

華東理工大學

计算机图形学

2023年10月

奉贤校区



华东理工大学



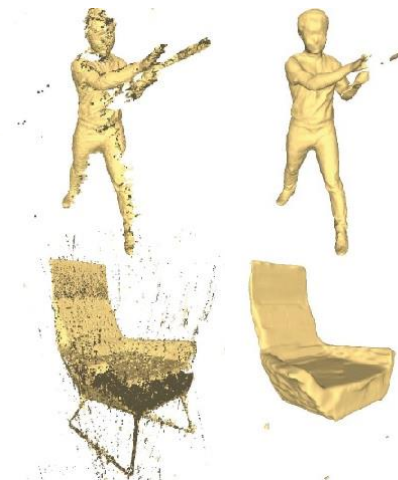
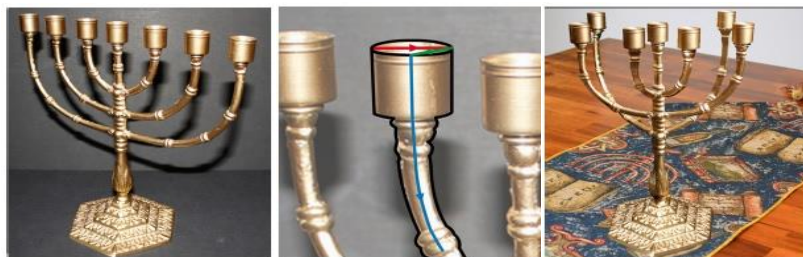
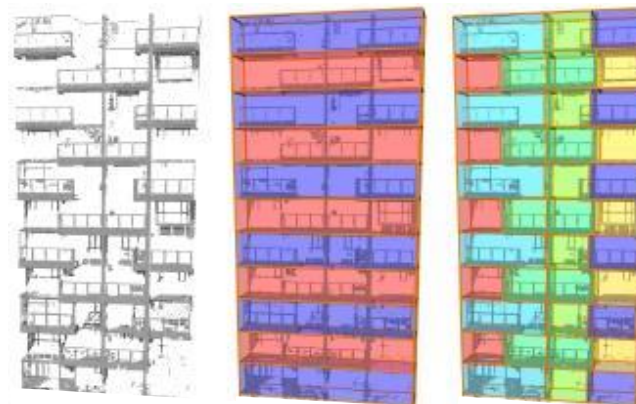
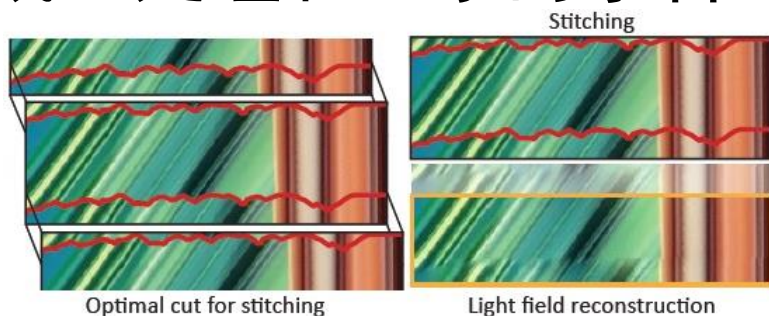
04

图形的表示与数据结构

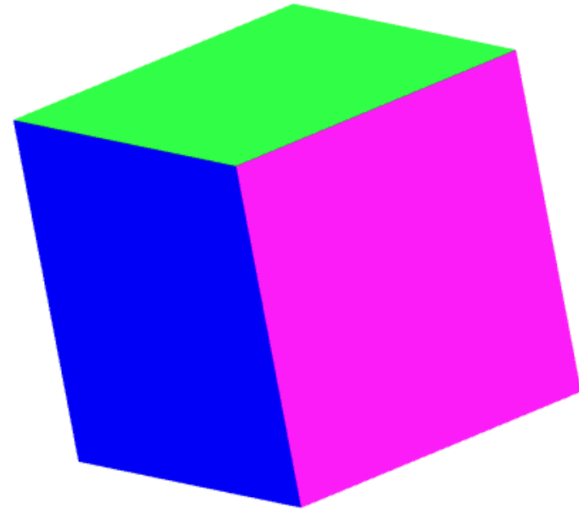
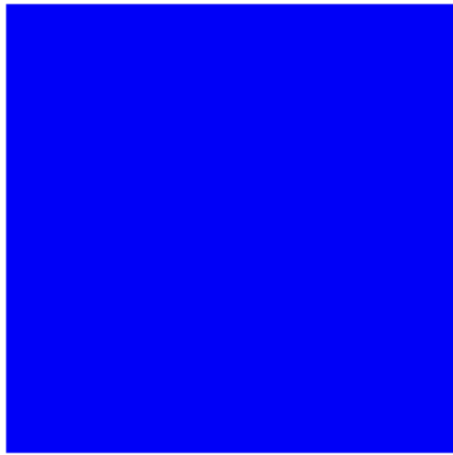
Representation and Data Structure of CG

Back to the beginning

- 计算机图形学是利用计算机研究图形的**表示**、生成、处理和显示的学科

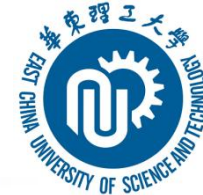


三维形体如何表示



- 如何记录这些物体？（边长为1）

思路:



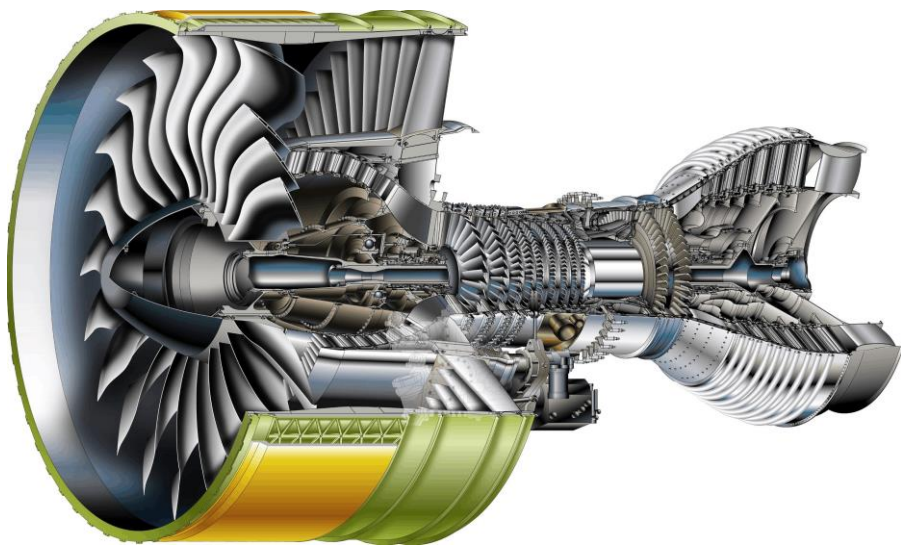
- 显示时直接表示，无数据结构，无持久性（无外部文件存储）

```
function quad(a, b, c, d){  
  var vertices = [  
    vec4(-0.5, -0.5, 0.5, 1.0),  
    vec4(-0.5, 0.5, 0.5, 1.0),  
    vec4(0.5, 0.5, 0.5, 1.0),  
    vec4(0.5, -0.5, 0.5, 1.0),  
    vec4(-0.5, -0.5, -0.5, 1.0),  
    vec4(-0.5, 0.5, -0.5, 1.0),  
    vec4(0.5, 0.5, -0.5, 1.0),  
    vec4(0.5, -0.5, -0.5, 1.0)  ];  
  var indices = [a, b, c, a, c, d];  
  for (var i = 0; i < indices.length; ++i)  
  {  
    positions.push(vertices[indices[i]]);  
  }  
}
```

```
function Cube()  
{  
  quad(1, 0, 3, 2);  
  quad(2, 3, 7, 6);  
  quad(3, 0, 4, 7);  
  quad(6, 5, 1, 2);  
  quad(4, 5, 6, 7);  
  quad(5, 4, 0, 1);  
}
```

