物体的多边形表示及数据结构

冯结青

浙江大学CAD&CG国家重点实验室

物体的多边形表示及数据结构

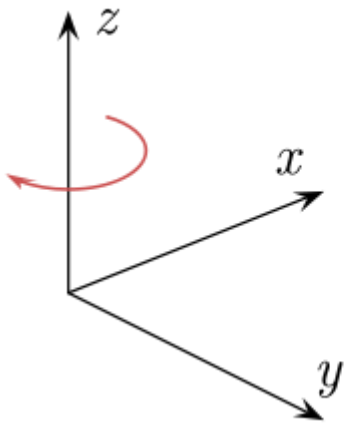
- 世界坐标系和景物(局部)坐标系
- 物体的多边形表示及其半边结构
- 图形学中常用的加速数据结构

物体的多边形表示及数据结构

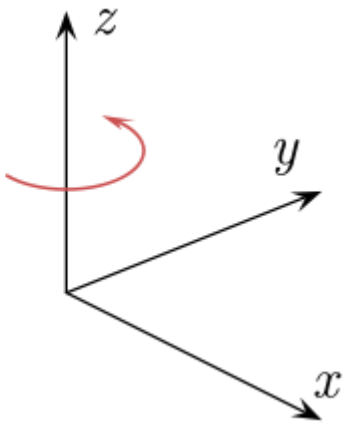
- 世界坐标系和景物(局部)坐标系
- 物体的多边形表示及其半边结构
- 图形学中常用的加速数据结构

世界坐标系和局部(景物)坐标系

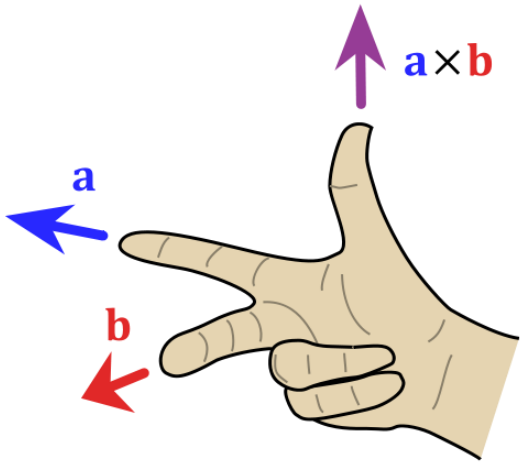
- 在计算机图形学中，常用的是空间直角坐标系
  - 空间任何一点可以用三个坐标(x, y, z)表示
  - 空间直角坐标系有两种：左手系和右手系



左手坐标系

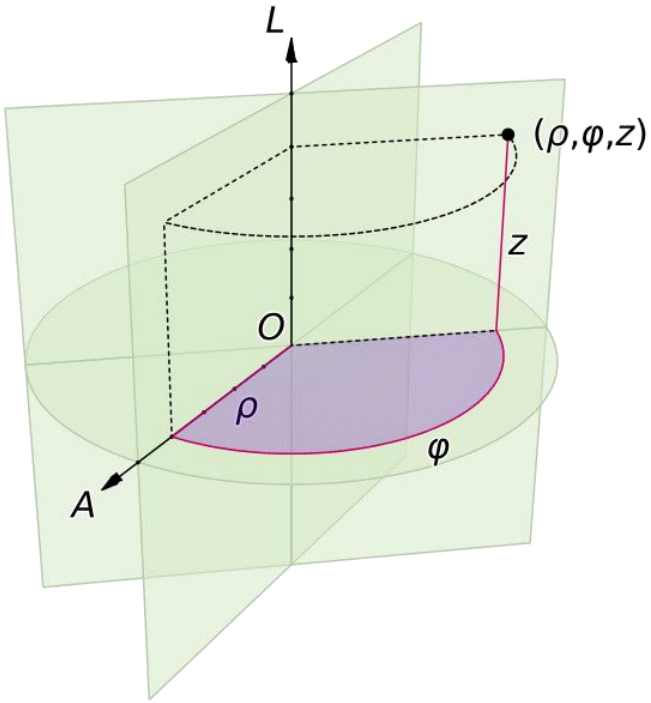


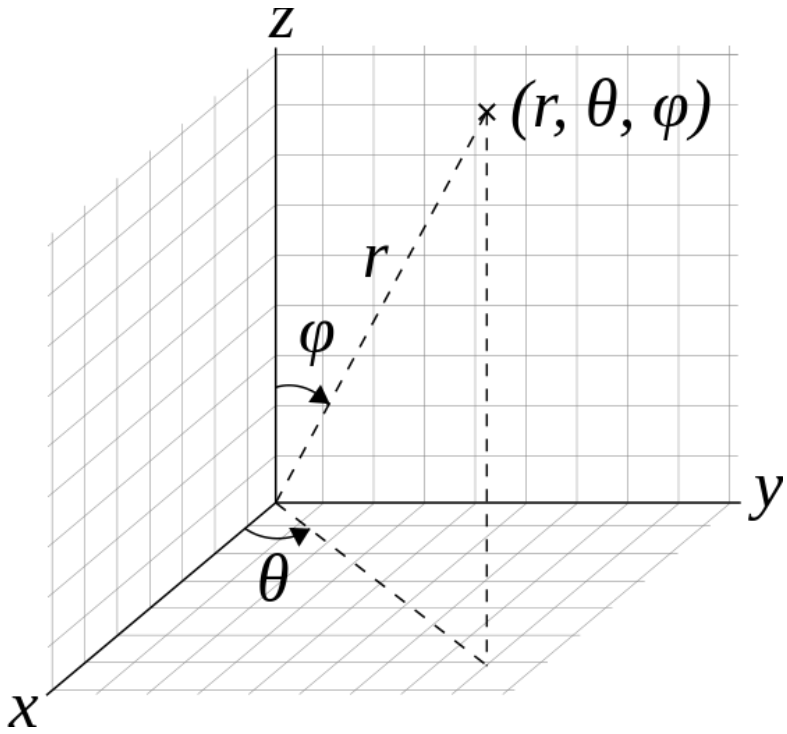
右手坐标系



右手坐标系

柱坐标系和球坐标系





柱坐标 (ρ, φ, z)：径向距离ρ、角坐标 φ、高度 z

球坐标(r, θ, φ)：径向距离 r、方位角 θ、极角 φ.

世界坐标系和局部(景物)坐标系

- 几何场景定义在一个世界坐标系中，它由许多几何物体组成
- 几何物体的描述与空间坐标系密切相关
  - 对于相同的几何物体，在不同的坐标系中会有不同的表示形式 。
  - 局部坐标系：几何物体的表示、操作与处理尽可能简单

以单位球面为例

表示1：

$x^2 + y^2 + z^2 = 1$　球心在原点

表示2：

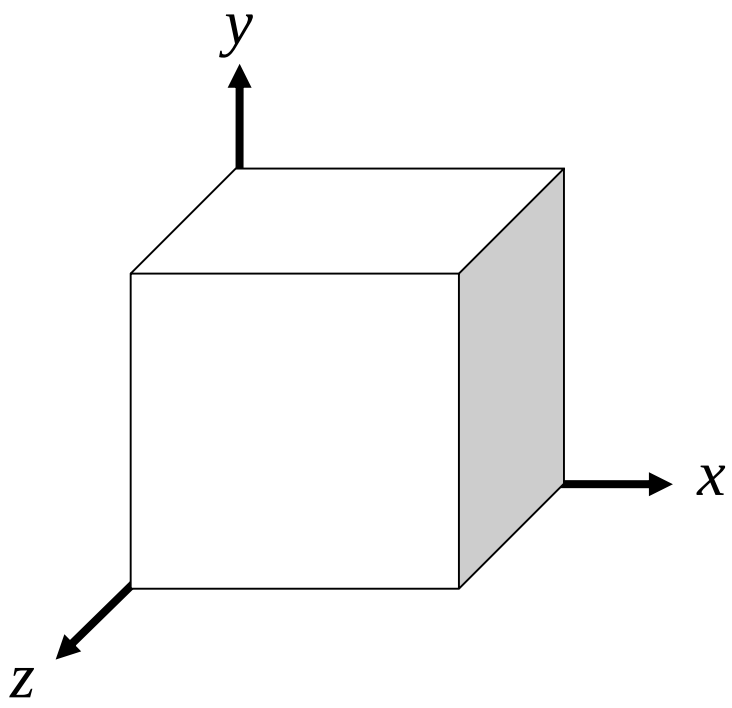
$(x-1)^2 + (y-1)^2 + (z-1)^2 - 1 = 0$　球心在(1,1,1)

最终表达式：

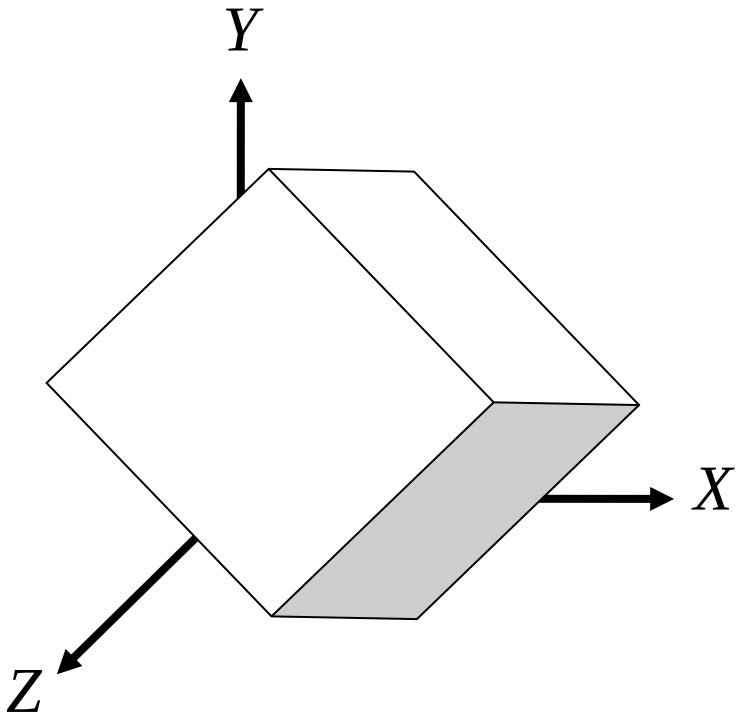
$$x^2 + y^2 + z^2 - 2x - 2y - 2z + 2 = 0$$

