改进 Marching Cubes 算法的肿瘤靶区三维重建与可视化

摘要—放疗是利用射线治疗恶性肿瘤的一种方法，目前治疗癌症最有效的手段之一。放疗的治疗和保护区域，一般都是存储在Modality是RTStruct的DICOM文件中，通常存储为包含在均匀分布的横截面图像（切片）上的 2D 轮廓。表面重建算法在肿瘤放射软件中实现，这样可以实现治疗和保护区域的三维展示，有助于医生更加直观、准确的对病变进行诊断，方便医生查看去制定更可靠的治疗计划，具有重要的临床意义。本文使用来自北京肿瘤医院的三位患者的三个不同部位的医学影像数据用于算法测试。Marching Cubes算法是医学影像三维重建中最广泛使用的等值面算法，但没有考虑相邻立方体中等值面之间的关系会导致重建的三维模型中有孔洞，本文通过改进Marching Cubes算法，先选择种子三角形，再依据体表面具有连通性理论，生长选出来的所有种子三角形，根据这些三角形重建出整个3D轮廓，采用中点选择法代替线性插值法,进一步提高三维重建的速度，由可视化工具包 VTK 和 OpenGL 借助 GPU 并行处理，完成勾画肿瘤区域的快速、精准三维重建与可视化，算法的执行效率整体提高。

关键词— Marching Cubes algorithm; 肿瘤靶区, 三维重建 可视化

1、介绍

放疗技术在肿瘤治疗方面具有不可或缺的应用地位。它是使用高能射线（如X射线或质子）来杀死或控制体内异常细胞的方法。在放射治疗中，靶区是指肿瘤或需要接受放射治疗的区域，它决定了放射束的照射范围，靶区勾画是非常重要的步骤，准确勾画靶区对于确保放疗的精准性和安全性至关重要。在放疗技术中医生会使用各种影像学方法，如CT扫描、MRI、PET-CT等，来获取患者体内肿瘤的准确位置和形状。使用影像学数据，能够帮助放射肿瘤学家和医生确定靶区的边界，确定肿瘤组织并尽量减少对正常组织的伤害。如何高效的重建出准确、光滑的三维勾画区域成为当前核心技术的关键性问题。

MC( Marching Cubes) 算法是在1987年由Lorensen和Cline提出的在三维数据场中进行等值面提取的经典算法。由于其三维重建效果好、算法原理简单等特点，特别适用于医学图像处理。但是随着对算法的深入了解及应用，MC算法还是存在以下几点不足:一是二义性，相邻立方体之间的等值面存在拓扑结构不匹配，导致重建出的外轮廓存在空洞现象;二是时间长，传统算法需要对所有体素进行遍历，由于存在很多空体素，即与等值面不相交的体素，占所有体素的65%以上，导致花费大量时间;三是MC算法通常会生成坏三角形，当等值面紧密经过节点时，会导致两个面或多个边切割靠近一个节点，靠近切割的位置都会形成两个三角形共享一个短角边，形成很多无用的微小和细长的三角形。针对以上问题，本文提出了改进Marching Cubes算法的肿瘤靶区三维重建与可视化方法: 首先选择种子三角形，根据这些三角形重建出整个3D轮廓，其次采用中点选择法代替线性插值法,进一步提高三维重建的速度，并使轮廓更加光滑，并利用VTK实现三维勾画肿瘤区域的显示、旋转、缩放等功能，方便医生在任意角度观察勾画的肿瘤区域，为准确制定治疗计划提供理论依据和技术支持。

2 、改进Marching Cubes算法

在传统的MC算法中，三维重建过程每个立方体的等值面按照其15种基本拓扑进行插值，如图4所示。其基本原理是: 循环遍历所有标量场数据中的立方体，给每个立方体顶点赋值，即二值化每个顶点，采用线性插值的方法,计算出各个交点的位置坐标，然后根据拓扑结构连接这些点，在立方体中形成0或多个三角面片，最后将这些三角面片拼接成近似的外表面。

