



地信网论坛

BBS.3S001.COM

三维重建算法研究和软件系统实现

答辩人： 徐淼华

导师： 赵振西 教授

专业： 计算机软件与理论

内容介绍

□ 绪论

PACS系统, DICOM格式,
三维重建的应用和研究内容, 介绍本文工作

□ 三维重建算法

表面重建、体重建

□ 软件系统实现

实现中的具体问题、设计模式的应用

□ 总结和展望

一、绪论

- PACS 简介
 - DICOM 简介
 - 三维重建的应用
 - 三维重建的研究内容
 - 本文的工作
-

1.1 PACS简介

- 含义

Picture Archiving and communication system

- 功能

- 优点

- 国内外开发现状

- 未来展望

1.2 DICOM标准

- 含义

Digital Imaging and communication in Medicine

- 发展历史

- 信息模型

E-R模型

1.4 三维重建的研究内容

- 数据预处理
 - 三维重建算法
 - 医学图象的分割
 - 模型的网格简化
 - 绘制与显示
-

1.5 论文主要工作

- 功能完整的三维重建软件
 - 图象预处理
 - MC算法构建表面几何模型以及模型的简化
 - 轮廓线编辑和轮廓线拼接
 - 三维模型的交互显示
 - 设计模式的应用
 - MC 算法的改进
 - 多轮廓线拼接问题的解决
-

二、三维重建算法研究

□ 表面抽取算法

■ 基于体素的等值面重建

□ MC算法

□ MT算法

□ 剖分立方体算法

■ 断层轮廓拼接

□ 体重建

■ 基于图像空间

■ 基于物体空间

■ 基于频域空间

2.1 Marching Cubes算法

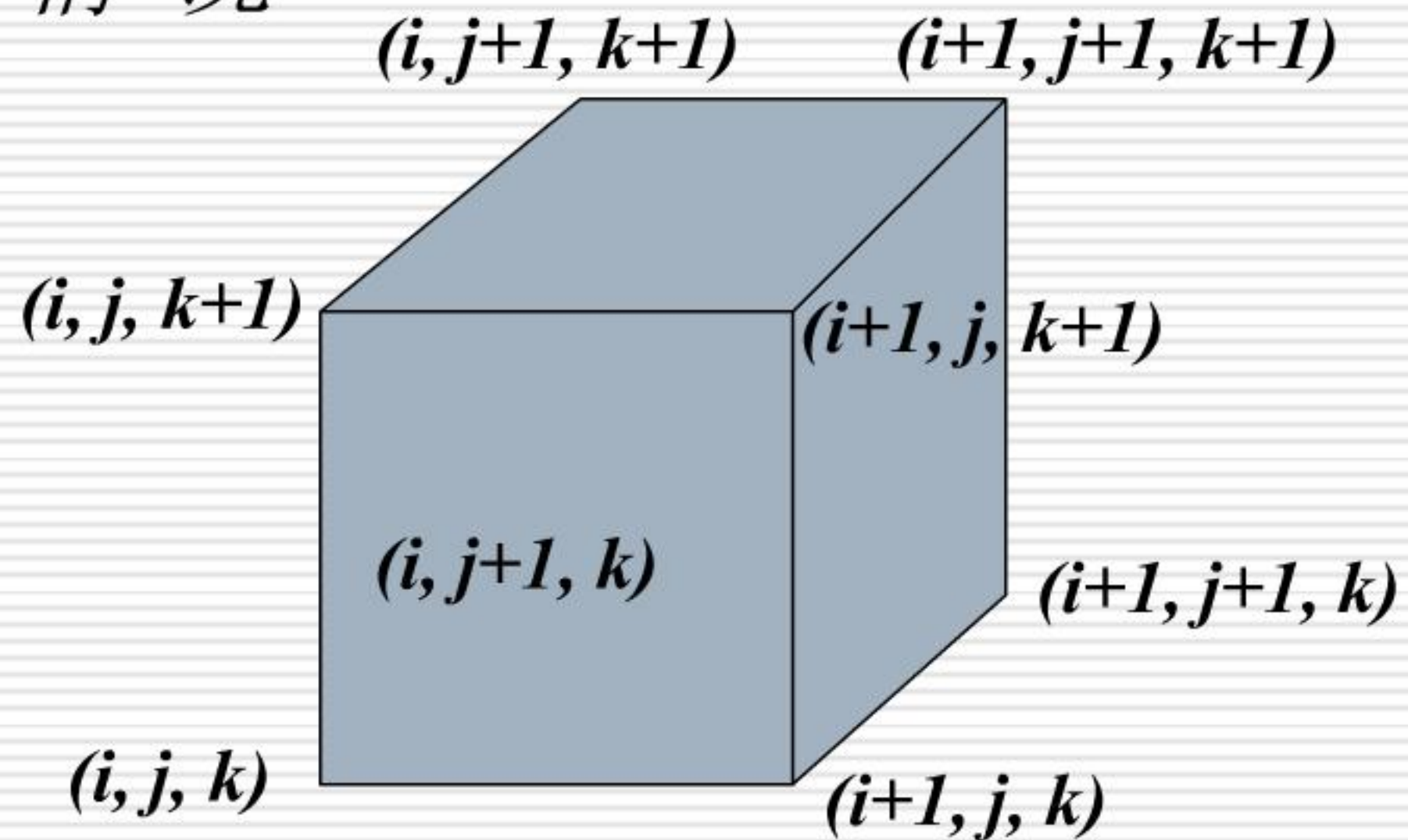
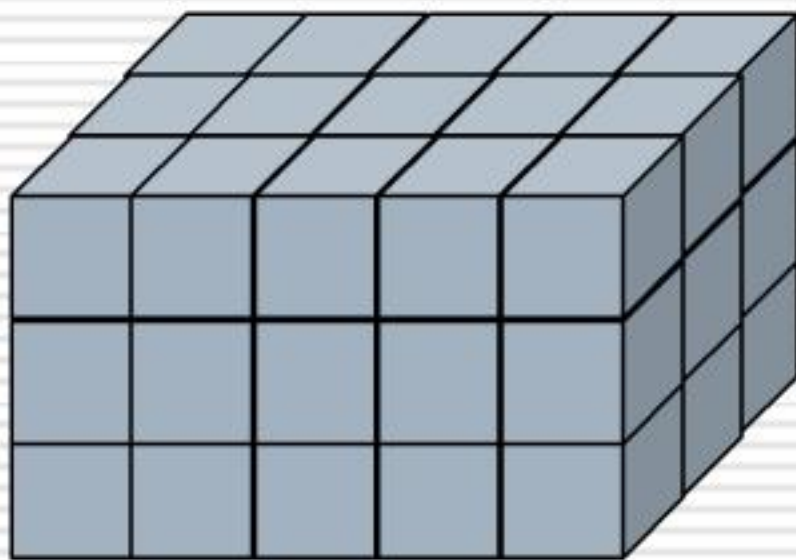
- 算法原理
 - 二义性问题
 - 解决二义性
 - 算法改进
-