静态场景（分类）

- 复杂工业零件表示

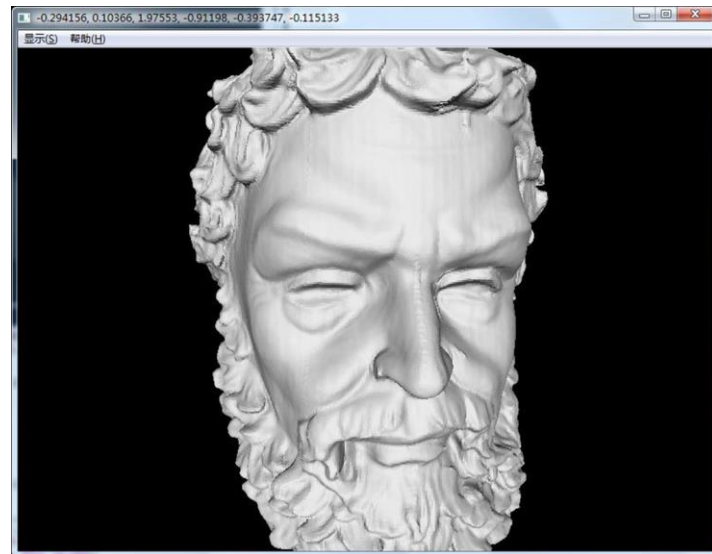


- 精度要求较低的图形

数据结构，持久性（外部文件存储）

静态场景（分类）

- 复杂工业零件表示
- 精度要求较低的图形



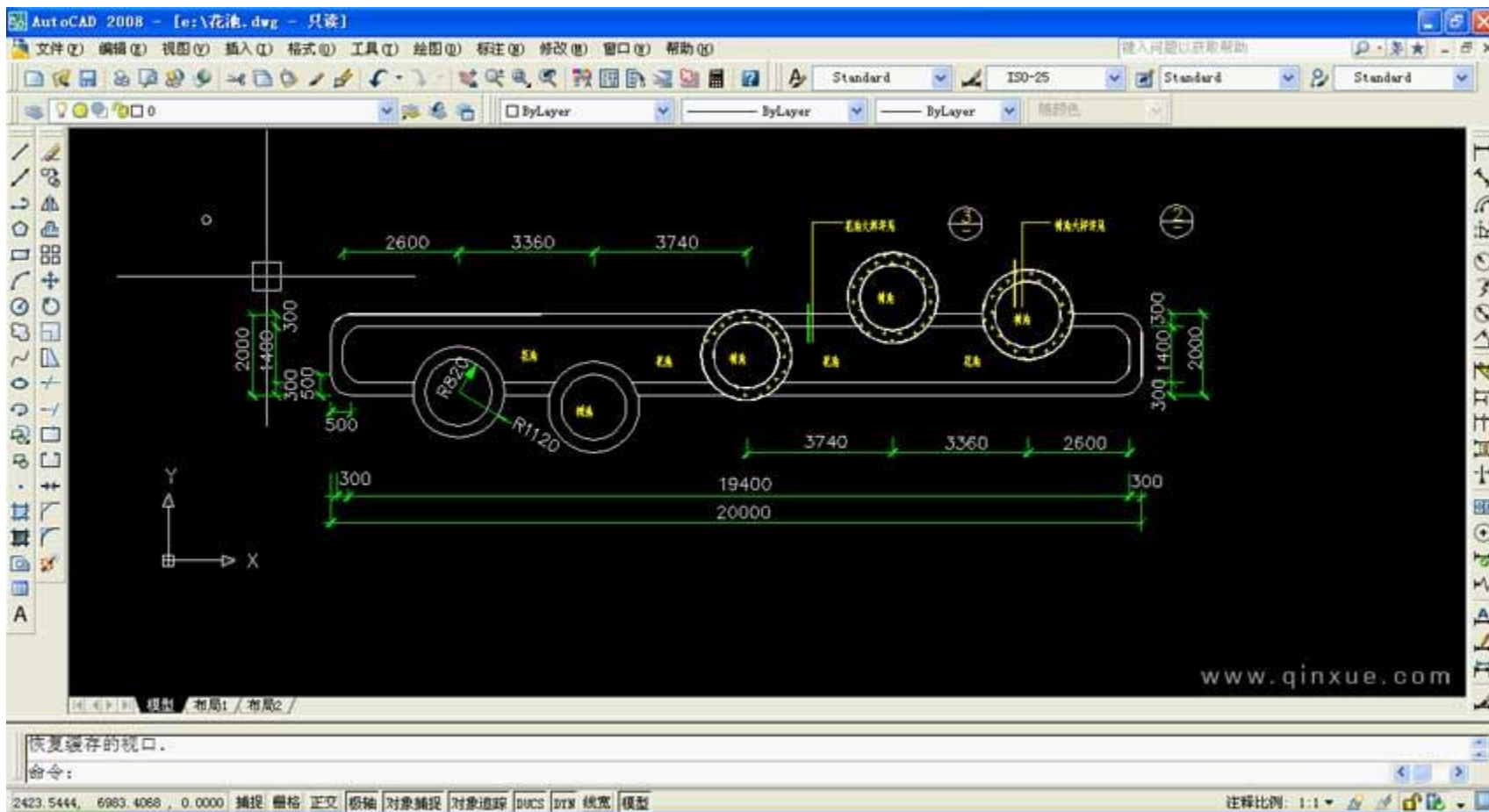
数据结构，持久性（外部文件存储）

动态场景的表示



不需要持久性，效果一样就行了

AutoCAD软件



- 从CAD二维版本说起

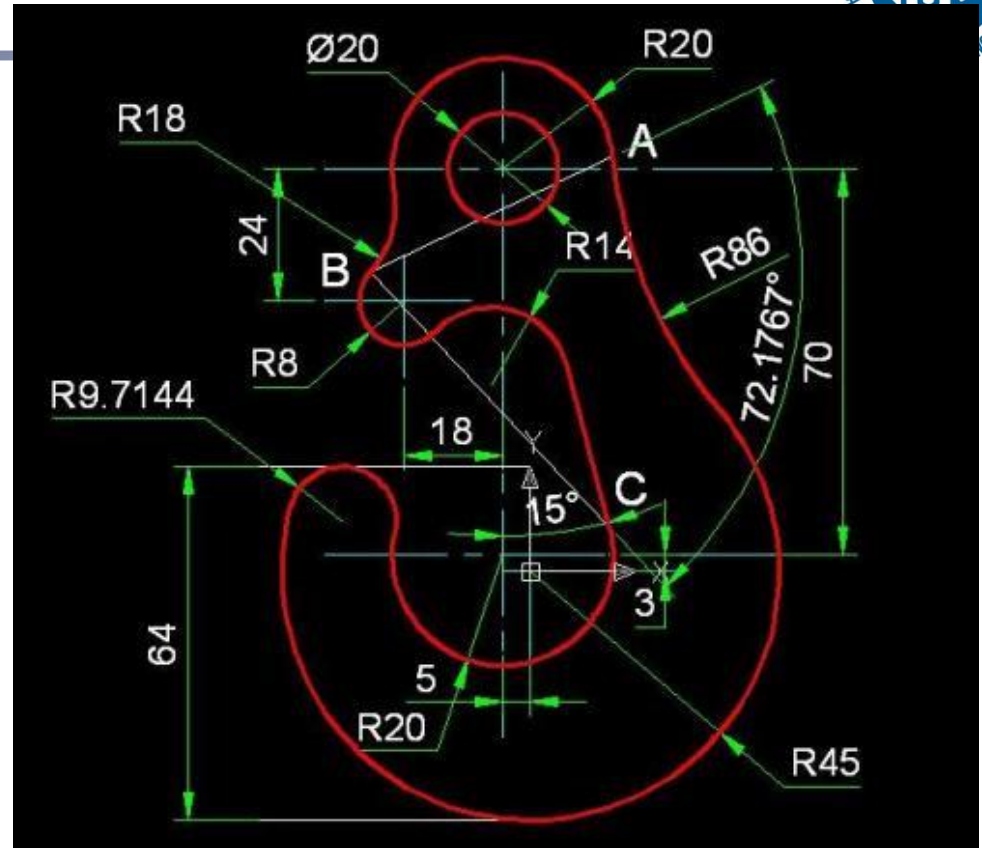
DXF被广泛使用，成为事实上的标准。绝大多数CAD系统都能读入或输出DXF文件。

DXF文件是由很多的“代码”和“值”组成的“数据对”构造而成，这里的代码称为“组码”（group code），指定其后的值的类型和用途。每个组码和值必须为单独的一行的。

DXF文件被组织成为多个“段”（section），每个段以组码“0”和字符串“SECTION”开头，紧接着是组码“2”和表示段名的字符串（如HEADER）。段的中间，可以使用组码和值定义段中的元素。段的结尾使用组码“0”和字符串“ENDSEC”来定义

DXF的组成

HEADER 段
CLASSES 段
TABLES 段
BLOCKS 段
ENTITIES 段
OBJECTS 段
THUMBNAILIMAGE 段



ENTITIES 段包含图形数据. 例如LINE 直线, ARC圆弧

读取基于DXF文件格式的图件数据

判断数据类型

文本

直线

多段
线

曲线

圆

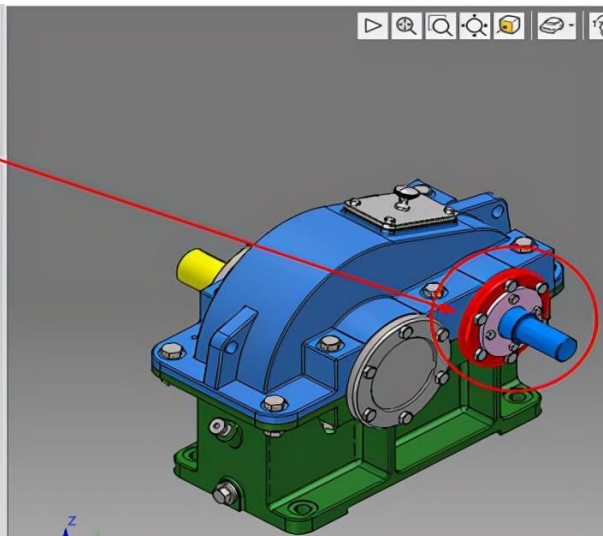
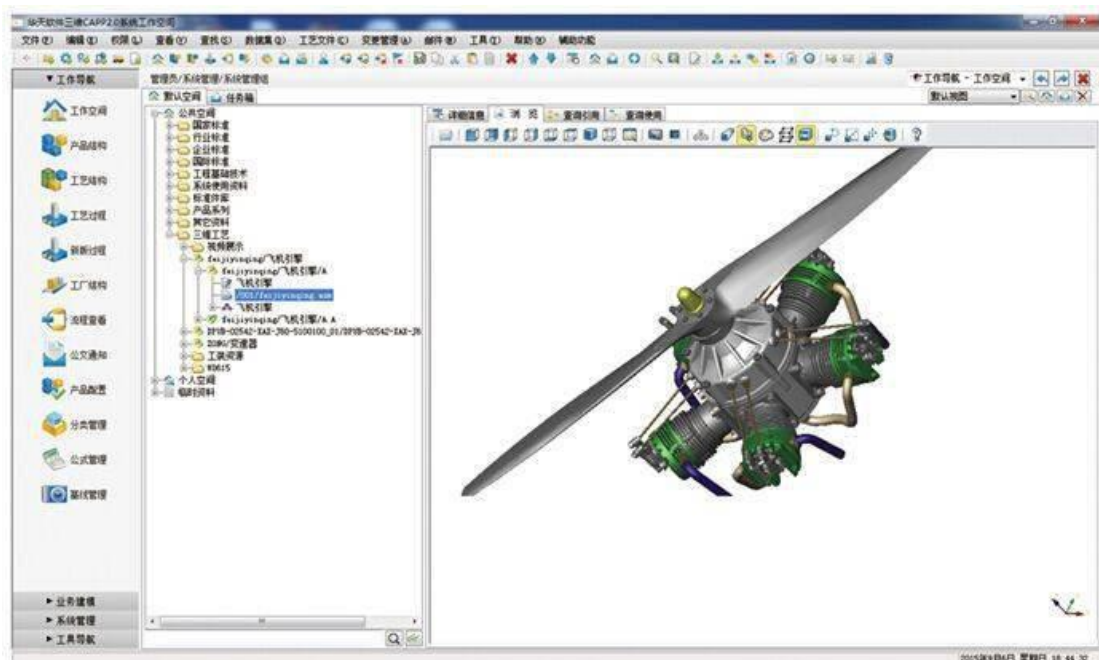
圆弧

