



虚拟现实技术及应用

主讲：纪庆革 副教授

中山大学数据科学与计算机学院
软件工程与应用研究所

Email: 1024180018@qq.com



第一讲： 虚拟现实技术概论

- ✚ VR的概念和发展
- ✚ VR系统的硬件组成
- ✚ VR系统的体系结构
- ✚ VR的研究内容
- ✚ 增强现实（AR）
- ✚ VR应用
- ✚ 虚拟现实的发展趋势



（一）VR的概念和发展

（1）虚拟现实技术的提出

1965年计算机图形学的奠基者 Ivan Sutherland发表了“ The Ultimate Display” 论文，提出了一种全新的图形显示技术。他在论文中提出使观察者直接沉浸在计算机生成的三维世界中，而不是通过窗户（计算机屏幕）来观察。

1) 观察者自然地转动头部和身体，他看到的场景就实时地发生变化。

2) 观察者能够以自然的方式直接与虚拟世界中的对象进行交互操作，触摸它们，感觉它们，并能听到虚拟世界的三维空间声音。



Ivan Edward Sutherland (born May 16, 1938)[1] is an American computer scientist and Internet pioneer, widely regarded as the "father of computer graphics".[2] His early work in computer graphics as well as his teaching with David C. Evans in that subject at the University of Utah in the 1970s was pioneering in the field. Sutherland, Evans, and their students from that era invented several foundations of modern computer graphics. He received the Turing Award from the Association for Computing Machinery in 1988 for the invention of Sketchpad, an early predecessor to the sort of graphical user interface that has become ubiquitous in personal computers. He is a member of the National Academy of Engineering, as well as the National Academy of Sciences among many other major awards. In 2012 he was awarded the Kyoto Prize in Advanced Technology for "pioneering achievements in the development of computer graphics and interactive interfaces".[3]



◆ 名言 知识是一种十分珍贵的东西，你要把它给予别人，才能获得它。



Ji课后练习：重写伊凡·苏泽兰简介。



（2）虚拟现实的狭义解释

虚拟现实对不同的人有不同的含义。比较狭义的解释是：虚拟现实是由计算机生成的、可交互的仿真环境。该虚拟环境具有以下三个特点：

- 1) 该环境将向介入者（人）提供视觉、听觉、触觉等多种感官刺激。
- 2) 该环境应给人以一种身临其境的沉浸感。
- 3) 人能以自然的方式与该环境中的一些对象进行交互操作，既不使用键盘鼠标等常规输入设备，而强调使用手势（数据手套）、体势（数据衣服）和自然语言等自然方式的交互操作。

(3) 一个典型的虚拟现实系统



(4) 虚拟现实的广义解释

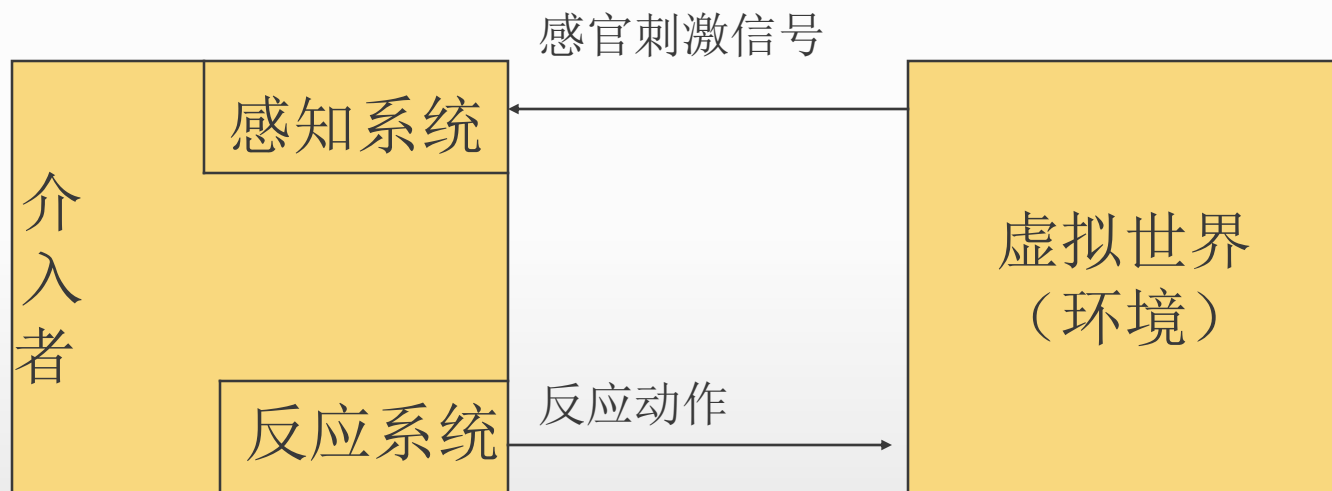
虚拟现实是由**计算机生成的**给人以**沉浸感**的虚拟环境。





- ◆ 中长期科学和技术发展.....
- ◆ 五、前沿技术
- ◆ 2.信息技术
- ◆ (8)虚拟现实技术
- ◆ 重点研究电子学、心理学、控制学、计算机图形学、数据库设计、实时分布系统和多媒体技术等多学科融合的技术，研究医学、娱乐、艺术与教育、军事及工业制造管理等多个相关领域的虚拟现实技术和系统。

(5) 虚拟现实的概念模型





(6) 虚拟环境系统提供的各种感官刺激

- ◆ 视觉
- ◆ 听觉
- ◆ 嗅觉
- ◆ 味觉
- ◆ 触觉（触觉，力觉）
- ◆ 身体感觉



(7) 虚拟环境中用户的动作检测和设备

用户反应动作	检测设备
头部运动	头部跟踪装置
肢体或躯体运动	跟踪器，力反馈装置，空间球
手指运动	数据手套，按钮装置，操纵杆
眼球运动	眼球跟踪器
语言	语音识别装置
受力	带力传感器的力反馈装置



(8) 虚拟现实的发展历程

- ◆ 1965年 Ivan Sutherland博士提出 “The Ultimate Display” 概念，1968年他研制成功头盔式显示器（HMD）
- ◆ 70年代以来图形学技术的进步，特别是90年代，可以实时显示复杂场景
- ◆ 军方飞行模拟器的研制和发展对VR的推动
- ◆ 1984年美国NASA的AMES中心研制成功VIVD系统，以及1986年的VIEW项目
- ◆ 1992年美国NSF召开VR研讨会
- ◆ 其他国际会议（IEEE VR；ACM VRST等），VR专著以及各种研究成果



（二）VR系统的典型设备和环境

✚ 显示和观察设备

- 沉浸式现实设备
- 非沉浸式显示设备

✚ 交互设备

- 数据手套（输入手的各种姿势）
- 三维鼠标
- 游戏杆

✚ 传感设备

- 数据手套
- 数据衣
- 位置跟踪装置

✚ 三维立体声系统

✚ 三维数据获取设备



(1) 显示和观察设备

产生沉浸感的方法和装置

1. 头盔式显示器
2. 快速响应的工作平台
3. 大屏幕投影
 - 洞穴状的投影屏幕 (CAVE)
 - 圆柱型的投影屏幕
 - 由矩形块拼接构成的投影屏幕
4. 全景图
5. 可支持多用户协作的投影屏幕
6. 具有三维空间感的声音

