哈爾濱工業大學

毕业设计（论文）

题　目　一种 CT 图像中骨骼边缘检测算法的研究与实现

专　　业　信息安全

学　　号　090420131

学　　生　姚　明

指 导 教 师　董开坤　副教授

答 辩 日 期　2013 年 6 月 23 日

哈尔滨工业大学

摘　要

随着计算机技术的不断发展，它已经渗透到了各个行业内部，其中包括医疗行业。就医疗方面，CT 图像中骨骼的自动分割是一个医疗自动化的重点和难点。

骨骼的法线方向在 CT 图像中的骨骼分割中起到了很重要的作用。但是，由于骨骼在不同的 CT 图像中在大小、像素强度等方面都有很大变化以及噪声的影响，所以想用单一的一种方法来计算 CT 图像中的骨骼法线方向是很难并且很不精确的。所以，我们采用了一种“先估算，后纠正”的方法来计算法线方向。

首先，使用高斯一阶导来计算法线方向，然后通过常见的轮廓跟踪算法找出大致骨骼边缘并通过几何的方法得出另一个法线方向，之后通过一个与高斯一阶导的可信度相关的权重值来将两种方法所取得的法线方向相结合，得到一种更为精确的法线方向。其中，权重值的确定跟高斯一阶导中的噪声方差相关联，这样，即确定了较好的方差值，又选择了一个好的权重值，从而得到好的法线方向。

最后，在每一个近似边缘点的法线方向上建立一维信号，然后利用常见的边缘检测算子来检测、更正边缘点。

经实验，这种方法在增强了轮廓跟踪算法的抗噪性，也提高了其准确率，并且在一定的噪声下也可以正常工作。但是，该算法也有并且必须手动指定种子点等缺陷有待提高。

关键词：轮廓跟踪；图像分割；法线方向；高斯函数；种子点

Abstract

With the development of computer technology, it has been penetrated into all kinds of fields, including medical. And in medical automation, segmentation of bones in CT images is very important and difficult.

The normal direction of the bone contour in CT images provides important information that can guide segmentation algorithms. Since the differences of sizes of bones and the intensity values between different images and the influence of noise, however, a single way to estimate the normal direction of bones can be very difficult and the result can be in inaccurate. So, we propose a "first estimation, then correction" way to get the normal direction of bones.

First, we use the first derivative of Gaussian to estimate the normal direction. Then, we find the approximate bones contour through the common way of contour tracing and also get the normal direction with geometrical ways. Next, combine the two different results of normal directions through a weight related to the reliability of the first way to balance these to estimates and get a better result. The weight is relevant to the variance of the noise. In this way, we can get the best variance and a good weight at the same time, which are both important to make the normal direction more precise.

Finally, construct a 1-D signal along the normal direction at every approximate edge point and detect the edge point or correct the edge point.

The experimental results show that this approach does better in anti-noise than contour-tracing algorithm and improves its rate of accuracy, but there also are some disadvantages. For example, its efficiency is low and we have to offer the seed points manually.

Key words： contour trace， image segmentation， normal direction， Gaussian， seed

目 录

.

第1章 绪论

1.1 课题来源及研究的目的和意义

对于当代的医学来说，计算机已经是一个必不可少的帮手。在患者进行身体检查时，CT 等各种各样先进的仪器已经是不可或缺的了。但是真的进入到病症鉴定以及手术等大型治疗手法的准备阶段，在中国，计算机帮助医生完成的工作还是微乎其微的。随着这几年计算机视觉的发展，进行机器诊断或者机器手术准备已经不是不可能的了。在很多发达国家，计算机都参与了病症的诊断甚至手术。

在医学领域，分割 CT 图像是引导手术中非常重要的一个部分，特别是在现代的整形外科手术中，骨头和软组织的分割无论是在手术前规划阶段和手术当中都是非常重要的。所以在 CT 图像的骨骼的分割方面如果能取得一定的进展，能够在帮助医生进行病症诊断，十分有实用意义。

从致密程度来看，大致可将骨骼分为皮质骨和松质骨，其中皮质骨比较致密，所以在分割的时候很容易与临近的软组织区分开；而松质骨却十分疏松，特别是在放射影像中看，松质骨十分接近于邻近的软组织。对于那些年纪偏大导致骨质疏松的患者来说，这更是一个难上加难的问题。

由上可见，CT 图像中骨骼的分割是近几年计算机处理医学图像中的一个重点，也是难点。如果能在此领域有一定的突破，对于医学上的帮助必将很大。

1.2 国内外在该方向的研究现状及分析

1.2.1 阈值法

当今在国内外都有很多比较成熟的 CT 图像的骨骼分割算法[1-6]。这些算法大部分都是基于分割物体是同质的这样一个假设，然后利用全局[7]或者局部[8]的阈值的方法来进行分割。然而，这种假设在脂肪和肌肉中一般都是成立的，骨骼却不然。这是因为各个部位的骨骼的形状和内部结构都各不相同，导致了图像中骨骼的像素强度变化很大，它既有跟脂肪一样低的像素点，也有能够达到图像中像素强度最高的点。正是因为这样，骨骼的像素强

度和肌肉与脂肪的像素强度经常会有部分的重叠，很难利用强度阈值的方法来从图像中分隔出骨骼。

此外，抛开上面的问题不谈，对于各种的阈值法进行分割，很重要的一个问题就是确定阈值。然而，由于噪声等各种外界因素的影响，确定阈值这一步就相当麻烦。对于全局阈值，方法比较简单，但是经常会导致局部出现阈值完全不符合导致分割失败的情况；对于局部阈值，确定方法就比较复杂了，感觉吃力不讨好。所以，主要的阈值法的实用价值都偏低。

1.2.2 基于区域的分割法

基于区域的图像分割方法[2]是根据在一个子区域中各像素的某些特性的一致性来判断目标的边界，这些特性如灰度、颜色和纹理。常利用图像多分辨率的表达结构，如金字塔结构。基于区域的分割方法常见的有两种形式，一种是从单个像素出发，逐渐合并从而形成目标分割区域，称为区域生长法；另一种是从全图出发，逐渐分裂切割直至得到目标分割区域，称为分裂合并法。

这算是一种基础的分割方法，其结果比较中规中矩，但是计算量又很大，所以，在医学图像领域，很少直接使用此种方法。但是，不可否认的是，像下面的动态轮廓模型等很多最新的方法都是以区域分割为基础的。

1.2.3 动态轮廓模型

另外一种现在比较流行的方法就是动态轮廓模型的方法[2]。这种方法已经被证明了在医学图像处理中是十分有用的[9-10]。

动态轮廓是一种保留连通性的逐次近似法，适用于图像分割问题。自从 Kass 第一次提出 snakes 模型，动态轮廓方法就开始被应用在图像分割和边界跟踪方面。动态轮廓的基本思想是通过一个初始的闭合曲线，也称为轮廓，根据图像中的某项参数迭代的进行收缩或是扩张操作。这些收缩和扩张操作被称为轮廓演化，演化的动力来自一个能量函数最小化过程。该能量函数类似传统的基于区域的分割方法，或是由几何模型的偏微分方程来拟合。

这种方法成功的关键在于初始的轮廓[11]，初始轮廓既可以人工提供也可以自动生成。如果初始轮廓能够接近真的骨骼的轮廓，那么动态轮廓一般就可以靠近真正的轮廓。然而，想要自动生成这样的初始轮廓是十分困难的。所以，这种方法还是不能很好的实现全自动的分割。

1.2.4 边缘过滤

一般的边缘检测或者说是边缘过滤也是一套常见的 CT 图像骨骼分割方法。这种方法就是利用了在图像中边缘的像素强度的变化要快于非边缘区域。可是这种方法中涉及到很多的可用参数[12]，这就使得选择这些参数的最优值变得十分困难，从而导致了生成的骨骼轮廓是非闭合的并且很多边缘点都是错误的。

此外，在骨骼的边缘检测算法中，理想的结果当然是只获得骨骼的边缘，但是这种算法经常会把肌肉或者脂肪的边缘轮廓也都包含进来，不符合要求。

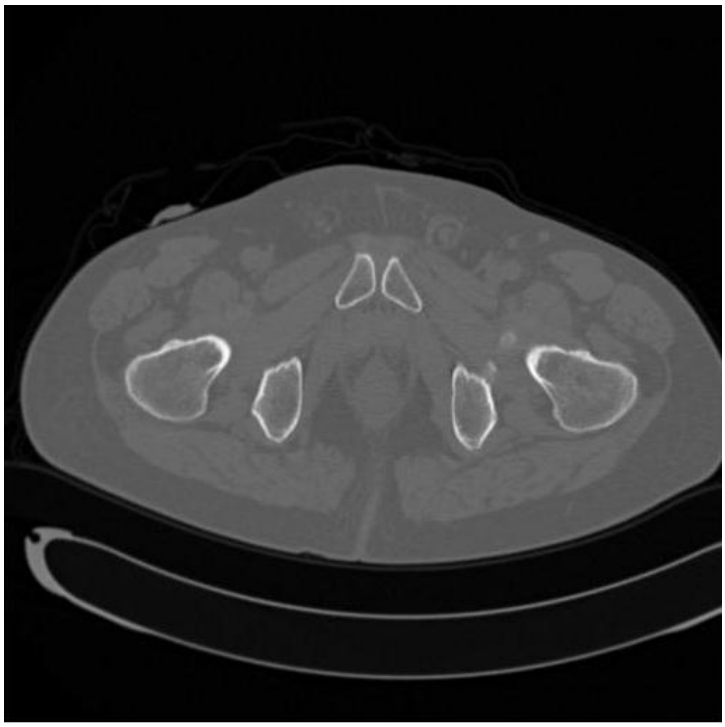


图 1-1 原 CT 图像显示

图 1-2 是边缘过滤算子中的一种——Canny 算子的结果显示[13]，图 1-1 是原图像，图 1-2 是 Canny 算子边缘检测的结果。在两图对比中可以清晰的看出上面所提到的这种方法所的缺点所在，即该方法找出来原图像中所有的轮廓，包括各种组织，而不是只有骨骼的轮廓。此外，可以清晰的从结果图中看出，该方法将原图像中所有的噪声也都显示了出来，这显然是不符合我们要求的。

图 1-2 Canny 算子的结果显示

在最后的结果实验中，我们会将 Canny 算子的边缘检测结果和我们算法的结果再次进行比较。

1.2.5 轮廓跟踪

轮廓跟踪是一种二值图像中很常用的一种边缘检测算法，顾名思义，轮廓跟踪就是通过顺序找出边缘点来跟踪边界的。经过一些修改,轮廓跟踪算法也可以适用于弧度图像。由于轮廓跟踪的基本结束条件就是跟踪到的下一个点是已有的边缘点，这就保证了轮廓跟踪算法所产生的结果一定是闭合的完整曲线，这无疑是一个很大的有点。也正是因为这样，我们才在算法中选用轮廓跟踪算法为基础。

但是，轮廓跟踪算法也有很多缺点和不足。

首先，轮廓跟踪算法是从种子点开始跟踪的，也就是其需要指定种子点。当然，现如今的技术中很多种子点都有可能实现自动获取，不过自动获取的种子点很难分辨是否属于我们想要的组织。所以，想要获得最理想的单纯指定的骨骼边缘轮廓，很多情况下还是需要手动指定种子点，没有实现完

全的自动。

其次，由于轮廓跟踪中对于跟踪的利用，导致轮廓跟踪算法有几率会产生重大错误。这个很好理解，毕竟该算法是跟着上一个点走的，如果有噪声影响了这个点的产生，使其偏离了真正的边缘点，那么其产生的下一个点只会偏离的越来越多，最终导致出现错误。这也是轮廓跟踪的一个致命伤。

综上所述，现在在 CT 图像中骨骼边缘检测这个领域中，有着各式各样的算法，它们都各有优点，但是也都有不足。我所希望做的就是取长补短，对于原有算法可以有一些改进。

1.3 主要研究内容

从上面提到的现今国内外已有的各种各样的骨骼分割技术不难看出，其实在这个领域已经有了一些比较成熟的算法，但是这些算法多多少少都是有一定的问题存在。所以选择一种较为成熟的算法，然后将其进行改进，有可能可以得出比较令人满意的结果。