表示1最为简单，**表示2**较为复杂。

世界坐标系和景物(局部)坐标系



局部坐标系



世界坐标系

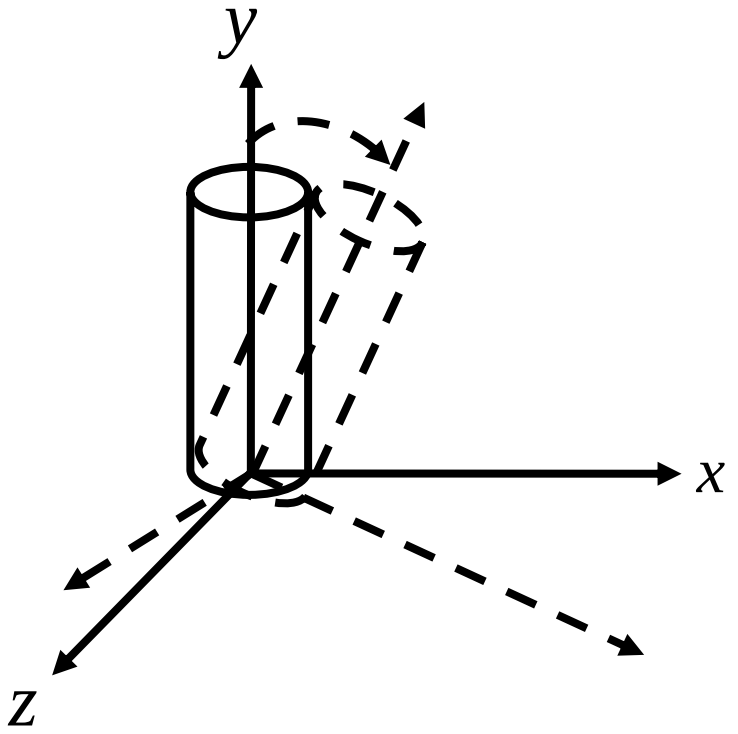
定义在局部坐标系和世界坐标系中的单位立方体

世界坐标系和局部(景物)坐标系

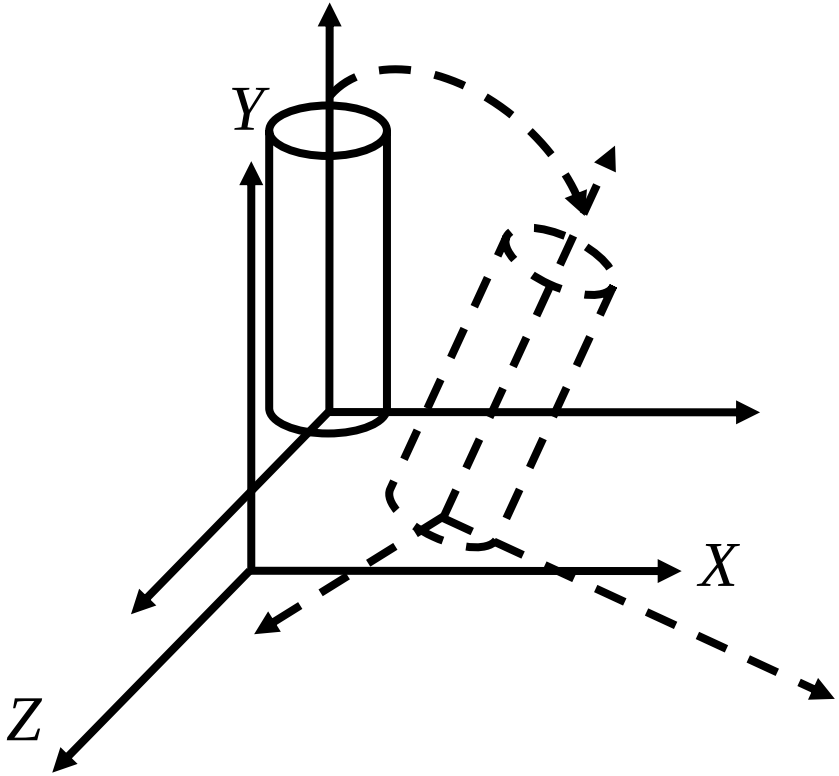
- 在局部坐标系中而不是在世界坐标系中直接表示几何物体
  - 几何物体具有简单的表示形式
  - 在同一几何场景中，同一个几何物体可能会多次出现，这些几何物体可以看作从局部坐标系到世界坐标系变换所得到的复制：

    标准体素 ＋ 变换 ＝ 新的物体
  - 局部坐标系有利于进行几何操作

局部和世界坐标系变换



局部坐标系中的变换



世界坐标系中的变换

在局部坐标系和世界坐标系中旋转一个圆柱面

世界坐标系和局部(景物)坐标系

- 模型变换：世界坐标系和局部(景物)坐标系之间的关系
  - 平移、旋转、放缩、剪切等
  - 上述变换的组合

物体的多边形表示及数据结构

- 世界坐标系和景物(局部)坐标系
- 物体的多边形表示及其半边结构
- 图形学中常用的加速数据结构

几何物体的多边形表示及其半边结构

- 几何物体的多边形表示
- 多边形表示的来源
- 多边形表示与2-流形
- 多边形表示的半边数据结构
- 多边形表示的不足

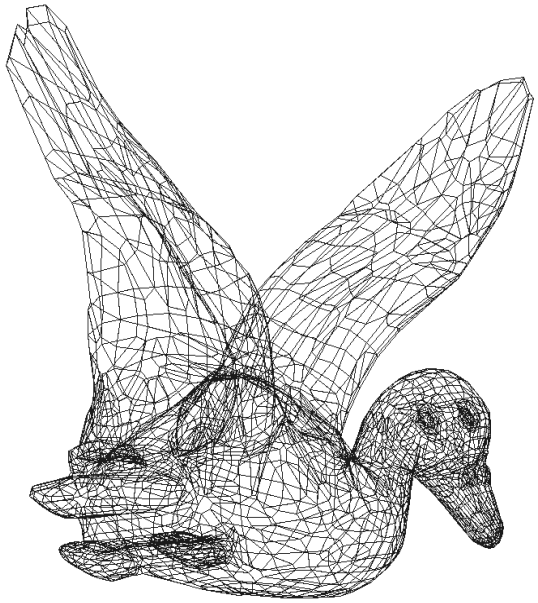
几何物体的多边形表示及其半边结构

- 几何物体的多边形表示
- 多边形表示的来源
- 多边形表示与2-流形
- 多边形表示的半边数据结构
- 多边形表示的不足

几何物体的多边形表示

- 物体的多边形表示：用大量平面多边形(三角形、四边形或者n-边形)来逼近几何物体的外形，也称为网格曲面
- 物体的多边形表示优点：
  - 表示简单：顶点、边、面
  - 可以表示具有任意拓扑的物体
  - 可以表示具有丰富细节的物体
  - 图形硬件支持多边形物体的加速绘制