头盔式显示器的特点

用户带着头盔；
双眼看到不同的图象；
跟踪器跟踪头的方向。



图 具有看穿功能 (see-through) 的头盔显示器





快速响应的工作平台





洞穴状的投影屏幕（CAVE）



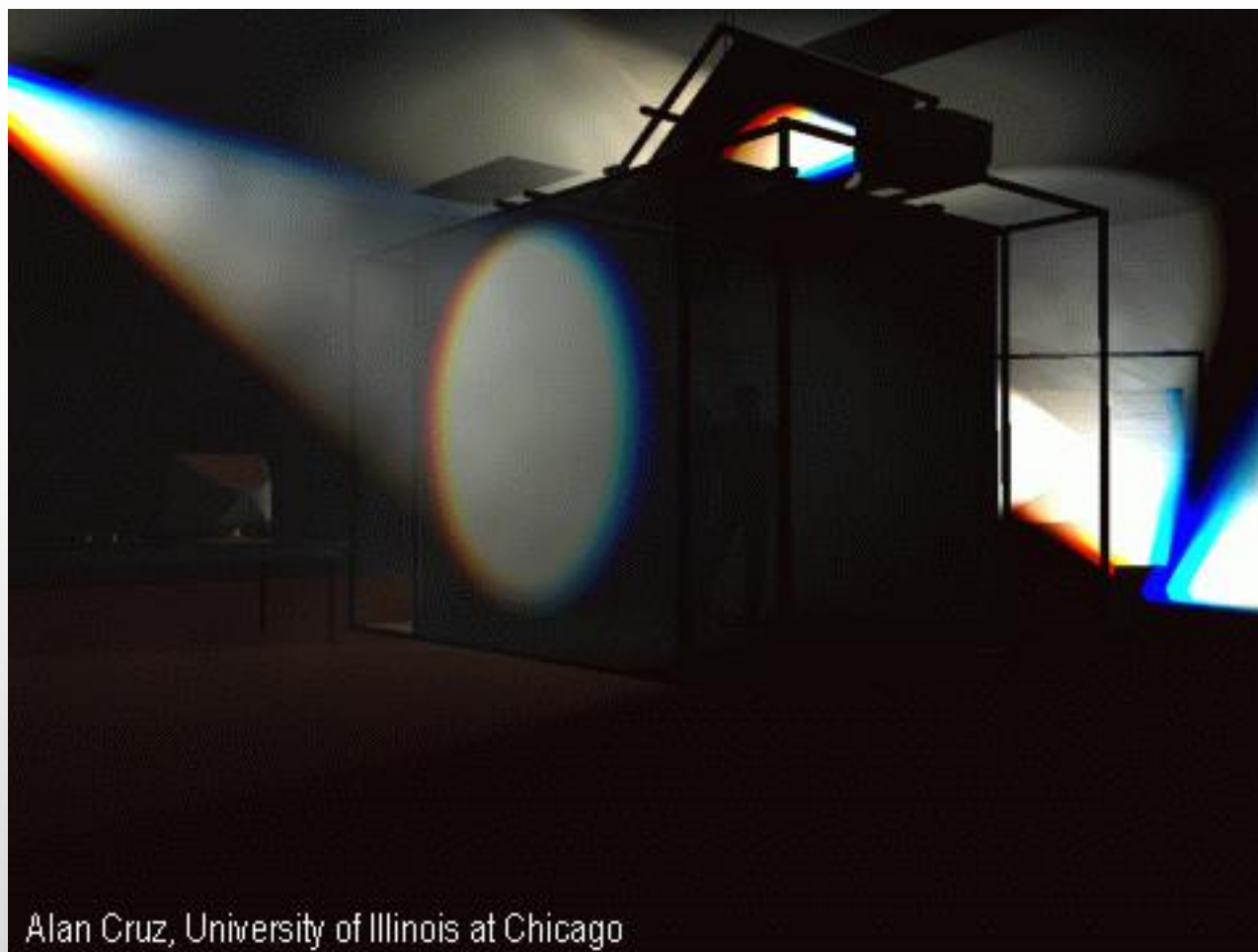


安装在浙江大学的CAVE



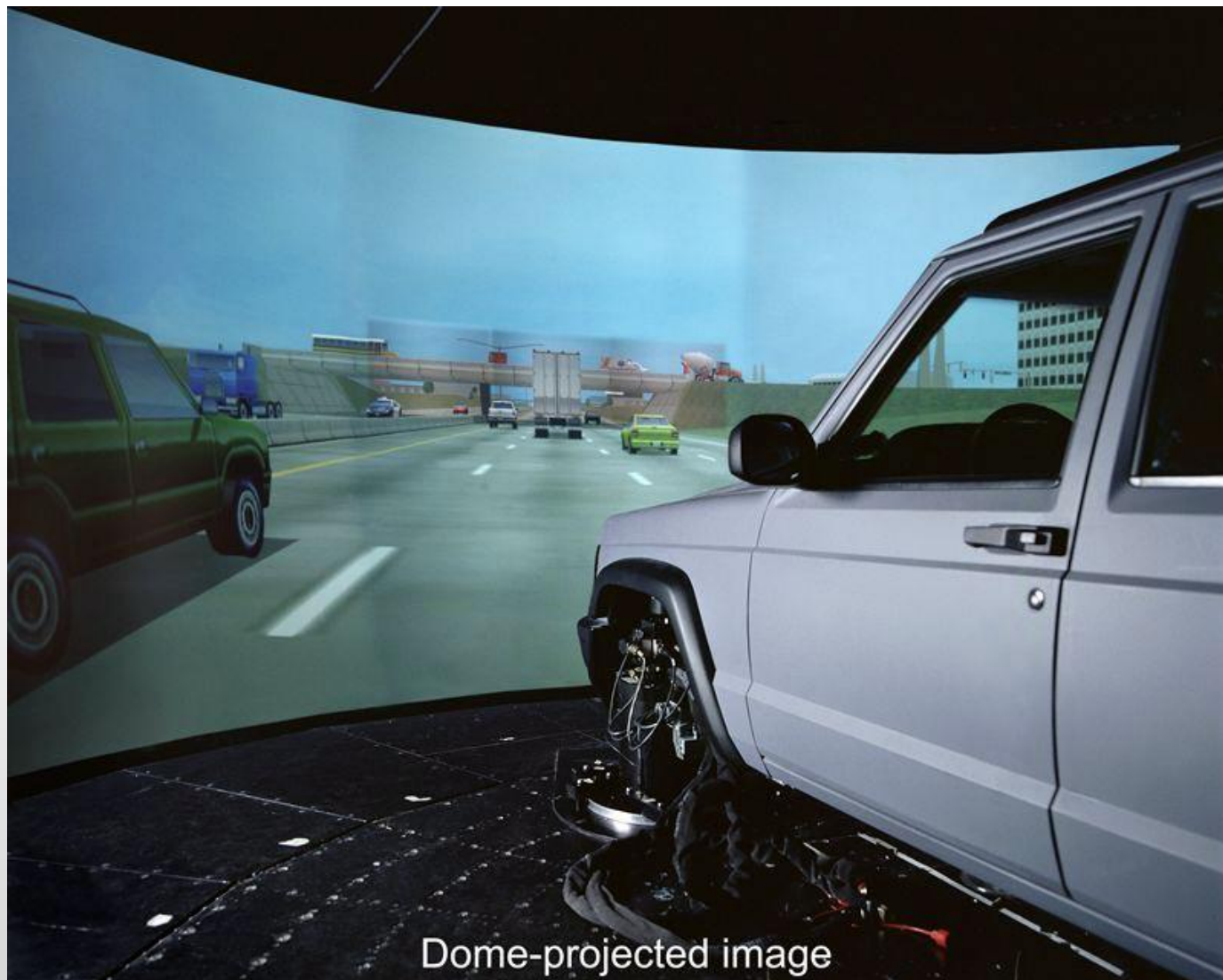


CAVE 示意图



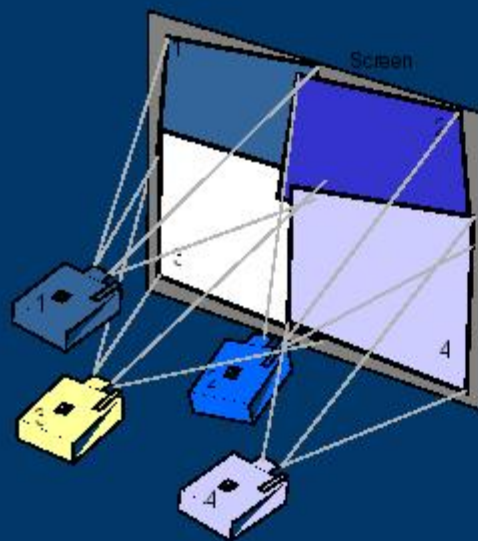


圆柱型的投影屏幕



由矩形块拼接构成的投影屏幕

Planar Tiled Display



全景图

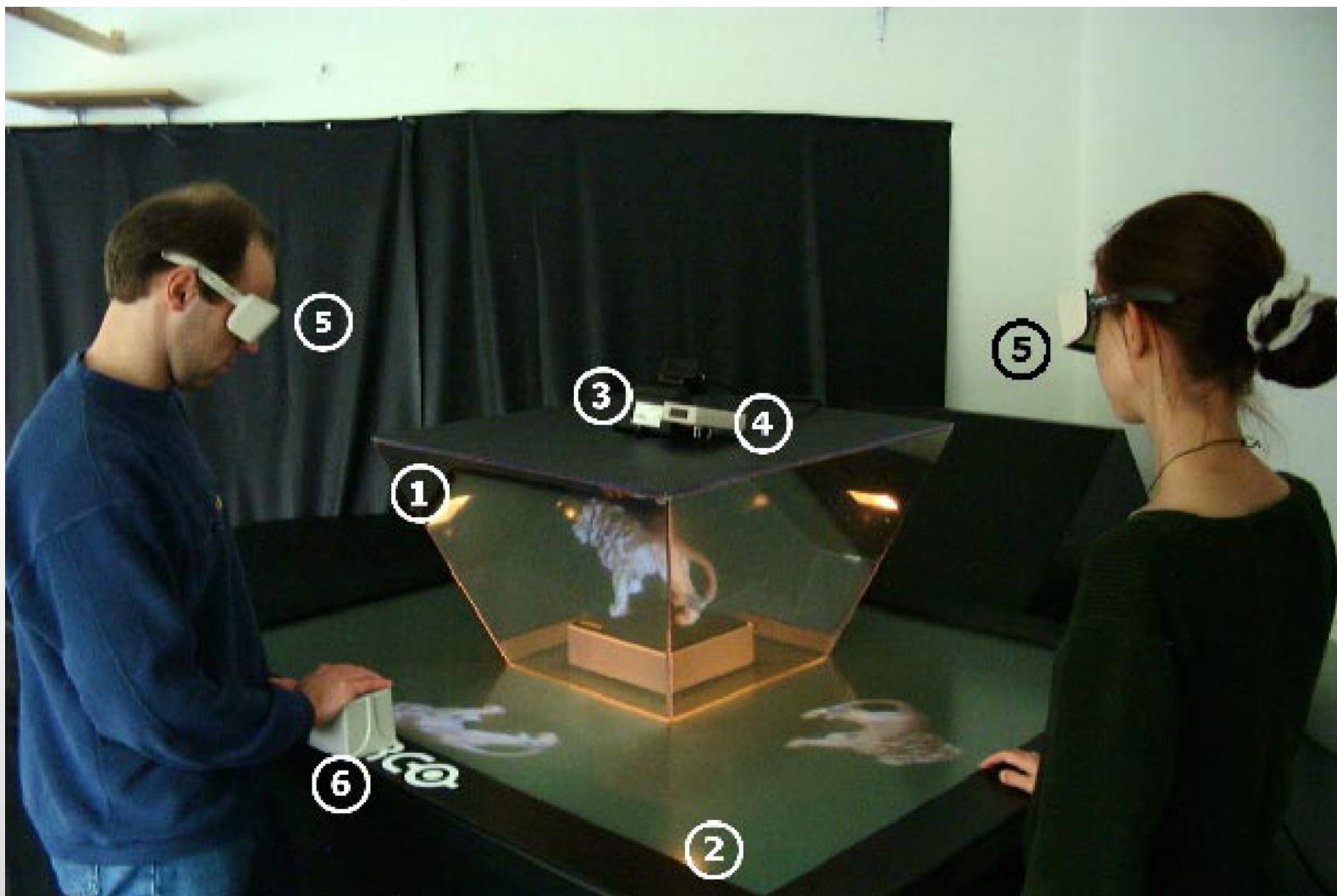
Unregistered HyperCam



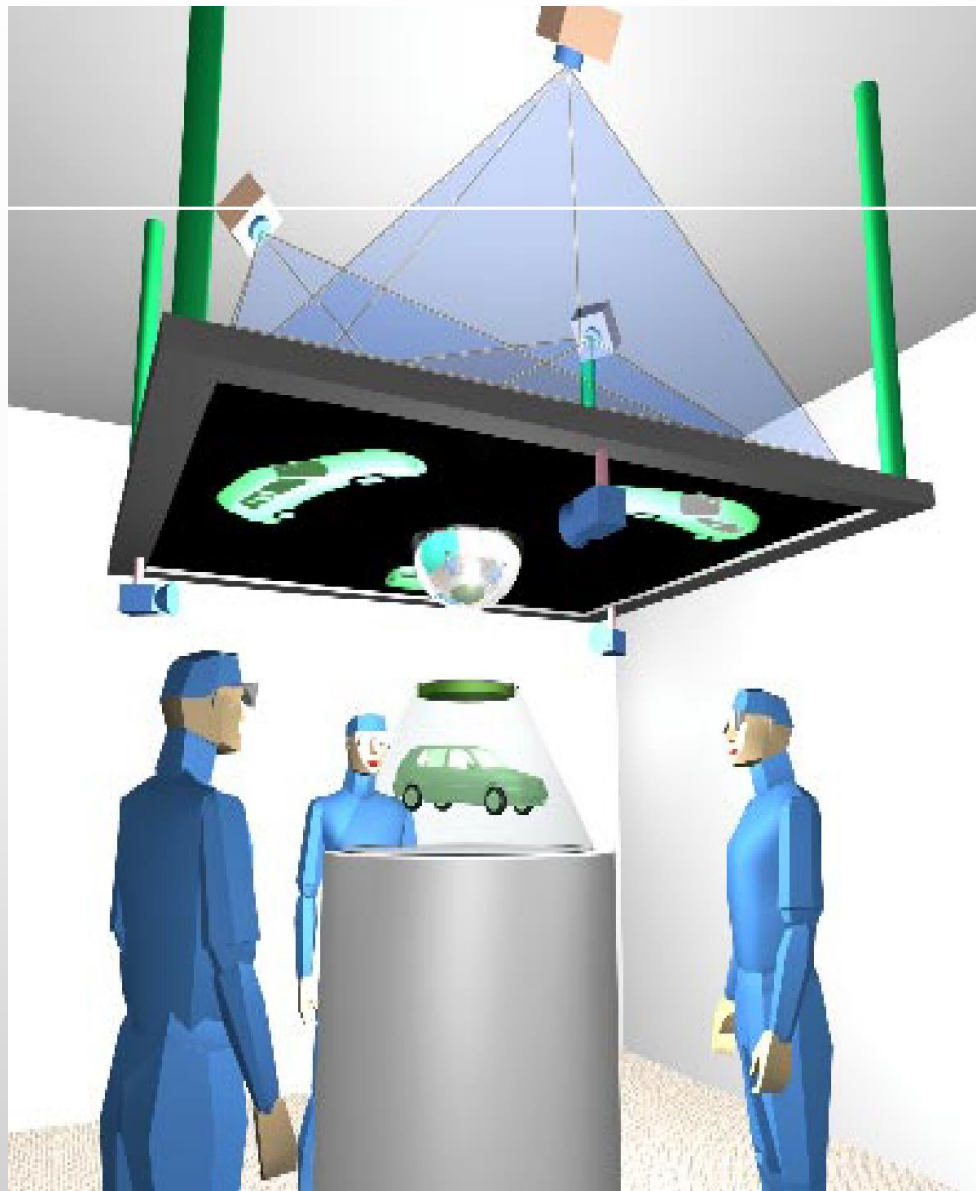
北京全景数字媒体



可支持多用户协作的投影屏幕

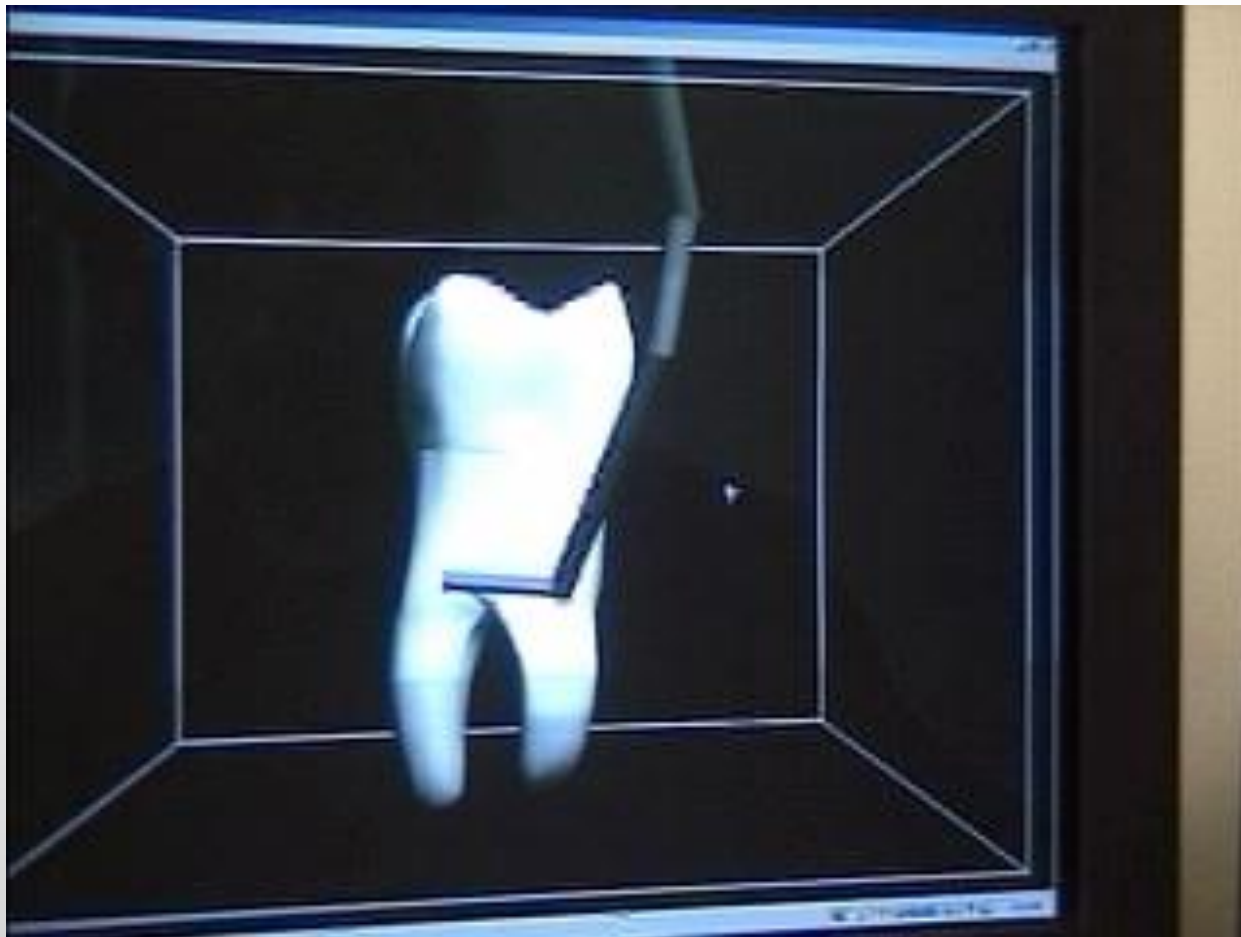


可支持多用户协作的投影屏幕的改进



(2) 交互设备

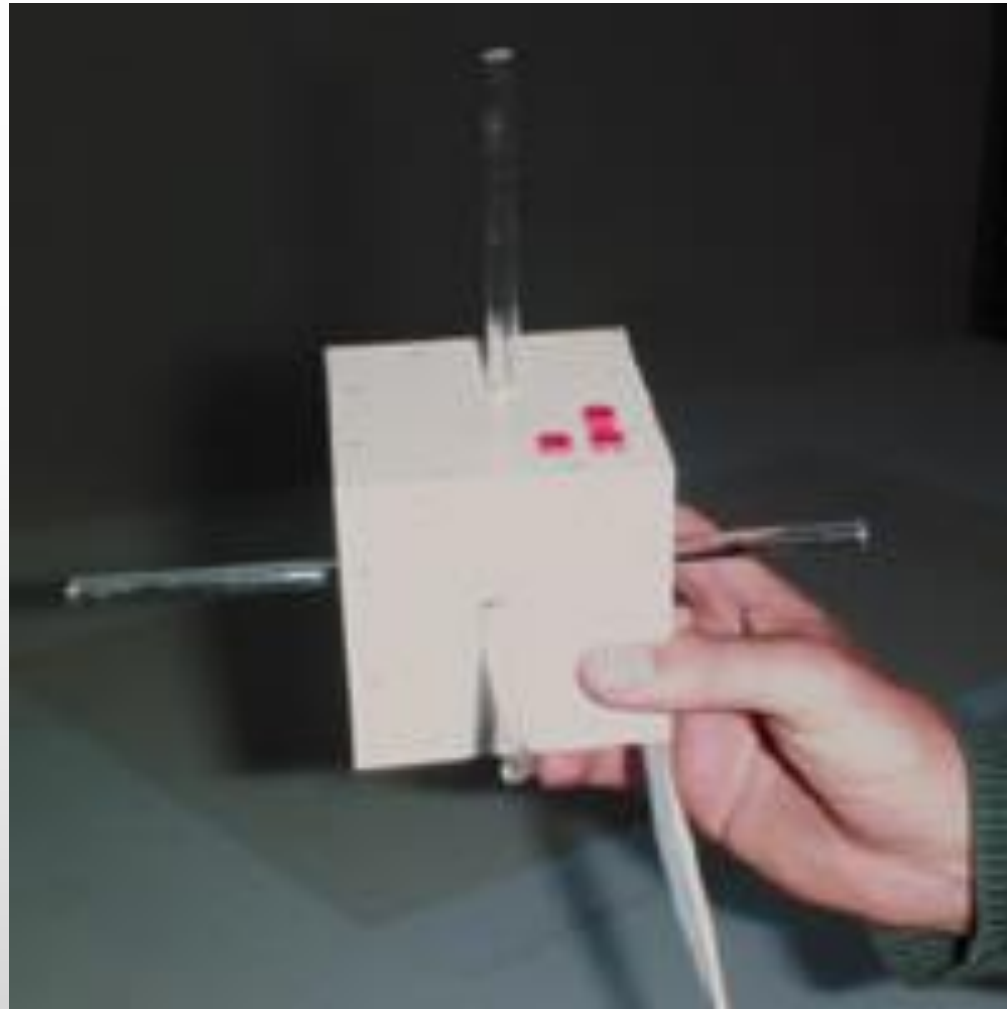
力反馈设备：



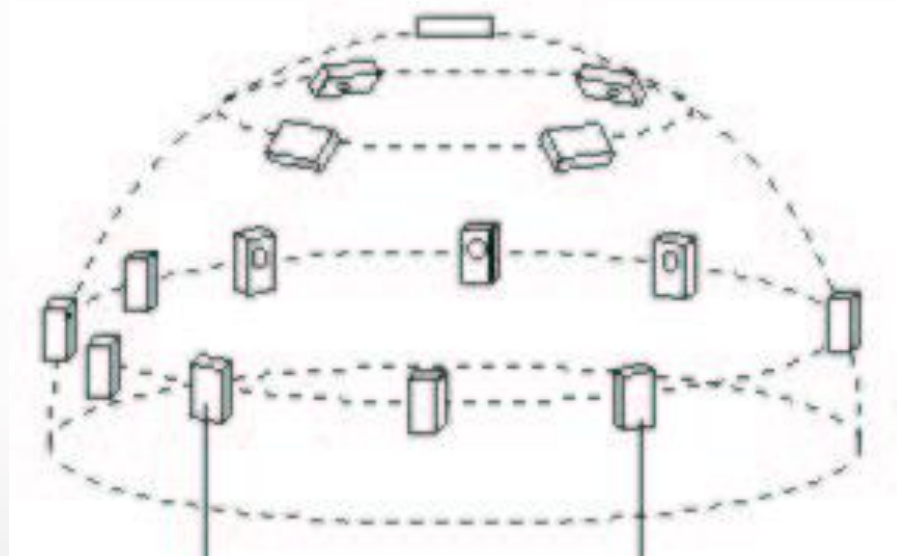
数据手套(cyberglove):



三维鼠标



(3) 三维声音系统



声音在VR中的作用：

- 1) 增强空间信息，尤其是当空间超出了视域范围。
- 2) 数据驱动的声音能传递对象的属性信息。
- 3) 声音是用户和虚拟环境的另一种交互方法。



(4) 三维数据获取设备

静态数据的获取：

三维扫描仪

动态数据的获取：

运动捕捉设备

用多摄像机实时重构三维数据

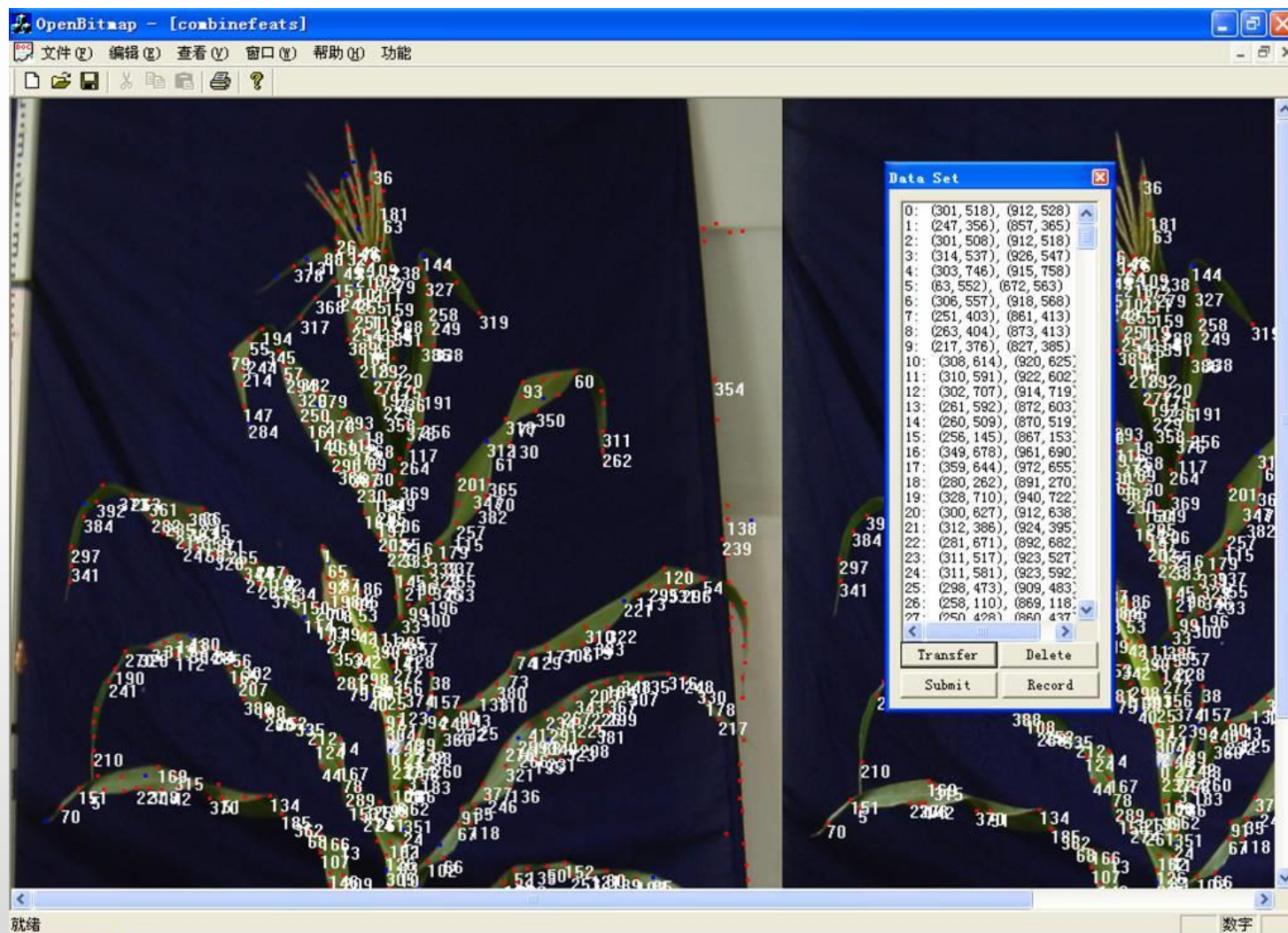
三维扫描仪 (来自cyberware 公司)