如今，有很多同样在做这方面算法的研究者的一些想法和算法，其中包括边缘跟踪算法。边缘跟踪算法是一种可以用来行程闭合的轮廓的算法。如果假设边缘点 i+1 的位置在 x_{i+1}，那么从第 i 个点出发的话，那么第 i+1 个点的位置就可以被表示成：

$$x_{i+1} = f(x_i) + \varepsilon_{i+1} \tag{1-1}$$

式中 $f(x_i)$ ——从 x_i 来预测 x_{i+1} 的函数；

ε_{i+1} —— $f(x_i)$ 计算过程中的误差值。

使用这种方法，只要可以找到合适的 ε_{i+1}，就可以从一个边缘点而不断的得到新的边缘点直到回到初始的边缘点或者某些条件不再满足。

减小这种算法中预测的误差，也就是寻找合适的 ε_{i+1} 的值，可以使用高斯一阶导、Canny 算子等边缘检测器[14-16]。这些边缘检测方法都是用于一维信号的，对于二维的图像来说，由于沿着法线方向像素变化最快，所以最好

的办法就是这些一维信号应该由那些处于预测点法线方向上的像素点组成。因此，很关键的一步就是怎样精确的找到预测点的法线方向。

1.4 本文结构

本文主要内容如下：

第 2 章中，介绍了本算法用到的一些重要的相关基础知识；

第 3 章中，介绍本算法的整体思想以及具体的实现；

第 4 章中，介绍算法的实验结果，以及与一些现有算法的结果的比较。

第2章 基础理论及相关技术介绍

2.1 数字图像及其分类

数字图像，是二维图像用有限多个的数字数值像素的表示。数字图像是由模拟图像数字化得到的，它是以像素为基本元素的，并且可以用数字计算机或数字电路存储和处理的图像。简单的说，可以使用计算机处理的图像都可以叫做数字图像。

每个图像的像素通常对应于二维空间中某一个特定的位置，并且有若干个与那该点相关的采样值组成数值。根据这些采样的数目及特性的不同，数字图像可以分为一下几种。

二值图像：每个像素的取值仅可以为 0 或 1 的图像。

灰度图像：图像中每个像素的取值范围是 0-255。0-255 之间表示不同的灰度级，其中 0 为最黑，255 为最白。这就是我们这次算法中主要接触的图像类型，读入图像后，图像矩阵中保存的是对应点的灰度值。

彩色图像：彩色图像是由三幅不同颜色的灰度图，即红绿蓝，组合而成。

立体图像：立体图像是对相同物体采用不同角度拍摄的一组图像，立体图像往往可以透露出图像的深层信息。

三维图像：三维图像是由一组堆栈的二维图像组成，每一幅二维图像表示该物体的一个横截面。

数字图像也用于表示在一个三维空间分布点的数据，其实 CT 图像就是这样的，但是对于边缘检测来说，我们不需要这么多的信息，只需要将其当作灰度图像去处理即可。

2.2 CT 图像的存储标准

CT 图像是由一定数目由黑到白不同灰度的象素按矩阵排列所构成。这些象素反映的是相应体素的 X 线吸收系数。不同 CT 装置所得图像的象素大小及数目不同。大小可能是是 $1.0\times1.0mm$，$0.5\times0.5mm$ 不等；数目可以是 256×256，即 65536 个，或 512×512，即 262144 个不等。显然，象素越小，数目越多，构成图像越细致，即空间分辨力高。

CT 图像是以不同的灰度来表示，反映器官和组织对 X 线的吸收程度。其中，黑影表示低吸收区，即低密度区，如肺部；白影表示高吸收区，即高密度区，如骨骼。但其实，正如我们上面所提到的，CT 图像中骨骼点的灰度变化范围十分广泛，这也就导致了我们很难使用简单的阈值法来处理 CT 图像，进行边缘检测。

在计算机中，CT 图像常常以 DICOM 标准进行存储，而 DICOM 即数字影像和通信标准。在医学影像信息学的发展和研究的过程中，由于医疗设备生产厂商的不同，造成与各种设备有关的医学图像存储格式以及传输方式各不相同，使得医学影像及其相关信息在不同系统、不同应用之间的交换变得十分繁琐。因此，DICOM 标准就是在这样的产生了。

DICOM 标准中涵盖了医学数字图像的采集、归档、通信、显示及查询等各方面的信息存储协议；以开放互联的架构和面向对象的方法定义了一整套包含各种信息的对象集合，其中包含了在医学图像处理的各个阶段所要用到一切信息。

当然，对于我们来说，只是对图像信息处理，进行边缘检测，在这过程中只需要读入矩阵的图像灰度值即可，DICOM 标准中的其他很多信息我们可以不需要考虑。在程序中处理 CT 图像，只需要将 DICOM 中的信息读入一个矩阵当中，然后对矩阵中的灰度值进行下一步处理即可。

2.3 图像噪声及其分类

图像噪声就是指图像中各种妨碍人眼对其信息接受的因素。噪声在理论上常常被定义为“不可预测，只能用概率统计方法来认识的随机误差”。因此我们可以将图像噪声看成是多维随机过程，继而我们就可以使用随机过程来描述噪声，即用其概率分布函数和概率密度分布函数。

目前大多数数字图像系统中，无论对输入图像如何操作，图像噪声也将同样受到这样的操作过程。在这些过程中的各种操作使得图像噪声的精确分析变得十分复杂。另一方面图像只是传输视觉信息的一种媒介，对图像信息的认识理解完全是由人的视觉系统所决定的。不同的图像噪声，人的感觉程度是不同的，这就是所谓人的噪声视觉特性课题。

图像噪声在数字图像处理领域中的重要性越来越明显，因为在任何情况下，我们都不能忽略图像噪声，否则可能造成重大错误。

在我们的算法中，也正是因为噪声的影响导致了很多时候不能得到理想

的结果，并且在生成边缘的过程中还会因为噪声的影响而导致出现边缘不闭合、出现无用点等各种各样的问题，这些问题都需要用一些手段去处理，从而将噪声对图像的影响降到最低。

需要强调的一点是，噪声在 CT 图像中是必然存在的，所以不可以忽视他，那样只会带来更大的误差甚至错误。

2.3.1 高斯噪声

第一种就是我们要用到的高斯噪声。也就是每一点都存在噪声，但噪声的幅值是随机分布，概率密度为高斯，其中最常见、最典型的高斯噪声是加性高斯白噪声。

加性高斯白噪声是一种最基本的噪声与干扰模型。我们此后再对噪声处理的时候用到的就是这种噪声，所以在此先进行介绍。

加性高斯白噪声需要分为 3 个部分去理解。首先，是加性噪声，它指的是叠加在信号上的一种噪声，通常记为 n(t)，而且无论有无信号，噪声都是始终存在的。因此通常称它为加性噪声。其次是白噪声，它是指噪声的功率谱密度在所有的频率上均为一相同的常数，则称这样的噪声为白噪声。最后，如果白噪声取值的概率分布服从高斯分布，则称这样的噪声为高斯白噪声。

该噪声信号为一种便于分析的理想噪声信号，实际的噪声信号往往只在某一频段内可以用高斯白噪声的特性来进行近似处理。由于加性高斯白噪声信号易于分析、近似，因此在数字图象处理、通信等领域运用十分广泛。也正是因为其易于处理、近似的特点，在本算法中也采用了加性高斯噪声来模拟噪声信号。

此外，本文算法中之所以采用加性高斯噪声来进行模拟，还有一个主要原因是，在这种模拟情况下，很容易由高斯一阶导求得图像的法线方向，这和我们的主要目的相同。

2.3.2 椒盐噪声

第二种常见的图像噪声是椒盐噪声。这种噪声的幅度基本相同，但噪声出现的位置是随机的。

椒盐噪声是由图像传感器，传输信道，解码处理等产生的黑白相间的亮暗点噪声。椒盐噪声往往由图像切割引起。

椒盐噪声又被人们分为椒噪声和盐噪声两种，分别对应于低灰度噪声和高灰度噪声。

2.4 数字图像噪声处理

正如上文所述，在数字图像处理领域，噪声一直是主要需要解决的问题之一。所以，在详细了解我们的算法实现之前，有必要了解一下常见的两种噪声处理方法。

2.4.1 均值滤波去噪

均值滤波是典型的线性滤波算法，它是指在图像上对目标像素给一个模板，该模板包括了其周围的临近像素，即其 8-邻域中出去本身像素以外其他的 8 个点，再用模板中的全体像素的平均值来代替原来像素值。这种方法也就是所谓的邻域平均法。

邻域平均法有力地抑制了噪声，但随着领域的增大，图像的模糊程度也愈加严重。为了尽可能地减少模糊失真，也可采用阈值法减少由于领域平均而产生的模糊效应。我们下文中在轮廓跟踪中会提到一种“跟踪虫”算法，其基本思想其实也利用了邻域平均这种均值滤波法。

除了邻域平均这种算数滤波器，均值滤波还包括几何均值滤波器、谐波均值滤波器和逆谐波均值滤波器。几何均值滤波器所达到的平滑度可以与算术均值滤波器相比，但在滤波过程中丢失的图像细节要少于算数滤波器。谐波均值滤波器对椒盐噪声中的盐噪声效果较好，也比较适用于高斯噪声。逆谐波均值滤波器效果不错，但是它有一些参数必须赋值正确，否则可能出现重大错误。

2.4.2 中值滤波去噪

因为图像中噪声的出现，使该点像素比周围的像素亮（或暗）了许多。如果在某个模板中，对像素进行由小到大排列的重新排列，那么最亮的或者是最暗的点一定被排在两侧。取模板中排在中间位置上的像素的灰度值替代待处理像素的值，即将每一像素点的灰度值设置为该点某邻域窗口内的所有像素点灰度值的中值，就可以达到滤除噪声的目的，这就是中值滤波去噪的主要思想。

中值滤波对于滤除图像的椒盐噪声非常有效。中值滤波器可以做到既去除噪声又能保护图像的边缘，从而获得较满意的复原效果。但是，由于本文中采用高斯噪声来模拟图像中噪声，所以文章中也没有涉及到相关中值滤波去噪的实际应用，所以此处不再做详细的介绍。

2.5 数字图像中的距离和邻域

2.5.1 像素间的距离

如上文所说，一幅数字图像由有限数量的像素组成，像素反映图像特定位置处的亮度信息。通常像素按照矩形采样栅格分布。我们用 2 维矩阵来读入并表示这样的数字图像，矩阵的元素是正整数，对应于亮度范围的量化级别。

对于一般所熟悉的连续图像，它上面有一些特性在数字图像中没有得到应用。例如距离，坐标 (i, j) 和 (h,k) 两点之间的距离可以有好多种表示形式，其中最常见的欧氏距离 D_E 如式（2-1）：

$$D_E[(i, j),(h,k)] = \sqrt{(i-h)^2 + (j-k)^2} \tag{2-1}$$

这样的距离表示很直观，但是如果放在数字图像中并不合适，因为其结果很有可能不是整数。

两点之间的距离也可以表示为在数字栅格上从起点移动到终点所需要的最少基本步数。如果只允许横向和纵向移动，那么距离就是 D_4，即：

$$D_4[(i, j),(h,k)] = |i-h| + |j-k| \tag{2-2}$$

在数字栅格中如果允许沿着对角线方向移动，就可以的到距离 D_8，即：

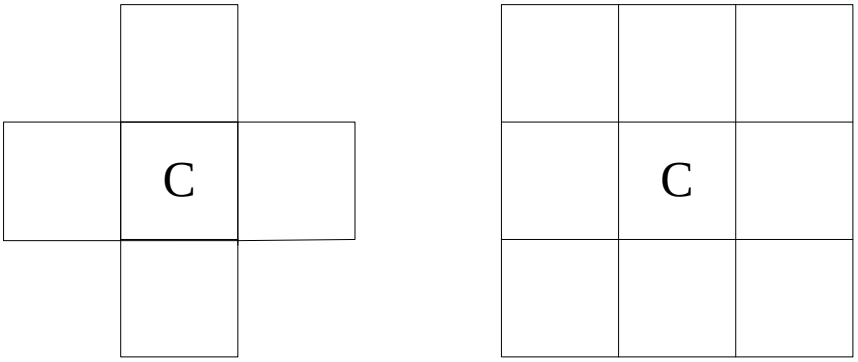
$$D_8[(i, j),(h,k)] = \max\{|i-h| + |j-k|\} \tag{2-3}$$