椭圆

图块

填充

CAD软件



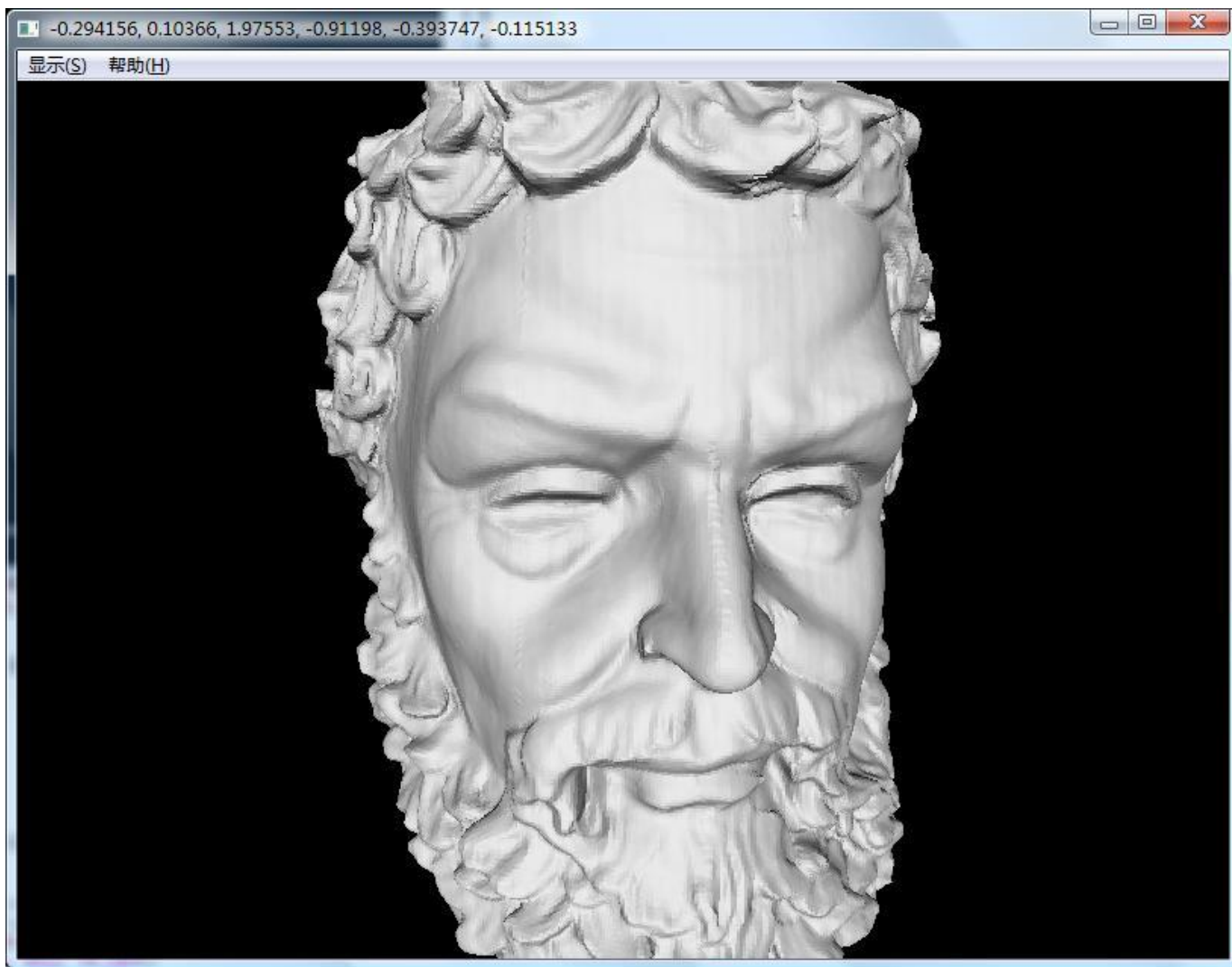
华天软件
中望
数码大方 (CAXA)

...

外形设计（犀牛等）



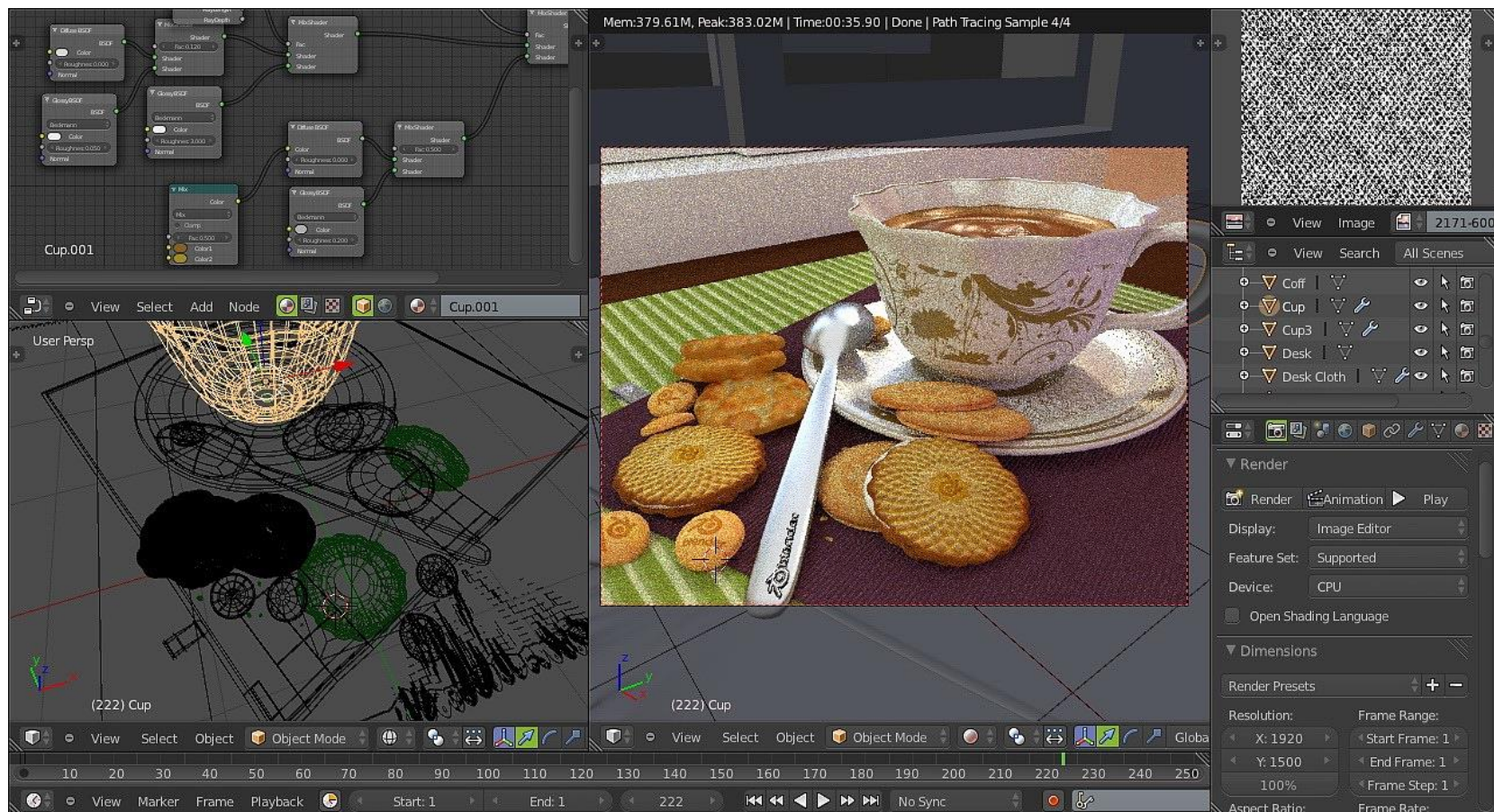
Mesh软件



Unity场景建模



Blender 软件



存在的问题

- 高精度的三维怎么表示？
- 三维表示的**唯一性**？ **真实性**？
- 如何表示三维的**内部**和**表面**？
- 只关注显示效果的表面如何快速表示？
- **非规则形状**如何表示？（动态的火焰、树叶、头发等等）



内容

- 基本概念
- 三维形体的表示
- 扩展内容： 网格表示

基本概念

1

造型技术

2

基本图形元素

3

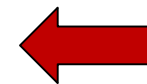
几何信息与拓扑信息

4

实体

5

正则集合运算



造型/建模技术 (Modeling)

- 把研究如何在计算机中建立恰当模型表示不同图形对象的技术称为**造型技术**。
- 有两类图形对象：
 - ✓规则对象：几何造型、几何模型。
 - ✓不规则对象：过程式模拟。

基本概念

1

造型技术

2

基本图形元素

3

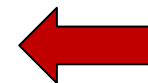
几何信息与拓扑信息

4

实体

5

正则集合运算



□**基本图形元素**：图素或图元、体素。

□**图素**是指可以用一定的几何参数和属性参数描述的最基本的图形输出元素。

□**段**是指具有逻辑意义的有限个图素（或体素）及其附加属性的集合。

基本概念

1

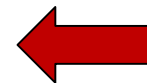
造型技术

2

基本图形元素

3

几何信息与拓扑信息



4

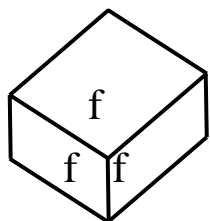
实体

5

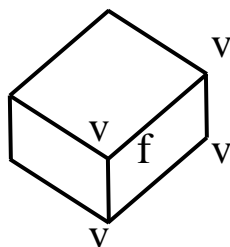
正则集合运算

几何信息与拓扑信息

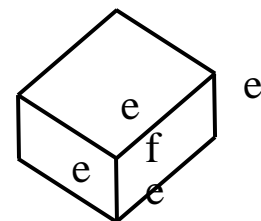
- 图形信息与非图形信息
 - **几何信息**：形体在欧氏空间中的位置和大小。
 - **拓扑信息**：形体各分量（点、边、面）的数目及其相互间的连接关系。



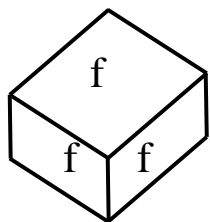
面相邻性 $f:\{f\}$



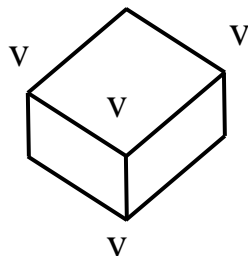
面-顶点包含性



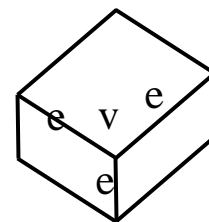
面-边包含性 $f:\{e\}$



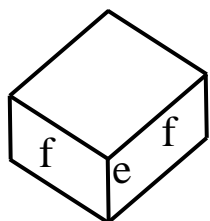
顶点-面相邻性



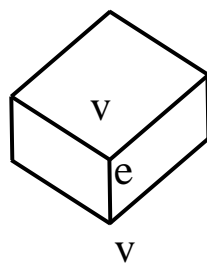
顶点相邻性



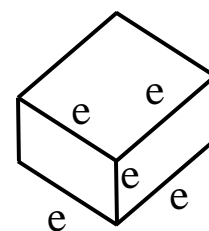
顶点-边相邻性 $v:\{e\}$



边-面相邻性 $e:\{f\}$



边-顶点包含性 $e:\{v\}$



边相邻性 $e:\{e\}$

图4.1 拓扑信息

几何信息与拓扑信息

- **刚体运动**：不改变图形上任意两点间的距离，也不改变图形的几何性质的运动。
- **拓扑运动**：允许形体作弹性运动，即在拓扑关系中，对图形可随意地伸张扭曲。但图上各个点仍为不同的点，决不允许把不同的点合并成一个点。