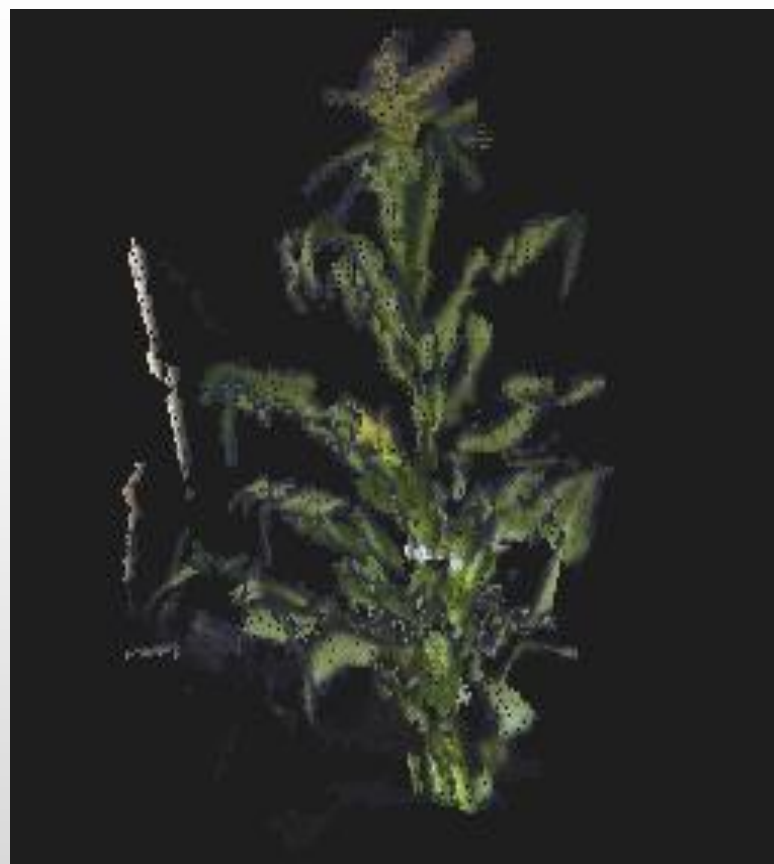


有多个摄像头的三维实时重构



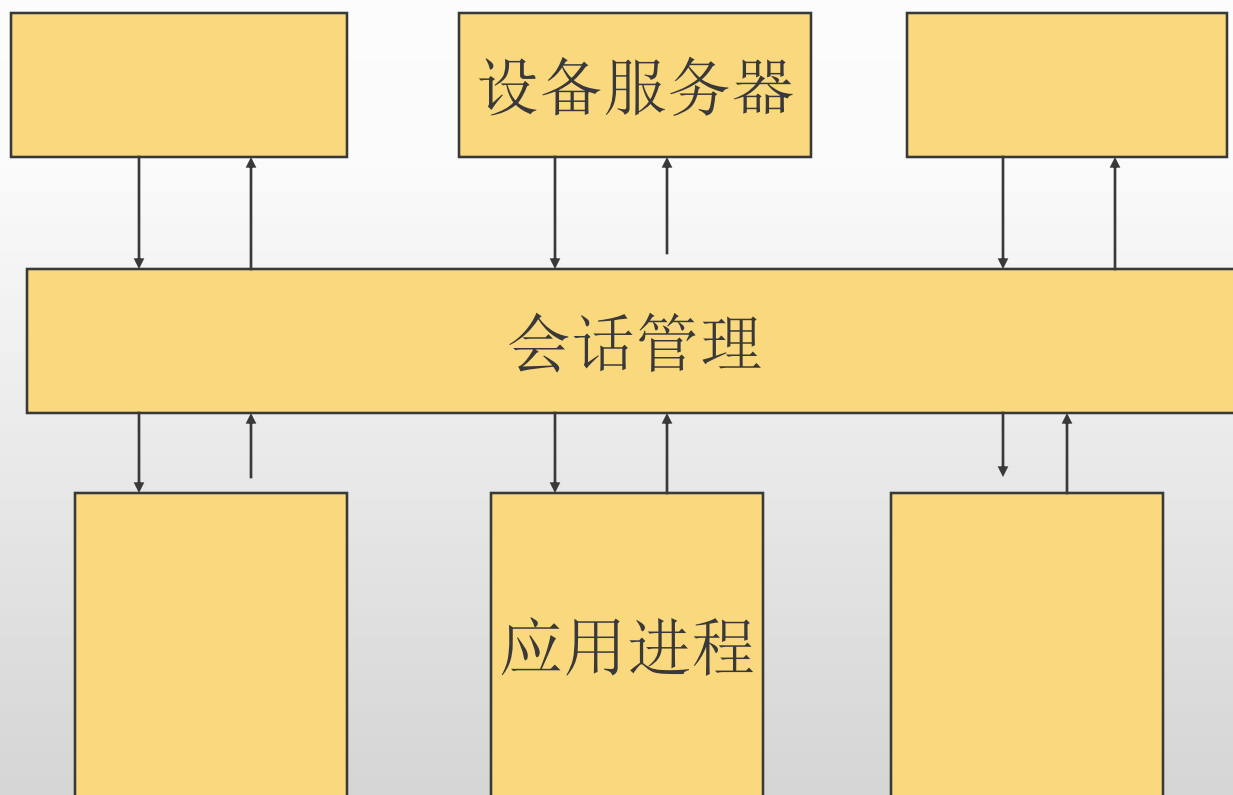


基于图片的三维重建



(三) VR系统的体系结构

(1) 非分布式VR系统体系结构





(2) 分布式虚拟现实系统（DVR）的结构

1) 数据模型

– 集中式结构

- *结构简单
- *实现比较容易
- *对网络通信带宽有较高的要求
- *中心服务器往往会成为整个系统的瓶颈

– 复制式结构

- *所需网络带宽较小
- *交互式响应效果好
- *比集中式结构复杂
- *维护多个备份的信息或状态一致性比较困难

2) 基于分布仿真 (DS) 模型的VR系统软件结构

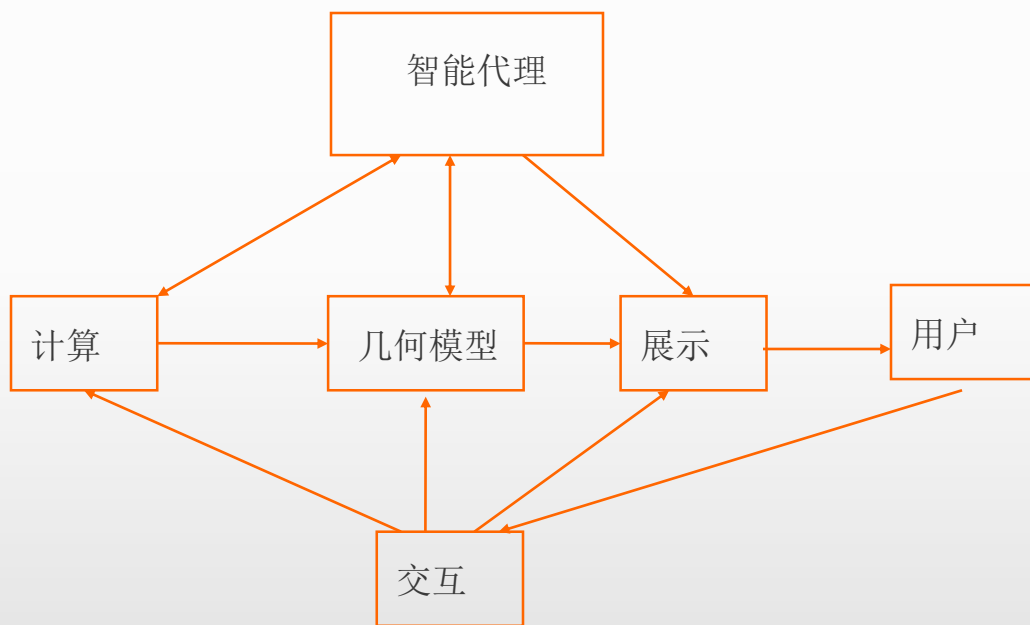


图 基于DS模型的VR系统软件结构

3) DVR系统的通用参考结构模型

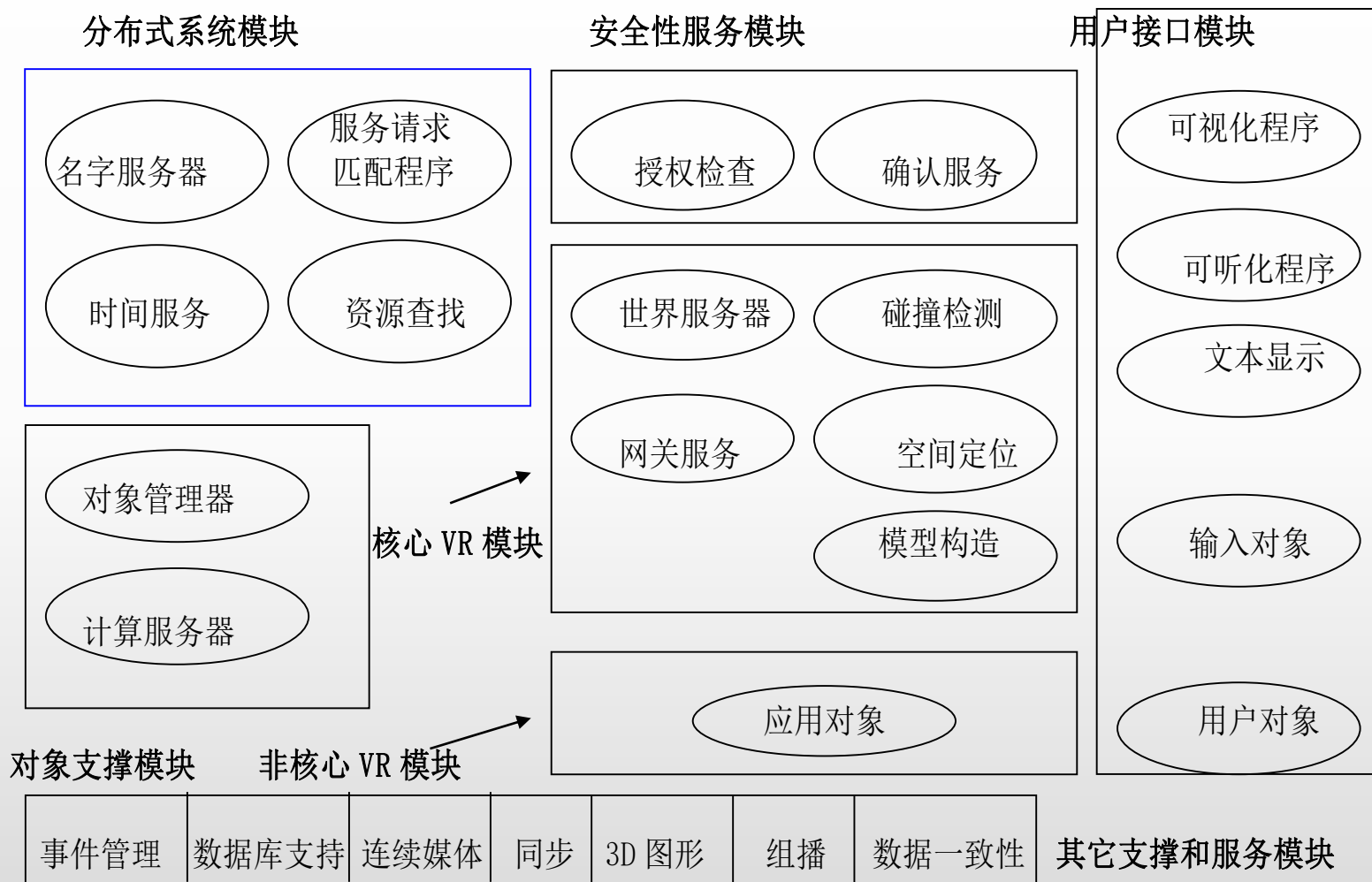
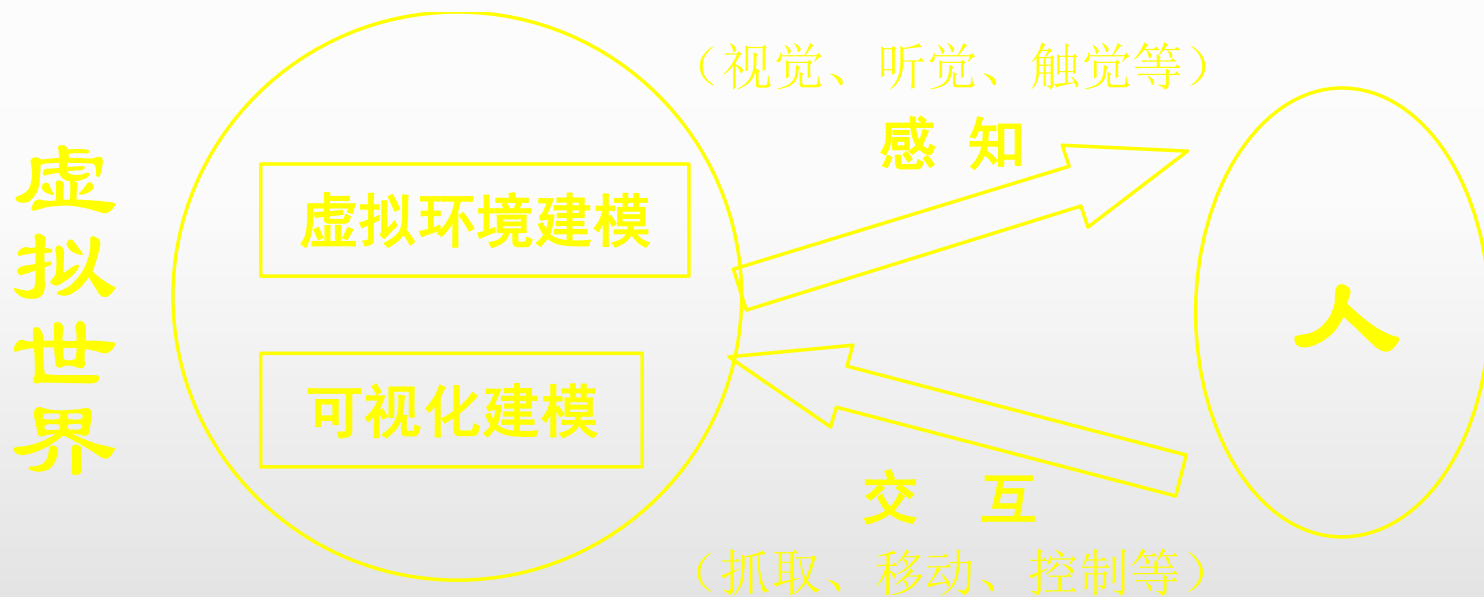


图3 DVE 的参考结构模型



（四）VR的研究内容





(1) 研究内容的多种分类

◆ 技术/应用:层面

- * 虚拟现实技术本身
- * 虚拟现实应用技术

◆ 软硬件层面

- * 虚拟现实硬件技术
- * 虚拟现实软件技术

◆ 技术层面

- * 建模（场景（基于：几何/图象），听觉，行为等）
- * 绘制（图形，声音，力觉）
- * 交互（人-机交互；人-人交互）



(2) 分布式虚拟现实

— 分布式虚拟现实(**Distributed Virtual Reality, DVR**)

在DVR系统中，位于不同物理位置的多个用户或多个虚拟环境通过网络相联结，进行信息共享和交互

— **DVR系统或工具的开发涉及多个领域**

- | | |
|------------|----------|
| * 虚拟现实 | * 分布式系统 |
| * 实时图形绘制技术 | * 异步系统设计 |
| * 数据库 | * 用户界面设计 |