不同的距离定义有着不同的使用意义，在之后邻域的介绍中就要用到

D_4 和 D_8 两种距离。

2.5.2 像素的邻接性

像素的邻接性也是一个重要的概念[17]。任意两个像素如果它们之间的距离 $D_4=1$，则称彼此是 4-邻接的。相似，8-邻接指的是两个像素间的距离 $D_8=1$。两种邻接性质如图 2-1 所示：



a) 点 C 的 4-邻域　　b) 点 C 的 8-邻域

图 2-1 像素邻接性

由一些彼此邻接的像素组成的重要集合，叫做邻域。比如，图 2-1 中 b)图除了点 C 都是和点 C 是 8-邻接的点，则这些点的集合叫做点 C 的 8-邻域。相同的，a)图中就是点 C 的 4-邻域。邻域的概念在边缘检测中十分常见，也是轮廓跟踪算法的一个重要基础。

2.6 本章小结

本章介绍了一些与本算法相关的预备知识。

首先介绍了数字图像的定义及其分类。

其次 CT 图像的相关特点以及在计算机中 CT 图像的标准存储协议 DICOM 协议的简单介绍。

然后介绍了图像噪声的相关概念并简单介绍了高斯和椒盐两种常见的噪声。

在这之后，还介绍了均值滤波和中值滤波两种常见的去噪方法。

最后介绍了数字图像中的距离和邻域的概念，为文章之后对这些概念的使用做好基础。

第3章 图像中骨骼边缘检测算法的实现

3.1 算法简介

本算法是基于上文所提到的轮廓跟踪算法的，其大致的流程如图 2-1 所示。

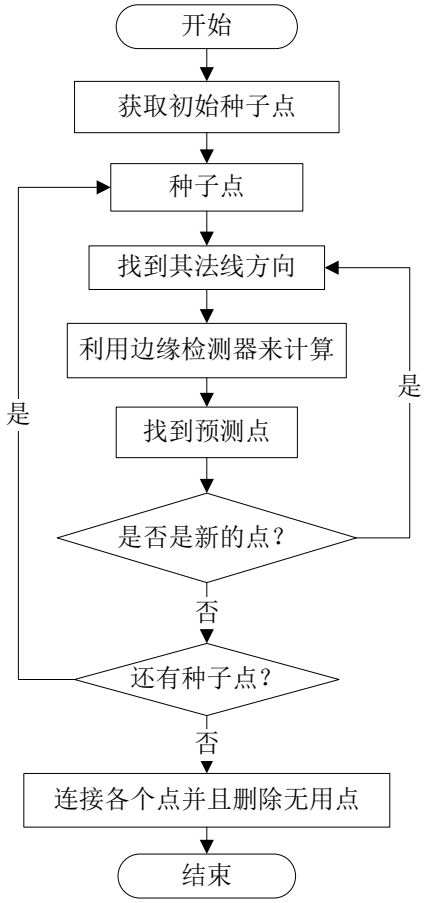


图 3-1 总体流程图

假设已经存在了一些手动标记的或者利用阈值法等方法自动获得的种子点，这些种子点没必要一定要精确的落在骨骼的轮廓上，只要它靠近骨骼轮廓即可。这个算法从一个种子点出发，计算每一个种子点的法线方向，然后用构建一组包含当前种子点法线方向上的像素点的像素强度的一维信号，然后利用一些常见的边缘检测算子或者边缘过滤器就可以寻找到这个种子点附

近的边缘点或者纠正这个种子点所在的位置。在这个一维信号中边缘点的偏差就可以被当作二维图像中以种子点为中心的边缘点的位置。从这个得到的边缘点起，可以继续预测下一个边缘点，直到预测到已有的点；在这之后，选择另一个种子点，然后重复上述过程，直到所有种子点都用完了，就可以找到图片完整的边缘了。

当然，由于这种一维信号的作用，使得图像中可能会出现一些散落的杂点，这些杂乱的点很明显是由于噪声造成的，其实算得上是对单纯的轮廓跟踪算法的纠正。所以，需要有一个衡量的标准来判断怎样的点才算得上是杂乱的点。在这种标准确立了之后，就可以将被判定为是杂点的点全都删除掉。

在理想情况下，我们得到的边缘应该是闭合的，而且其实单纯的轮廓跟踪算法本身就是保证轮廓闭合的。但是，由于图片中噪声的存在，再加上该算法在轮廓跟踪之后又使用了其他方法来纠正原有的结果，所以最终边缘的轮廓很有可能是非闭合的，这显然不符合要求。所以，在这之后应该采取一些其他的方法来对结果进行修正，保证其闭合。例如，Canny 的滞后阈值处理就是解决轮廓非闭合的一种常用方法[8]。在这种方法中，采用了两个独立的阈值，一大一小，那些像素强度在这两个阈值之间的点，只有存在于另一个边缘点旁边的时候才会被确定为边缘点。但是在本文的算法中，并没有使用这种算法，而是更加便利的采取了二次边缘跟踪的方法。在那些非闭合的轮廓中，只需要找到轮廓的终点，然后再次利用边缘跟踪的方法，直到到达轮廓的另一个终点，这样就可以很顺利的实现了最终结果的闭合性。而且这种方法也比较简单。

在确定了轮廓的闭合之后，还有一些轮廓可能会出现一些不规则的分支以及类似上面提到的杂点。这些无用点只需要经过简单的处理就可以分辨的出来，然后从轮廓中删除掉，不会影响最终结果。

由于噪音的作用以及不同图片之间的像素强度跨度较大，所以精确的边缘法线方向是很难得到的，这种事边缘跟踪算法为何鲁棒性这么差的原因。本文当中的算法正是希望可以通过一些简单的研究和尝试来改变边缘跟踪算法的这个缺陷而采取了“先估算，后纠正”的一个方案，等于是将两种方法获得的法线方向通过一些手段而综合起来，以求综合这两种方法的优点，极大可能的减小法线方向的误差，从而能够获得比单纯的边缘跟踪的算法好一些的图像边缘。

接下来，在本章的第 2 部分，主要介绍了关于在灰度图像中使用轮廓跟

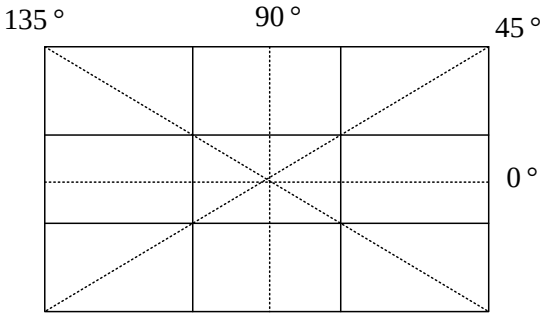
踪的方法；而如果由轮廓跟踪算法的结果来获取边缘的法线方向在第 3 部分中描述；第 4 部分介绍了使用高斯一阶导来获取图片的法线方向的方法；第 5 部分则是说明了如何利用轮廓跟踪获得的法线方向来矫正高斯一阶导获得的法线方向；至于怎样确保两种方法的均衡来获取一个最佳的法线方向值将在第 6 部分讨论；第 7 部分就是利用已经确定法线方向来构建一维信号，从而检测最终的边缘；最后我们将在第 8 部分讨论如何对最终的结果进行处理来确保无用点被删除以及边缘的闭合性。

3.2 灰度图像中的轮廓跟踪算法

轮廓跟踪，顾名思义，就是通过顺序找出边缘点来跟踪边界的。在二值图像中，轮廓跟踪是一种十分常用的获取轮廓的方法，也十分简单。但是在灰度图像中，轮廓跟踪却比较复杂，首先就涉及到有关梯度幅度的求法。

3.2.1 梯度幅度的计算

数字图像在成像、数字化和传输的过程中,由于受到各种干扰而形成噪声。而噪声和边缘是图像中像素与相邻区域的像素在灰度级上均具有跳跃特性的信号,因此图像的边缘和噪声具有很大的相似性,都可以通过图像上像素点的一阶导数来描述。在实际的应用中, 图像上每个像素点在 8-邻域内具有 8 个邻域和 0 , 45 , 90 , 135 四个检测方向，如图 3-2 所示。



a) 8-邻域方向示意图

（i-1,j-1）	（i-1,j）	（i-1,j+1）
（i,j-1）	（i,j）	（i,j+1）
（i+1,j-1）	（i+1,j）	（i+1,j+1）

b) 8-邻域相对坐标示意图

图 3-2 像素 8-邻域示意图

在 8-邻域内可以通过计算四个方向的一阶偏导数有限差分均值来确定像素点的梯度幅值[18]。设 $I(i, j)$ 表示 8-邻域内中心像素 (i, j) 的灰度值，$M(i, j)$表示 8-邻域内中心像素 (i, j) 的梯度幅值，$P_x(i, j)$表示 8-邻域内中心像素 (i, j)的四个方向的一阶偏导数。则四个方向的一阶偏导数的计算分别

如式（3-1）、式（3-2）、式（3-3）、式（3-4）所示。

$$P_{0^o}[i,j] = |I(i,j+1) - I(i,j-1)| \tag{3-1}$$

$$P_{45^o}[i,j] = |I(i-1,j+1) - I(i+1,j-1)| \tag{3-2}$$

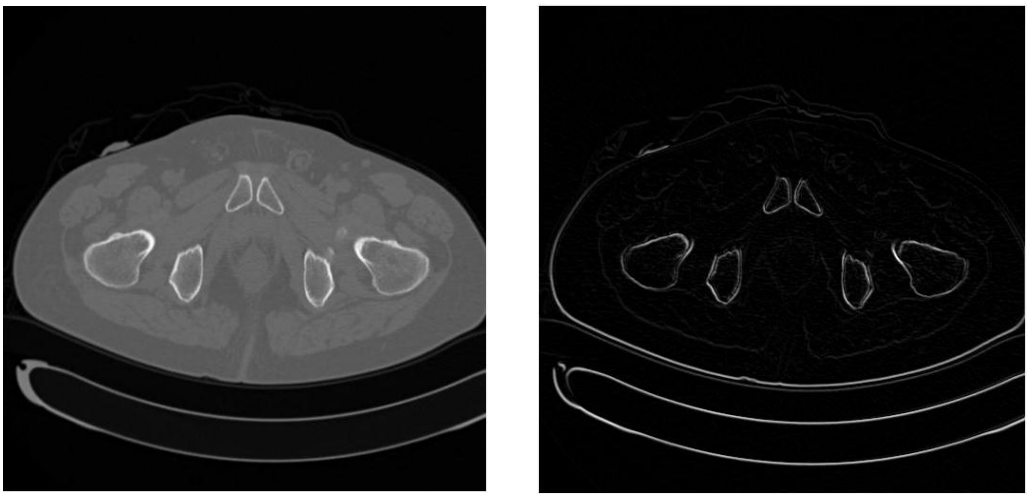
$$P_{90^o}[i,j] = |I(i+1,j) - I(i-1,j)| \tag{3-3}$$

$$P_{135^o}[i,j] = |I(i+1,j+1) - I(i-1,j-1)| \tag{3-4}$$

则中心像素（i,j）的梯度幅度为：

$$M[i,j] = (P_{0^o}[i,j] + P_{45^o}[i,j] + P_{90^o}[i,j] + P_{135^o}[i,j])/4 \tag{3-5}$$

根据式（3-5），很容易计算出灰度图像的梯度幅度值，计算效果大概如图 3-3 所示：



a) 原图像　　b) 梯度幅值图

图 3-3 原图像和梯度幅值的对比

在图 3-3 中，左边的 a)图是原图像，右边的 b)图是用上述方法所求得的梯度幅度值图。明显，梯度幅图中的边界和其他部位的灰度值相差更大，更容易分辨。边缘点基本是梯度幅度值最大的一批点。所以，计算出梯度幅度对于灰度图像的轮廓跟踪来说是十分关键的一步。

