Investigation of feature recognition in clinical CT images as a step towards adaptive radiotherapy

Alix MOAWAD

Supervisors:
Kenneth Long, Ruth McLauchlan

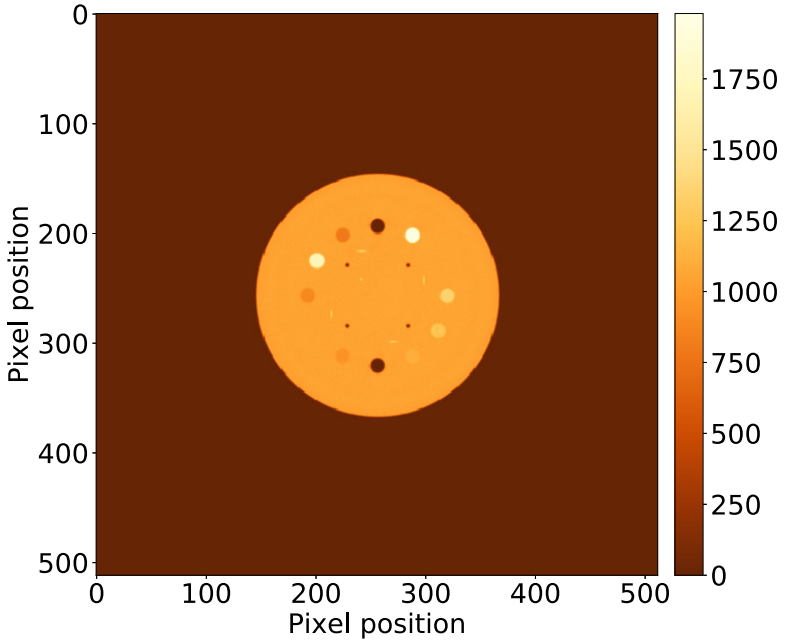
March 18, 2020

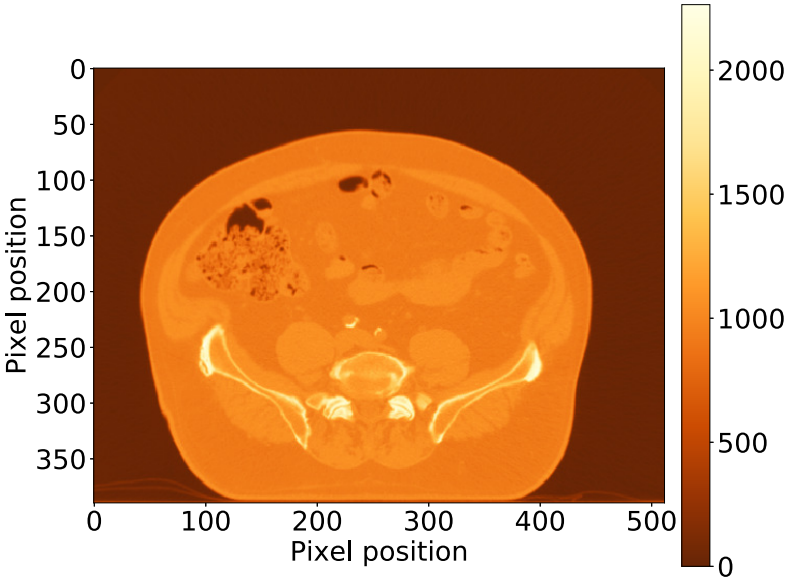
Imperial College London

Study materials

CT-scans of two different study materials will be examined :

- A phantom (3D image cut into 88 slices)
- A 'real-life' subject (3D image cut into 81 slices)





Outline

Two main questions:
→ How to **identify** all the borders within a 3D image?
→ How to **distinguish** its different regions?