DVR的实例





(3) 实时绘制技术

- 场景简化
- 快速消隐
- 纹理化对象
- 限时绘制

（五）增强现实

增强现实允许用户看到真实世界，并且把虚拟对象叠加在真实世界之上。





(1) 增强现实系统的定义

- ◆ 真实世界和虚拟世界融为一体的系统
- ◆ 具有实时人-机交互功能
- ◆ 真实世界和虚拟世界是在三维空间上整合的。

(2) 增强现实的关键技术

- ◆ 聚焦和对比度
- ◆ 系统的可移动性
- ◆ 整合技术
- ◆ 实时高精度传感器



(六) 虚拟现实的应用

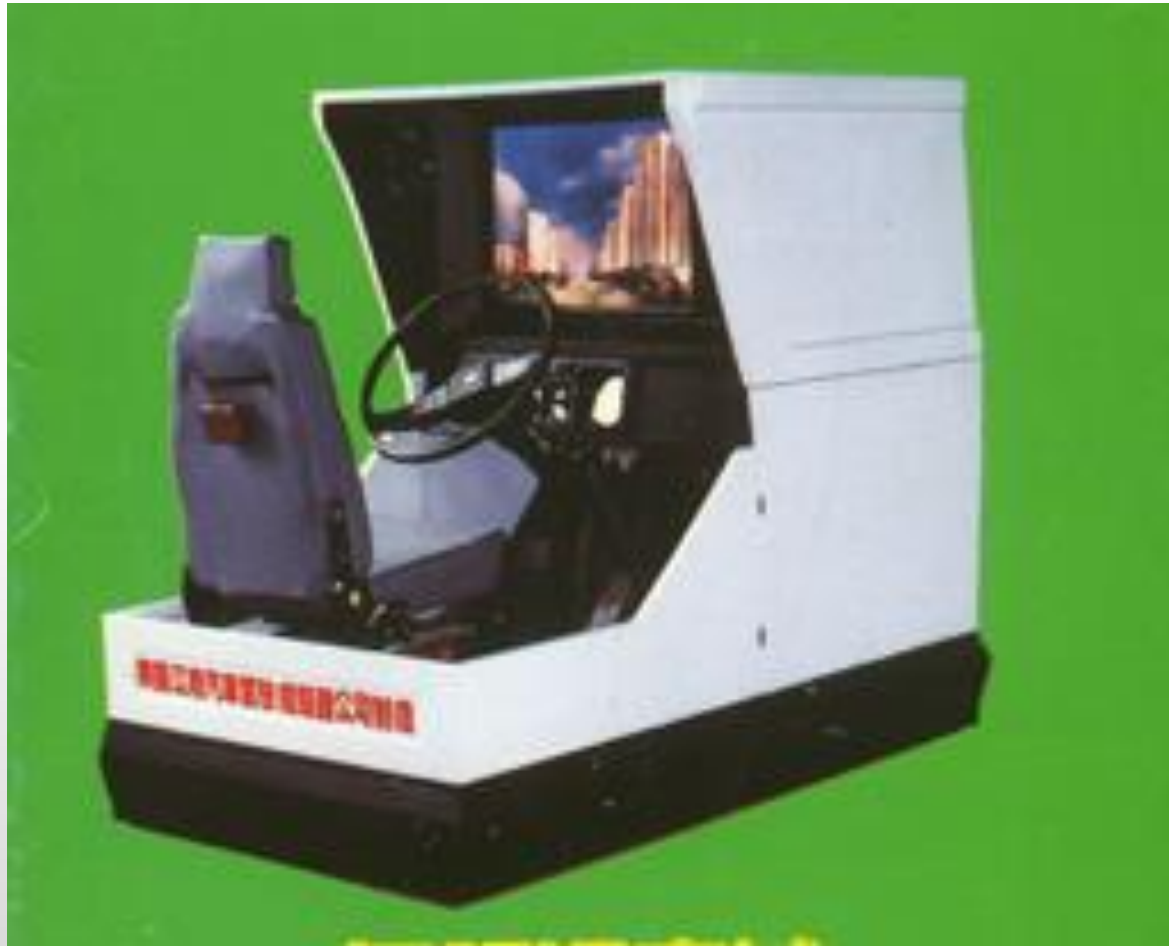
* 应用（一）

- 虚拟战场
- 核爆炸、战略导弹的虚拟杀伤
- 虚拟飞行/驾驶
- 虚拟建筑漫游
- 虚拟风洞实验
- 虚拟设计、虚拟制造

战场仿真



虚拟驾驶模拟



虚拟建筑漫游

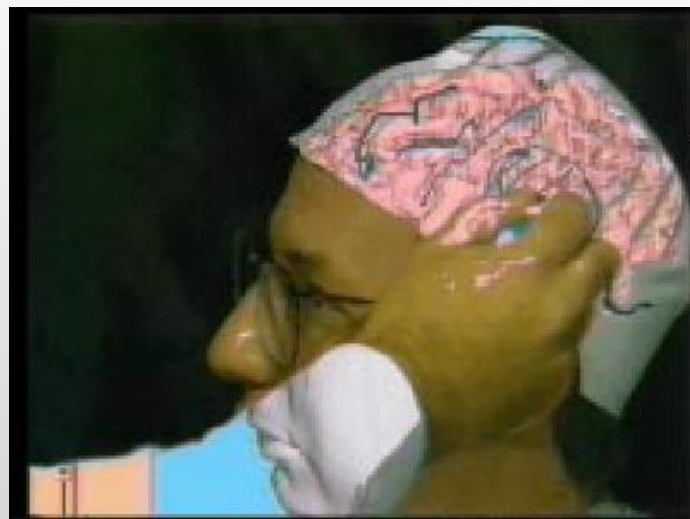
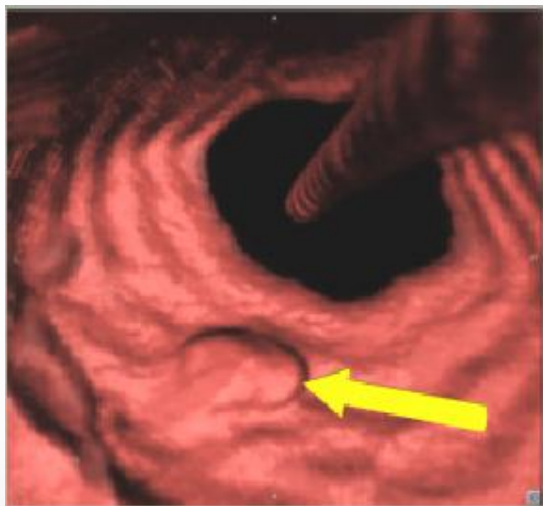




* 应用（二）

- 虚拟课堂
- 虚拟手术和远程手术
- 城市规划和建筑设计的虚拟演示
- 虚拟商店、虚拟旅游、虚拟剧场
- 虚拟体育仿真
- 虚拟文化遗产保护（虚拟博物馆）

虚拟手术



虚拟故宫漫游

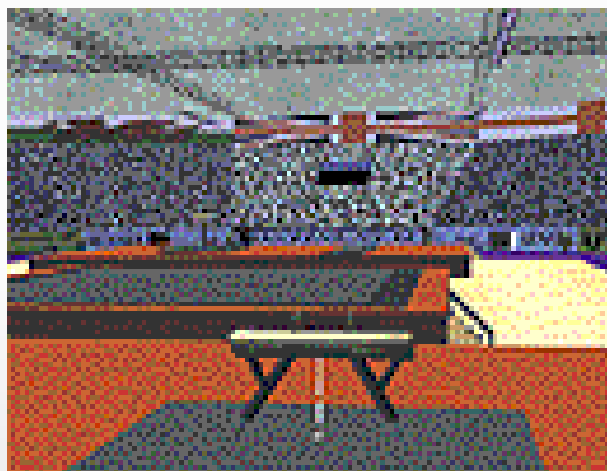


虚拟博物馆



虚拟体操训练系统

(Sheffield Hallam University)

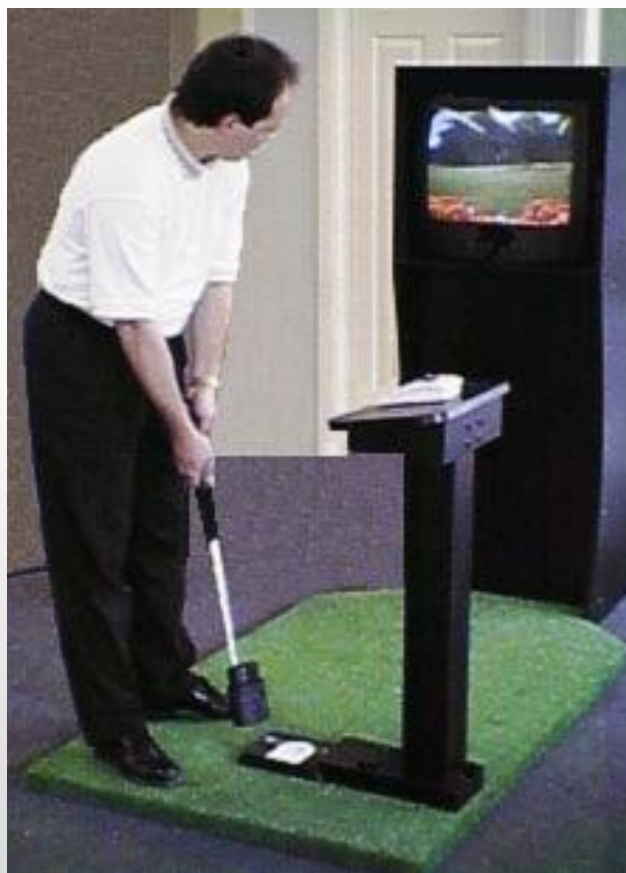


虚拟滑雪训练系统

(Reebok Sports Club in New York City)

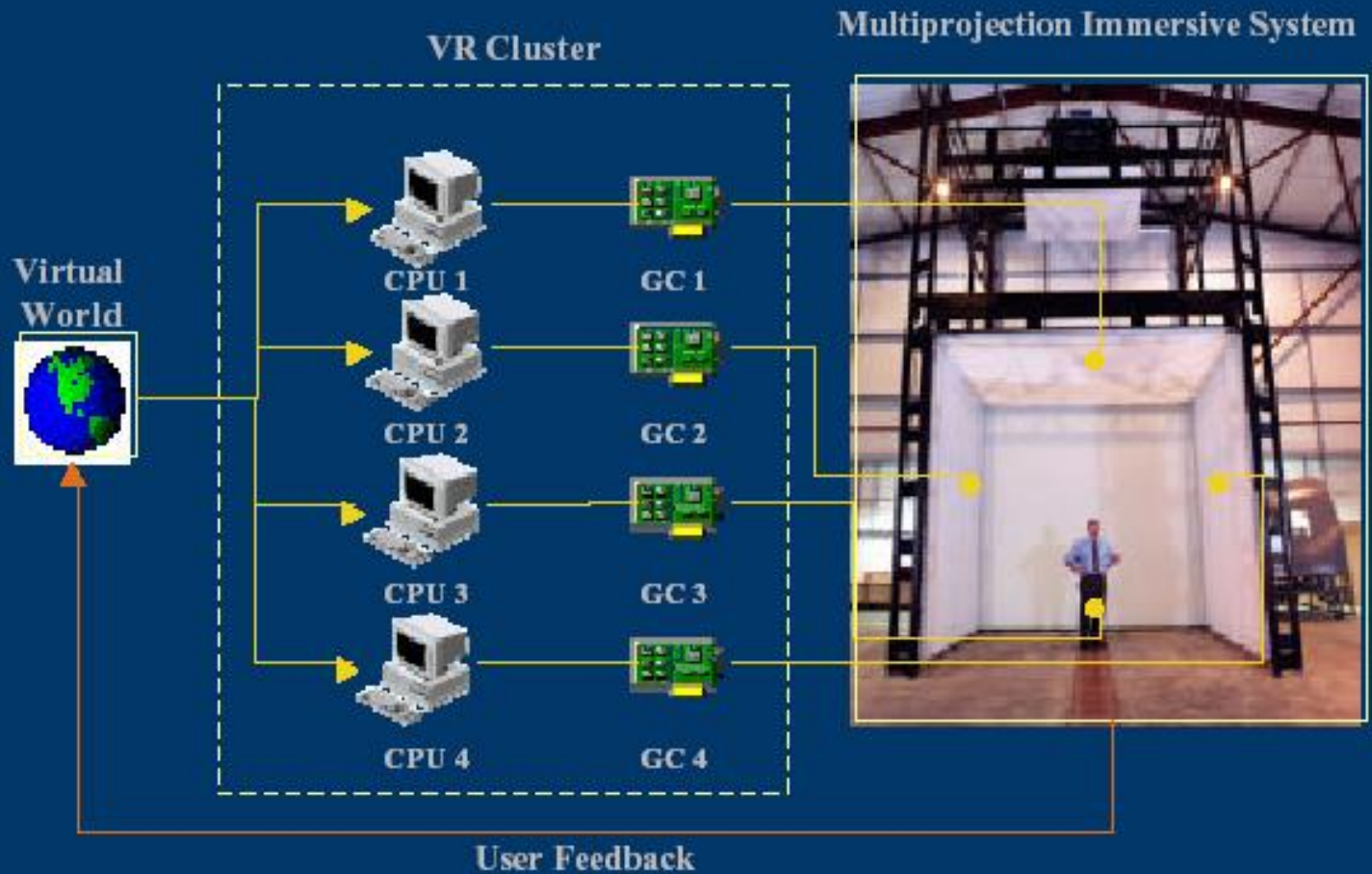


其他例子



PC CLUSTER

Putting all together





（七）虚拟现实的发展趋势

（1）虚拟现实硬件的发展趋势

- 1) 正在大力发展基于 PC 的低价格的虚拟现实系统
- 2) 基于多个视频输入的系统受到重视。例如：由多个摄像头组成的跟踪器, 由多个摄像头组成的三维重构系统等。



(2) 虚拟现实软件的发展

发展特点:

- 1) 非常规的输入设备在主流的操作系统中不被支持。
- 2) 管理不同的输入的工具包正在出现。

软件资源:

VRJuggler

- 1) 开放源代码, 其功能类似CAVElib。
- 2) 具有输入和显示功能。
- 3) 输入层不支持手势识别。
- 4) 设计者有雄心打算将其发展成一个操作系统。

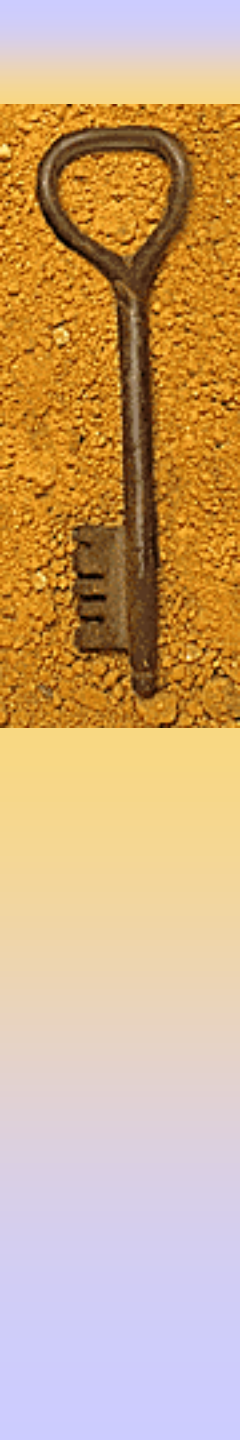


(3) 虚拟现实应用的发展

- ◆ 1) 网络化==》网络VR
- ◆ 2) 单投影应用==》多投影应用
- ◆ 3) 智能化==》智能虚拟环境
- ◆ 4) 虚拟现实==》增强现实，混合现实
- ◆ 5) 应用向各个行业扩展：
 - * 与商业结合==》虚拟商城
 - * 与体育结合==》虚拟体育仿真
 - ○ ○ ○


最终形成网络化的虚拟社会。

第二讲 实时图形绘制技术

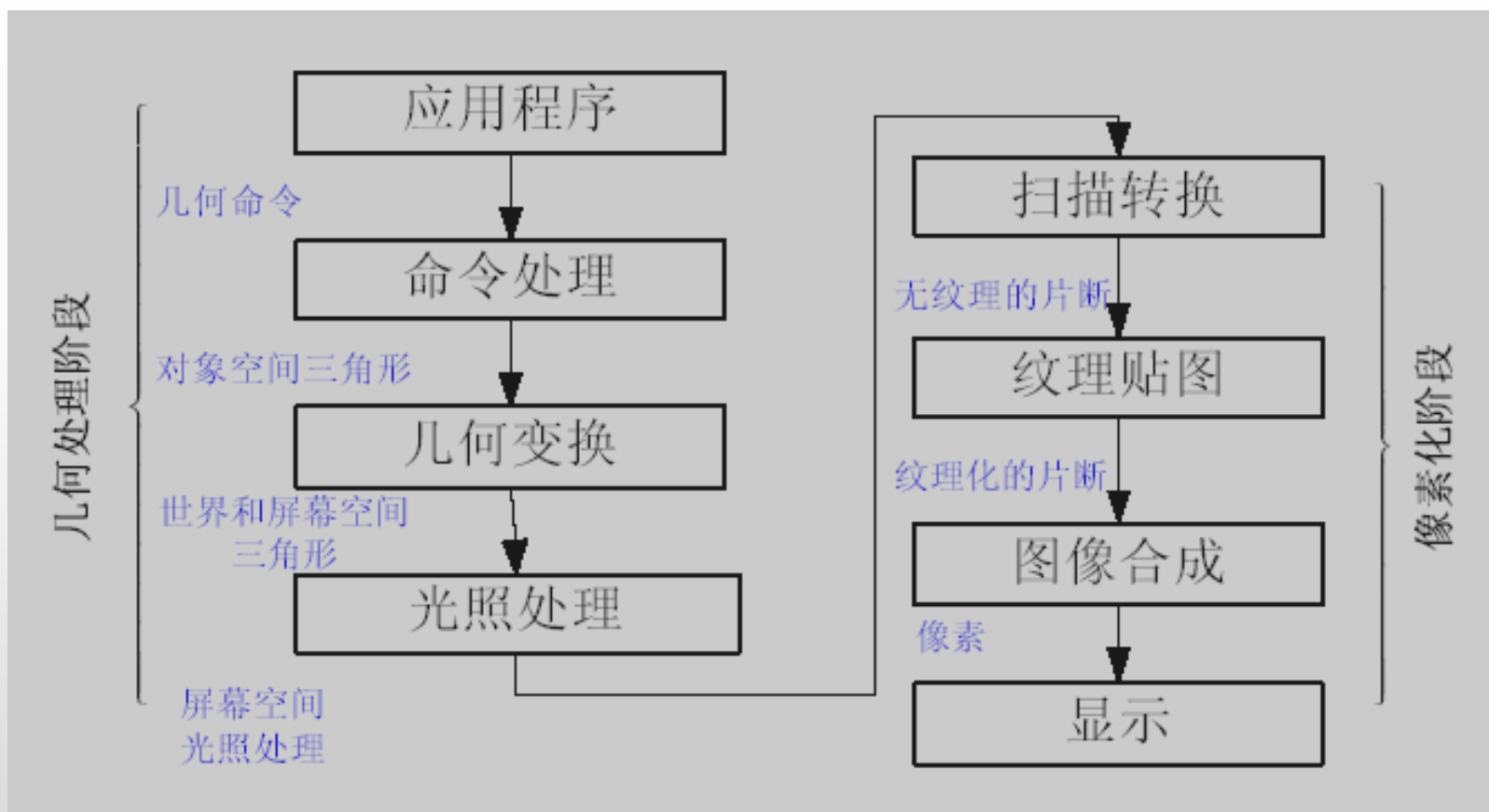
- 
- 图形绘制流水线
 - 网格简化
 - 快速消隐
 - 纹理映射
 - IBR&IBM