3.2.2 轮廓跟踪算法

在得出了梯度幅度值之后，就可以开始进行灰度图像的轮廓跟踪[19-20]。其大致过程如图 3-4。

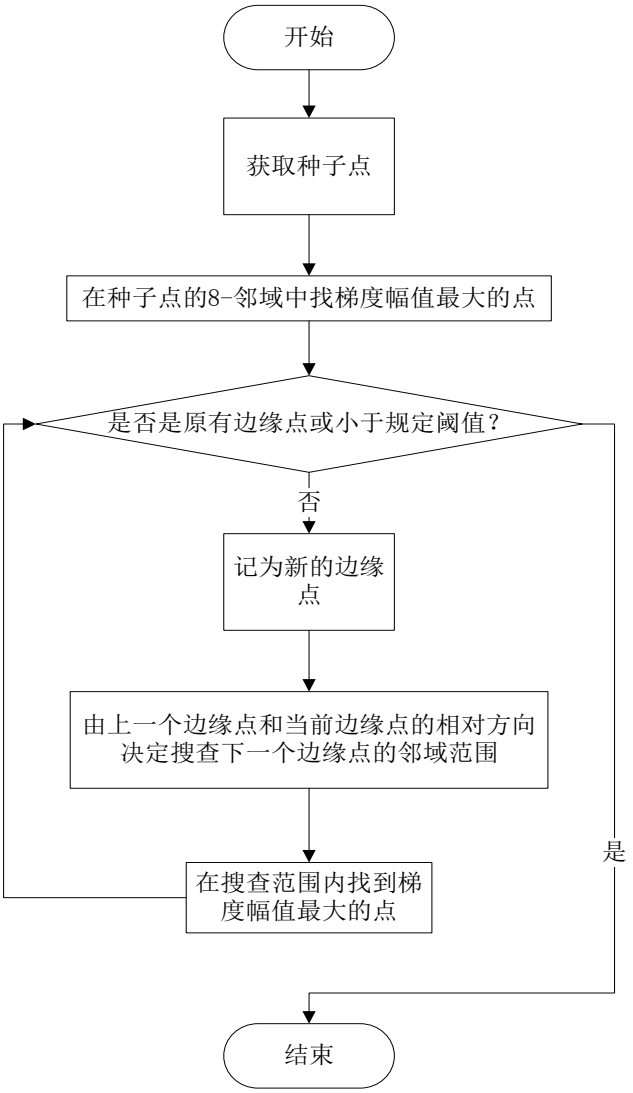
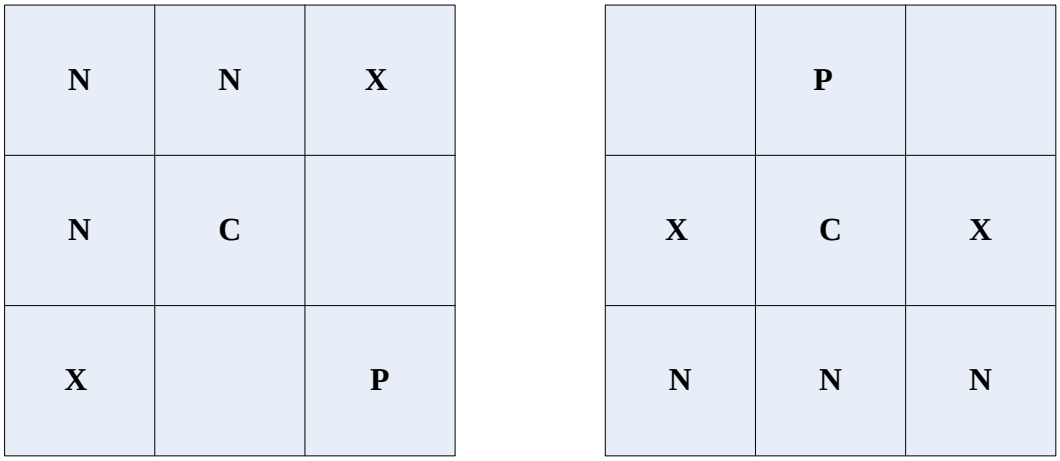


图 3-4 轮廓跟踪流程图

如图 3-4 所示，以种子点为起始点，搜索以起始点为中心的 8-邻域，找出具有最大灰度级的邻域点作为第 2 个边界点。如果有两个邻域点具有相同的最大灰度级，就任选一个。从这一点开始，就开始根据上一个边界点和当前边界点来判断下一个边界点的搜索范围，即，在以当前边界点为中心的 8-邻域内，下一个边界点的搜索范围是前一个边界点位置相对的邻点和这个

邻点两旁的两个点，如图 3-5 所示。下一个边界点就是上述三点中具有最高灰度级的那个点。如果所有三个或两个相邻边界点具有同样的最高灰度级，我们就选择中间的那个点。如果两个非邻接点具有同样的最高灰度级，我们可以任选其一。



a) 跟踪方向示例一　　b) 跟踪方向示例二

图 3-5 轮廓跟踪方向判定

在图 3-5 中，P 表示上一个轮廓点，C 表示当前轮廓点，N 就表示下一个轮廓点的搜索范围。值得一提的是，在某些极端情况下，图中的 X 点也有可能需要被列入到搜查范围中，但是这只是少数情况，绝大多数情况下还是只在 N 中搜索下一个点即可。

只要根据这样的方法不停地迭代，直到找到的下一个点的梯度幅值小于所定阈值或者下一个点已经存在与边缘点之中才停止搜索。显然，两种停止条件中的后者才是真正的找到了令人满意的闭合轮廓应遇到的情况，前者明显是由于噪声而造成的跟踪算法的中间停止。但是根据实验结果显示，跟踪算法中的阈值设置还是十分必须的，毕竟在非理想条件下，噪声必然存在，如果忽视的话，可能会导致程序走入死循环或者在根本不是边缘的地方绕了很多圈然后结束，这都是我们不想看到的结果。反之，如果程序因为阈值的关系在一些比较模糊的地方停止了，还可以根据后续的一些修复来完善它。所以，阈值的设置是十分必要的。

在轮廓跟踪结束后，正常情况下应该是行程了一段闭合的轮廓。由此轮廓就可以进行算法的后续步骤。

3.2.3 轮廓跟踪的优化

在一个理想无噪声的单调点状物图像中，上述算法将描画出最大梯度边界；但是，对于轮廓跟踪算法来说，一定数量的噪声可能使跟踪暂时或永远偏离边界，这也正是我们上面所提过的轮廓跟踪算法的最致命的缺点。噪声的影响可以通过跟踪前对梯度图像进行平滑或采用“跟踪虫”的方法来降低。即使这样，轮廓跟踪也无法保证一定可以产生理想的闭合轮廓，还是有可能出现严重的偏离。

跟踪虫是一种按下述方式工作的算法。首先，定义一个矩形平均窗 n，这就是所谓的“跟踪虫”，通常整个窗口具有相同的权值，如图 3-6 所示。最近两个或几个边界点定义了当前的边界方向。图 3-6 中显示的 3 个矩形就是候选 3 个点的跟踪虫

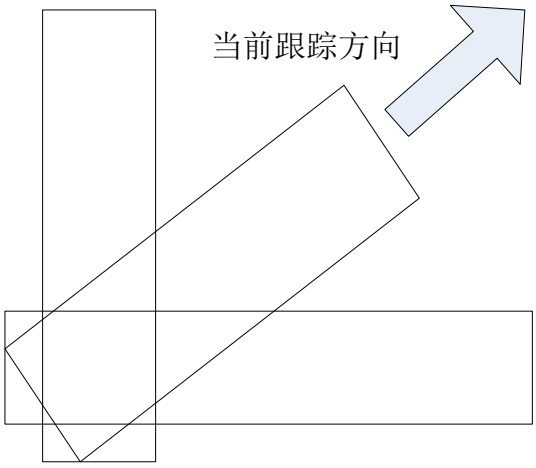


图 3-6 跟踪虫算法

虫的背部则以当前边界为中心，以当前的边界方向为轴。随后跟踪虫可向任一边转一个角。

跟踪虫所在的每一个位置，对虫覆盖的区域的平均梯度幅度进行计算。当虫位于最高平均梯度幅度位置时，可以从虫的前部选择一个点作为下一个边界点。显然，此处将比较单独的下一个边界点的梯度幅值改成了比较跟踪虫所在区域内所有点的均值，这样就实现了在一个更大的空间执行先前描述过的边界跟踪过程，大尺寸的跟踪虫可以完成梯度图像的平滑，从而降低了它对噪声的敏感化。它也限制了边界方向的急剧改变。

虫的大小和形状也可以改变以求达到最佳性能。可通过减小旁视角增大

虫的受到上一个点的影响的作用。实际上，虫的确切形状对其性能影响似乎并不明显。梯度跟踪虫通常在噪声很低的图像或人工干预能防止灾难性的偏差的情况下才有用。

在我们的算法中，简单测试了跟踪虫来对轮廓跟踪进行干涉，最终发现效果并不明显，并且还降低了计算速度，反而弊大于利，所以最后并没有采纳这种方法。但是这种方法的确是对轮廓跟踪优化的主要方法之一。

3.3 由边缘来获取边缘点法线方向

上述的轮廓跟踪算法生成了初步的二值边缘图像，为了完成整个算法，需要将由这个轮廓跟踪的结果二值图像来获取每一个边缘点的法线方向。

在此之前，我们必须首先假设骨骼的边缘是平滑并且连续的，只有这样才可以从上面方法预测的边缘点中得出边缘点的法线方向。

对于每一个边缘点，要想获得其法线方向，需要将其跟其 8-邻域内的另外两个边缘点放在一起，然后过这 3 个点做圆弧。然后这 3 点所做圆弧的圆心和中间点也就是当前求法线点的连线即为改点法线。

具体做圆弧的方法很简单，只要 3 点中任意两点连线，然后分别做其中垂线，几条中垂线的交点即为圆弧的圆心。特殊情况下，若中垂线没有交点，说明中垂线平行，即这 3 个点三点一线，这种情况下，过当前点做与三点连成线的垂线即为当前点法线。

找到法线后，需要判断其方向。有很多方法都可以实现，例如区域标识等[21]。但是由于考虑到 CT 图像的骨骼中，凹曲线并不常见，所以没必要采取复杂的方法来标识区域，而是采用的是简单的根据切线两边的边缘点个数来判断法线方向的方法，即当前点的法线必定是由切线的边缘点较多的一次指向边缘点较少的一侧。

法线的方向一旦确定，这部分的算法就基本完成了，只需要将法线方向转换成和坐标轴 X 轴的夹角即可。

3.4 由高斯一阶导来估算法线方向

法线方向的估算是基于图像边缘附近的点的像素强度沿法线方向的变化要远快于沿切线方向的变化这一事实的。在 CT 图像中，噪声是不可避免的，而噪声很有可能会模糊甚至掩盖掉边缘。因此，想要更好的估算边缘的

法线方向，就必须减少噪声的影响。若将图像中的噪声大概认定为加性高斯噪声[22]，那么式（3-6）中的G_n，即二维对称高斯函数的一阶导数，是在获取法线方向和抑制噪声方面运用的相当广泛的一种方法。

$$G_n = \frac{\partial G}{\partial n} = n \cdot \nabla G \tag{3-6}$$

$$G(x) = \exp(-\frac{\| x \|^2}{2\sigma^2}) \tag{3-7}$$

式中 $x \in R^2$；

σ——噪声方差的估算值。

那么点x_0上的法线方向n可以被估算为：

$$n(x_0) = \frac{\nabla(G(x - x_0) * I)}{\| \nabla(G(x - x_0) * I) \|} \tag{3-8}$$

式中 *——卷积；

I——图像矩阵。

要用式（3-8）在用来估算法线方向，必须保证以下两点假设成立。

首先，σ的值必须可以通过某种方法确定。由于图像中存在灰度阶跃变化和加性高斯噪声，噪声的方差σ就可以使用图像的自相关函数来估算[23]。但是，若真的在图像中使用这种方法，结果必将不尽如人意，因为自相关函数十分不精确。所以，该算法中通过大量不同的σ值进行实验，从而找到特定的某一幅图像的最佳σ。此外，在一些图像引导的手术中，需要用到上百幅的 CT 图像，所以如果真的想要投入到实际运用中去，那么σ的最佳值必须由算法自动选择出来。

其次，另外一个假设是如果坐标轴旋转，使用式（3-8）所估算出的法线方向结果必须保持不变。正如上文所说，法线方向的估算的基础就是沿着法线方向像素强度的变化是最快的。但是在估算出法线方向之前，法线方向是未知的，所以算法中必须自己建立坐标轴来计算。这一点根据我们的计算方法来看应该是可以成立的。