几何物体的多边形表示



(a) 线框图

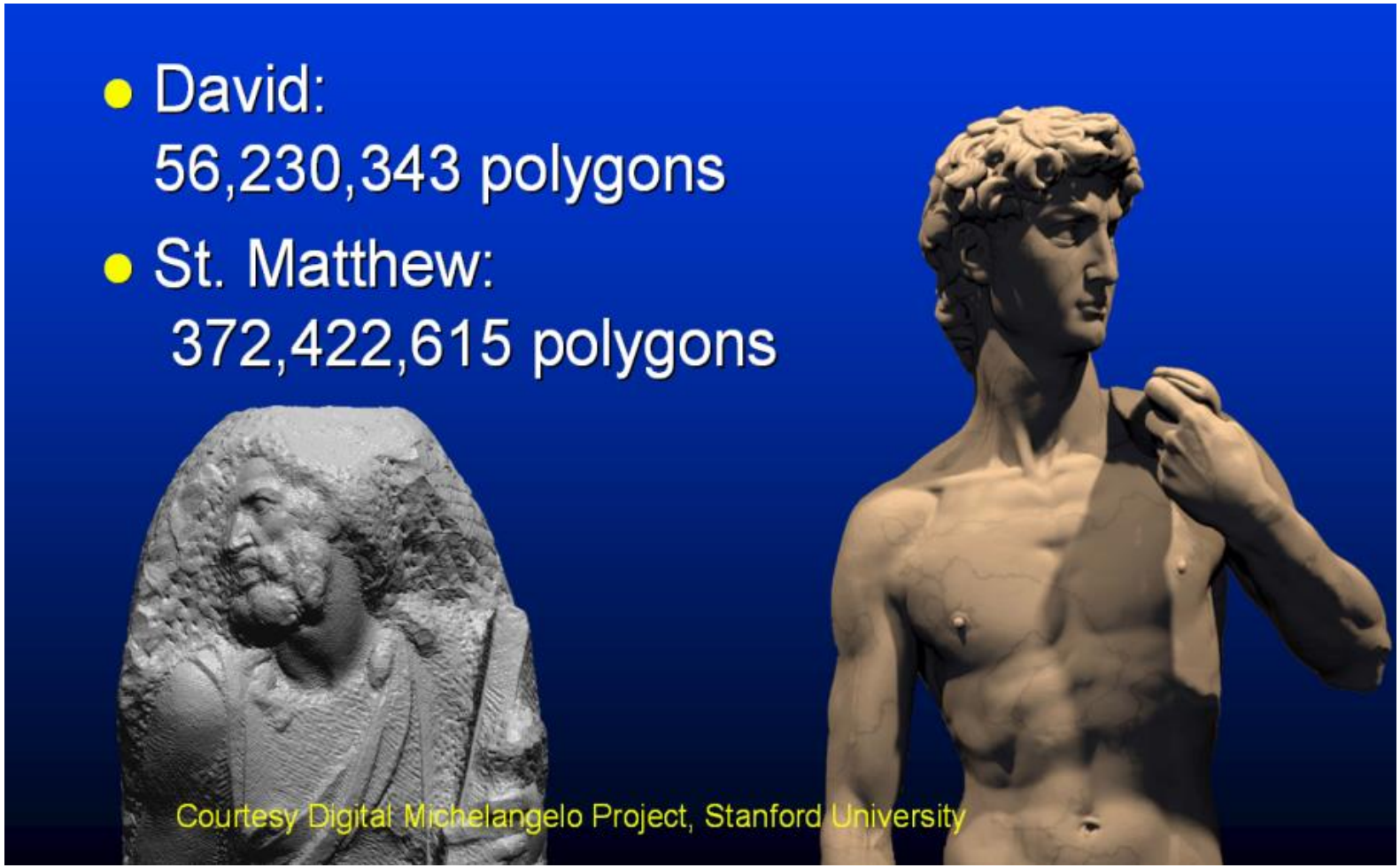
(b) 原始法向着色图

(c) 平均法向着色图

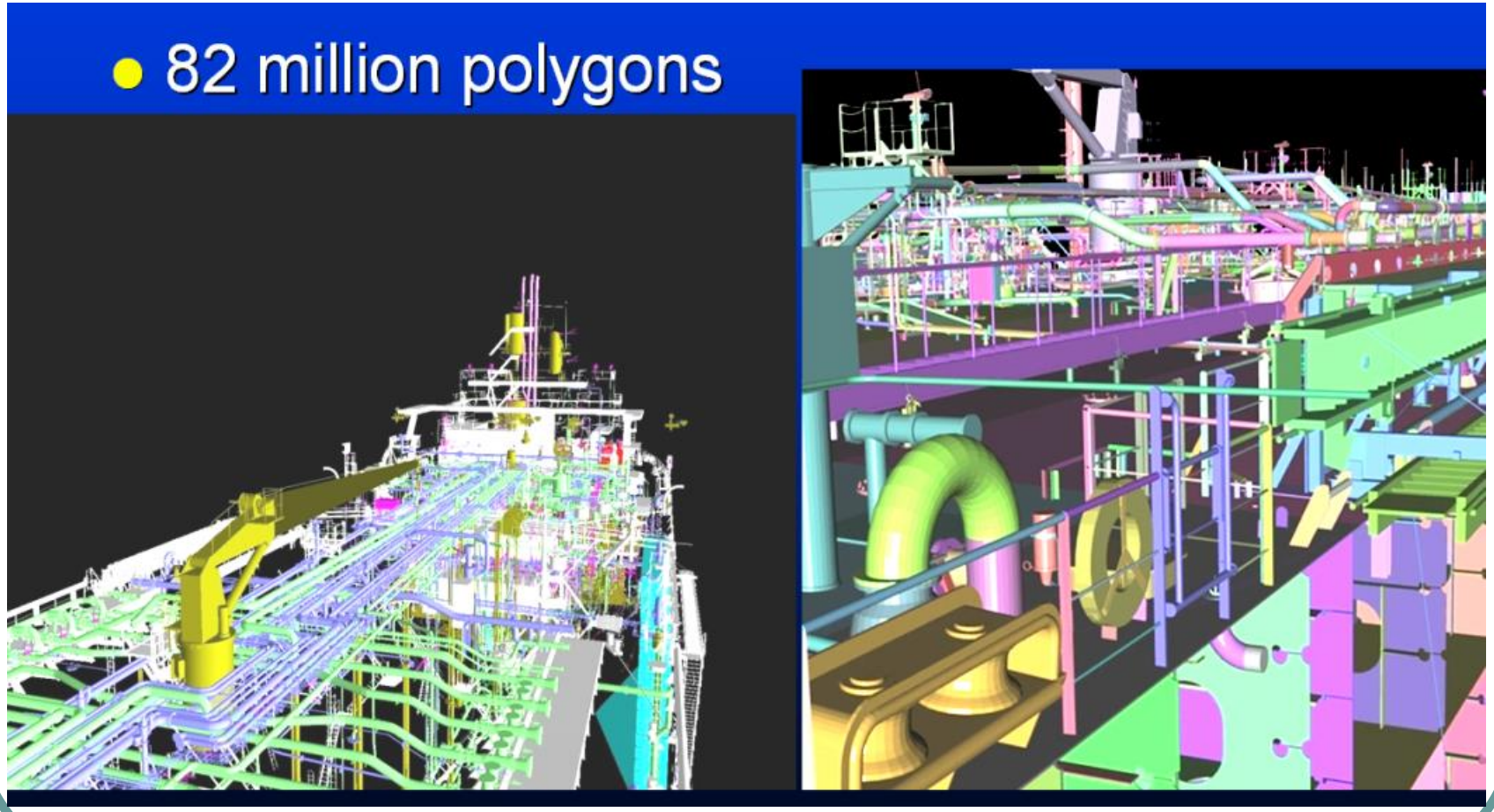
多边形表示的野鸭模型：6656个面片，3474个顶点

多边形表示的 大规模场景与模型

多边形表示的 大规模场景与模型



多边形表示的 大规模场景与模型



几何物体的多边形表示及其半边结构

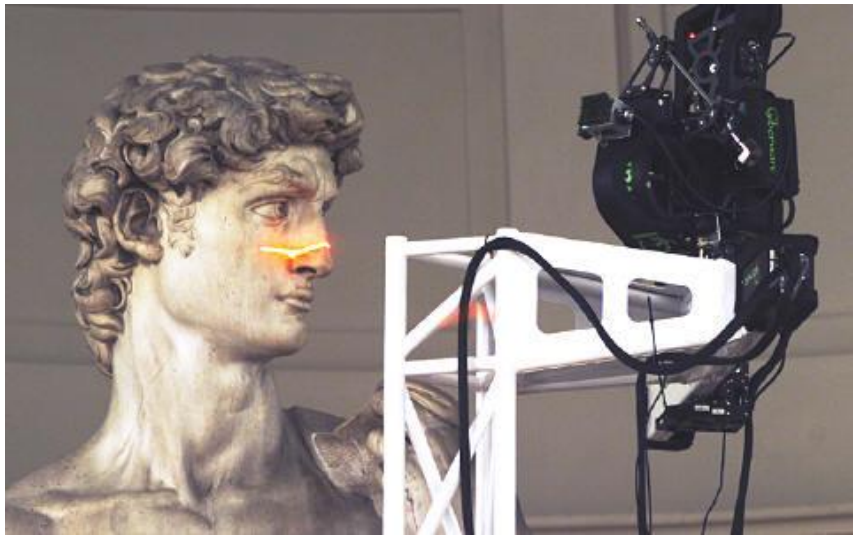
- 几何物体的多边形表示
- 多边形表示的来源
- 多边形表示与2-流形
- 多边形表示的半边数据结构
- 多边形表示的不足

多边形表示的物体主要来源

- 三维测量与扫描
  - 原始数据一般为三维空间中的点集
  - 采用适当的重建算法才能得到其多边形表示
  - 该方法适用于数学公式难以直接描述的、自然界存在的物体

接触式坐标测量仪

激光三维扫描

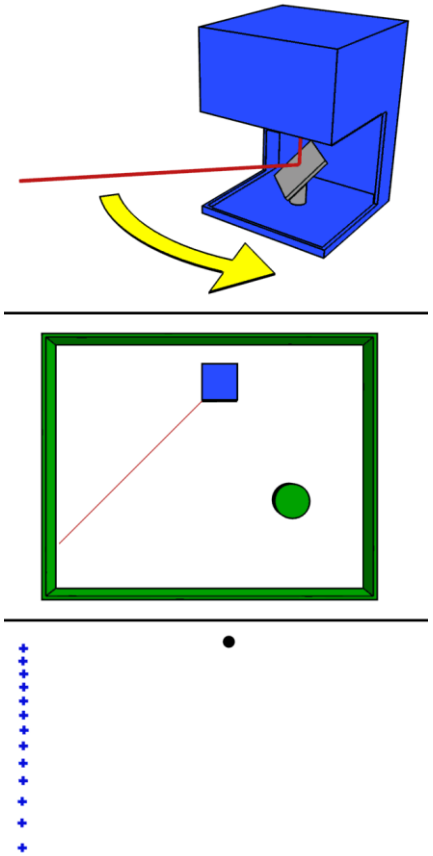


激光雷达(LiDAR/LIDAR/LADAR)

- Light Imaging, Detection, And Ranging
- Light Detection And Ranging

- 适用于建筑物、岩石、地貌等大范围场景

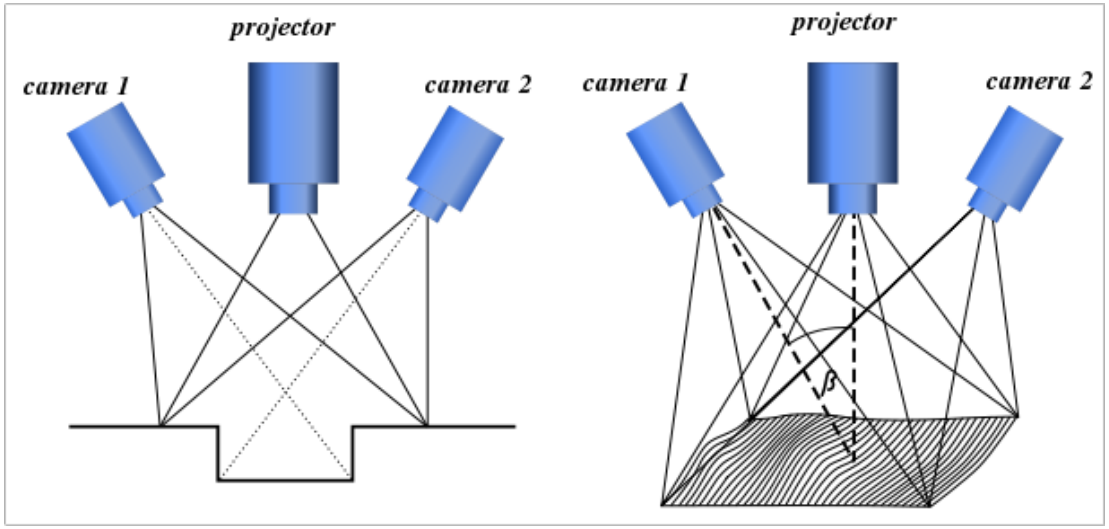
激光雷达(LiDAR/LIDAR/LADA)



LiDAR原理示意图

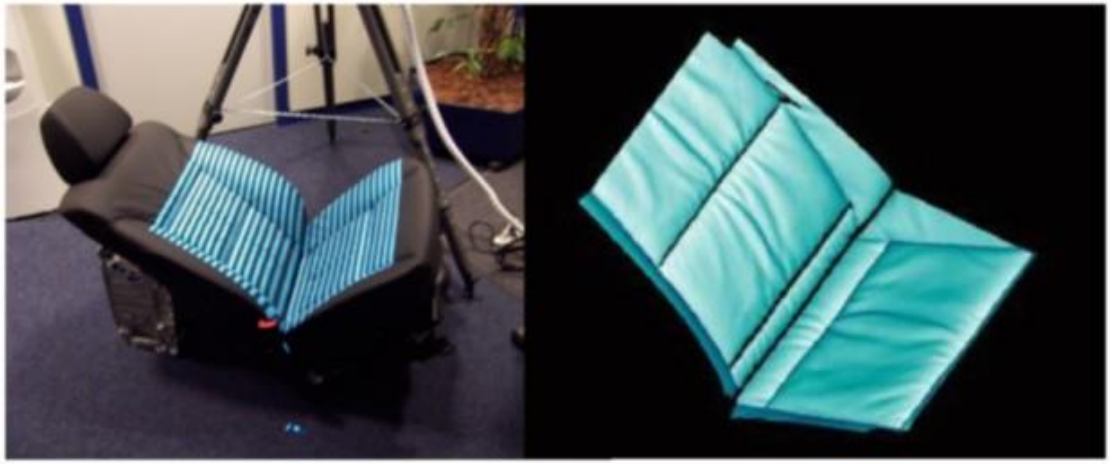
Leica HDS-3000

主动式立体视觉扫描(结构光)

