

摘要

医学影像分割技术是医学影像中的重要技术,医学影像分割的效果直接决定着计算机辅助诊断系统的成败。经过多年的研究,国内外学者已经提出了多种算法,但绝大多数算法都是针对某一具体问题的。

本文主要研究基于水平集的医学影像分割。水平集方法的出现,极大地推动了活动轮廓模型的发展。水平集方法和曲线演化模型相结合,克服了传统 Snake 模型的很多缺陷,极大地拓展了活动轮廓模型的应用领域。

本文首先对现有的医学影像分割算法进行了比较全面的综述。在介绍过几何活动轮廓模型和水平集方法之后,讨论了水平集方法的两种快速算法,并总结出水平集方法的优缺点。针对水平集方法计算量比较大这一缺点,本文提出了一种无需重新初始化的基于图像边缘信息的变分水平集图像分割算法,从而完全消除了真正影响水平集计算量的重新初始化步骤,在加速轮廓线演化的同时提高了算法的鲁棒性。针对水平集方法演化速度较慢的缺点,本文提出了一种基于稀疏场方法的水平集分割算法,该方法减少了每次迭代时更新的像素,增加了每次迭代中的时间步长,在加速演化过程的同时提高了演化计算的精度,适宜处理医学影像这类数据量较大的图像。最后,论文指出了进一步的研究方向。

关键词: 医学影像分割 水平集 活动轮廓 变分水平集 稀疏场

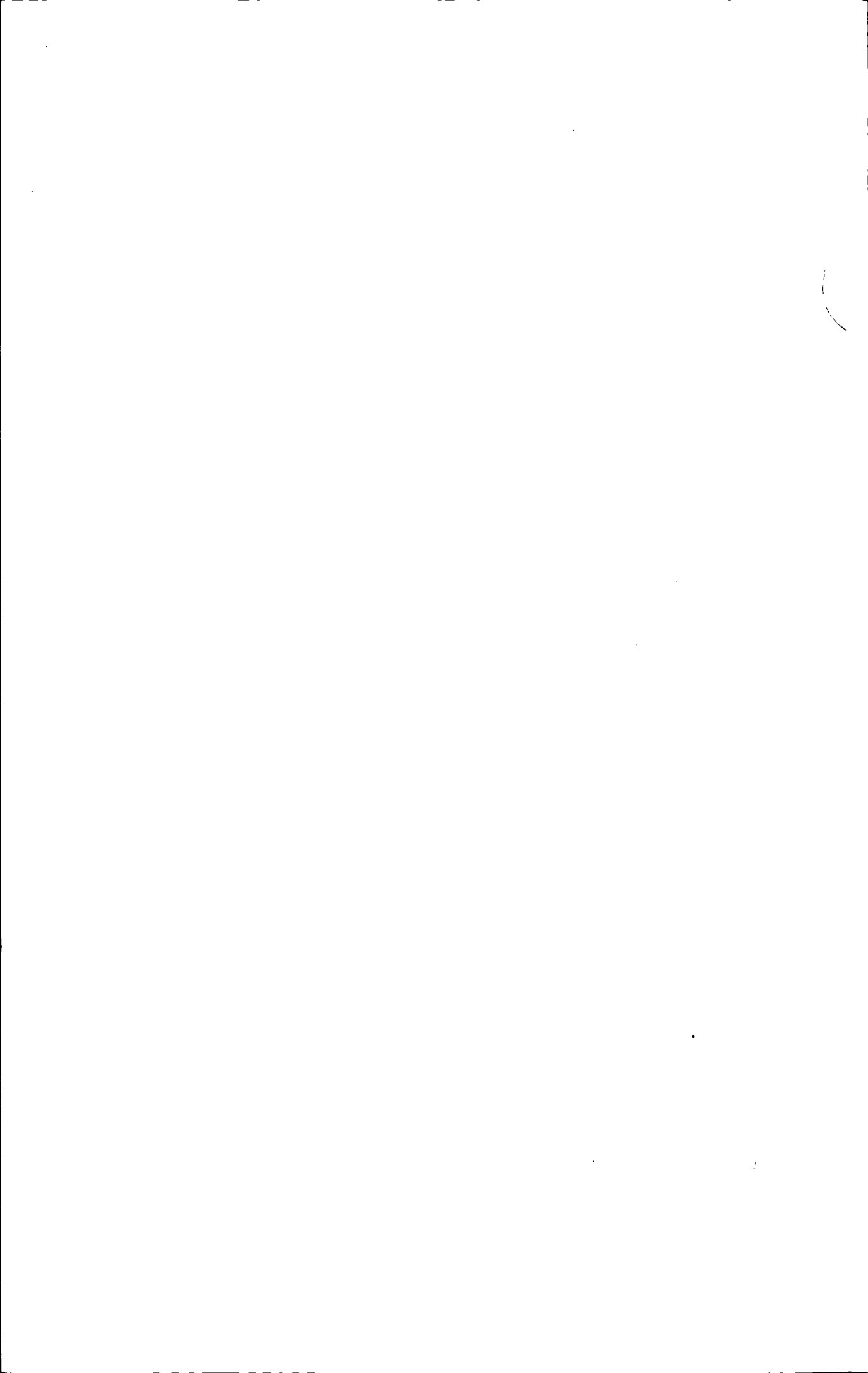
Abstract

Segmentation of medical images plays a key role in medical imaging technology, and the effect of which will impact the results of computer assisted diagnosis. Recent years, a variety of algorithms has been proposed, most of which, however, concentrate on a specific issue.

This thesis concentrates on the research on segmentation of medical images based on level set, which greatly promoted the development of active contour model. The cooperation of level set methods with curve evolution theory conquers multiple drawbacks of traditional snake model, and highly widens the application areas of active contours model.

The thesis commences with comprehensive comparisons of algorithms available for segmentation of medical images. Then, geometric active contour model and level set methods are introduced, as well as two fast algorithms for level set methods before an evaluation of level set methods is made. Considering the large calculation amount of level set methods, it is presented in the thesis a variational level set algorithm for image segmentation, based on edge information, and without re-initialization, thus completely eliminating the need of costly re-initialization, accelerating contour evolution, and enhancing robustness. Moreover, considering poor evolution speed of level set methods, it is proposed in the thesis a level set segmentation algorithm based on sparse field methods, which reduces pixels to be renewed and increases time step each iteration, thus accelerating evolution, improving calculation accuracy, and applicable to such images with abundant data as medical images. The thesis concludes with the further research direction in future.

Keywords: Segmentation of medical images Level set Active contour
Variational level set Sparse field



目录

第一章 绪论	1
1.1 研究背景及意义	1
1.2 医学影像分割的研究进展及现状	2
1.3 本文的主要研究成果及内容安排	3
第二章 医学影像分割算法研究	5
2.1 图像分割的定义	5
2.2 医学影像分割算法的分类	6
2.2.1 基于阈值选取的分割方法	6
2.2.2 基于区域的图像分割方法	6
2.2.3 基于边缘检测的分割方法	7
2.2.4 基于特定理论的分割方法	10
2.3 小结	12
第三章 基于水平集方法的医学影像分割方法研究	13
3.1 水平集理论简介	13
3.2 曲线演化理论	13
3.2.1 水平集理论基本原理	14
3.2.2 水平集方法的数值计算	17
3.3 水平集函数演化的快速算法	19
3.3.1 窄带法	19
3.3.2 快速行进法	20
3.4 水平集方法的优缺点	22
3.5 小结	23
第四章 无需重新初始化的变分水平集方法研究	25
4.1 变分水平集方法简介	25
4.2 无需重新初始化的变分水平集图像分割方法	26
4.2.1 算法原理	26
4.2.2 算法实现	28
4.2.3 实验结果与分析	30
4.3 小结	34
第五章 基于稀疏场方法的水平集分割研究	35

5.1 传统稀疏场定义.....	35
5.2 稀疏场算法.....	36
5.2.1 初始化.....	36
5.2.2 轮廓演化.....	37
5.2.3 实验结果与分析.....	41
5.3 相似技术.....	44
5.4 小结.....	45
第六章 总结与展望.....	47
6.1 本文工作总结.....	47
6.2 未来发展展望.....	47
致谢.....	49
参考文献.....	51
研究成果.....	55

第一章 绪论

1.1 研究背景及意义

自 1895 年伦琴发现了 X 射线成像以来,伴随着物理学、电子学、计算机技术日新月异的发展,应用于医学研究和临床诊断的医学影像技术也得到了突飞猛进的发展。医学影像技术种类繁多,包括 X 射线、计算机 X 射线断层扫描(Computed Tomography, CT)、核磁共振成像(Magnetic Resonance Imaging, MRI)、超声(Ultrasound)、同位素扫描等。医学影像技术在定量分析、实时监控和治疗规划等方面都具有很大的潜力,但是,仅靠医学专家对图像的人工分析进行诊断治疗是一项非常繁重的工作,并且准确性和实时性都受到一定程度的限制;而如果通过计算机对医学影像进行分析,可以使医学专家摆脱繁重的人工观察和诊断,并且还可以提供病患者更加全面而精确的辅助诊断数据。医学影像技术的日益成熟,使得现代医疗诊断已经越来越离不开影像技术,开展医学图像处理研究对提高医学影像数据的利用价值和降低临床误诊概率具有深远的意义。因此,医学图像计算机辅助诊断长期以来备受国内外研究人员的关注。

在运用医学影像技术对内部组织器官进行定量分析、实时监控和治疗规划的过程中,经常需要对人体某种组织和器官的形状、边界、截面面积以及体积进行测量,从而得到该组织病理或功能方面的重要信息,因此,精确的测量对疾病的诊断和治疗有着非常重要的临床意义。图像边界清晰与否在医学上也十分重要,例如,X 光片和 CT 片需要勾画出骨骼的明确边界,才能判断出骨骼是否发生病变,因此有必要对图像的边界进行准确提取。对人体各种组织的正确分割不仅可以为临床组织病变提供计算机辅助诊断依据,而且也是图像三维重建和医学图像可视化的基础。

医学影像分割在医学图像处理研究方面具有重要的意义:

(1) 它是其它后续图像处理(如三维重建、图像配准、计算机辅助诊断系统)的基础,为后续研究提供准确可靠的某一特定组织结构的完备数据集。

(2) 便于提取特定目标组织的特征参数,如提取目标几何的、物理的、病理的、统计的等特征参数,建立并依次建立信息数据库或图谱,为诊断医师提供有价值的参考信息。

(3) 对目标区域进行自动或半自动化定位,将分割的结果进行三维重建,并实现可视化,为放射治疗、化学治疗、外科手术、计算机辅助诊断等治疗方案定位目标。

(4) 便于保存和管理图像信息, 分割后的图像在不丢失有用信息的前提下, 实现图像数据的最大压缩比。这在图像保存、传输、远程诊断治疗和 PACS(Picture Archiving and Communication Systems)系统中具有重要的实际应用价值。

1.2 医学影像分割的研究进展及现状

医学影像分割研究发展到今天仍然没有得到圆满解决, 其中一个重要的原因是医学图像的复杂性和多样性。由于医学图像的成像原理和组织本身的特性差异, 而且图像的形成受到诸如噪音、局部体效应、场偏移效应和组织运动等的影响, 医学图像与普通图像相比, 不可避免地具有模糊、不均匀性等特点。另外, 人体的解剖组织结构和形状比较复杂, 而且人与人之间有相当大的个体差异, 这些都给医学影像分割带来了困难。

医学影像分割经历了一个从人工分割到半自动分割现在正向自动分割发展的过程。早期的人工分割是由医疗工作者从医学图像中通过手工描绘出不同结构的边界, 然后提取出所要解剖结构的区域信息。随着计算机和图像处理技术的发展, 人工操作逐渐被半自动化分割方法所取代, 通过一定的人机交互, 由计算机完成分割。此类方法大大加快了分割速度并且提高了分割精度, 而且减少了人为因素的干扰, 但是, 操作者的经验和知识仍然是制约图像分割的一个重要因素。

近年来, 由于大量新兴技术在图像分割中的应用, 医学影像分割领域也涌现出了一些自动分割技术^{[1][2][3][4]}, 但这些方法复杂度较高, 运算量较大, 目前仍处在实验阶段, 真正得到临床应用的并不多^[5]。

目前, 医学影像分割方法的研究主要有以下四个趋势^[6]:

(1) 提高算法的自动化程度, 尽量减少人工干预。

(2) 降低算法的复杂性, 提高算法效率。医学图像的数据量通常比较大, 算法越复杂, 需要提取的特征就越多, 计算量就越大, 对计算机的存储量、计算精度、计算速度等性能的要求就越高, 这必然导致算法在实际应用中受到一定程度的限制。

(3) 提高算法精度。这是医学影像分割领域一直关注的一大难题, 分割结果的好坏直接影响到医生对疾病的诊断, 错误的分割结果甚至可能导致医生对疾病做出误诊。

(4) 提高算法的鲁棒性。由于个体的一些不确定因素, 如图像模式、特征参数、计算方法、初始化处理、分析步骤、阈值设置、人为因素等, 要在图像分割算法中获得非常好的鲁棒性是比较困难的。因此, 如何提高图像分割算法的鲁棒性是医学影像分割中亟待解决的一大难题。

1.3 本文的主要研究成果及内容安排

本论文以作者攻读硕士学位期间承担课题的工作为基础,主要研究了水平集方法在医学影像分割上的应用,这种方法提高了分割算法的鲁棒性,在提高演化计算精度的同时加速了演化过程,从而改善了图像的分割效果。论文的主要内容包括:

(1) 讨论了当前主要的医学影像分割算法,分析了每种算法的研究现状和技术特点;

(2) 重点研究了基于水平集方法的医学影像分割算法,对水平集方法的基本理论和数值计算流程进行了专门的分析和研究,讨论了水平集方法的两种快速算法,窄带法和快速行进法。通过对这两种算法的分析和研究,总结出水平集算法的优缺点,为水平集算法的改进提供基础;

(3) 介绍了一种无需重新初始化的基于图像边缘信息的变分水平集图像分割算法。这种算法完全消除了真正影响水平集计算量的重新初始化步骤,从而加速了轮廓线的演化,提高了算法的鲁棒性,同时使得初始化方法更加灵活;

(4) 讨论了一种基于稀疏场方法的水平集分割算法。该算法试图把窄带宽度缩小到极限,减少每次迭代时更新的像(体)素,同时可以提高每次迭代中时间步长的值,这样就加快了演化过程。另外,稀疏场算法还可以提高演化计算的精度。该算法适宜处理医学影像这类数据量比较大的图像。

全文共分为六章,具体的安排如下:

第一章:绪论。阐述了本课题的研究背景和意义,介绍了医学影像分割的研究进展及现状,最后给出本文的主要研究成果及内容安排。

第二章:医学影像分割算法研究。分类讨论了现有的医学影像分割算法,比较详尽地介绍了每一种分割算法并简单地评述了其优缺点。

第三章:基于水平集方法的医学影像分割方法研究。详细说明了水平集分割算法的基本原理和数值计算方法,分析了曲线演化理论和水平集偏微分数值计算方法等理论知识,介绍了水平集的快速算法窄带法和快速行进法的技术原理和实现流程。

第四章:无需重新初始化的变分水平集方法研究。讨论了这一基于图像边缘信息的几何活动轮廓算法,并对该算法的原理以及实现做了详细阐述,最后给出该算法的实验结果。

第五章:基于稀疏场方法的水平集分割研究。介绍了传统稀疏场,详细阐述了稀疏场算法,给出了实验结果,并对稀疏场方法进行了改进,从而加速了演化的速度。

第六章：总结与展望。对本论文研究工作进行总结，对以后的研究进行展望，指出下一步的研究方向。

第二章 医学影像分割算法研究

2.1 图像分割的定义

图像分割(Image Segmentation)^[7]就是将图像表示为物理上有意义的连通区域的集合,即把图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标的技术和过程,它是由图像处理到图像分析的关键步骤。人们一般是通过对图像的不同特征,如边缘、纹理、颜色、亮度等的分析达到图像分割的目的。图像分割通常是为了进一步对图像进行分析、识别、跟踪、理解、压缩编码等,分割的准确性直接影响后续任务的有效性,因此具有十分重要的意义。

令集合 R 代表整个图像区域,对 R 的分割可看作将 R 分成若干个满足如下五个条件的非空子集(子区域) R_1, R_2, \dots, R_n ^[8]。

(1) $\bigcup_{i=1}^n R_i = R$ 。完备性:分割所有全部子区域的总和(并集)应能包括图像中所有像素或将图像中每个像素都划分进一个子区中。

(2) $\forall i, j, i \neq j, R_i \cap R_j = \emptyset$ 。独立性:一个像素不能同时属于两个子区域,即各子集互不重叠。

(3) $P(R_i) = true, i = 1, 2, \dots, n$ 。相似性:即属于同一个子区域的像素应具有某些相同或相近的特性。

(4) $P(R_i \cup R_j) = false, i \neq j$ 。互斥性:即属于不同子区域的像素应具有某些不同的特性。

(5) R_i 是一个连通的区域, $\forall i$ 。连通性:同一个子区域中的像素点是连通的。

其中 \emptyset 表示空集, $P(R_i)$ 是对子区域 R_i 中所有元素的逻辑谓词,即特性的相似性准则。

实际应用中,总是将图像划分为背景子区域(用 R_1 表示)和不同的目标物子区域(用 R_i 表示, $i = 2, 3, \dots, n$),其中的划分满足以上定义,则 $R_i (i = 1, 2, 3, \dots, n)$ 就称为 R 的分割。