对于式（3-8）来说，计算时可以采用先求高斯函数的梯度算子，然后再和图像进行卷积的方法来简化计算。这样，该算法就估算出了整幅图像的法线方向了。

3.5 对估算的法线方向的矫正

由于较为精确的估算法线方向是十分困难的，所以该算法中使用了一种“先估算，后纠正”的方案来对估算结果进行修正。

对于高斯一阶导计算出来的法线方向来说，属于使用边缘像素强度的数学方法来估算的。而上面 2.3 中所进行的计算是使用轮廓跟踪方法获得的边缘点然后使用几何的方法来获取的，两种方法各有优缺点，如果可以综合两种方法，预期应该会达到一个较为精确的结果。

在骨骼的边缘是闭合的前提下，我们选择一个权值 q 来权衡两种方法之间的比重，即，若最终计算出的较为精确的第 i 点法线方向被记为 $\bar{\theta}_i$，则：

$$\bar{\theta}_i = q\theta_G^i + (1-q)\theta_C^i \tag{3-9}$$

式中 θ_G^i ——第 i 个点由高斯一阶导估算的法线方向；

θ_C^i ——第 i 个点由 2.3 中处理轮廓扫描后结果所得出的法线方向。

如果我们能够找到一种方法来确定 q 值，那么式（3-9）就应该可以得出一个比上述两种方法单独计算的法线方向都要精确的结果。但是，对于图像中不同的像素点，点附近的像素强度等变化都不同，所以，如果不同图像、亦或者相同图像的不同像素点都采用完全相同的 q 值，那么 q 的设立就是去了意义。所以，为了保证精确性更高，对于不同的图像以及相同图像的不同点来说，我们都应该才去不同的 q 值，即，公式（3-9）应该改写为：

$$\bar{\theta}_i = q_i\theta_G^i + (1-q_i)\theta_C^i \tag{3-10}$$

式中 q_i ——第 i 个点的高斯权重值。

接下来的主要任务就是要指定方案确定权重值 q_i 的具体取值。如上文所说，实际运用中 CT 图片的边缘检测必将是大量的，所以无论使用何种方法，必须保证 q_i 的确定是由算法自动完成的。人工作业既无法确保精确，

也没办法保证效率。

3.6 估算的可信度评估以及两种方法的权重选择

从式(3-10)不难看出，可以将 q_i 设定为使用式（3-8）即高斯一阶导的方法计算图像法线方向的可信度。那么，为了选择合适的 q_i，可以从对高斯一阶导可信度的评判方式出发。

在理想情况下，对于一个给定边缘的图像，其边缘点的法线方向不应该由于计算参数的不同而改变。但是有上述高斯一阶导即式（3-8）的计算结果会由于方差 σ 的取值不同而变化。所以，也许可以通过观察不同的 σ 取值所产生的不同的法线方向值来确定 q_i 的取值。

假设像素点 i 是一个临近的边缘点，那么如果使用一个选定的区间 $[\sigma_{\min},\sigma_{\max}]$ 中的不同的 σ 来计算其法线方向时，所产生的结果变化较小，即小于一个选定的值 $\theta_{tol}>0$ 时，就认定为估算结果是可信的，并且设定 $q_i=1$，即通常情况下，q_i 可以大致定义为：

$$q_i=\frac{\sigma_u^i-\sigma_l^i}{\sigma_{\max}-\sigma_{\min}} \qquad (3\text{-}11)$$

式中 $\sigma_u^i\le\sigma_{\max}$，$\sigma_l^i\ge\sigma_{\min}$；

$[\sigma_l^i,\sigma_u^i]$——该区间是满足条件要求的 σ 的取值区间。

上述条件是：该区间中中所有的 σ 取值估算得到的点 i 的法线方向之间的差值都小于或者等于上面所规定的 θ_{tol}，并且 $[\sigma_l^i,\sigma_u^i]$ 也是满足这一条件的最大区间。

也就是说，在 $[\sigma_l^i,\sigma_u^i]$ 区间外的 σ 值所求的法线方向的差异都大于 θ_{tol}。

在这种方法下，我们可以找到生存周期最长的θ值。例如，图 3-7 中就是我们计算出的某一个近似边缘点上的$\theta-\sigma$曲线。从图中很容易可以看出在其中的一段σ上θ的变化很小，小于θ_{tol}，这样，就可以确定出σ_l^i和σ_u^i的值。从而确定出q_i的值。此外，这种方法同时也帮助我们确定了该点上θ的最佳取值即$\left\langle \theta_i^{l,u} \right\rangle$。

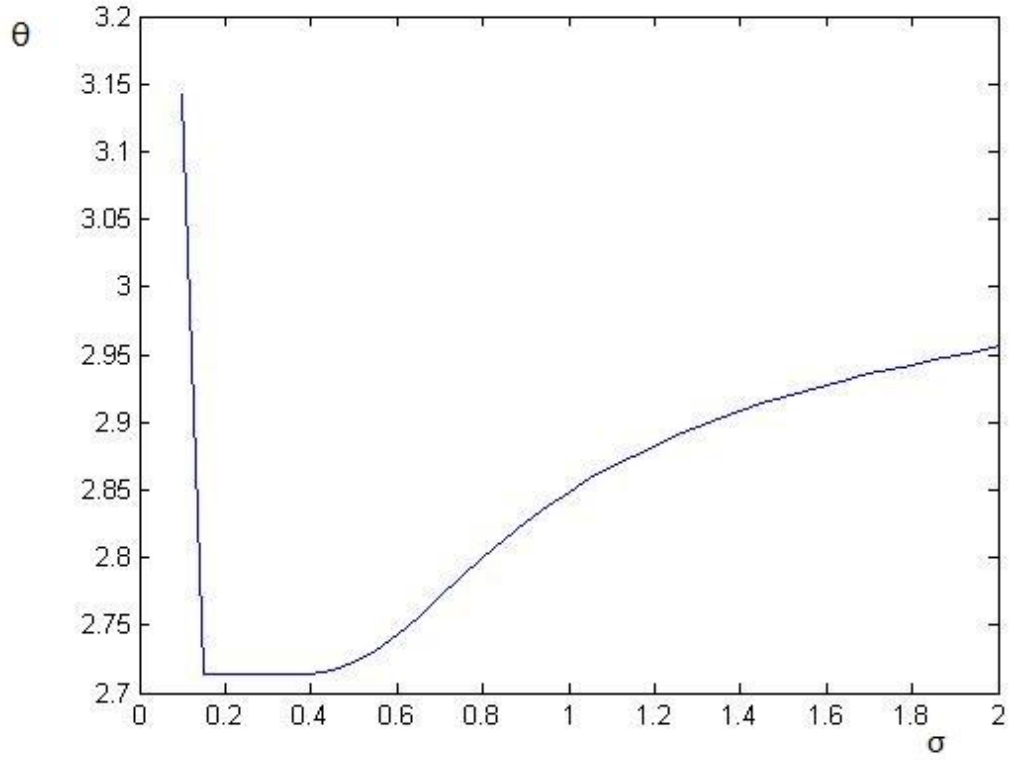


图 3-7 高斯一阶导估算的θ值与σ的变化关系

显然，我们应当将$\left\langle \theta_i^{l,u} \right\rangle$作为点$i$上的法线方向的最佳取值。而这种对于不同点、不同图像来分别计算的方法无疑要比将q_i认定为一个常量来估算要精确的多。在$\sigma_{\max}$和$\sigma_{\max}$的取值方面，$\sigma_{\min}$的取值可以选择的比较小（该

算法中取 0.1），$\sigma_{\max}$ 的取值也可以选择的很大（该算法中取 2.0）。相对而言，θ_{tol} 的取值需要谨慎一些，毕竟是 θ_{tol} 在控制着估算的精确度。根据我们的实验以及调整，最终该算法中取 $\Delta\theta_{tol}=10^{\circ}$，在这样的取值下，算法整体表现更好，结果更加精确。

在上述的算法中，q_i 的取值完全决定于 $\sigma_{\max}$、$\sigma_{\max}$ 和 θ_{tol} 的取值。例如，如果 θ_{tol} 选择过大，那么 $\sigma_u^i \to \sigma_{\max}$，$\sigma_l^i \to \sigma_{\min}$，就会导致 $q_i \to 0$；而如果 θ_{tol} 选择过小，那么 $q_i \to 0$。所以，在实际运用中，还是应该根据不同的情况而尝试更多的三种参数的组合选择，最后根据自己的需要确立最好的组合。

此外，还有很重要的一步就是设立重比例因子来制约 q_i，即我们选择一个重比例因子 γ，然后在所有涉及到 q_i 的地方用 γq_i 来代替。这样做是为了降低权重值受 $\sigma_{\max}$、$\sigma_{\max}$ 和 θ_{tol} 的影响的敏感程度。至于 γ 的取值，可以根据自己的假设来定，建议应该至少大于 0.5，毕竟应该对于 q_i 的信任程度高一些。在本算法中，我们假设至少有 80%的像素点的估算法线方向有 90%以上的信任度，所以 $\gamma=0.72$。当然，这个不是一定的，实际应用中应该按照实际情况来调节 γ 的取值，最终使得结果尽量精确。

最后，确定高斯一阶导估算出的法线方向的可信度为 γq_i，则等式（3-10）变成了：

$$\bar{\theta}_i = \gamma q_i \theta_G^i + (1-\gamma q_i)\theta_C^i \qquad (3\text{-}12)$$

此外，我们还需要由 σ_l^i, σ_u^i 来确定点 i 上的最佳的 σ 取值以便之后的边缘

检测算子可能会用到。毫无疑问，最佳的σ肯定应该在$[\sigma_l^i,\sigma_u^i]$这个区间上，由于我们使用的是最大梯度的边缘检测算子，所以应该选择会使得$\|\nabla G(\sigma)*I\|$能够取得最大值的σ作为点i的最佳σ取值，如式（3-13）。

$$\{\sigma\,|\max\{\|\nabla G(\sigma)*I\|\},\sigma\in[\sigma_l^i,\sigma_u^i]\} \qquad (3\text{-}13)$$

3.7 构建一维信号进行边缘检测

一旦按照上面的过程得到了估算并且矫正过的法线方向，并且获得了点i的最佳σ取值，就可以以点i为中心然后由沿着点i法线方向上的点的像素强度构建成一个一维信号S_i。正如上面所说，我们认定由“先估算，后纠正”的方法计算得出的法线方向是很值得信任的，只有这样，才能保证一维信号S_i在边缘点处变化的最快，而这就是我们接下来进行边缘检测的唯一标准。

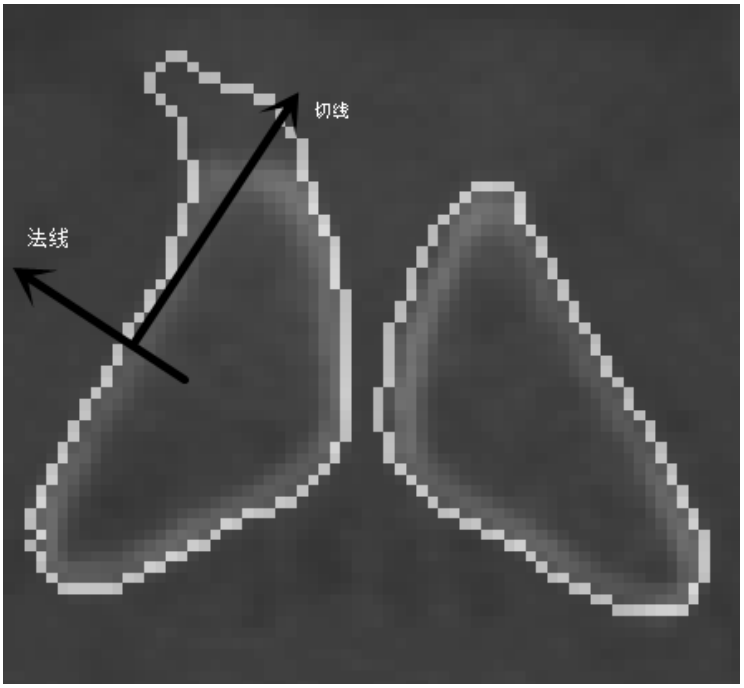


图 3-8 构建一维信号示意图

对于构建一维信号，其含义如图 3-8 所示，通过上述算法，我们找到了法线方向，那么对于轮廓跟踪结果中的所有边缘点，我们在以其为中心、按照其法线方向构建一维信号，在这个以为信号上，我们选取中心点以及中心点附近的其他的几个点，如法线方向和法线反方向分别选取 2 个点，然后再在这几个点中进行边缘检测。

