

冠状动脉中心线提取

2018.12.5

1 简介

1.1 步骤和实现方式

本次任务是从冠状动脉增强图像提取血管中心线。步骤和实现方式大致如下：

- 图像二值化：读入.mha 格式 CT 图像，阈值处理；
- 空洞填充
- 图像细化：类似腐蚀，取最大内切球心的集合
- 端点分叉点检测：考虑 26 邻域内像素个数，卷积实现
- 断裂分支重连：寻找连接点，条件判断，Dijkstra 最小代价连接
- 构建中心线：在分叉点集基础上追踪，数组存储在 Cell 中

1.2 运行说明

coronary_refine.m 是主要的运行函数。其他函数和脚本：branchReconnect 输入细化后的图像和权重（原始 CT volume 的像素值为可能性），其中调用了三维的 Dijkstra 函数；directConnect 脚本很简短地实现三维图像中两点连直线，但因为用了最短路径所以没有采用；其余函数都是由比较冗长的小功能封装成的。两张图片运行时间小于一分钟。

2 实现方法

2.1 阈值

为了不让阈值化后丢失的成分过多，对后续分支重连的步骤造成困难，这里选择了较小的阈值 $0.1 * \text{原图最大值}$ (2^{16})。这也导致最后结果中分支会显得比 0.5 的阈值下丰富很多，但算法能够原图 (mha) 保证最终中心线和真实血管走向的一致性。

2.2 空洞填充

一开始使用的是 imfill 函数，通过查看源代码可见这个函数调用了 imcomplement 和 imreconstruct 对二值图像进行填充。imfill 对三维图像的处理速度较慢，最终使用形态学库函数 bwmorph3 中的 fill 功能进行处理。

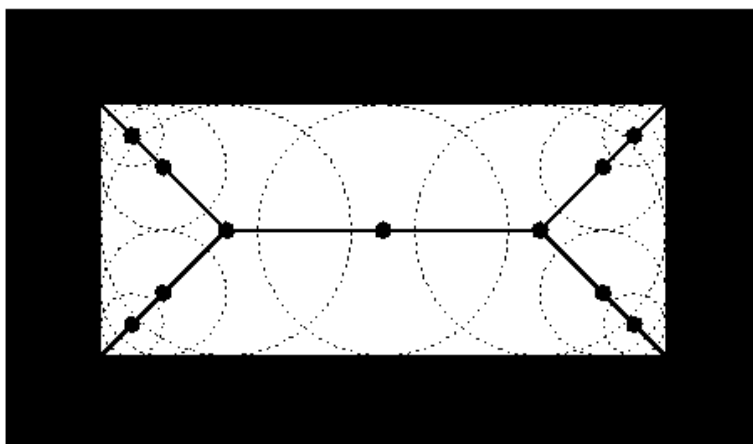


图 1: Skeleton of a rectangle defined in terms of bi-tangent circles.

2.3 图像细化

程序中调用了 `bwskel` 来实现。Thinning 在文献中有两种最为常见的方法，一种被称为 “Onion peeling”¹，顾名思义用不断的腐蚀操作来一层一层地剥开血管，难点是设置一定的条件来保证原有拓扑结构。这个方法也是 `bwskel` 的参考文献中使用的方法。² 还有一种细化方法也和腐蚀有些类似，基本思路是求连通域内部的内切圆心（三维为球心）集合，如图一。

2.4 基于卷积的端点分叉点检测

虽然形态学库函数中同样有 `branch` 和 `endpoint` 的功能，但这两个功能的 `feature` 都导致它们并不适合直接使用。比如 `bwmorph3` 中 `branch` 会返回所有分叉点以及分叉点各自的相邻点。面对如此古怪的 `feature`，不如构造简单的卷积核来求端点分叉点。

- 分叉点检测

首先考虑 $3 \times 3 \times 3$ 全 1 的卷积核。在二值、细化图像非分叉部分，其响应应该为 3。如果将响应大于 3 的视为分叉，其结果中会有很多处于真正的分叉点附近、实际却为原图空白部分的点被误判成分叉。原因就是分叉附近往往点较为密集，空白点的 26 邻域内也容易出现多个 1，导致超出阈值。解决方法很简单，要让卷积能区分出原中心线上的点和空白格，只要在 `kernel` 的中心加大权重，这样空白格的响应和值为 1 的点差距会变得很大，从而被排除在外。代码如下（因为 `convolution` 包含 `padding`，最终结果还需删除 `padding` 部分）：

¹A Sequential 3D Thinning Algorithm and Its Medical Applications

²Ta-Chih Lee, Rangasami L. Kashyap and Chong-Nam Chu Building skeleton models via 3-D medial surface/axis thinning algorithms. *Computer Vision, Graphics, and Image Processing*, 56(6):462-478, 1994.