根据图像的某些局部特征(如灰度、颜色或几何性质)的相似性和互斥性,将图像分割成若干个子区域,在每个子区域内部具有相似特性,而相邻子区域的特性不相同。因此,图像局部特性的相似性和互斥性可作为图像分割的依据。

若过分强调分割区域的相似性,分割区域很容易产生大量的空白和不规整的边缘;若过分强调分割后不同区域之间的性质差异,则会造成非同质区域的合并和丢失一部分有意义的边界。图像处理和机器视觉界的研究者们为了研究满足上

述定义的分割算法做了长期的努力。图像分割的算法都是针对某一类型的图像或某一具体应用的，通用方法和策略仍面临巨大的困难，同时并不存在一个判断分割是否成功的客观标准，因此，图像分割被认为是计算机视觉中的一个瓶颈。

2.2 医学影像分割算法的分类

现有的医学影像分割算法主要分以下几类：基于阈值选取的分割方法、基于区域的图像分割方法、基于边缘检测的分割方法以及基于特定理论的分割方法等^{[9][10]}。

2.2.1 基于阈值选取的分割方法

基于阈值选取的图像分割方法是提取目标物体与背景在灰度上的差异，把图像分为具有不同灰度级的目标区域和背景区域的组合。阈值法对物体和背景对比较强的图像分割有着很强的优势，计算较为简单，并且可以用封闭和连通的边界定义不交叠的区域，是图像分割中最有效且实用的技术之一。根据获取最优分割阈值的途径可以把阈值法分为全局阈值法、动态阈值法、模糊阈值法和随机阈值法等。尽管阈值选取的方法很多，但至今还没有找到一种对所有图像都可以有效分割的方法，一种阈值方法只能适用于某一类或几类图像，因此一些学者在研究新的阈值选取方法之外，同时将新的数学工具融合到原有的阈值方法中，并取得了较好的分割效果。

阈值分割技术^[11]是一种并行分割方法，它的关键在于阈值的选择，不同的阈值可能会引起完全不同的分割结果，阈值的个数和取值往往要根据实际的要求，由一些先验知识和实验统计数据来确定。阈值法对物体和背景对比较弱的图像，或者物体和背景的灰度值范围有较大重叠的图像，阈值分割则难以得到准确的结果。另外，由于阈值方法孤立地考虑每个像素，没有考虑像素之间的空间关系，所以对噪声非常敏感。

2.2.2 基于区域的图像分割方法

基于区域的分割方法是典型的串行区域技术，依赖于图像的空间局部特征，如灰度、纹理及其他像素统计特性的均匀性。典型的基于区域的分割方法有区域生长法、分裂-合并法等。它们的主要优点是对噪声不敏感，但是，常常会造成图像的过分分割问题(over segmented)，并且分割结果很大程度上依赖于种子点(seed points)的选择，分割所得到的区域的形状也依赖于所选择的分割算法。

1. 区域生长法

区域生长(Region Growing)的基本思想是将具有相似性质的像素集合起来构成区域。具体实现是对每个分割的区域找个种子像素作为生长的起点,根据某种事先确定的生长或相似准则,在种子像素周围邻域中寻找与种子像素有相同或相似性质的像素,并将这些像素合并到种子像素所在的区域中。将这些新像素当做新的种子像素继续进行上述的过程,直到再没有满足条件的像素可被包括起来,这样,一个区域就生成了。

区域生长需要选择一组能正确代表所需区域的种子像素,确定在生长过程中能将相邻像素包括进来的相似性准则以及制定让生长停止的条件或准则。选取的种子像素可以是单个像素,也可以是包含若干个像素的小区域;相似性准则可以是灰度级、彩色、纹理、梯度等特性;大部分区域生长准则使用图像的局部性质;生长准则可根据不同原则制定,而使用不同的生长准则会影响区域生长的过程。

区域生长法的优点是计算简单,对于较均匀的连通目标有较好的分割效果。它的缺点是需要人为确定种子点,对噪声敏感,可能导致区域内有空洞。另外,它是一种并行算法,当目标较大时,分割速度较慢,因此在设计算法时,要尽量提高效率。

2. 分裂-合并

分裂-合并分割方法^[12]差不多是区域生长的逆过程;它从整个图像出发,不断分裂得到各个子区域,然后再把前景区域合并,实现目标提取。

分裂-合并的假设是对于一幅图像,前景区域是由一些相互连通的像素组成的。因此,如果把一幅图像分裂到像素级,那么就可以判定该像素是否为前景像素。当所有像素点或者子区域完成判断以后,把前景区域或者像素合并就可得到前景目标。

分裂-合并分割方法的关键是分裂-合并准则的设计。这种方法对复杂图像的分割效果较好,但算法较复杂,计算量大,分裂还可能破坏区域的边界。

2.2.3 基于边缘检测的分割方法

边缘检测方法^[13]是人们研究得比较多的一种方法,人们通过检测图像中不同区域的边缘即检测灰度级或者结构具有突变的地方,可以达到分割图像的目的。这种基于边缘检测的图像分割的方法不依赖于已处理像素的结果,它适于并行化,但缺点是对噪声敏感,而且当边缘像素值变化不明显时,容易产生假边界或不连续的边界。

边缘可以分为阶跃边缘和屋脊边缘两种,阶跃边缘是图像中局部区域灰度变换最剧烈的点构成的曲线;而屋脊边缘是图像灰度的局部极值点构成的曲线。一

一般而言, 阶跃边缘反映了图像中不同区域的界限, 而屋脊边缘则反映了图像中的线状或点状结构。

人们为边缘检测设计了各种检测算子, 这些算子总体上可以分为两类^[14], 其中 Robert 算子、Sobel 算子、Kirsh 算子都属于一阶微分算子, 而 Laplacian 算子、LOG 算子、Canny 算子^[15]都属于二阶微分算子。一阶微分算子利用的是图像的一阶导数(X 或 Y 方向上的方向导数)在阶跃边缘处取得极大值而在屋脊边缘处为 0 的特性; 而二阶微分算子利用的是图像的二阶导数在阶跃边缘处取 0 值而在屋脊边缘处取得极大值的特性。

根据边缘检测采用方式的不同, 边缘检测方法可以细分为如下几类。

(1) 基于局部图像函数的方法: 将灰度看成高度, 用一个曲面来拟合一个小窗口内的数据, 然后根据该曲面来决定边缘点。

(2) 基于多尺度的边缘检测方法: 实际上是用不同尺度的滤波算子对图像进行卷积, 并考察由此得到的边缘点随尺度的变化而具有的性质, 结果多种不同尺度的信息来最终决定边缘点。其中可以利用多分辨率的思想从初始图像用规则或不规则的方式逐步降低分辨率, 得到金字塔形的一个图像序列, 再在此基础上进行图像分割。

(3) 基于图像滤波法: 基于对平滑滤波后的图像求其一阶导数的极大值或二阶导数的过零点, 以此来决定边缘, 它的核心问题是要设计一个合适的滤波器, 通常使用的滤波器有可控滤波器、LOG 滤波器、B 样条滤波器等。

(4) 基于边界曲线拟合方法: 用平面曲线来表示不同区域之间图像的边界线, 并试图根据图像梯度等信息找到能正确表示边界的曲线, 以此来达到分割图像的目的。

(5) 状态空间搜索法: 也称图(Graph)搜索法或启发式(Heuristic)搜索法, 该方法用图结构表示边界点和边界段

(6) 动态规划法: 通过把一个 N 步过程化为 N 个单步过程的方法使算法复杂度降低, 根据动态规划的原理, 将全局最优化成局部最优之和, 因此是一个多步决策的过程。

(7) 边界跟踪法: 也称边缘点链接(Edge Point Linking)法。它由图像梯度图中一个边缘点出发, 依次搜索并连接相邻边缘点, 从而逐步检测出边界的方法。

(8) Hough 变换法: 利用图像全局特性而直接检测目标轮廓的一种常见的方法, 该方法的主要优点是受噪声和曲线间断的影响较小。

下面介绍另外一些新发展起来的边缘检测方法。

1. 基于数学形态学的边缘检测方法

数学形态学是一种非线性滤波方法, 它建立在积分几何和随机集理论基础上的整套变换、概念和算法。基于数学形态学的分割方法利用膨胀、腐蚀两种数

学形态学基本运算及其不同逻辑组合构成的开运算和闭运算对图像进行处理,然后再通过边缘强度算子就可以检测出图像的边缘^[16]。它提供了并行的,具有鲁棒性的图像分割技术,它不仅能够得到图像中各种几何参数的间接测量,而且适合于描述图像的随机性质^[17]。

将数学形态学用于边缘检测,既能有效地滤除噪声,又可保留图像中的原有细节信息,具有较好的边缘检测效果;用数学形态学运算进行边缘检测也存在着一定的不足,比如结构元素单一的问题。

形态学理论在图像分割中的应用比较有代表性的是 Luc Vincent 等^[18]人提出的分水岭(Watershed)算法,研究人员至今已提出了多种基于分水岭的形态学分割方法^[19]。

2. 基于分形理论的边缘检测方法

分形分割方法^[20]是近年来新出现的图像分割方法。自 1975 年 Mandelbrot 系统地提出了分形几何学的理论^[21]以来,它已经逐渐成为研究和处理复杂和不规则图形的有力工具。

分形理论用分形维数这一概念来衡量描述自然现象的不规则程度,分形分割方法的基本思想是任何图像都可以看作是由某种纹理构成的,大部分纹理图像都可以用分形模型进行描述。分形集的特征是具有自相似性,属于同一分形集的像素应该有相同的分形维数。这种方法用于图像分割的特点在于分形维数直观上是与物体表面的粗糙程度相吻合的,而自然界中不同的纹理粗糙度则有很大的差别,因此可以用它作为有效的特征参数来区别不同的纹理。但是,一般情况下,基于纹理分析的图像分割分辨率都比较低,因此,无论是医学图像,还是自然图像的分割,分形理论的应用都是比较少的。

3. 基于遗传算法的边缘检测方法

遗传算法是基于自然选择和基因遗传学原理的随机搜索算法,其核心问题是寻找求解优化问题的效率与稳定性之间的有机协调,即“鲁棒性”。研究者进行了大量的研究工作,并成功地将它们运用到各种类型的优化问题上。在分割复杂的图像时,人们往往采用多参量进行信息融合,在多参量参与的最优值的求解过程中,优化计算是最重要的,把自然进化的特征应用到计算机算法中,将能解决很多困难。遗传算法的出现为解决这类问题提供了新颖而有效的方法,它不仅可以得到全局最优解,而且大大缩短了计算时间。

对于图像的边缘提取,采用二阶的边缘检测算子处理后再进行过零点检测,其计算量很大,硬件实时资源占用大而且速度慢,因此,有人提出了一种二次搜索寻优的阈值选取策略。通过遗传算法进行边缘提取阈值的自动提取,能够显著地提高阈值选取的速度,可以对视觉系统所产生的边缘图像进行阈值的实时自动选取,增强了整个视觉系统的实时性和鲁棒性。

4. 基于参数活动轮廓模型(Snake)的边缘检测方法

Kass^[22]于 1987 年提出的 Snake 模型又称为参数活动轮廓模型,它融合了分割过程的三个阶段,检测得到的目标边界是一条光滑连接的曲线。Snake 是能量极小化的曲线,内力约束它的形状,外力引导它的行为,图像力将其拖向显著的图像特征。Snake 方法及其改进方法在静态图像处理方面主要用于检测图像边缘、曲率、边界以及区域分割和骨架提取等。

Snake 模型^{[23][24]}的主要思想是定义一个能量函数,由初始位置向真实轮廓逐渐靠近时,寻找此能量函数的局部极小值,即通过对能量函数的动态优化来逼近目标的真实轮廓。此能量函数主要由内部能量函数与外部能量函数组成。内部能量函数考虑包络本身的连续性和各点曲率的大小;外部能量函数则主要涉及到图像的一些具体情况,如图像灰度变化的梯度等因素。

Snake 可以描述成一条参数曲线: $v(s) = (x(s), y(s))$, 其中 s 是弧长。定义以 $v(s)$ 为变量的能量函数 E_{snake} , 即

$$E_{snake} = E_{int} + E_{image} + E_{con} = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))) ds \quad (2-1)$$

式中, E_{int} 表示内部能量; E_{image} 表示图像能量; E_{con} 表示外部能量; $v(s)$ 是 Snake 沿路径 s 上的点; $x(s)$, $y(s)$ 是 $v(s)$ 的坐标, 路径 s 的长度归一化到 1。Snake 模型的目标是寻找 E_{snake} 的最优解, 当 E_{snake} 达到某一极小值, 说明 Snake 达到某种平衡。

Snake 的行为是由外力和内力来控制的, 内力起平滑性限制的作用, 外力引导初始轮廓线向图像特征逼近, 初始轮廓线在内、外力的作用下寻找其能量函数的极小值。

Snake 模型具有一些经典方法无法比拟的优点, 如图像数据、初始估计、目标轮廓及基于知识的约束统一于一个特征提取过程中; 适当初始化后, 其能够自主地收敛于能量极小值状态; 尺度空间中由粗到精地极小化能量可以极大地扩展捕获区域和降低计算复杂性。但 Snake 模型也存在缺点, 如对初始位置敏感, 需要依赖其他机制将 Snake 放置到图像特征附近; 由于其的非凸性, 它有可能收敛到局部极值点, 甚至发散。

基于边缘的分割方法定位精度比较高, 但受噪声影响比基于区域的方法却要大得多, 所以对于医学图像而言, 仅仅利用基于边缘的分割技术是难以达到目的的。近年来的绝大部分边缘分割技术要与其它模型结合才能得到满意的结果。

2.2.4 基于特定理论的分割方法

图像分割至今尚无通用的自身理论。随着各学科许多新理论和新方法的提出,

出现了许多与一些特定理论、方法相结合的图像分割方法。

1. 基于聚类分析的图像分割方法

特征空间聚类法进行图像分割是将图像空间中的像素用对应的特征空间点表示,根据它们在特征空间的聚集对特征空间进行分割,然后将它们映射回原图像空间,得到分割结果。K 均值、模糊 C 均值聚类(FCM)算法是最常用的聚类算法。

K 均值算法是先选 K 个初始类均值,然后将每个像素归入均值离它最近的类并计算新的类均值,迭代执行前面的步骤直到新旧类均值之差小于某一阈值。模糊 C 均值算法是在模糊数学基础上对 K 均值算法的推广,是通过最优化一个模糊目标函数实现聚类,它不像 K 均值聚类那样认为每个点只能属于某一类,而是赋予每个点一个对各类的类属度,适宜处理事物内在的不确定性。利用模糊 C 均值(FCM)非监督模糊聚类标定的特点进行图像分割,可以减少人为的干预,且较适合图像中存在不确定性和模糊性的特点。FCM 算法对初始参数极为敏感。有时需要人工干预参数的初始化以接近全局最优解,提高分割速度。另外,传统 FCM 算法没有考虑空间信息,对噪声和灰度不均匀敏感。

2. 基于模糊集理论的分割方法

模糊图像分割技术把图像分割看成是对一个模糊集的操作。它的基本思想是:首先通过隶属度函数 G (模糊算子)将图像映射成模糊隶属度矩阵(模糊特征平面);然后在模糊特征平面上利用模糊增强算子的非线性变换增强图像的边缘信息并削弱非边缘信息,然后对其做 G^{-1} 运算得到增强的图像,最后通过简单的阈值化处理即可得到图像的边缘。

基于模糊集理论的分割方法是以模糊数学为基础,利用隶属度值决定图像中由于信息不全面、不准确、矛盾、含糊等造成的不确定性问题。目前,模糊技术在图像分割中应用的一个显著特点就是它能与现有的许多图像分割技术相结合,形成一系列集成模糊分割技术,如模糊聚类、模糊阈值、模糊边缘检测技术等,这些技术为医学影像分割技术展示了乐观的前景。但是,由于图像模糊测度的定义和提取比较困难,而且模糊算法也不容易设计,因此,模糊图像分割实施起来非常复杂。对于医学超声图像的分割,因为图像质量较差,所以更是难以运用模糊理论来实现。

3. 基于神经网络的分割方法

近年来,人工神经网络识别技术已经引起了广泛的关注,并应用于图像分割。基于神经网络的分割方法^[25]的基本思想是通过训练多层感知机来得到线性决策函数,然后用决策函数对像素进行分类来达到分割的目的。这种方法需要大量的训练数据,神经网络存在巨量的连接,容易引入空间信息,能较好地解决图像中的噪声和不均匀问题。选择何种网络结构是这种方法要解决的主要问题。

4. 基于小波变换的分割方法

小波变换是近年来得到广泛应用的数学工具，它在时域和频域都具有良好的局部化性质，而且小波变换具有多尺度特性，能够在不同尺度上对信号进行分析，因此在图像处理和分析等许多方面得到应用。

基于小波变换的阈值图像分割方法^[26]的基本思想是：首先由二进制小波变换将图像的直方图分解为不同层次的小波系数，然后依据给定的分割准则和小波系数选择阈值门限，最后利用阈值标出图像分割的区域。整个分割过程是由粗到细，由尺度变化来控制，即起始分割由粗略的 $L^2(R)$ 子空间上投影的直方图来实现。如果分割不理想，则利用直方图在精细的子空间上的小波系数逐步细化图像分割。分割算法的计算量与图像尺寸大小呈线性变化。

2.3 小结

本章主要讨论了现有的医学影像分割算法。首先给出了图像分割的定义、图像分割的意义以及图像分割的依据，随后对现有的医学影像分割算法分类进行了阐述。由于医学影像分割算法众多，并且涉及的学科广泛，限于篇幅本章仅列举了一些主要的分割方法，比较详尽地叙述了每一种分割算法并简单地评述了其优缺点。