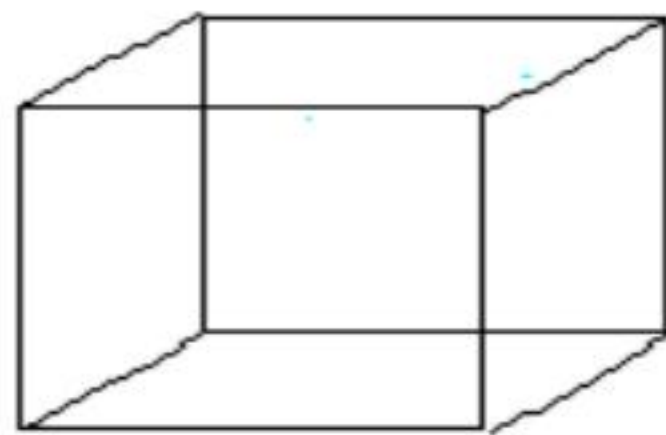
2.1.1 MC算法原理

1. 确定包含等值面的体元

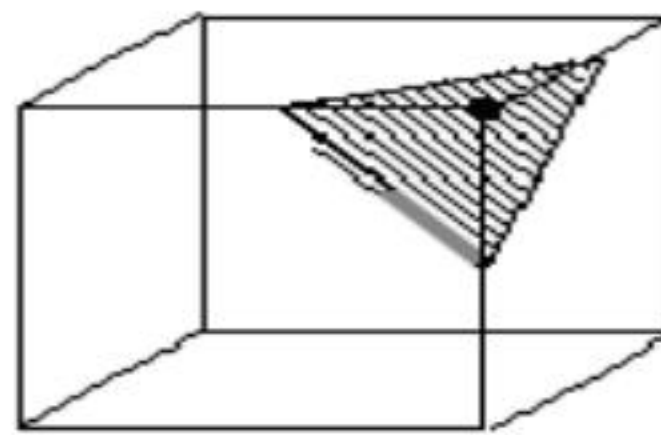
$$flag(i, j, k) = 0 (1)$$

$2^8 = 256 \rightarrow 15$ 种情况

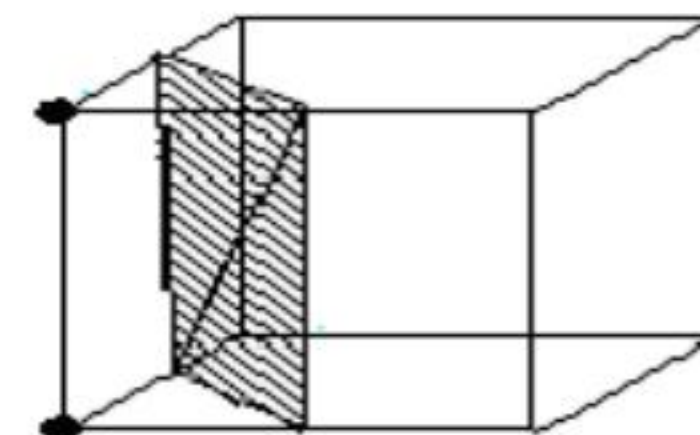




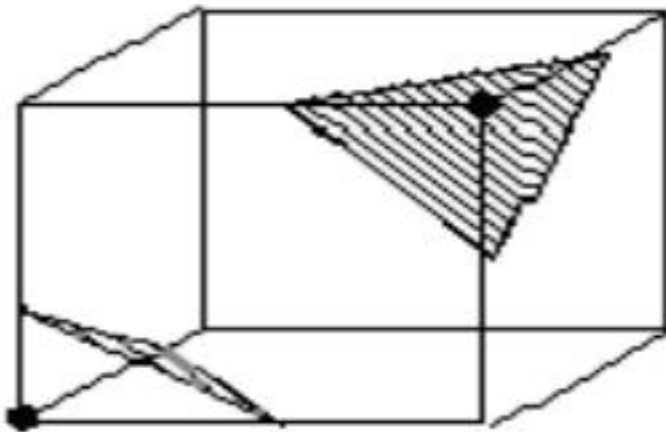
Case 0



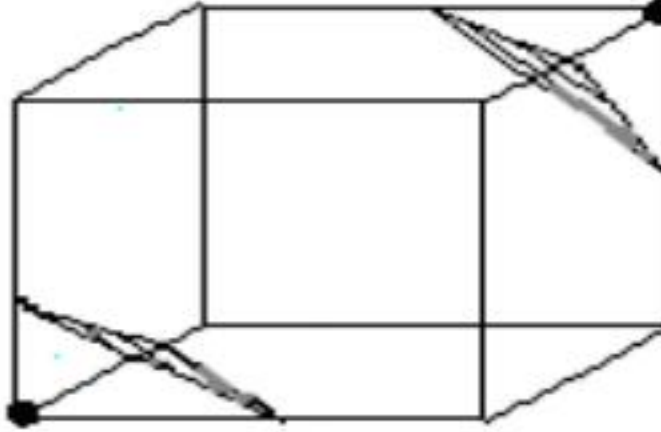
Case 1



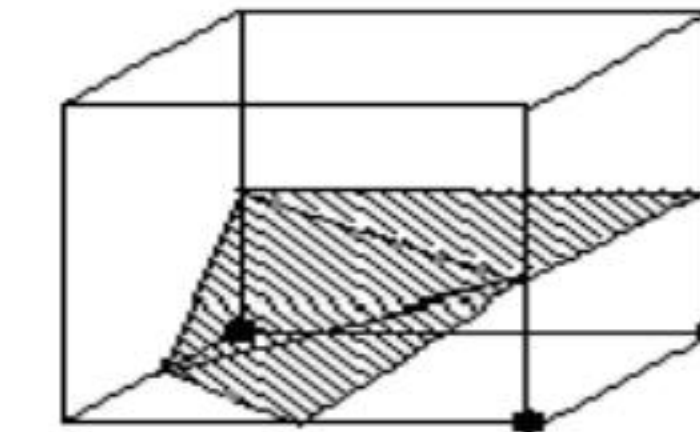
Case 2



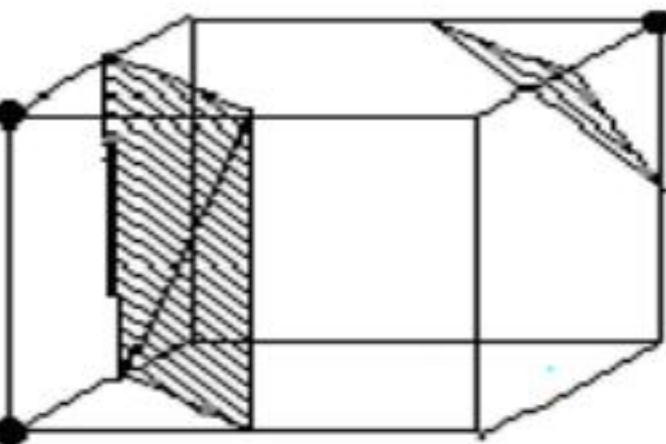
Case 3



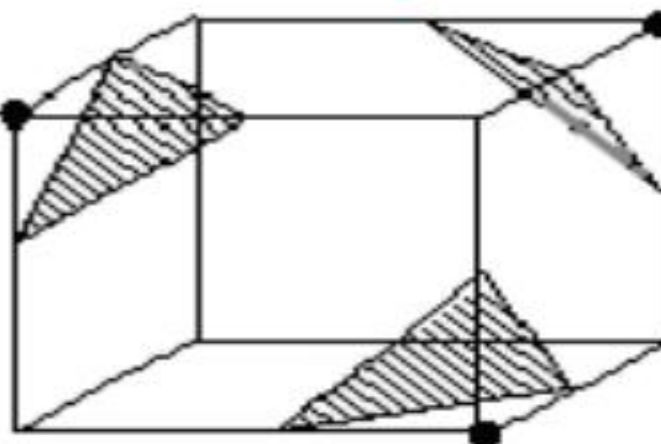
Case 4



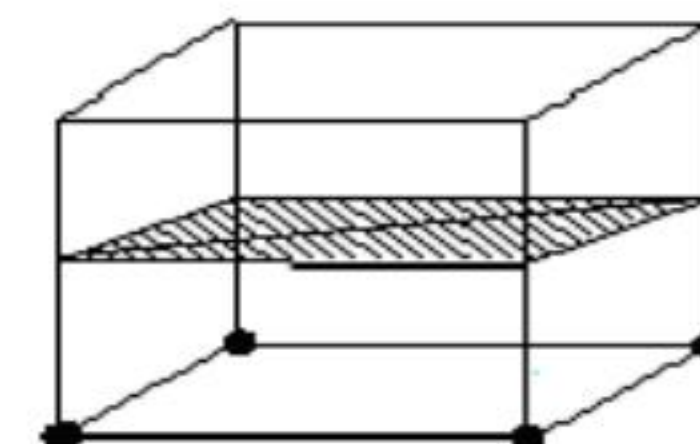
Case 5



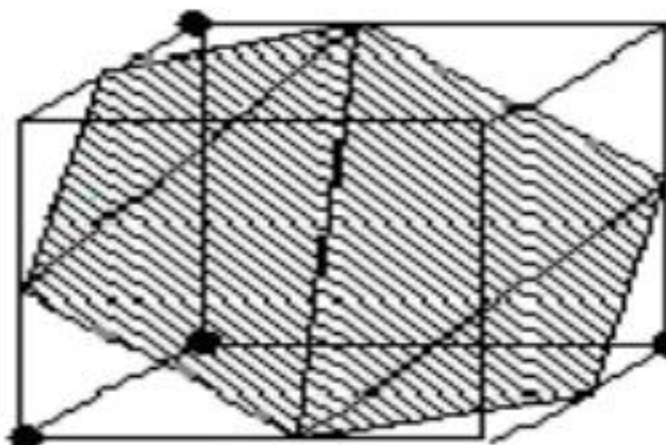
Case 6



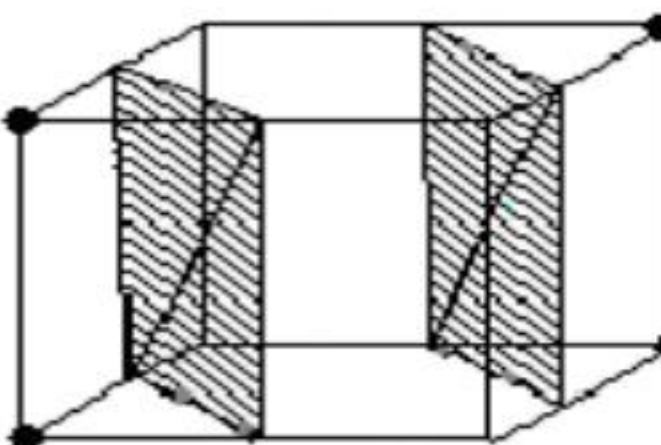
Case 7



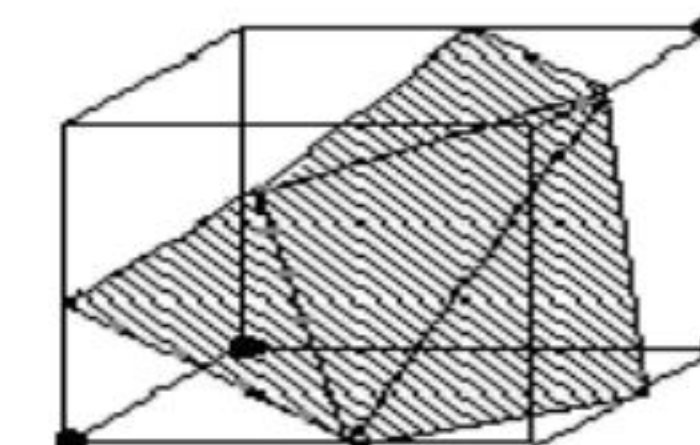
Case 8



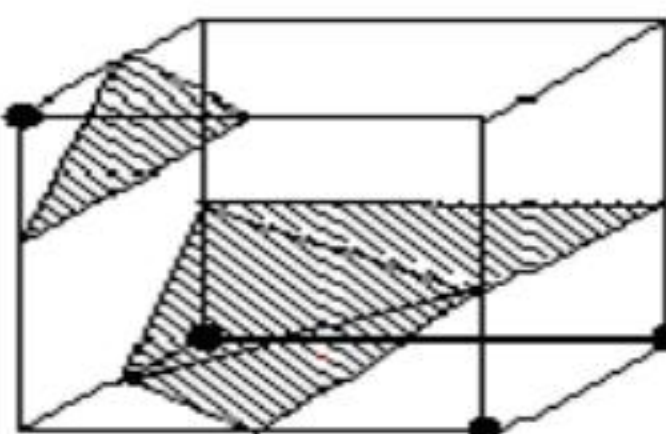
Case 9



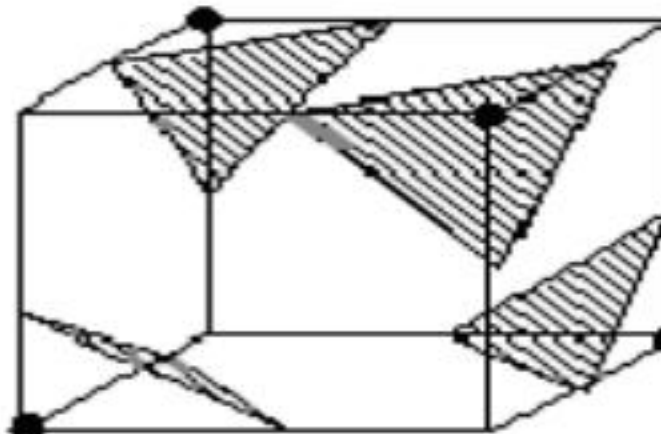
Case 10



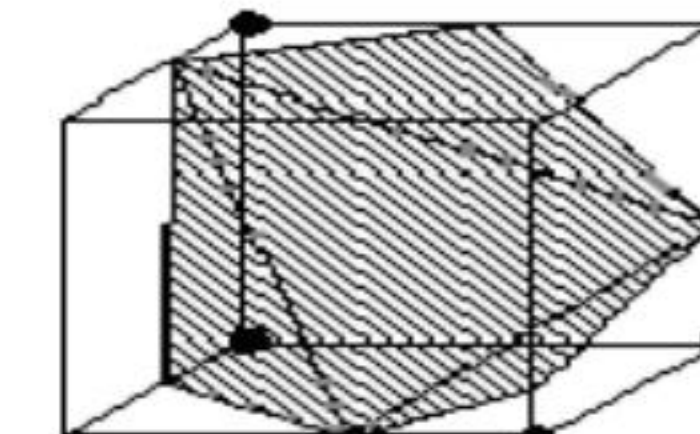
Case 11



Case 12

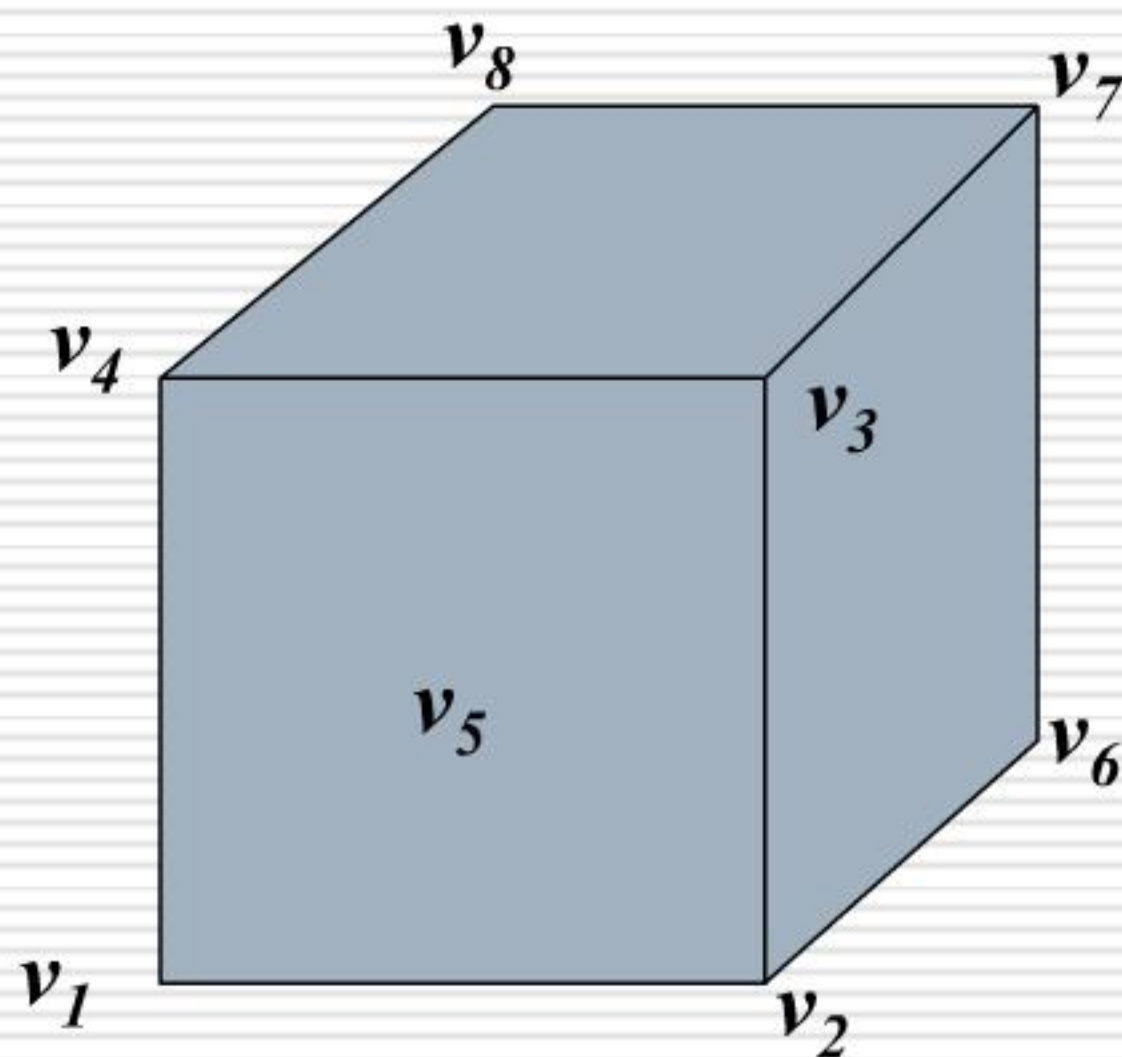


Case 13



Case 14

体元状态表 (1 byte)



v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
-------	-------	-------	-------	-------	-------	-------	-------

