Segmentation and 3D Surface Reconstruction of Bones from CT Scans

Feroze Naina
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA - 15213
fdheenmo@andrew.cmu.edu

Onder Erin
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA - 15213
ondere@andrew.cmu.edu

Abstract

Cross-sectional image views of a patient's internal organs can be obtained from specialized medical imaging devices such as Magnetic Resonance Imaging (MRI) and Computerized Tomography (CT) scans. These images can be used to generate 3D models for use by doctors to diagnose medical conditions such as fractures and tissue damage. Our goal is to understand various methods of segmenting out bone tissues from the image slices and developing a 3D model of the segmented bone. This would be useful for visualizing the surface of a single bone among multiple bones in a CT scan. In our project, we used CT scans of the Iliac bone and the two forearm bones - Radius and Ulna.

1. Introduction

Computed Tomography (CT) scan uses a series of X-ray images of the human body taken from different angles and applies computer processing to create cross-sectional image slices. CT scans provide more detailed information than conventional X-rays and are used for diagnostic and therapeutic purposes. It's has many advantages over traditional X-rays. Figure 1 shows a typical forearm CT scan. The left bone in the CT scan is the Radius and the right bone is Ulna as depicted below.

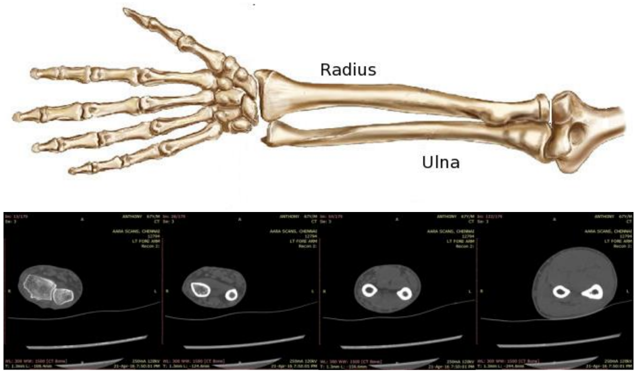

Figure 1: Forearm CT scans

CT scans of the human body typically have bimodal intensity distribution – bones and musculature. In the image below, the initial peak with high pixel intensity are the bone tissues. The second peak with lower intensity values are muscle tissues. The change in contrast is used to detect bone-tissue by implementing global thresholding after normalization: Figure 2 shows a sample CT scan histogram with bimodal distribution.

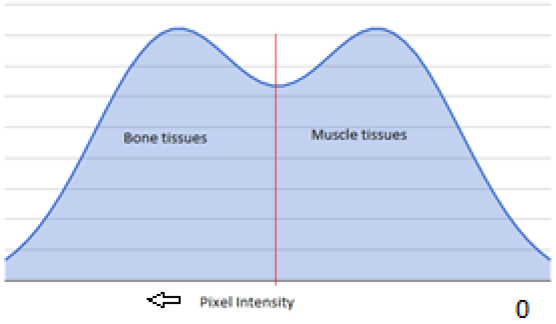

Figure 2: Histogram of CT scans

Common issues in CT scan images are the lack of consistent contrast difference across image slices. This is due to the change relative density of bones and muscle across imaging regions.

2. Segmentation Techniques

Common medical segmentation techniques based on gray levels are:

1. Amplitude segmentation
2. Edge based
3. Region growing
4. Model based segmentation

In this report, we are focusing primarily on amplitude, edge based and region growing segmentation methods.

2.1. Global Thresholding

The intensity value of CT scans is a linear function of the material density for a given location which can be represented as

$$\boldsymbol{I_i}(x, y) = C * \boldsymbol{\sigma_i}(x, y) \tag{1}$$

where I_i is 2D image intensity matrix and σ_i is the 2D density function matrix of the tissue for the i^{th} image slice. Because bone density is significantly higher than any other tissue in a human body, the intensity value of a bone tissue is significantly larger than rest of the tissue intensities. This distinct feature of bones is used to differentiate the bone tissue with the rest of the tissues for segmentation of bones by using a global thresholding. The intensity threshold is chosen as a function of each slice image's maximum intensity which can be represented as

$$\boldsymbol{I_i}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 1 & if \quad \boldsymbol{I_i}(\boldsymbol{x}, \boldsymbol{y}) \geq th \\ 0 & if \quad \boldsymbol{I_i}(\boldsymbol{x}, \boldsymbol{y}) < th \end{cases} \tag{2}$$

where $th = 0.85 * \max \boldsymbol{I_i}(\boldsymbol{x}, \boldsymbol{y})$.

