

文章编号:1005—0523(2006)02—0062—05

基于灰度形态学的图像预处理及其三维重构研究

詹学峰, 蒋先刚, 胡晓燕

(华东交通大学 交通信息工程及控制研究所, 江西 南昌 330013)

摘要:对三维表面重构所用图片的预处理、三维重构 Marching Cubes 算法及其算法的具体编程实现进行了详细地讨论和验证, 其相关技术在工程软件设计中获得了预计的结果.

关键词:灰度形态学; 三维重构; Marching Cubes

中图分类号:TP391.41

文献标识码:A

1 概述

长期以来, 临床医生对病情只能凭借经验和有限的影像资料做出判断, 缺乏直观而具体的样本模型, 因而带有一定的盲目性和片面性. 而且, 简单地凭借经验和二维医学影像资料对器官实体及病灶形状进行空间想象, 其判断误差是不可避免的, 容易造成对病症的误诊. 为了解决这一系列问题, 就有了“医学器官三维重构”这一概念. 医学器官三维重构就是从一系列二维的断层影像图片中重构出组织或器官的三维影像的过程, 目的是为了给医生一个直观动态的虚拟解剖实体, 便于对病症的正确诊断.

一般通过 CT、MRI 机扫描得到的图像, 由于设备以及图像自身的原因, 经常会有一些噪声. 这些噪声点会对图像重构的真实性和整体性造成影响. 因此, 在进行三维重构之前, 有必要对原始扫描图片进行预处理. 针对所处理的图像的特点, 图像预处理的目标是既要去除噪声, 又要充分保留原始图像中的整体信息. 在数字图像处理中, 经常用到的滤波方法有中值滤波、均值滤波, 也可以在频率域滤波进行处理. 在此, 我们主要运用了灰度形态学处理的方法, 并且取得了满意的效果.

2 图像灰度形态学预处理技术

数学形态学的基础是集合论. 通过集合论的术语以及

图形描述就可以表示形态学的各种处理过程. 基本的形态学图像处理主要包括图像的膨胀、腐蚀、开运算、闭运算. 下面我们将详细叙述三维重构所需图片的前置处理技术.

形态学处理表现为一种邻域运算的形式. 我们把一种特殊定义的邻域称之为“结构元素”(Structure Element), 在每个像素位置上, 它与图像对应的区域进行特定的运算, 运算的结果为相应像素的输出图像. 形态学运算的效果取决于结构单元的大小、内容以及逻辑运算的性质. 结构单元在形态学处理中的作用相当于信号处理中的“滤波窗口”. 一般的形态学图像处理的对象是二值图像, 而将这个概念扩展到灰度图像, 即用灰度形态学处理技术, 将达到更佳图像预处理效果. 假设在此处理的是可以表示成 $f(x, y)$ 和 $b(x, y)$ 的数字图像函数, $f(x, y)$ 是输入图像, $b(x, y)$ 是结构元素. $b(x, y)$ 在某种意义上可以看成是一个子图像函数, 而 f 和 b 代表图像中每一个 (x, y) 坐标的灰度值.

1) 膨胀(Dilation)

用结构元素 b 对函数 f 进行的灰度膨胀表示为: $f \oplus b$, 其定义为:

$$(f \oplus b)(s, t) = \max \{ f(s-x, t-y) + b(x, y) \mid (s-x, t-y) \in D_f; (x, y) \in D_b \}$$

其中 D_f 和 D_b 分别是 f 和 b 的定义域.

从上式中可以看出, 如果结构元素 $b(x, y)$ 为正值, 图像的灰度值趋向于变大. 所以输出图像会比输入图像看上去更加明亮, 并且能减弱或者消除一些比结构元素面积小的、暗的细节部分. 图像明亮的程度与结构元素的大小以及形状

收稿日期:2005—10—15

基金项目:江西省教育厅 2004 年科研基金资助项目.

作者简介:詹学峰(1981—), 男, 江西九江人, 华东交通大学在读研究生, 主要研究领域:工业监控、数字图像处理.

有很大关系.

2) 腐蚀(Erosion)

用结构元素 b 对函数 f 进行的灰度腐蚀表示为: $f \ominus b$, 其定义为:

$$(f \ominus b)(s, t) = \min\{f(s + x, t + y) - b(x, y) \mid (s + x, t + y) \in D_f; (x, y) \in D_b\}$$

其中 D_f 和 D_b 分别是 f 和 b 的定义域.