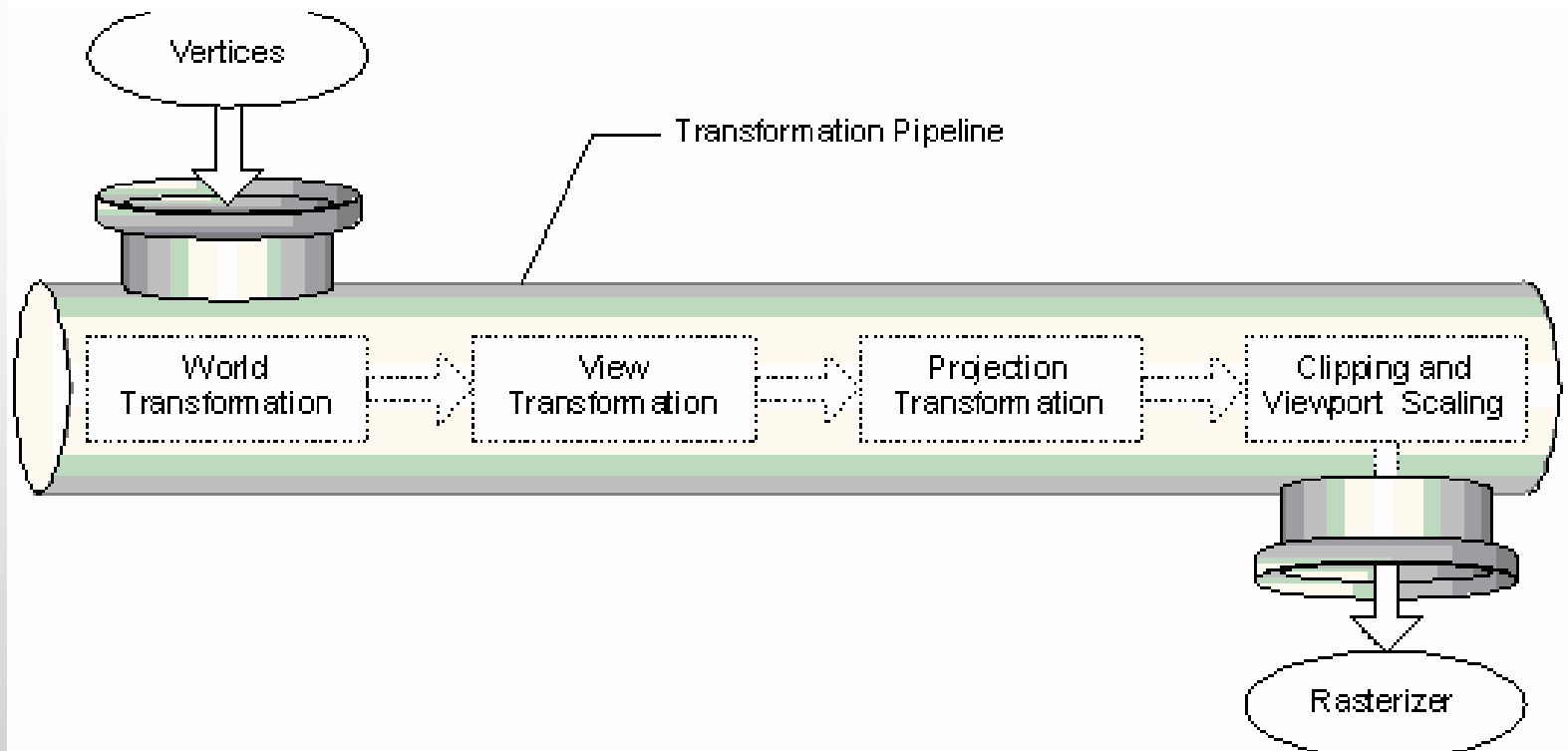
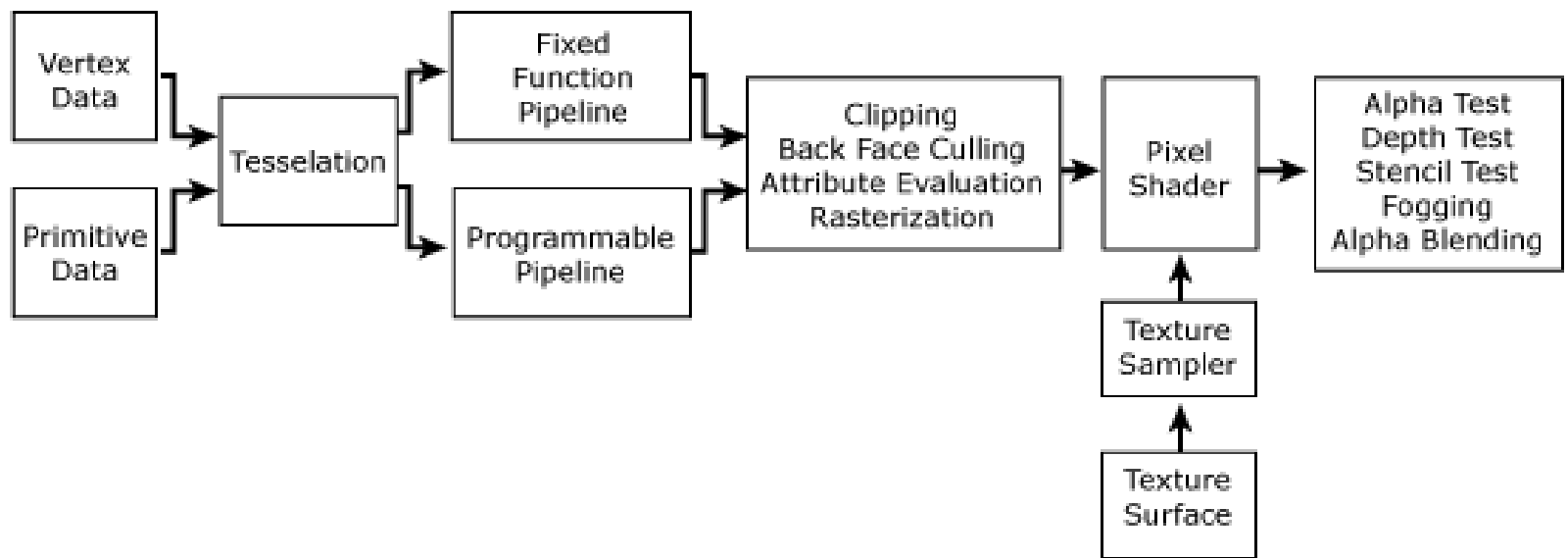


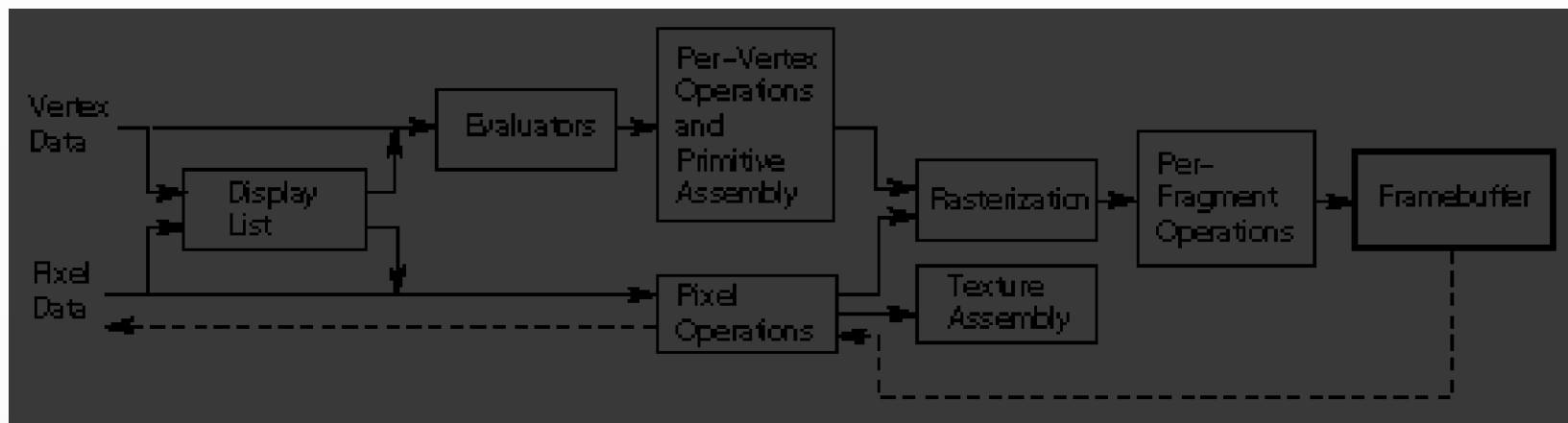
真实感图形生成的步骤

- 
- ◆ 场景造型
 - ◆ 取景变换、透视投影
 - ◆ 视域裁剪、消除隐藏面
 - ◆ 光亮度计算

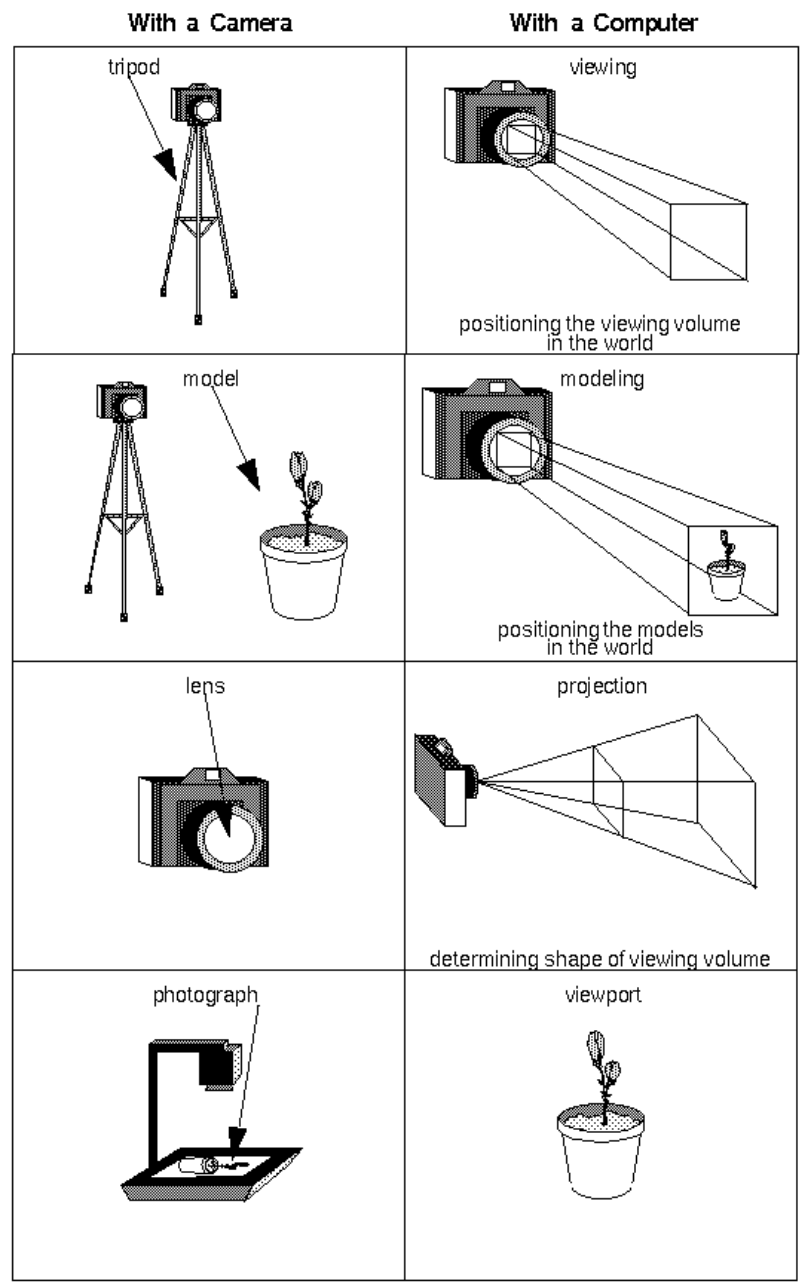
典型三维图形系统的绘制流水线



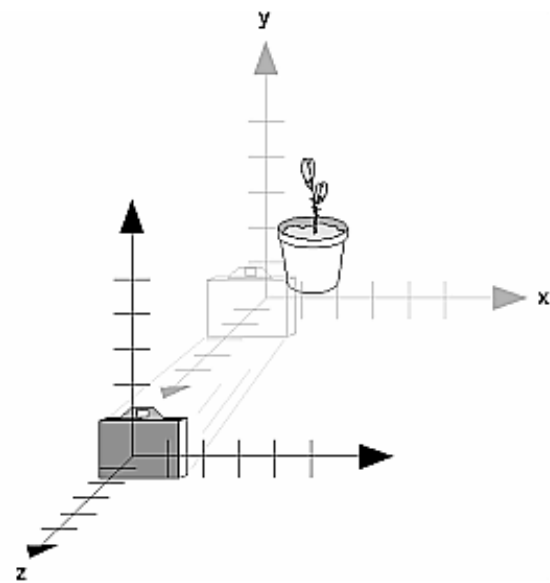
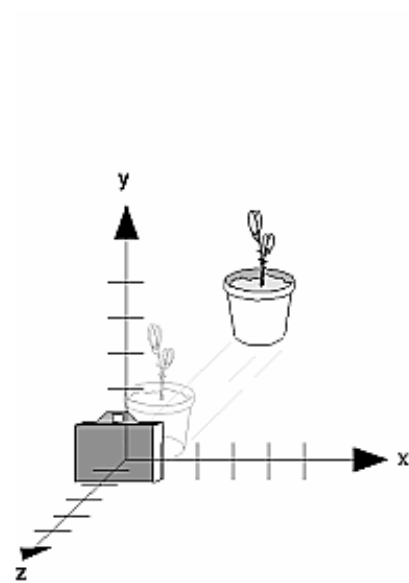
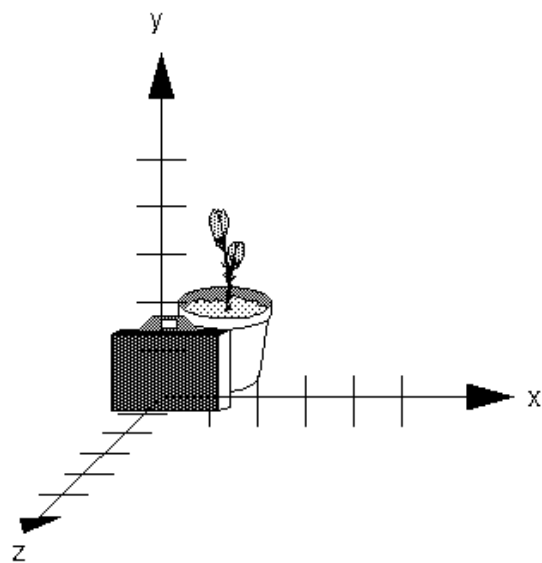
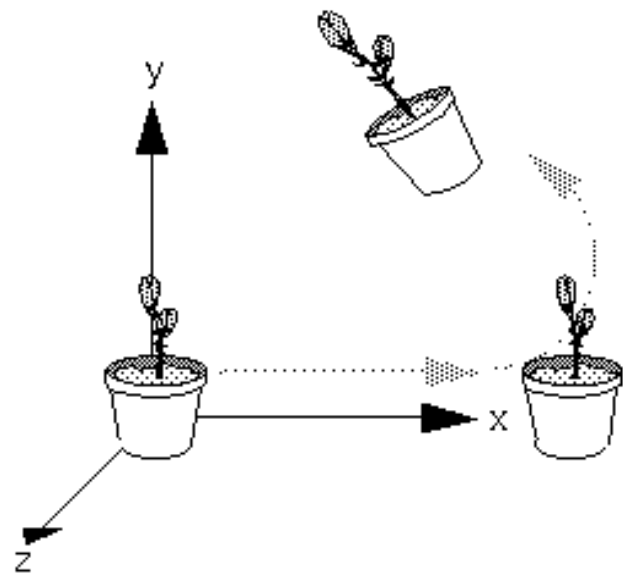
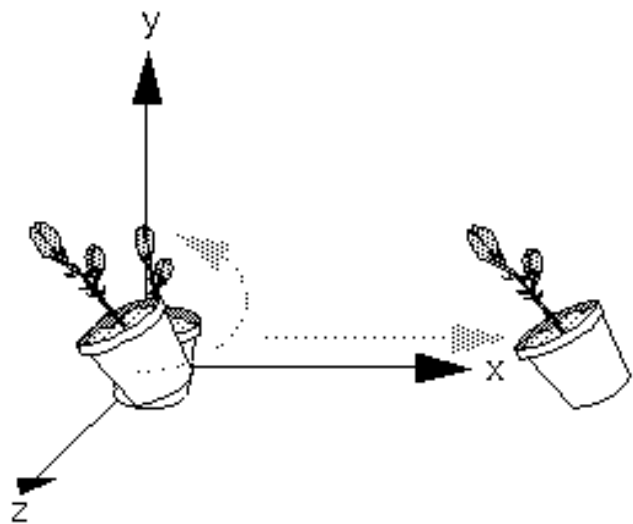