基本概念

1

造型技术

2

基本图形元素

3

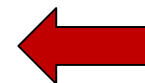
几何信息与拓扑信息

4

实体

5

正则集合运算



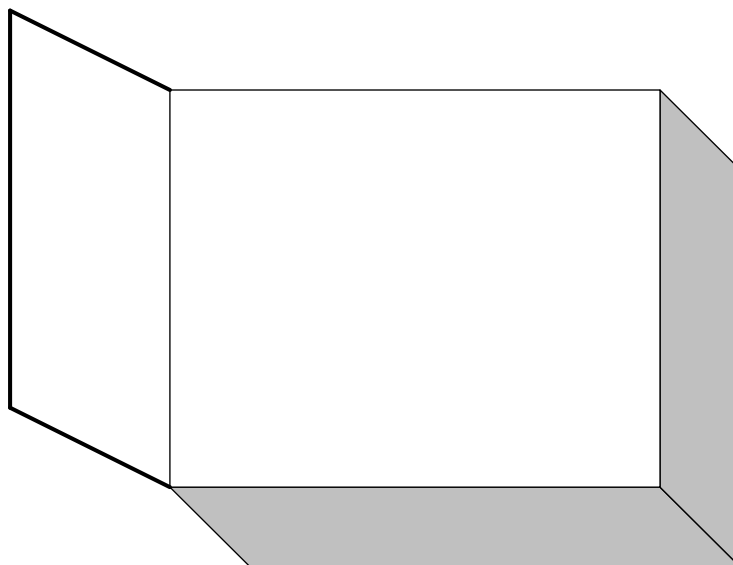


图4.2 带有悬挂边的立方体

实体

- **点的领域：** 如果 P 是点集 S 的一个元素，那么点 P 的以 R ($R>0$) 为半径的领域指的是围绕点 P 的半径为 R 的小球（二维情况下为小圆）。
- **开集的闭包：** 是指该开集与其所有边界点的集合并集，本身是一个闭集。
- **正则集：** 由内部点构成的点集的闭包就是正则集，三维空间的正则集就是正则形体。

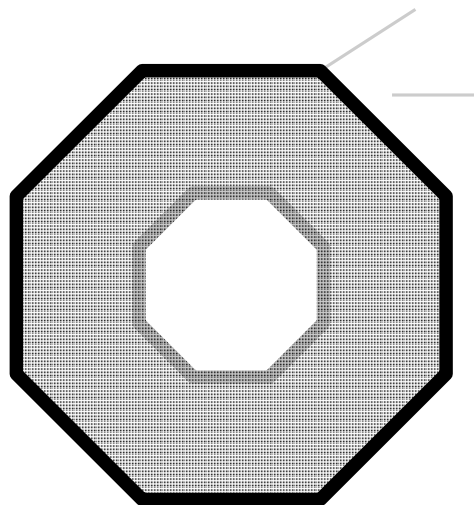
□ 组成三维物体的点的集合可以分为两类：

- **内点：**为点集中的这样一些点，它们具有完全包含于该点集的充分小的领域。
- **边界点：**不具备此性质的点集中的点。

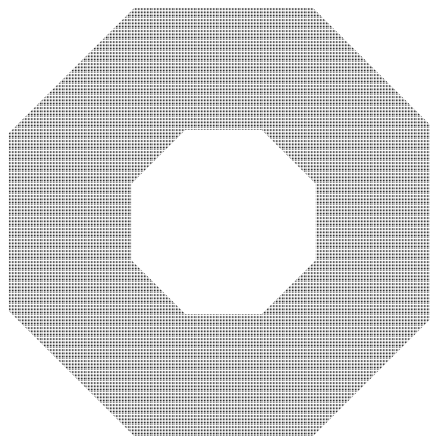
□ 定义点集的正则运算 r 运算为:

$$r \cdot A = c \cdot i \cdot A$$

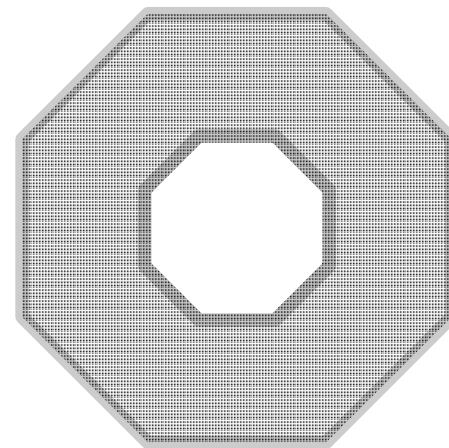
□ 正则运算即为先对物体取内点再取闭包的运算。
 $r \cdot A$ 称为 A 的正则集。



(a) 带有孤立点和边的
二维点集A



(b) 内点集合 $i \cdot A$



(c) 正则点集 $c \cdot i \cdot A$

图4.3 实体的例子

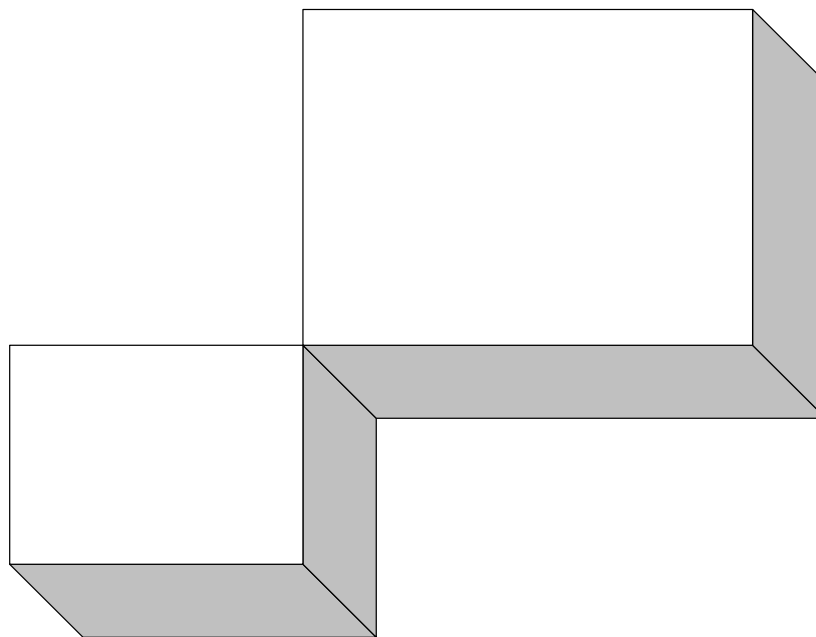
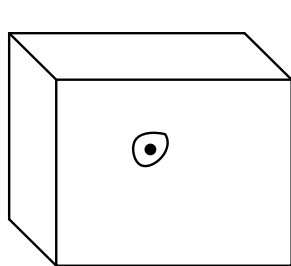


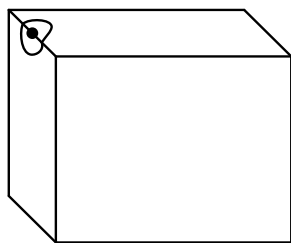
图4.4 正则形体

实体

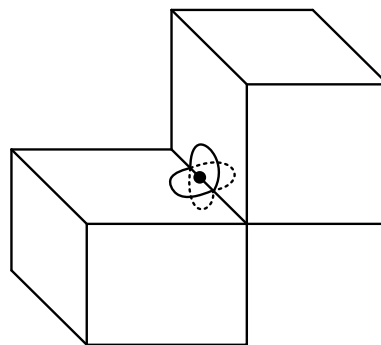
- **二维流形**指的是对于实体表面上的任意一点，都可以找到一个围绕着它的任意小的领域，该领域与平面上的一个圆盘是拓扑等价的。



