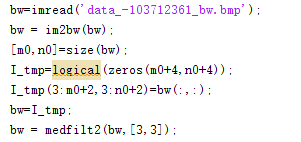
基于视觉显著性的层次形状分解项目技术细节

1. 轮廓及骨架提取

使用的是提供的函数GetContour2和SkeletonGrow1，在对轮廓提取之前，对图像进行了扩展和滤波，因为有些图像的轮廓到达了边界或者图像有些噪声。具体操作是：



所提供的骨架提取函数提取的骨架图像因为有很多非单一像素，所以需要对求得的骨架做细化处理，具体为：



1. 分叉点获取

函数FindJunctionPoint对分叉点进行获取，因为已经对骨架做了细化的处理，所以直接对骨架上的点进行分支数计算来得到是否分叉点，在其中加入了一个参数，即分支延长系数N\_limitpoint，根据不同的骨架形状设置不同的延长系数对一些不必要的小分支进行过滤。

1. 轮廓划分

RangeDivide函数完成轮廓划分功能，根据离骨架的尾端点最近的轮廓点来进行划分。

1. 骨架划分

GetSkelRange函数完成骨架划分功能并完成分叉点优先级的划分，寻找骨架区域的过程由FindSkelRange函数完成，每次由尾端点出发，找到的非分叉点在骨架中删除，直到找到的点为分叉点为止，并记录下每次分支的起始点、终止点以及其坐标，如果该分叉点未被标记优先级，则其优先级设置为该尾端点优先级加1，如果已被标记优先级，则选择其相邻分叉点的优先级最小值加1与其自身优先级比较的最小值作为新的优先级。

1. 分割点获取

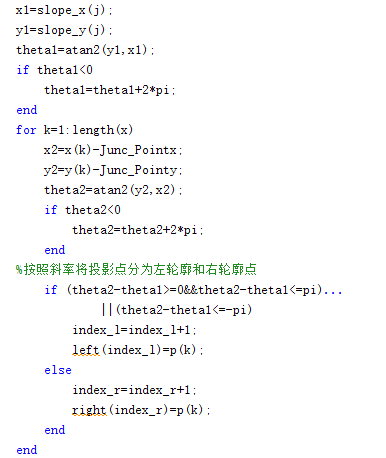
CreateSplitPoint函数完成对分割点候选点的获取，对于每一个分叉点，求得其在上述轮廓划分的每个区域中求取投影点，为简单起见，其投影点即为该分区域离分叉点的最近点。

1. 分支斜率获取

GetJuncSlope2函数完成对骨架分叉点每条分支进行斜率计算，即对每条分支内的点进行求和平均，该平均点与分叉点即构成该分支的斜率，这里加入一个参数N\_Slope，因为有些分支方向会变化，所以引入这个参数即为控制代入斜率计算的骨架点数，具体参数设置要看具体骨架图像。同时，这里引入第一步优化，对相邻比较近的分叉点所对应的分支不做斜率计算，具体方法即是从该分叉点出发的分支的另一端分叉点与其相距的距离同该分叉点的最大内切圆半径（即与轮廓相距最近的点的距离）进行比较。