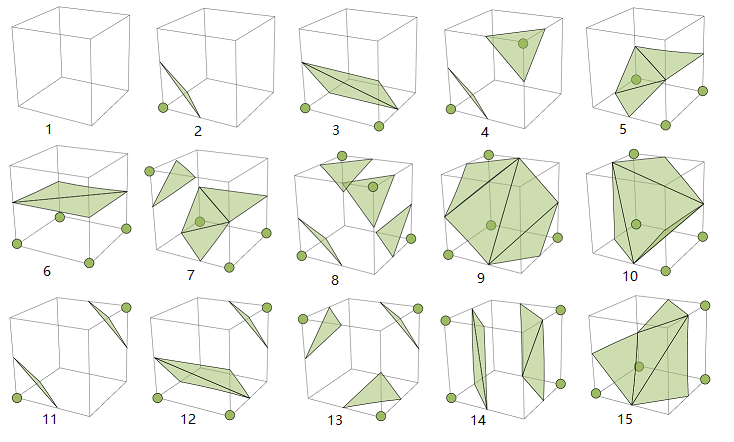


图1 传统MC 算法的 15 种拓扑结构

由于传统的MC算法是将所有的立方体全部循环来生成立方体中的等值面，但是由于立方体数据很大，再加上有很多空的立方体，即有不含等值面的立方体存在，会导致大量的时间浪费，并且MC算法在计算时只是独立处理每个立方体，没有考虑相邻立方体的位置关系，会导致偶尔相邻的立方体之间会存在孔洞的现象，如图5所示，黄色的矩形为缺失的孔洞。已知任何物体的外轮廓都是连续的曲面，在3D重建中亦是如此，因此经过相邻立方体留下的等值面并不是独立存在的，而是相互连接的。

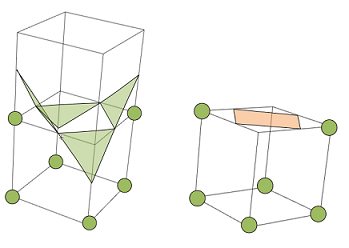


图2 由于二义性导致产生孔洞例子

为了解决上述问题本文提出了三角形生长的连通算法，本算法的核心是利用现有等值面在相邻立方体中形成公共边，再依据体表面具有连通性理论再在相邻立方体中生成等值面。如图3所示，两个立方体包含同一个面ABCD，如果4个顶点不是同时大于或小于给定的值，则种子三角形共边相邻的立方体中也包含等值面。线段EF为种子三角形在公共面的交线，因为该面同属于右面的立方体，所以在右面立方体中的等值面也必然与此面相交于该交线。由此证明在右面的体素中也含有等值面。

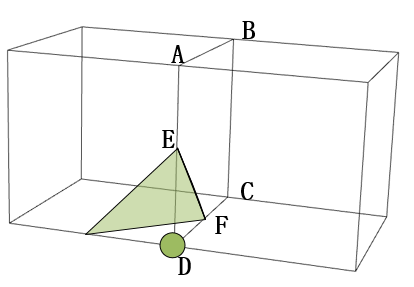


图3 相邻立方体等值面交线例子

立方体六个面中是否存在等值面交线可以通过判断四个顶点的状态。四个顶点状态只有如下四种状态即图4所示，一个顶点和其他顶点状态不同有一种情况，两个顶点和其他顶点状态不同有两种情况，其他为所有顶点状态相同，因此处于1、2、3三种状态中任意一种，则与种子三角形公共面交线的立方体中就包含等值面。

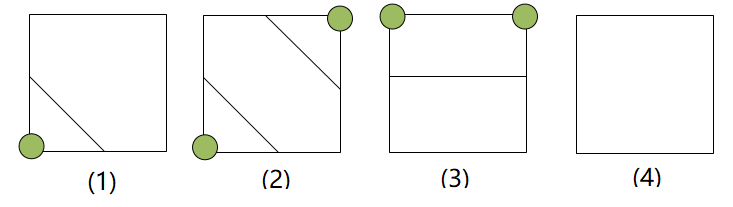


图4 立方体与种子三角形交线的情况

本文的算法利用连通队列和散列表来辅助整个重建过程。连通队列存储种子三角形信息及相邻立方体相交线信息，散列表存储每个立方体顶点状态和坐标及种子三角形相关信息。算法实现步骤为：

1. 选择种子三角形并将种子三角形信息及相邻立方体相交线信息放入连通队列，并加入散列表中。
2. 从连通队列中取出种子三角形及交线信息，根据拓扑结构对立方体边进行插值，并连接交点形成新的三角等值面。
3. 将步骤2形成的三角形信息及相邻立方体相交线信息放入队列中，同时将种子三角形及交线信息及对应的立方体顶点状态和坐标信息放入散列表中。
4. 重复步骤2和3，当队列为空时则算法停止。

