Определение непрерывности костной стенки пазух

Общая проблематика

Основная задача состоит в том, что нам нужно как бы пройтись по контуру пазухи и понять, что стенка, которая прилегает к этому контуру - имеет какие-то резкие отклонения, которые продолжаются какое-то продолжительное время, а не какой-то резкий одноразовый скачок интенсивности стенки.

В качестве основного алгоритма был выбран такой набор действий:

- Перебрать слайсы по одной оси (то есть рассматривать не как 3D изображение);
- Внутри сегментации на конкретном слайсе найти все её контуры;
- Для каждой точки мы либо уже получили нормали или их нужно найти;
- Провести отрезки параллельно нормали каждой точки и выбрать по некой эвристике интенсивность, которая будет отвечать за стенку или возможности её отсутствия;
- Применить какие-то алгоритмы для определения непрерывности стенки.

Объяснения данного решения

- От анализа 3D изображения мы сразу отказались, так как изменение в одномерных данных ещё как-то можно анализировать, а вот что такое обойти контур для 3D и представить его как последовательности интенсивности изображения? Не понятно.

Способ через 3D представление

Внутри 3D Slicer'a можно представить фрагментацию, как модель (которая по сути будет потом экспортироваться для 3D формата файлов).

Но проблемы, которые появляются при его использовании:

- Хоть мы и будем иметь представление одновременно, как и нормалей, так и точек, но эти точки представлены в системе координат RAS (по сути представление в мировых координатах), которые никак не связаны с системой IJK (по сути координатная система изображения). [1)]
- То есть невозможно было применить какую-то функцию для определения, что точка принадлежит какому-то определённому слайсу в системе IJK. Конечно есть возможность перевода из одной системы координат в другую, но тогда в каких-то слайсах появляются, либо слишком много точек, либо в каких-то местах уже их не хватает, из-за чего может теряться точность определения стенки, ведь нарушается логика непрерывности данных. Так ещё в RAS все координаты всех вершин были разные;
- Также не понятно, как перебрать эти точки в порядке их контура. Только переводить эти точки в как бы бинарную маску изображения, но из-за пункта выше могут появляться разрывы, которые будут сбивать алгоритмы по нахождению контура.

Способ через фрагментацию, как бинарное представление

Также 3D Slicer даёт возможность получить представление фрагментации в виде набор вокселей.

Поэтому мной был произведён такой алгоритм:

- Перебираем слайсы по одной оси (в качестве данной оси взята аксиальная проекция, так как на ней лучше всего понятен разрыв кости (но в алгоритме можно подменить на любую проекцию);
- Находим контуры с использованием библиотеки OpenCV и функции findContours с параметрами cv2.RETR_LIST и cv2.CHAIN_APPROX_NONE, так как нам не важна иерархия контуров, а также важно, чтобы были выданы все точки контура, а не отбрасывались из-за их аппроксимации;
- Далее нам стоит найти нормали. Я решил сделать итерацию по точкам, где нормаль находится, как векторное произведение вектора направленного перпендикулярно проекции, которую мы выбрали, и следующей точке в контуре - тем самым мы находим однозначную нормаль;
- Но если у нас вокселей слишком мало, то нормали будут слишком резко изменяться относительно друг-друга и не будут отражать изменяющийся рельеф пазухи. Тем самым было решено, что нужно аппроксимировать нормали. То есть для начала мы считаем нормали, как описано в предыдущем пункте, а затем применяем усреднение по своей нормали и нормалям ближайших соседей и получаем более гладкое изменение нормалей. Тут главное делать это в новом массиве, а иначе в наши вычисления уже могут попасть усреднённые нормали. В качестве кол-ва соседей по каждой из сторон было взято 2 (то есть всего 5 нормалей участвует в аппроксимации) и это даёт очень хороший результат;
- Также стоит отбрасывать контуры, которые имеют меньше, чем необходимое кол-во нормалей, потому что, как минимум, это какие-то выбросы и не являются теми частями пазух, что можно нормально проанализировать;
- Далее мы для каждого контура проходимся по каждой его точке и пускаем линию вдоль нормали с выбранной эмпирическим путём длинной. Для итерации по этой линии я попользовался данным алгоритмом - ссылка.
- В качестве функции был взят просто максимум на отрезке. Потому что использовать как-либо, например, расстояние, не очень стоит, ведь сегментации из-за природы снимков может иметь разное расстояние до стенки, но при этом стенка всё ещё есть;
- Далее же мы применяем для каждого контура алгоритм Change Point Detection, который подробнее описан в разделе ниже;
- И визуализируем точки разрыва с помощью средства 3D Slicer'a под названием Markups.

Попытки в другие способы (неудачные решения)

- Контуры также можно было бы найти по полярному углу или применив алгоритм ConvexHull, но самая главная проблема, что у нас пазуха не является выпуклой фигурой. Она может быть невыпуклой и в таком случае логика сломается;
- В случае использования 3D модели, в качестве эвристики нахождения контура можно было бы взять евклидово расстояние, но из-за проблем, что у нас контур может

обрываться или иметь слишком много точек, то могут быть множество ошибок. Были попробованы эвристики, где выбор следующей точки был основаны на скалярном умножении, но в целом это не придавало каких-то хороших результатов;
- Были попытки посчитать нормаль относительно расположения двух точек, но там сразу появлялась проблема того, что если направления до этих точек прямо пропорциональны относительно друг-друга, то нормаль превращалась бы в 0. Да и появлялась проблема в направлении нормали, ведь центр масс, как ориентир, нельзя использовать, так как фигура невыпуклая.

Change Point Detection

Данный алгоритм идеально подходит для определения точек разрыва в одномерных данных, таких как временные ряды или последовательности.

Сравнение алгоритмов и их тестирование на наших данных

В качестве наглядного сравнения был использован jupiter notebook.

> Сюда бы в идеале сделать таблицу со сравнением.

Инструменты для визуального тестирования предположений

В качестве оценки всех выше попробованных мною идей был использован инструмент 3D Slicer'a под названием Markups с помощью которого я рисовал кривые для контура, а также рисовал отрезки с помощью которой находилась нужна интенсивность, чтобы понять насколько удовлетворительными получаются нормали и сами длин отрезок.

> Сюда стоит впихнуть скриншот с контуром и отрезками.

Возможности для улучшения

- Можно для контура добавлять промежуточные точки, чтобы увеличить точность, если какие-то нормали интерполируются не так;
- Нужно в идеале показывать не просто точки разрыва, а показывать кривую по которой происходит этот разрыв.

[1)]
Координатные системы - https://slicer.readthedocs.io/en/latest/user_guide/coordinate_systems.html

From:
http://wiki.osll.ru/ - **Open Source & Linux Lab**

Permanent link:
http://wiki.osll.ru/doku.php/projects:otolaryngologist:sinus_bone_continuity?rev=1714996423

Last update: **2024/05/06 14:53**