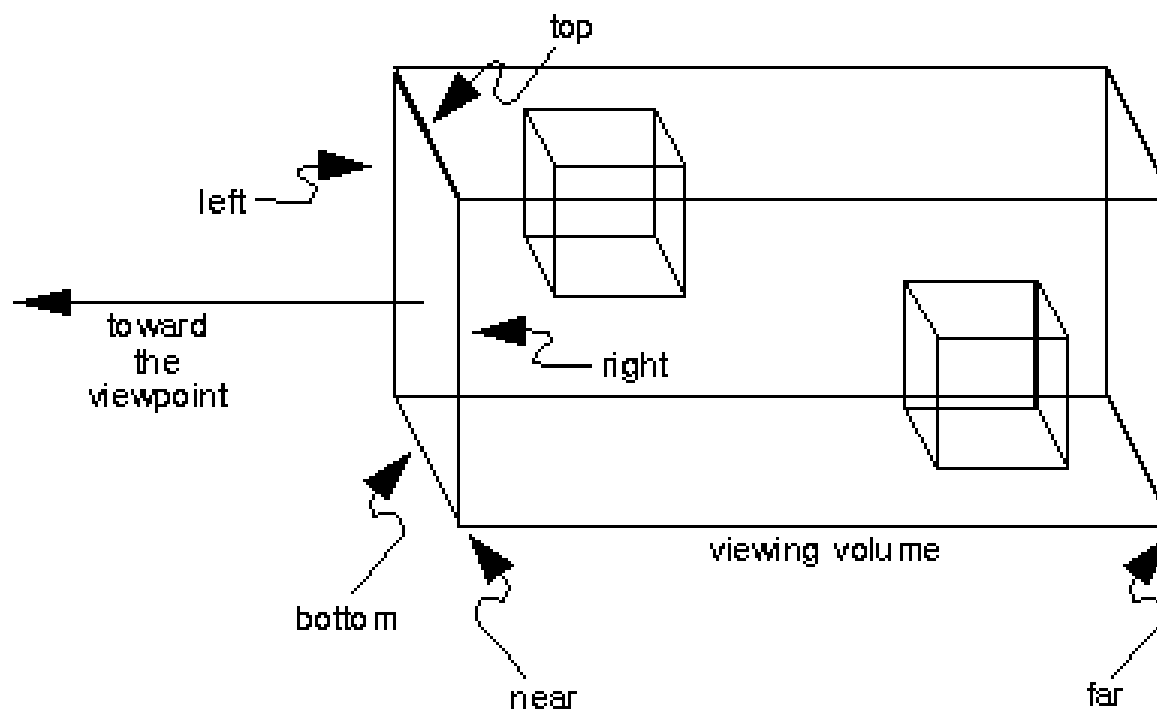
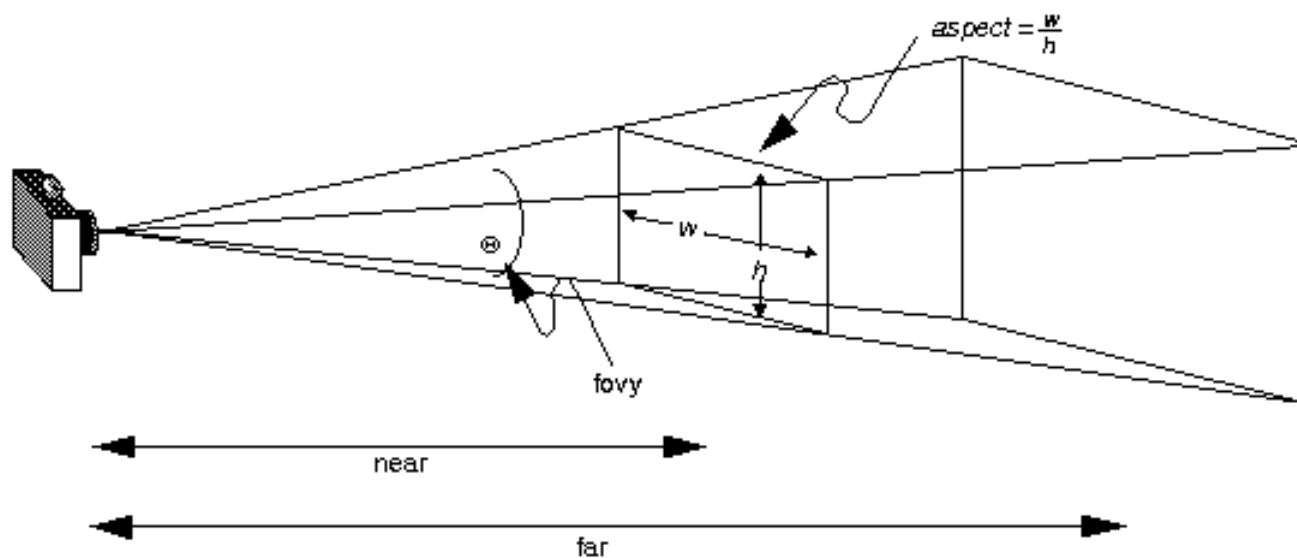


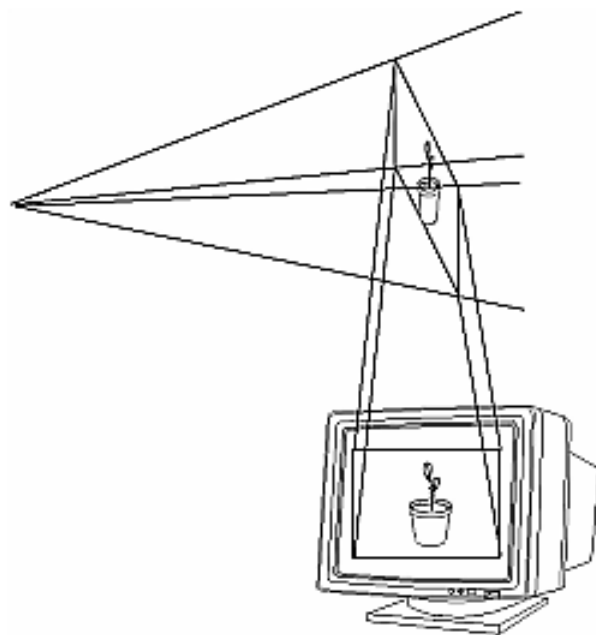
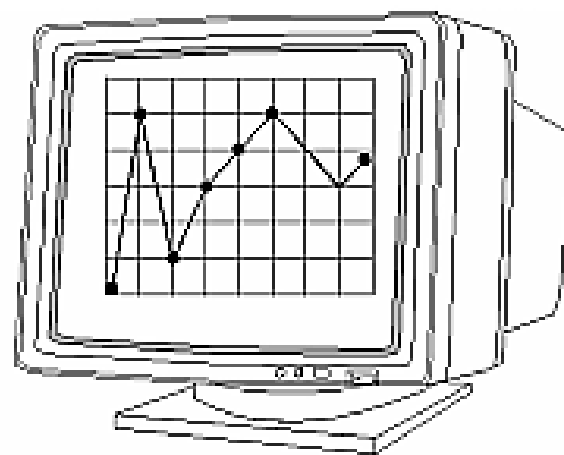
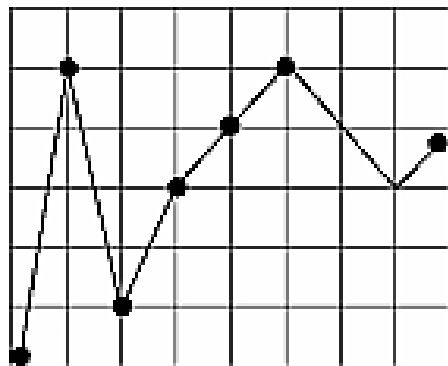
OpenGL绘制流水线



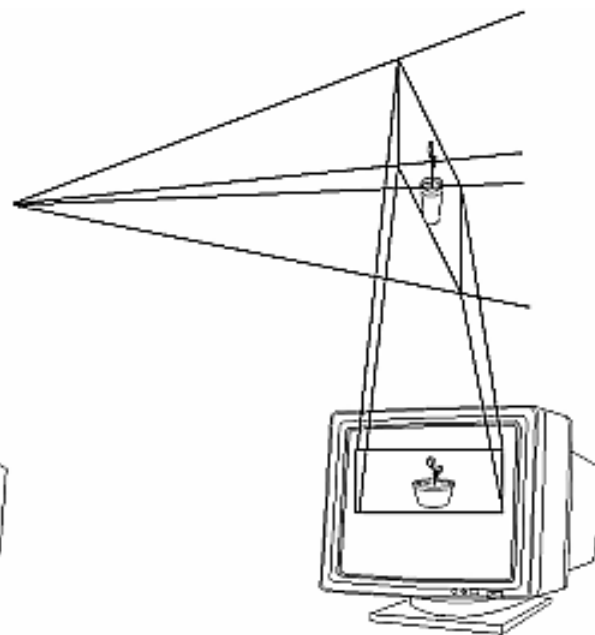
计算机成像过程与照相过程的类比







undistorted



distorted



影响图形绘制性能的因素

- ◆ 几何复杂度
- ◆ 光照模型复杂度
- ◆ 纹理映射

问题：如何提高绘制速度？

针对不同的因素采取不同的方法。

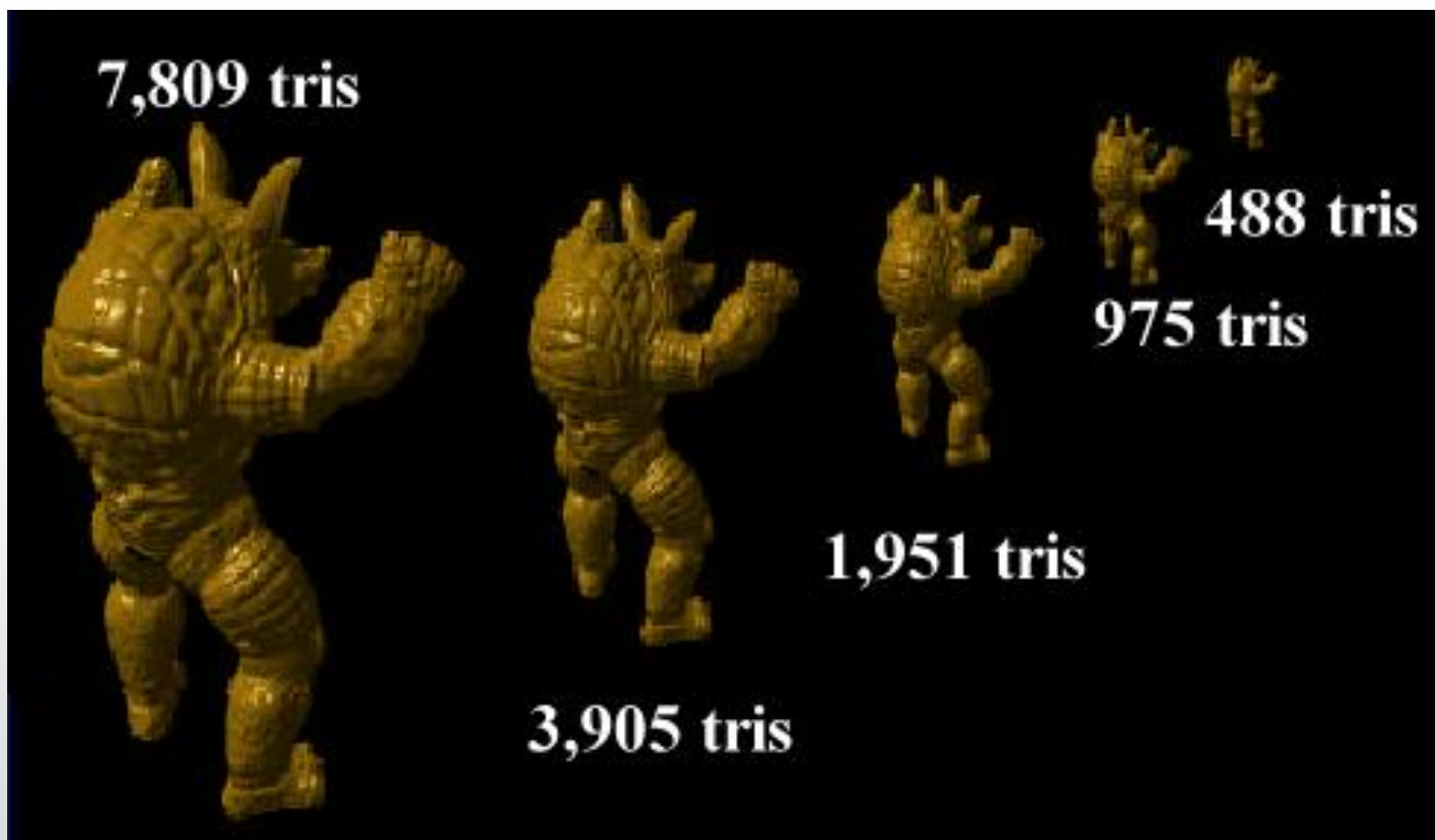
降低场景几何复杂度

- ◆ 场景简化 和 网格多分辨表示
- ◆ 快速消隐
- ◆ 基于BSP树的绘制
- ◆ 纹理映射技术
- ◆ 基于图象的绘制



网格简化技术

场景简化





层次细节模型:

在大规模的虚拟环境中，很多对象很小，获距离视点很远。按透视规律，在屏幕上只占很小的图象区域。为了更好地利用这一特点，一个对象应该用多层次细节表示。在交互绘制过程中，使用简单的对象的表示可以提高帧速率

- 根据观察者的心理、生理特点和场景画面限时计算的要求，在视觉等价的前提下，对要绘制的复杂场景进行简化,生成可表达不同层次细节的场景模型。
- 在多层次细节建模和三角形网格简化方面，我们设计和实现了一系列新的算法，并提出了新的控制全局误差和局部误差的方法。

LoD网格-1



- ◆ 基于顶点删除的三角形网格模型简化 **Schroeder 1992**
 - 基于距离准则的顶点删除，顶点移去留下的空洞由一个局部的三角面片机制重新生成新的三角面片。
- ◆ 基于重新划分的多边形网格模型简化 **Turk 1992**
 - 引入新的顶点来优化三角形网格，生成新的几何模型。新的顶点均匀地分布在原对象表面。原来的顶点被逐步移去，表面进行局部的三角化，以最好地满足对象表面的连续性。
- ◆ 整体网格优化算法，全局能量优化方程 **Hoppe 1993**
 - 基于表面重构的三角网格简化方法。它引入了一种新的能量公式来建模多边形简化以及与简化模型与原模型的相似度。用能量函数来衡量原网格模型与简化模型的变形程度。算法可以找到使变形最小的优化的顶点分布。

LoD网格-2

◆ 基于顶点聚类的网格简化算法

Rossignac 1993

- 不考虑输入模型和输出模型的拓扑结构。结果，算法对于退化的输入数据具有很好的鲁棒性，对任意类型的几何数据都可以得到很高的压缩比。缺点是，简化模型的人工痕迹严重，模型的局部特征保持不好。
- 不同的选择准则来选取那些聚类的顶点。Rssignac提出了一种简单有效的一致量化方法。Schaufler则采用了分层的聚类方法。