(a) 二维流形



(b) 二维流形



(c) 非二维流形

图4.5 正则形体

- 实体：对于一个占据有限空间的正则形体，如果其表面是二维流形，则该正则形体为实体。

对应真实的物体。

基本概念

1

造型技术

2

基本图形元素

3

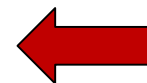
几何信息与拓扑信息

4

实体

5

正则集合运算



正则集合运算

- 有效实体的封闭性。
- 把能够产生正则形体的集合运算称为**正则集合运算**。

正则集合运算

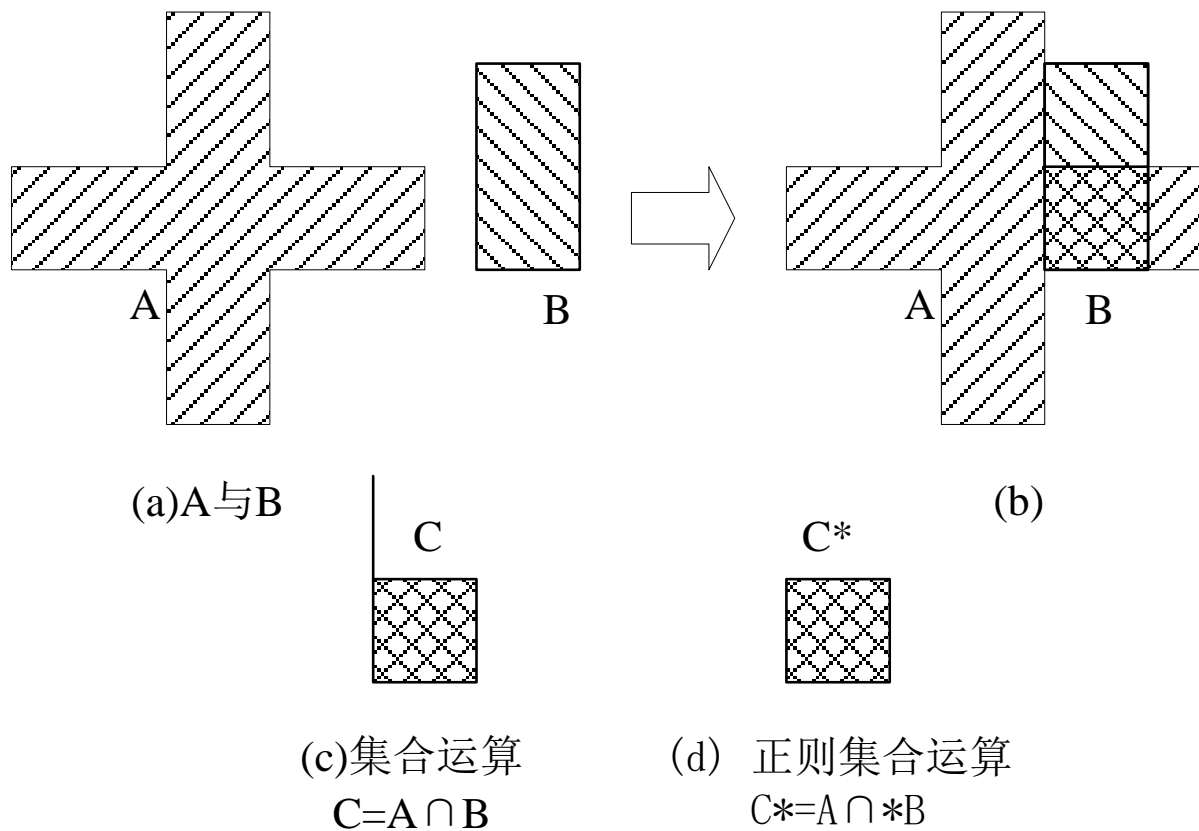


图4.6 集合运算与正则集合运算

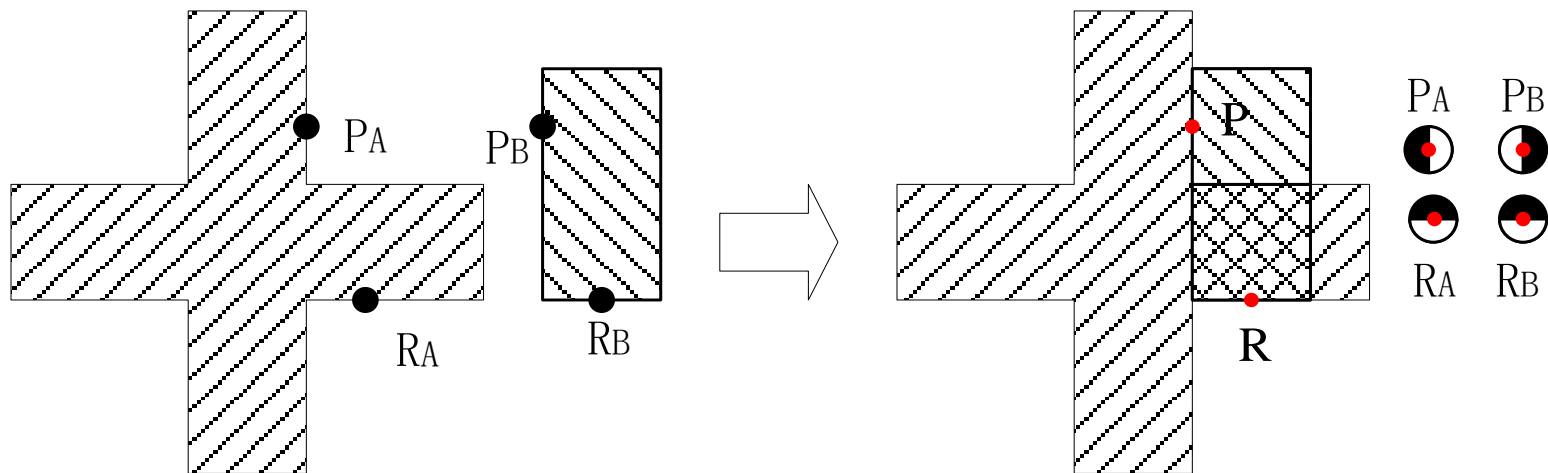


图4.7 基于点的领域概念生成正则形体

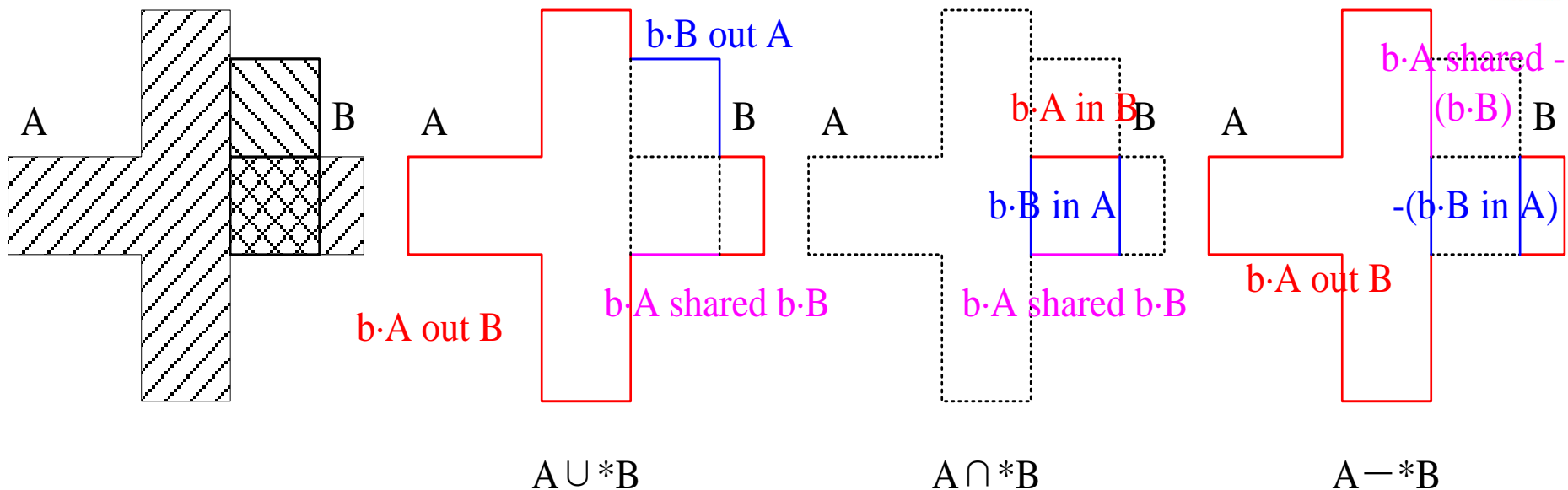


图4.8 正则集合运算 $A \cup^* B$, $A \cap^* B$, $A -^* B$ 的结果 (实线表示结果形体的边界)

$$b \cdot (A \cup^* B) = \{b \cdot A \text{ out } B, b \cdot B \text{ out } A, b \cdot A \text{ shared } b \cdot B\}$$

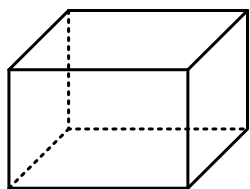
$$b \cdot (A \cap^* B) = \{b \cdot A \text{ in } B, b \cdot B \text{ in } A, b \cdot A \text{ shared } b \cdot B\}$$

$$b \cdot (A -^* B) = \{b \cdot A \text{ out } B, -(b \cdot B \text{ in } A), b \cdot A \text{ shared } -(b \cdot B)\}$$

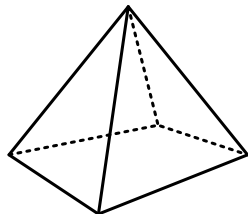
欧拉公式

- 欧拉公式证明简单多面体的顶点数 V 、边数 E 和面数 F 满足如下关系： **$V-E+F=2$** 。
- 非简单多面体需对欧拉公式加以扩展。令 H 表示多面体表面上孔的个数， G 表示贯穿多面体的孔的个数， C 表示独立的、不连接的多面体数，则扩展后的欧拉公式为： **$V-E+F-H=2 (C-G)$** 。

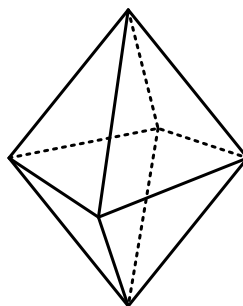
平面多面体与欧拉公式



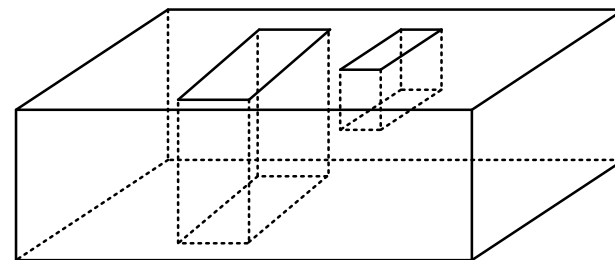
$$8 - 12 + 6 = 2$$



$$5 - 8 + 5 = 2$$



$$6 - 12 + 8 = 2$$



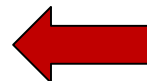
$$24 - 36 + 15 - 3 = 2 \times (1 - 1)$$

图4.9 平面多面体与欧拉公式

三维形体的表示

1

规则对象的表示方法



2

非规则对象的表示方法

3

常用的显示表示-网格



图形的层次结构

- **线框模型与实体模型（实体造型技术）**
- **可以将实体模型的表示大致分为三类：**
 - **边界表示（Boundary representation, B-reps）**
 - **构造实体几何表示**
 - **空间分割（Space-partitioning）表示**