从上式中同样可看出,如果结构元素 $b(x, y)$ 为正值,图像的灰度值趋向于减小,所以输出图像会比输入图像看上去更加阴暗,并且能减弱或者消除一些比结构元素面积小的、亮的细节部分.图像阴暗的程度同样取决于结构元素的大小及形状.

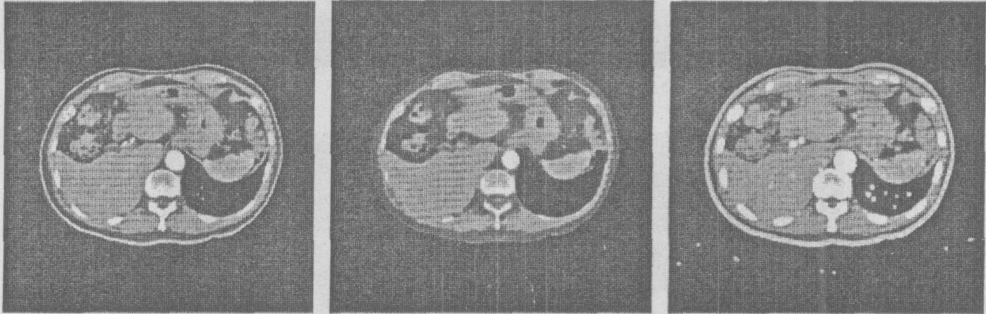


图 1 原始图像 图 2 腐蚀后的图像 图 3 膨胀后的图像

3) 开运算(Opening)

用结构元素 b 对函数 f 进行的开运算表示为: $f \circ b$, 其定义为:

$$f \circ b = (f \ominus b) \oplus b$$

开运算操作就是对图像先进行腐蚀后膨胀.先进行的腐蚀可以去除一些小的亮的细节同时会让图像更加阴暗;其后的膨胀操作又去除一些小的暗的细节,并且会增加图像的亮度,但不会引入前一腐蚀操作所去除的部分.其方法经常用于去除较小的明亮细节,而且能够相对地保持整体的灰度级和较大的明亮区域不变.可以用来消除细小物体、平滑较大物体的边界的同时并不明显改变其面积.

沿着 $f(x, y)$ 曲面的上侧面滚动.经过开运算, $f(x, y)$ 曲面中所有比结构元素范围小的波峰和尖锐程度都减小了,从而能够删除图像中在尺度上小于结构元素的几何结构,起到了平滑图像的外凸边缘的作用;经过闭运算, $f(x, y)$ 曲面中所有比结构元素范围小的波谷都被平滑,从而具有填充图像中那些比结构元素小的断裂或凹入部分的功能,起到了平滑图像的内凸部分的作用.所以从直观上看,灰度图像的开运算和闭运算具有聚类功能,即能够把图像中某个范围内灰度相近、数量占优势的图像信息聚到一起,从而使得图像信息不易丢失.

4) 闭运算(Closing)

用结构元素 b 对函数 f 进行的闭运算表示为: $f \bullet b$, 其定义为:

$$f \bullet b = (f \oplus b) \ominus b$$

闭运算操作就是对图像先进行膨胀后腐蚀.先进行的膨胀可以去除一些小的暗的细节同时会让图像更加明亮;其后的腐蚀操作又去除一些小的亮的细节,并且会减小图像的亮度,同样不会引入前一膨胀操作所去除的部分.其方法经常用于去除图像中的暗细节部分,而相对地保持明亮部分不受影响.可以用来填充物体内部细小空洞、平滑其边界的同时并不明显改变其面积.

在灰度级形态学中实现图像平滑处理的一种途径是先使用形态学开运算操作,而后进行闭运算操作.进行这两种操作后的最终结果是除去或减少了局部亮的和暗的影响因素或噪声.

相应的各种图像处理方式的实现效果如上图所示.作为对比,图 8 和图 9 给出了均值滤波和中值滤波的效果.我们在此采用的结构元素 $b(x, y)$ 是一个具有单位高度的 5×5 像素的平行六面体,使用上述方法对一个灰度图像进行处理.从五种形态学操作的效果图中可以看出,腐蚀操作缩小了图像中物体的轮廓且图像变得更暗,图像中较暗的空洞也变大了;膨胀操作扩大了物体的轮廓且图像变得更亮,图像中的许多空洞已经没有了;经过开运算操作,图像中那些小的、明亮细节尺寸变小了而暗的、灰度的效果没有明显变化,因此图像略有变暗且轮廓基本没变;经过闭操作,小的、暗的细节尺寸变小了而明亮的部分的效果没有明显变化,因此图像略有变亮但轮廓基本没变;经过形态学平滑操作,图像中那些小的明暗细节尺寸都减小了,有效地填充了原始图像中一些小的空洞,并且明亮程度和轮廓都基本与原始图像一样.