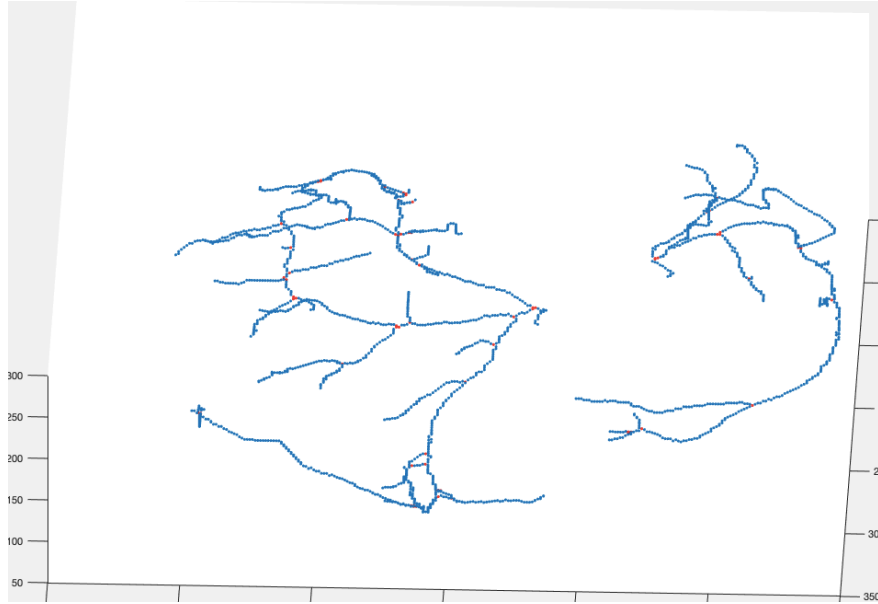


图 2: Convolution-based branch detection

```
kernel = ones(3,3,3);
a(2,2,2) = 2;
branch = convn(kernel, sk)>=5;
```

- 端点检测

类似地如果考虑 $3 \times 3 \times 3$ 全 1 的卷积核，按照直觉，末端的响应应该为 2。然而在和中心线平行的空白邻域内也有大量的点响应为 2，导致结果一眼看上去是原细化图像的”壳“。解决方法同样是加大 kernel 中心点的权重，记为 w ，周围不变；那么响应为 $w+1$ 的就可以确定是端点。

因为检测都是基于卷积，以 convn 等函数的优化程度，其总运行时间也不过 1-2 秒，相比遍历轻便许多。下面以 branch 为例考察其结果，图中分叉处标红。

可以看到几何位置正确且没有遗漏。实际上，如果使用 bwmorph3 的 branch 库函数，二者结果粗看完全相同。关于分支点检测，看文献时常有一些八仙过海各显神通的操作，比如图 3 中这种用球体沿着中心线切血管表面再取球心的方法：

2.5 断裂分支重连

代码在 branchReconnect 函数中。

重连的问题可以描述成将图片中小的片段连接到 coronary artery 的端点或主干上，并丢弃 CT 图中原有的 noise 和 artifacts。可见所有待连接线至少有一端是在 endpoints 集合中的。于是重连的基本思路就是遍历细化图像的所有 endpoints，寻找其周围空间内和自己不在一个连通域的点，对其中条件适当的点予以连接。

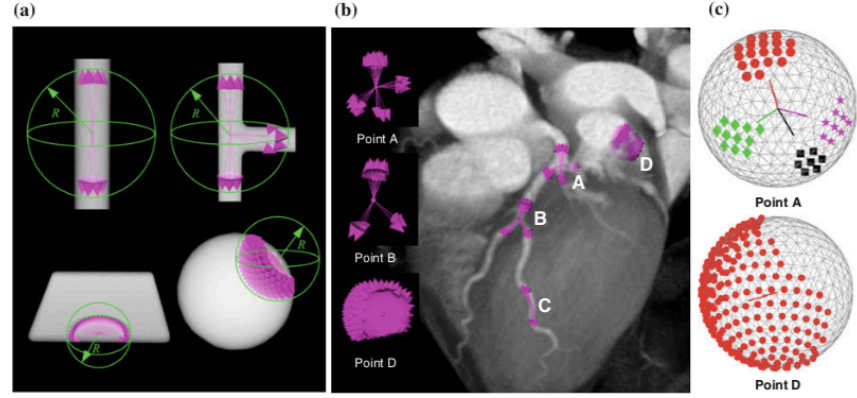


Fig. 3 **a** Ray-casting results in 4 synthetic datasets. **b** Ray-casting results in a CCTA image. Points A, B, C are located in the coronary arteries and point D is located at the boundary of cardiac cavity. **c** shows connected component labeling results at points A and D with different colors. Each preserved ray was

mapped to a point on the *spherical* surface. Gray lines represent the neighborhood relationship of the points on the *spherical* surface. Lines from *sphere* center are the average directions of each group