◆ 边折叠与视点有关的累进网格

Hoppe 1997

- 基于边折叠操作，产生一个无损的，连续分辨率的三角网格。简化表示通过一系列重复的边折叠操作产生，很容易转换为递进的重构过程。

LoD简化网格的一些评价参数

- ◆ 网格简化算法的鲁棒性
- ◆ 简化网格的生成和解析速度
- ◆ 简化后的对象shape是否保持不变
- ◆ 简化后的网格是否保持拓扑结构不变



LoD几何模型

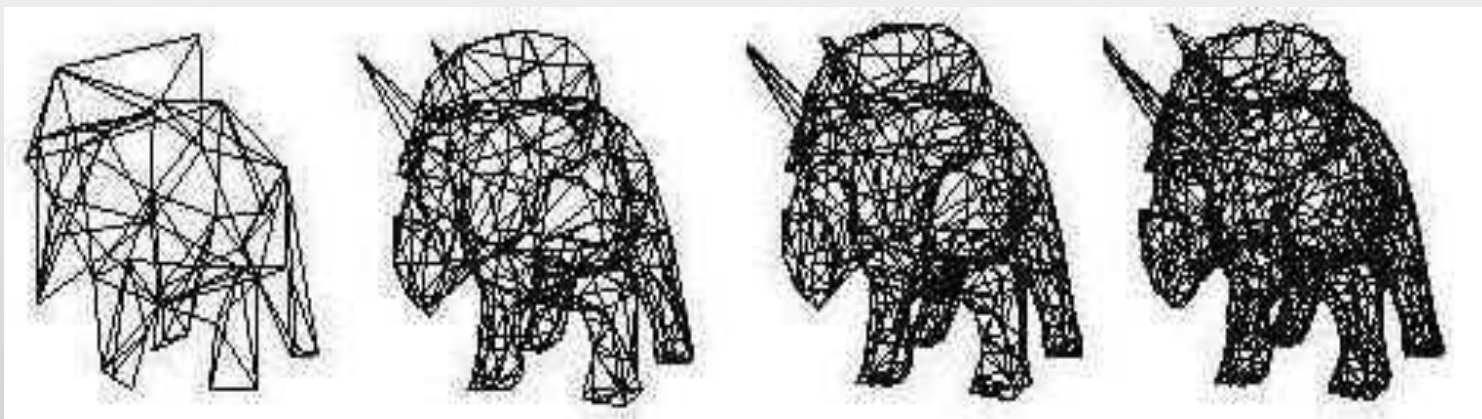
◆ 传统的网格的表示

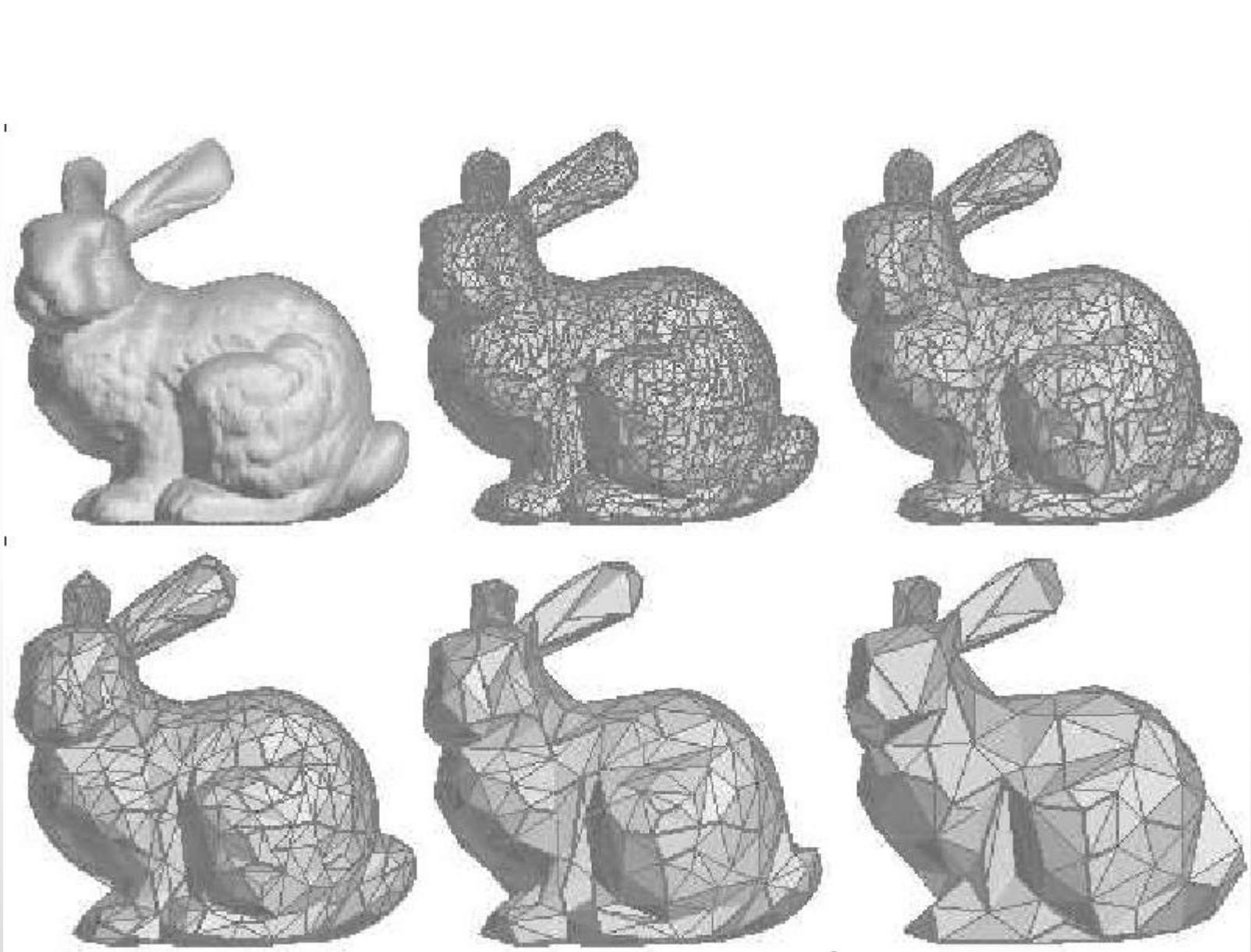
数据量庞大，不易于存储、传输 -- 实时性

◆ 很多情况下，我们并不需要一个对物体的细节刻画得很详细的模型

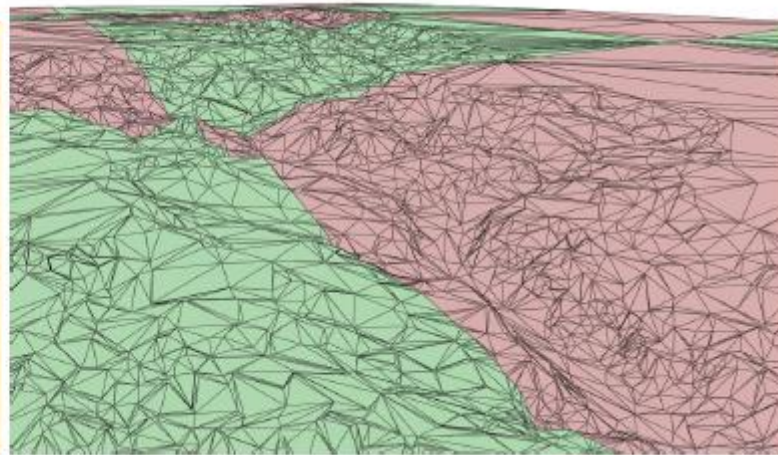
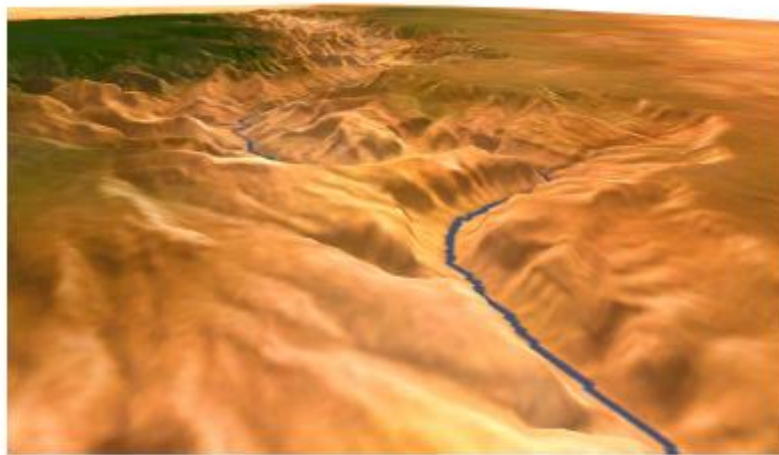
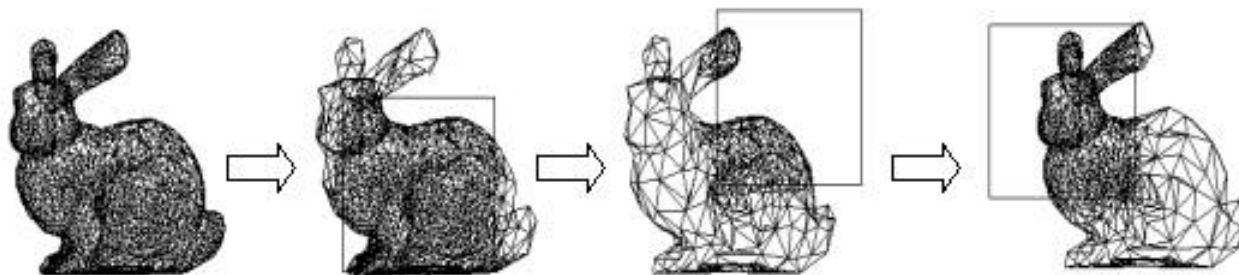
较远或物体在运动 -- 较粗糙网格

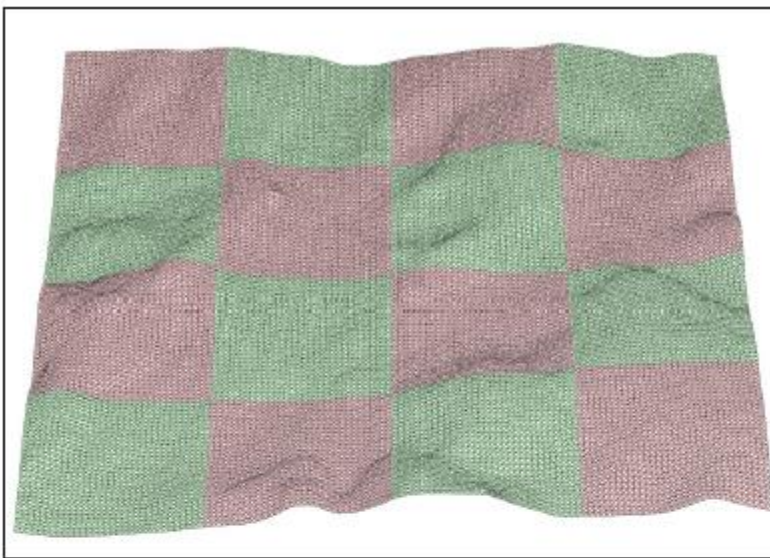
物体较近或物体静止 -- 精细网格



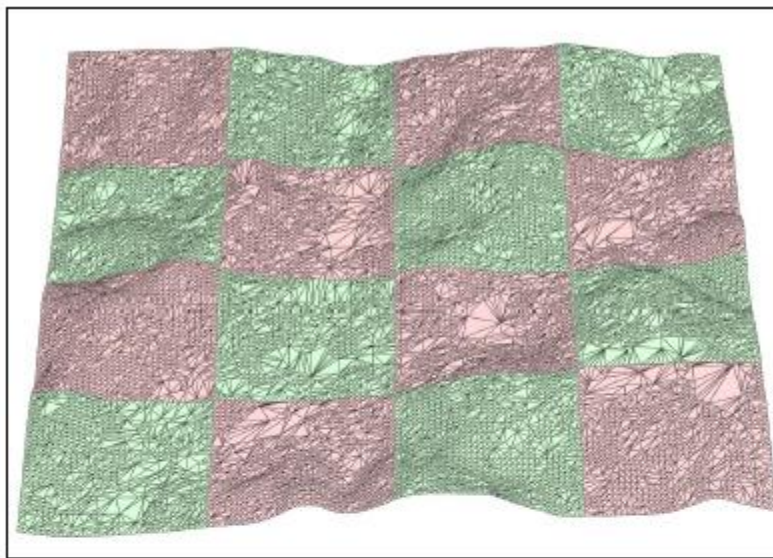


基于视点的网格简化

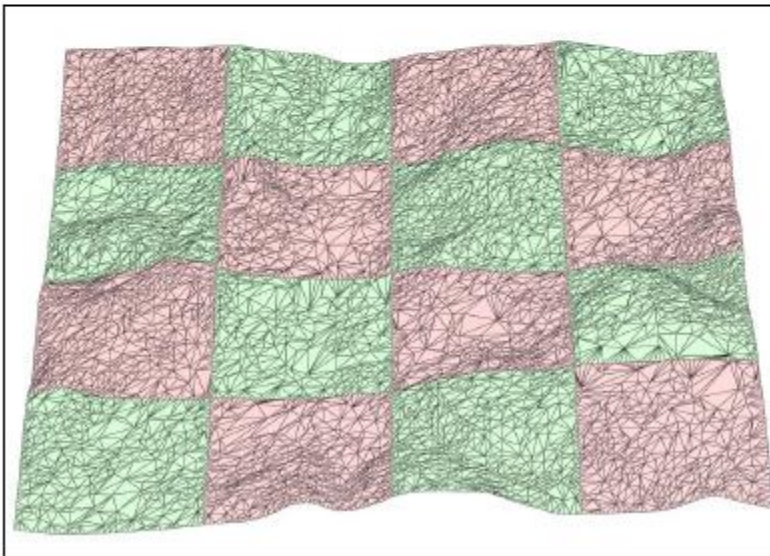




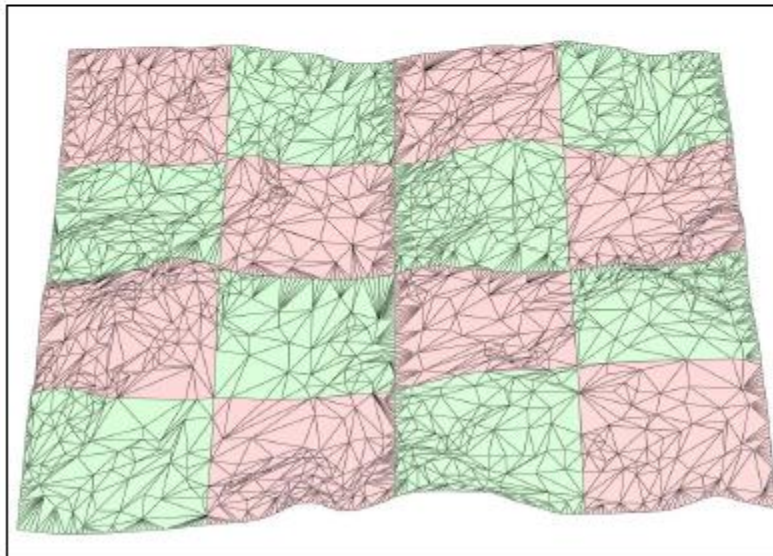
(a) Original mesh (129×129 vertices; 32,768 faces; $\max \delta=0$)



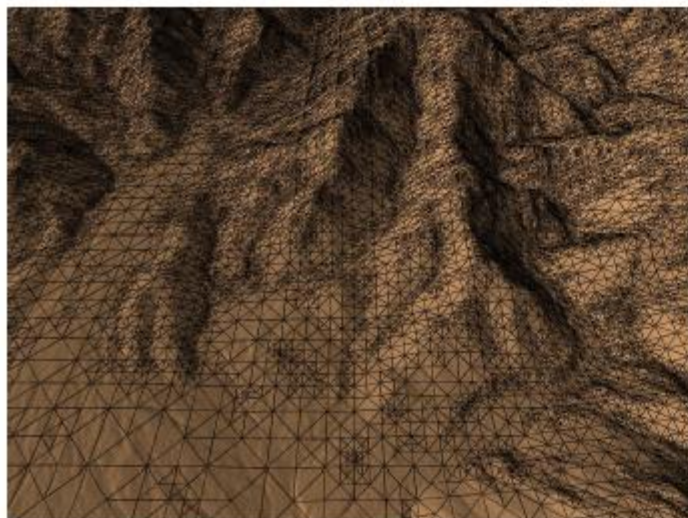
(b) Pre-simplified mesh (21,622 faces; $\max \delta=0.03\%$)



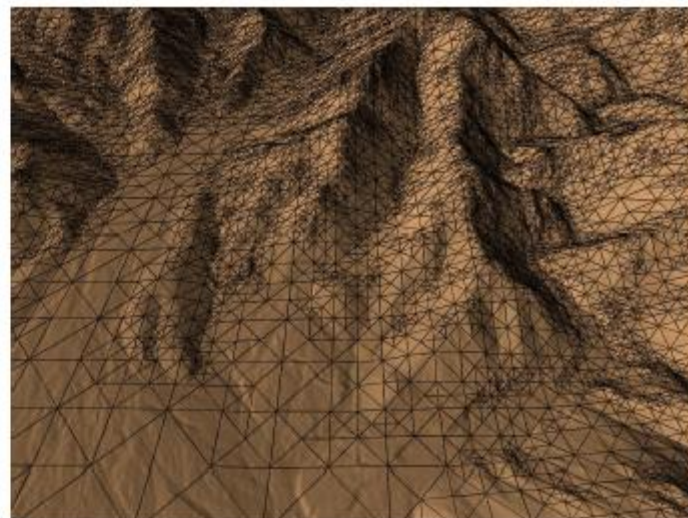
(c) Level 0 simplification (11,048 faces; $\max \delta=0.04\%$)



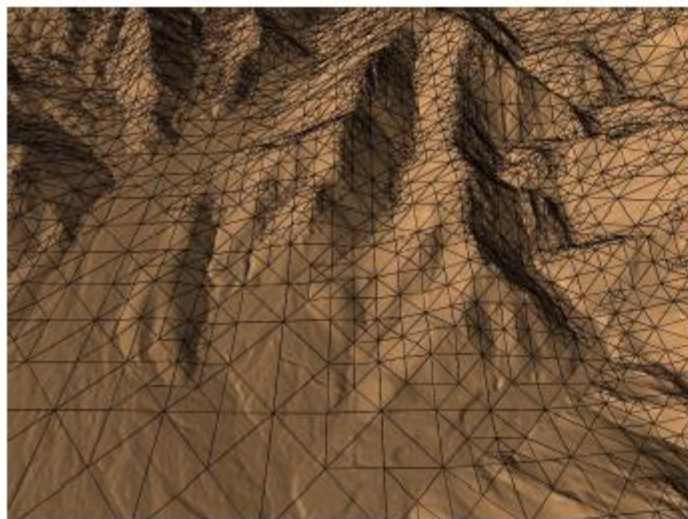
(d) Level 1 simplification (2,504 faces; $\max \delta=0.1\%$)



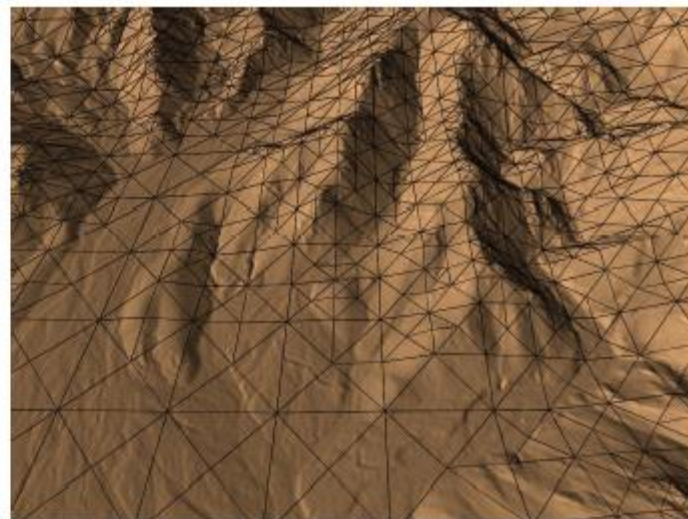
3a. $\tau = 0.5$, 62,497 polygons



3b. $\tau = 1.0$, 23,287 polygons



3c. $\tau = 2.0$, 8,612 polygons



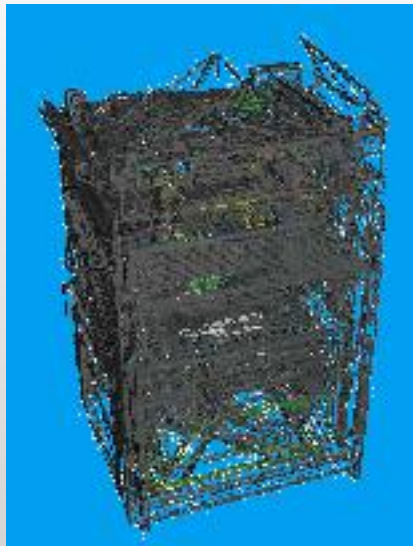
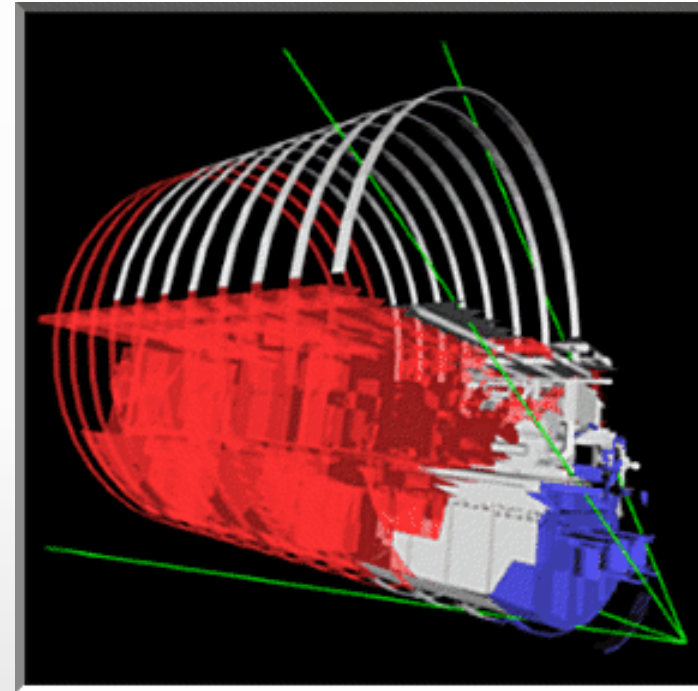
3d. $\tau = 4.0$, 3,385 polygons



