平移到 E 处即得.

由流程图 2 可知,在每一绘制循环中,我们均需同步执行两个步骤,即从当前采样图象 I_c 向下一采样图象 I_n 的 warping 过渡和预测生成新的采样图象 I_f . 不妨记生成这 3 帧采样图象的摄像机方位分别为 (E_c, D_c) , (E_n, D_n) 和 (E_f, D_f) ,相应地,其坐标分量亦以下标 c, n 和 f 来标识. 为保持画面变化的平稳性,我们首先在主机上利用 (E_c, D_c) , (E_n, D_n) 和鼠标状态预测出 (E_f, D_f) ,并输送给从机执行画面绘制. 若已知 (E_c, D_c) 的线速度矢量为 (V_c, ω_c) ,则 (E_f, D_f) 可预测为

$$(E_f, D_f) = (E_n, D_n) + (\Delta E_n, \Delta D_n),$$

其中

$$\begin{aligned} s_n &= \|V_c\| + a, \\ \Delta E_n &= (s_n \sinh_n \cosp_n, s_n \cosh_n \cosp_n, s_n \sin_p_n), \\ \Delta D_n &= (k(m_y - c_y)/R_y, 0, k(m_x - c_x)/R_x). \end{aligned}$$

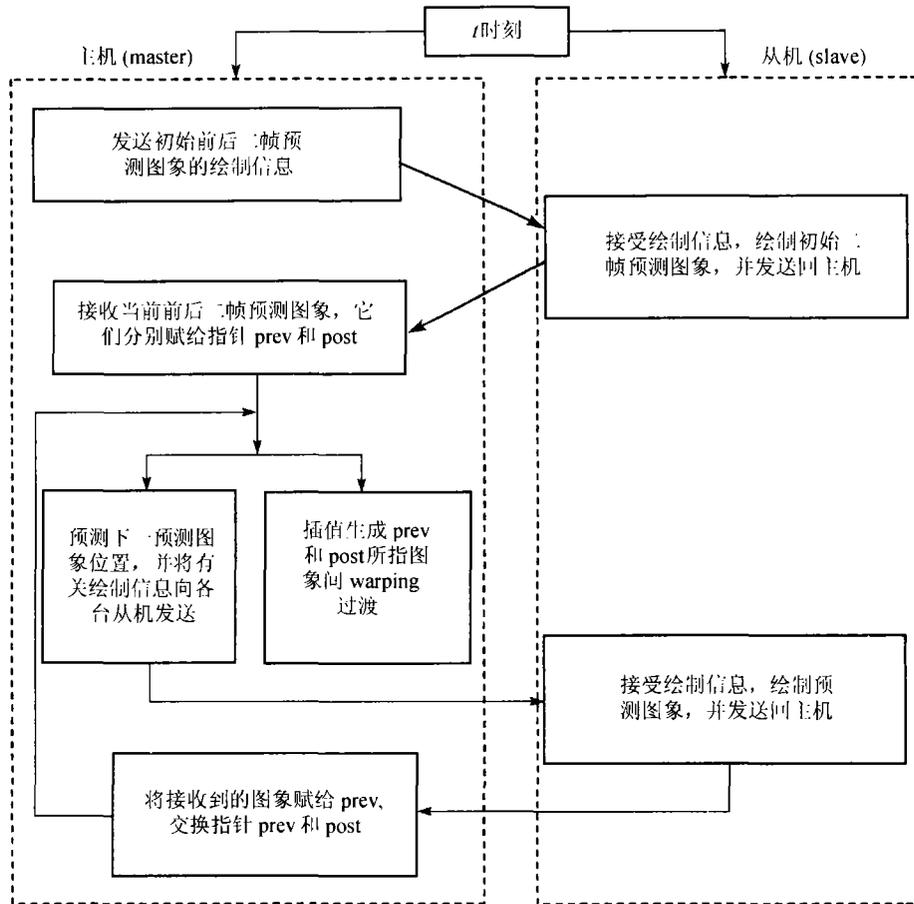


图 3 分布式绘制算法流程

这里, R_x 和 R_y 分别为图象的水平 and 垂直方面分辨率, (m_x, m_y) 为鼠标的屏幕坐标, (c_x, c_y) 为图象中心的屏幕坐标, a, k 为用户给定的常数, 它反映了鼠标按压时间、移动距离与其控制参数间的关系. 按压时间越长, a 值变化越大, 摄像机的运动变化亦越快; 而 k 值越大, 摄像机的方向变化就越灵敏. 因此, 摄像机在 (E_n, D_n) 处的线速度矢量 (V_n, ω_n) 为

$$V_n = s_n \frac{E_f - E_c}{\|E_f - E_c\|}, \omega_n = \|\Delta D_n\| \frac{D_f - D_c}{\|D_f - D_c\|}.$$