1. **Section 1**
   - Laplacian Method
   - Canny Edge Detector
   - Otsu's Method

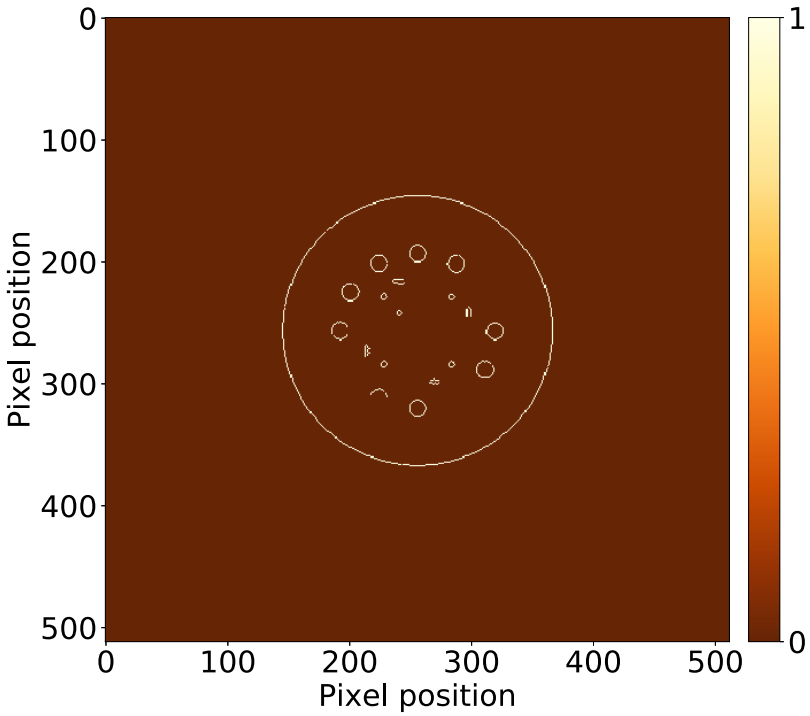
2. **Section 2**
   - A recursive algorithm
   - An iterative algorithm

Outline

Laplacian Method

The maximum intensity variations correspond to the **zero crossings** of the Laplacian operator. The method used here is based on three steps:

- **Laplacian** computed at each pixel using central finite differences
- Search for the **zero crossings** of the Laplacian
- **Threshold method** by percentage: less than 15% of the brightest contours displayed



Canny Edge Detector

Canny Edge Detector is an image processing method which can be split into 4 steps:

- Filter out any noise with a **Gaussian Filter**
- Find the **intensity gradient** of the image
- Apply a **non-maximum suppression** algorithm
- Track the edges by **hysteresis**

Otsu's Method

- The Otsu's Method is a global thresholding technique used on the **gradient magnitude** image to generate a **high threshold**.
- A **low threshold** is then found by taking a percentage of the high threshold (typically 50%).
- We ultimately used a **edge tracking by hysteresis** method to outline the borders after applying a **non-maximum suppression** algorithm.

Otsu's Method

Canny Edge Detector and Otsu's Method: a comparaison

Observation: Canny Edge Detector displays absolutely all contours within an image while the Otsu's method only keeps global contours. The thresholds of the Canny Edge Detector are defined manually whereas they are defined statistically in Otsu's method.

$\rightarrow$ Which method should we use?

Outline

A recursive algorithm

Observation : A stackoverflow prevented us to apply this algorithm to entire 3D scans. Indeed, Python's stack was exhausted, due to deep recursion. $\rightarrow$ How to resolve this situation?

An iterative algorithm

In a list E are stored the coordinates (x,y,z) of every pixel considered as belonging to a border and the color assigned to it.

- A color is attributed to the first pixel (x_1,y_1,z_1) on the list E.
- We search for pixels of coordinates (x,y,z) in the list E such that :

$$\begin{cases} x_1 - 1 \leq x \leq x_1 + 1 \\ y_1 - 1 \leq y \leq y_1 + 1 \\ z_1 - 1 \leq z \leq z_1 + 1 \end{cases} \tag{1}$$

- Two possibilities:
  - If pixels with the above conditions are found, then they are given the same color and we apply the same process for those new pixels.
  - If not, the second pixel (x_2,y_2,z_2) in the list E is studied with a new color and the same process is applied.

Conclusion and further work