算法流程图如图5所示。

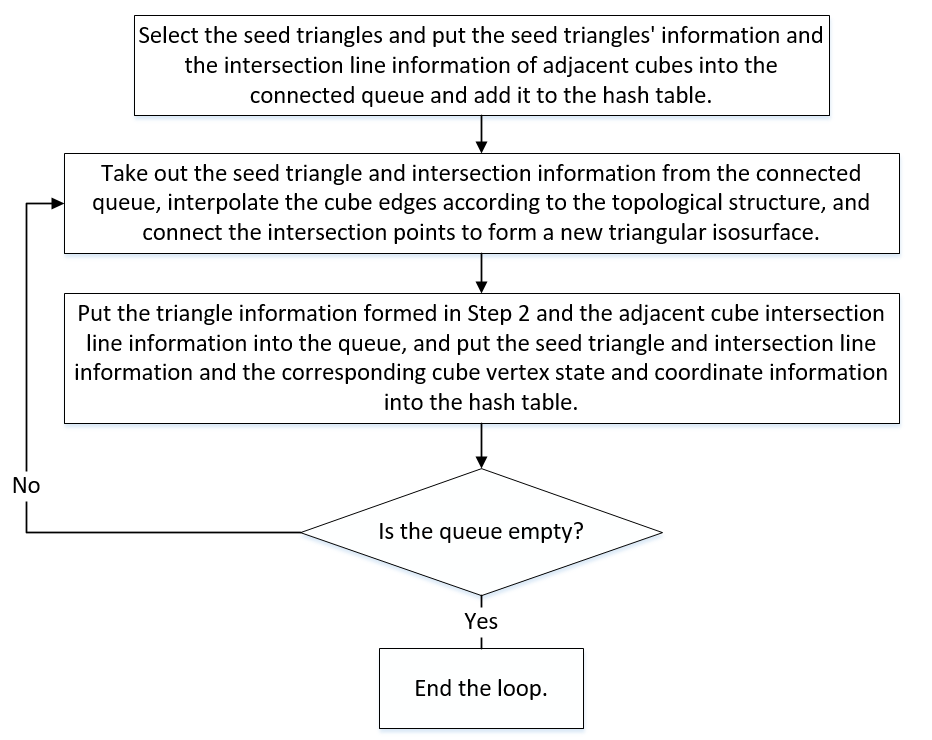


图5 算法流程图

根据结构简单、占比大的原则，本文选择图1中2的拓扑结构作为种子三角形，再根据等值面具有连通性理论再连通出相邻立方体中的等值面。在算法流程结束后，取出散列表中的立方体信息，进行等值点和法向量的计算。传统的计算等值点和法向量公式如下：

P = P1 + (isoValue － V1)(P2 － P1)/(V2 － V1) (1)

N = N1 + (isoValue － V1)(N2 － N1)/(V2 － V1) (2)

其中P表示等值点坐标;P1 、P2 为两个端点坐标;isoValue为阈值；V1、V2 表示两个端点的权值；N1 、N2代表 两个端点的法向量。传统的线性插值法会导致当等值面都紧密经过节点时，会产生很多很小的三角形。为了避免这种微小的三角形，本文提出中点选择法进行插值重建，不用传统MC算法中的通过线性插值计算三角等值面的顶点，而是通过用该顶点所在边的中点得到。利用中点替代三角等值面顶点这样可以取消插值计算步骤，减少计算时间，假设P1 、P2 是立方体一侧的两个顶点，P代表中点，则由中点选择法可得：

P = (P1 + P2 )/2 (3)

N1 、N2 分别为P1 、P2的法向量则P的法向量N为：

N = (N1 － N2)/2 (4)

中点选择的优点是:

1. 误差精度范围在0.5个立方体边长大小，与传统MC算法重建出来的可视化差异很小。
2. 如果先将坐标值放大到没有小数，这样就能进行整数运算，避免浮点型运算。
3. 可以减少运算量，缩短运算时间，同时避免形成微小的三角形，减少相邻立方体内紧密相连的节点数，使局部表面更加平坦，如图6所示。

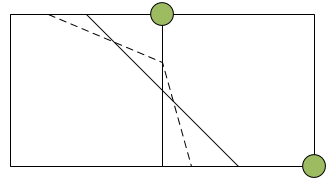


图6 中点选择法示意图

3、重建与可视化的实现

3.1、创建标量场数据并重建

1. 读入DICOM文件并解析。
2. 创建大小为CT图像长、宽、层数大小的三维标量场数据(rows \* cols \* slices)。
3. 根据图像层厚、像素间距、图像左上角坐标位置换算标量场数据里每个体素的物理坐标，如图7所示。