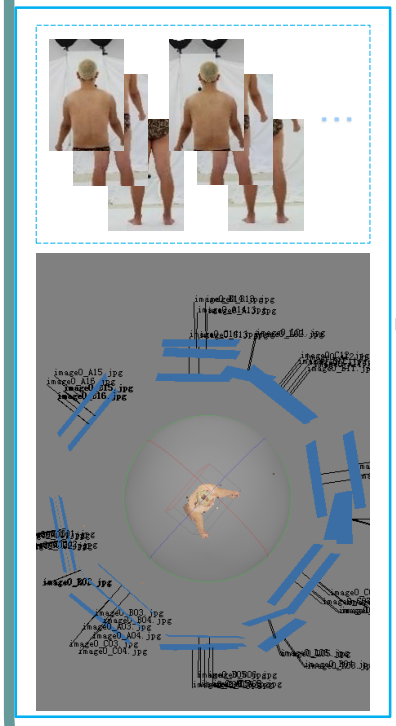


具有两个相机的条纹记录系统

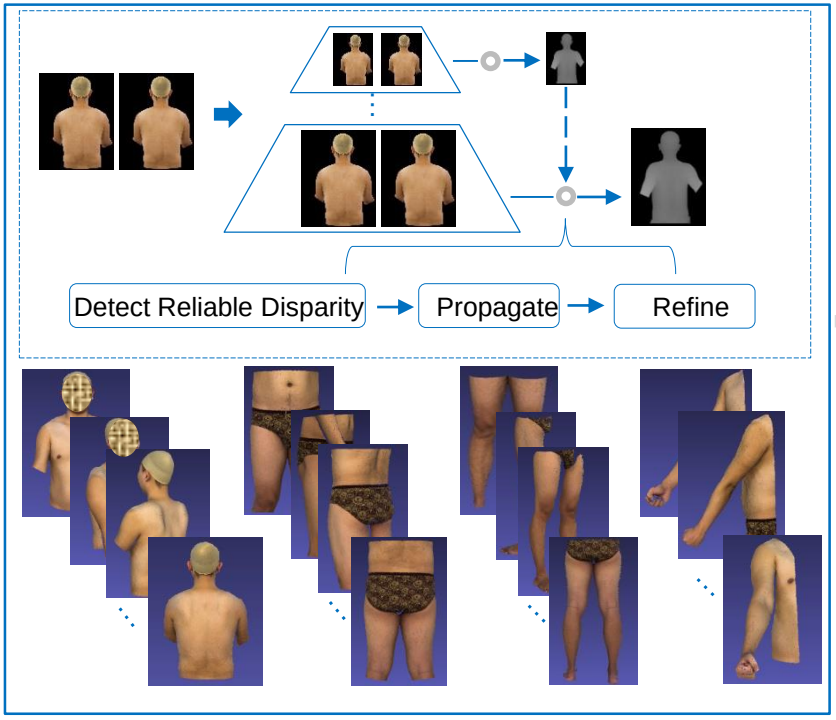
汽车座椅的三维扫描



被动式立体视觉扫描



Capture and Calibration

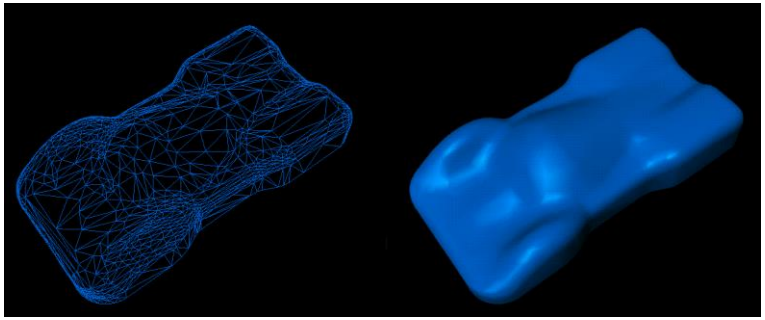


Pairwise Point-cloud Recovery by Hierarchal Stereo matching

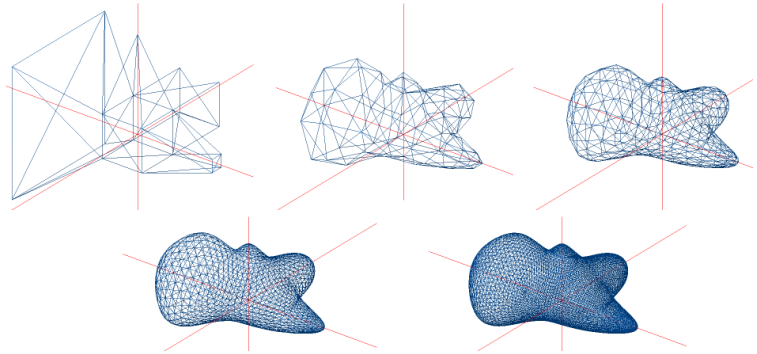
Registration and Meshing

多边形表示的物体主要来源

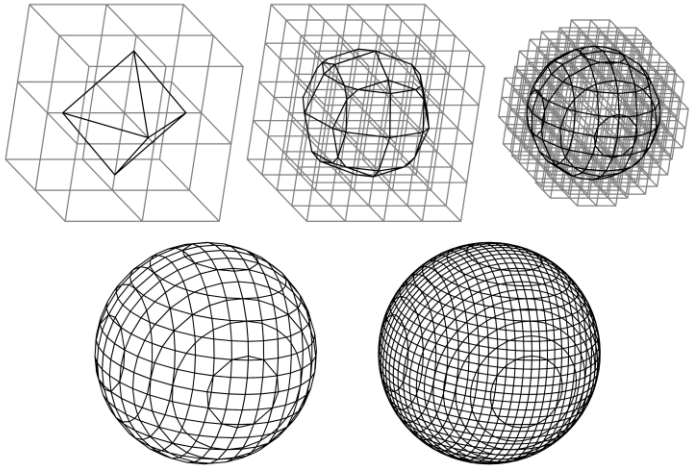
- 解析数学公式的逼近
  - 常用的几何物体数学表示方法包括参数曲面、细分曲面、隐式曲面等
  - 满足精度的曲面物体多边形逼近



参数曲面的多边形逼近



细分曲面的多边形逼近

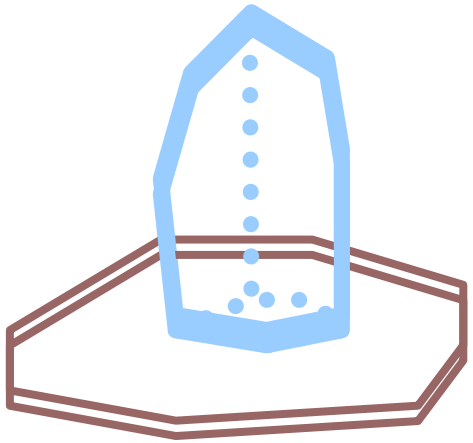


隐式曲面的多边形逼近

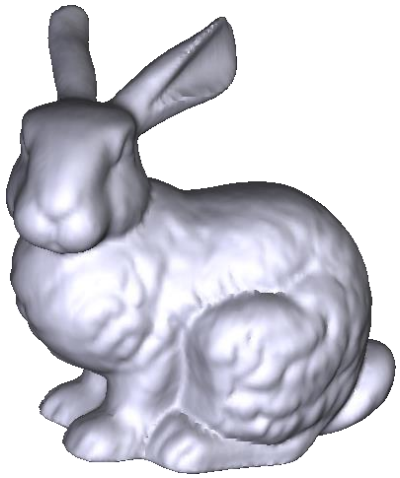
几何物体的多边形表示及其半边结构

- 几何物体的多边形表示
- 多边形表示的来源
- 多边形表示与2-流形
- 多边形表示的半边数据结构
- 多边形表示的不足

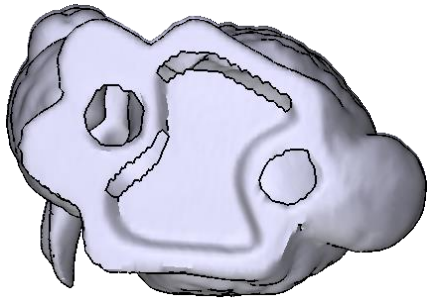
多边形与流形(Manifold)



非流形



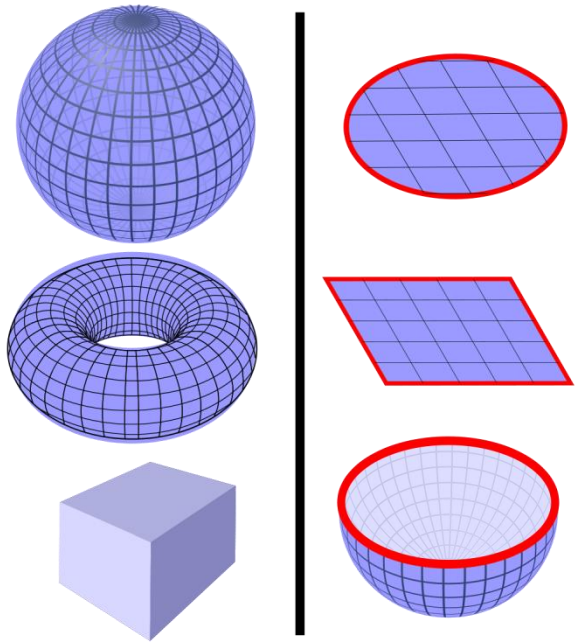
闭流形

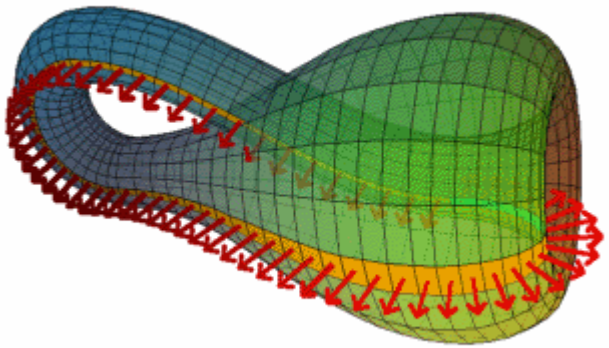


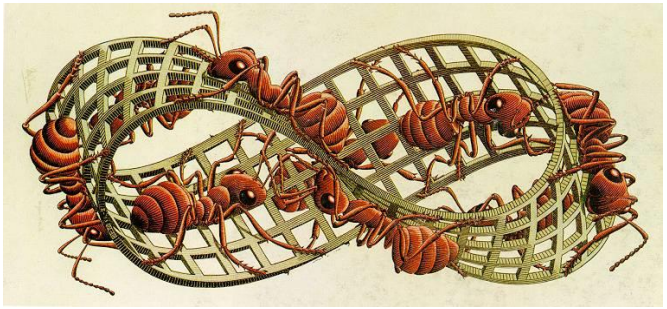
开流形

2-流形的定义和几何直观

- 2-流形上的每一点处，都存在一个位于流形内的任意小开邻域，该邻域拓扑同胚于平面上的开圆盘 (在边界处，该邻域拓扑同胚于半圆盘)

