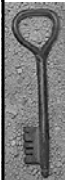




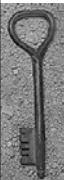
三维重建技术（面显示与体显示）介绍

田捷
中国科学院自动化研究所
www.3dmed.net
www.mitk.net



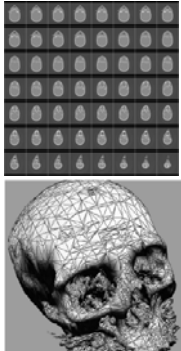
三维医学图像技术的意义


- ◆ 改变传统的阅片方式（二维到三维）
- ◆ 给医生提供真实感三维图形
- ◆ 任意角度观察
- ◆ 辅助医生临床诊断



三维重建的过程


- ◆ 输入：由CT、MR等设备扫描得到的一系列的两维切片数据
- ◆ 输出：组织（器官）的三维形状





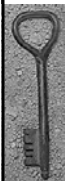
三维重建的分类

- ◆ 面显示（Surface Rendering）
- ◆ 体显示（Volume Rendering）



三维重建的分类——面显示

- ◆ 只提取感兴趣的某一种物质（如骨骼）
- ◆ 计算速度快，显示清晰
- ◆ 一般用密集的三角网格来表达
- ◆ 应用广泛（图形引导手术、虚拟内窥镜等）
- ◆ 可以实现多层的面显示，以观察整体效果



面显示的例子（骨骼）

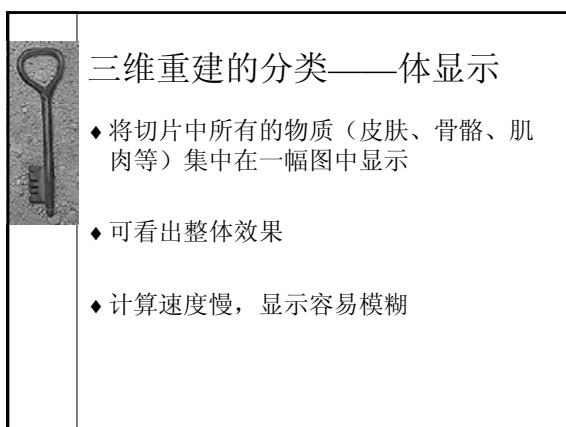




面显示的例子（皮肤）

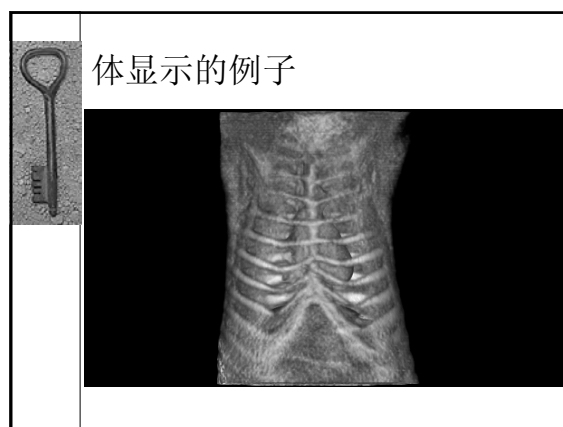


多层面显示的例子

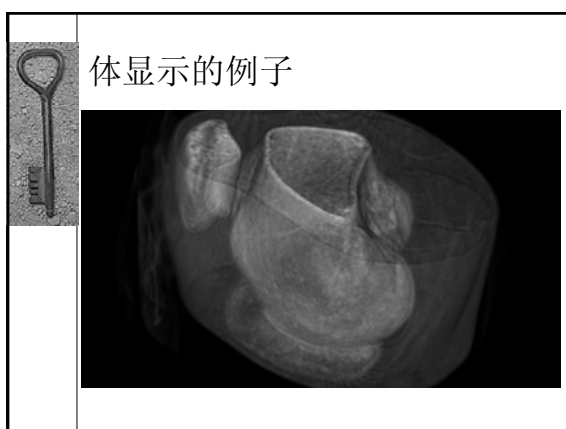


三维重建的分类——体显示

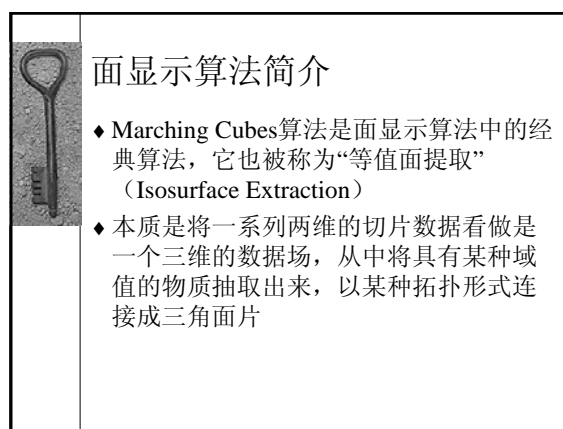
- ◆ 将切片中所有的物质（皮肤、骨骼、肌肉等）集中在一幅图中显示
- ◆ 可看出整体效果
- ◆ 计算速度慢，显示容易模糊