三维形体的表示

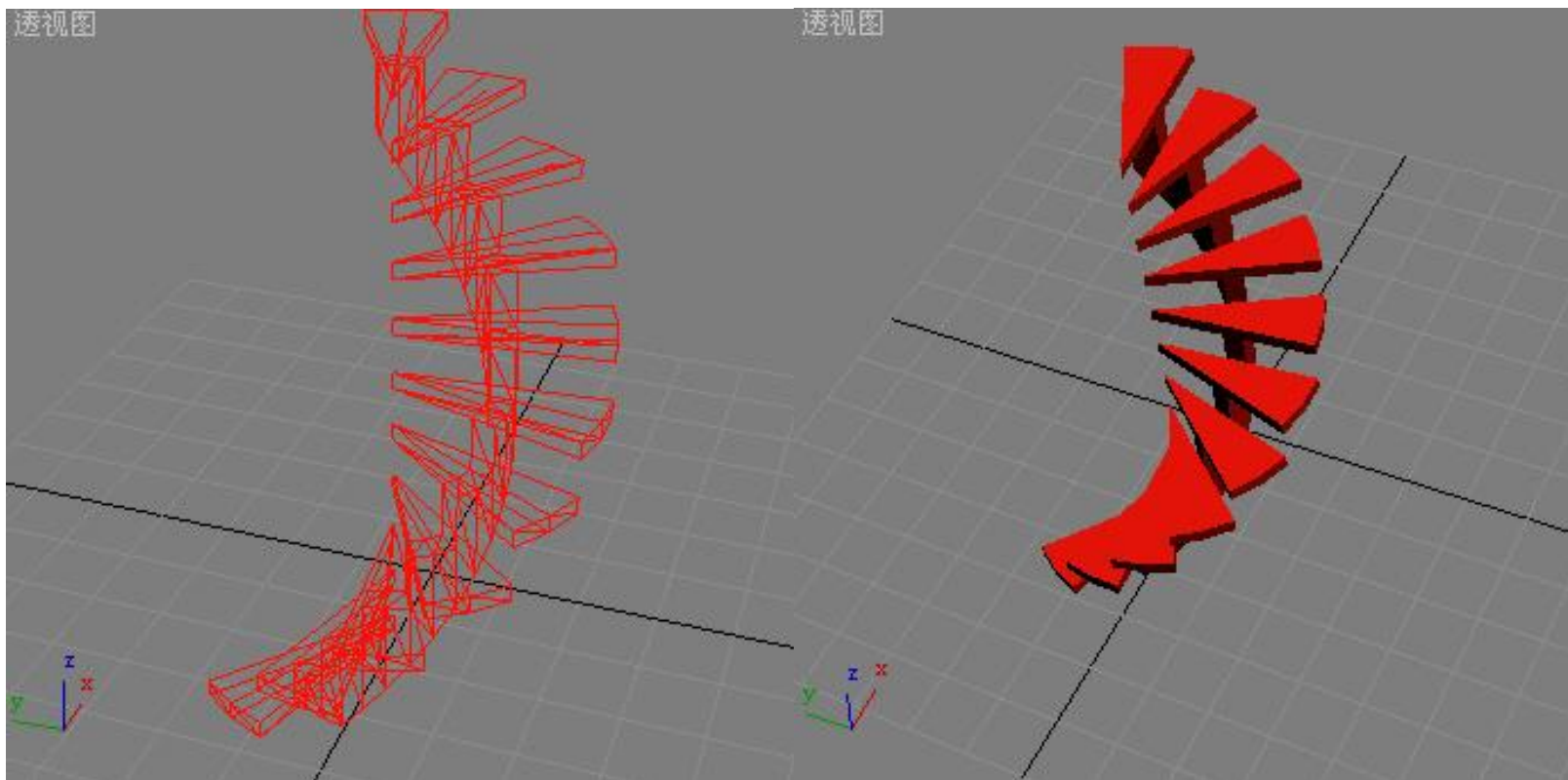


图4.9 线框和实体显示的三角楼梯

三维形体的表示

- 多边形表面模型
- 扫描表示
- 构造实体几何法
- 空间位置枚举表示
- 八叉树

多边形表面模型

- **边界表示 (B-reps)** 的最普遍方式是**多边形表面模型**，它使用一组包围物体内部的平面多边形，也即平面多面体，来描述实体。

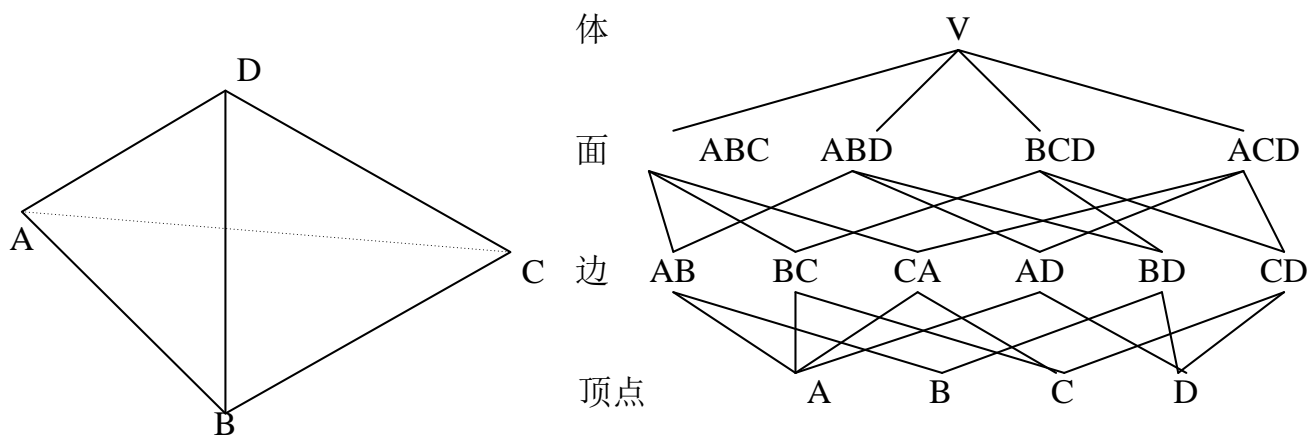


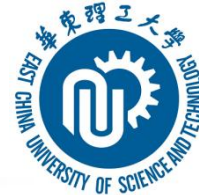
图4.10 四面体及其点、边、面的关系

□ 几何信息

- 建立3张表：顶点表、边表和多边形表来存储几何数据。
- 实体模型中，用多边形顶点坐标值以及多边形所在平面方程方式保存实体单个表面部分的空间方向信息。

曲面呢？

多边形表面模型——数据结构



□ 拓扑信息：翼边结构表示 (Winged Edges Structure)

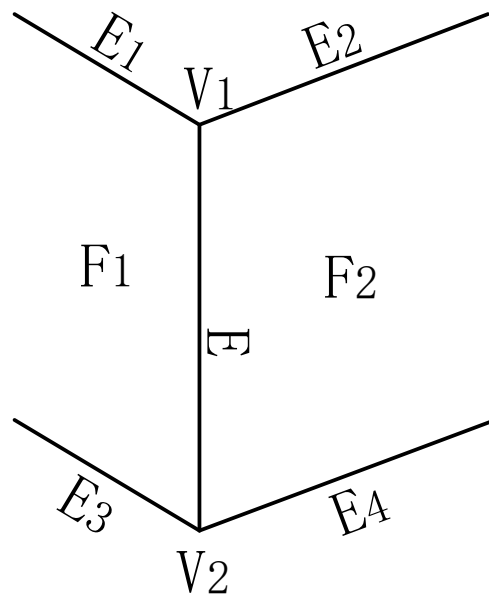
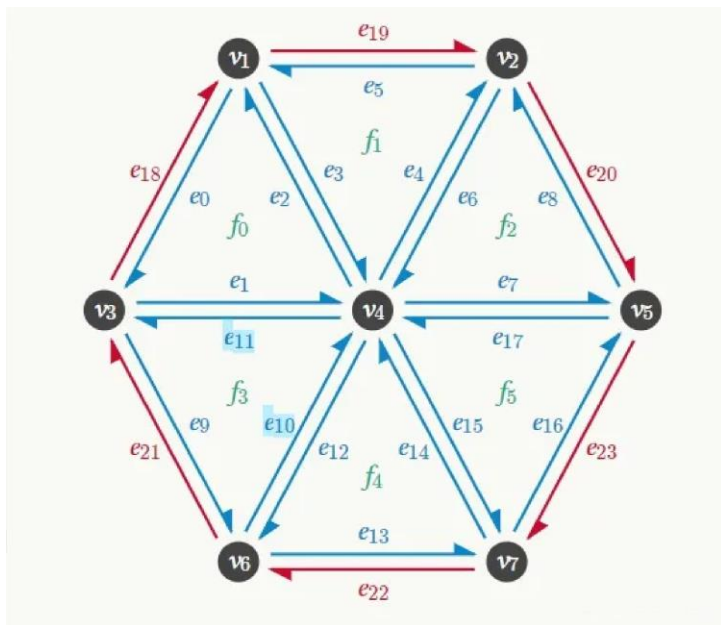


图4.11 翼边结构表示

注意：现在所有几何引擎采用是“半边结构表示”

半边数据结构

- 不是存储网格的边（Edge），而是存储半边（Half Edge）。
- 半边是边的一半并且是通过沿其长度分割边来构造的。
- 我们将构成边的两个半边称为半边对（Pair）。
- 半边是定向的，半边对的两条边有相反的方向。



□ 属性信息

用属性表来存储多边形面的属性，指明物体透明度及表面反射度的参数和纹理特征等等。

多边形表面模型

- 多边形网格：三维形体的边界通常用多边形网格（polygon mesh）的拼接来模拟显示或者仿真；精细表示的用NURBS曲面来表示加工制造。
- 例子

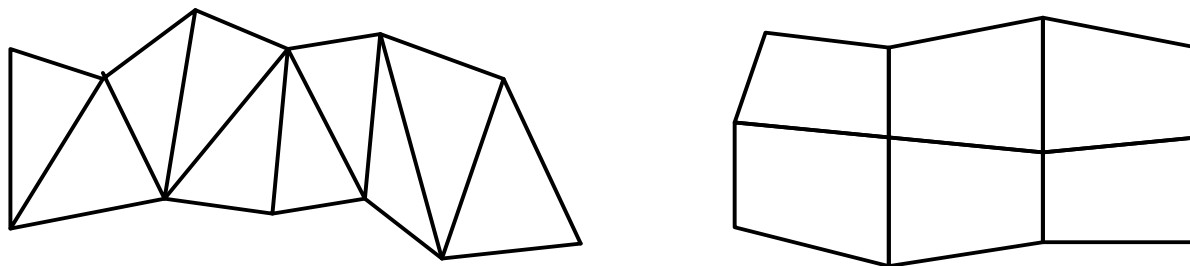
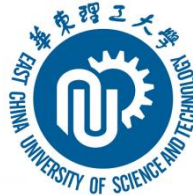


图4.12 三角形带与四边形网格

扫描表示 (sweep representation)



- **扫描表示法** (sweep representation) 可以利用简单的运动规则生成有效实体。
- 包含两个要素
 - 一是作扫描运动的基本图形（截面）；
 - 二是扫描运动的方式。

扫描表示 (sweep representation)

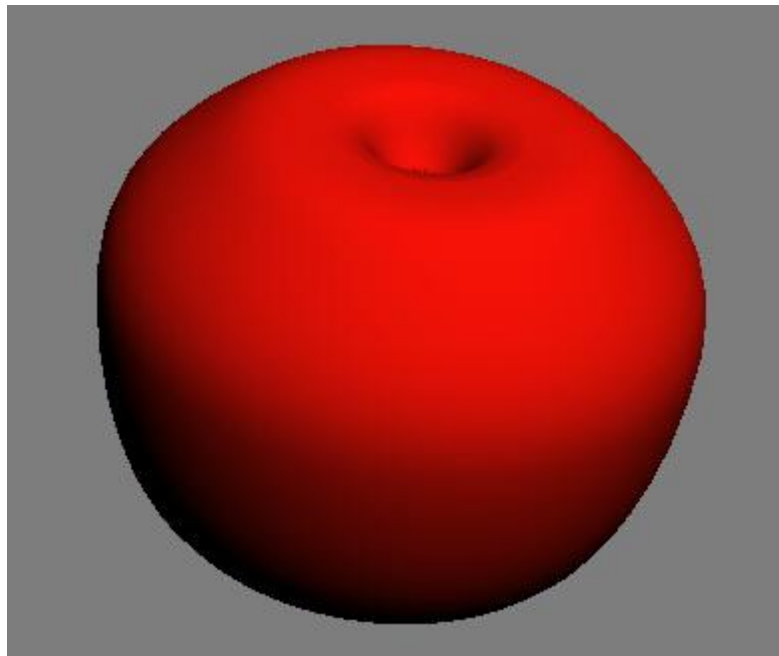
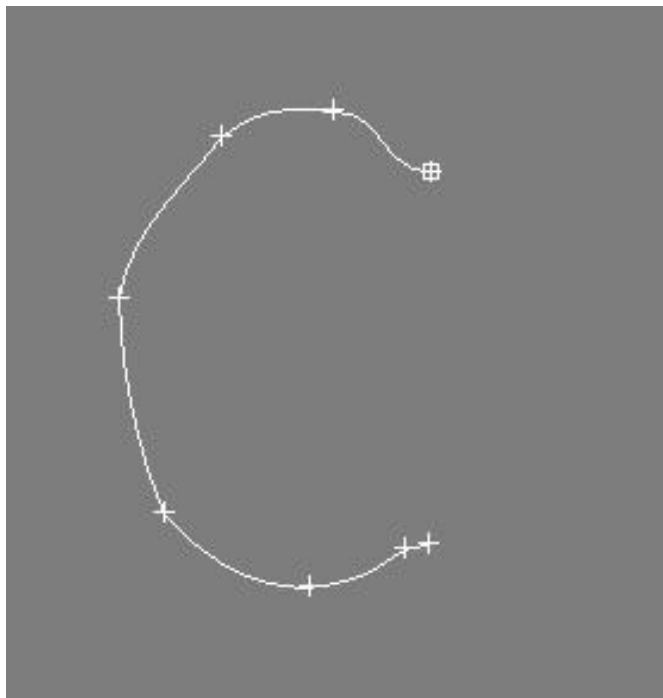