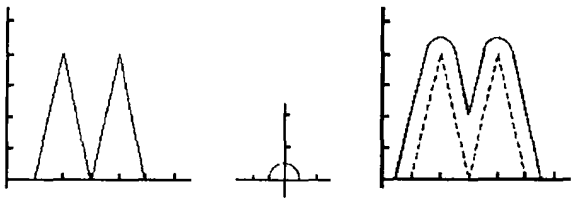


图 4 一般函数膨胀示意图

灰度图像的开运算和闭运算在几何上可以理解为把由结构元素 $b(x, y)$ 表示的曲面沿着由 $f(x, y)$ 表示的曲面滚动,开运算是沿着 $f(x, y)$ 曲面的下侧面滚动,而闭运算是

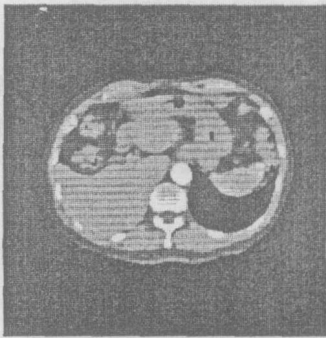


图 5 开运算后图像

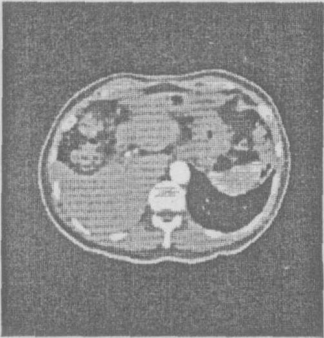


图 6 闭运算后的图像



图 7 平滑处理后的图像



图 8 均值滤波后的图像

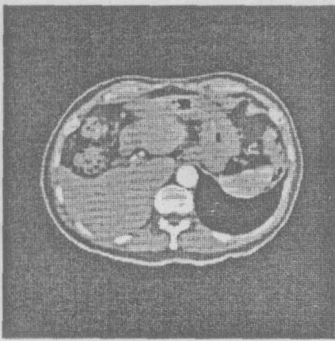


图 9 中值滤波后的图像

原始图像中一般存在许多噪声,噪声的存在将极大影响三维重构的效果.为获取适合进行三维重构的图像,一般对处理后的图像有如下要求:(1)填充图像中一些细小空洞;(2)保持物体的轮廓基本不变;(3)去除图像中一些多余噪声点.一般的滤波处理只是为了减少随机噪声平滑图像,使得图像出现不同程度的模糊.这就不能完全满足要求,特别是填充细小空洞,因此容易丢失切片图像的整体信息.所以,一般的处理是针对二维图像,不大适合重构.而灰度形态学中的平滑要经过开运算和闭运算,分别减小了图像中小的明暗的细节,在很大程度上填充了那些细小空洞,有利于进行良好的重构.

3 三维重构

三维空间数据场的描述有两类可视化算法,一类是直

接体绘制算法,即不构造中间几何图元而直接在屏幕上绘制;另一类是首先在三维空间数据场中构造出中间几何图元,再由传统的计算机图形学技术实现画面绘制.我们在此使用第二类表面重建算法,图像的三维重构主要采用 Marching Cubes 算法,并且通过 OpenGL 显示三角面片来实现.

3.1 Marching Cubes 算法介绍

设想被重构物体所在的三维空间是由一个个体元组成.我们将断层切片集分层读入内存,其中每相邻的两层数据构造多个立方体形状的体元(如图 10).由于每个立方体有 8 个顶点,故我们对这 8 个顶点进行采样,可以得到它们在对应断层切片上的灰度值.通过判断顶点灰度值的大小就可以得知物体表面与哪些体元相交,或者说穿过哪些体元.如果相交,则通过插值的方法获得物体表面与体元相应边的交点.然后求出相应三角面片上的法向量,最后通过 OpenGL 对该三维形体进行重构.