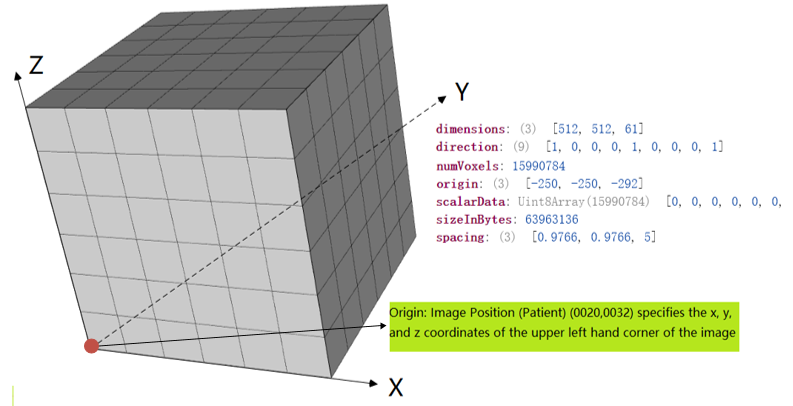


图7 三维标量场模型

1. 将DICOM-RTSTRUCT文件中读取轮廓点位置映射到上面创建好的三维标量数据场中，在轮廓内部的值为1，其他为0。具体步骤如下:
   1. 将轮廓点的物理坐标转成图像每个体素空间索引(i,j,k)，如图8所示。图像坐标系描述了如何获取有关解剖结构的图像。 医学扫描仪会创建规则的矩形点和单元格阵列，这些点和单元格从左上角开始。 i轴向右增加，j轴向底部增加，k轴向后增加。原点表示在解剖坐标系中的第一个像素，图像间隔表示每个轴体素间的距离，使用原点和间隔，就能把对应轮廓点的索引位置算出来，计算方法如下：

function getPointIJK(point=[x,y,z]){

var origin = [x1, y1, z1];

var spacing = [a, b, c];

var i = int((x - x1) / a);

var j = int((y – y1) / b);

var k = int((z – z1) / c);

return {i, j, k};

}

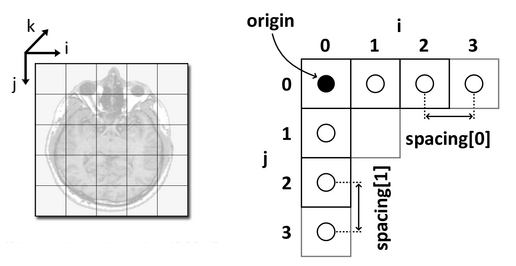


图8 图像坐标系系统

1. 根据上一步索引(i,j,k),找到对应的体素，赋值为1。
2. 再找到轮廓内部的点，并赋值为1
3. 将创建好的标量场数据利用改进后的MC算法进行表面重建。

3.2、靶区外轮廓三维可视化

3.2.1、VTK及可视化管道介绍

VTK是一个全球顶尖的底层可视化工具包，它包含了二维绘图、三维计算机图形、建模、图像处理、体绘制等众多可视化算法和高级建模技术，同时充分利用线程和分布式内存并行处理技术提升运行速度和可扩展性。适合于涉及有限差分和有限元求解的医学成像和工程工作。VTK除了基于OpenGL的桌面版开发，还推出了基于Webgl的VTK.js（也在逐步支持WebGPU），可以为浏览器应用提供高级的科学可视化功能。本文应用VTK.js在浏览器端实现靶区的三维可视化。

VTK的可视化管线基本组成有三种: 数据源、数据过滤器和数据映射器。数据源是可视化管线的起点，它可以从文件、内存或其他来源读取数据，然后将数据转换为VTK的标准数据结构，如多维数组、网格、点云等。数据过滤器是可视化管线的中间环节，它可以对数据进行各种操作，如裁剪、平滑、提取、变换、计算等。数据过滤器可以有多个输入和多个输出，可以实现复杂的数据处理逻辑。数据映射器是可视化管线的终点，它可以将数据转换为图形原语，如点、线、面、体等。数据映射器还可以将数据的属性，如颜色、透明度、纹理等，映射到图形原语上，从而实现数据的可视化表达。

3.2.2、可视化的实现

本文把重建后的三角面片作为输入，以此为数据源创建vtkPolyData, 然后创建vtkActor和vtkMapper, 并将vtkActor和vtkMpaaer绑定，将vtkPolyData的输出数据给到vtkMapper,最后将vtkActor添加到vtkFullScreenRenderWindow中渲染出来，实现靶区的三维重建，三维可视化的管线流程图如图9所示。