第三章 基于水平集方法的医学影像分割方法研究

3.1 水平集理论简介

水平集方法(Level Set Method)^{[27][28]}是求解曲线演化(Curve Evolution)的一种方法,它以隐含的方式表达平面闭合曲线,避免了对曲线演化过程的追踪和参数化。因此,水平集方法比较容易处理曲线拓扑结构的变化,如分裂或合并等。正是由于这些特性,水平集方法已经成为人们广泛研究的对象,并被应用于图像分割等领域。

水平集方法(Level Set Method)最初由 Osher 和 Sethian 提出,用于解决遵循热力学方程下的外形变化过程^[29],由于火苗外形的高动态性和拓扑变化的随意性,所以用参数化的曲线或曲面来描述火苗的这种变化显然是非常费力的。

该方程利用偏微分方程(Partial Differential Equations, PDE)作为数值分析(Numerical Analysis)方法与技术手段被广泛运用于轮廓面(Interface)或轮廓线的运动追踪。由于水平集方法将二维(三维)的闭合曲线(曲面)的演化问题转化为高维空间水平集函数曲面演化的隐含方式求解,适应于对拓扑结构变化的处理,其计算精度高,算法稳定,因而得到广泛应用。

3.2 曲线演化理论

描述曲线几何特征的两个重要参数是单位法向矢量 N 和曲率 k 。 N 描述曲线的方向, k 则描述曲线的弯曲程度。曲线演化理论^[30]就是仅利用曲线的单位法矢和曲率等几何参数来研究曲线随时间的变形,而在参数化活动轮廓模型中,曲线的演化依赖的是参数化曲线的导数。

设演化曲线为 $C(s,t) = (x(s,t), y(s,t))$, s 是参数化变量, t 是时间,并设曲线的内向单位法矢为 N ,曲率为 k ,则曲线沿单位法矢方向的演化方程^[31]为

$$\frac{\partial C}{\partial t} = V(C)N \quad (3-1)$$

式中, $V(C)$ 是速度函数,决定曲线 C 上各点的演化速度。

最常用的曲线演化是曲率演化和常量演化。

曲率演化的偏微分方程为

$$\frac{\partial C}{\partial t} = \alpha kN \quad (3-2)$$

式中, $\alpha > 0$ 是常量; k 是曲线的曲率。

Grayson 证明, 任意形状的简单初始闭合曲线(本文所研究的曲线仅限于简单闭合曲线, 即闭合、无交叉), 在式(3-2)的驱动下, 将会逐渐平滑, 并收敛成一点。

常量演化的偏微分方程为

$$\frac{\partial C}{\partial t} = V_0 N \quad (3-3)$$

式中, V_0 是决定曲线演化速度和方向的常量。与曲率演化不同, 常量演化会导致曲线出现尖角, 并导致拓扑结构发生变化(分裂或合并)。

由上述分析可知, 曲率演化和常量演化的曲线变形特性刚好相反, 前者去除曲线的角点并使曲线更加光滑, 而后者则使曲线产生角点。

将曲线演化模型应用到图像分割中, 利用图像数据耦合曲线的演化速度, 使曲线演化最终停止在目标边缘位置, 这就是几何活动轮廓模型的主要内容, 曲线演化的过程则是通过水平集方法来实现的。

3.2.1 水平集理论基本原理

1. 水平集的定义

水平集方法中, 平面闭合曲线被隐含地表达为三维连续函数曲面 $\phi(x, y)$ 的一个具有相同函数值的同值曲线, 通常是 $\{\phi = 0\}$, 称为零水平集, 而 $\phi(x, y)$ 称为水平集函数。 $\phi(x, t)$ 按照如下方式定义

$$\phi(x, t) > 0 \quad x \in \Omega$$

$$\phi(x, t) < 0 \quad x \notin \Omega$$

$$\phi(x, t) = 0 \quad x \in \partial\Omega = \Gamma(t)$$

式中, Γ 为给定 R^n 空间的曲面, Ω 为以其边界围成的开连通区域。这样, 通过分析 $\phi(x, t)$ 随时间运动的情况, 就可以用其函数值为零的点的集合来描述曲面 Γ 。在数值计算的过程中, 这种定义方式可以很好地描述曲面拓扑结构上的裂变和融合。

【举例】

函数 $\phi(x, y, z) = x^2 + y^2 + z^2$ 对应于常数 c 的水平集是以 $(0, 0, 0)$ 为球心, \sqrt{c} 为半径的球面^[32]。

在一个二维平面的圆, 如 $x^2 + y^2 = 1$ (如图 3.1(a)所示) 可以看成是二元函数 $\phi(x, y) = x^2 + y^2$ 的 1 水平(如图 3.1(b)所示, $Z=1$ 时的曲线)。因此, 计算这个圆的变化时就可以先求 $\phi(x, y)$ 的变化, 再求其 1 水平集。

图 3.1 说明了水平集函数表达闭合曲线的方式, 其中图 3.1(a) 是一条平面闭合曲线, 图 3.1(b) 是其水平集函数的隐含表达, 图中的曲线就是隐含为水平集的平面

曲线。如图 3.1(c), (d)所示, 其中图 3.1(c)是水平集函数更新后其零水平集曲线的形状, 而图 3.1(d)中的零水平集曲线已经分裂为两条。

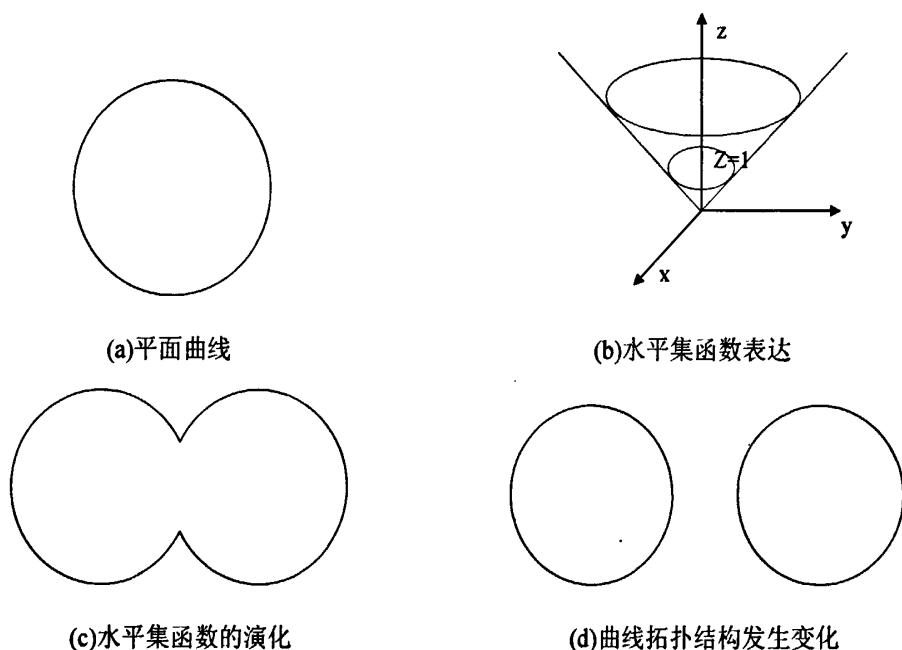


图 3.1 基于水平集方法的曲线演化

2. 水平集的数学表达

下面简要说明水平集方法的数学表达式以及和几何曲线演化模型的关系。

设连续函数 $\phi(x, y, t) (\mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R})$ 是闭合曲线 $C(p, t) (0 \leq p \leq 1)$ 在 t 时刻的隐含表达, 即 t 时刻, $C(p, t)$ 对应于 $\phi(x, y, t)$ 的零水平集

$$\begin{cases} C(p, 0) = \{(x, y) | \phi(x, y, 0) = 0\} \\ C(p, t) = \{(x, y) | \phi(x, y, t) = 0\} \end{cases} \quad (3-4)$$

ϕ 的零水平集所对应的平面闭合曲线

$$\phi(c(t), t) = 0 \quad (3-5)$$

为了使 $\phi(x, y, t)$ 在演化过程中一直满足曲线演化的偏微分方程

$$\frac{\partial C}{\partial t} = V(C)N \quad (3-6)$$

对式(3-5)求全微分, 得

$$\nabla \phi \cdot \frac{\partial C}{\partial t} + \frac{\partial \phi}{\partial t} = 0 \quad (3-7)$$

式中, $\nabla \phi$ 是 ϕ 的梯度。根据水平集函数的定义, ϕ 沿着曲线 C 的方向的变化量为零, 满足下式

$$\phi_s = 0 = \phi_x x_s + \phi_y y_s = \langle \nabla \phi, C_s \rangle \quad (3-8)$$

根据式(3-8)知 $\nabla\phi$ 与 C_s 正交, 即 ∇ 就垂直于闭合曲线 C 的切线 C_s , 因此, $\nabla\phi$ 和 C 的法线同向。假设 C 内部的函数 ϕ 为负, 外部的函数 ϕ 为正。那么水平集 C 的内向单位向量为

$$N = -\frac{\nabla\phi}{|\nabla\phi|} \quad (3-9)$$

将式(3-8)和式(3-9)代入式(3-7), 可得

$$\phi_t = -\nabla\phi \cdot V(k)N = \nabla\phi \cdot V(k) \frac{\nabla\phi}{|\nabla\phi|} = V(k)|\nabla\phi|$$

设 $V_{ext}(x)$ 为由 $V(C)$ 扩展到所有水平集的速度场, 得

$$\frac{\partial\phi}{\partial t} = V_{ext} |\nabla\phi| \quad (3-10)$$

由方程(3-10), 得 $C(p,t)$ 的曲率为

$$k = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \quad (3-11)$$

3. 水平集方法实现中的问题

具体实现水平集方法时, 还需要考虑以下 3 个问题:

(1) 符号距离函数的生成

水平集函数 ϕ 的选择, 通常选取由初始闭合曲线 C_0 生成的符号距离函数 (Signed Distance Function), 记为 SDF。设 $\phi(x,t=0), x \in R^2$ 是 SDF, 则

$$\phi(x,t=0) = \pm d(x) \quad (3-12)$$

式中, $d(x)$ 是点 x 到初始曲线 $C(t=0)$ 的距离, 它的符号根据点 x 在闭合曲线 $C(t=0)$ 的内外部而定。如果 x 位于 $C(t=0)$ 的内部, 取正号(或负号), 反之则取负号(或正号)。

对于任意闭合曲线, 其符号距离函数的计算量比较大。因此, 如何快速稳定地计算任意闭合曲线的 SDF, 对于提高水平集方法的效率和稳定性非常重要。

(2) 速度场的拓展

一般速度函数 $V(C)$ 只定义在零水平集, 其他水平集没有定义, 不过单位法矢和曲率在所有的水平集都有定义。因此, ϕ 演化时, 需要将 $V(C)$ 推广到所有的水平集, 变为拓展的速度场 $V_{ext}(x)$ 。比如, 由图像的边缘信息定义边缘力只在零水平集才有定义, 因而需要将零水平集的边缘力拓展到 ϕ 的整个定义域。然而, 许多拓展速度场的方法常常导致水平集函数 ϕ 不再保持为 SDF, 从而引起单位法矢和曲率计算的误差。因此, 如何选择合适的方法拓展速度场, 也是实现水平集方法时需要认真考虑的问题。

(3) 在几何活动轮廓线的实际应用中, 常常利用常量演化对活动轮廓线产生大尺度的变形作用。然而, 常量演化经常使平滑的水平集函数产生尖角, 一旦形成尖角, 对 ϕ 进一步的演化就不确定了, 因为在尖角位置, 单位法矢的方向是不确定的。如何正确处理常量演化所导致的水平集函数奇异性的问题, 也是实现水平集函数必须考虑的。

总之, 水平集函数实现活动轮廓线演化具有如下优点:

(1) 只要速度函数 V 是平滑的, 水平集函数 $\phi(x, y, t)$ 将始终保持为一个函数, 而 $\phi(x, y, t)$ 的零水平集即演化曲线 C , 却可以随 ϕ 的演化很自然地改变其拓扑结构, 可以分裂、合并、形成尖角等。

(2) 由于 $\phi(x, y, t)$ 演化时始终保持为函数, 因此很容易实现数值近似算法, 可以利用离散网格结构的有限差分方法便可以很容易地实现函数 $\phi(x, y, t)$ 的演化。

(3) 曲线 C 内在的几何特征, 比如内向单位法矢和曲率等, 可以直接由 $\phi(x, y, t)$ 计算出来。

(4) 水平集方法很容易拓展到高维情况。比如可以很方便地拓展成为三维闭合曲线的演化, 这对于三维图像的分割是非常有用的。

3.2.2 水平集方法的数值计算

为了利用水平集方法实现曲线演化, 可以采用适当的数值计算方法。下面我们将详细讨论式(3-10)的数值表达, 包括水平集函数的网格表达, 一阶和二阶微分的计算, 内向单位法向量以及曲率的计算等。

由于水平集函数在演化过程中始终保持为一个函数, 因此, 可以用离散网格来表达水平集函数 $\phi(x, y, t)$ 。设离散网格的间隔为 h , 并设在 n 时刻, 结点 ij 处的水平集函数为 ϕ_{ij}^n , 则 $\phi_{ij}^n = \phi(ih, jh, n\Delta t)$, 这里的 Δt 是时间步长, 则式(3-10)可以离散化为

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} + V_{ij}^n |\nabla_y \phi_{ij}^n| = 0 \quad (3-13)$$

式中, V_{ij}^n 表示 n 时刻拓展速度函数 V_{ext} 位于网格点 ij 处的值。

式(3-13)中, $\phi(x, y, t)$ 的有限数值差分 $\nabla_y \phi_{ij}^n$ 必须采取适当的形式, 以避免常量演化中导致的水平集函数奇异性问题。式(3-13)是一种 Hamilton-Jacobi 类型的偏微分方程的数值解, 正确处理水平集函数奇异性问题必须基于波前传播的“熵守恒”理论, 熵守恒原理应用在偏微分方程的求解上, 就是偏微分方程的迟滞解方法。

下面介绍式(3-10)的迟滞解方法:

$$\text{设 } D_{i,j}^{-x} = \phi_{i,j} - \phi_{i-1,j}$$

$$D_{i,j}^{+x} = \phi_{i+1,j} - \phi_{i,j}$$

$$D_{i,j}^{0x} = (\phi_{i+1,j} - \phi_{i-1,j})/2$$

$$D_{i,j}^{-y} = \phi_{i,j} - \phi_{i,j-1}$$

$$D_{i,j}^{+y} = \phi_{i,j+1} - \phi_{i,j}$$

$$D_{i,j}^{0y} = (\phi_{i,j+1} - \phi_{i,j-1})/2$$

用来表示对 ϕ 的有限差分操作数, 则式(3-10)可以重写为

$$\phi_t + H(\phi_x, \phi_y) = 0 \quad (3-14)$$

上式称为 Hamilton-Jacobi 方程, $H(\cdot)$ 称为哈密顿操作数, 这里给出其一阶近似解。对于图像分割应用, 由于方程(3-10)的速度函数通常可以写成如下形式

$$V = V_{prop} + V_{curve} + V_{adv} \quad (3-15)$$

$V_{prop} = V_0$ 是常量扩张速度(气压力), $V_{curve} = -\varepsilon k$ 是曲率速度, $V_{adv} = \mu(x, y, t) \cdot N$ 是水平对流速度, $\mu(x, y, t) = (u(x, y, t), v(x, y, t))$, 则方程(3-10)的近似解为

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + \Delta t \left\{ \begin{array}{l} - \left[\max(V_{oj,0}) \nabla^+ + \min(V_{oj,0}) \nabla^- \right] \\ \left[\max(u_{ij}^n, 0) D_{ij}^{-x} + \min(u_{ij}^n, 0) D_{ij}^{+x} \right] \\ \left[\max(v_{ij}^n, 0) D_{ij}^{-y} + \min(v_{ij}^n, 0) D_{ij}^{+y} \right] \\ + \varepsilon k_{i,j} \left[(D_{i,j}^{0y})^2 + (D_{i,j}^{0x})^2 \right]^{1/2} \end{array} \right\} \quad (3-16)$$

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + \Delta t \left\{ \begin{array}{l} - \left[\max(V_{oj,0}) \nabla^+ + \min(V_{oj,0}) \nabla^- \right] \\ \left[\max(u_{ij}^n, 0) D_{ij}^{-x} - \min(u_{ij}^n, 0) D_{ij}^{+x} \right] \\ \left[\max(v_{ij}^n, 0) D_{ij}^{-y} + \min(v_{ij}^n, 0) D_{ij}^{+y} \right] \\ + \varepsilon k_{i,j} \left[(D_{i,j}^{0y})^2 + (D_{i,j}^{0x})^2 \right]^{1/2} \end{array} \right\}$$

式中

$$\nabla^+ = \left[\max(D_{i,j}^{-x}, 0)^2 + \min(D_{i,j}^{+x}, 0)^2 + \max(D_{i,j}^{-y}, 0)^2 + \min(D_{i,j}^{+y}, 0)^2 \right]^{1/2}$$

$$\nabla^- = \left[\min(D_{i,j}^{-x}, 0)^2 + \max(D_{i,j}^{+x}, 0)^2 + \min(D_{i,j}^{-y}, 0)^2 + \max(D_{i,j}^{+y}, 0)^2 \right]^{1/2}$$