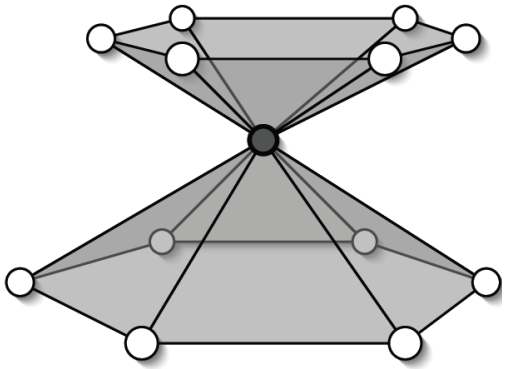


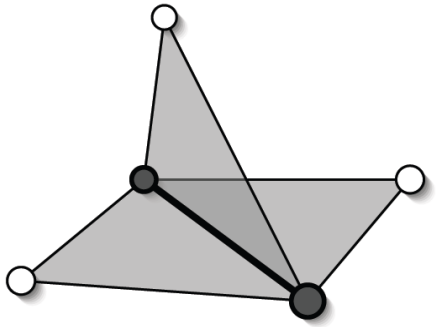


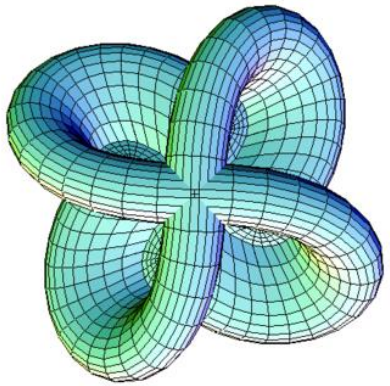


2-流形的定义和几何直观

- 非流形顶点：两个流形交于一点
- 非流形边：一条边与三个以上的面相交
- 自相交多边形







几何物体的多边形表示及其半边结构

- 几何物体的多边形表示
- 多边形表示的来源
- 多边形表示与2-流形
- 多边形表示的半边数据结构
- 多边形表示的不足

多边形物体表示方法：OBJ格式

- 由Wavefront Technologies首先提出的一个开放式的几何物体文件格式
- 支持：多边形、自由曲线/曲面、基本元素组合、显示和绘制属性
- 没有尺寸，可以在注释处添加缩放因子
- 其它文件格式

https://en.wikipedia.org/wiki/Polygon_mesh

多边形物体表示方法：OBJ格式

- 顶点坐标表(x,y,z)：每个顶点涉及多个面片，顶点数小于面片数(大约1/2)。鸭子模型中含有3474个顶点
- 纹理坐标表(u,v)：控制纹理在表面上的位置。鸭子的身体、脚趾、眼睛和嘴具有不同的颜色
- 法向表 (n_x,n_y,n_z)：可以控制物体绘制时的光滑程度。
  - (面片)法向：绘制出来的多边形物体棱角分明，见图(b)
  - (顶点)法向：则绘制出来的多边形物体是光滑的，见图(c)
- 面表：由指向顶点(及对应的纹理坐标和法向)指针组成，逆时针方向为外法向。鸭子模型含有6656个面

一个简单的多边形物体文件格式

顶点坐标表	$v_i=(x_i, y_i, z_i)$ $i=1,2,\ldots,$顶点数目
纹理坐标表	$vt_p=(u_p, v_p)$ $p=1,2,\ldots$纹理坐标数目
法向表	$vn_a=(nx_a, ny_a, nz_a)$ $a=1,2,\ldots$法向数目
面表	$f_s=(\ldots,v_i/vt_p/vn_a, v_j/vt_q/vn_b, v_k/vt_r/vn_c,\ldots)$ $s=1,2,\ldots,i,j,k,\ldots$面片数

- 详细信息见课件中相关文件

设计高效的网格曲面数据结构

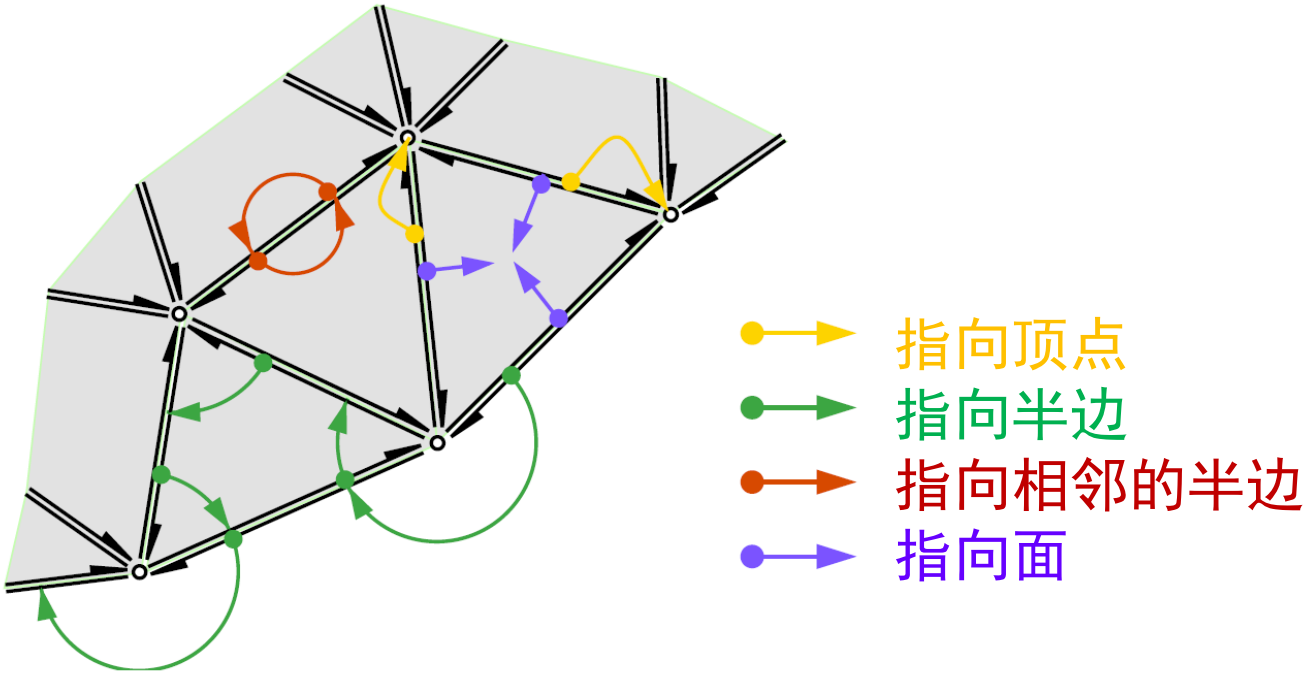
- 存储空间的考虑
- 拓扑上的考虑
  - 闭流形或开流形？
  - 三角网格曲面或任意多边形曲面？
  - 是否表示非流形曲面？
  - 层次结构或者单一分辨率的曲面？

设计高效的网格曲面数据结构

- 算法上的考虑：数据结构所适用的算法
  - 绘制网格曲面
  - 几何形状编辑
  - 拓扑连接关系的改变
  - 在顶点、边、面上附着其它信息
  - 邻接关系的查询：顶点的邻边、邻面？边的顶点、邻接面？面的顶点、边、相邻的面…
  - 曲面是否可定向的？

网格曲面数据结构：半边结构

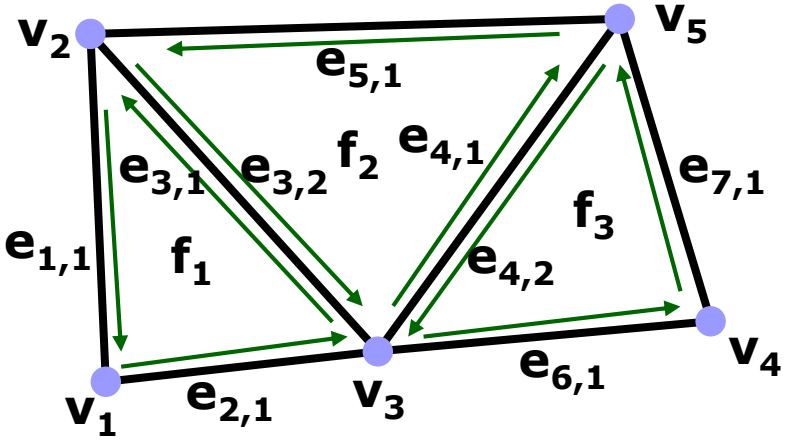
- 半边结构(Half-Edge Structure)：可定向的二维流形及其子集



半边结构

- 每条边被记为两条半边，每条半边记录：
  - 起始顶点的指针
  - 邻接面的指针(如果为边界，指针为NULL)
  - 下一条半边(逆时针方向)
  - 相邻的半边
  - 前一条半边(可选)
- 面：记录任一条边的一条半边
- 顶点
  - 坐标值
  - 指向以此顶点为起始端点的半边

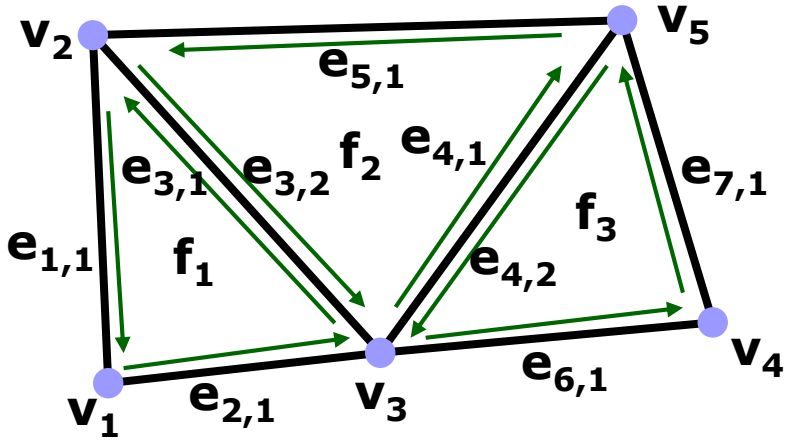
半边结构的实例



顶点	坐标	以此为起点的半边
v_1	(x_1,y_1,z_1)	$e_{2,1}$
v_2	(x_2,y_2,z_2)	$e_{1,1}$
v_3	(x_3,y_3,z_3)	$e_{4,1}$
v_4	(x_4,y_4,z_4)	$e_{7,1}$
v_5	(x_5,y_5,z_5)	$e_{5,1}$