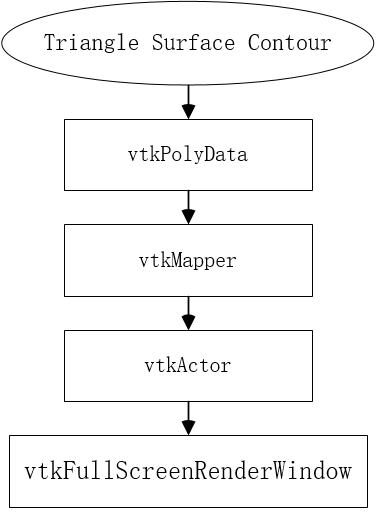


图9 三维可视化的管线流程图

3.3、实验及结果分析

本文实验中的数据来自北京肿瘤医院的三位患者的三个不同部位，分别是头部、肺部、眼部的勾画数据。从轮廓到外轮廓的结果如图10所示。

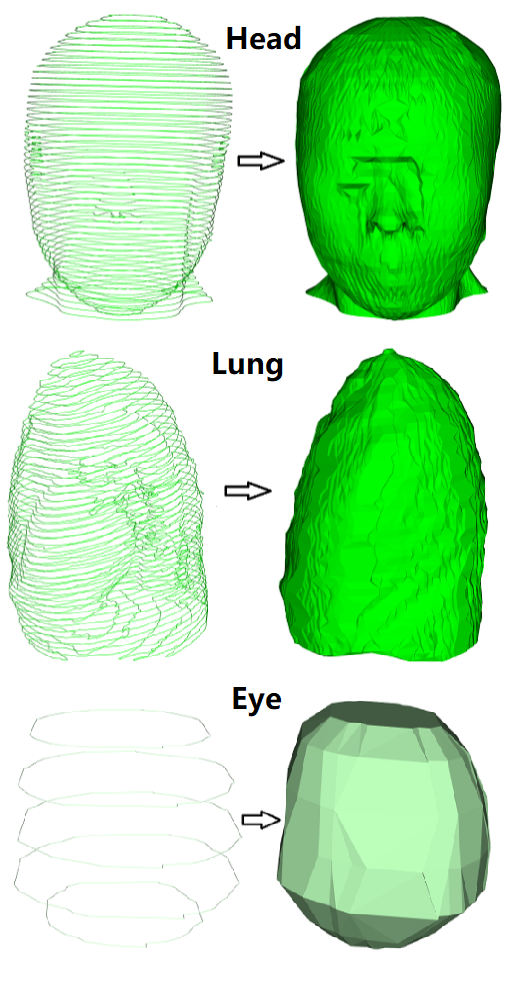


图10 勾画数据的轮廓到外轮廓结果

为了清晰的比较传统的MC算法和改进后的MC算法在三维重建效果和时间上的差异，采用一样的环境下分别运行这两种算法，效率对比如表1所示。从实验结果来看，本文改进后的算法效率得到明显提升，而且计算的三角面片数量也比传统MC算法要少，而且在重建中避免了孔洞的产生，整体来看，本文算法的表现优于传统MC算法。Table 1 Comparison of the 3D reconstruction results of the three sets of data in the experiment

|  |  |  |  |
| --- | --- | --- | --- |
| Organ | Algorithm | Time (ms) | Triangle faces |
| Head | MC | 492.2 | 9200 |
| Ours | 419.9 | 8653 |
| Lung | MC | 422.8 | 5740 |
| Ours | 373.1 | 5297 |
| Eye | MC | 225.5 | 189 |
| Ours | 193.6 | 172 |

4、结束语

本文提出了依据三角形生长的连通性理论改进的MC算法，同时用中点选择法代替线性插值法。首先，改进后的算法解决了传统MC算法由于二义性引起的孔洞现象。其次，缩短了三维重建的时间。本算法的核心思想是选出种子三角形，再依据体表面具有连通性理论选出其他等值面，这样可以避免遍历空体素和其他无用的微小和细长的三角形，这样可以提高算法在三维重建时的效率缩短运行时间。算法分析实验表面：利用本文算法可以减少对应靶区重建时间，减少了等值面的提取。但是提取的外轮廓不够圆滑，可以清晰的看见三角面片的组成，还需要对三维重建后的外轮廓进行光滑处理。