体显示的例子



体显示的例子



面显示算法简介

- ◆ **Marching Cubes**算法是面显示算法中的经典算法，它也被称为“等值面提取”（**Isosurface Extraction**）
- ◆ 本质是将一系列二维的切片数据看做是一个三维的数据场，从中将具有某种域值的物质抽取出来，以某种拓扑形式连接成三角面片

Marching Cubes算法描述

- 切片数据可以看做是一些网格点组成的，这些点代表了密度值

Marching Cubes算法描述

- 每次读出两张切片，形成一层 (Layer)

Marching Cubes算法描述

- 两张切片上下相对应的八个点构成一个Cube，也叫Cell, Voxel等

Marching Cubes算法描述

- 对一个Cube的8个顶点分别进行分类：
 - 顶点密度值 < 域值, Outside (1)
 - 顶点密度值 ≥ 域值, Inside (0)

Marching Cubes算法描述

- 由Cube的8个顶点的Inside (Outside) 状态得到一个0-255之间的索引值


v8	v7	v6	v5	v4	v3	v2	v1
INDEX							

Marching Cubes算法描述

- 由此索引值去查询一个长度为256的查找表，得到三角片三个顶点所在的边号

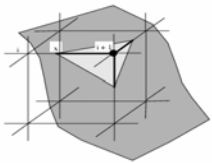
Case 4
INDEX = 01000001

triangle 1 = e1, e9, e4
triangle 2 = e6, e7, e12

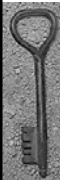


Marching Cubes算法描述

- 得到边号以后，在此条边上进行线性插值运算得到三角片顶点的坐标

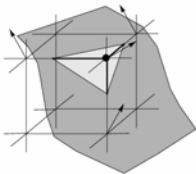


$$x = i + (\text{value} - D(i)) / (D(i + 1) - D(i))$$



Marching Cubes算法描述


- 法向量的计算也同样采用线性插值，首先使用中心差分法计算两个顶点的梯度值，然后由这两个梯度值插值得到三角片顶点的法向量



$$G_x = D(i + 1, j, k) - D(i - 1, j, k)$$

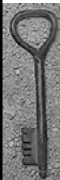
$$G_y = D(i, j + 1, k) - D(i, j - 1, k)$$

$$G_z = D(i, j, k + 1) - D(i, j, k - 1)$$



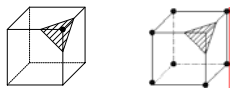
Marching Cubes算法描述


- 算法的难点：查找表的构造
 - 因为一个Cube有8个顶点，每个顶点有Inside和Outside两种状态，所以一个Cube里头三角片的分布总共可能有 $2^8=256$ 种组合
 - 如果手工去造表的话，不仅容易出错，而且工作量也太大
 - 反映了图形学算法的特点：容易理解，但是细节太烦琐



Marching Cubes算法描述

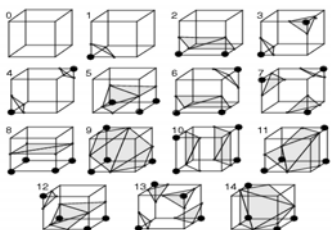
- Cube具有旋转（Rotation）对称性，旋转不影响等值面的拓扑结构
- 另外，所有的Inside变为Outside，同时所有的Outside变为Inside，则等值面的连接方式也不会改变（Inversion对称）



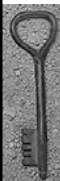


Marching Cubes算法描述

- 考虑到Rotation和Inversion对称两种情况后，可以用15种Basic Cube来覆盖所有256种可能的情况



- 由这15种Basic Cube可以很容易地构造出长度为256的查找表



Dividing Cubes算法简介

- 由于Marching Cubes算法得到的三角面片数量相当巨大，并且很多三角片投影到屏幕上以后小于一个像素的大小，因此Marching Cubes的作者又开发了一个更简化的算法：Dividing Cubes
- 绘制的基本元素由三角片变成了点，无需考虑拓扑结构
- 速度更快，尤其适合于医学图形领域
- 目前新兴的基于点的绘制（Points Based Rendering）为这一算法赋予了新的意义