The resulting image is a binary image showing the bone structure. The main advantage of this technique is being very fast and easy to implement. Especially when there is a set of slice images, speed is an important issue for CT scans. However, because of constant 85% thresholding, some of the slice images fail under this constant thresholding value due to the variation in contrast at each image.

2.2. Adaptive Thresholding

In adaptive thresholding, the image is divided into smaller windows and each window is segmented separately. The reason for this is that smaller image regions are more likely to have approximately uniform illumination, thus being more suitable for thresholding. The size of the neighborhood has to be large enough to cover sufficient foreground and background pixels, otherwise a poor threshold is chosen. On the other hand, choosing regions which are too large can violate the assumption of approximately uniform illumination. In our implementation, we use a windows size of 11 pixels x 11 pixels.

An approach to finding the local threshold is to statistically examine the intensity values of the local neighborhood of each pixel. The statistic which is most appropriate depends largely on the input image. Simple and fast functions include the mean of the local intensity distribution.

Like global thresholding, adaptive thresholding is used to separate desirable foreground image objects from the background based on the difference in pixel intensities of each region. Global thresholding uses a fixed threshold for all pixels in the image and therefore works only if the intensity histogram of the input image contains neatly separated peaks corresponding to the desired subject(s) and background(s). Hence, it cannot deal with images containing, for example, a strong illumination gradient which may occur in medical imaging data.

Local adaptive thresholding, on the other hand, selects an individual threshold for each pixel based on the range of intensity values in its local neighborhood. This allows for thresholding of an image whose global intensity histogram doesn't contain distinctive peaks.

2.2.1 Otsu's method

In computer vision and image processing, Otsu's method, named after Nobuyuki Otsu, is used to automatically perform clustering-based image thresholding, or, the reduction of a gray level image to a binary image. The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal, or equivalently (because the sum of pairwise squared distances is constant), so that their inter-class variance is maximal. Consequently, Otsu's method is roughly a one-dimensional, discrete analog of Fisher's Discriminant Analysis.

In Otsu's method we exhaustively search for the threshold that minimizes the intra-class variance (the variance within the class), defined as a weighted sum of variances of the two classes:

$$\sigma_w^t(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\,\sigma_1^2(t) \tag{3}$$

Weights $\omega_{0,1}$ are the probabilities of the two classes separated by a threshold t and $\sigma_{0,1}^2$ are variances of these two classes.

The class probability $\omega_{0,1}(t)$ is computed from the L histograms:

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \tag{4}$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \tag{5}$$

Otsu shows that minimizing the intra-class variance is the same as maximizing inter-class variance:

$$\begin{aligned} \sigma_b^2(t) &= \sigma^2 - \sigma_w^2(t) \\ &= \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \\ &= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \end{aligned} \tag{6}$$

which is expressed in terms of class probabilities ω and class means μ.

while the class mean $\mu_{\{0,1,T\}}(t)$ is:

$$\mu_0(t) = \sum_{i=0}^{t-1} ip(i)/\omega_0 \tag{7}$$

$$\mu_1(t) = \sum_{i=t}^{L-1} ip(i)/\omega_1 \quad (8)$$

$$\mu_T(t) = \sum_{i=0}^{L-1} ip(i) \quad (9)$$

The following relations can be easily verified:

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T \quad (10)$$

$$\omega_0 + \omega_1 = 1 \quad (11)$$

The class probabilities and class means can be computed iteratively. This idea yields an effective algorithm.

Algorithm:

1. Compute histogram and probabilities of each intensity level
2. Set up initial $\omega_i(0)$ and $\mu_i(0)$
3. Step through all possible thresholds t = 1 … maximum intensity
4. Update ω_i and μ_i
5. Compute $\sigma_b^2\ (t)$
6. Desired threshold corresponds to the maximum $\sigma_b^2\ (t)$.

2.3. Watershed segmentation

Watershed transform finds "catchment basins" and "watershed ridge lines" in an image by treating it as a surface where light pixels are high and dark pixels are low. It produces closed contours: each minimum or each marker corresponds to one region. Flooding a topographic surface fills some minima, and the watershed of the flooded surface has fewer catchment basins. The catchment basins of successive floodings form a hierarchical segmentation. It is possible to flood a surface so as to impose minima at some predetermined places: this leads to marker based segmentation. Marker based watershed segmentation is performed on single CT slice and propagated to other images.