2.1.1 MC算法原理

2. 求等值面与体元边界的交点（线性插值）

3. 求等值面的法向（梯度矢量）

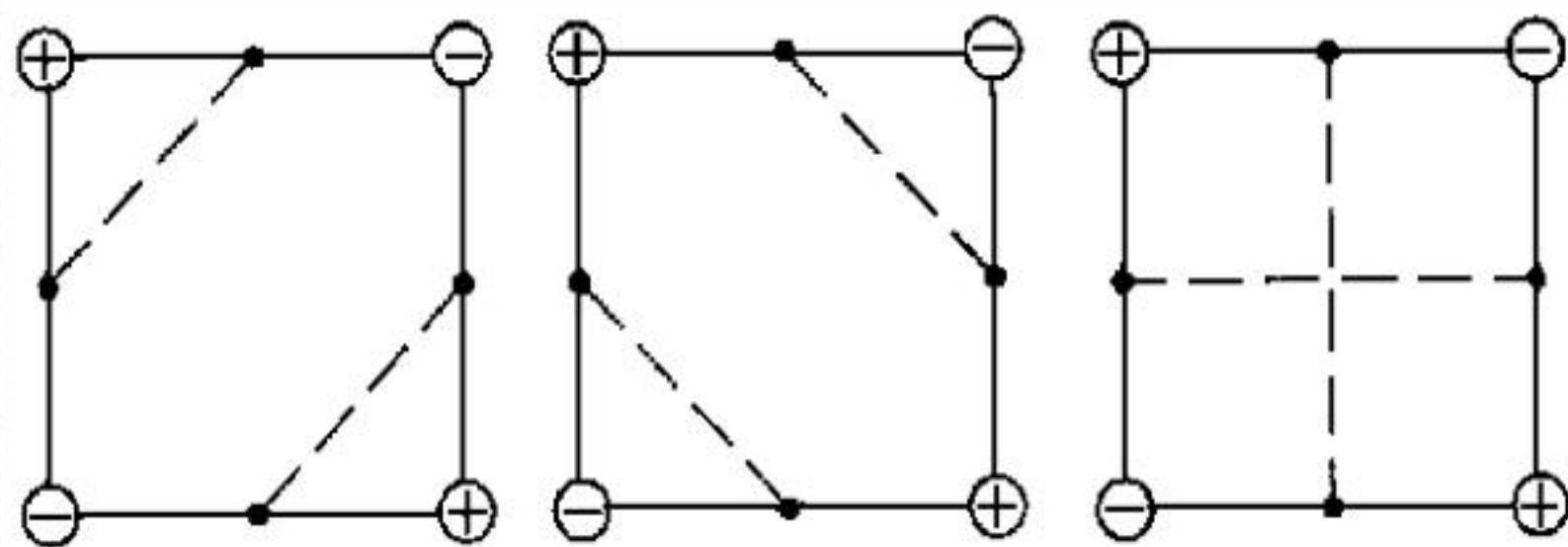
① 中心差分求体元各角点处的梯度

② 体元边界上采用线性插值求交点梯度

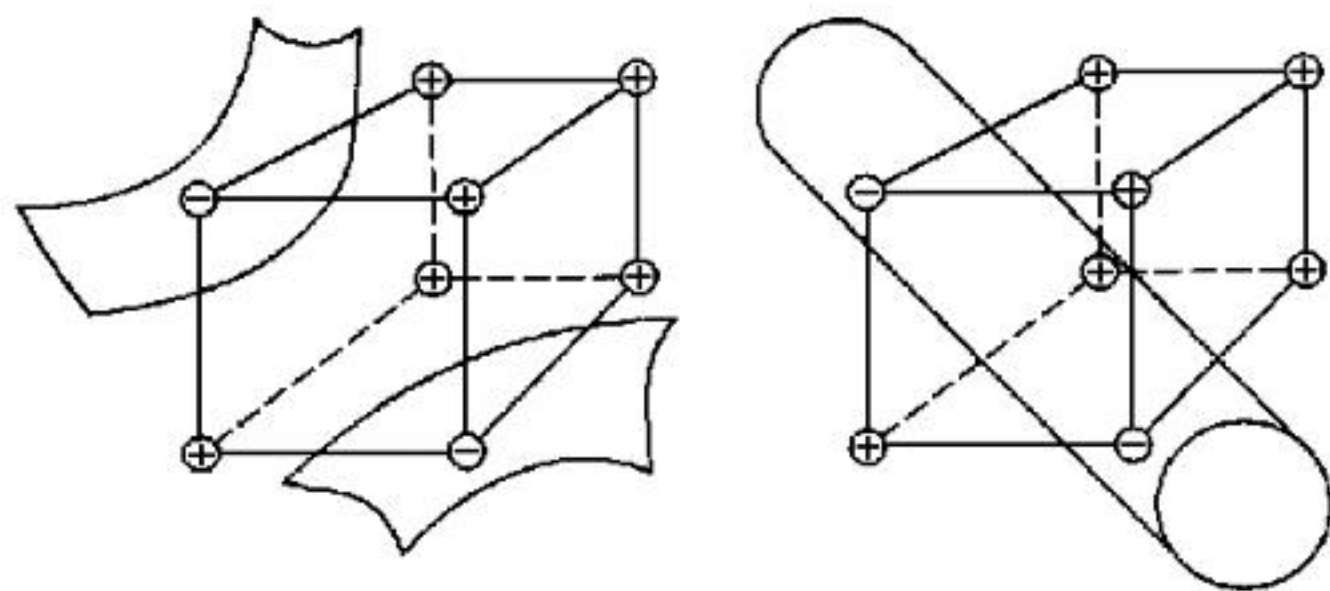
③ 三角面片采用[Gourand模型](#)绘制

$$\begin{aligned} Grad_x &= \frac{f(x_i+1, y_j, z_k) - f(x_i-1, y_j, z_k)}{2\Delta x} \\ Grad_y &= \frac{f(x_i, y_j+1, z_k) - f(x_i, y_j-1, z_k)}{2\Delta y} \\ Grad_z &= \frac{f(x_i, y_j, z_k+1) - f(x_i, y_j, z_k-1)}{2\Delta z} \end{aligned}$$

2.1.2 MC算法的二义性

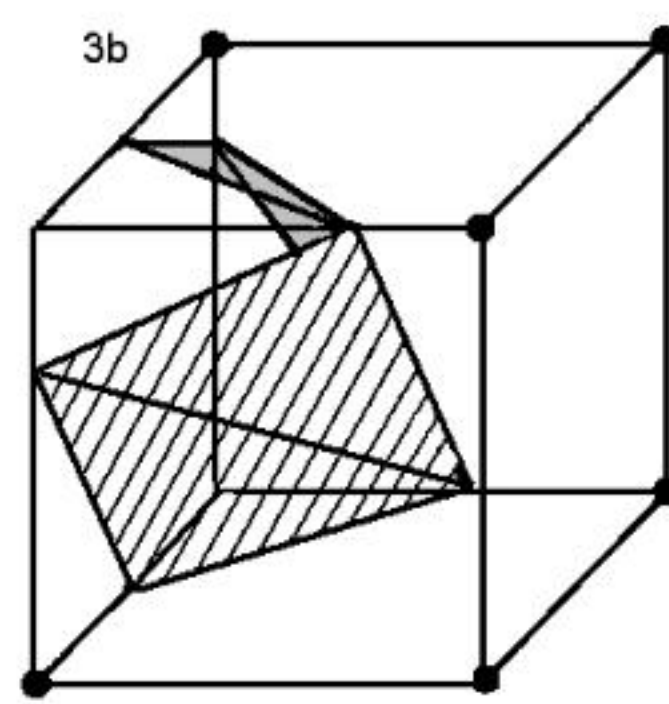
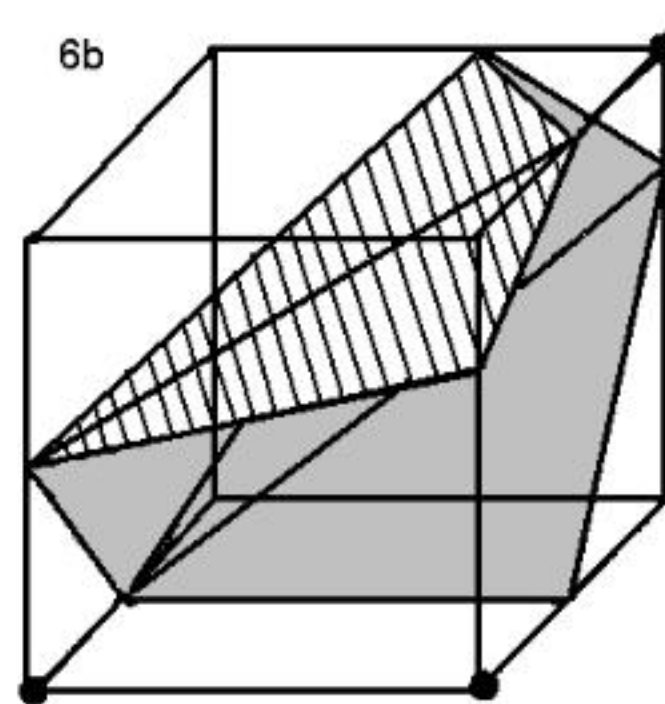
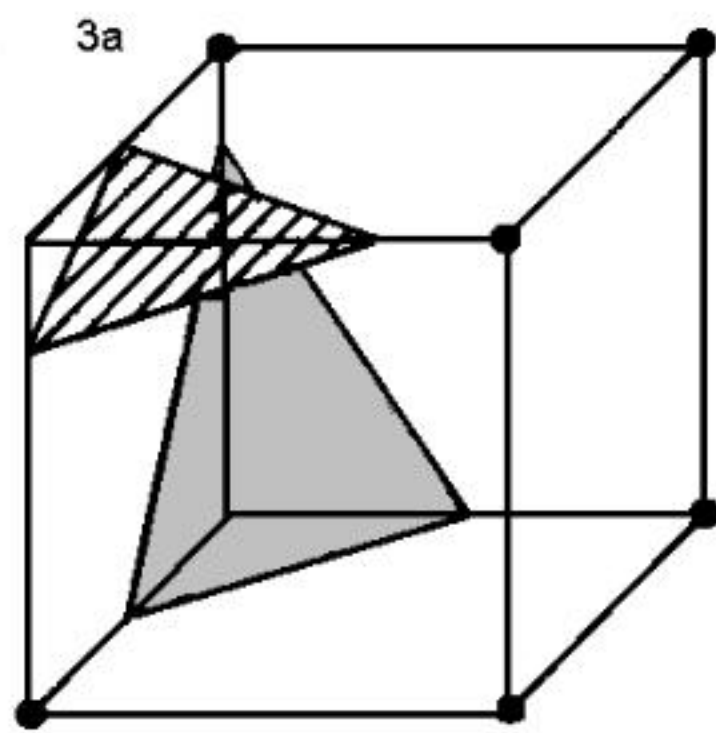
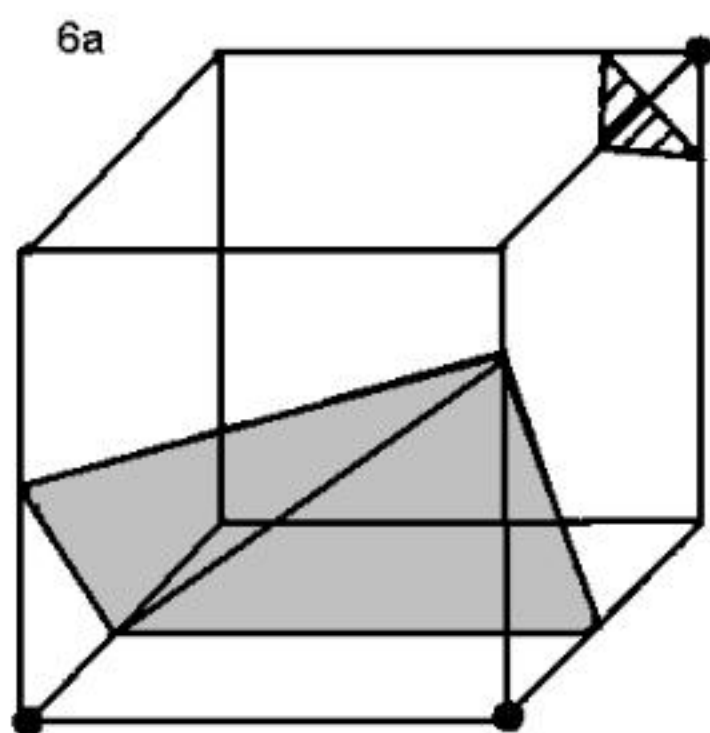


(a) 连接方式二义性的二维表示



(b) 连接方式二义性的三维表示

2.1.2 MC算法的二义性



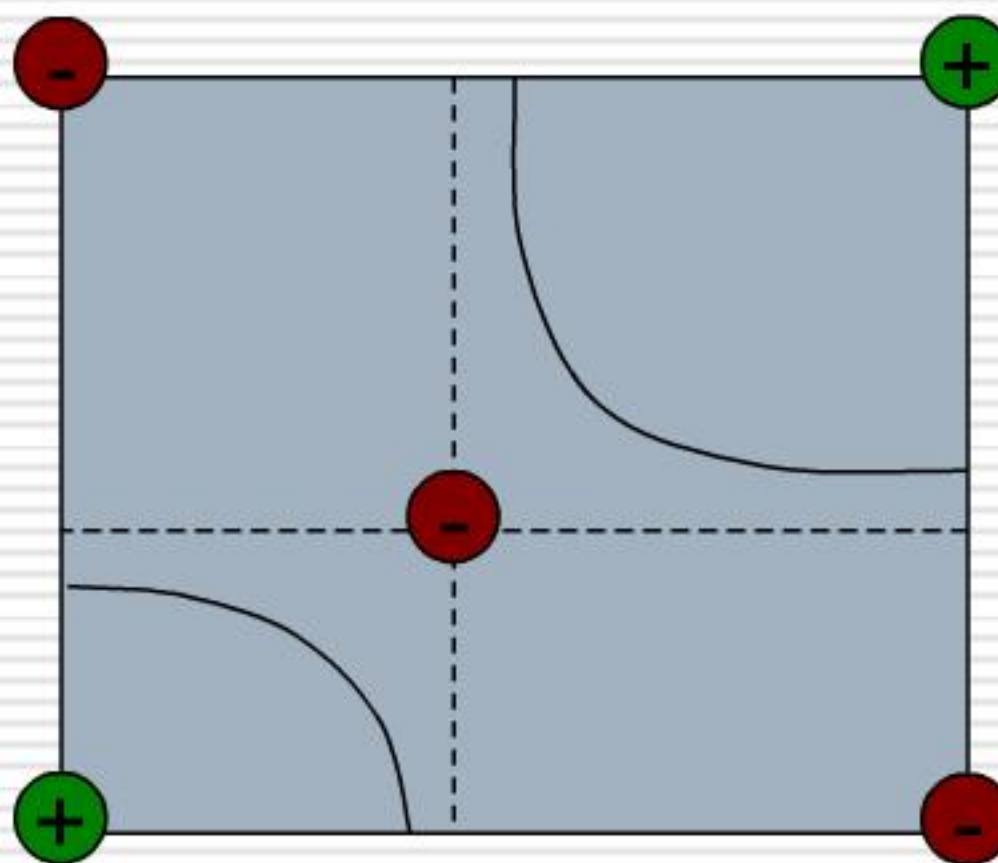
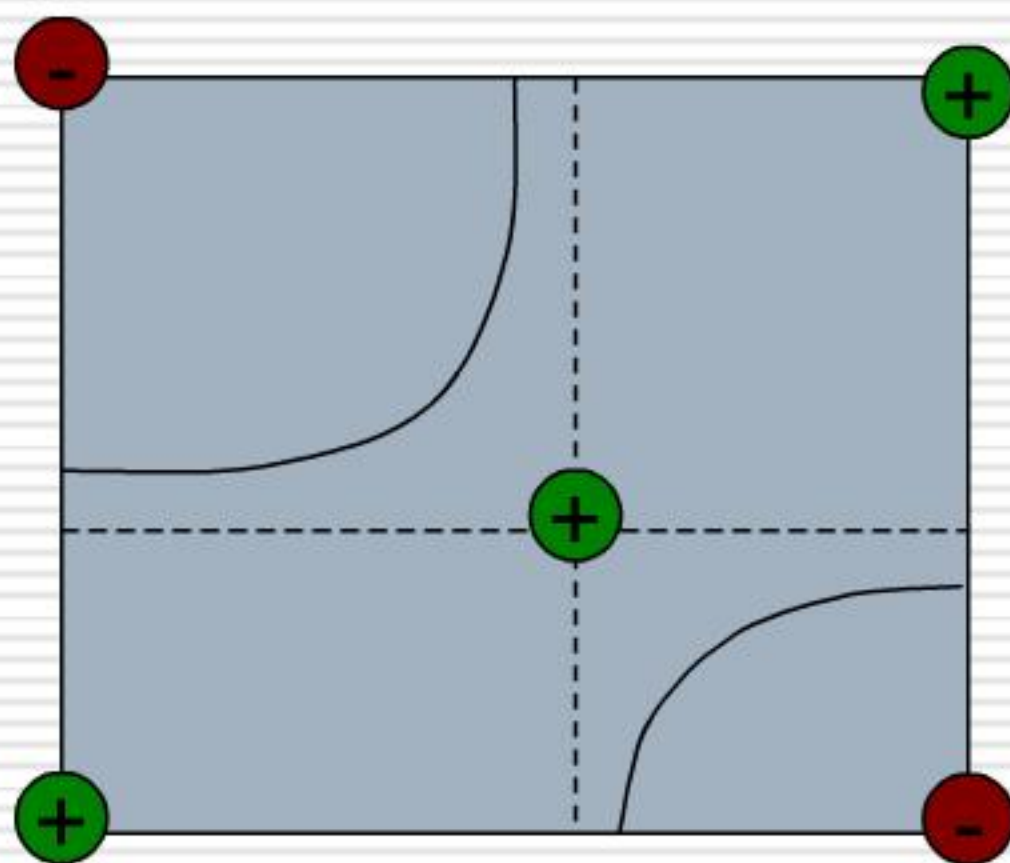
拓扑不一致造成孔隙