Dividing Cubes算法描述

- 同Marching Cubes算法一样，处理的基本对象都是一个Cube（体现了分而治之的策略）

Dividing Cubes算法描述

- 对每个Cube进行分类
 - 所有的八个顶点都是Inside则为Interior Cube
 - 所有的八个顶点都是Outside则为Exterior Cube
 - 其它则为Surface Cube

Dividing Cubes算法描述

- 将Surface Cube在空间上分割为与最终图像解析度相同的Sub Cube
 - 如果数据集的规模为 $256 \times 256 \times 128$ ，而最终显示的窗口的大小为 512×512 ，则在x、y方向上分割一次，在z方向上分割两次
 - 新的Sub Cube的顶点密度值由三线性插值得到
 - 对于每个Sub Cube，继续进行上一步骤（分类）

Dividing Cubes算法描述

- 对于最终得到的Surface Cube，计算它的中心点的坐标、法向量（由梯度得到）
- 最终形成一个点集，每个点都具有法向量

Point Surface Representation

体显示算法简介

- 主要是研究光线在带颜色的、半透明的材质中传播的理论。

体显示算法简介

- 由光的传输理论，经过一定的简化处理，可以得到体显示的方程为：

$$L(x) = \int_x^{x_B} e^{-\int_x^{x'} \mu(x'') dx''} \epsilon(x') dx'$$

体显示算法简介

- ◆ 将体显示方程离散化后可以得到

$$L(x) = \sum_{i=0}^{n-1} c_i \cdot \prod_{j=0}^{i-1} (1 - \alpha_j)$$

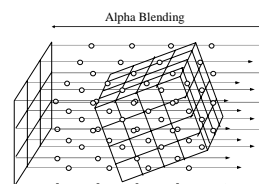
$$= c_0 + c_1(1 - \alpha_0) + c_2(1 - \alpha_0)(1 - \alpha_1) + \cdots$$

$$+ c_{n-1}(1 - \alpha_0) \cdots (1 - \alpha_{n-2})$$

$$= c_0 \text{ over } c_1 \text{ over } c_2 \text{ over } \cdots \text{ over } c_{n-1}$$

- ◆ 其中 $c_i = C_i \times \alpha_i$
 C_i ——颜色
 α_i ——阻光度

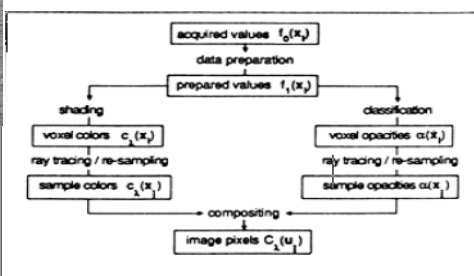
光线透射算法 (Ray Casting)



Ray Casting 的步骤

- ◆ 对投影平面上的每一个像素，投射出一条射线穿过体数据
- ◆ 在这条射线上按照一定的间隔采样，计算出采样点的颜色 (C_i) 和阻光度 (α_i)
- ◆ 按照体显示方程的离散化形式，进行颜色合成，计算出这个像素的最终颜色

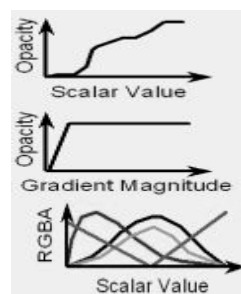
算法的整体流程

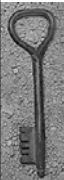


分类 (Classification)

- ◆ 目的：得到每个体素处的阻光度值
- ◆ 两类方法：
 - 由密度值和阻光度的传递函数 (Transfer Function) 决定
 - 用分割的方法 (这里不作介绍)

分类 (Classification)






明暗处理 (Shading)

◆ Phong光照模型

$$I = I_a K_a + I_d [K_d (N \cdot L) + K_s (N \cdot H)^n]$$


- K_a ——材质的环境光属性
- K_d ——材质的漫反射光属性
- K_s ——材质的镜面反射光属性
- n ——材质的光滑程度
- I_a ——光源的环境光入射强度
- I_d ——光源的反射光入射强度
- N ——体素的法向量 (用梯度来计算)
- L ——光源的方向矢量
- H ——入射光和视线夹角一半处的矢量



明暗处理 (Shading)

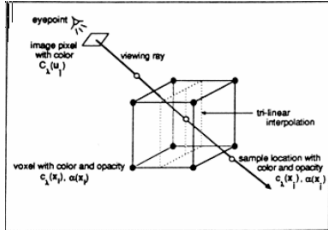
◆ Gouraud明暗处理

- 对于一个面，只计算顶点处的光照强度 (按照Phong光照模型)
- 对于面内部的点，按照扫描线顺序进行光照强度的线性插值



重采样滤波器

◆ 一般采用三线性插值 (tri-linear interpolation)





结束

◆ 谢谢!