In the study of image processing a watershed of a gray scale image is analogous to the notion of a catchment basin of a height map. In short, a drop of water following the gradient of an image flows along a path to finally reach a local minimum. Intuitively, the watershed of a relief corresponds to the limits of the adjacent catchment basins of the drops of water.

There are different technical definitions of a watershed. In graphs, watershed lines may be defined on the nodes, on the edges, or hybrid lines on both nodes and edges. Watersheds may also be defined in the continuous domain. There are also many different algorithms to compute watersheds. Watershed algorithm is used in image processing primarily for segmentation purposes.

Different approaches may be employed to use the watershed principle for image segmentation.

Local minima of the gradient of the image may be chosen as markers, in this case an over-segmentation is produced and a second step involves region merging.

Marker based watershed transformation make use of specific marker positions which have been either explicitly defined by the user or determined automatically with morphological operators or other ways.

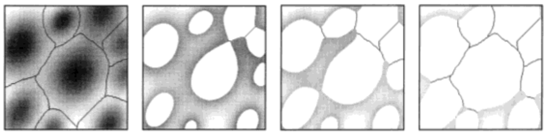

Figure 3: Watershed region growing

2.3.1 Meyer's flooding algorithm

One of the most common watershed algorithms was introduced by F. Meyer in the early 90's. This method is implemented in MATLAB.

1. A set of markers, pixels where the flooding shall start, are chosen. Each is given a different label.
2. The neighboring pixels of each marked area are inserted into a priority queue with a priority level corresponding to the gradient magnitude of the pixel.
3. The pixel with the lowest priority level is extracted from the priority queue. If the neighbors of the extracted pixel that have already been labeled all have the same label, then the pixel is labeled with their label. All non-marked neighbors that are not yet in the priority queue are put into the priority queue.
4. Redo step 3 until the priority queue is empty.

The algorithm works on a gray scale image. During the successive flooding of the grey value relief, watersheds with adjacent catchment basins are constructed. This flooding process is performed on the gradient image, i.e. the basins should emerge along the edges. Normally this will lead to an over-segmentation of the image, especially for noisy image material, e.g. medical CT data. Either the image must be pre-processed or the regions must be merged on the basis of a similarity criterion afterwards. Figure 4 shows the radius bone segmented out using watershed propagation technique.

A hierarchical watershed transformation converts the result into a graph display (i.e. the neighbor relationships of the segmented regions are determined) and applies further watershed transformations recursively. This can be used to create labels such that the bone marrow is segmented separately inside the bone, and the muscle tissues can be segmented separately inside a limb.

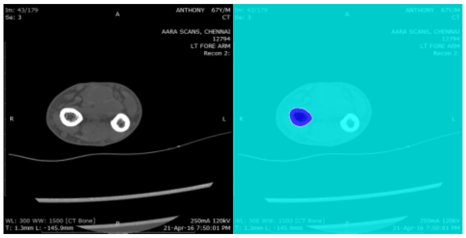

Figure 4: Watershed segmentation of radius bone using markers

3. Segmentation Propagation

Reasonable amount of CT scans include multiple bones instead of a single bone. In this case, it is important to segment a single bone out of the overall bones. Therefore, on top of the current segmentation techniques, segmentation propagation technique needs to be added. Additionally, because the bone location varies in each slice due to the orientation of the body part during the scans, segmentation algorithm has to be adaptive to these changes. The best solution we propose is propagating the user provided segmentation window along the CT scan image set.

This technique requires a user input to track a certain bone and it tracks the same bone at a constant window during the whole CT scan image set.

The user provides a polygon to encircle desired bone area to be tracked and segmented out. The algorithm uses the previous image's selected window as a mask for the next slice image and finds the best matching window location on the next image by using SSD algorithm and updates the mask and the location of the mask. The SSD algorithm is given in the equation below:

$$\sum_{(i,j)\in W}\left(I_{mask}(i,j) - I_{original}(x+i,\ y+j)\right)^2 \quad (12)$$

The update of the mask at each slice image is

$$I_{mask_{k+1}} = I\left(\min_{x,y}\sum_{(i,j)\in W}\left(I_{mask_k}(i,j) - I_{original}(x+i,\ y+j)\right)^2\right) \quad (13)$$

where k is the slice index.