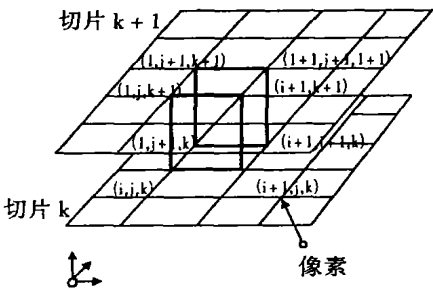


图 10 体元的构造及顶点采样

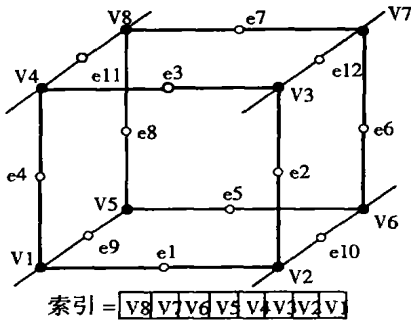


图 11 等值三角片的定义

取一个立方体为例(如图 11),假设整个立方体只有点 v_1 在被重构物体表面内,则物体表面经过了该立方体,并且近似地可以用三角形片 $\triangle e_1e_4e_9$ 取代.不同的相交情况就用不同三角面片列取代,显然,由于立方体八个顶点,每个顶点有两种状态(要么在表面内,要么在表面外),这样就有 $2^8=256$ 种情况.利用立方体的对称性将这 256 种情况简化为 14 种.

通过依次分析组成对象空间的所有体元,从而获得每一个体元与被重构物体表面之间的关系.根据上述等值面与体元相交的 14 种情形,得出所有发生相交情况的体元.连接所有这些相邻的体元内的三角面片,就基本形成了物体表面.然后对这些三角形片进行绘制,就可以把三维形体显示出来了.每个立方体内的三角片构成是通过连接各边交点来实现的,各边与原始图形表面的交点要用线性插值的方法得到.

3.2 Marching Cubes 算法的程序实现

基于 MC 算法三维重构的实现主要有以下几步:
1) 构造由体元组成的三维空间.
根据切片的大小及其数量,分别在 xyz 方向的 $(-1, +1)$ 范围内构造三维空间.

```
For cy := 0 to GridSizeY do //GridSizeY 为切片数量
For cz := 0 to GridSizeXZ do //GridSizeXZ 为切片大小
For cx := 0 to GridSizeXZ do Begin //GridSizeXZ 为切片大小
Grid[cx, cy, cz].P.X := 2 * cx / GridSizeXZ - 1; // x 方向从 -1 到 +1
Grid[cx, cy, cz].P.Y := 1 - 2 * cy / GridSizeY; // y 方向从 -1 到 +1
Grid[cx, cy, cz].P.Z := 2 * cz / GridSizeXZ - 1; End; // z 方向从 -1 到 +1
```

其中,Grid 三维数组用于构造三维空间.然后由此三维空间里每 8 个相邻的点,按照如图 11 所示体元的顶点排列顺序,构造我们所需要的体元.标定好各个体元之后,我们就可以根据下面介绍的索引表进行三角面片排列.

2) 构造三角片排列索引表 LookUpTable 文件.
在 LookUpTable 文件中定义了两个数组 EdgeTable 和 TriangleTable.其中,EdgeTable 数组代表按照图 11 所示规则的所有 256 种等值面与体元相交的情况.例如 '\$109'=000100001001 表示图 11 中体元的 e_1 、 e_4 、 e_9 边分别被物体表面所截.TriangleTable 数组则相应地记录了每一种相交情况的三角面片的构成.

```
EdgeTable: array [0..255] of Integer //记录 256 种情况的截点所在的边
{ $0, $109, $203, $30a, $406, $50f, $605, $70c,
$80c, $905, $a0f, $b06, $c0a, $d03, $e09, $f00,
$190, $99, $393, $29a, $596, $49f, $795,
```

```
$69c,
$99c, $895, $b9f, $a96, $d9a, $c93, $f99,
$e90, ... ..}
TriangleTable : array [0..255, 0..15] of Integer //记录 256 种情况三角形的构成
{ (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1),
( 0, 8, 3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1),
( 0, 1, 9, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1),
( 1, 8, 3, 9, 8, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1),
-1, -1, -1), ... ..}
```

3) 判断各个体元与表面是否相交.
下面为某体元中第一个顶点的灰度采样值与等值面进行比较的程序段.

```
If (GridCube.GridPoint[0].Value <= SelectPart.MaxValue)
and (GridCube.GridPoint[0].Value >= SelectPart.MinValue) then
CubeIndex := CubeIndex or 1;
如果有相交情况出现的话,利用 EdgeTable 数组查找体元的各个边与被重构物体表面相交的情况,再通过插值公式计算截点信息.
If (EdgeTable[CubeIndex] and 1) <> 0 then VertList[0] := Interpolate(GridPoint[0].P, GridPoint[1].P, GridPoint[0].Value, GridPoint[1].Value);
最后根据由 TriangleTable 数组对应的三角面片信息,由 OpenGL 将所有出现的三角面片显示出来.
```

3.3 运行结果及对比分析

从三维重构的线框模式中,我们可以很清楚地看到该器官形体的各个部分都是由多个三角面片组成的.而从重构的效果可以看出,经过灰度形态学平滑后产生的三维形体比之平滑前的少了许多杂质部分,并且有效地保持了基本的整体轮廓和形态.这便于我们更清楚有效地辨认以及判断医学器官的实际构成,从而做出正确的诊断分析.