一旦从机接收到新的预测采样图象的摄像机方位 (E_f, D_f) , 系统即在从机上采用 Iris Performer 软件平台来绘制预测采样画面并输送回主机, 主机则同步地采用样条插值技术来实现从画面 I_c 向画面 I_n 摄像机方位的光滑 $(C^1$ 或 $G^1)$ 过渡. 假设摄像机以恒定的加速度从 (E_c, D_c) 运动到 (E_n, D_n) , 使得摄像机的运动速度由 (V_c, ω_c) 渐变为 (V_n, ω_n) , 则由运动学方程, 我们可计算出从 (E_c, D_c) 运动到 (E_n, D_n) 所需的时间和加速度. 因而, 时间归一化后的摄像机的运动方程可表达为如下的分量形式

$$\begin{cases} E_{\alpha}(t) = E_{c_{\alpha}} + \frac{E_{n_{\alpha}} - E_{c_{\alpha}}}{V_{c_{\alpha}} + V_{n_{\alpha}}} [2V_{c_{\alpha}} + t(V_{n_{\alpha}} - V_{c_{\alpha}})] t, \\ D_{\alpha}(t) = D_{c_{\alpha}} + \frac{D_{n_{\alpha}} - D_{c_{\alpha}}}{\omega_{c_{\alpha}} + \omega_{n_{\alpha}}} [2\omega_{c_{\alpha}} + t(\omega_{n_{\alpha}} - \omega_{c_{\alpha}})] t, \end{cases} \quad t \in [0, 1]$$

式中下标 α 分别表示相应矢量的 3 个分量. 这 6 条分量样条曲线就构成了所需的 C^1 插值样条. 图 4 绘出了插值过渡样条的构造. 直接采样该样条曲线即可得到中间任一过渡画面的摄像机方位. 为此, 本系统对前后相邻帧预测图象之间设置一恒定的过渡画面帧数 N . 这些采样点对参数区间 $[0, 1]$ 实行均匀采样, 即取 $t = i/(N + 1) (i = 1, 2, \dots, N)$, 它们在上述插值样条上呈非均匀分布. 显然, 这是一个逐步加速或减速的过程, 从而满足了预测采样点附近摄像机的运动速度渐变的条件, 保证了运动过程中的平稳性. 测试表明, 这一方法取得了良好的运动控制效果.

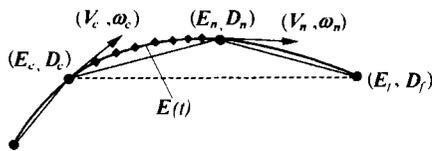


图 4 插值过渡样条的构造

3.2 景物的交互操作

为使生成预测采样图象时可随时改变场景中景物的几何位置, 又能实现从当前帧画面到下一帧场景已局部改变的画面间的连续的采样画面过渡. 系统采用了如下的方法: 当用户采用鼠标或其它硬件设备拾取某一个景物时, 系统自动找到该景物并通知主机和从机这一

景物将被交互操作, 我们将当前所有位置被移动景物放在一队列中, 从机在绘制下一帧预测采样图象时, 不再绘制这一队列中的任何景物. 而在稍后的图象 warping 过渡中, 我们采用从后一帧预测采样图象向当前帧采样图象 warping 方法来生成过渡画面的背景, 在此基础上, 再用硬件 Z-buffer 算法直接将移动景物队列中所含景物绘制上去, 即得最终画面. 由于过渡画面中仍含有深度信息, 故最终画面的上遮挡关系是完全正确的.

3.3 恒定帧频显示

在虚拟环境系统中, 各个视点及视线方向处的场景复杂度存在着很大差异, 因而, 生成各帧画面所需的时间亦相差很大. 这样, 在漫游过程中, 画面的刷新频率将呈跳跃性变化, 严重影响视觉效果, 并存在着显示帧频不稳定现象.