2.1.3 消除二义性

$$f(x, y, z) = a_0 + a_1x + a_2y + a_3z + a_4xy + a_5yz + a_6zx + a_7xyz$$

$$x = \frac{a_2 + a_5z_0}{a_4 + a_7z_0}$$

$$y = \frac{a_1 + a_6z_0}{a_4 + a_7z_0}$$

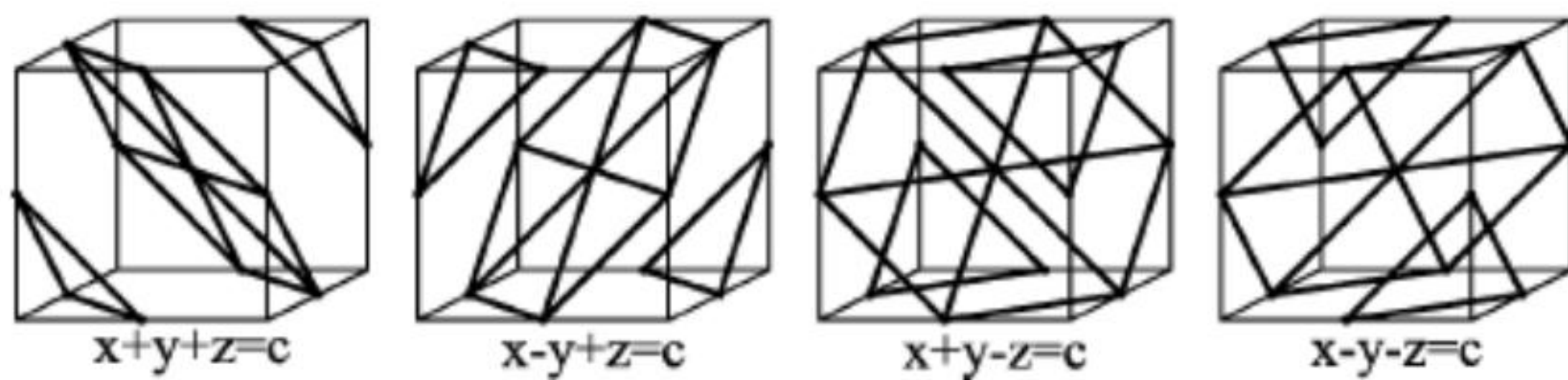
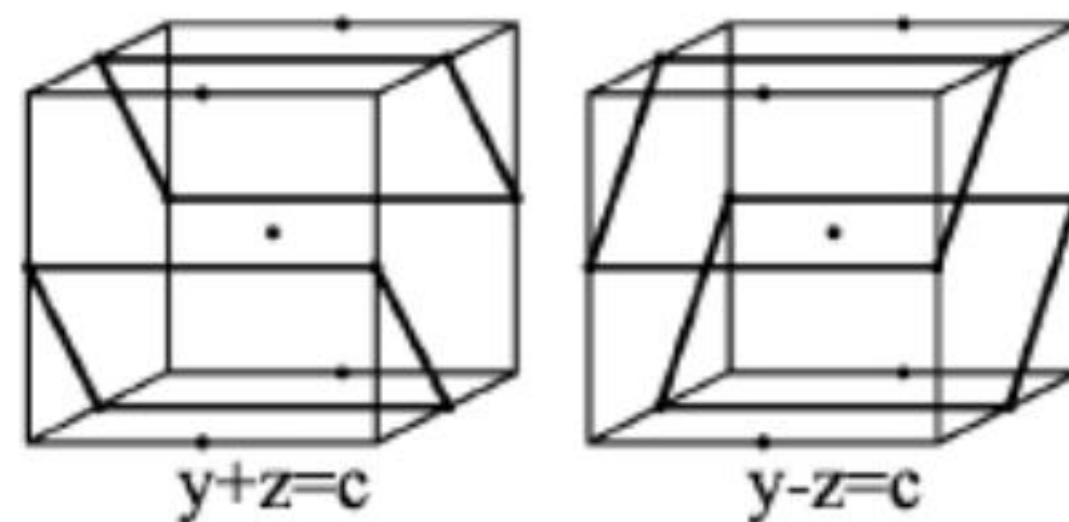
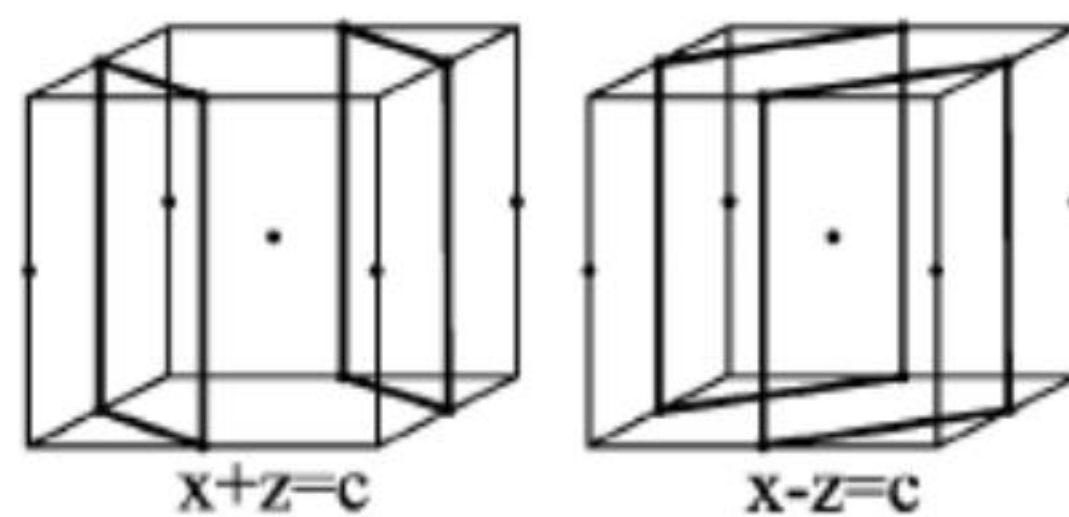
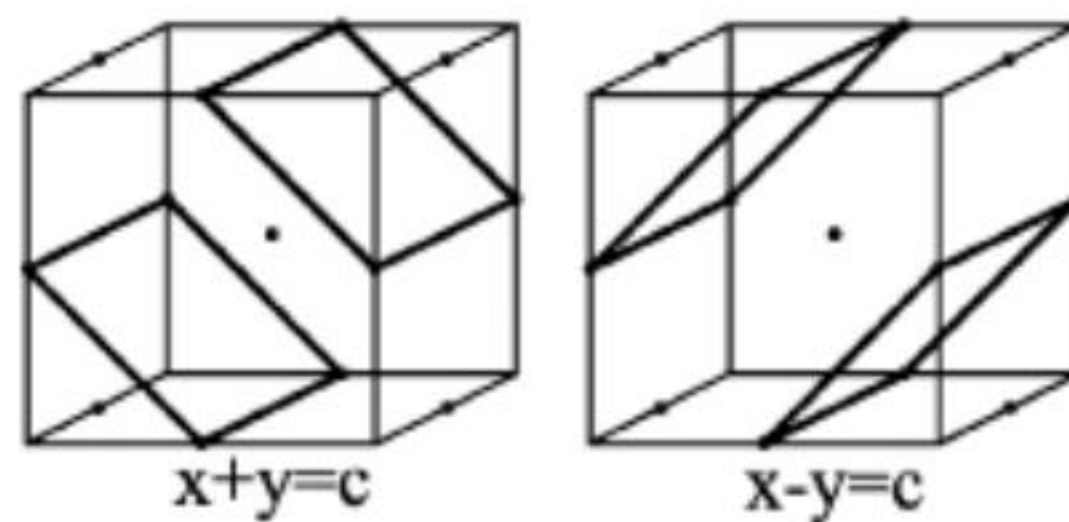
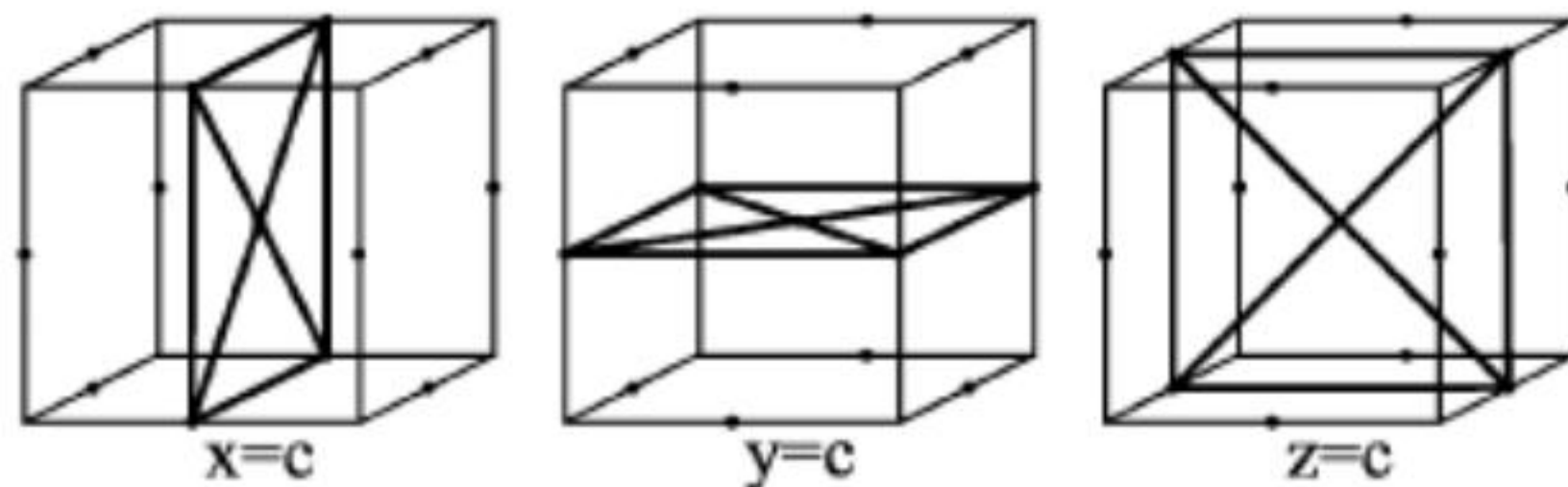


2.1.4 算法改进

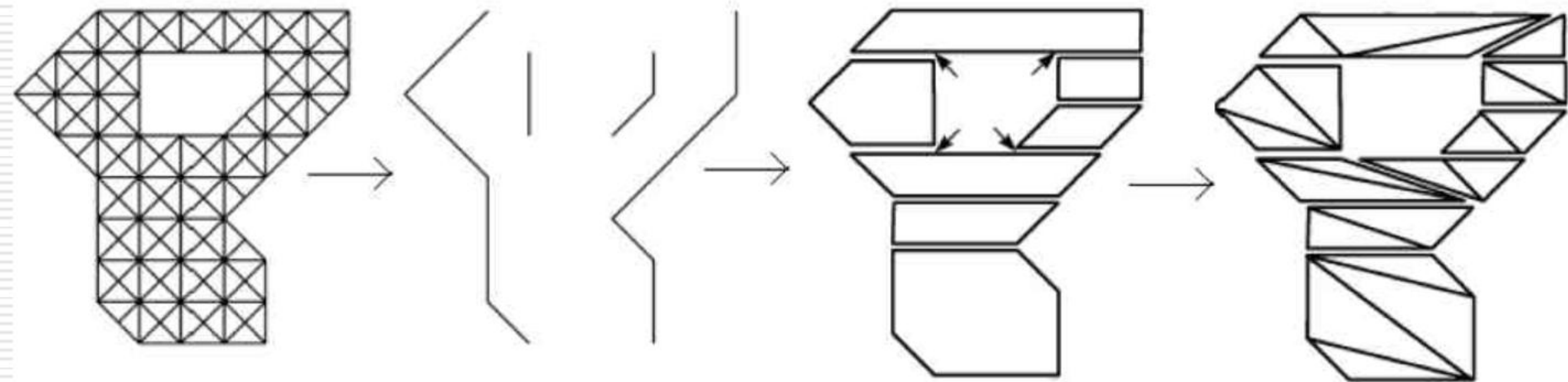
□ 改进方法二

- 扫描：假设三角形面片的顶点只能落在立方体体元边界中点和体元中心点，因为生成的形状个数是有限的
 - 合并：因为方向取值有限，考虑法向后只有26种情况，所以非常适合合并。合并的过程是：先合并成大的多边形，然后重新划分成三角形
 - 插值：通过线性插值计算出最后所得的三角形面片的顶点坐标以及法向量。
-