图4.13 利用旋转构造苹果造型

扫描表示 (sweep representation)

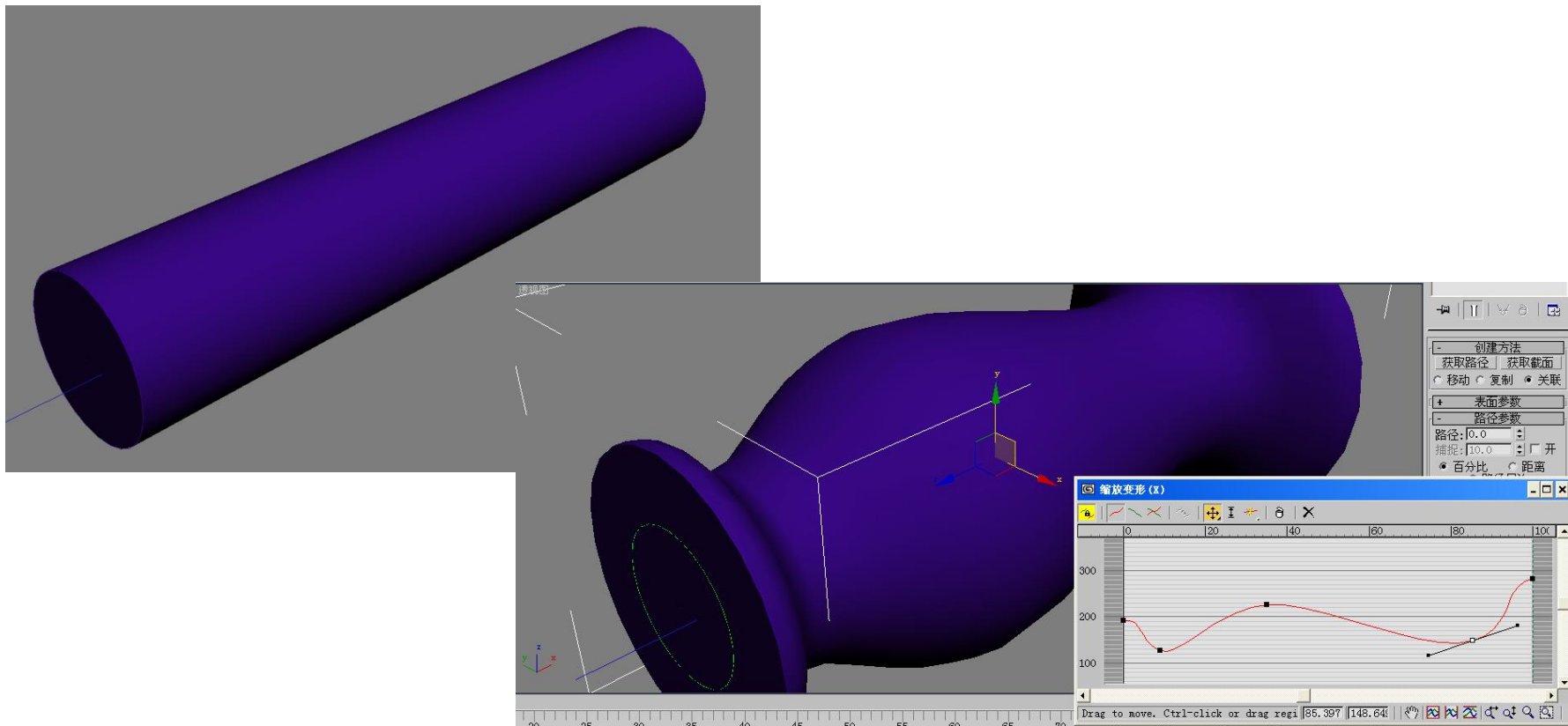
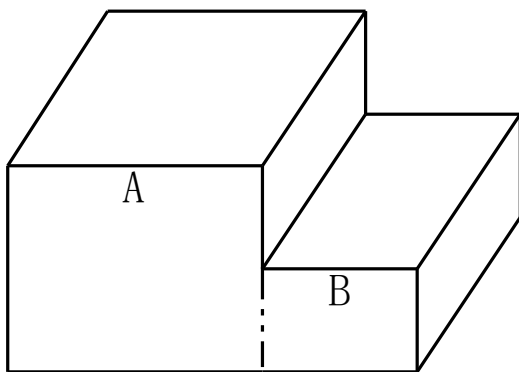


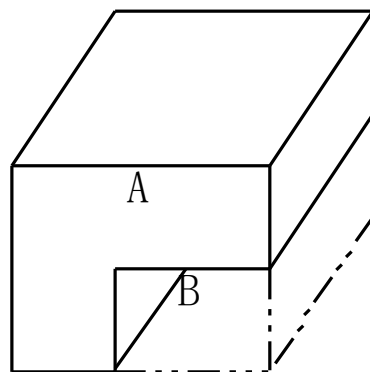
图4.14 利用放样实现截面的变化

构造实体几何法

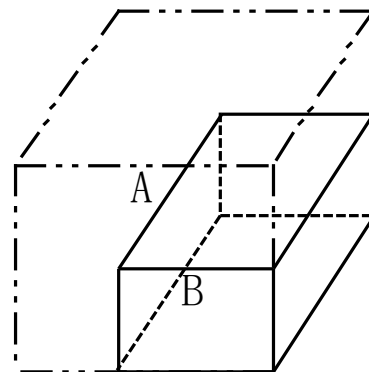
- **构造实体几何法**（CSG, Constructive Solid Geometry）由两个实体间的并、交或差操作生成新的实体。



(a) A, B形体的并



(b) A, B形体的差

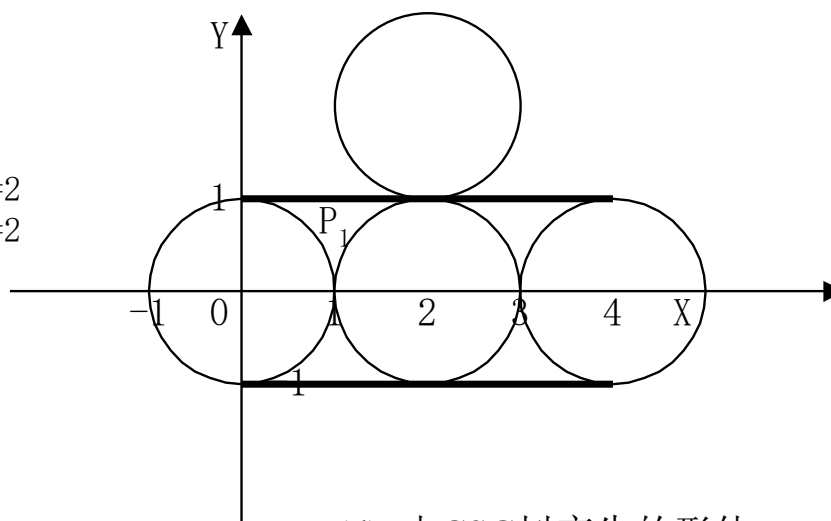
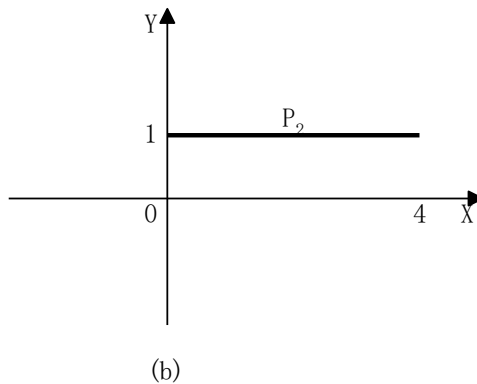


(c) A, B形体的交

图4.15 构造实体几何法

构造实体几何法

- 在构造实体几何法中，集合运算的实现过程可以用一棵二叉树（称为**CSG树**）来描述。
 - 树的叶子是基本体素或是几何变换参数；
 - 树的非终端结点是施加于其子结点的正则集合算子（正则并、正则交和正则差）或几何变换的定义。



(d) 由CSG树产生的形体

图4.16 由CSG树产生二维形体的实例

构造实体几何法

- **优点：** 如果体素设置比较齐全，通过集合运算就可以构造出多种不同的符合需要的实体。
- **缺点一：** 集合运算的中间结果难以用简单的代数方程表示，求交困难。
- **缺点二：** CSG树不能显式地表示形体的边界，因而无法直接显示CSG树表示的形体。

构造实体几何法

- 解决：光线投射算法

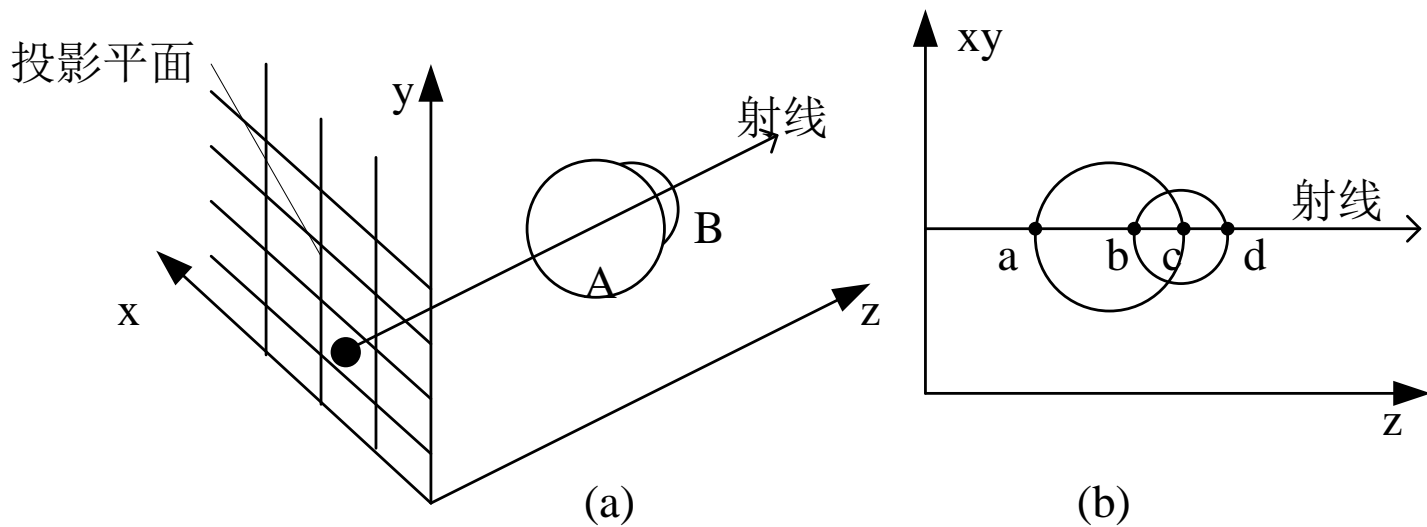


图4.17 光线投射算法
(实体 $A \cup B$ 取 ad , 实体 $A \cap B$ 则取 cb , 实体 $A - B$ 则取 ab)