在最后的实验中，我们尝试了两种边缘检测算子，一种是高斯一阶导作为边缘检测算子，另一种是梯度幅度的方法。

对于高斯一阶导，我们采用有限一维∇G。选用它的主要原因有两个：第一，较为简单易用；第二，在 Canny 等著名的边缘检测算子中，这种方法多次被运用，其优势不言而喻。在本算法中，通过使用矫正过的法线方向、最佳的σ值以及上述一维信号，∇G应该可以表现的不错。∇G是一种梯度最大的边缘检测算子，则在一维信号中拥有最大的边缘强度的点将会被认定为边缘点。

对于梯度幅度，是后来为了实验而加上去的。事实证明，梯度幅度的方法在有些情况下表现的甚至会更优于高斯一阶导。而且梯度幅度使用起来也相当简单，就是将一维信号放到梯度幅值图中，然后寻找其方向上的灰度值最大的点即可。

3.8 后期处理

正如上文所说，原本的轮廓跟踪算法是保证边界闭合的。但是经过对法线方向的处理已经最后构造信号进行边缘检测的处理后，很容易会出现边界不闭合、出现无用点的情况。遇到这种情况，只需要一些简单的处理即可。

3.8.1 二次轮廓跟踪

在理想情况下，轮廓必然是闭合的。但是由于一些噪声的影响，最终生成的结果很有可能非闭合。在这种情况下，会有很多种选择来处理。在本算法中，采取的是二次轮廓跟踪的方法。其流程如图 3-9 所示。

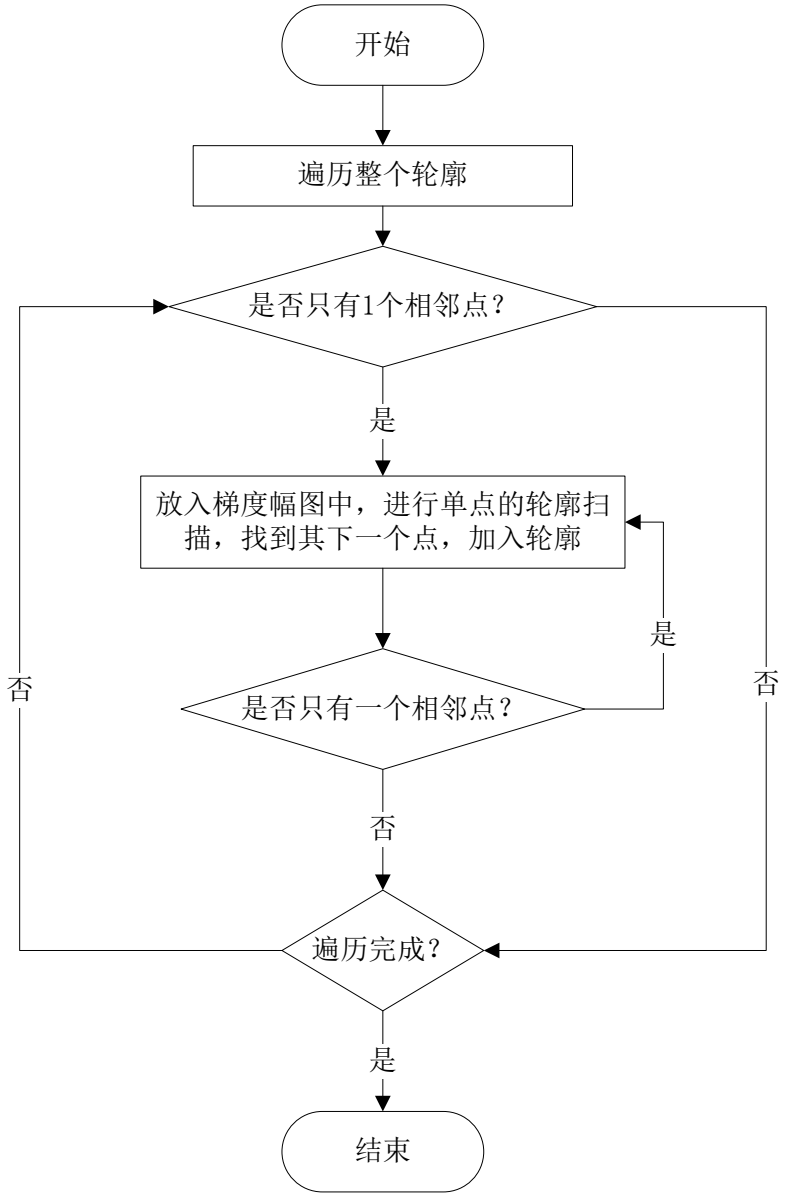


图 3-9 二次轮廓跟踪算法流程图

如图 3-9 中所示，遍历结果中的所有轮廓点，然后找出其中在其 8-邻域中只有一个相邻点的那些点。这些点就是轮廓断开的地方。然后将这个断开的点作为当前点，跟原来的方法一样去寻找下一个点，每找到一个点就将其加入轮廓中，然后判断它是否是单邻居点，如果是则继续用相同的方法知道找到一个双邻居点。这就说明已经把断开的地方连接上了。然后用相同的方法继续遍历整个轮廓，知道轮廓中找不到单邻居点。

这种算法的出来保证了轮廓的闭合性。

3.8.2 删除无用点

在对轮廓进行了这么多的处理之后，自习观察轮廓，法线还是能够看到一些散落的无用点。如图 3-10 所示。

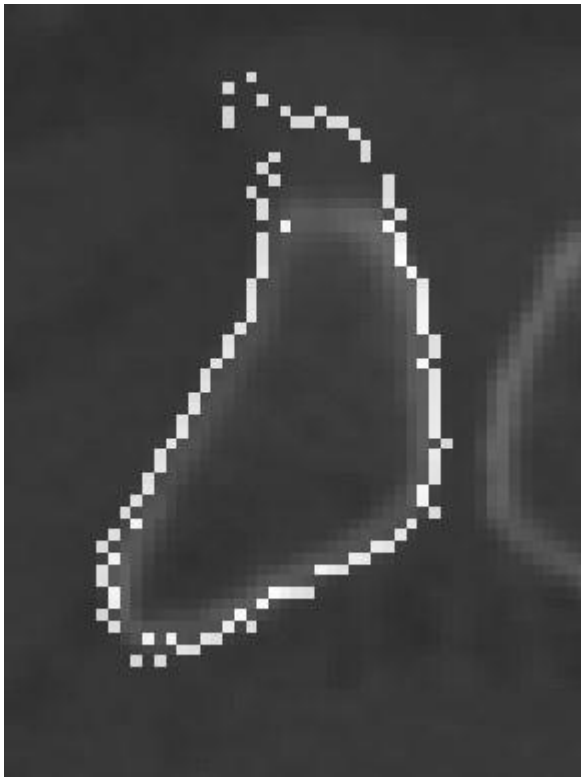


图 3-10 无用点举例

对于这样的情况，需要根据实际需要来指定规则删除无用点。在本算法中，规定了散落的 1-2 个单独的点被视为无用点，检测出来删除即可。然后再使用上述的二次轮廓跟踪的方法使得轮廓闭合。之后那些跟轮廓完全无连接的片段无论多长也都定义为无用点一律删除。

当然，可能在一些其他的情况下单独的 3-4 个点等也需要被列入无用点中，这个需要灵活变动。但是在大多数情况下，上述的规则应该可以应对。

此外本章所描述的两种后期处理方法即二次轮廓跟踪和删除无用点的方法需要相互配合，两者之间的顺序也很有可能会影响结果。在本算法中采取的是，先删除 1-2 个的无用点，然后进行二次轮廓跟踪，形成闭合轮廓之后，再去删除那些和闭合的轮廓完全不相连的无用点。

3.9 本章小结

本章着重介绍了整个估算-矫正算法的实现。

首先，介绍了估算矫正算法实现的基础，也就是轮廓跟踪算法的主要实现方法。紧接着有讲述了如何将轮廓跟踪的结果边缘点转化成边缘点的法线方向。

其次，介绍了使用高斯一阶导的方法来估算法线方向的方法。

之后，着重介绍了将高斯一阶导估算的法线方向和轮廓跟踪转化的法线方向相结合的方法，即估算-矫正算法的核心。接着，又介绍了如何分配上述算法中两种方法所占有的权重，计算高斯一阶导的可信度。

后来，介绍了根据上面矫正好的法线方向来构建一维信号从而完成边缘检测。

最后，介绍了对于算法最终结果出现一些问题时的简单处理方法，使得结果变得更加完善。

第4章 实验结果及分析

在本章的实验结果中，列举出了两幅最有代表性的图像的实验结果。两幅原图分别为图 4-1、图 4-2。

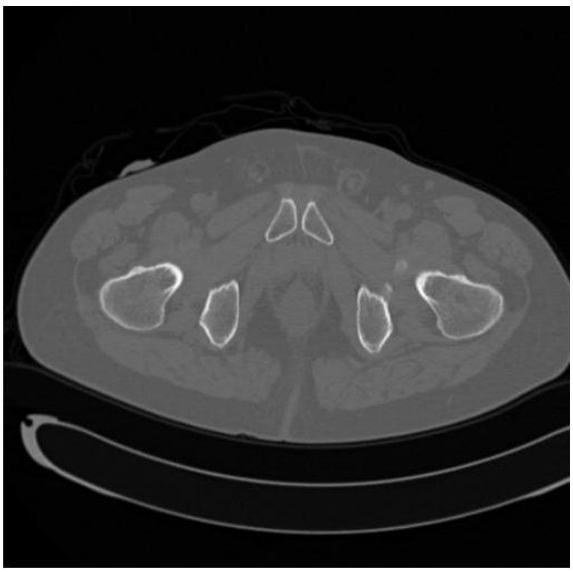


图 4-1 实验图像一原图像

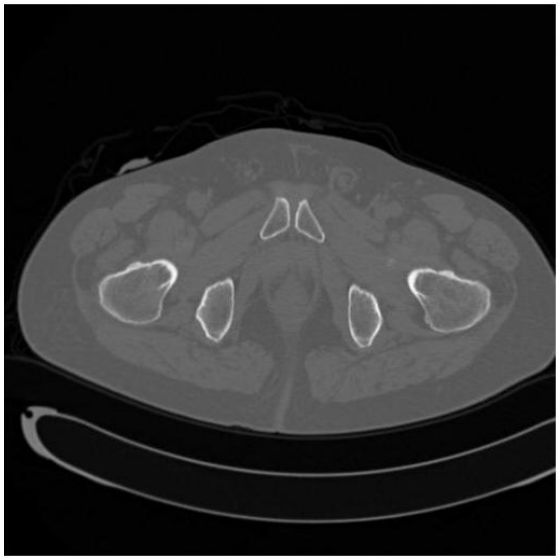


图 4-2 实验图像二原图像

以下所有实验结果都是基于图 4-1，图 4-2 中的两幅原图像。

图 4-3 程序实验界面

图 4-3 为算法的整体界面，由图中可以看出，该界面已经提供了整个算法所需的中间过程的显示。足够进行实验。

4.1 梯度幅度实验结果

根据文中提到的方法，首先第一步就是要求得原图像的梯度幅图，图 4-1 和图 4-2 的梯度幅图结果分别如图 4-4 和图 4-5 所示。

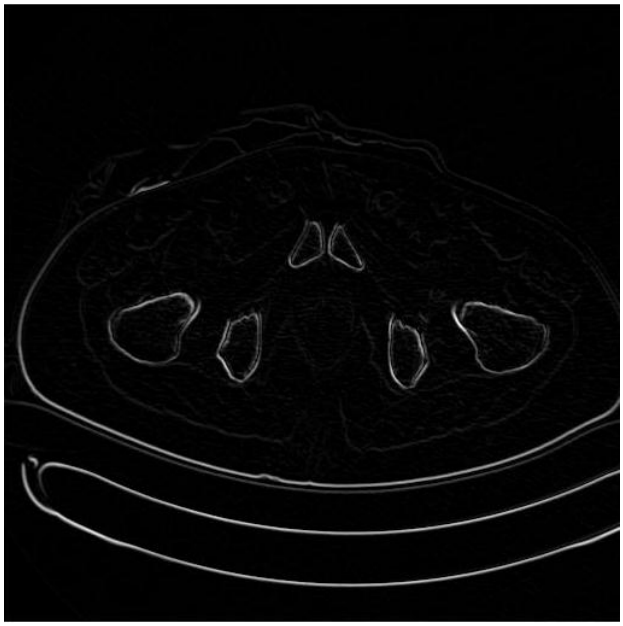


图 4-4 实验图像一梯度幅值图

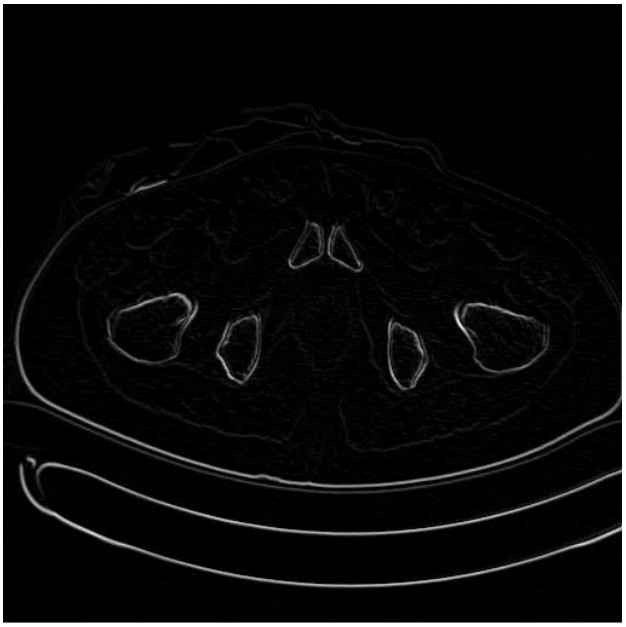


图 4-5 实验图像二梯度幅值图

由图 4-4 和图 4-5 可以看出,梯度幅值图像的确可以明显的加强边缘区域和非边缘区域的差别,为之后的轮廓跟踪提供了十分重要的依据.