扫描:



合并



插值：和MC算法相同

2.2 轮廓拼接

□ 单轮廓拼接

■ 凸、单轮廓线拼接

□ 全局搜索策略

□ 启发式算法

■ 最短对角线

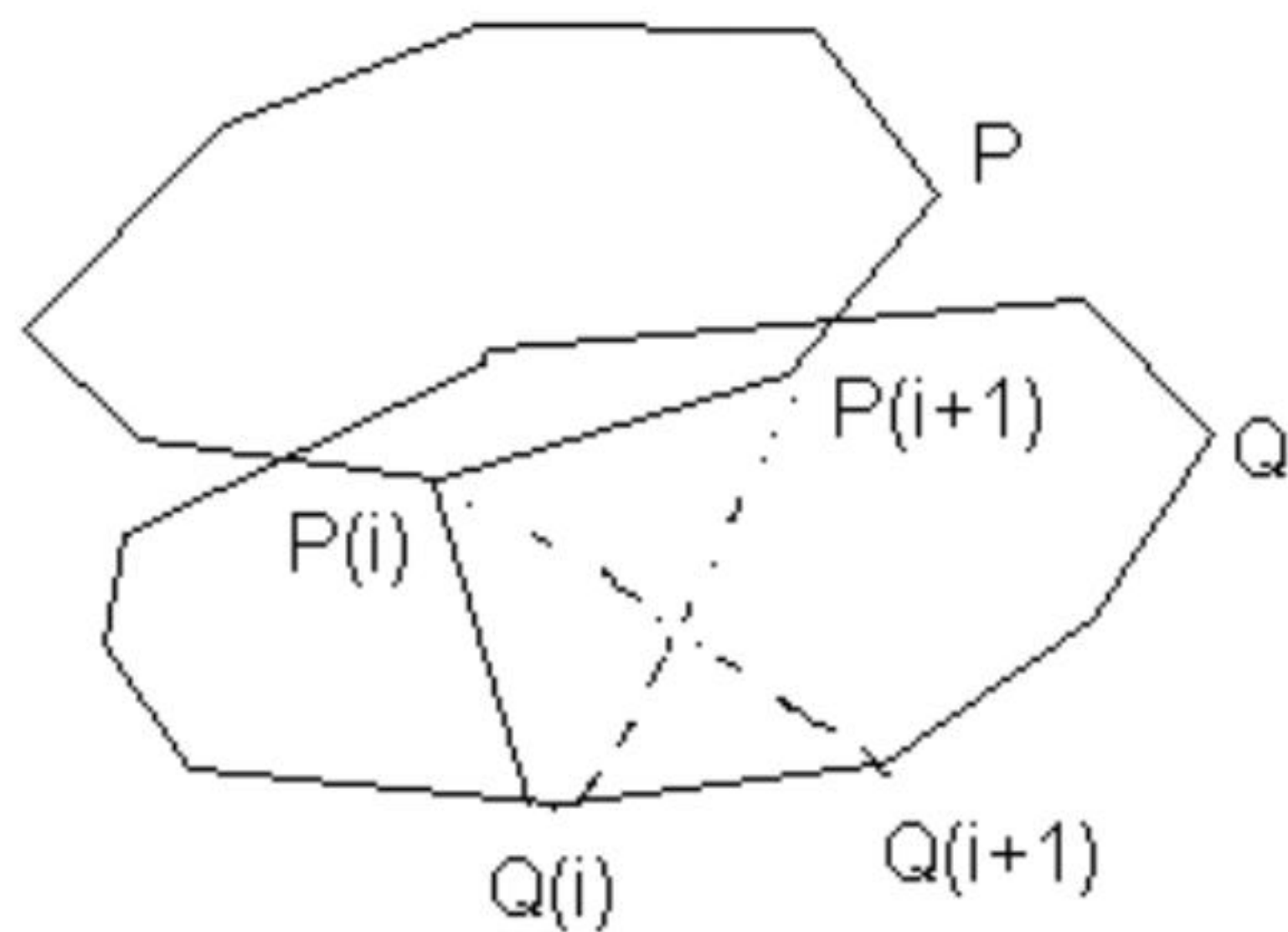
■ 最大体积、

■ 轮廓线渐进

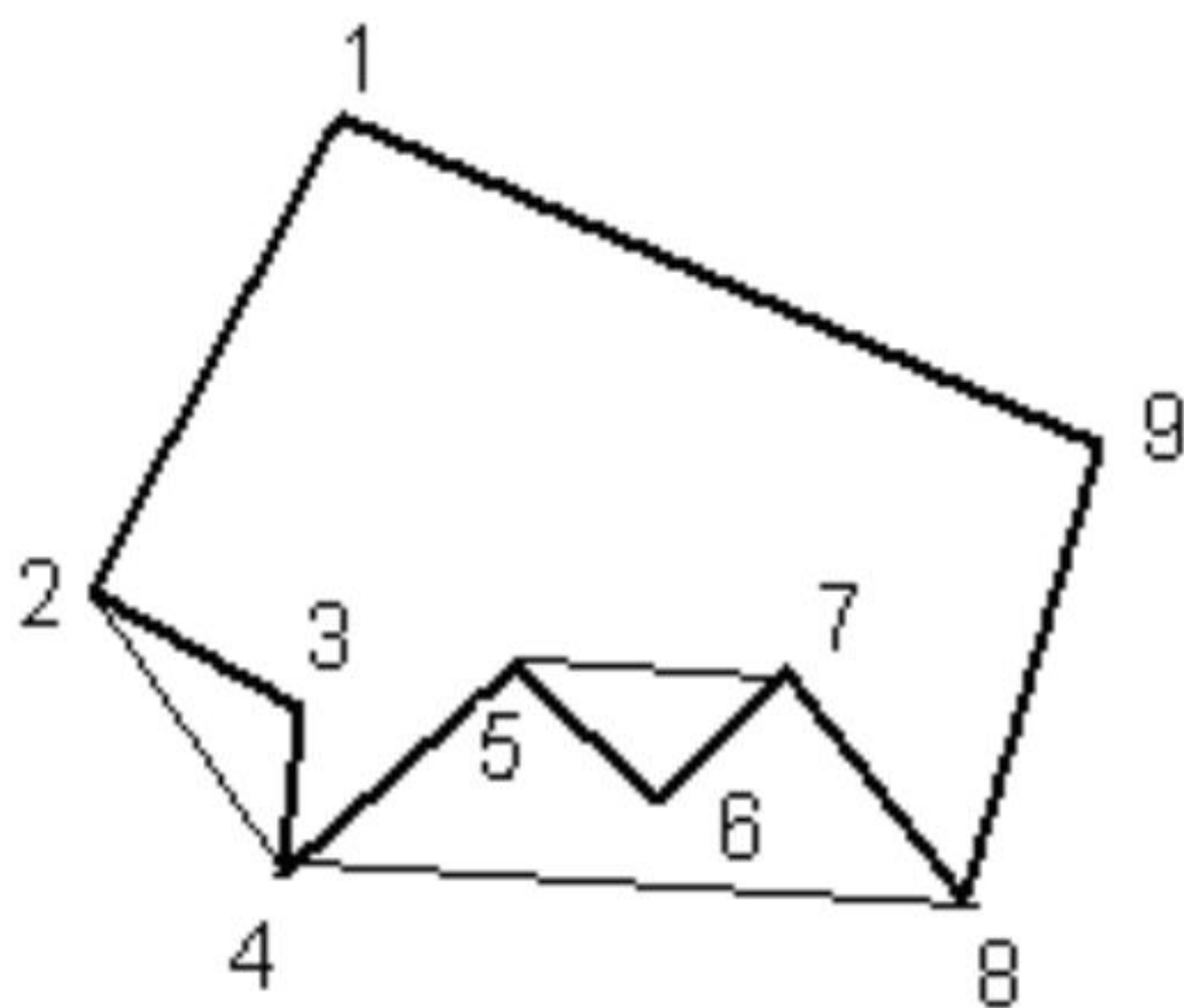
■ 非凸单轮廓线拼接

□ 多轮廓线拼接

最短对角线法



非凸单轮廓线拼接



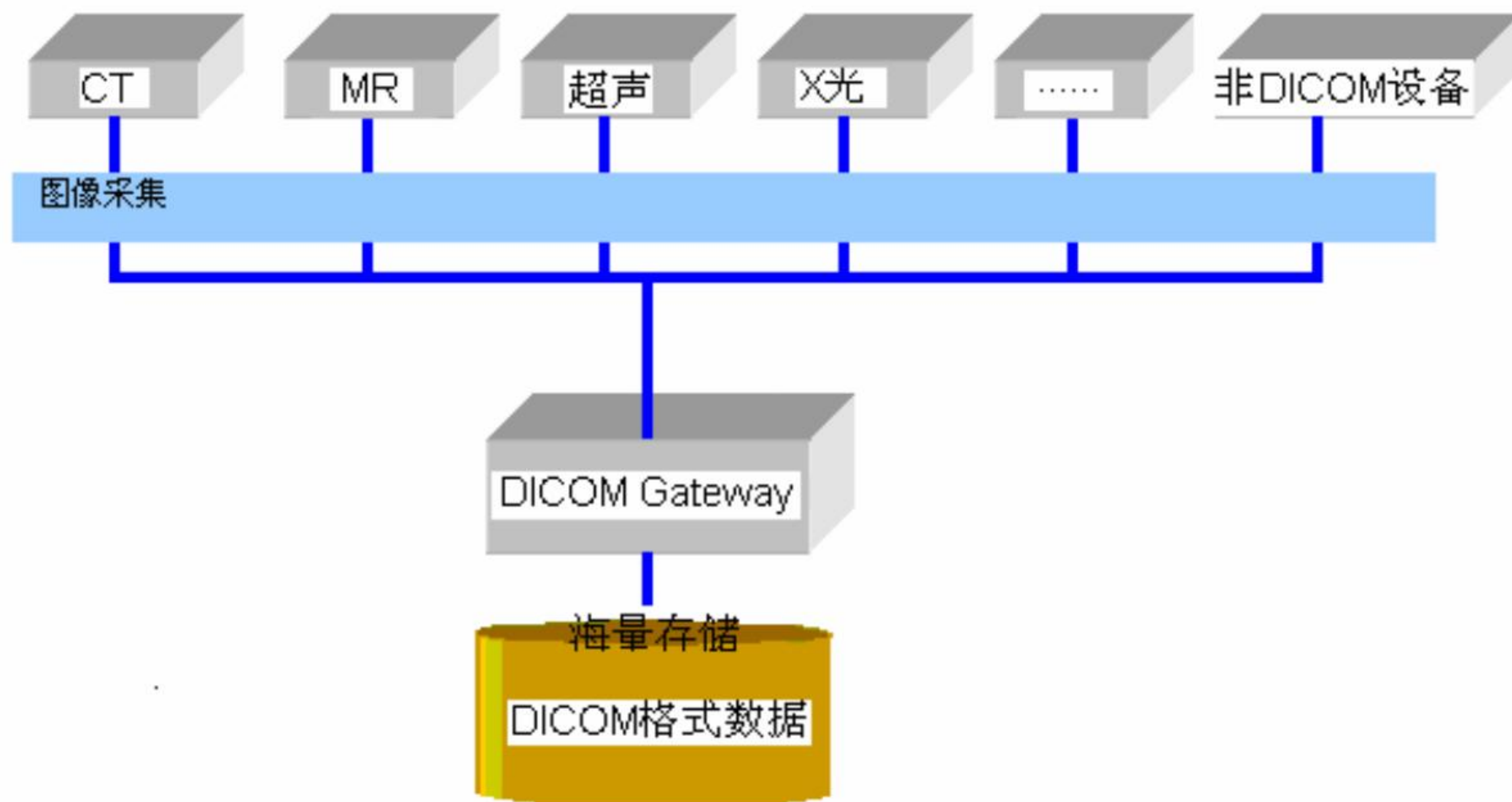
多轮廓线拼接

- 连接的困难
 - 形体重构转化成体数据中等值面构造
 - 体数据的构造
 - 场函数
 - 判断轮廓线内外的方法——扫描线算法
 - 表面生成
 - MC算法
-

三、 三维重建软件的设计与实现

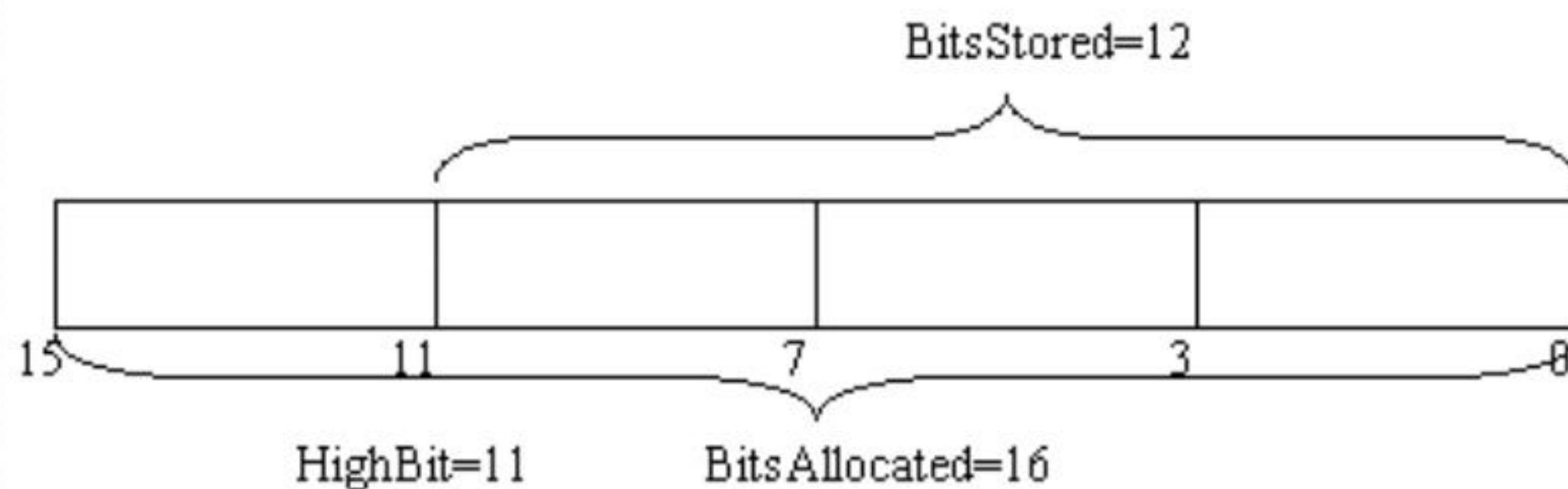
- DICOM网关
 - 高分辨率图像的显示方法
 - 轮廓线编辑
 - 封装OpenGL
 - 程序类图
 - 设计模式的使用
-