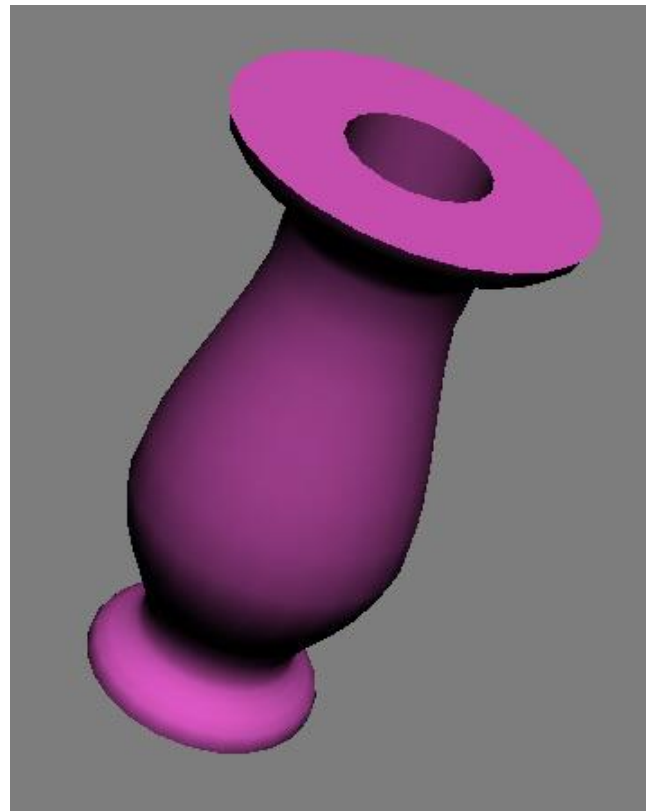
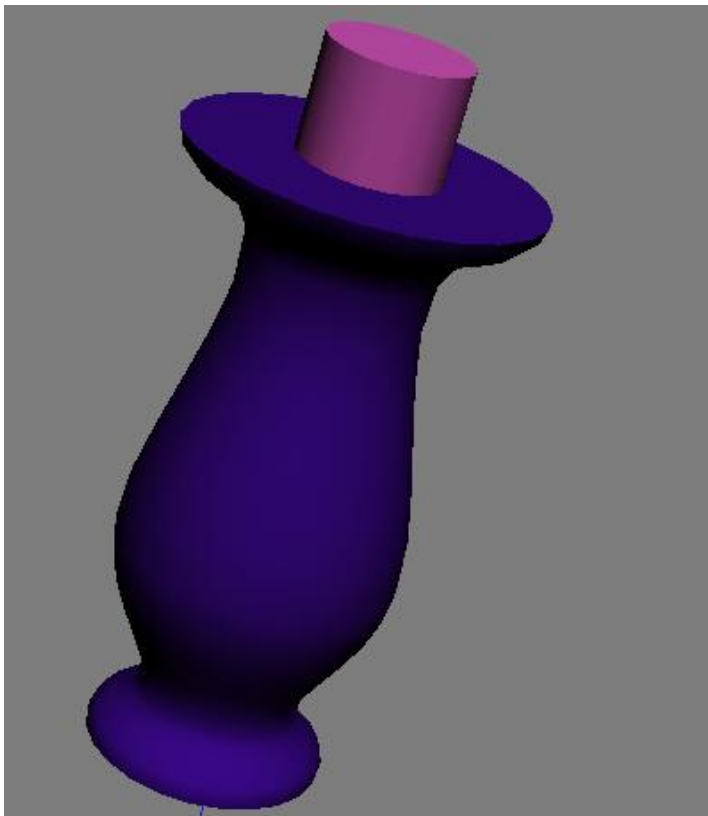
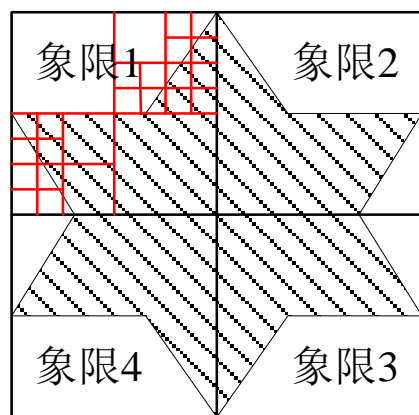


图4.18 利用差运算实现打孔

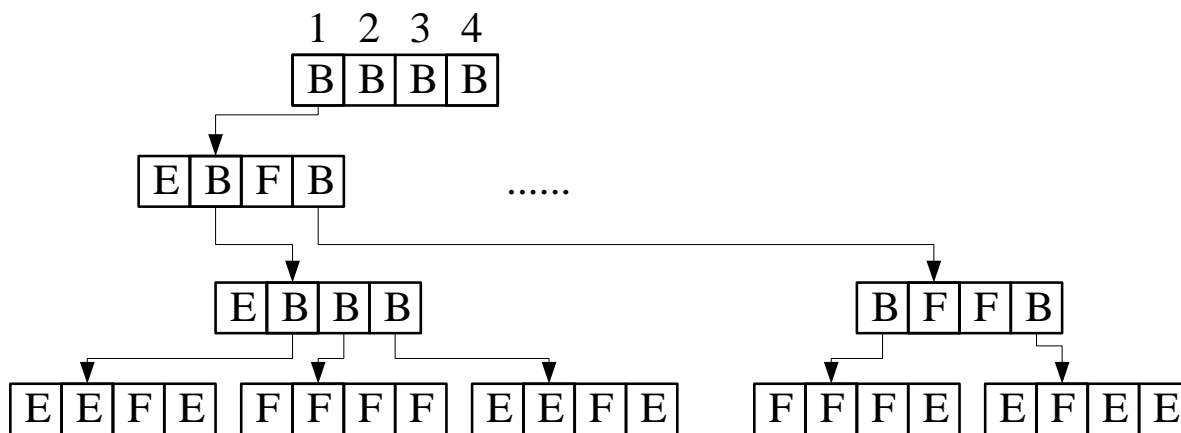
空间位置枚举表示

- **空间位置枚举表示法**将包含实体的空间分割为大小相同、形状规则（正方形或立方体）的体素，然后，以体素的集合来表示图形对象。
 - 二维情况，常用二维数组存放。
 - 三维情况下，常用三维数组 $p[i][j][k]$ 来存放。

- **八叉树 (octrees)** 又称为分层树结构，它对空间进行自适应划分，采用具有层次结构的八叉树来表示实体。



(a)



(b)

图4.19 二维图的四叉树表示

问题呢？

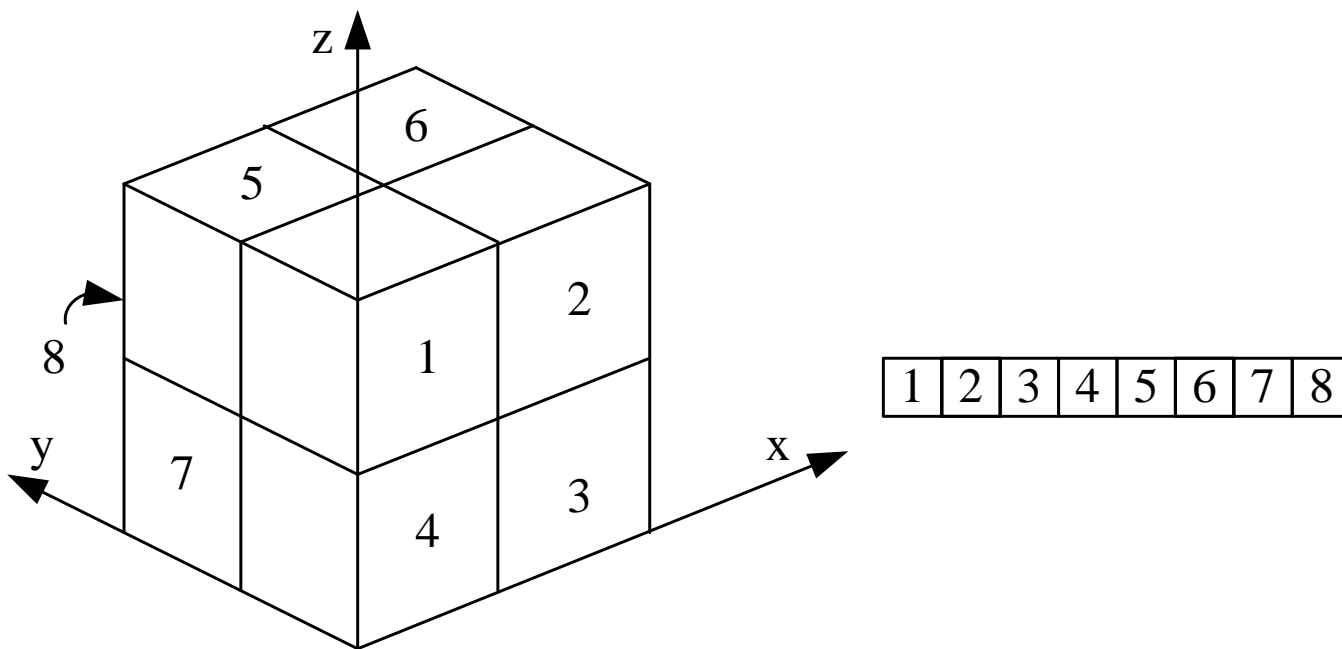


图4.20 三维空间分成八个卦限及其节点表示

BSP树

- **二叉空间分割 (Binary Space Partitioning, BSP) 树**方法是一种类似于八叉树的空间分割方法，它每次将一实体用任一位置和任一方向的平面分为二部分（不同于八叉树方法的每次将实体用平行于笛卡尔坐标平面的三个两两垂直的平面分割）。

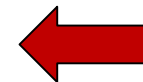
三维形体的表示

1

规则对象的表示方法

2

非规则对象的表示方法



3

常用的显示表示-网格



图形的层次结构



实验课

- 今晚6:00开始
- 215 计算机
- 218 软件

谢谢！

*Thank
You*