SSD algorithm provides very high accuracy to track the desired bone tissue with more than 87% accuracy with thresholded CT scan images. The main reason of this high accuracy is having very distinct features with high intensity values of the thresholded CT scan images.

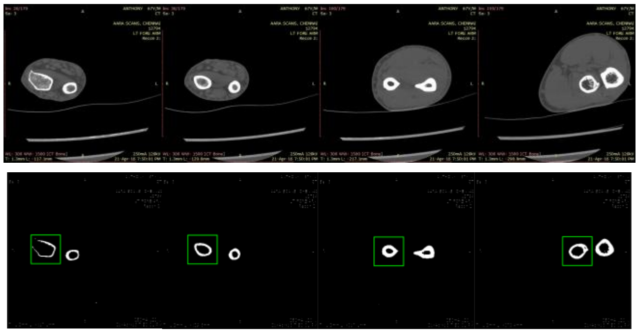

Figure 5: Green box indicates the segmented region propagation for the Radius bone.

4. Acquiring Outline from the Segmented Images

In order to have 3D point cloud generation, the outline of the segmented bone images need to be acquired. In order to have this, canny edge detection algorithm is implemented on the image to acquire outline and the internal features. The fudge factor is chosen as 1.5. To make sure the whole bone presents a closed shape, the bone image dilated by using 5x5 isotropic dilation window. Then, flood fill algorithm is implemented to have the enclosed bone image as one solid image. To get the outline, the flood-filled image is eroded with 5x5 erosion window and the resulting image is subtracted from the original flood-filled image to achieve the only outline of the bone. The same technique is applied to the each image slice in a CT scan set. This process is described in the figure below:

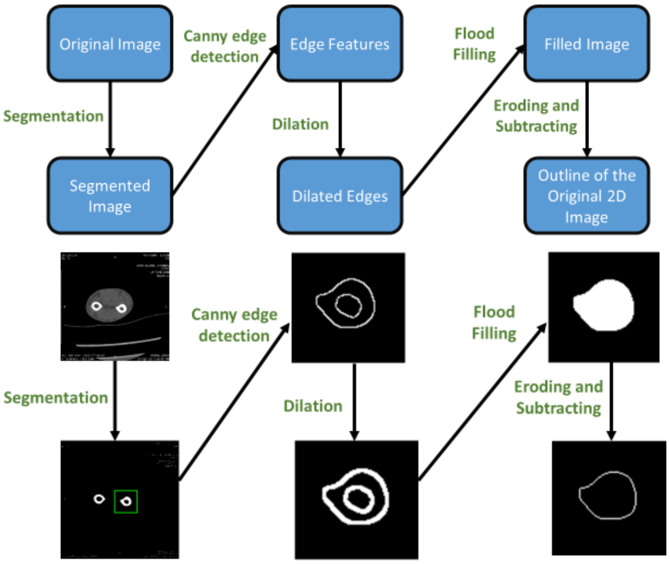

Figure 6: Schematics of the main steps for segmentation and getting 2D outline point cloud of a forearm CT scan

5. 3D Visualization

The pointcloud is generated by stacking the x and y co-ordinates of the outline in each CT slice along the voxel thickness size of 1.5mm. This is then converted to an ASCII PCD format file to allow us to easily visualize the desired segmented bone in the PCL point cloud viewer.