1. 左右轮廓划分

没有写成单独的函数，在测试文件test5.m中，其具体方法是：根据上述求得的分支斜率，对每个分叉点的所有投影点进行划分，具体步骤如下图：



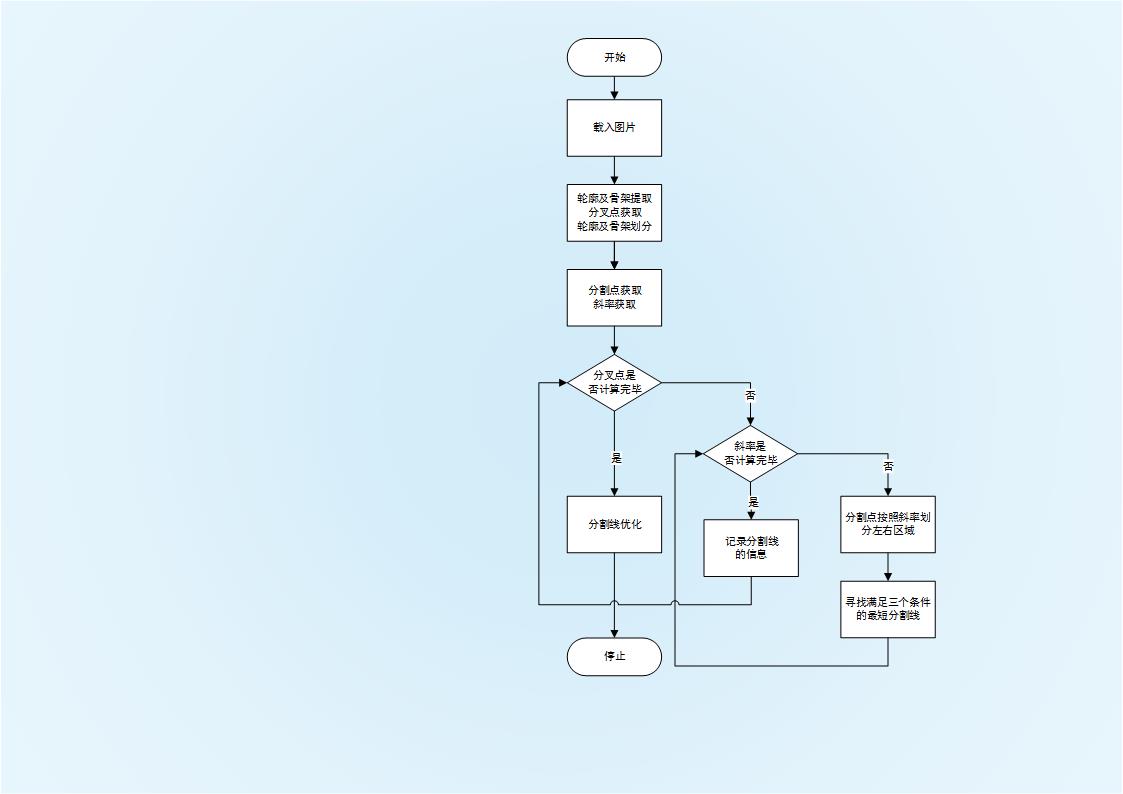
1. 分割线获取

同样是在测试文件test5.m实现的，设置了三个条件进行分割线的选择：1、距离短；2、离分叉点在一定范围内；3、优先选择同锐角，如果没有的话则换成两候选点的分割点中心点在于斜率在同一方向上。并对每一条求得的分割线进行记录。

1. 优化分割线

优化分为三部分：第一个部分基于分叉点合并的分割线优化，这部分在分支斜率获取函数GetJuncSlope2中实现，不过换了种形式，是对相距较近的两点之间不进行分割线获取，分叉点依然保留；第二个部分基于视觉显著度的分割线优化，这部分由GetPVS函数完成，对于每一条分割线进行两点间轮廓距离求解，因为分割线起始点与终止点已经被记录了下来，所以很容易求得其轮廓距离，然后再根据视觉显著度的定义求出该分割线对应的显著度，与视觉显著度阈值进行比较，如果大于阈值则该分割线保留，如果小于阈值，则删除该分割线，即将其优先级设为-1；第三个部分基于分割线优先级的优化，同样是在GetPVS函数中实现的，具体实现为：首先按照优先级大小从低到高开始求解分割线对应的视觉显著度，如果保留该分割线则将这分割线两端的距离设置为该分割线的长度，否则则设置为该分割线对应轮廓线的长度，在计算轮廓长度时，如果该轮廓内有分割线，则带入计算的是该分割线的长度，并且在所有优先级计算完毕后再对所有的分割线计算一遍。

技术流程图如下：



函数说明：

|  |  |
| --- | --- |
| 函数 | 功能 |
| SkeletonGrow1 | 骨架提取 |
| GetContour2 | 轮廓提取 |
| FindJunctionPoint | 分叉点获取 |
| RangeDivide | 轮廓区域划分 |
| GetSkelRange | 骨架区域划分 |
| CreateSplitPoint | 分叉点轮廓投影获取 |
| GetJuncSlope2 | 分叉点所有分支斜率获取 |
| GetPVS | 视觉显著度获取 |