4 结束语

通过对医学图像的三维重构实现,我们可以得到更加直观的效果,比只看二维图片能获得更多的信息.在原始切片预处理过程中,由灰度形态学的方法,通过使用不同的结构元素对灰度图像进行处理,就能够得到较好的效果,有效地起到平滑图像的作用,从而可能对图像的三维重构产生更好的效果.由于采用的是基本的 MC 算法,下一步我们有必要对算法进行改进,尽量减少程序的计算冗余,缩短重构时间,提高重构效率.

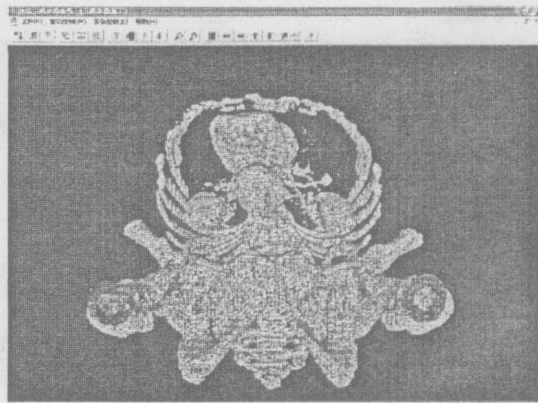


图 12 平滑前(线框模式)

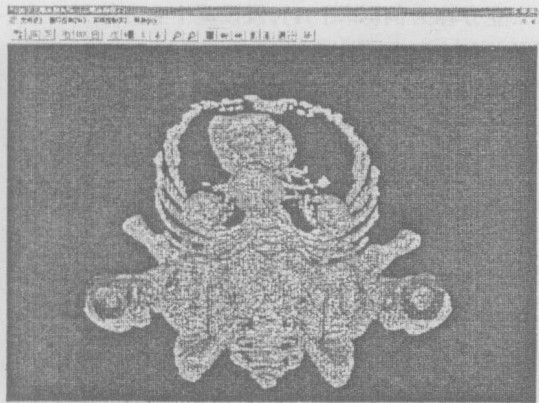


图 13 平滑后(线框模式)

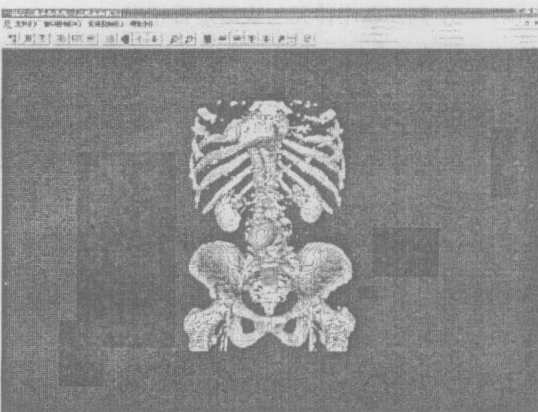


图 14 平滑前重构效果



图 15 平滑后重构效果

参考文献:

[1] 费广正,等. 可视化 OpenGL 程序设计[M]. 北京:清华大学出版社,2001.

[2] David F·Rogers. 计算机图形学算法基础[M]. 北京:电子工业出版社,2002.

[3] 唐泽圣,等. 三维数据场可视化[M]. 北京:清华大学出版社,1999. 12.

[4] 彭 波,崔永普,等. 用于数字监控系统的图像去噪算法的研究与实现[J]. 北京:中国农业大学学报,2004, 9 (5).

[5] 王树文,闫成新,等. 数学形态学在图像处理中的应用[J]. 北京:计算机工程及应用,2004,23.

Research on Preprocessing of Slices Based on Gray Morphology and 3D Reconstruction Technology

ZHAN Xue-feng, JIANG Xian-gang, HU Xiao-yan

(Information & Control Engineering Institute, East China Jiaotong University, Nanchang 330013, China)

Abstract: This paper discusses the technologies of slices' preprocessing before 3D reconstruction, 3D surface reconstruction algorithm—"Marching Cubes" and the programming implementation of the algorithm. It reaches intending result in the testing software.

Key words: gray Morphology; 3D reconstruction; marching cubes