3.1 DICOM网关

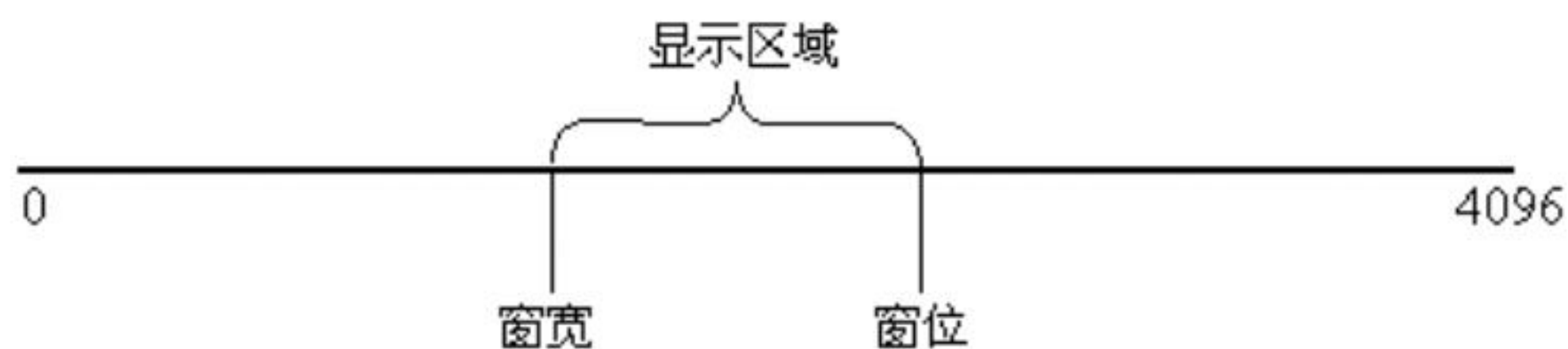


3.2 高分辨率数据的显示方法

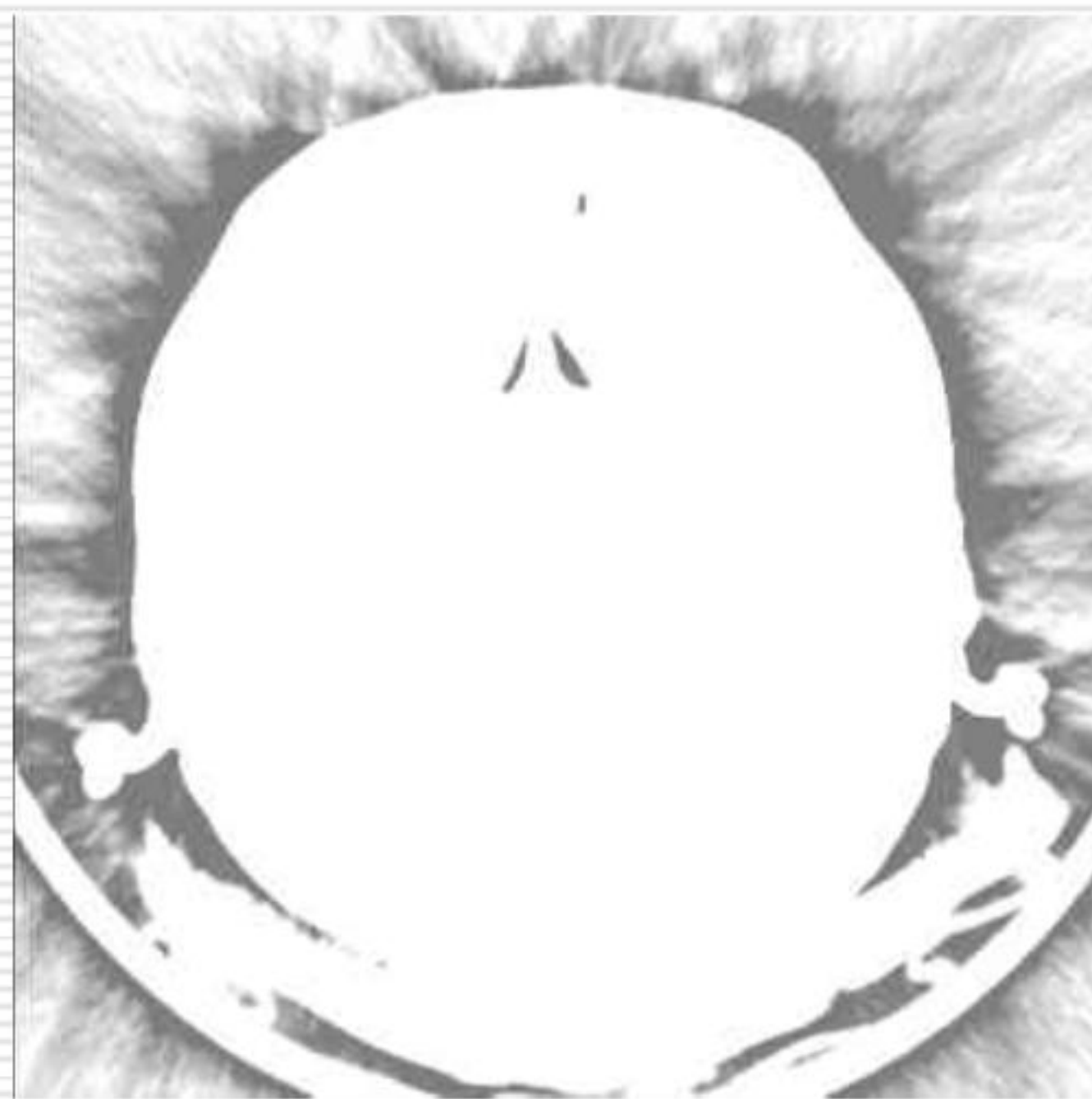
CT数据格式



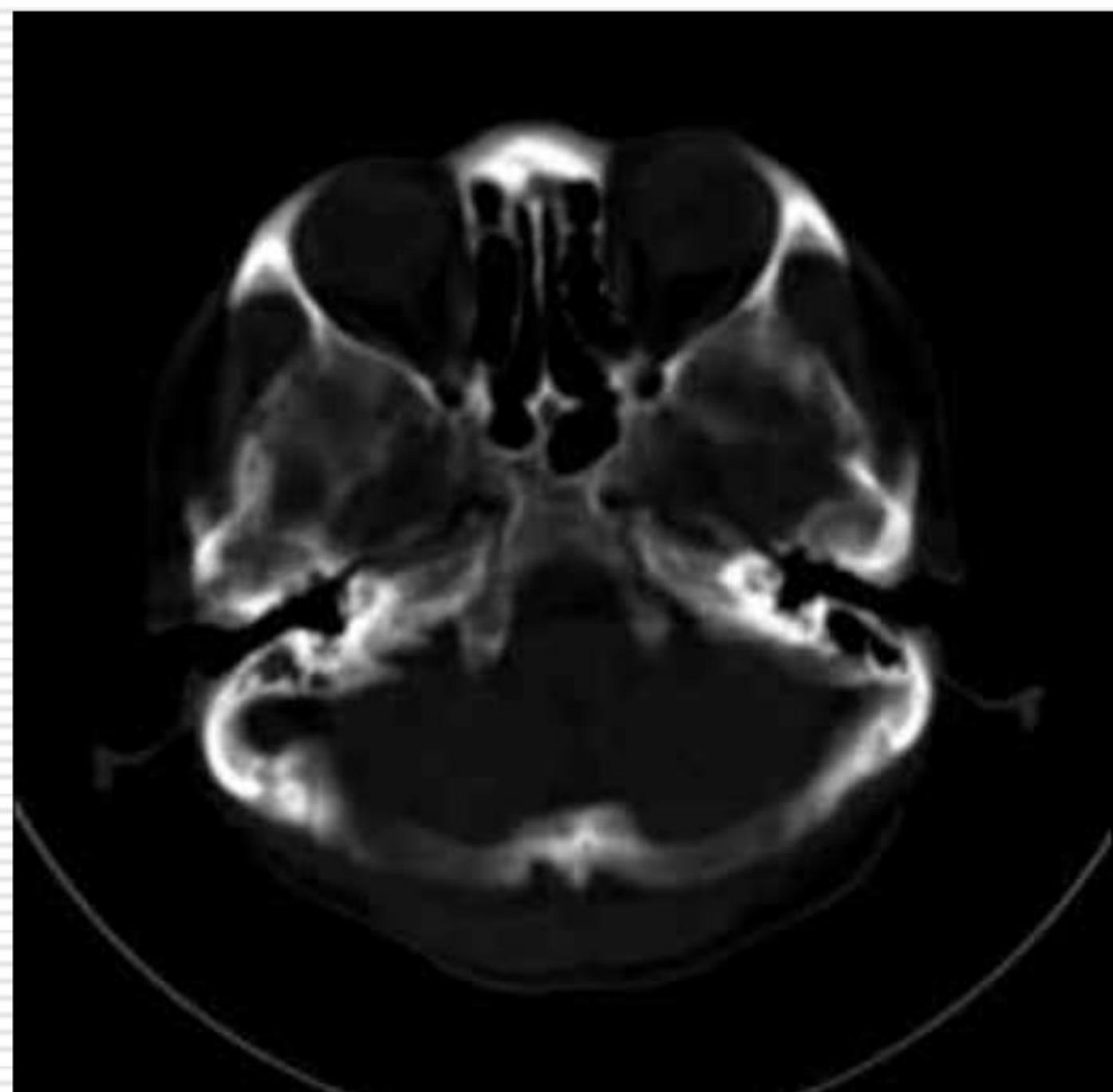
窗宽窗位



调低窗宽窗位



调高窗宽窗位



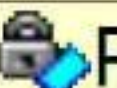
3.3 轮廓线编辑

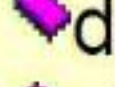
MultiPoly3d

 Poly2D
 errorRange
 imageWidth
 imageSrc

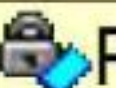
 updateLayer()
 UpdateAllLayer()
 setErrorRange()
 setImageWidth()
 setImageSrc()
 MultiPoly3d()
 Draw()

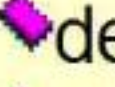
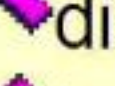
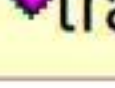
MultiPoly2d

 Polys

 MultiPoly2d()
 add()
 del()
 distance()
 draw()