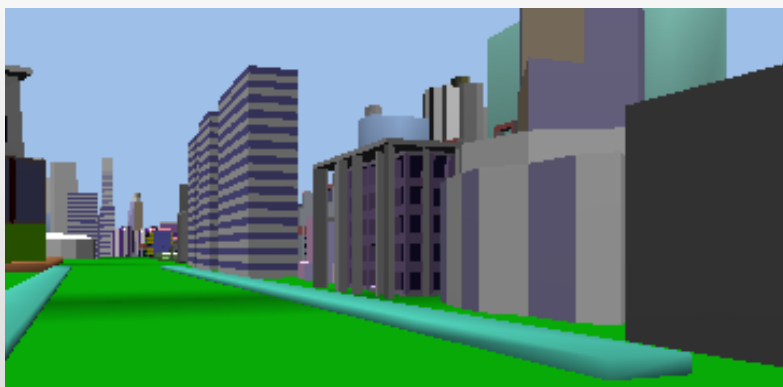
快速消隐技术

- ◆ 视见体裁减
- ◆ 背面剔除
- ◆ 遮挡剔除(Occlusion Culling)

遮挡剔除(Occlusion Culling)



快速消隐的例子





基于BSP树的场景绘制技术

- ◆ BSP树方式的场景组织
- ◆ BSP树的遍历

注：一个二元空间划分树（binary space partitioning）被称为BSP树，使用一个分离面（splitting plane）对每一层一分为二，从而实现对空间的划分。



纹理映射

◆ 基本思想:

实际纹理非常复杂，难以解析描述。
采用图象来描述表面纹理细节。

IBMR (Image-based Modeling and Rendering)

◆ 关键问题:

如何在光照明模型中融入纹理的描述？

如何将纹理绘制在景物表面上？



光亮度计算

1. 依据：光照明模型

- 局部光照明模型：朗伯模型、Phong模型
- 整体光照明模型：Whitted模型，光能辐射度方程（隐式）

追求目标：光照效果的真实感



2. 计算方法：绘制算法

- 扫描线方法
- 光线跟踪方法
- 光能辐射度方法

追求目标：画面生成的实时性



纹理映射的基本原理

纹理生成过程实质上是将所定义的纹理映射为反映某种三维景物表面的属性，并参与后续的光照明计算。

表面属性：与光照明模型及表面几何有关的各种参数，如表面法向、漫/镜面反射率等。

纹理映射的实现:

- 交互确定纹理属性

$$I_{local} = k_a I_a + k_d (N \cdot L) + k_s (N \cdot H)^n$$

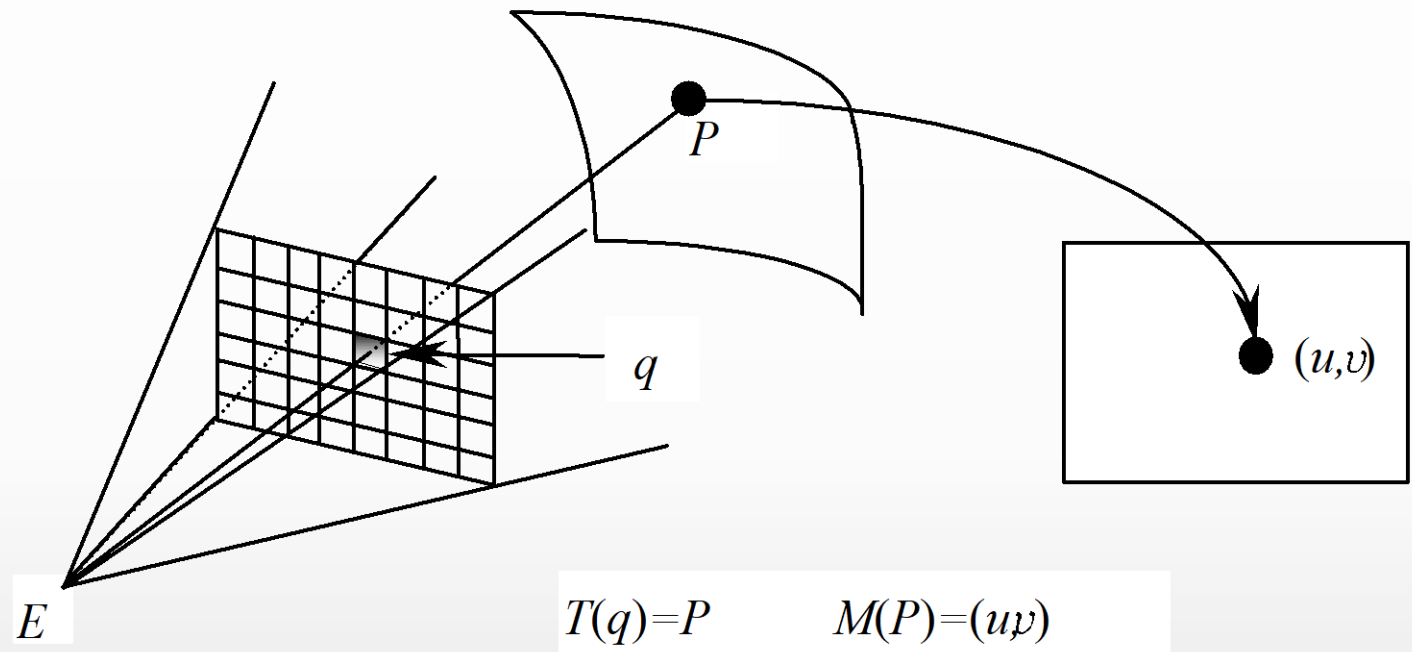
$$I = I_{local} + s I_s + t I_t$$

参数不再是常数，逐点变化

建立纹理空间与景物空间及景物空间与屏幕空间之间的映射关系：

$$M: \textit{ObjectSpace} \rightarrow \textit{TextureSpace}$$

$$T: \textit{ScreenSpace} \rightarrow \textit{ObjectSpace}$$





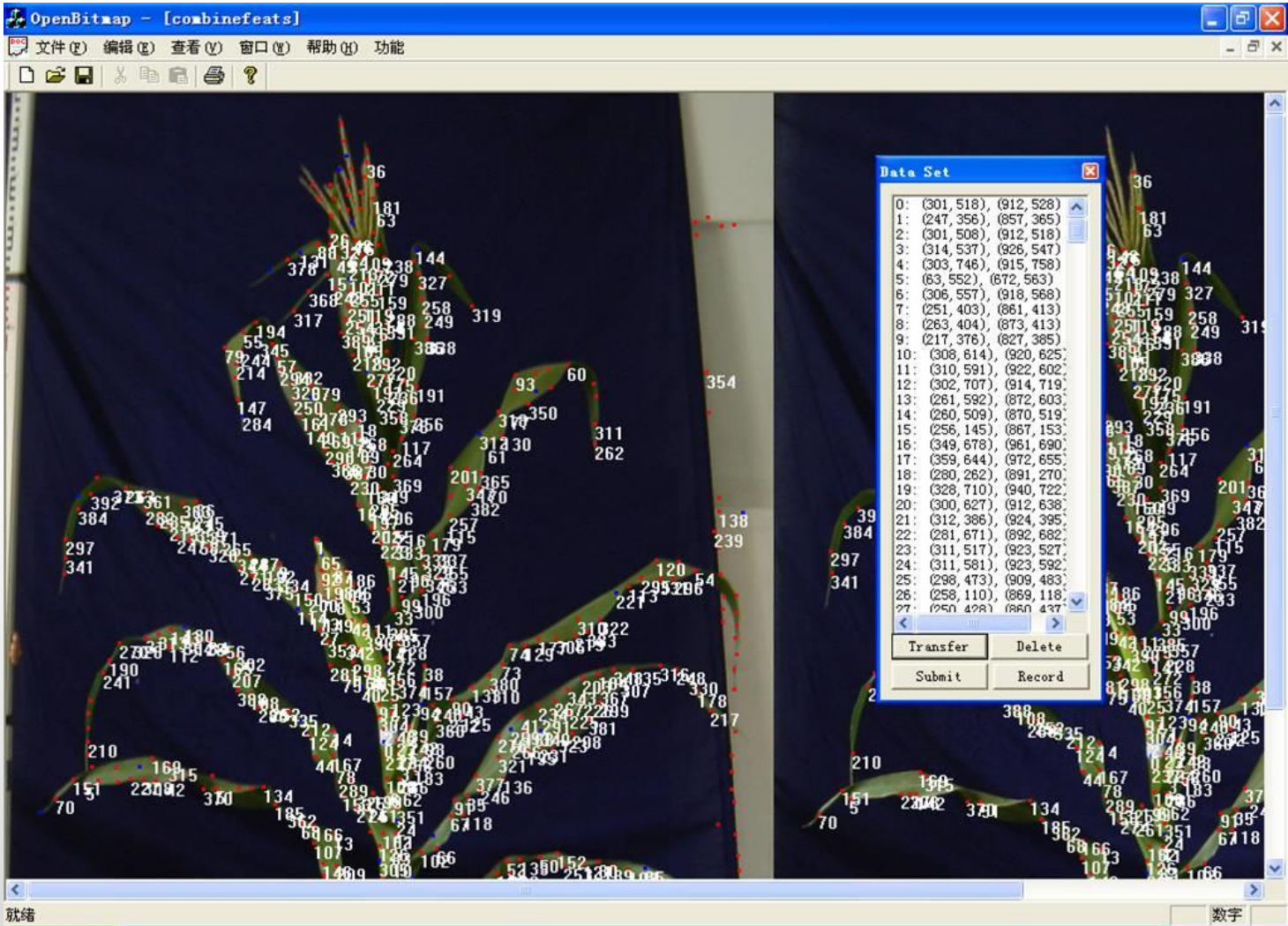
景物表面的纹理属性主要有以下几种：

- (1) 表面颜色, 即表面的漫反射率;
- (2) 镜面反射分量, 即表面的镜面反射率
- (3) 透明度;
- (4) 表面法向, 即扰动表面法向来产生表面的凹凸纹理;
- (5) 环境的漫反射和镜面反射效果;
- (6) 光源强度和色彩分布。

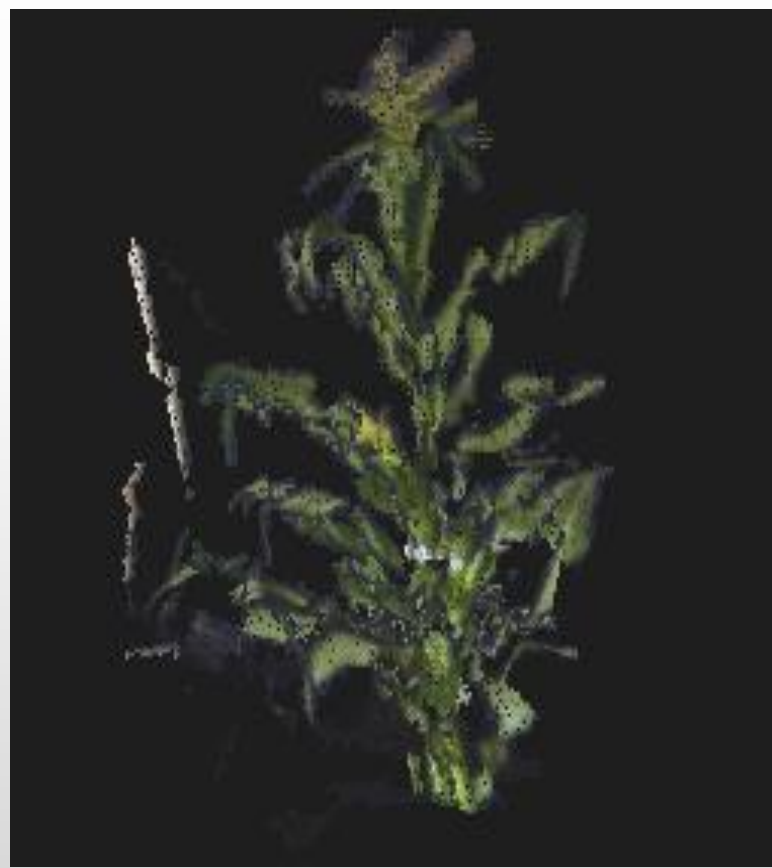


基于图象的建模与绘制

- ◆ 基于图象的建模—IBM
- ◆ 基于图象的绘制—IBR



基于图片的三维重建



基于图象的绘制—IBR

- ◆ 基于立体视觉的方法
- ◆ 基于视图插值的方法
- ◆ 基于拼图和分层的方法
- ◆ 基于全视函数的方法

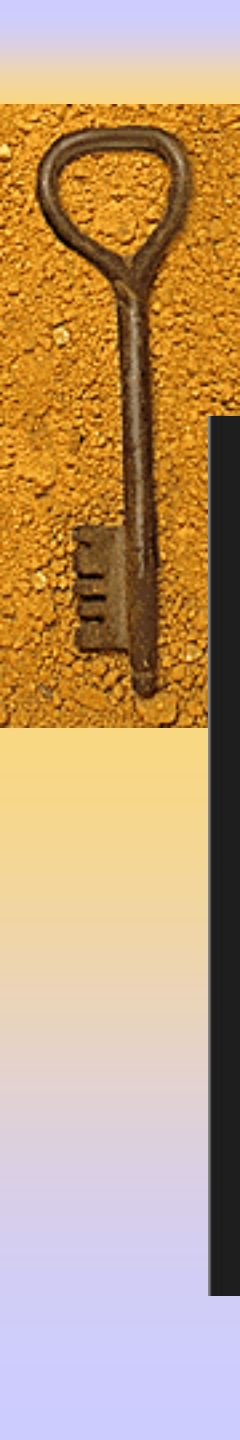


◆ 基于全景图的绘制算法

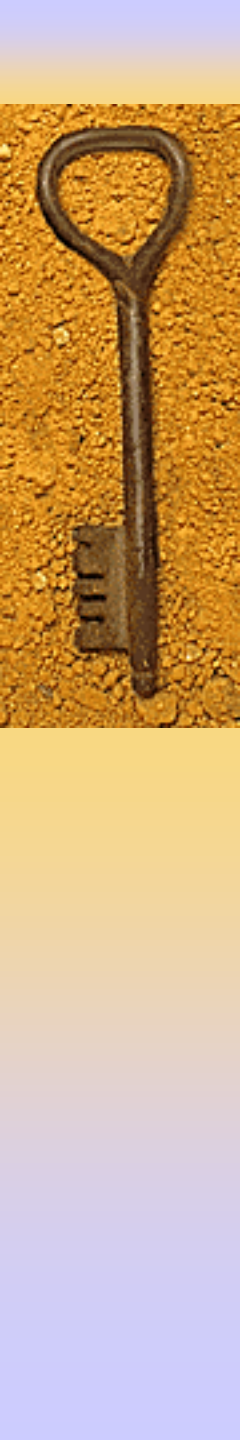


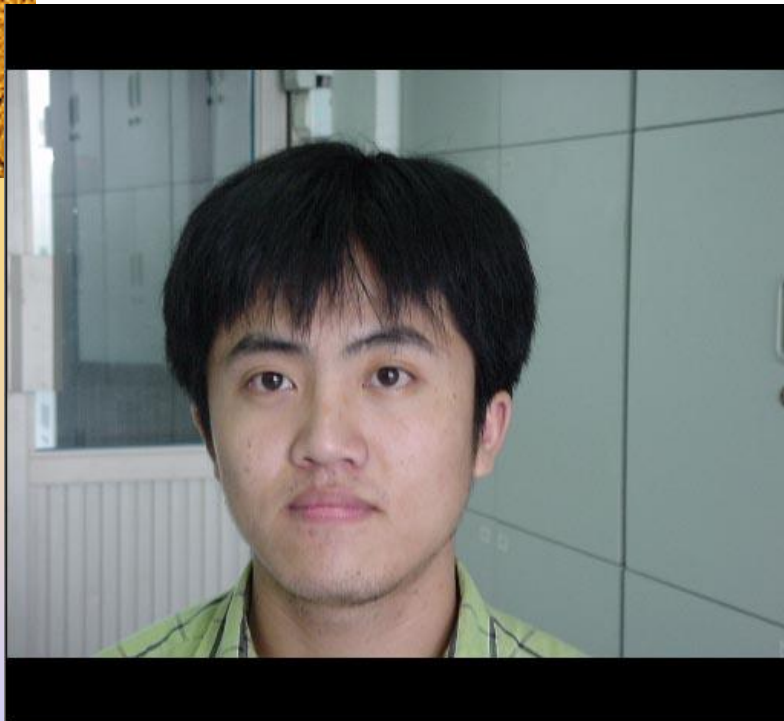


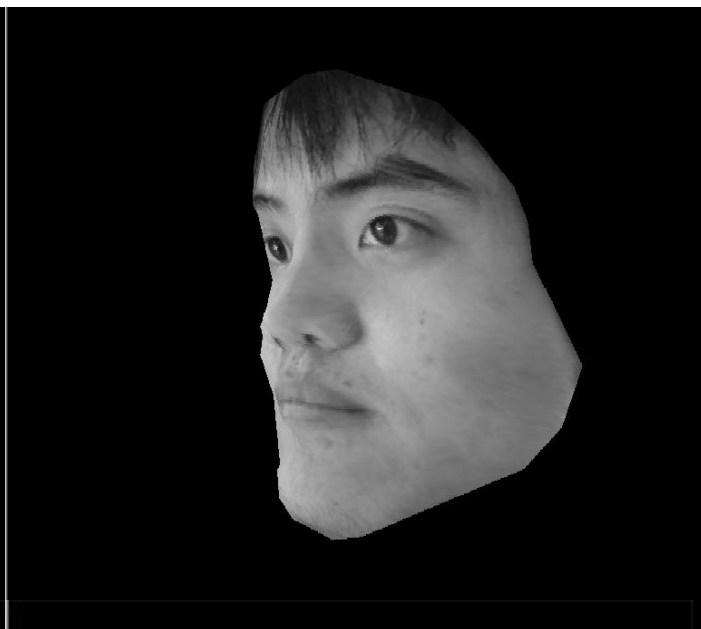














感谢潘志庚研究员！本课件是在潘老师的课件基础上补充而成的

2009版

2016版

2017版