Neuro Imaging Documentation - Draft, DO NOT DISTRIBUTE

Yves Baumann

28.11.2021

1 Problem Statement

Given a 3D scan of a tumor and two arteries, decide whether the Tumor encloses one or both of the arteries completely.
Input: You are given a 256×256×256 scan. Each pixel has one of four values. The pixel is 0 if it is neither tumor nor and artery. It is 1 if the pixel is a Tumor. It is 2 if the pixel is part of the right artery, and it is 3 if it is part of the left artery.
Output: Output whether there exists a 2d slice of the scan, where tumor encloses one or both of the arteries. If the Tumor does enclose one of the arteries, output which artery and for which slices the tumor encloses the artery.

2 Solution Idea

We solve this problem by running an adapted Bread-First-Search on each slice of the scan:
Think of the slice as an undirected Graph, where each pixel represents a vertex and each vertex has edges to all of it's 8 direct neighbors. The vertices have values corresponding to their respective pixel values. We now greedily "color" all neighboring vertices of the artery with value zero, and repeat this procedure until the artery has finished "growing". Finally, if one of the border pixels has the same color as the artery, it must hold that there exists a path from the artery to the border and thus the tumor cannot enclose the artery on that slice completely!

3 Pseudo Code

Algorithm 1 Adapted BFS

```
Require: Scan of size 256×256×256
  for each 2D slice and artery in the scan do
      q = Queue
      for each pixel in the slice do
          if pixel belongs to artery then
              q.put(pixel)
          end if
      end for
      while q.notEmpty() do
          Current = q.get()
          for all neighbors of vertex Current do
              if Vertex is uncolored then
                  Color Vertex same as artery
                  q.put(Vertex)
              end if
          end for
      end while
      for each Vertex on the border do
          Check whether it has the same color as an artery
      end for
  end for
```