与基于几何的虚拟环境系统不同, 除了场景复杂度的变化外, 在考虑恒定显示帧频时, 我们还必须考虑父、子进程的相互等待等问题. 为此, 我们采用了如图 5 所示的恒定帧频显示算法.

4 执行结果

我们采用 MultiGen 软件来构造场景, 并采用 Performer 软件开发平台来生成预测采样深度图象, 然后用图象 warping 技术生成过渡画面, 在本实验室的 SGI Octane 和 O2 图形工作站上实现了本系统. 我们分别测试了以 O2 和 Octane 互为主机和从机两种分布环境下的系统性能, 并将它们与基于几何的绘制模式进行了比较(见表 1, 图象分辨率均为 512×512). 图版 I 给出了测试的 4 个场景.

从表 1 可看出, 由于 O2 工作站的计算能力较差, 将它作为主机, 系统在作复杂场景的图象

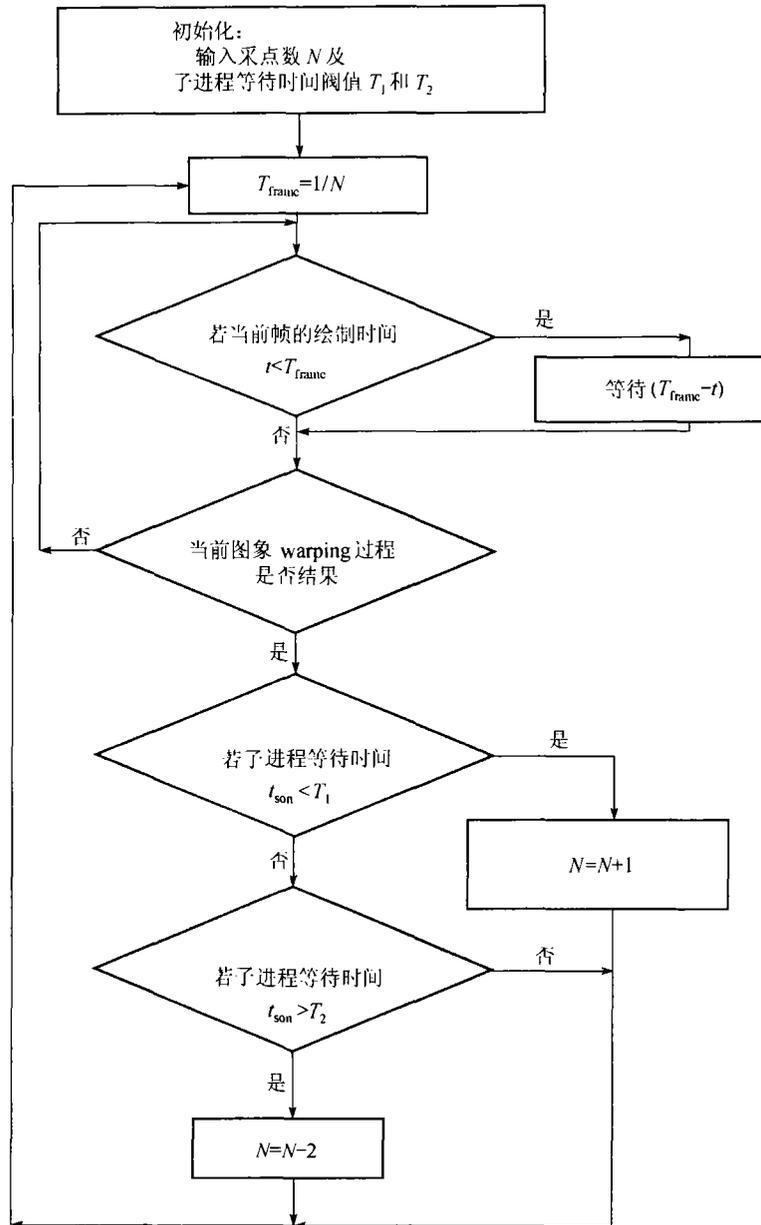


图 5 恒定帧频显示算法流程

warping 时,仍无法达到实时. 而将它作为从机时,对高度复杂的场景,它可能难以以二帧以上的帧频生成预测画面. 容易发现,本系统与场景的复杂性的依赖程度较少,因而,对高度复杂的场景本系统仍具有很高的显示帧频.