图 3: branch detection

2.5.1 最小代价路径

代码在 DijkstraConnect 中。因为原.mha volume 实际上是可能性矩阵，那么将其归一化后用 1 减（相当于取反），就得到了 cost(penalty) 三维数组。这里权重非负，求最短路径的算法是 Dijkstra。matlab 中有 graph 相关库函数，但在三维 volume 中实现这样的搜索仍然比较繁琐，因为待搜索的图的边数为 $O(n^3)$ ，其中 n 为边长（以正方体为例）。同时为了减小计算复杂度，在上一步获得了始末点之后，把路径搜索的区域先缩小到始末点之间的长方体再进行搜索。

2.5.2 连接条件

在获得了路径后，还需进行一定的判断才能将刚画出来的路径安置到原图，这是为了避免错误连接的出现。判断路径是否合理本程序采用了两个约束。一是路径上每个点的平均代价，当代价超过某个阈值时，这条路径很可能是被 Dijkstra 强行撮合，强扭的瓜不甜。二是禁止血管的”方向发生突变“，这里计算两个向量的夹角：1 出发点向外延长线的方向向量 2 从出发点到目标点的向量。这里完全没有将目标点处的方向纳入考虑，因为血管几乎可以从任何角度搭接到主动脉上，所以如果使用了这个角度，反而会让一些该连的线被丢弃。此外，放弃连接的条件是以上两个条件相与，也就是当方向特别流畅或者代价特别小的时候，仍然允许连接。用两重条件也增强了鲁棒性。

若无约束，可能出现的极端情况：

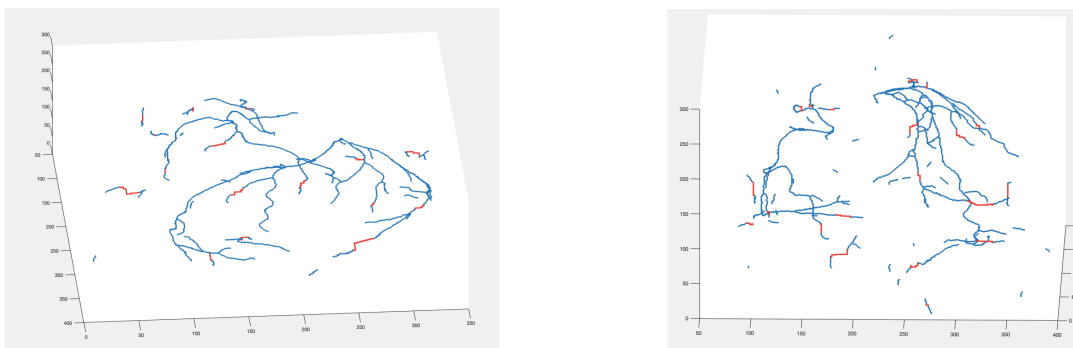
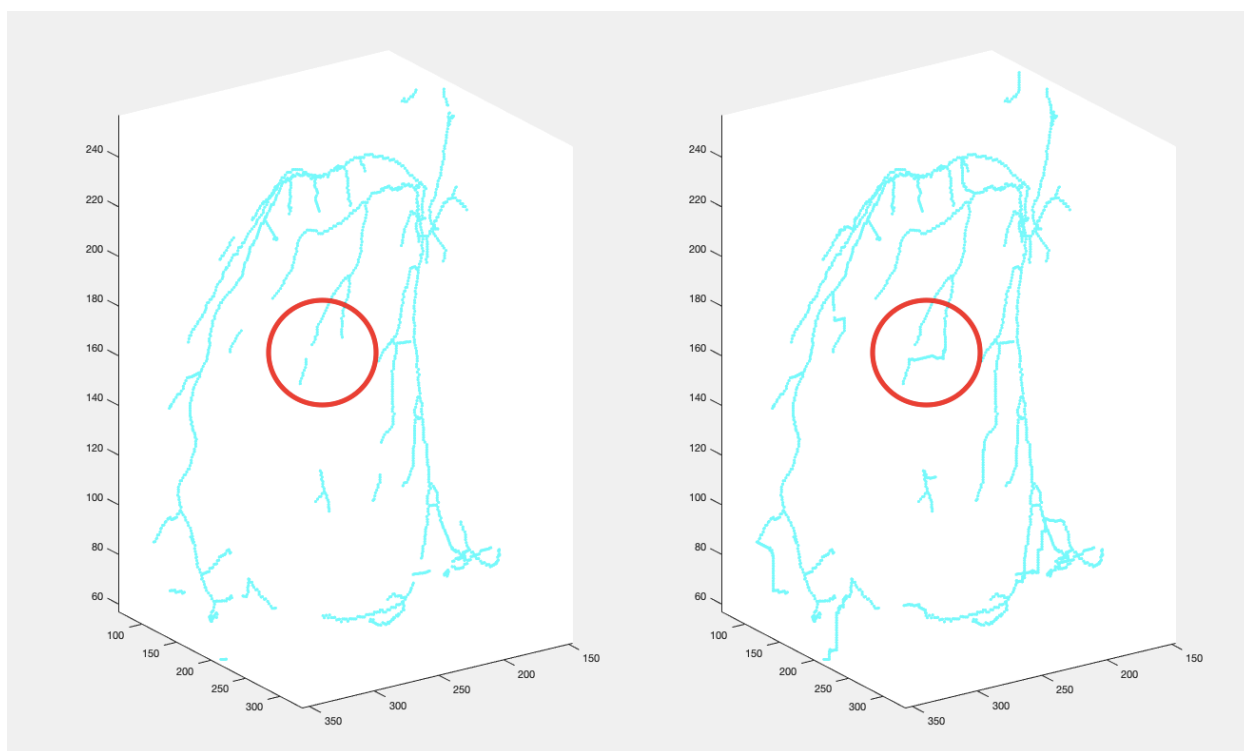