此外,噪声的影响在梯度幅度图像中也得到了降低。在梯度幅度的结果图像中，噪声的影响变得很小。

4.2 轮廓跟踪结果实验

图 4-6 和图 4-7 给出的是单纯的轮廓跟踪之后的结果:

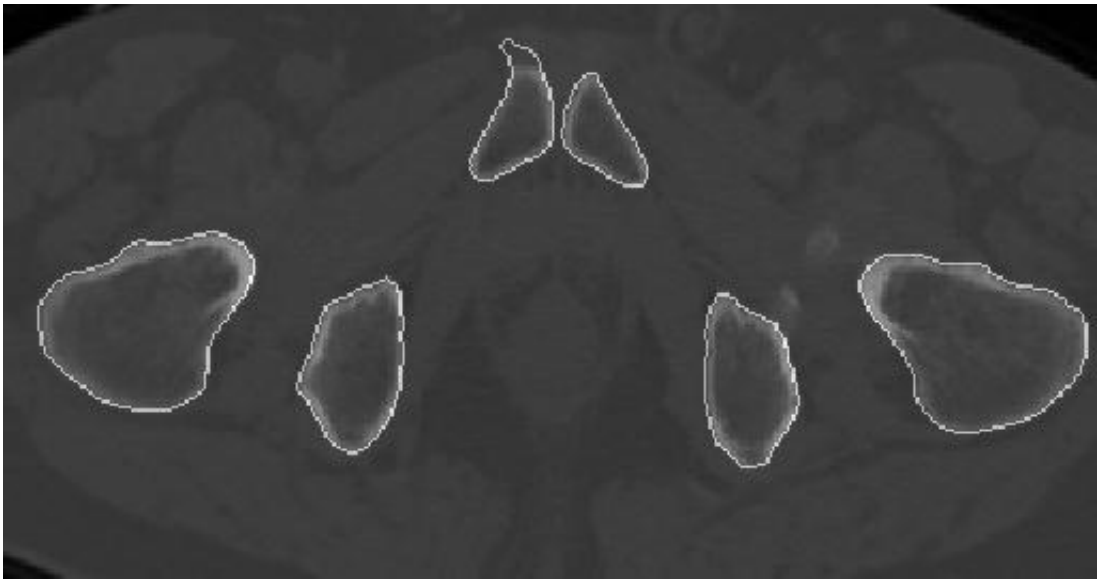


图 4-6 实验图像一的轮廓跟踪结果

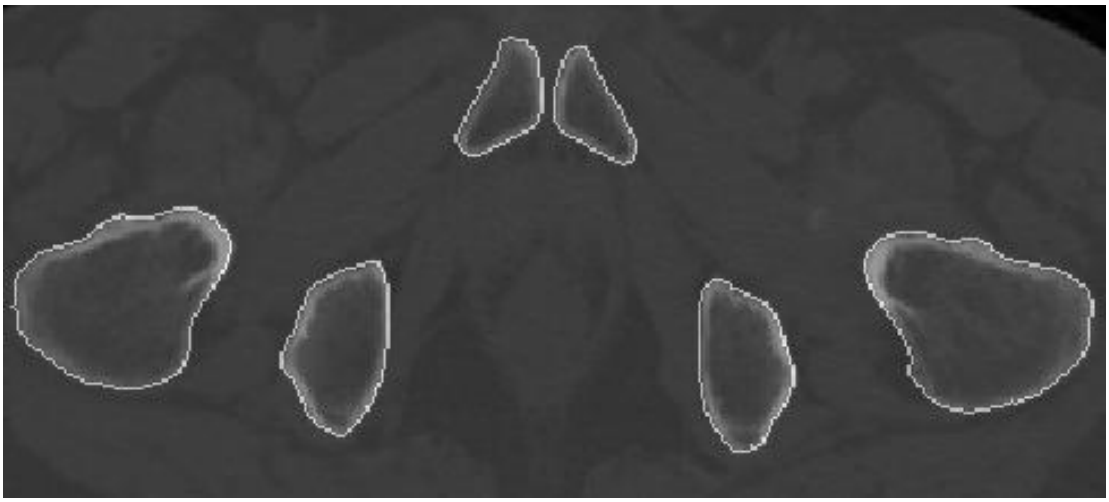


图 4-7 实验图像二的轮廓跟踪结果

由图 4-6 和图 4-7 不难看出，在轮廓跟踪算法正常运转的情况下，其结果还是比较令人满意的。当然，在实验当中也遇到一些由于噪声影响太大导致算法无法正常运转的情况，但是根据情况适当调整阈值以及搜索区域之后，大部分情况下轮廓跟踪可以正常的运转。

不过，图 4-6 上面偏左的那块骨头轮廓的上部明显出现了较大偏差，由于顶部噪声较大，算法直接将其当作了轮廓来处理。出现这么大的偏差，在正常的算法中是绝对不允许的，这也就是轮廓跟踪算法的缺点，一旦一个点走错，接下来就没办法纠正，只会越走越偏。

4.3 整体算法结果实验

图 4-8 和图 4-9 给出的是整体算法完成之后的结果:

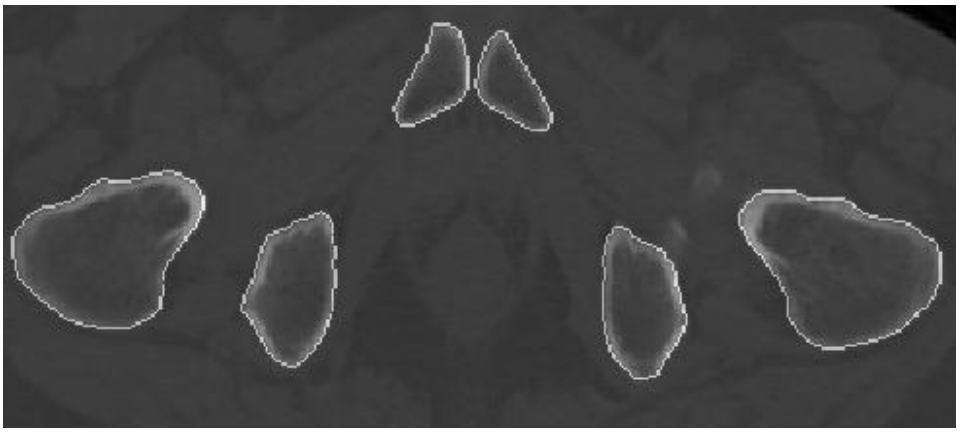


图 4-8 实验图像一的整体算法结果

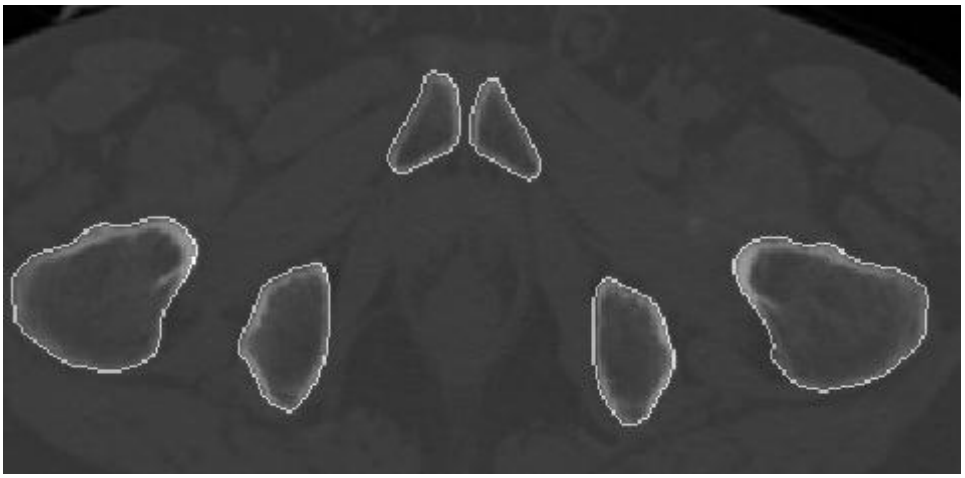


图 4-9 实验图像二的整体算法结果

不难看出，在图 4-8 和图 4-9 中，整体轮廓都比较好，虽然如果放大来看一些细节的话多少还是有些缺陷，但是整体效果不错。并且相比较单纯的轮廓跟踪算法都有了一些提升，优势比较明显。

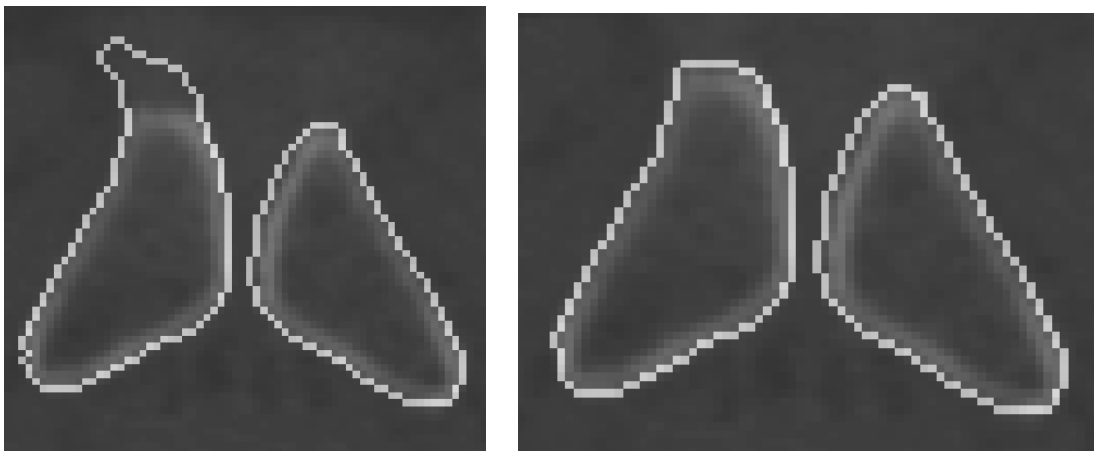
并且，在后期的一些二次轮廓检测、删除无用点等算法的处理下，整个轮廓保证闭合，并且没有出现明显的无用点。

总之，单单看最终的整体算法结果图还是比较令人满意的。接下来需要将该结果跟一些其他常见的边缘检测算法相比较。

4.4 实验结果对比

4.4.1 与轮廓跟踪的比较

图 4-10 是相同的一副图像在单纯的轮廓检测算法下和在我们的整个算法下的结果对比：



a) 单纯轮廓跟踪结果　　b) 完整算法结果

图 4-10 相同图像在于单纯轮廓检测算法和我们整体算法下结果的对比

在图 4-10 中，a)是单纯轮廓检测的结果，b)是整体算法的结果。

将两图这样发大对比来看，差距还是相当大的。特别是左边的那块骨骼，在轮廓跟踪下的结果 a)的上方明显受到噪声的干扰而完全走偏。而在 b)中上方却完全修复了这一问题，使得在肉眼看来整体轮廓很完整并且比较正确。