表 1 不同绘制模式的性能比较

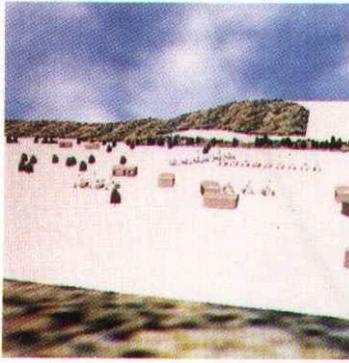
绘制方式	场景 1		场景 2		场景 3		场景 4	
	平均可见四 边形/个	显示帧频 /帧·s ⁻¹	平均可见四 边形/个	显示帧频 /帧·s ⁻¹	平均可见四 边形/个	显示帧频 /帧·s ⁻¹	平均可见四 边形/个	显示帧频 /帧·s ⁻¹
几何绘制模式(O2)	584 50	2.5	3 000	8.6	24 450	4.6	55 000	2.6
几何绘制模式(Octane)	584 50	5.1	3 000	18.0	24 450	10.0	55 000	3.6
O2 主机, Octane 从机	2 000 ~ 3 600	8.2	1 500	15.0	1 000 ~ 10 000	8.5	1 800 ~ 2 500	10.4
Octane 主机, O2 从机	2 000 ~ 3 600	11.6	1 500	20.0	1 000 ~ 10 000	14.8	1 800 ~ 2 500	14.3

5 结论

本文介绍了一个基于几何和图象的混合式绘制模式的桌面型虚拟现实系统,该系统既可以并行实现,亦可以分布式实现. 由于在图象 warping 过程中引入了几何信息,因而本系统能方便地实现用户对景物的直接交互操作,从而克服了传统基于图象绘制技术的虚拟现实系统只适应于静态场景的缺陷. 更为重要的是,本系统由于采用基于几何和图象的混合式绘制模式,因而系统对场景复杂性的关联程度得到较大的降低,从而极大提高了系统的实时响应能力. 实验结果表明,本系统能在中档图形工作站上实现实时的交互操作.

参 考 文 献

- 1 Chen S E, Williams L. View interpolation for image synthesis. *Computer Graphics*, 1993, 27(2): 279
- 2 McMillan L, Bishop G. Plenoptic modeling: an image-based rendering system. *Computer Graphics*, 1995, 29(4): 39
- 3 Greene N, Kass M, Miller G. Hierarchical Z-buffer visibility. *Computer Graphics*, 1993, 27(2): 231
- 4 Shade J, Lischinski D, Salesin D H, et al. Hierarchical image caching for accelerated walkthroughs of complex environments. *Computer Graphics*, 1996, 30(3): 75
- 5 Sillion F X, Drettakis G, Bodelet B. Efficient imposter manipulation for real-time visualization of urban scenery. *Computer Graphics Forum*, 1997, 16(3): 207
- 6 Torborg J, Kajiya J T. Talisman: commodity real-time graphics for PC. *Computer Graphics*, 1996, 30(3): 353
- 7 Carolina C, Sandin D J, Defanti T A. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. *Computer Graphics*, 1993, 27(2): 135
- 8 Chen S E. QuickTime VR: an image-based approach to virtual environment navigation. *Computer Graphics*, 1995, 29(4): 29
- 9 Raskar R, Welch G, Cutts M, et al. The office of the future: a unified approach to image-based modeling and spatially immersive displays. *Computer Graphics*, 1998, 32(3): 179
- 10 郑文庭, 鲍虎军, 彭群生. 基于几何和图象的混合式图形实时绘制算法. *自然科学进展*, 2000, 10(4): 366
- 11 Lihner V D, Defanti T A. Distributed virtual reality: supporting remote collaboration in vehicle design. *IEEE Transaction on Computer Graphics and Applications*, 1997, 17(2): 13
- 12 Pan Z G, Shi J Y, He Z J. A distributed graphics processing support environment. *Chinese Journal of Advanced Software Research*, 1995, 2(2): 161



(a)



(b)



(c)



(d)

4 个测试场景

(a) ~ (d) 分别为：场景 1 ~ 4