面	半边
f_1	$e_{1,1}$
f_2	$e_{3,2}$
f_3	$e_{4,2}$

半边结构的实例



半边	起点	相邻半边	面	下条半边	前条半边
$e_{3,1}$	v_3	$e_{3,2}$	f_1	$e_{1,1}$	$e_{2,1}$
$e_{3,2}$	v_2	$e_{3,1}$	f_2	$e_{4,1}$	$e_{5,1}$
$e_{4,1}$	v_3	$e_{4,2}$	f_2	$e_{5,1}$	$e_{3,2}$
$e_{4,2}$	v_5	$e_{4,1}$	f_3	$e_{6,1}$	$e_{7,1}$

关于半边结构

- 优点
  - 查询时间 O(1)
  - 操作时间 (通常) O(1)
- 缺点
  - 只能表示可定向流形
  - 信息冗余

关于半边结构

- CGAL (Computational Geometry Algorithms Library)

  http://www.cgal.org

- OpenMesh (RWTH-Aachen University)

  http://www.openmesh.org

  https://www.graphics.rwth-aachen.de/software/openmesh/

- trimesh2 (gfx @ Princeton University)

  https://gfx.cs.princeton.edu/proj/trimesh2/

物体的多边形表示及数据结构

- 几何物体的多边形表示
- 多边形表示的来源
- 多边形表示与2-流形
- 多边形表示的半边数据结构
- 多边形表示的不足

物体多边形表示的不足之处

- 线性逼近，难以满足模型放大需求
- 外形修改/编辑困难
- 几何属性的解析计算困难
- 在表示复杂拓扑和具有丰富细节的物体时，数据规模庞大，对于建模、编辑、绘制、存储是一个巨大的负担

数字几何处理

物体的多边形表示及数据结构

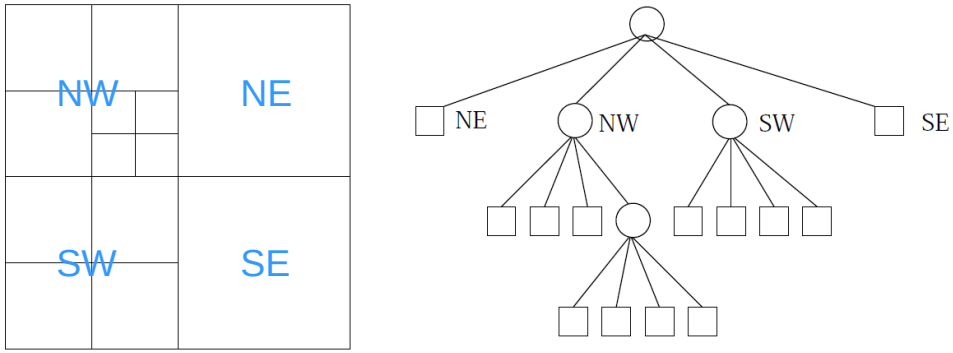
- 世界坐标系和景物(局部)坐标系
- 物体的多边形表示及其半边结构
- 图形学中常用的加速数据结构

图形学中常用的加速数据结构

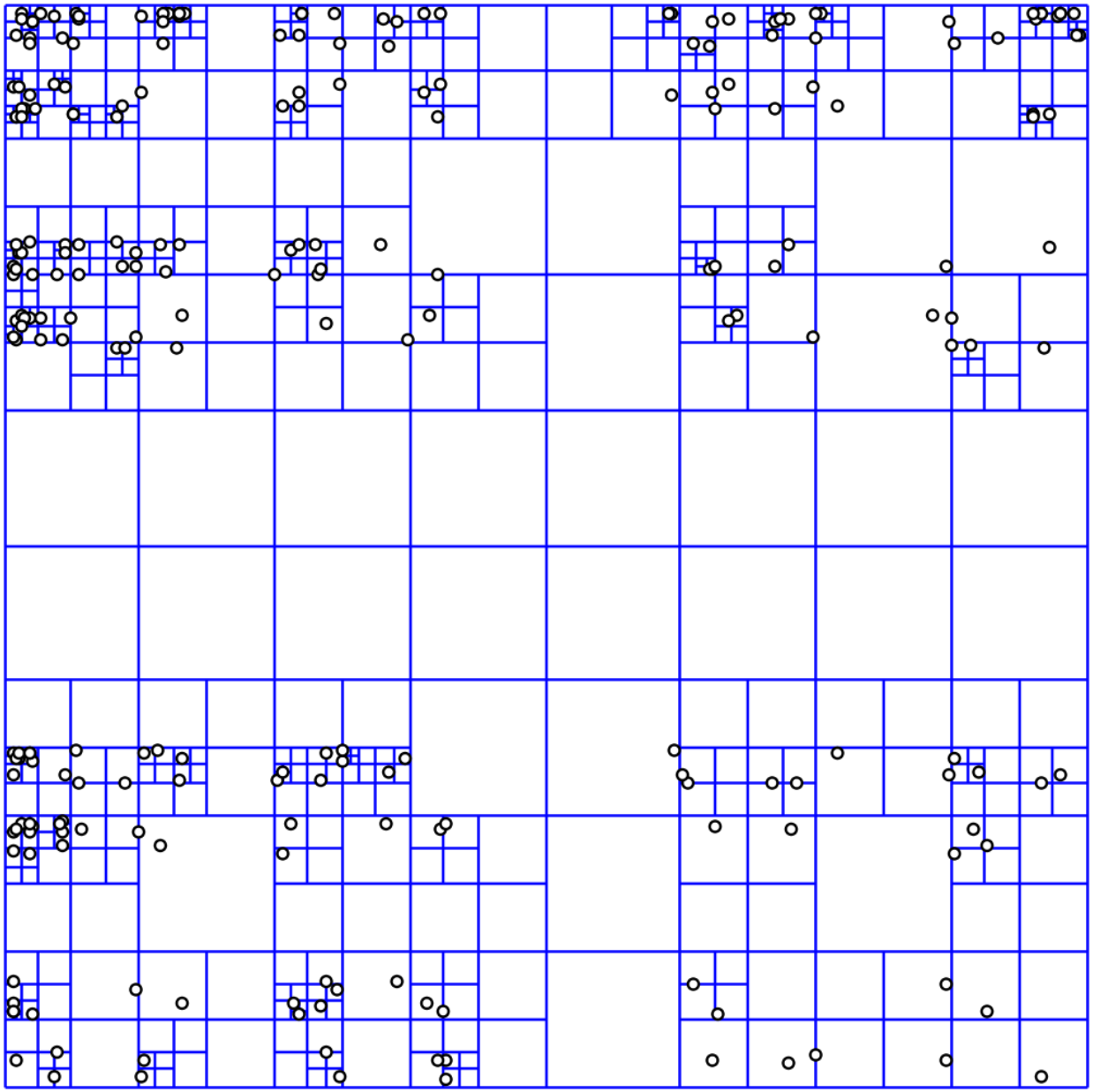
- 用规则空间代理作为模型或部件的索引，加速排序和查找。
  - 四叉树与八叉树(Quadtree & Octree)
  - k-d-树(k-d-tree)
  - 二叉空间剖分树(BSP tree, Binary Space Partition tree)
  - 层次包围盒(BVH, Bounding Volume Hierarchies)

四叉树(Quadtree)

- 定义：具有根节点的树，每一个内部节点具有四个子节点。每一个节点对应一个矩形。
- 在图形学中，四叉树可存储、表示、快速查询各种几何对象



一个四叉树例子



四叉树构造(以平面点集为例)

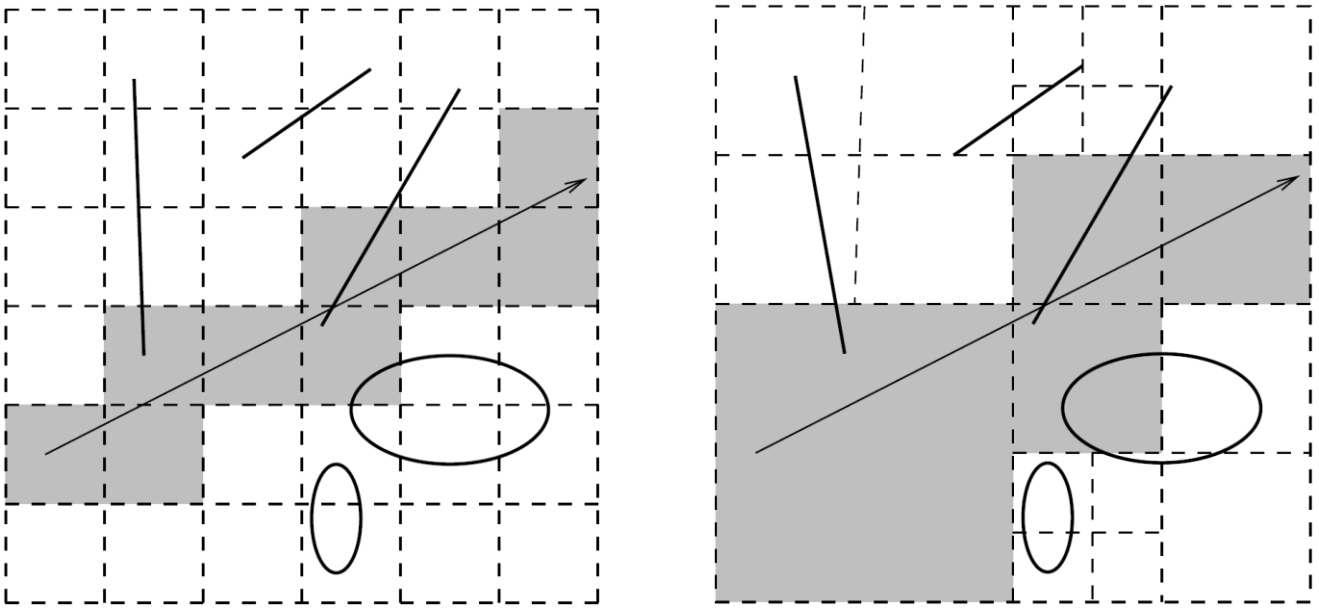
- 给定：平面点集P和初始正方形 Q=$[x_1, x_2]\times[y_1, y_2]$
- 目标：每个叶节点包含一个点
- 四叉树构造算法：
  - 如果P中只有一个点，则四叉树只有一个叶节点，存储Q和P；
  - 否则进行递归剖分：记$x_{mid}=(x_1+x_2)/2$、 $y_{mid}=(y_1+y_2)/2$；四个子节点正方形Q_{NE}、Q_{NW}、Q_{SE}、Q_{SW}；对应的四个子点集为

$$
\begin{aligned}
P_{NE} &:= \{p \in P : p_x > x_{mid} \wedge p_y > y_{mid}\},\\
P_{NW} &:= \{p \in P : p_x \le x_{mid} \wedge p_y > y_{mid}\},\\
P_{SW} &:= \{p \in P : p_x \le x_{mid} \wedge p_y \le y_{mid}\},\\
P_{SE} &:= \{p \in P : p_x > x_{mid} \wedge p_y \le y_{mid}\}.
\end{aligned}
$$