这种差分方法称为逆向有限差分法(Upwind Finite Difference)。通过差分方程(3-16), 可以利用迭代法来不断更新水平集函数, 更新完成后, 利用轮廓检测方法提取更新后的水平集函数的零水平集, 这样就得到更新后的闭合活动轮廓线。而且, 闭合活动轮廓线的曲率可以直接采用中值差分方法来数值计算, 即采用 D_{ij}^{0x} 和

D_{ij}^{0y} 来近似, 而内向单位法向量则可以按照下式计算

$$\begin{aligned} n_{ij}^* &= \frac{(\phi_x, \phi_y)}{\sqrt{\phi_x^2 + \phi_y^2}} \\ &= \frac{(D_y^{+x}, D_y^{+y})}{\sqrt{(D_y^{+x})^2 + (D_y^{+y})^2}} + \frac{(D_y^{-x}, D_y^{+y})}{\sqrt{(D_y^{-x})^2 + (D_y^{+y})^2}} \\ &\quad + \frac{(D_y^{+x}, D_y^{-y})}{\sqrt{(D_y^{+x})^2 + (D_y^{-y})^2}} + \frac{(D_y^{-x}, D_y^{-y})}{\sqrt{(D_y^{-x})^2 + (D_y^{-y})^2}} \end{aligned} \quad (3-17)$$

归一化得

$$N_{ij} = \frac{n_{ij}^*}{|n_{ij}^*|} \quad (3-18)$$

3.3 水平集函数演化的快速算法

直接根据数值方法计算水平集函数的演化, 需要对整个图像定义域中的所有点的水平集函数值进行更新, 这种方式的计算量比较大, 特别是在图像比较大的情况下。事实上, 在水平集函数零水平集的演化过程中, 如果对整个水平集函数更新, 显然会产生很多无效的计算。因此, 人们研究出许多种快速演化的方法。

常用的两种快速演化水平集方法的数值计算方法是窄带法(Narrow Band)和快速行进法(Fast Marching)。

3.3.1 窄带法

窄带法最初由 Chop 提出^[33], Adalsteinsson 和 Sethian 给出详细的实现方法。基本思想是只演化位于零水平集周围很窄的一个带状区域水平集函数的值。更新水平集函数的时候, 只对带状阴影内部的网格点(又称启动点)更新, 如图 3.2 所示。

由于窄带宽度一般较窄, 窄带内需要新的启动点不多, 因此, 更新水平集函数的计算量大大减小。但是, 窄带法也存在一些问题, 在经过几次迭代后, 零水平集的位置可能超过窄带的范围, 因此, 需要经常更新窄带状水平集函数, 并且它是一种局部而不是全局求解水平集的方法。

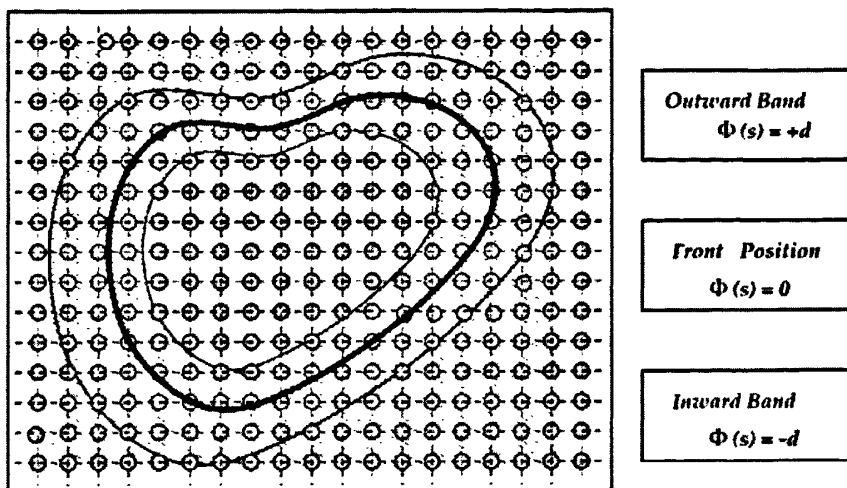


图 3.2 窄带法示意图

3.3.2 快速行进法

这里考虑曲线演化的特殊情况，式(3-6)中的速度 $V(x)$ 总是保持符号不变，即总是正值，或总是负值，则式(3-6)称为静态 Hamilton-Jacobi 方程，因为时间变量可以消除。

Sethian 提出了求解该类方程边界值问题的快速行进法(Fast Marching)^[34]。如果 $T(x)$ 表示零水平集通过点 x 的时间边界，则 $T(x)$ 满足如下方程

$$|\nabla T|V = 1 \quad (3-19)$$

利用逆向差分法(Upwind Difference)，得到

$$\left[\max(D_{i,j}^- T, 0)^2 + \min(D_{i,j}^+ T, 0)^2 + \max(D_{i,j}^- T, 0)^2 + \min(D_{i,j}^+ T, 0)^2 \right]^{1/2} = \frac{1}{V_{i,j}} \quad (3-20)$$

式中 D^- 和 D^+ 分别是后向与前向差分，例如 $D_{i,j}^- T = \frac{T_{i,j} - T_{i-1,j}}{h}$ ， $D_{i,j}^+ T = \frac{T_{i,j+1} - T_{i,j}}{h}$ ，

h 是网格大小。

由式(3-20)可知，边界传播方向是从 T 比较小的点向 T 比较大的点运动，根据这样的特点，Sethian 提出了快速行进法来迅速传播边界值 T 。它的基本思想是在传播边界外围构造一个启动窄带，窄带内的点的到达时间未定，利用逆向机制(Upwind Scheme)将当前边界向外传播，就像水波扩散一样，凡是扩散到的点，就冻结其波前到达时间，然后再根据当前的波前构造新的启动带，如此循环往复，就可以得到整个平面上各点的到达时间 T ，如图 3.3 所示。

图中， $T(x,y)$ 表示零水平集通过点 (x,y) 的到达时间，为一个锥形曲面，它在

初始曲线(蓝色区域)处横断 xy 平面。在任意高度 T 处, 曲面表示在时间 T 到达的点的集合, 称为到达时间曲面。

下面具体介绍快速行进法算法的步骤^{[35][36][37]}。根据网格中各点距离闭合曲线 C_0 的远近, 将网格点分为三类(如图 3.4 所示):

- ① 启动(Alive)点 已经被波前传播过的点, 到达时间已知;
- ② 活跃(Active)点 当前波前即将经过的点, 到达时间未知;
- ③ 远(Faraway)点 远离波前的点, 到达时间未知。

根据上述定义, 快速行进法的基本步骤为:

(1) 初始化

- ① 将 C_0 所在网格点标记为“Alive”, 并且时间 $T=0$ 。
- ② 考察每个“Alive”点的四邻点, 如果有标记不是“Alive”的, 则以“Active”标记之, 并赋予到达时间 $T_{ij} = \frac{1}{V_{ij}}$ 。然后将所有标记为“Active”的点放入一个排序堆中, 排序堆按照各点的 T 从小到大的顺序排序。
- ③ 剩余的既不是“Active”也不是“Alive”标记的点, 标记为“Faraway”, 到达时间标记为 $T_{ij} = \infty$ 。

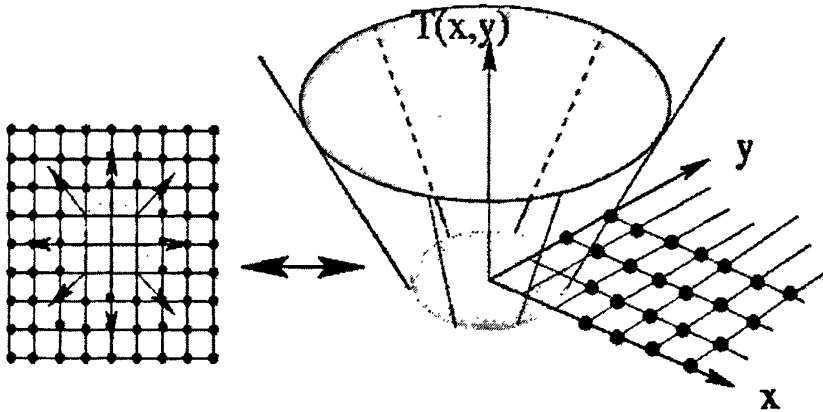


图 3.3 快速行进法示意图

(2) 前向行进

- ① 输出 T 最小的“Active”点。从排序堆输出窄带区中 T 最小的点 $P_{\min}(i_{\min}, j_{\min})$, 标记为“Alive”, 并从排序堆删除。
- ② 更新邻点。选择点 P_{\min} 的四连通邻点 $(i_{\min}-1, j_{\min})$, $(i_{\min}, j_{\min}-1)$, $(i_{\min}+1, j_{\min})$, $(i_{\min}, j_{\min}+1)$ 中标记不是“Alive”的点 N_p , 计算其到达时间 T ; 计算时, 要选择 P_{\min} 的四邻接点中标记为“Alive”的邻点的 T 代入方程(3-20)。计算完毕后, 如果 N_p 的当前标记为“Faraway”, 则将其修改为“Active”, 插入排序堆中; 如果当前标记是“Active”, 则利用新的 T 值更新该点的 T , 并重新调整 N_p 在排序堆中的位置, 保持排序堆的顺序。

③ 收敛检查。如果 $T_p > T_{threshold}$ ，或排序堆空，则行进结束，否则重新行进。

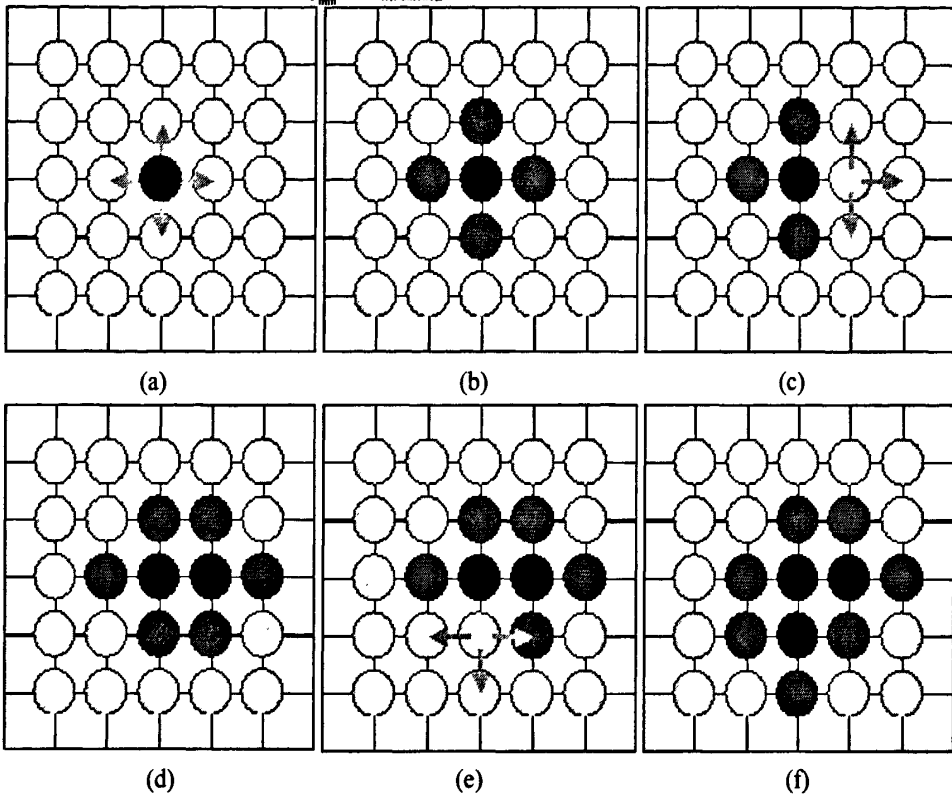


图 3.4 快速行进法步骤示意图

图 3.4 中的黑色圆圈表示启动(Alive)点，深灰色圆圈表示活跃(Active)点，灰色圆圈代表远(Faraway)点。图 3.4(a)表示快速行进法的开始，图 3.4(b)、(c)、(d)、(e)、(f)表示快速行进法的演化过程。

从快速行进法的具体步骤中，我们很容易知道，在最坏的情况下，快速行进法需要遍历一幅图像的所有像素，当然，这也取决于速度函数的定义以及相应参数的取值。

快速行进法的关键在于对速度函数的选取。由于其速度函数选取了图像的梯度信息，也就是图像的边缘信息，而没有充分地利用图像区域的全局信息，这必然导致了在图像灰度比较接近、边界模糊的地方，分割结果发生错误。

3.4 水平集方法的优缺点

水平集方法描述曲面运动具有以下几个优点：

(1) $\phi(x,t)$ 保持为光滑函数， $\phi(x,t)=0$ 所形成的等值面始终与运动曲面相对应，可以很容易地描述曲面的拓扑变化；

(2) 在计算过程中，可以利用有限差分法结合离散网格在数值上的近似求解，另外，函数 ϕ 的梯度也可以利用空间导数得到很好的近似；

(3) 通过水平集函数 $\phi(x, t)$, 可以比较容易地获得曲面的几何特征, 通常情况下, 曲面的法向单位向量为 $h = \frac{\nabla\phi}{|\nabla\phi|}$, 而曲面相对应的曲率为 $k = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}$;

(4) 低维计算时需要不断的重新参数化, 而高维中不需要, 而且高维中的计算结果更加精确;

(5) 水平集方法能够很好地描述二维空间中的曲线运动和三维空间中的曲面运动。

水平集方法描述曲面运动具有以下几个不足:

- (1) 水平集分割结果易产生边界泄露现象;
- (2) 对于多类型图像分割问题, 存在速度场参数的选择;
- (3) 由于水平集的运算量比较大, 所以分割速度较慢。

水平集方法的一个重要不足是时间花费极其惊人。因为它不仅是针对边界零水平集的函数求解, 而且还要针对整个图像所有的水平集。因此, 如何提高水平集方法的运算速度, 就成了后来相当长一段时间里, 乃至如今都必须认真研究的一个重要课题。

3.5 小结

本章主要讨论了基于水平集方法的医学影像分割方法。第一节对水平集理论做了简单介绍; 第二节介绍了曲线演化理论, 其中包括水平集理论的基本原理、水平集方法的数值计算; 第三节介绍了窄带法和快速行进法两种水平集函数演化的快速算法; 第四节则详细阐述了水平集方法的优缺点。



第四章 无需重新初始化的变分水平集方法研究

4.1 变分水平集方法简介

在基于偏微分方程(PDE)的图像分割方法中,活动轮廓模型是一种典型的基于变分和水平集方法的图像分割模型。一般地,活动轮廓模型根据其表达和实现方式的不同可分为两种,一种是参数化的 Snake 模型(参数活动轮廓模型, Parametric Active Contour), 另一种是几何活动轮廓模型(Geometric Active Contour)。

传统的参数化 snake 模型是一种典型的基于变分的能量极小化模型。其主要思想是,为给定的模型构造特定的能量函数,然后利用变分法对能量函数极小化,得到模型演化的偏微分方程。这种基于变分的能量极小化图像分割模型的最大优点是,将图像分割问题转化为能量函数的极小化问题,能量极小化主要通过变分法实现,那么,使能量函数达到极小值的解即为图像分割的结果。另外,在能量模型中可以方便地融入与模型自身相关的内部变形能量和外部约束能量,使模型在总能量的驱动下朝目标边缘演化,从而实现图像的分割。然而,由于传统的 Snake 方法是基于 Lagrange 的表示方式,所以很难处理演化曲线拓扑结构发生变化的情形。

用水平集表达的几何活动轮廓模型,曲线拓扑结构的变化可以在演化过程中得到自然的处理。其基本思想是,将平面曲线的演化隐含地表示在高维超平面水平集函数中,通过特定的偏微分方程演化水平集函数,间接地求解曲线的演化。

早期的几何活动轮廓模型,利用曲线演化理论,先得到一个用拉格朗日方式描述的二维曲线演化的偏微分方程,再利用水平集方法将其转化为相应的水平集表达形式;或者利用变分法,极小化一个关于平面参数化曲线的能量函数,得到曲线演化的梯度下降流,然后再转化为水平集演化的欧氏表达。上述两种方式都是分两个步骤得到水平集函数的演化方程。

然而,水平集函数演化的偏微分方程也可以直接通过极小化关于水平集函数自身的能量函数而得到。这样,由于这是一个基于变分的能量极小化模型,故具有上述参数化 Snake 模型的优点;由于采用水平集函数的表达方式,所以又具有水平集方法的优点,因此是一种非常理想的方法,这种方法称为变分水平集方法^[38]。变分水平集方法的关键是,首先建立一个能量模型,并且能量模型中的内、外部能量都使用水平集函数表示,再利用变分法对这个能量函数极小化,便得到水平集演化的偏微分方程。

相对于传统的由纯粹的 PDE 驱动的水平集图像分割方法,基于变分水平集的

图像分割方法可以在能量函数中自然地融入附加约束信息,如基于图像区域、边缘或物体形状先验知识的信息,因而可以产生鲁棒性更强的结果。本章我们将介绍一种无需重新初始化的变分水平集模型。

4.2 无需重新初始化的变分水平集图像分割方法

4.2.1 算法原理

传统的水平集方法在水平集每次迭代以后,都需要对水平集函数进行重新初始化(Re-initialization),以此来修正迭代造成的水平集位置的偏离,从而确保水平集在迭代以后不会偏离 SDF。虽然这样保证了演化计算的稳定性,但也花费了很多时间,而且重新初始化也非常复杂,严重阻碍了人们对水平集方法的广泛应用。

J Gomes 与 O Faugeras 指出^[39],重新初始化水平集函数显然是水平集理论与实践的矛盾之处。在理论上,重新初始化是完全没有必要的;在实践中,为了保证水平集在演化过程中不会偏离 SDF 太远,时间步长必须选得足够小,这样做又增加了迭代次数,从而增加了演化的时间。