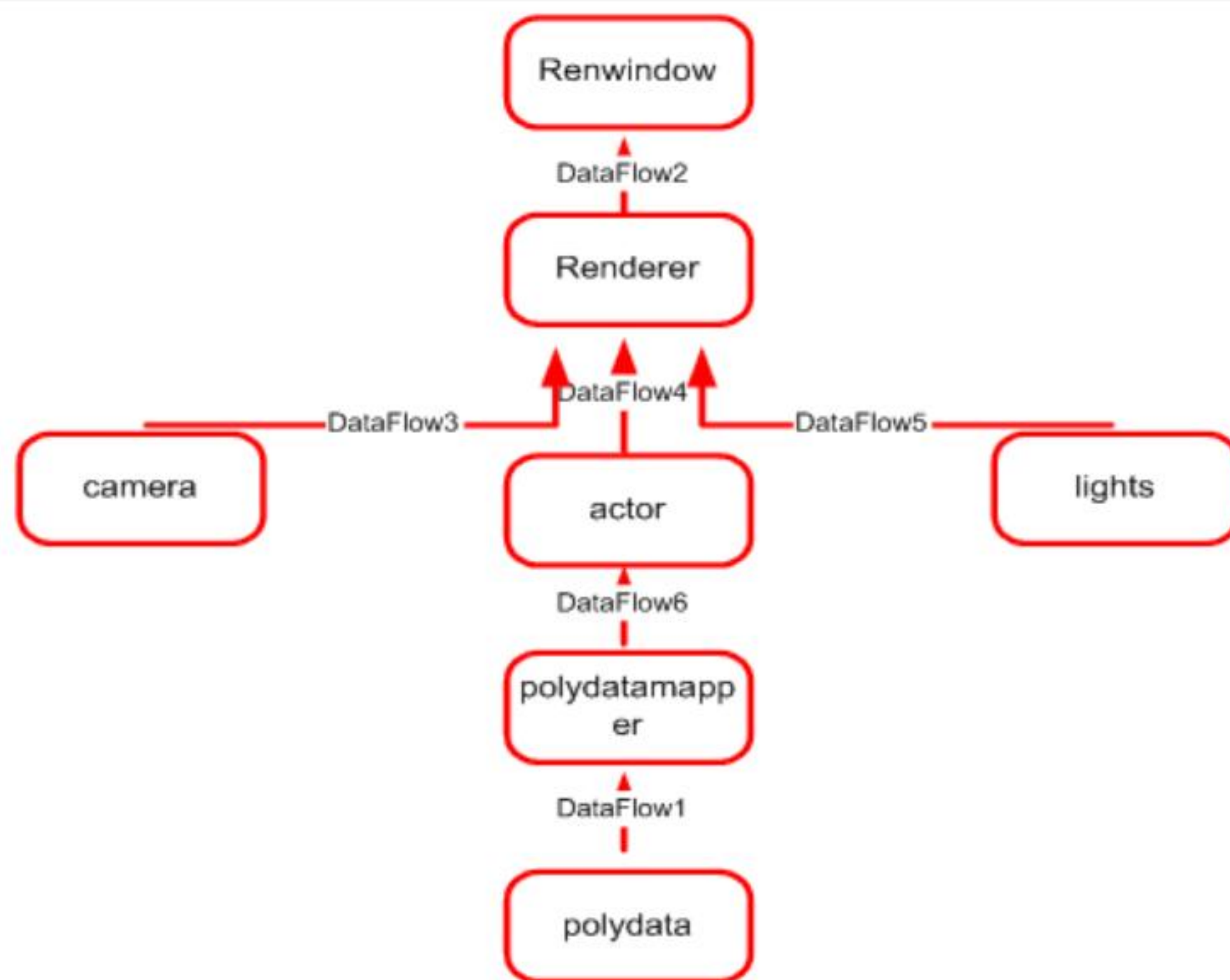
ClosedPoly2d

 Points

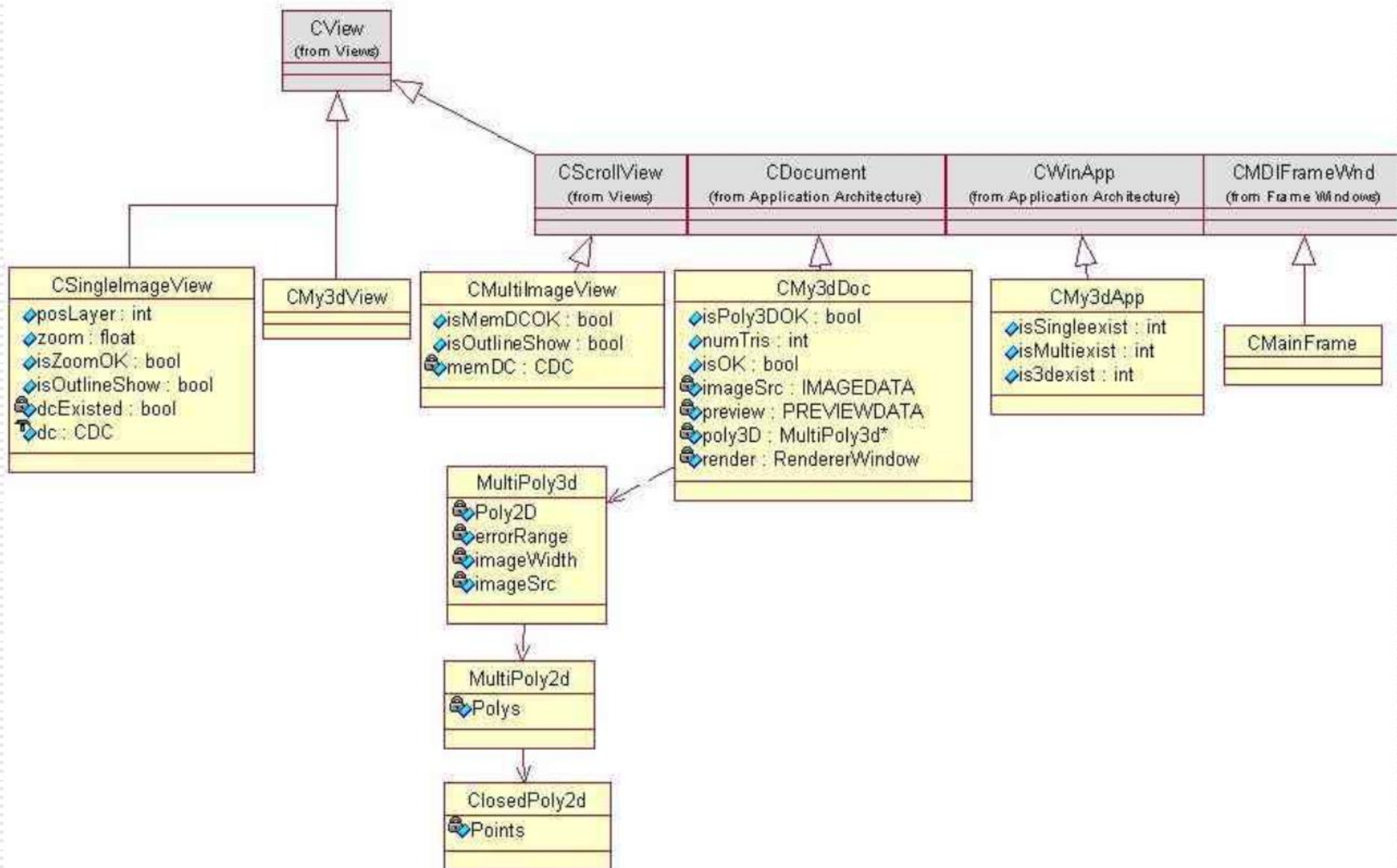
 add()
 del()
 distance()
 tranverse()

3.4 封装OpenGL

- 我们调用Renwindow的函数来请求显示三维模型，Renwindow设置一些相应的窗口属性，然后调用Renderer的相关函数
- Renderer负责转换世界坐标系、视点坐标系和屏幕坐标系，通过设置视点位置、光源和调用Actor的函数来实现把几何模型转换成图像的功能。
- Actor是渲染场景中的一项条目，根据对应的属性做适当设置后，调用PolydataMapper的相关函数。
- PolydataMapper负责解释三维模型中的点如何解释成为OpenGL基本命令里面的参数，几乎所有的OpenGL函数都出现在这个类中。



3.5 程序类图



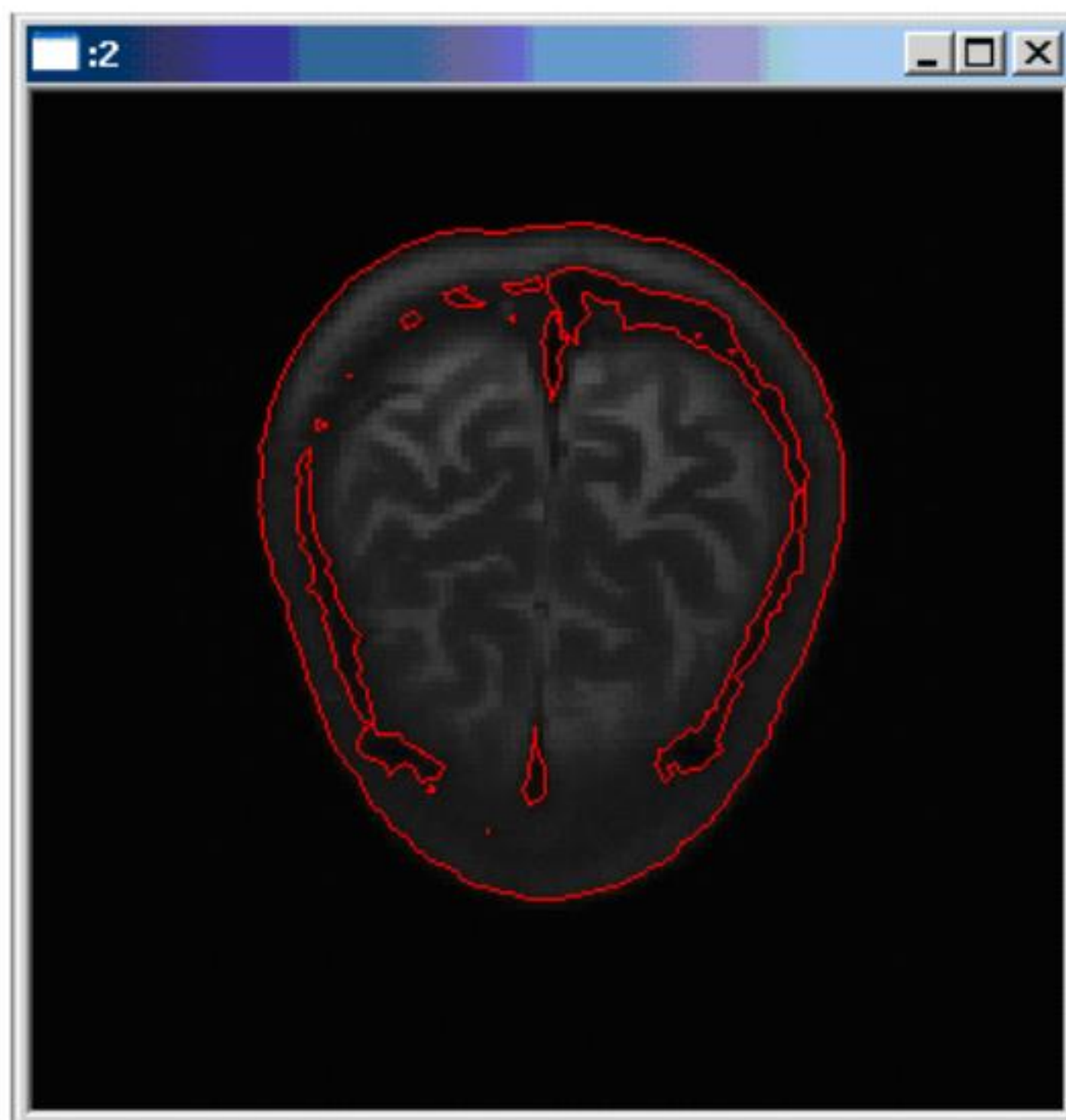
CSingleImageView

CSingleImageView
◆ posLayer : int
◆ zoom : float
◆ isZoomOK : bool
◆ isOutlineShow : bool
◆ dcExisted : bool
◆ dc : CDC
◆ CSingleImageView()
◆ RepaintUsingImage()
◆ prepareDC()
◆ deleteDC()
◆ drawMultiPoly2d()
◆ OnDraw()
◆ ~CSingleImageView()
◆ AssertValid()
◆ Dump()
◆ OnSingleoutlineShow()
◆ OnSingleoutlineHide()
◆ OnSinglePrev()
◆ OnSingleNext()
◆ OnZoomIn()
◆ OnZoomOut()
◆ OnSize()

(a) 类图



(b) 工具栏

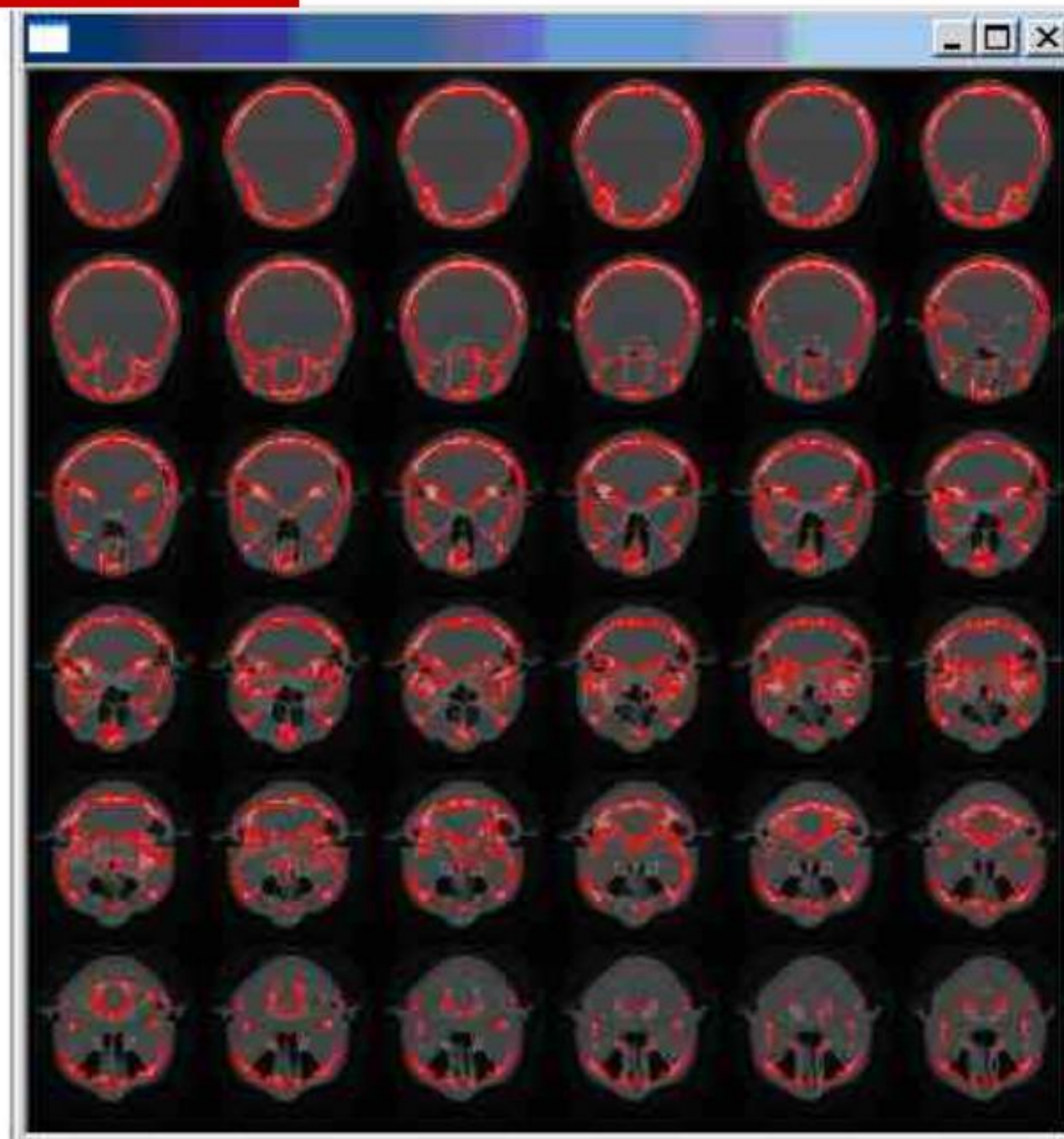


(c) 显示窗口

CMultiImageView

CMultiImageView	
isMemDCOK	bool
isOutlineShow	bool
memDC	CDC
CMultiImageView()	
prepareMemDCandDrawAllBitmap()	
DrawOneBitmap2MemDC()	
DrawAllOutline2MemDC()	
DrawOneOutline2MemDC()	
OnDraw()	
OnInitialUpdate()	
~CMultiImageView()	
AssertValid()	
Dump()	
OnLButtonDblClick()	
OnReconstruct()	
OnReconstruct2()	
OnTest()	
OnOutlineShow()	
OnOutlineHide()	