图 4: Minimum cost reconstruction

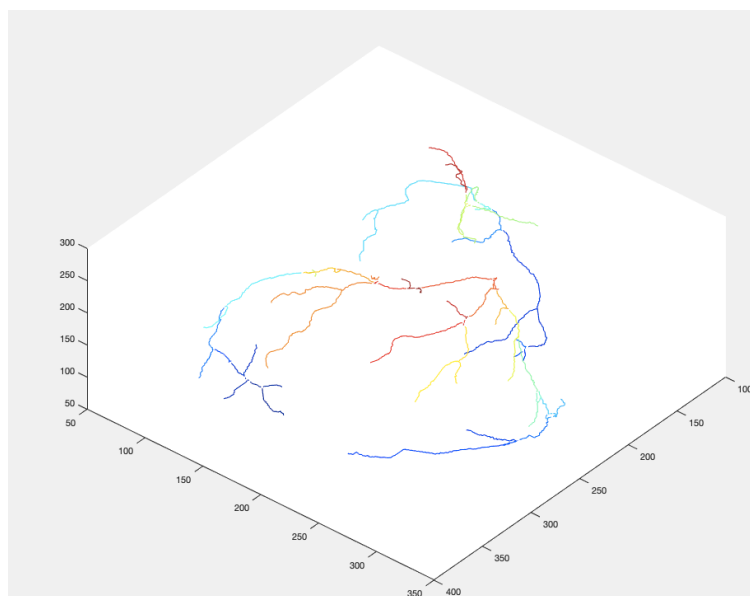
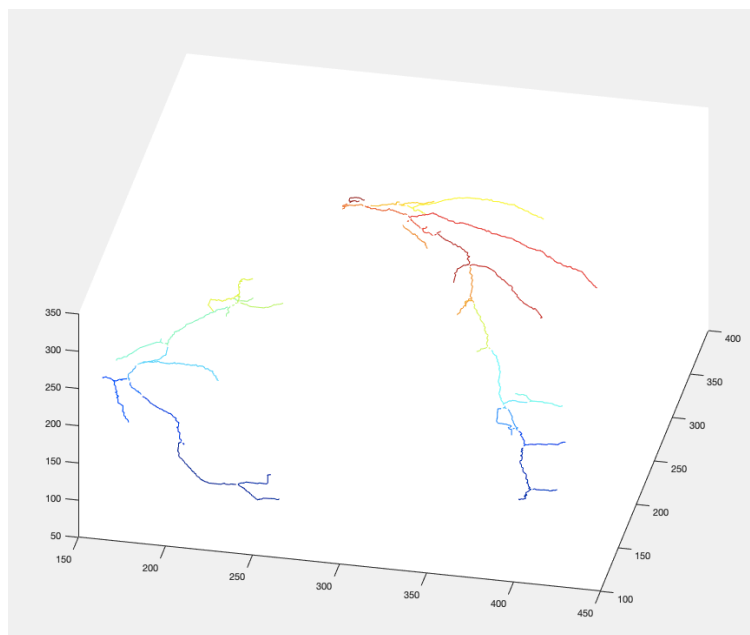