虽然,学者们提出了多种重新初始化方法,但是这些方法往往会使得零水平集偏离其原始位置,同时如何适时地、正确地实施重新初始化也是一个比较麻烦的问题。下面我们介绍一种无需重新初始化的基于图像边缘信息的几何活动轮廓算法^[40]。

1. 内部能量

设 ϕ 为水平集函数,符号距离函数(SDF)满足一个非常好的性质,即 $|\nabla\phi|=1$;反之,满足 $|\nabla\phi|=1$ 的任意函数 ϕ 都可表示成一个 SDF 加上一个常数^[41]。所以引出如下积分

$$P(\phi) = \int_{\Omega} \frac{1}{2} (|\nabla\phi| - 1)^2 dx dy \quad (4-1)$$

作为函数 ϕ 与 Ω 中的 SDF 接近程度的一个衡量标准。这一标准在我们的变分水平集公式中将扮演重要角色。

根据上述定义的函数 $P(\phi)$,我们提出以下变分公式

$$\varepsilon(\phi) = \mu P(\phi) + \varepsilon_m(\phi) \quad (4-2)$$

式中, $\mu > 0$ 为一常数,它控制着“惩罚”水平集 ϕ 偏离 SDF 的力度, $P(\phi)$ 称为函数 ϕ 的内部能量,在曲线演化过程中,内部能量 $\mu P(\phi)$ 用来控制水平集函数,使其保持为 SDF;基于图像数据的能量 $\varepsilon_m(\phi)$ 用于驱动 ϕ 的零水平集曲线向图像的边缘演化,我们称它为外部能量。

我们用 $\frac{\partial \varepsilon}{\partial \phi}$ 表示函数 ε 的 Gateaux 导数, 以下演化方程

$$\frac{\partial \phi}{\partial t} = -\frac{\partial \varepsilon}{\partial \phi} \quad (4-3)$$

是最小化函数 ε 的梯度流^[42]。因此, 用 Gateaux 导数明确定义的特殊函数 $\varepsilon(\phi)$ 可以根据函数 ϕ 以及它的导数来计算和表达。

函数 ϕ 根据式(4-3)表示的梯度流进行演化, 在 ϕ 的演化过程中, 零水平集曲线通过外部能量 $\varepsilon_m(\phi)$ 进行演化; 与此同时, 由于内部能量的“惩罚”作用, 演化函数 ϕ 将自动保持为一个近似的 SDF。因此, 在提出的公式中重新初始化步骤被完全消除了, 这一概念将在下面详细阐述。

2. 外部能量

在图像分割中, 活动轮廓是向目标边缘运动的动态曲线。为达到这一目标, 我们明确定义了一个能驱使零水平集曲线向目标边缘运动的外部能量。

设 I 为原始图像, g 为边缘检测函数, 定义如下

$$g = \frac{1}{1 + |\nabla G_\sigma * I|^2} \quad (4-4)$$

式中, G_σ 是标准差为 σ 的高斯核。当轮廓线位于图像的边缘处时, 图像的高斯梯度较大, 边缘检测函数 g 较小, 曲线的演化速度几乎为零, 轮廓线就会停止在图像的边缘处。

我们定义函数 $\phi(x, y)$ 的外部能量如下

$$\varepsilon_{g,\lambda,\nu}(\phi) = \lambda L_g(\phi) + \nu A_g(\phi) \quad (4-5)$$

式中, $\lambda > 0$ 和 ν 均为常数。 $L_g(\phi)$ 和 $A_g(\phi)$ 定义如下

$$L_g(\phi) = \int_{\Omega} g \delta(\phi) |\nabla \phi| dx dy \quad (4-6)$$

$$A_g(\phi) = \int_{\Omega} g H(-\phi) dx dy \quad (4-7)$$

式中, δ 为单变量 Dirac 函数, H 为 Heaviside 函数。

假设 ϕ 的零水平集可以用可微的参数化曲线 $C(p), p \in [0, 1]$ 表示, 那么 $L_g(\phi)$ 表示的是在正形投影公式 $ds = g(C(p)) |C'(p)| dp$ 中 ϕ 的零水平集曲线的长度^[43], 这就是能量 $L_g(\phi)$ 的几何意义。

能量函数 $A_g(\phi)$ 用来加速曲线的演化, 可以看出, 当边缘检测函数 g 为常数 1 时, $A_g(\phi)$ 即为 $\Omega_\phi = \{(x, y) | \phi(x, y) < 0\}$ ^[44] 区域的面积, 因此, 我们可以把 $A_g(\phi)$ 看

成是区域 Ω_c 的加权面积。 $A_g(\phi)$ 的系数 v 可正可负，它取决于初始轮廓线相对于待分割目标的位置。一般来说，当初始轮廓线位于目标外部时， v 为正值，这样可以促使轮廓线快速收缩；反之，当初始轮廓线位于目标内部时， v 为负值，可以加快轮廓线的膨胀速度。

于是，我们得到完整的能量函数

$$\begin{aligned} \varepsilon(\phi) &= \mu P(\phi) + \lambda L_g(\phi) + v A_g(\phi) \\ &= \mu \int_{\Omega} \frac{1}{2} (|\nabla\phi| - 1)^2 dx dy + \lambda \int_{\Omega} g\delta(\phi) |\nabla\phi| dx dy + v \int_{\Omega} gH(-\phi) dx dy \end{aligned} \quad (4-8)$$

将式(4-8)代入式(4-3)，函数 ϕ 满足欧拉-拉格朗日方程 $\frac{\partial \varepsilon}{\partial \phi} = 0$ ，最小化能量函数

$\varepsilon(\phi)$ ，可得

$$\frac{\partial \phi}{\partial t} = \mu \left[\Delta\phi - \operatorname{div} \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \right] + \lambda \delta(\phi) \operatorname{div} \left(g \frac{\nabla\phi}{|\nabla\phi|} \right) + v g \delta(\phi) \quad (4-9)$$

式中， Δ 为 Laplacian 算子， $\nabla\phi$ 是 ϕ 的梯度， div 为散度。这一梯度流即是零水平集函数的演化方程。

第二、三项由外部能量而来，用于驱动 ϕ 的零水平集曲线向目标边缘运动。第一项中的

$$\Delta\phi - \operatorname{div} \left(\frac{\nabla\phi}{|\nabla\phi|} \right) = \operatorname{div} \left[\left(1 - \frac{1}{|\nabla\phi|} \right) \nabla\phi \right] \quad (4-10)$$

为一个扩散项，因子 $1 - \frac{1}{|\nabla\phi|}$ 可以理解为扩散率。

当 $|\nabla\phi| > 1$ 时，扩散率为正值，这个扩散项就是一般意义上的扩散，它使水平集函数 ϕ 更加平坦，于是 $|\nabla\phi|$ 减小；反之，当 $|\nabla\phi| < 1$ 时，反向扩散使得 $|\nabla\phi|$ 增大。通过这种扩散调节，使得 $|\nabla\phi| \approx 1$ ，从而使水平集函数 ϕ 始终接近符号距离函数。

4.2.2 算法实现

1. 偏微分方程的求解

该算法采用有限差分的方法来求解式(4-9)。在实际应用中，Dirac 函数采用正则化的 $\delta_\varepsilon(x)$ 进行计算， $\delta_\varepsilon(x)$ 定义如下：

$$\delta_\varepsilon(x) = \begin{cases} 0, & |x| > \varepsilon \\ \frac{1}{2\varepsilon} \left[1 + \cos\left(\frac{\pi x}{\varepsilon}\right) \right], & |x| \leq \varepsilon \end{cases} \quad (4-11)$$

本章中所有实验选取 $\varepsilon = 1.5$ 。

所有的空间偏导数 ϕ_x^n 、 ϕ_y^n 用中心差分来近似计算^[45]，即

$$\phi_x^n = \frac{\phi_{i+1,j}^n - \phi_{i-1,j}^n}{2\Delta h}, \quad \phi_y^n = \frac{\phi_{i,j+1}^n - \phi_{i,j-1}^n}{2\Delta h} \quad (4-12)$$

时间偏导数 $\frac{\partial\phi}{\partial t}$ 用前向差分来近似计算，即 $\frac{\partial\phi}{\partial t} = \frac{\phi^{n+1} - \phi^n}{\tau}$ 。

于是，可以得到曲线演化的迭代方程

$$\begin{cases} \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\tau} = \mu \left[\operatorname{div} \left[\left(1 - \frac{1}{|\nabla\phi_{i,j}^n|} \right) \nabla\phi_{i,j}^n \right] \right] + \lambda \delta_\varepsilon(\phi_{i,j}^n) \operatorname{div} \left(g \frac{\nabla\phi_{i,j}^n}{|\nabla\phi_{i,j}^n|} \right) + \nu g \delta(\phi_{i,j}^n) \\ \phi(0, x, y) = \phi_0(x, y) \end{cases} \quad (4-13)$$

2. 时间步长的选择

在实现上述提出的水平集方法时，可以选择的时间步长 τ 的范围明显大于在传统水平集方法中使用的时间步长范围，在这个算法中， τ 可以设在 0.1~100 之间。实验中，为了保持水平集演化的稳定性，我们发现，时间步长 τ 和系数 μ 必须满足 $\tau\mu < \frac{1}{4}$ 。

采用较大的时间步长可以加速演化，但是，如果时间步长的选择过大，在边缘定位时可能出错，因此，选择更大的时间步长与边缘的精确定位是一对矛盾。一般地，对于大多数图像我们选择 $\tau \leq 10.0$ 。

3. 水平集函数的初始化

在传统的水平集方法中，必须将水平集函数 ϕ 初始化为符号距离函数(SDF) ϕ_0 ，如果初始水平集函数明显不同于 SDF，那么，重新初始化步骤也不能将这个函数重新初始化为一个 SDF。在我们的算法中，不仅重新初始化步骤得以完全消除，而且水平集函数也不再需要初始化为 SDF。

这里，我们提出以下函数作为初始函数 ϕ_0 。令 Ω_0 是图像区域 Ω 的一个子集， $\partial\Omega_0$ 是 Ω_0 的所有边界点的集合，它们能够被一些简单的形态学操作有效识别。于是，初始函数 ϕ_0 定义如下：

$$\phi_0(x, y) = \begin{cases} -\rho & (x, y) \in \Omega_0 - \partial\Omega_0 \\ 0 & (x, y) \in \partial\Omega_0 \\ \rho & \Omega - \Omega_0 \end{cases} \quad (4-14)$$

式中, $\rho > 0$ 是一个常数。 ρ 一般选取 $\rho \geq 2\varepsilon$, ε 为正则化 Dirac 函数 $\delta_\varepsilon(x)$ 定义中的宽度。

与传统的由轮廓计算得来的 SDF 不同, 上述初始水平集函数是从图像区域 Ω 中的任意一个子区域 Ω_0 中计算而来, 水平集函数的这种基于区域的初始化方法不仅计算高效, 而且在一些情况下应用也较为灵活。例如, 如果感兴趣区域可以通过一些途径粗略和自动获取, 那么我们可以使用这些粗略获得的区域作为区域 Ω_0 来构造初始水平集函数 ϕ_0 。接下来, 初始水平集函数将根据演化方程稳定演化, 同时它的零水平曲线可以收敛于感兴趣区域的精确边缘。

虽然这种初始函数 ϕ_0 明显偏离了 SDF, 但是由于能量函数中的惩罚函数 $P(\phi)$ 的调节作用, 使水平集函数 ϕ 在其零水平集附近能够近似等于 SDF, 这对于边缘的定位没有太大影响。

4. 水平集函数收敛的判断

水平集函数收敛的判断条件为:

$$Q = \frac{\sum_{|\phi_{i,j}^n| < \text{const}} |\phi_{i,j}^{n+1} - \phi_{i,j}^n|}{M} \leq h^2 \tau \quad (4-15)$$

式中, M 为满足 $|\phi_{i,j}^n| < \text{const}$ 的网格点数目, 其中 const 为实验常数。实验中, 一般取 $h \leq \text{const} \leq 2h$, h 为空间步长, 即离散网格大小。

算法的实现步骤为:

- (1) 初始化: 设置初始轮廓, 可以设置为圆形、矩形或是一条非闭合曲线, 根据公式(4-14)计算水平集函数的初始值 ϕ^0 ;
- (2) 利用迭代公式(4-13), 计算 ϕ^{n+1} ;
- (3) 检查本次迭代是否收敛, 如果收敛, 则停止计算; 否则, 转至步骤 2, 继续计算;
- (4) 更新轮廓线, 获得分割图像。

4.2.3 实验结果与分析

我们用多幅图像对上述算法进行了实验, 获得了较好的分割效果。在所有实验中, 我们都采用式(4-14)所定义的初始化方法。

图 4.1 给出了一幅像素大小为 83×65 的显微镜下观测到的两个细胞图像的轮廓演化过程。可以看到, 两个细胞的某些边缘是非常模糊的, 我们用这个图像来论证这种方法在处理弱目标边缘时的鲁棒性。

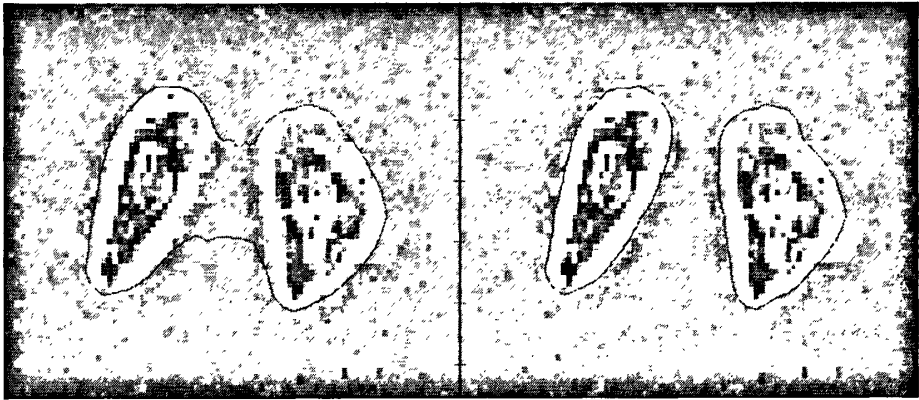
实验中, 我们选择了一个矩形区域作为区域 Ω_0 来计算初始水平集函数 ϕ_0 , 如图 4.1(a) 中所示。如图 4.1 所示, 演化曲线自动分裂, 将两个细胞分割出来。这一

结果表明, 我们的方法在提取弱目标边缘时获得了令人满意的分割结果, 而这用传统的水平集方法却很难实现。



(a) 初始轮廓

(b) 迭代 50 次

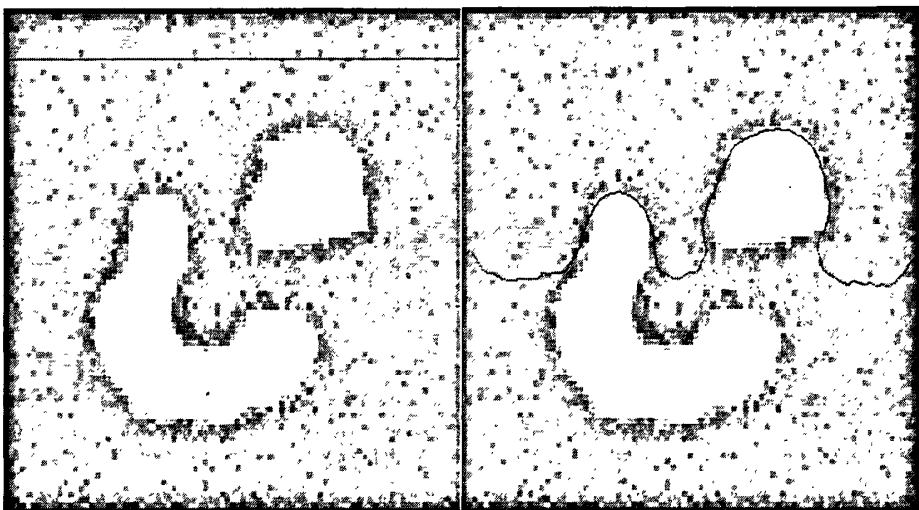


(c) 迭代 100 次

(d) 迭代 300 次

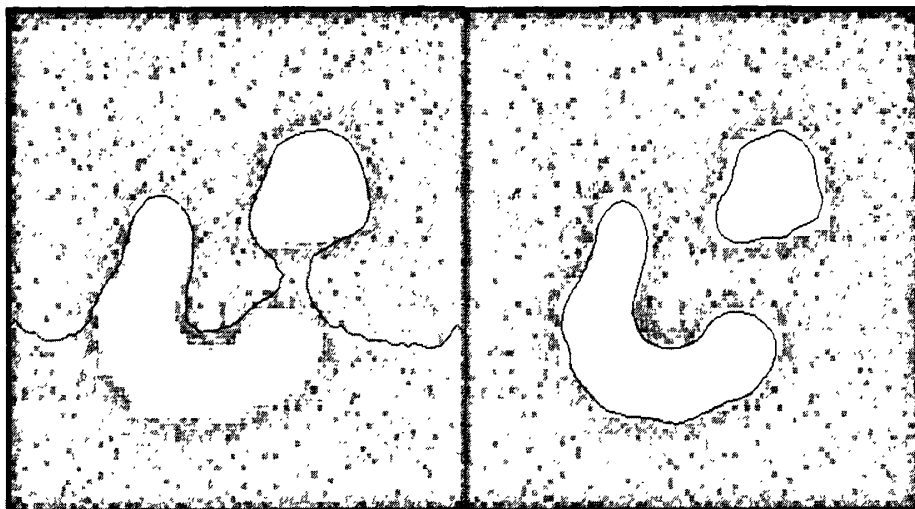
图 4.1 细胞图像(83×65)分割结果,

参数 $\lambda = 5.0$, $\mu = 0.04$, $\nu = 1.5$, 时间步长 $\tau = 5$



(a) 初始轮廓

(b) 迭代 140 次



(c) 迭代 180 次

(d) 迭代 400 次

图 4.2 两个目标物图像(84×84)分割结果,

参数 $\lambda = 5.0$, $\mu = 0.04$, $\nu = 3.0$, 时间步长 $\tau = 5$

图 4.2 给出了一幅像素大小为 84×84 的两个目标物图像的轮廓演化过程。在实验中, 我们使用参数 $\lambda = 5.0$, $\mu = 0.04$, $\nu = 3.0$, 时间步长 $\tau = 5$, 这个时间步长明显大于在传统水平集方法中使用的时间步长。