四叉树性质

- 四叉树深度：平面点集P对应的四叉树深度至多为：log(s/c)+3/2，s为初始正方形边长，c为点集中任意两点距离的最小值
- 四叉树空间复杂度和构造时间复杂度：一个深度为d、存储有n个点的四叉树，其节点数目为O((d+1)n)，构造时间为O((d+1)n)
- 邻近节点搜索复杂度：一个深度为d的四叉树T，搜索节点v的给定方向的邻近节点的时间为O(d+1)

Quadtree(Octree)的应用举例

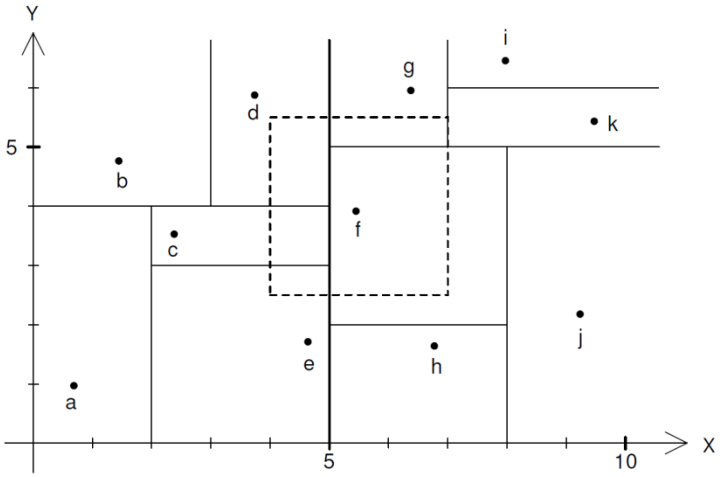


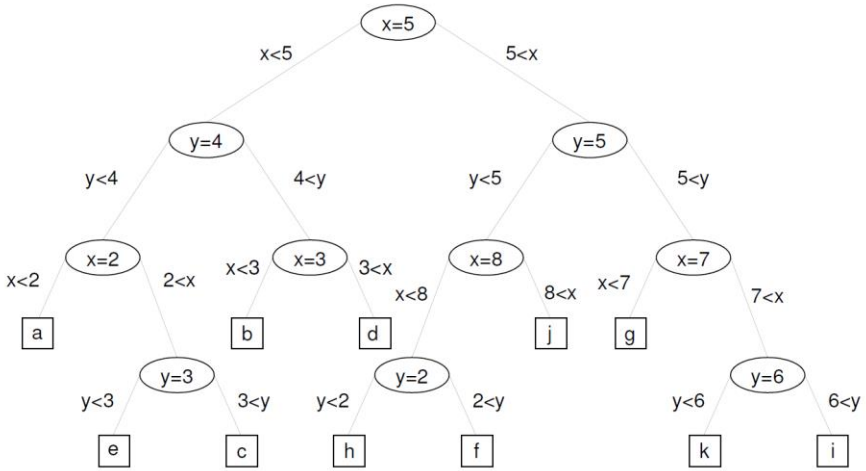
通过对场景中的物体进行(/八叉树)剖分，从而快速进行光线与物体的求交测试 (3D-DDA Traversal)

kd-树(k-d-tree, kd-tree)

- 定义：用于组织k维空间中点集的空间剖分数据结构(二叉树)。是一种特殊的二叉空间剖分树(binary space partitioning trees)
- 建立多维数据搜索关键字，用于空间范围快速查询、最近邻快速查询

kd-树实例





平面点集的kd-树(k=2)及其矩形区域搜索：中间节点对应剖分线，叶节点对应于平面上的点。建树目标为每个叶节点含有一个点。

kd-树构造(以平面点集为例)

- 给定：平面点集D (k=2)，假定所有点的X-坐标不同、Y-坐标也不相同
- 目标：构造kd-树，使得每个叶节点包含一个点
- kd-树构造算法：
  - 首先搜索一个X-坐标分割值 X=s；根据剖分线X=s，将点集D分为两个子集：

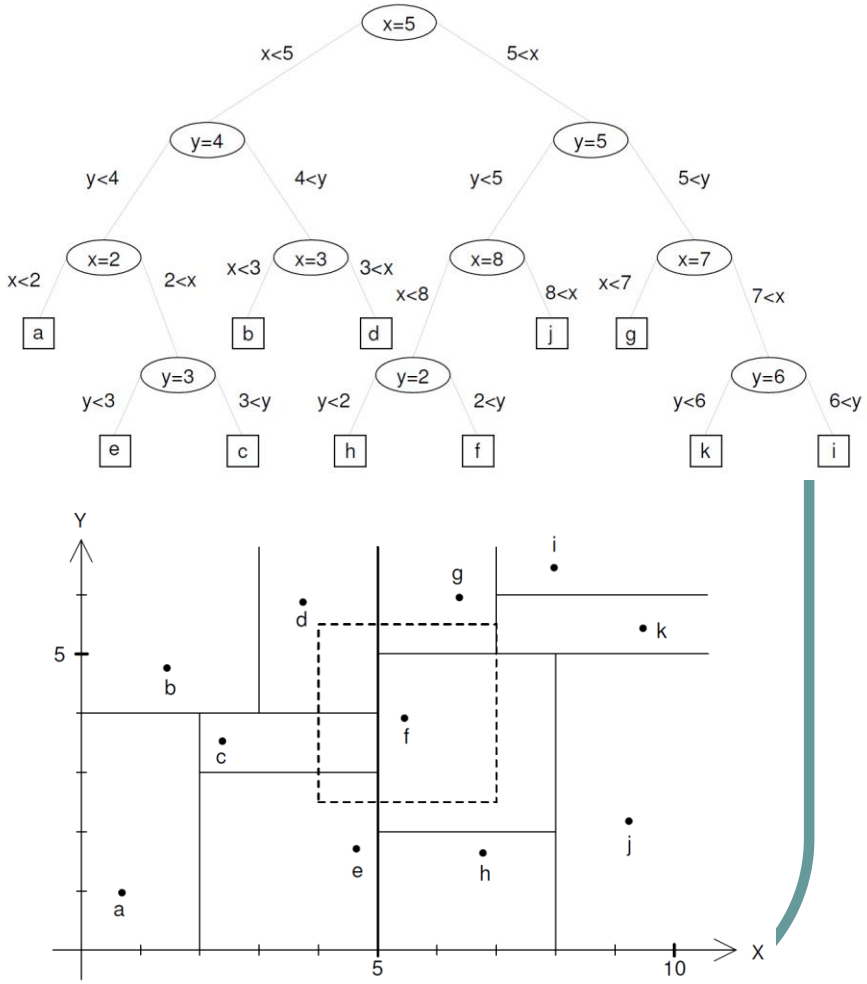
$$D_{<s} = \{(x,y) \in D; x < s\} = D \cap \{X < s\}$$
$$D_{>s} = \{(x,y) \in D; x > s\} = D \cap \{X > s\}.$$

  - 然后，对于两个子集$D_{<s}$和 $D_{>s}$，分别选择剖分线$Y=t_1$和$Y=t_2$
  - 递归执行上述步骤，直至每一个叶节点包含一个点

kd-树的性质

- 每个中间节点对应一个剖分线(/超平面)，叶节点对应一个点(/几何元素)

- 每个节点对应一个矩形R(v)(/超长方体)：即该节点回朔至根节点的剖分线(/超平面)所剖分的半平面(/半空间)的交集



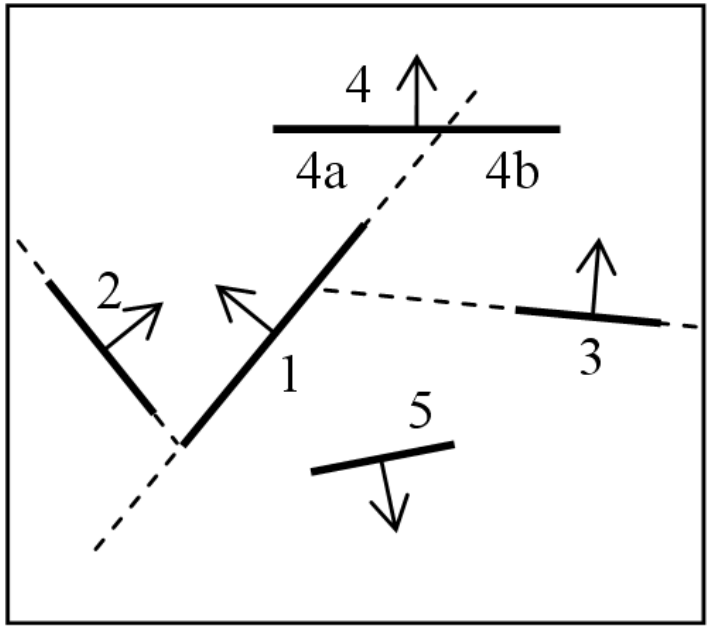
kd-树的性质

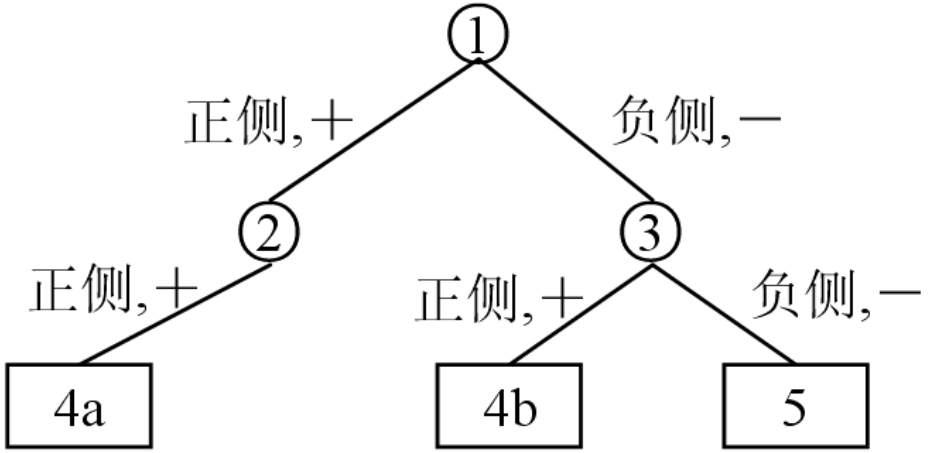
- 给定平行于坐标轴的矩形Q，搜索位于Q中的点：
  - 计算所有节点v∈Q的矩形R(v)，如果R(v)∩Q≠Φ；
  - 判断位于上述非空的R(v)中点是否位于Q中；
- 给定平面上*n*个点的点集，构造一棵平衡kd-树的时间复杂度为O(nlogn)、空间复杂度为O(n)，搜索平行于坐标轴的矩形中的点的复杂度为O(a+$\sqrt{n}$)，其中a为结果的点的个数