4.4.2 与 Canny 边缘检测算子的比较

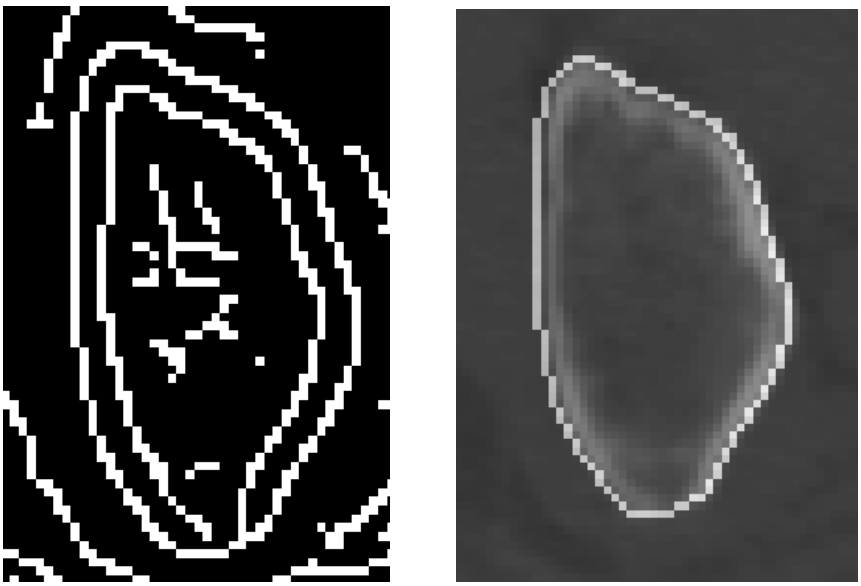
图 4-11 是实验图像一在 Canny 边缘检测算子的检测下所得到的完整的图像:

图 4-11 实验图像一在 Canny 算子下的结果

将图 4-11 和图 4-8 进行简单的对比即可法线，Canny 算子的结果中不但画出了我们想要的骨骼的轮廓，同时还画出了很多其他的组织的轮廓已经很多噪声点。这无疑是我们算法的优势之一。

图 4-12 是相同的一副图像在我们的整体算法下和 Canny 边缘检测算子下的结果对比，其中 a)是 Canny 边缘检测算子的结果，b)是我们自己算法的结果。

在这样细致的局部对比下，不难发现，a)图下方有一点上内轮廓和外轮廓相连了，并且内轮廓不闭合。按照 a)的结果去寻找外轮廓的话一定会收到和内轮廓相连一点的困扰。虽然我们的算法只检测外轮廓，但是保证了外轮廓的清晰可见。此外，Canny 算子同时还画出了很多噪声等的无用点，这点跟本算法就跟相差较远了。



a) Canny 算子局部结果　　b) 本算法局部结果

图 4-12 相同图片在 Canny 边缘检测算子下和本算法下的结果对比

4.5 实验结果分析

综上分析，总体来说，我们的算法整体表现不错。对于一般的 CT 图像来说可以画出其大致的骨骼边缘。

和原本的单纯的轮廓跟踪算法相比较，有了较大的提升，增加了轮廓跟踪算法的抗噪性。和 Canny 等常见的边缘检测算子来比较，也在一些方面体现出了优势，比如，边缘一定保证闭合、不会标出任何不相关的组织或噪声作为边缘点。

当然，在实验中也发现了一些问题，比如，轮廓跟踪固有的大问题就是容易由于某一点的错误导致整体出现大的偏差，在这一点上，我们的算法虽然有了一些改进，但是还是有类似的问题，需要进一步控制。其次，对于阈值的选择，有时候在图片环境出现大的变化的时候可能需要做出一些改变，否则会造成结果不理想。

除了分割结果之外，在测试时还发现了算法整体计算速度偏慢，特别是在对图像进行高斯一阶导求值时，效率较低，总结其原因大概有下列两点：

第一，高斯一阶导是对于整个图像进行求值，涉及像素点过多，所以整体计算速度偏慢。

第二，实验算法在 Matlab 环境下实现，所以整体偏慢。

对于这些问题，需要更多的实验和调整来更正。

4.6 本章小结

本章介绍了文章所介绍的算法的实验结果。

首先展示的是单纯的轮廓跟踪算法的实验结果。接着就是整个估算-纠正算法的实验结果。

之后，又展示了一些比较有特点的结果的相互比较，并且描述了不同结果之间的差别。

最后，分析了整个实验结果，讨论了结果的优缺点。

结　论

本文介绍了一种较为创新的估算-矫正算法来对 CT 图像中的骨骼进行边缘检测。

在 CT 图像中，噪声较多，现有的各种算法都多少有一些自己都优点，但是也都有不足的地方。本文选择了轮廓跟踪算法作为基础算法，由其得出的结果去计算出边缘点的法线方向，并用此方向来矫正高斯一阶导估算出来的边缘的方向。矫正过程中，通过对高斯一阶导的可信度评估来行程一个权重参数，权衡两个法线方向所占比重，最终得出的法线方向应该优于两种方法单独计算的结果。

在这之后，沿着法线方向构建一维信号，然后利用边缘检测算子来找出边缘点，形成边缘。

整体完成之后，我们使用多张图片对算法进行了系统的实验，并且也选用了一些常见的其他算法来进行对比。从最终的实验来看，这个算法还是有一定的优势的：

第一，跟单纯的轮廓跟踪算法相比较，增强了其抗噪性，也能够修复一部分轮廓跟踪算法中的错误轮廓。

第二，跟常用的 Canny 边缘检测相比较，保证了轮廓的闭合性，并且只会单独的画出需要的骨骼轮廓，不会画出噪声或者其他组织等无用点。

第三，由于算法本身是基于轮廓跟踪算法的，所以算法继承了轮廓跟踪算法的最大的优点，即形成的轮廓一定是闭合的。

但是，必须承认，该算法也有很多缺点需要改进：

首先，由于以轮廓跟踪算法为基础，对于轮廓跟踪的致命缺点，即，容易被一个错误点引导后续整个边界出现问题，虽然有了一定的修正，但是没有完全解决，多少还是有一些这样的问题。

其次，本算法需要手动指定种子点，这一点导致不能实现完全的自动化运行，这也是个比较致命的缺陷。在实验中，我也尝试了很多方法来指定种子点，但是总是不能保证整幅图像的每块骨头轮廓都可以被找到。这一点有待继续加强。

此外，由于本算法在计算法线方向时，其估算值要对整张图片进行估算，这样就大大的降低了算法的效率，计算速度偏慢。当然，计算速度过慢也是和 Matlab 环境有分不开的关系的，以后可以用 C++等其他语言实现来

尝试改变算法效率。

综上所述，本算法相比于一些常用的其他算法而言，有一些自己的优点，但是也是存在着自己很多的不足和缺陷，有待进一步的完善。

致　谢

大四下学期半年的忙碌生活转眼间就过去了，我们的毕业设计也到了最后的收尾阶段。仔细回想这半年的时光，如果没有在各方面老师的监督和教导以及同学们之间的相互帮助，我是很难完成这次毕业设计的。

所以，在整片文章的最后，我要感谢这半年了对我有过帮助的老师和同学们：

首先要感谢我的指导老师董开坤老师，在整个毕业设计期间，无论我遇到什么问题去找董开坤老师，董老师一定详细耐心的教导我。在我的开题报告上以及中期检查时都给我提出了很多宝贵意见，让我的毕业设计得到了进一步的完善。

其次我要感谢实验室的郭长勇老师、周生俊学长和胡鑫学长，这半年来，我从对图像处理的一窍不通到现在自己完成了整个算法的编码工作和这几位老师和学长的帮助是分不开的。无论何时我去找他们讨论技术问题，他们都详细给我解答；他们还推荐了一些相关书目给我参考，在我的算法遇到问题的时候帮我出谋划策。毫不夸张的说，这几位都是我在图像处理方面的启蒙老师。

除了实验室的几位老师、学长，我还要感谢我的班主任周广禄老师。作为我们的班主任，他一直关心我们的学习生活，经常询问我的毕设进度，提醒我抓紧时间。在论文的修改等方面，他也给了我很多意见和帮助。

我还要感谢迟乐军老师和伯彭波老师在验收程序是给我提出的宝贵意见。

此外，我还要感谢我身边的各位同学。在大学的最后半年时光里，我们互相帮助，相互支持的完成了毕设，没有这帮好朋友、好战友在我遇到困难的时候支持我、在我松懈的时候鞭策我，我很难完成今天的任务。

最后，感谢大学四年来所有的老师和同学，这份毕业设计算是对我大学4年的总结，而你们，是大学4年来一路陪我走过的人。

感谢你们所有人，因为你们我才能完成这次毕业设计！

参考文献

1. M. L. Comer and E. J. Delp. Segmentation of textured images using a multiresolution gaussian autoregressive model. IEEE Trans. Image Process. 1999,8:408-420
2. 曹军. 基于动态轮廓模型的图像分割算法研究. 太原理工大学硕士学位论文. 2010:20-25
3. D. Wang. A multiscale gradient algorithm for image segmentation using watersheds. Pattern Recognit. 1997,30(12):2043-2052
4. 吴良武,侯建华,张勇,秦绪佳,欧宗瑛.用邻域运算从 CT 图像中分割骨骼. 中国生物医学工程学报.2003,3:199-240
5. R. Adams and L. Bischof. Seeded region growing. IEEE Trans. Pattern Anal. Mach. Intell. 1994,16(6):641-647
6. 李银波,洪波,高上凯,刘凯. 人体 CT 切片图像中骨骼的分割. 生物医学工程杂志. 2004,21(2):169-173
7. 杨晖. 图像分割的阈值法研究. 辽宁大学学报. 2009,2:135-137
8. 王亮亮,王黎,高晓蓉,王泽勇. 两种改进的局部阈值分割算法. 现代电子技术.2009,14:78-80
9. D. Williams and S. Mubarak. A fast algorithm for active contours and curvature estimation. CVGI: Image Understand. 1992,55(1):14-26
10. 姚滨. 动态轮廓线模型在图像分割中的应用与研究. 西安电子科技大学硕士学位论文. 2008:13-24
11. 周继鹏,耿国华,周明全. 一种新的动态轮廓模型. 计算机研究与发展. 1998,35(8):734-738
12. H. Soltanian-Zadeh and J. P. Windham. A multiresolution approach for contour extraction from brain images. Med. Phys. 1997,24(12):1844-1853
13. 李牧,闫继红,李戈,赵杰. 自适应 Canny 算子边缘检测技术. 哈尔滨工程大学学报. 2007,09:1002-1007
14. J. Canny. A computational approach to edge detection. IEEE Trans.Pattern Anal. Mach. Intell. 1986,8:679-714
15. Junna Shang, Feng Jiang. An algorithm of edge detection based on soft morphology. Signal Processing. 2012,1:166-169

16. Gupta, A. DGW-canny: An improvised version of Canny edge detector. Intelligent Signal Processing and Communications Systems. 2011,12:1-6
17. Milan Sonka, Vaclav Hlavac, Roger Boyle. Image Processing, Analysis, and Machine Vision. 艾海舟，武勃. 第二版. 人民邮电出版社,2003:18-20
18. 姚刚,刘勇,雷帮军,董方敏. 自适应梯度幅值和形态学组合滤波算法. 计算机应用. 2010,30(12):3241-3242
19. Jinwei Shi, Chaoyong Guo. The Extraction of Circle Contour Based on Improved Boundary Tracing Algorithm. Intelligent Human-Machine Systems and Cybernetics. 2012,2:104-107
20. Liu Yan, Zhu Min. A new contour tracing automaton in binary image. Computer Science and Automation Engineering. 2011,2:577-581
21. 郑杰,姬红兵,杨万海. 一种基于快速区域标识的交互式体切割算法. 西安电子科技大学学报. 2007,34(1):49-53
22. 汤伟,黄永灿,高国伟. 基于加性高斯噪声的图像滤波器的设计与分析. 黑龙江科技信息. 2009,29:29
23. 李慧慧,冯前进,陈武凡. 引入高斯函数的互信息法多模态图像配准. 中国医学物理学杂志. 2010,06:2238-2243