在传统的水平集方法中, 初始轮廓线一般选取一条简单闭合曲线, 即不存在交叉的闭合曲线。而图 4.2(a) 中我们使用了一条非闭合的直线作为初始轮廓线, 那么这条直线以下的区域为区域 Ω_0 , 演化曲线将两个目标物都提取了出来, 这是传统水平集方法做不到的。

图 4.3 给出了一幅像素大小为 128×128 的脑部 CT 图像的去骨分割过程。在实验中, 我们使用参数 $\lambda = 5.0$, $\mu = 0.04$, $\nu = 3.0$, 时间步长 $\tau = 5$, 选择矩形区域作为区域 Ω_0 。

这个模型的最大优点是^[46], 式(4-9)使得真正影响水平集计算量的重新初始化步骤得以完全消除。只要初始化时给定一个适当的水平集函数, 则由于第一项内部能量的约束作用, 可以迫使它接近符号距离函数(SDF), 这也是变分水平集方法的一大优点, 而传统的水平集方法是无法做到这一点的。与传统的水平集方法相比, 式(4-9)对实物图像能够有很好的分割效果, 并且解决了传统方法无法分割带模糊边缘的图像的问题; 灵活的水平集函数初始化方法, 使得初始轮廓线的选择更加自由, 并且计算也简便了很多。

该模型有效地利用了图像的边缘梯度信息, 对图像边缘具有较好的局域化作用, 对具有较好对比度的图像一般具有不错的分割效果。但是, 由于它仅利用了图像的边缘梯度信息, 而没有考虑图像的全局区域信息, 使得模型对边缘模糊图像、噪声图像、纹理图像的分割效果不理想。

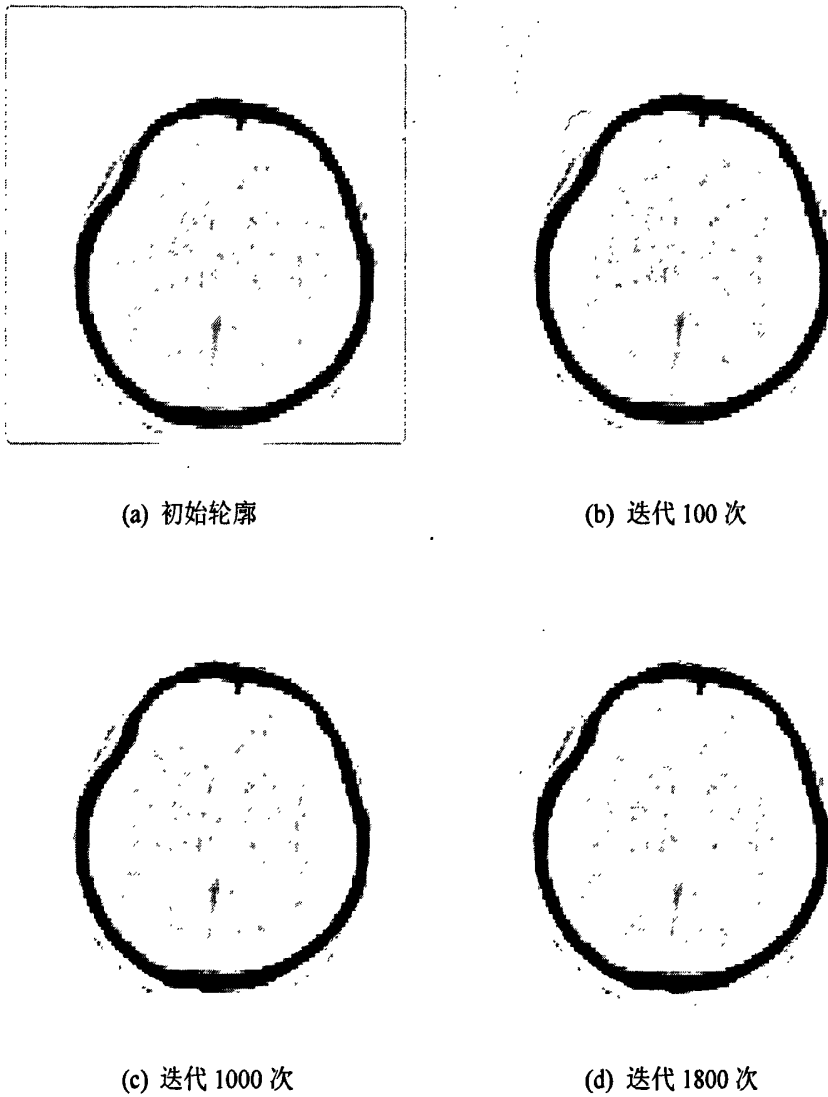


图 4.3 脑部 CT 图像(128×128)的去骨分割结果

参数 $\lambda = 5.0$, $\mu = 0.04$, $\nu = 3.0$, 时间步长 $\tau = 5$

式(4-2)中的 $\varepsilon_m(\phi)$ 主要依靠主动轮廓线的长度信息(式 4-5 的第一项表示长度, 第二项用来加速演化), 所以迭代次数很多, 影响了演化速度。本章中所列举的实验的迭代都至少数百次(不管图像是大是小), 这在很多情况下都会带来很多麻烦, 例如, 在图像边缘处可能造成一定程度的“过光滑”。

该模型中的初始轮廓线的选取依然受到很大限制。一般地, 初始轮廓线需包围所有待分割的物体, 而且初始轮廓线的设定不能跨越同质区域, 这些都大大影响了选取时的灵活性。另外, 对于某些深度凹陷, 常常会检测失败。这一模型采用的是显式迭代解法, 虽然其迭代时间步长可选取一个适当的较大值, 但超过一定值时, 会导致图像边缘定位不准确, 甚至严重影响图像分割的最终效果。

4.3 小结

本章我们介绍了一种无需重新初始化的基于图像边缘信息的几何活动轮廓算法。它是一个变分水平集模型，这个模型的最大优点是完全消除了真正影响水平集计算量的重新初始化步骤。本章首先对变分水平集方法做了简单介绍，然后对该算法的原理以及实现做了详细阐述，最后给出该算法的实验结果。实验结果表明，该方法在处理弱目标边缘时具有较强的鲁棒性。

第五章 基于稀疏场方法的水平集分割研究

体模型把空间划分为体素(voxel, 简称为 vx), 不妨设体素是单位大小正方体。Ross T. Whitaker 基于街区(City Block)距离模型计算的简单快速, 提出了稀疏场算法^[47], 试图把窄带宽度缩小到极限, 减少每次迭代时更新的体素, 同时可以提高每次迭代中时间步长的值, 这样就加快了演化过程。另外, 稀疏场算法还可以通过子分体素方式提高差分计算的阶数, 提高演化计算的精度。

5.1 传统稀疏场定义

基于理想的欧式距离模型, 稀疏场活动集定义^{[48][49][50]}如下:

$$L_0 = \left\{ vxl_{u,v,w} \mid -\frac{1}{2} \leq \phi_{u,v,w} \leq \frac{1}{2} \right\} \quad (5-1)$$

外层体素集定义如下:

$$L_n = \begin{cases} \left\{ vxl_{u,v,w} \mid n - \frac{1}{2} < \phi_{u,v,w} \leq n + \frac{1}{2} \right\} & n > 0 \\ \left\{ vxl_{u,v,w} \mid n + \frac{1}{2} > \phi_{u,v,w} \geq n - \frac{1}{2} \right\} & n < 0 \end{cases} \quad (5-2)$$

其中, n 是整数。这样的定义体现了作者提出稀疏场算法的初衷, 在理想距离模型中是严格的。但是在多维欧式空间的实际计算中, 由于数值计算的非连续性和计算精度, 该定义是不严格的, 有一定的冗余。由于活动集必须满足拓扑分离性, 所以在欧式距离模型中比较可靠的距离定义如下:

$$L_0 = \left\{ vxl_{u,v,w} \mid -\frac{\sqrt{3}}{2} \leq \phi_{u,v,w} \leq \frac{\sqrt{3}}{2} \right\} \quad (5-3)$$

同样, 外层体素集在多维欧氏空间不能用距离定义。Whitaker 基于距离值符号关系定义活动集。记一个体模型为 $VM = \{P, \phi\}$, P 是所有体素中心点的集合, ϕ 是 X 中每个元素的距离值; 单位向量 $e_1 = \{1, 0, 0\}$, $e_2 = \{0, 1, 0\}$, $e_3 = \{0, 0, 1\}$, 每个体素的 6 连通邻定义为:

$$C(vxl_{u,v,w}) = \{ vxl_{r,s,t} \mid (r,s,t) = (u,v,w) \pm e_p, 1 \leq p \leq 3 \} \quad (5-4)$$

定义相邻算子为:

$$A(vxl_{u,v,w}, vxl_{r,s,t}) = \begin{cases} TRUE & vxl_{r,s,t} \in C(vxl_{u,v,w}) \\ FALSE & vxl_{r,s,t} \notin C(vxl_{u,v,w}) \end{cases} \quad (5-5)$$

定义 ϕ 值符号关系算子为

$$O(vxl_{u,v,w}, vxl_{r,s,t}) = \begin{cases} TRUE & \phi_{u,v,w} * \phi_{r,s,t} \leq 0 \\ FALSE & \phi_{u,v,w} * \phi_{r,s,t} > 0 \end{cases} \quad (5-6)$$

传统稀疏场算法活动集定义相当于如下形式:

$$L_0 = \{ vxl_{u,v,w} \mid \exists vxl_{r,s,t} (A(vxl_{u,v,w}, vxl_{r,s,t}) \wedge O(vxl_{u,v,w}, vxl_{r,s,t})) \} \quad (5-7)$$

式(5-7)定义的活动集是厚度为2的对象离散边界。而活动集内外分别需要厚度为1的外层体素集。

5.2 稀疏场算法

为方便起见,我们使用二维术语“曲线”来表示曲面,用曲线演化方法表示曲面演化方法,用像素表示体素。

在稀疏场方法(SFM)^[51]中,使用了几个点序列来表示零水平集以及零水平集邻近的点。通过使用这些序列并且谨慎地往适当的序列中移入或从序列中移出一些点,从而得到了一种非常有效的表示 ϕ 的方法。

这些序列的实现方式与双向链表类似。这些序列满足以下属性,元素可以动态添加,而且可以从序列中移出元素。这一类数据结构在大多数编程语言中可以获得(例如,C++中的向量类)。

在 SFM 中,五个不同的水平分别用五个序列表示:

$$\begin{aligned} L_0 &\rightarrow [-0.5, 0.5] \rightarrow Lz \\ L_{-1} &\rightarrow [-1.5, -0.5] \rightarrow Ln1 \\ L_1 &\rightarrow (0.5, 1.5] \rightarrow Lp1 \\ L_{-2} &\rightarrow [-2.5, -1.5] \rightarrow Ln2 \\ L_2 &\rightarrow (1.5, 2.5] \rightarrow Lp2 \end{aligned}$$

每个序列存储像素点在图像中的 x 、 y 和 z 方位。除序列外,还用到两个数组。首先是 ϕ 数组,它与图像区域大小相同。第二个数组称为标志映射,它与 ϕ 数组大小相同。这种标志映射用于记录每个点的状态,提供一种查找某个点属于某一序列的方法,标志映射仅包含这些值 $\{-3, -2, -1, 0, 1, 2, 3\}$ 。

5.2.1 初始化

步骤1处理一幅二值图像(init),并为标志映射(label)返回一个完全初始化的数组, ϕ (phi)和五个表示水平-2到2的序列。二值图像(init)有两个值0和1,1表示前景像素,0表示背景像素。此外,符号N(p)用于表示一点p的邻域,它表示二维

中的4邻域(上、下、左、右)和三维中的6邻域(上、下、左、右、前、后)。

步骤1 初始化

//先决条件 label和phi

- 1: 对于区域中各点p
- 2: if(init(p) == 0) label(p) = 3, phi(p) = 3
- 3: if(init(p) == 1) label(p) = -3, phi(p) = -3

//寻找零水平集

- 4: 对于区域中各点p
- 5: if(init(p) == 1 并且N(p)中任意点N(p) == 0)
- 6: 将p添加到Lz中
- 7: label(p) = 0, phi(p) = 0

//寻找+1 和-1 水平集

- 8: 对于Lz中各点p
- 9: 对于N(p)中各点q
- 10: if(label(q) == -3) 将q添加到Ln1中, label(q) = -1, phi(q) = -1
- 11: if(label(q) == 3) 将q添加到Lp1中, label(q) = 1, phi(q) = 1

//寻找+2和-2水平集

- 12: 对于Ln1中各点p
- 13: 对于N(p)中各点q
- 14: if(label(q) == -3) 将q添加到Ln2中, label(q) = -2, phi(q) = -2
- 15: 对于Lp1中各点p
- 16: 对于N(p)中各点q
- 17: if(label(q) == 3) 将q添加到Lp2, label(q) = 2, phi(q) = 2

5.2.2 轮廓演化

只要这一类数据结构通过步骤1完成初始化,水平集就可以通过变形来达到最小化一些分割能量的目的。这样的能量有许多种,而为活动轮廓分割算法选择合适的分割能量是众多研究中的首要问题^[52]。一种非常流行的活动轮廓能量称为Chan-Vese能量^[53],如下:

$$E = \int_{in} (I - \mu_1)^2 + \int_{ex} (I - \mu_2)^2 \quad (5-8)$$

第一项为内部能量，第二项为外部能量，对应的演化方程为

$$F = (I - \mu_1)^2 - (I - \mu_2)^2 \quad (5-9)$$

注意到， F 应该仅在零水平集边上计算，而且在每次迭代时应该归一化为这样 $\|F\| < 0.5$ 。

1. 在零水平集附近更新 ϕ

只要 F 的计算和归一化完成，步骤 2 就可以描述如何在零水平集边沿和其他 4 个序列的临近点上更新 ϕ 。另外，还有 5 个序列用于临时存储正在改变状态的点：

$$\begin{aligned} S_0 &= Sz \rightarrow \text{正向 } L_0 \text{ 运动的点} \\ S_{-1} &= Sn1 \rightarrow \text{正向 } L_{-1} \text{ 运动的点} \\ S_1 &= Sp1 \rightarrow \text{正向 } L_1 \text{ 运动的点} \\ S_{-2} &= Sn2 \rightarrow \text{正向 } L_{-2} \text{ 运动的点} \\ S_2 &= Sp2 \rightarrow \text{正向 } L_2 \text{ 运动的点} \end{aligned}$$

步骤 2 可总结如下：

- (1) 扫描 L_0 ，在各点处将对应的 F 值加到已存在的 ϕ 值上；
- (2) 将 ϕ 的新值在范围 $[-.5 \quad .5]$ 之外的点从 L_0 中移出，然后添加到 S_{-1} 或 S_1 中，这意味着这些点将分别改变状态到 L_{-1} 或 L_1 中；
- (3) 扫描 L_{-1} 和 L_1 ，更新 ϕ 值，这样这些点距离 L_0 中它们的最近邻相差正好一个单位。如果没有 L_0 邻存在，这些点将被分别移到 S_{-2} 或 S_2 中；
- (4) 如果 L_{-1} 和 L_1 中的点的 ϕ 值在更新后落在为 L_{-1} 或 L_1 指定的范围之外，那么这些点将被移到 S_0 、 S_{-2} 或 S_2 中；
- (5) 扫描 L_{-2} 和 L_2 ，更新 ϕ 值，这样这些点距离 L_{-1} 或 L_1 中它们的最近邻相差正好一个单位。如果没有 L_{-1} 或 L_1 邻存在，这些点将从所有序列中移出，从而 ϕ 值和标志映射将变为 -3 或 +3.；
- (6) 如果 L_{-2} 和 L_2 中的点的 ϕ 值在更新后落在指定的范围之外，那么这些点将从 L_{-2} 和 L_2 中移出。如果它们的值过低，将被移到 S_{-1} 或 S_1 ；如果它们的值过高，将从所有序列中移出。从所有序列中移出的点应该在 ϕ 中有它们对应的点，而且标志映射变为 -3 或 +3。

步骤 2 更新水平集序列

//更新零水平集

- 1: 对于 Lz 中各点 p
- 2: 将 $F(p)$ 加到 $\phi(p)$ 上

- 3: if($\phi(p) > .5$), 将 p 从 L_z 中移出, 将 p 添加到 $Sp1$
- 4: if($\phi(p) < -.5$), 将 p 从 L_z 中移出, 将 p 添加到 $Sn1$