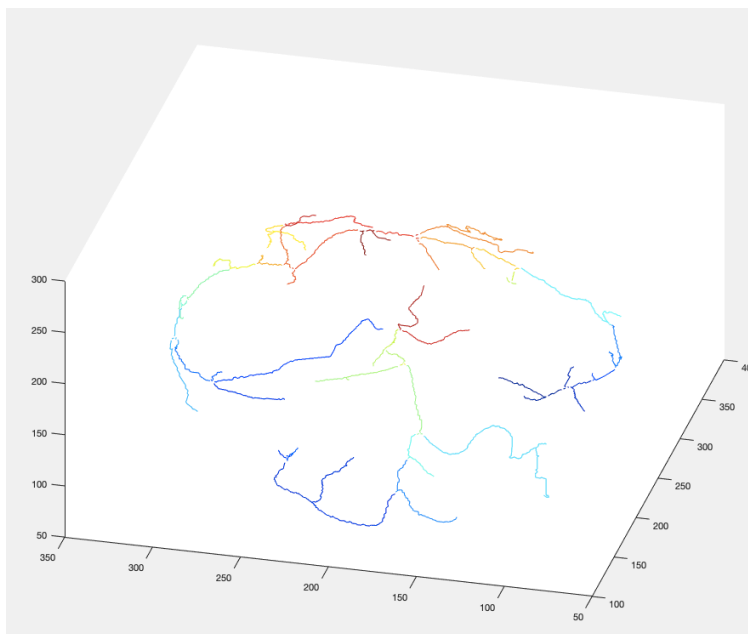
连接效果（红色部分）:

2.6 构建中心线:

代码在 `coronary_refine` 中。这里为了分开各分支，并保持点的有序性，只能在骨架上进行遍历，这里采用的是类似深度优先的实现方式（但因为没有代价、权重等考虑，简单许多）。主要的思路是：维护一个 `closed` 表，记录所有已经搜索过的位置；每次搜索从分叉点开始，到无法找到新邻居结束，搜索的同时构建 `branch`。

3 最终效果





4 问题与缺陷

1. `bwskel` 函数在处理粗血管交汇的三角形区域时，会形成一个环状的 `skeleton`，而不是真正的从血管组织中心出发的三叉树。这个结果很像是用球腐蚀形体，球心所能到达的边界形成的环，怀疑可能和 `bwskel` 的原理有关，但没有找到源码。
2. Dijkstra 的连接条件中，角度的计算或许要综合考虑更长的分支的方向，而不是仅从端点附近几个邻域计算。
3. 很多参考文献都使用了 Frangi's Vesselness Filter，是专门用来进行血管增强的滤波器。我也尝试用它进行图片的预处理，但因为这个滤波器需要设定一组方差 σ 进行多次迭代取最优，电脑跑了两个就几乎死机，遂放弃。
4. 网上搜索 coronary artery 有大量的相关论文，看了参考文献发现有大量思路直观，实现上复杂的方法，因为技术不行，没法付诸实现，比较遗憾。

5 参考文献

Ta-Chih Lee, Rangasami L. Kashyap and Chong-Nam Chu Building skeleton models via 3-D medial surface/axis thinning algorithms. *Computer Vision, Graphics, and Image Processing*, 56(6):462-478, 1994.

Asma Kerkenia, Asma Benabdallaha, et al. A coronary artery segmentation method based on multiscale analysis and region growing. *Computerized Medical Imaging and Graphics* 48 (2016) 49-61.

- M. Schaap et al. Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. *Medical Image Analysis* 13 (2009) 701–714
- Z. Li et al. A coronary artery segmentation method based on multiscale analysis and region growing. *Biomedical Signal Processing and Control* 16 (2015) 1–8
- Z. Li et al. An automatic and efficient coronary arteries extraction method in CT angiographies. *Biomedical Signal Processing and Control* 36 (2017) 221–233
- Ka'lm'an Pala'gyi¹, Erich Soran A Sequential 3D Thinning Algorithm and Its Medical Applications
- Martin Aastrup Olsen*, Daniel Hartung et al. Convolution Approach for Feature Detection in Topological. 978-1-4244-9900-7/11/ ©2011 IEEE Skeletons Obtained from Vascular Patterns
- Guanyu Yang et al. Automatic centerline extraction of coronary arteries in coronary computed tomographic angiography. *Int J Cardiovasc Imaging* (2012) 28:921–933