二叉空间剖分树

- 二叉空间剖分树： BSP tree, Binary Space Partition tree
- kd-树的推广
  - kd-树的分割面：坐标平面
  - BSP-树的分割面：任意平面
- 应用于景物空间的物体排序与消隐

二叉空间剖分树



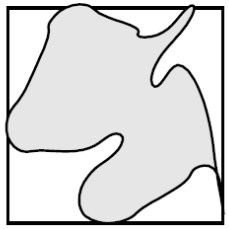


层次包围盒

- 层次包围盒： Bounding Volume Hierarchies，简称BVH

- 与上述层次数据结构的区别
  - 四叉树(八叉树)、kd-数、BSP树：面向**空间**剖分
  - 层次包围盒：面向**几何物体**的剖分
    - 避免对所有物体的过渡剖分

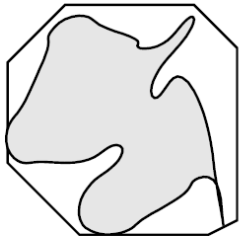
常见包围盒的类型



AABB

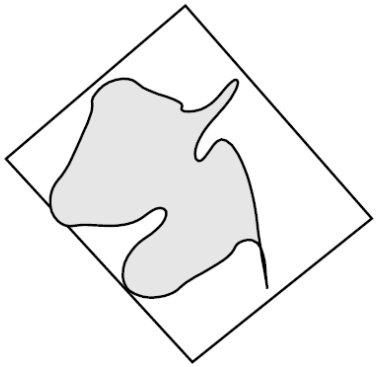
Axis-Aligned Bounding Box

sphere



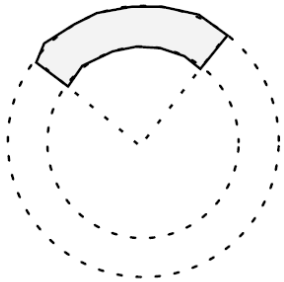
DOP

Discrete Oriented Polytope

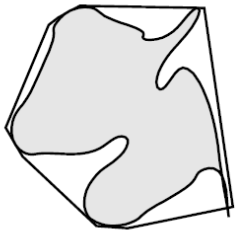


OBB

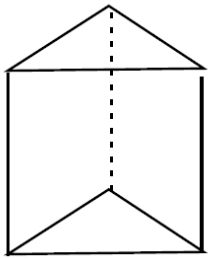
Oriented Bounding Box



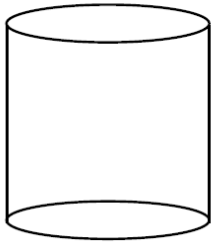
spherical shell



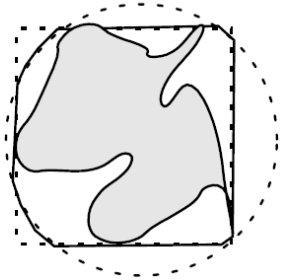
convex hull



prism

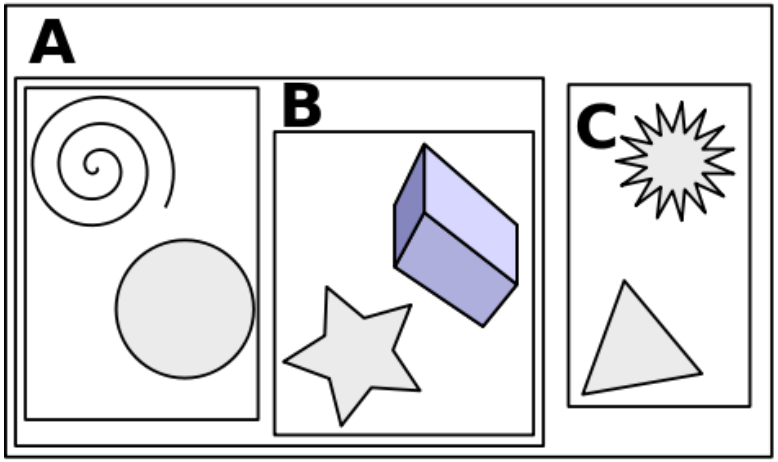


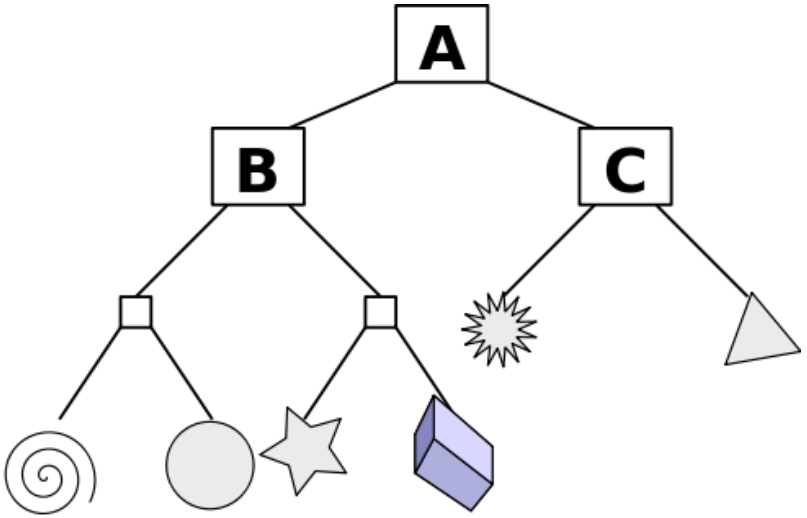
cylinder



intersection of other BVs

层次包围盒举例





层次包围盒

- 定义：记$O=\{o_1, \ldots, o_n\}$为一个几何物体集合，关于O的层次包围盒BVH(O) 为
  1. 如果$|O|=e$，则BVH(O)为一个叶节点，存储了集合O和其包围盒BV；
  2. 如果$|O|>e$，则BVH(O)为根节点为v、具有n(v)个子节点$v_1,\ldots,v_{n(v)}$的树。每一个子节点v_i是物体子集合$O_i \subset O$的一个层次包围盒$BVH(O_i)$，其中$\cup O_i=O$，v还存储了物体集合O的包围盒BV

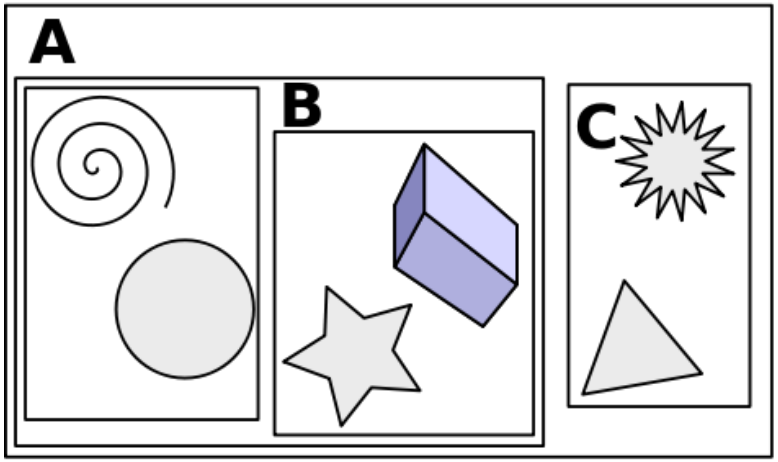
层次包围盒

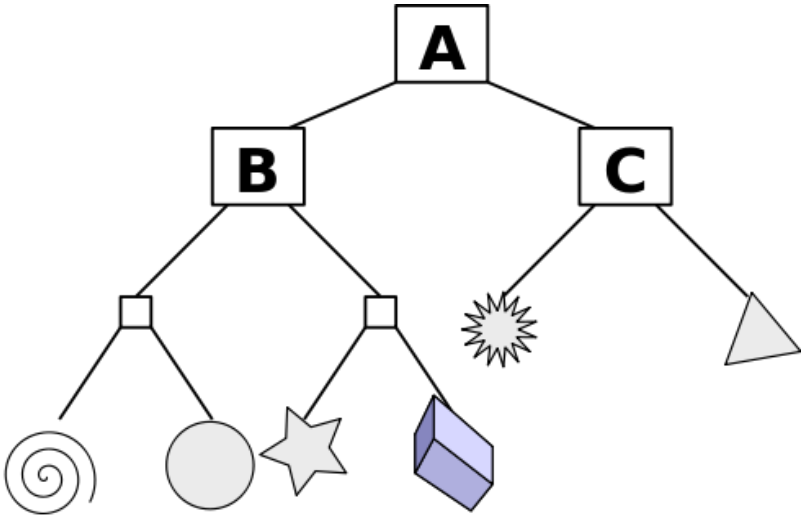
- 关于BVH定义中的两个参数
  - e：可以取1，也根据应用取较大的数。
    - 例如在排序中，如果集合中的物体较少，直接取所有物体的包围盒进行操作。递归操作存在代价
  - 子节点数目n(v)：大部分情况是二叉树，也可以是多叉树。不同层次中的节点数可以不是常数。
    - 算法实现：常数目节点可以简化处理
  - e和n(v)：线性搜索/操作与递归建树之间平衡

层次包围盒

- BVH设计原则：
  - 同一个几何物体尽可能属于一个节点(根据应用不同，也可能属于多个节点)
  - 尽可能将物体集合分成相互分离的子集
  - 对每一个节点选择恰当的包围盒。通常是下属性质之间的平衡：
    - 紧致性(tightness)
    - 内存消耗
    - 进行查询时的代价

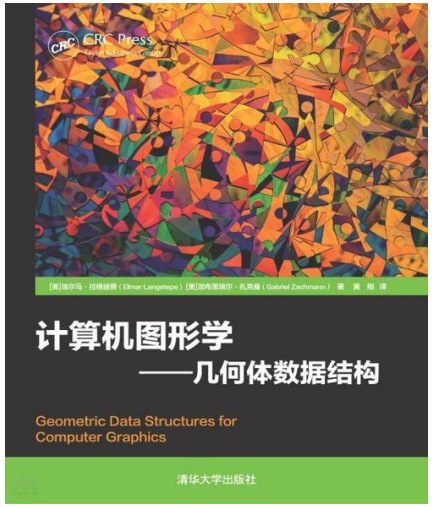
层次包围盒举例





相关参考资料

- wavefront OBJ 文件格式说明
- Geometric Data Structures for Computer Graphics (Siggraph 2003 Tutorial 16)
- Polygon Mesh、K-D Tree、BVH from Wiki



作者: Elmar Langetepe/Gabriel Zachmann
出版社: 清华大学出版社
原作名: Geometric Data Structures for Computer Graphics
译者: 黄刚
出版年: 2019-8-1
ISBN: 9787302527930