//更新-1 和+1 水平集

- 5: 对于 $Ln1$ 中各点 p
- 6: 如果 p 在 $N(p)$ 中没有 $label(q) = 0$ 的邻
- 7: 将 p 从 $Ln1$ 中移出, 将 p 添加到 $Sn2$
- 8: 否则
- 9: $M =$ 在 $N(p)$ 中 $label(q) \geq 0$ 的最大点 q
- 10: $\phi(p) = M - 1;$
- 11: if($\phi(p) \geq -0.5$), 将 p 从 $Ln1$ 中移出, 将 p 添加到 Sz
- 12: if($\phi(p) < -1.5$), 将 p 从 $Ln1$ 中移出, 将 p 添加到 $Sn2$

- 13: 对于 $Lp1$ 中各点 p
- 14: 如果 p 在 $N(p)$ 中没有 $label(q) = 0$ 的邻
- 15: 将 p 从 $Lp1$ 中移出, 将 p 添加到 $Sp2$
- 16: 否则
- 17: $M =$ 在 $N(p)$ 中 $label(q) \leq 0$ 的最小点 q
- 18: $\phi(p) = M + 1;$
- 19: if($\phi(p) \leq 0.5$), 将 p 从 $Lp1$ 中移出, 将 p 添加到 Sz
- 20: if($\phi(p) > 1.5$), 将 p 从 $Lp1$ 中移出, 将 p 添加到 $Sp2$

//更新-2 和+2 水平集

- 21: 对于 $Ln2$ 中各点 p
 - 22: 如果 p 在 $N(p)$ 中没有 $label(q) = -1$ 的邻
 - 23: 将 p 从 $Ln2$ 中移出, $label(p) = -3$, $\phi(p) = -3$
 - 24: 否则
 - 25: $M =$ 在 $N(p)$ 中 $label(q) \geq -1$ 的最大点 q
 - 26: $\phi(p) = M - 1;$
 - 27: if($\phi(p) \geq -1.5$), 将 p 从 $Ln2$ 中移出, 将 p 添加到 $Sn1$
 - 28: if($\phi(p) < -2.5$), 将 p 从 $Ln2$ 中移出, $label(p) = -3$, $\phi(p) = -3$
- 29: 对于 $Lp2$ 中各点 p
 - 30: 如果 p 在 $N(p)$ 中没有 $label(q) = 1$ 的邻
 - 31: 将 p 从 $Lp2$ 中移出, $label(p) = 3$, $\phi(p) = 3$

- 32: 否则
- 33: $M =$ 在 $N(p)$ 中 $\text{label}(q) \leq 1$ 的最小点 q
- 34: $\text{phi}(p) = M+1;$
- 35: if($\text{phi}(p) \leq 1.5$), 将 p 从 L_{p2} 中移出, 将 p 添加到 S_{p1}
- 36: if($\text{phi}(p) > 2.5$), 将 p 从 L_{p2} 中移出, $\text{label}(p) = 3$, $\text{phi}(p) = 3$

2. 处理改变状态的点

一旦序列 L_0 、 L_{-1} 、 L_1 、 L_{-2} 和 L_2 中的所有点被访问, 就有必要去处理在迭代过程中已经改变状态的点。这在步骤 3 中描述, 总结如下:

- (1) 扫描 S_0 , 将各点移动到 L_0 上, 更新标志映射的值为 0;
- (2) 扫描 S_{-1} 和 S_1 , 将各点移动到 L_{-1} 或 L_1 上, 更新标志映射的值为 -1 或 +1。如果在 L_{-1} 或 L_1 上的点有 ϕ 值等于 -3 或 +3 的邻, 将这些邻添加到合适的 S_{-2} 或 S_2 序列中, 同时设置它们的 ϕ 值与当前点的值相差一个单位;
- (3) 扫描 S_{-2} 和 S_2 , 将各点移动到 L_{-2} 或 L_2 上, 根据符号更新标志映射的值为 -2 或 +2。

通过运行步骤 2 和步骤 3 完成一个完整的迭代过程, 而且 F 可以根据轮廓的新位置来重新计算。这一过程循环往复, 直到收敛完成。

步骤 3 处理改变状态的点

//将点移入零水平集

- 1: 对于 S_z 中各点
- 2: $\text{label}(p) = 0$, 将 p 添加到 L_z , 将 p 从 S_z 中移出

//将点移入 -1 和 +1 水平集

//并且确保 -2, +2 邻

- 3: 对于 S_{n1} 中各点
- 4: $\text{label}(p) = -1$, 将 p 添加到 L_{n1} , 将 p 从 S_{n1} 中移出
- 5: 对于 $N(p)$ 中各点 q
- 6: if($\text{phi}(q) = -3$) $\text{phi}(q) = \text{phi}(p) - 1$, 将 q 添加到 S_{n2}
- 7: 对于 S_{p1} 中各点
- 8: $\text{label}(p) = 1$, 将 p 添加到 L_{p1} , 将 p 从 S_{p1} 中移出
- 9: 对于 $N(p)$ 中各点 q
- 10: if($\text{phi}(q) = 3$) $\text{phi}(q) = \text{phi}(p) + 1$, 将 q 添加到 S_{p2}

//将点移入-2 和+2 水平集

11: 对于 S_{n2} 中各点

12: $label(p) = -2$, 将 p 添加到 L_{n2} , 将 p 从 S_{n2} 中移出

13: 对于 S_{p2} 中各点

14: $label(p) = 2$, 将 p 添加到 L_{p2} , 将 p 从 S_{p2} 中移出

3. 有效更新状态

在计算轮廓线的运动位移时, 当一个点越过零水平集时通常需要追踪, 这样这点将由一个内部点变为一个外部点, 反之亦然。在零水平集上的点将 F 值加到 ϕ 值上(步骤 2 中第 2 行)之前和之后, 这可以通过检查 ϕ 的符号来完成。

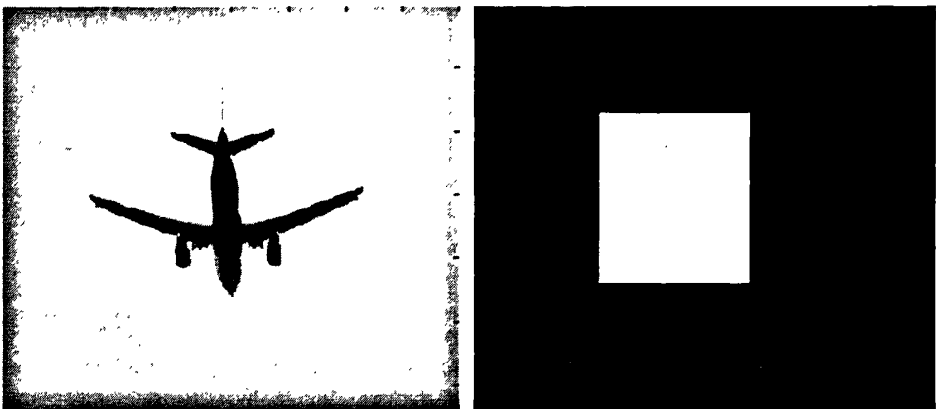
当一个点的符号由负变为正时, 可以将它添加到序列 L_{in2out} 中; 如果符号由正变为负则是 L_{out2in} 。接下来, 可以在迭代后处理这两个序列从而更加有效的更新状态。例如, 如果在计算 F 时使用了内部和外部平均值, 与每次迭代时重新计算内部和外部平均值相比, 基于个别点越过曲面的运动来改变状态则更加有效。

5.2.3 实验结果与分析

我们用多幅图像对上述算法进行了实验, 获得了较好的分割效果。

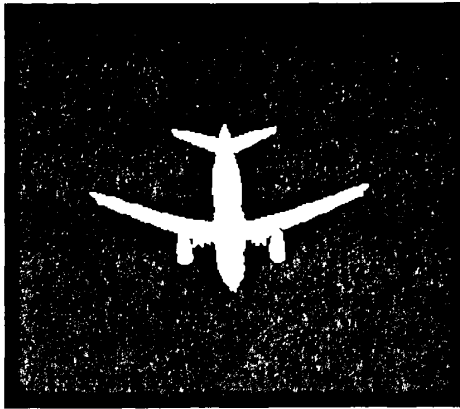
图 5.1 给出了一幅像素大小为 400×320 的飞机图像的分割结果。可以看到, 该算法得到了比较好的分割结果。在实验中, 为了方便比较, 用比较醒目的红色标记了分割结果, 我们使用参数 $\lambda = 0.3$, 迭代次数为 600 次。

图 5.2 给出了一幅像素大小为 400×400 的脑部 CT 图像的去骨分割过程。实验中, 我们使用参数 $\lambda = 0.3$, 迭代次数为 1400 次, 同样用红色标记分割结果。



(a) 原始图像

(b) 初始化



(c) 最终的分割结果



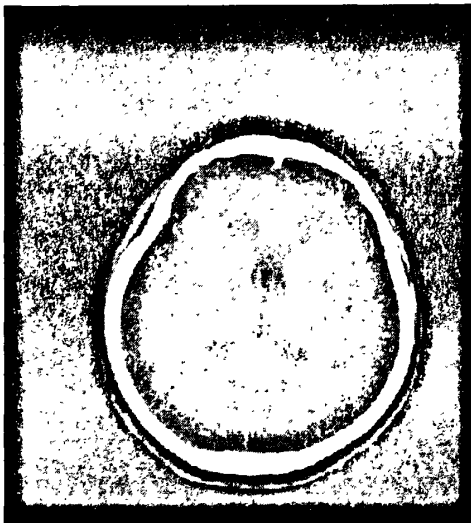
(d) 用红色标记分割结果的图像

图 5.1 飞机图像(400×320)分割结果,

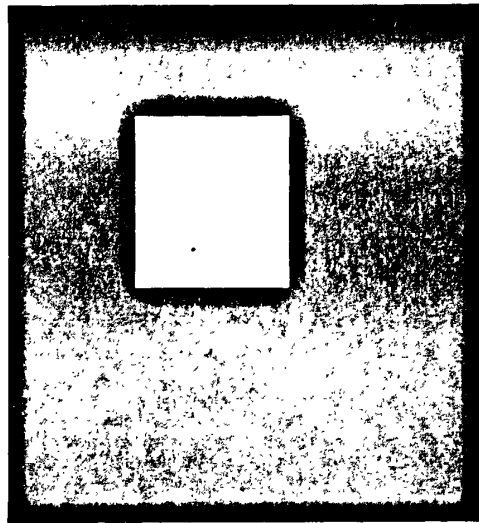
参数 $\lambda = 0.3$, 迭代次数 600 次

稀疏场方法(SFM)有几个非常好的特点使得它成为一种在图像处理中颇有价值的工具。事实上,它使用序列来一直追踪在零水平集附近的点,这意味着曲线更新的速度只取决于曲线的长度,而不是图像的大小,这在处理非常大的医学影像数据时很有价值。而且,这种方法的精确度如 Whitaker 的方法中所示,这表明,在不考虑特定曲线实现正在影响结果的前提下,通过使用 SFM,这样行为实现能量就很好理解了。

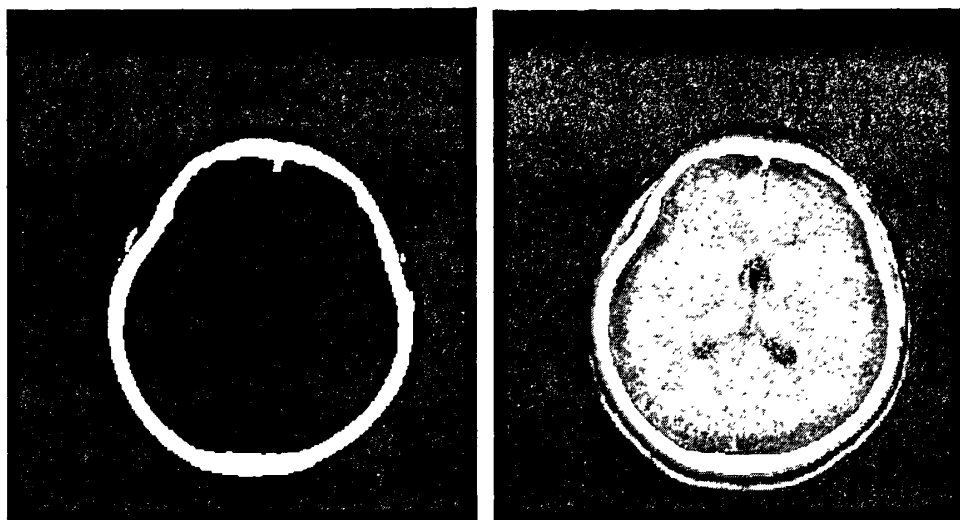
SFM 的一个限制条件是,由于力仅在零水平集的边沿上计算,这样就不太可能突然出现新的曲线。取而代之的是,初始轮廓可以仅通过生长、分裂和合并来得到它的初始形状。但是,这是在理论上,实际上,“正确”的行为应该是,通过使用一些实现来得到一种能力,使得新的轮廓线突然出现并在图像信息中填补漏洞和弥合缺口。



(a) 原始图像



(b) 初始化



(c) 最终的分割结果

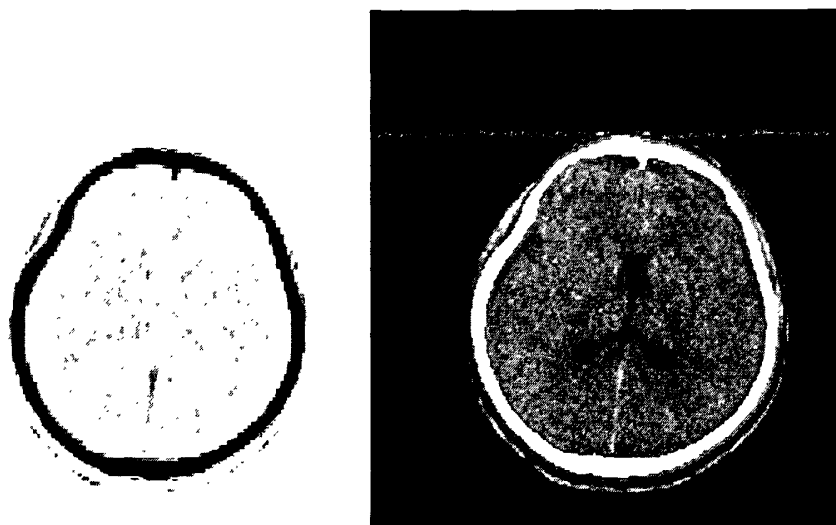
(d) 用红色标记分割结果的图像

图 5.2 脑部 CT 图像(400×400)的去骨分割结果

参数 $\lambda = 0.3$, 迭代次数 1400 次

比较图 4.3 和图 5.2 中脑部 CT 图像的分割效果可以看出, 基于稀疏场方法的水平集分割算法效果明显好于无需重新初始化的变分水平集方法。

图 5.3 对上述脑部 CT 图像分别利用变分水平集方法和稀疏场方法进行分割的结果做了比较。



(a) 变分水平集方法

(b) 稀疏场方法

图 5.3 脑部 CT 图像(400×400)分割结果比较

表 5.1 和表 5.2 分别对无需重新初始化的变分水平集方法和基于稀疏场方法的水平集分割算法的迭代次数和运行时间做了比较。从表中可以看出, 对于同一幅图像, 得到相似的分割结果时, 稀疏场方法迭代的次数明显少于变分水平集方法, 运行时间也明显少于变分水平集方法。

图像	变分水平集方法	稀疏场方法
脑部 CT(512×512)	1800	1400
飞机(400×320)	1000	600

表 5.1 迭代次数比较(单位: 次)

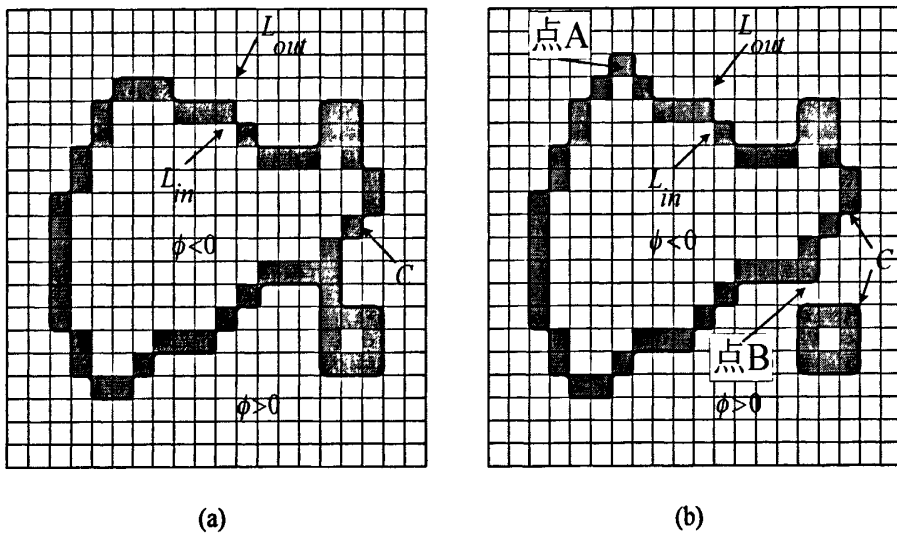
图像	变分水平集方法	稀疏场方法
脑部 CT(512×512)	600.6	3.4
飞机(400×320)	142.5	1.42

表 5.2 运行时间比较(单位: 秒)

5.3 相似技术

稀疏场方法(SFM)可以非常高效地实现完整的数值水平集方法。然而, 如果不要精确解的话, 可以获得更好的效果。有一种改进方法^[54], 保留 3 个序列, 分别对应零、+1 和-1 水平集。这一相似的方法使用整数值-1、0、和 1, 而不是在这些序列中表示 ϕ 用到的浮点型。若采用这一简化, 可以获得更快的速度。