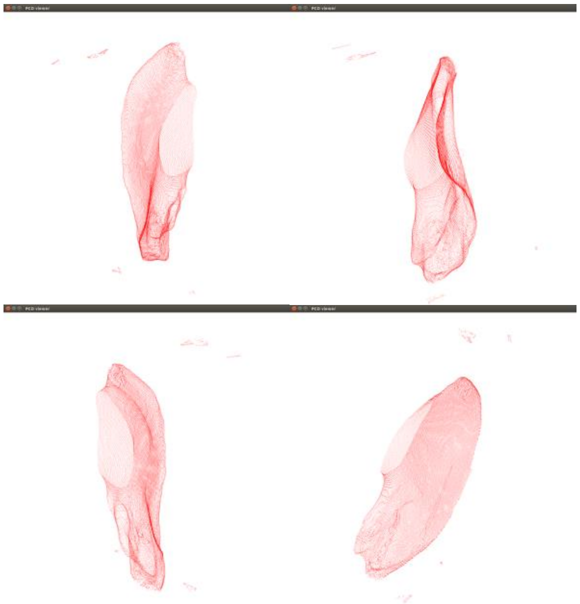

Figure 7: Iliac bone visualized in PCL viewer

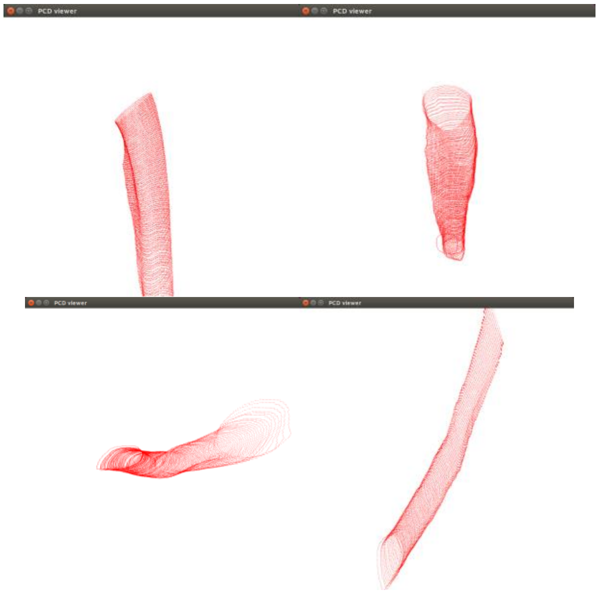

Figure 8: Radius (above) and Ulna (below) visualized in PCL viewer

6. Results

A constant thresholding value to apply every image is not the best approach practically because every image has different intensity variation. Therefore, global thresholding was not sufficient enough. Also, by applying global thresholding, some information is lost which might result in losing some bone features. Failure examples of global thresholding algorithm on an iliac bone is given below.

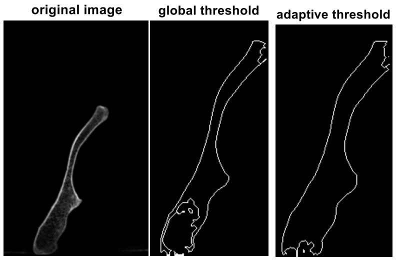

Figure 9: Comparison between global thresholding and adaptive thresholding. The image in the middle is an example case for a failure for global thresholding algorithm. Adaptive thresholding is more robust for segmentation as third image presents.

Acquiring outline from the segmented image uses dilation and erosion algorithms. Due to the algorithms, the original is prone to have modified. In order to keep the modification low, the window sizes are chosen as small as possible as 5x5 to 11x11. The results show that 2D outlines provides very well fitting with the original bone image as long as the window sizes are kept below 11x11**.**

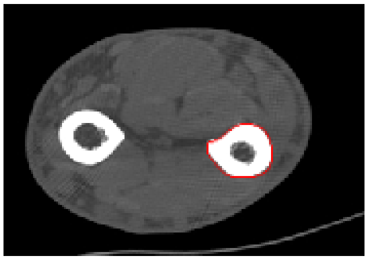

Figure 10: Red contour shows the extracted 2D outline point cloud. The accuracy of the method can be seen from the image on the right which is a comparative image with the original CT scan data.

Thus we have compared various medical image segmentation techniques and have analyzed the advantages and disadvantages of each. We have also successfully segmented out a single bone from the forearm and visualized it in the 3D.

7. References

[1] S. Beucher and F. Meyer, “The morphological approach to segmentation: The watershed transform”, in Mathematical Morphology in Image Processing, vol. 12, , pp. 433–481, 1993.

[2] Neeraj Sharma and Lalit M Aggarwal, “Automated medical image segmentation techniques”, Journal of Medical Physics, Vol. 35, No. 1, pp. 3-14, 2010.

[3] Rein van den Boomgaard and Richard van Balen, “Methods for fast morphological image transforms using bitmapped binary images”, CVGIP: Graph. Models Image Process, Vol. 54, No. 3, pp. 252-258, 1992.

[4] Serge Beucher, “The Watershed Transformation”, http://cmm.ensmp.fr/~beucher/wtshed.html

[5] Wikipedia, “Watershed (image processing)”, https://en.wikipedia.org/wiki/Watershed_(image_processing)