(a) 类图

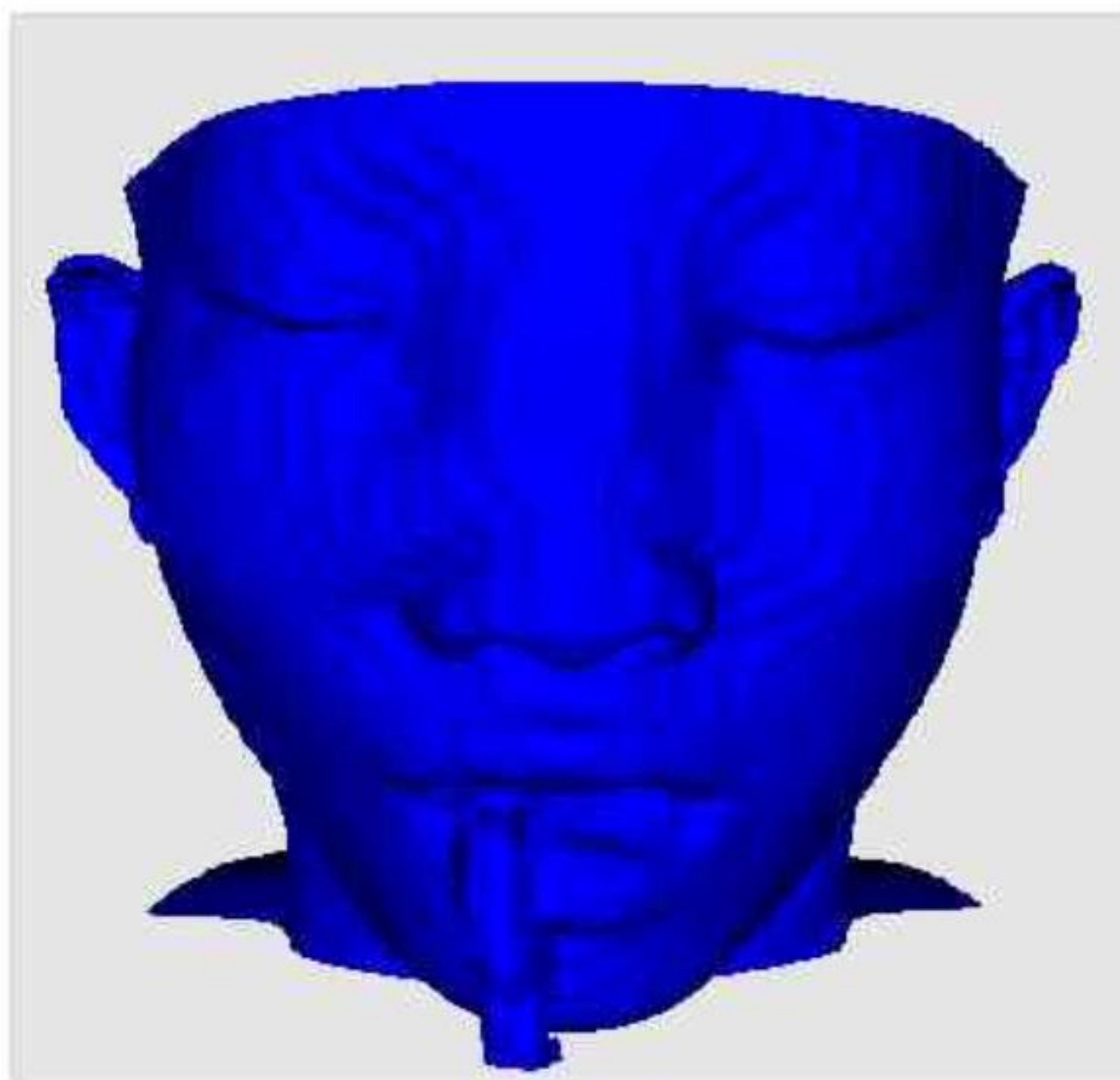


(b) 运行界面

CMy3dView



(a) 类图



(b) 运行窗口

图3.15 CMy3dView

3.6 设计模式的使用

- ❑ Singleton模式
 - ❑ Command模式
 - ❑ Strategy模式
 - ❑ Adapter模式
-

Singleton模式——代码示例

```
Class Singleton{
public:
    Singleton* Instance();
protected:
    Singleton();
Private:
    static Singleton*
    _Instance;
}
```

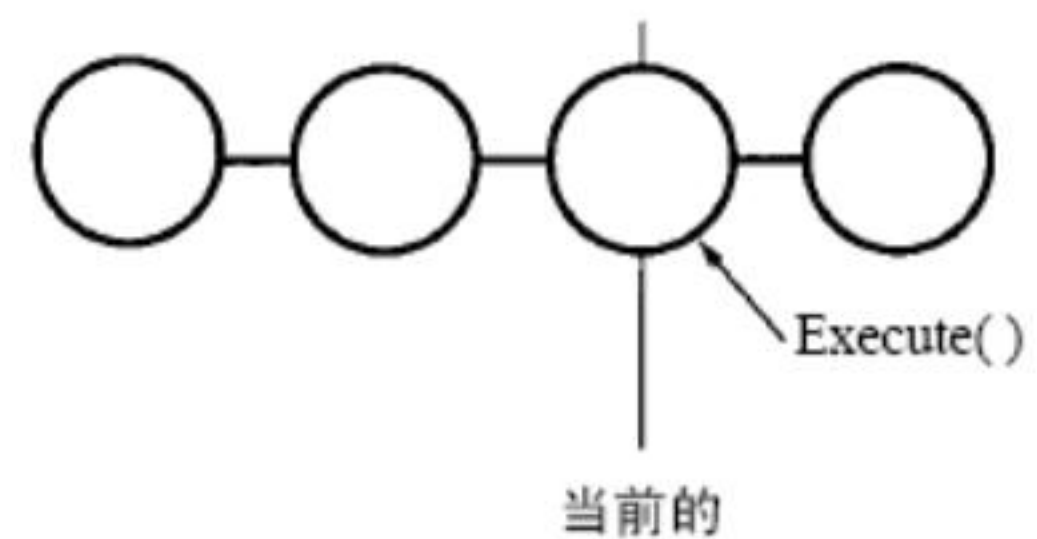
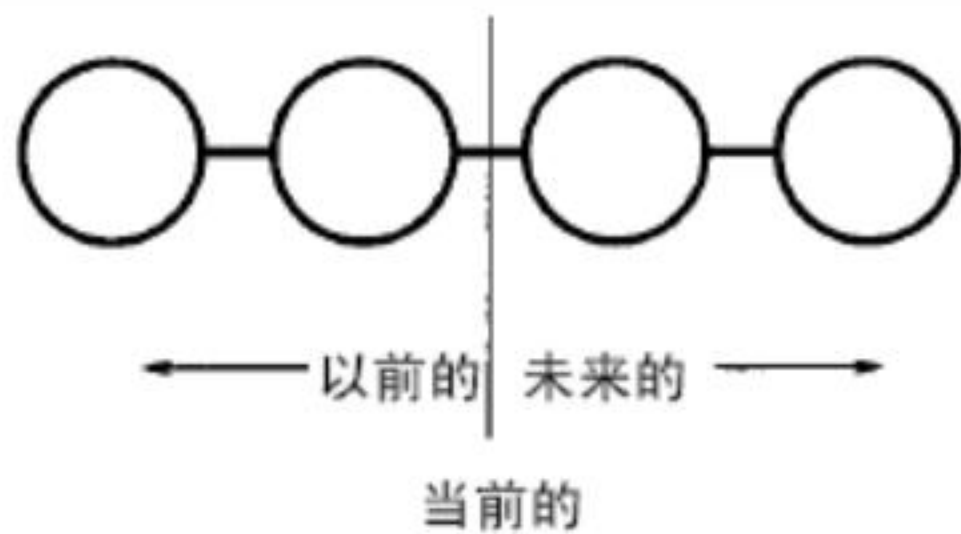
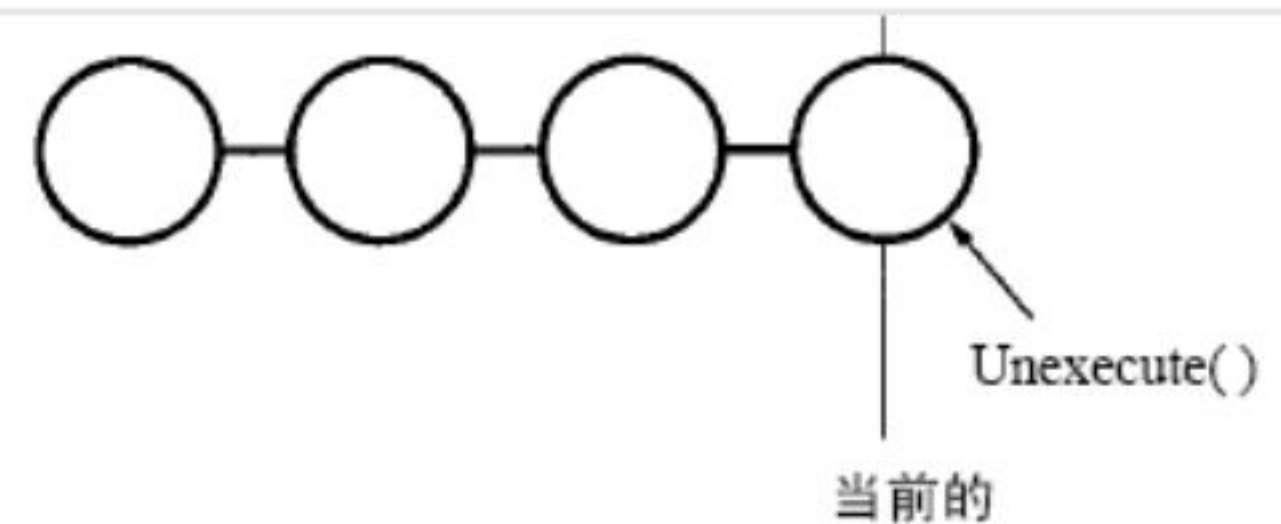
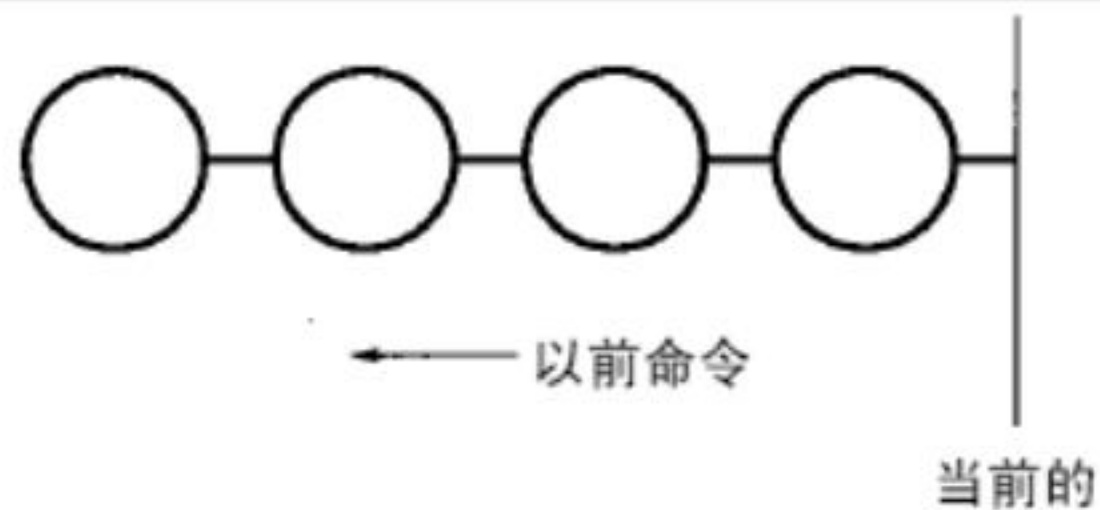
```
Singleton::_Instance = 0;
Singleton*
Singleton::Instance(){
    If( _Instance==0 ){
        _Instance = new
        Singleton;
    }
    return _Instance;
}
```

使用Singleton::Instance()来访问实例

Singleton模式——效果

	原来的代码	使用singleton模式后的代码
全局变量	<pre>MultiPoly3d *Poly3d; bool IsPoly3dExist=false;</pre>	不需要
每次要使用实例的地方 （注意程序中会出现多处， 所以代码简化非常有用）	<pre>MultiPoly3d *nowVisit; if(IsPoly3dExist) nowVisit = Poly3d; else nowVisit= new MultiPoly3d(); nowVisit->CallFun.....</pre>	<pre>MultiPoly3d::Instance()- ->CallFun...</pre>

Command模式



四、工作总结和展望

- 成功开发了医学影像三维重建软件**My3d**
 - 既支持自动得到重建模型又允许用户的介入
 - 允许用户在图像精度和运算时间间进行取舍
 - 功能完整，操作方便
 - 化简**MC**算法重构得到的三角面片
 - 解决了多轮廓线拼接的困难
-

四、工作总结和展望

- 模拟手术
 - 交互技术
 - 模型剖分和手术开窗
 - 图像分割的准确性
 - 场函数的选择
-

谢 谢 ！