在这一水平集方法中, 函数 ϕ 的零水平集表示为曲线 C , 并定义在如图 5.4(a) 中的一个固定网格中。这里我们选择 ϕ 为负时在曲线 C 内, 而 ϕ 为正时在曲线 C 外。

图 5.4: (a)水平集方法中曲线 C 的表示。两个相邻的网格点序列 L_m 和 L_{out} 可以在这一网格中

被唯一定义。(b)表示的曲线 C 的运动可以通过在 L_m 和 L_{out} 间转换点来实现。

由图 5.4(a)中我们可以看出, L_m 是在 C 内的相邻网格点序列, L_{out} 是在 C 外的相邻网格点序列。对于一个固定的曲线 C , ϕ 的定义可以是任意的, 但是两个相邻的网格点序列 L_m 和 L_{out} 则是唯一的。反过来, 如果给出两个序列, 曲线的位置取

决于网格采样间隔的精度。这促使我们来分析,表示的曲线的运动与两个序列 L_{in} 和 L_{out} 之间的关系。

图 5.4(b)中给出了图 5.4(a)中曲线 C 的演化结果。在网格点 A 的位置,曲线向外运动,同时点 A 的 ϕ 值由正变为负;在网格点 B 的位置,曲线向内运动并且分裂为两条曲线,同时点 B 的 ϕ 值由负变为正。除了用于处理 PDE 的繁重的运算量,而且很难满足实时的要求外,这一切在水平集方法中可以得到很好的实现。但是,如果我们只关心曲线 C 的最终位置的话,我们使用曲线 C 与 L_{in} 和 L_{out} 之间的关系可以很容易得到相同的结果。将曲线运动到网格点 A 之外,我们仅需要把它从 L_{out} 中变换到 L_{in} 中。相同地,我们仅需要把网格点 B 从 L_{in} 中变换到 L_{out} 中使得曲线向内运动。在 L_{in} 和 L_{out} 中的所有点通过应用这些步骤,我们能够用最小的运算量在沿着曲线任何一处的网格点处将 C 向内或向外运动。这些构成了我们的快速实现算法的基础。

为了避免我们的方法与用于曲线演化的拉格朗日类型的方法之间产生混淆,我们注意到在 L_{in} 和 L_{out} 中的点不在曲线 C 上。它们用于定义离散网格上曲线的邻,而且曲线仍然通过 ϕ 来表示。

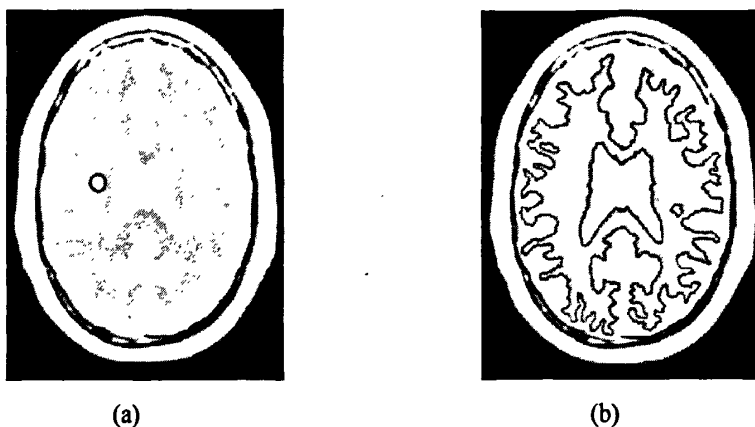


图 5.5 脑部 MRI 图像分割

图 5.5 给出了一幅脑部 MRI 图像的分割效果图。图 5.5(a)中所示为原始图像与初始轮廓(黑色圆圈),图 5.5(b)中所示为这一快速算法对这幅脑部 MRI 图像的分割结果。

5.4 小结

本章主要讨论了一种基于稀疏场方法的水平集分割算法。第一节给出了传统稀疏场的定义;第二节详细阐述了稀疏场算法,从初始化到轮廓演化,最后给出了该算法的实验结果,结果表明曲线更新的速度只取决于曲线的长度,而不是图像的大小,这种方法适宜处理医学影像这类数据量比较大的图像;第三节介绍了

一种与稀疏场算法相似的技术，将 5 个序列简化为 3 个序列，从而获取更快的分割速度。

第六章 总结与展望

6.1 本文工作总结

医学影像分割技术是医学影像领域内的重要课题之一,医学影像分割经过了长时间的发展,逐渐由手动分割向自动化及交互分割方向发展,而借助各种理论至今已提出了成千上百种各种类型的分割算法。尽管研究者在图像分割方面做了大量的研究工作,但由于尚无通用的分割理论,因此已经提出的分割算法大都是针对具体问题的,并没有一种适用于所有图像的通用的分割算法。但是可以看出,医学影像分割方法正朝着更快速、更精确的方向发展,通过与各种新理论和新技术结合将不断取得突破和创新。

本文主要研究了水平集方法在医学影像分割上的应用,这种方法提高了分割算法的鲁棒性,在提高演化计算精度的同时加速了演化过程,从而改善了图像的分割效果。

本文的主要的研究成果总结为以下几点:

(1) 讨论了当前主要的医学影像分割算法,分析了每种算法的研究现状和技术特点;

(2) 重点研究了基于水平集方法的医学影像分割算法,对水平集方法的基本理论和数值计算流程进行了专门的分析和研究,讨论了水平集方法的两种快速算法,窄带法和快速行进法。通过对这两种算法的分析和研究,总结出水平集算法的优缺点,为水平集算法的改进提供基础;

(3) 介绍了一种无需重新初始化的基于图像边缘信息的变分水平集图像分割算法。这种算法完全消除了真正影响水平集计算量的重新初始化步骤,从而加速了轮廓线的演化,提高了算法的鲁棒性,同时使得初始化方法更加灵活;

(4) 讨论了一种基于稀疏场方法的水平集分割算法。该算法试图把窄带宽度缩小到极限,减少每次迭代时更新的像(体)素,同时可以提高每次迭代中时间步长的值,这样就加快了演化过程。另外,稀疏场算法还可以提高演化计算的精度。该算法适宜处理医学影像这类数据量比较大的图像。

6.2 未来发展展望

医学影像分割是一个传统而具有挑战性的课题。结合本文的研究工作和应用的实际需求,今后的研究任务应该着眼于以下几个方面:

(1) 本文对无需重新初始化的变分水平集模型研究还不够深入。由于这个模型仅利用了图像的边缘梯度信息,而没有考虑图像的全局区域信息,使得模型对边缘模糊图像、噪声图像、纹理图像的分割效果不理想。此外,该模型的迭代次数过多,初始轮廓线的选取依然受到很大限制。这些都需要不断深入研究才能够不断优化。

(2) 对于基于稀疏场方法的水平集分割方法的研究还有很大的空间。将原有方法中的 5 个序列减少为 3 个的相似技术,可以快速实现基于稀疏场方法的水平集分割,这一方法还有待深入研究。如果能将序列数继续减少,可能会影响实验结果的精确度,但可以减少迭代次数,这是我们进一步研究的方向和趋势。

(3) 本文的研究虽然是基于二维图像,但是本文提出的方法都很容易扩展到三维图像。对三维体数据图像直接进行分割是一项很有意义的工作,未来的研究将致力于此方面的工作。

(4) 到目前为止,还没有一种适用于所有图像的分割算法,因此在进行具体的图像分割时,算法选择是很有必要的,即考虑图像分割的评价方法,这在本文中并没有过多的涉及。因此,深入研究一些客观和定量的标准来评价分割算法和分割图像的质量,以帮助合理选择和优化算法,并为研究新的算法提供指导,是非常有意义的。

致谢

论文完成之际，首先要向我的导师杨兵教授表示深深的谢意，论文的顺利完成得益于杨老师的悉心指导。杨老师渊博的学识、敏锐的洞察力、严谨的治学态度深深地影响着我，成为我在科研工作中的榜样。在两年多的学习生活中，杨老师待人谦和诚恳、风趣幽默，亦师亦友。在此，谨向两年多来培养、帮助和关怀过我的导师致以由衷的敬意和诚挚的谢意！

感谢我的父母和家人。我的父母含辛茹苦地将我养育成人，在我身上倾注了他们一生的心血，我的家人在我这么多年的学习生涯中对我一如既往的支持、理解和鼓励，使我能够坦然面对坎坷的人生和残酷的现实，他们始终是我前进的力量源泉。

感谢我的师兄袁晓光、杨大峰、徐常志、李震、朱康生、刘向涛、王伊、师姐李媛媛，在他们的帮助和指导下，使我迅速适应研究生生活，在生活、学习和科研上少走了很多弯路。

感谢我的同级黄琴波、胡玉成在我学习和生活中对我的支持和帮助。

感谢我的好朋友张涛、吴金建、陈崇雨、王敏华、安欢、吕涛、李晓博、张骅、张华伟、王琦、李杰、党杨涛、王明明、周智勋、于政、李深、王飞、陈婧、丹洋、王保刚等，感谢你们一直以来对我的支持和鼓励，与我分享快乐、分担忧愁，给予我人生中最美好的友谊。

感谢我的师弟陈俊、韦明祥、王冠宇、杨弋、师妹王媛媛、王黎明在学习生活中给予我的支持。

同时衷心地感谢参加我论文评阅的老师们，感谢在百忙之中抽出宝贵的时间评审我的论文。

衷心感谢所有关心、支持、帮助和鼓励过我的老师、同学以及亲朋好友们！感谢一路与你们同行！

参考文献

- [1] C. Tsai. Automated segmentation of brain MR images[J]. Pattern Recognition, 1995, 28(12):1825~1837.
- [2] Zheng L, Jin J, Hugues T. Unseeded region growing for 3D image sementation[J]. Journal of Research and Practice in Information Technology, 2001, 2:31~37.
- [3] M. Stella Atkins, Blair T. Machiewich. Fully Automatic segmentation of the Brain in MRI[J]. IEEE Trans Medical Imaging, 1998, 17(1):98~107.
- [4] Zi-kuan Chen, Sabee Molloi. Automatic 3D vascular tree construction in CT angiography[J]. Computerized Medical Imaging and Graphics, 2003, 27:469~479.
- [5] 叶峰. 医学图像提数据分割及其可视化的研究[D]. 苏州: 苏州大学, 2008.
- [6] 何东健, 耿楠, 张义宽. 数字图像处理[M]. 西安: 西安电子科技大学出版社, 2003.
- [7] 李弼程, 彭天强, 彭波等. 智能图像处理技术. 北京: 电子工业出版社, 2004.
- [8] 许录平. 数字图像处理. 北京: 科学出版社, 2007.
- [9] 魏伟波, 潘振宽. 图像分割方法综述. 世界科技研究与发展, 2009, 31(6): 1074~1078.
- [10] 林瑶, 田捷. 医学图像分割方法综述. 模式识别与人工智能, 2002, 15(2): 192~204.
- [11] 孟令旗. 基于水平集的医学图像分割算法研究. 上海大学硕士毕业论文, 2008, 1.
- [12] Horowitz SL et al. Picture segmentation by a directed split-and-merge procedure. Proceedings of second international joint conference on pattern recognition, 1974:424~433.
- [13] 魏伟波, 芮筱亭. 图像边缘检测方法研究[J]. 计算机工程与应用, 2006, 42(30): 88~91.
- [14] M. Kunt. Edge detection: a tutorial review. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1982, 7:1172~1175.
- [15] J. Canny. A computational approach to edge detection. IEEE Trans. PAMI, 1986, 8(6):679~698.
- [16] Lee JSJ, et al. Morphologic edge detection. IEEE J. Robotics and Automation, 1987, 3(2):142~156.
- [17] Bosworth JH, Acton ST. Morphological scale-space in image processing. Digital

- Signal Processing, 2003, 13:338~367.
- [18] Luc Vincent, Pierre Soille. Watersheds in digital spaces:an efficient algorithm based on immersion simulations[J]. IEEE Trans. On Pattern Analysis and Machine Intelligence, 1991, 13(6):583~598.
- [19] Meyer F, Beucher S. Morphological segmentation[J]. Journal of Visual Communication and Image Representation, 1990, 1:21~46.
- [20] 罗强, 任庆利, 杨万海. 基于分形理论的图像边缘提取方法[J]. 通信学报, 2001, 22(11): 104~109.
- [21] 薛东辉, 朱耀庭, 朱光喜等. 分形方法用于有噪图像边缘检测的研究. 通信学报, 1996, 17(1): 7~11.
- [22] Kass M, Witkin A, Terzopoulos D. Snake:active contour models. Pro First Inter Conf Computer Vision, London, 1987:259~268.
- [23] Xu C, Prince J L. Snakes, shape and gradient vector flow. IEEE Trans on Image Processing, 1998, 7(3):359~369.
- [24] Menet S, Saint P M, Medion G B. Snakes:implementation and application to stereo. Pittsburgh Pennsylanis:IEEE Computer Society Press, 1990:720~726.
- [25] Ghosh A, Pal NR, Pal Sk. Image segmentation using a neural network[J]. Biol Cybern, 1991, 66:151.
- [26] 陈武凡. 小波分析及其在图像处理中的应用[M]. 北京: 科学出版社, 2002.
- [27] Sethian J A. An analysis of flame propagation[D]. Ph.D. dissertation, Dept. Math., Univ. California, Berkeley, CA, 1982.
- [28] Sethian J A. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Material Science[J]. London:Cambridge University Press, 1999.
- [29] Osher S, Sethian J. Fronts Propagating with Curvature-Dependent Speed:Algorithms Based on Hamilton-Jacobi Fomulations[J]. Computational Physics, 1988, 79.
- [30] 唐立群, 郭庆昌, 李永华. 数字图像模式识别方法分析. 哈尔滨: 哈尔滨工程大学出版社, 2008.
- [31] 李俊. 基于曲线演化的图像分割方法及应用研究. 上海交通大学博士毕业论文, 2001, 6.
- [32] <http://mathworld.wolfram.com/LevelSet.html>.
- [33] Chop D. Computing minimal surface via Level Set curvature flow. Journal of Computational Physics, 1993, 106:77~91.
- [34] Sethian J A. A Review of the Theory, Algorithms, and Appliactions of Level Set

- Methods for Propagating Interfaces[J]. Acta Numerica, Cambridge University Press, 1996.
- [35] Mikkel B. Stegmann. Fast Marching Level Sets. 2001.
- [36] Sethian J A. A Fast Introduction to Fast Marching Methods. 2001.
- [37] 朱付平, 田捷, 林瑶等. 基于 Level Set 方法的医学图像分割. 软件学报, 2002, 13(9): 1866~1872.
- [38] Zhao H K, Chan T F, Merriman B et al. A variational Level set approach to multiphase motion. Journal of computational Physics 127, 1996, no0167:179~195.
- [39] J. Gomes and O. Faugeras. Reconciling distance functions and Level Sets. J. Visual Communic. and Image. Representation, 2000, 11:209~223.
- [40] Chunming Li, Chengyang Xu, Changfeng Gui et al. Level Set Evolution Without Re-initialization:A New Variational Formulation. IEEE International conference on Computer Vision and Pattern Recognition(CVPR) San Diego, 2005:430~436.
- [41] V. I. Arnold. Geometrical Methods in the Theory of Ordinary Differential Equations. New York:Springer-Verlag, 1983.
- [42] L. Evans Partial Differential Equations. Providence:American Mathematical Society, 1998.
- [43] B. Vemuri and Y. Chen. Joint image registration and segmentation. Geometric Level Set Methods in Imaging, Vision, and Graphics, Springer, 2003:251~269.
- [44] S. Osher and R. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Spring-Verlag, New York, 2002.
- [45] 李萌. 基于偏微分方程的图像分割算法研究与应用. 南京理工大学硕士毕业论文, 2006, 1.
- [46] 常先堂. 基于 PDE 的变分水平集图像分割方法. 大连理工大学硕士毕业论文, 2006, 6.
- [47] Whitaker T. and Breen D.. Level-Set Models for the Deformation of Solid Objects, Proc of 3th Intl. Workshop on Implicit Surfaces, 1999:19~35.
- [48] Whitaker R, Breen D, Museth K. A level-set approach to 3D reconstruction from range data. International Journal of Computer Vision, 1998, 29(3):203~231.
- [49] 翁建广, 庄越挺, 潘云鹤. 基于改进稀疏场算法的水平集形状过渡. 软件学报, 2006, 17(7): 1544~1552.
- [50] 翁建广. 基于体模型的三维形状检索和形变研究. 浙江大学博士研究生毕业论文, 2005, 6.
- [51] Shawn Lankton. Sparse Field Methods-Technical Report. 2009, 7.
- [52] A. Blake and M. Isard. Active Contours. Springer, Cambridge, 1998.

- [53] T. Chan and L. Vese. Active contours without edges. IEEE Transaction on Image Processing, 2001, 10(2):266~277.
- [54] Y. Shi and W. Karl. A fast Level set method without solving pdes. In Proceedings of ICASSP, volume2, 2005.

研究成果

在硕士研究生期间取得的研究成果如下：

一、参加科研项目情况：

[1] 项目名称：大空间火灾探测系统。

参与工作：负责系统方案可行性论证、设备选型、部分硬件电路设计。

[2] 项目名称：西安华海医疗信息技术股份有限公司三维影像工作站 4.x 版本。

参与工作：负责相关算法的课题研究。

二、发表的论文情况：

[1] 郭元卡. 基于 Level Set 方法的图像分割. 电子科技, 2011, 24(5).

