



Edge Detection and Segmentation of Multiple Contours from CT Scan Images

Amba D. Bhatt¹, Ujjaval Gupta², Vishal Wagholikar³ and Uday V. Pise⁴

¹Motilal Nehru National Institute of Technology, Allahabad, India, amba_bhatt@yahoo.com

²Motilal Nehru National Institute of Technology, Allahabad, India, ujjaval88@gmail.com

³Motilal Nehru National Institute of Technology, Allahabad, India, vwagholikar@yahoo.in

⁴Motilal Nehru National Institute of Technology, Allahabad, India, udayvpise@rediffmail.com

ABSTRACT

An image consists of a number of objects. Some applications require us to identify particular objects and isolate them from the image. This target is achieved by image segmentation. Segmentation of multiple objects is a challenging area of scientific research. The objective of image segmentation is to identify the desired objects, distinguish and separate them from the image consisting of several other objects and background. In this paper, we have used edge detection as a tool to carry out image segmentation. The process involves several steps like image preprocessing, Edge detection based on Canny Edge Detection method, boundary tracing using Moore neighborhood tracing algorithm and use of point in polygon algorithm for image segmentation. We have proposed a novel algorithm to segment multiple objects from an image. The method produces good results with appreciable accuracy.

Keywords: image segmentation, edge detection, multiple contours.

DOI: 10.3722/cadaps.2012.501-516

1 INTRODUCTION

Image segmentation is the process of partitioning the image into multiple regions or sets of pixels [23]. Its purpose is to distinguish objects from background. Basically, it is used to locate objects and boundaries. Image segmentation eliminates unnecessary pixels from the image and hence it simplifies the processing and analysis of an image. Image segmentation is a subfield of image analysis. It simplifies the analysis of images by drastically reducing the volume of data to be processed by retaining only the essential data. Thus, image segmentation results in subdivision of image into set of regions that collectively cover the entire image or a set of contours extracted from the image. These

Computer-Aided Design & Applications, 9(4), 2012, 501-516

© 2012 CAD Solutions, LLC, <http://www.cadanda.com>

regions are segmented on the basis of common characteristics like grey level intensity or colors. Image segmentation methods are broadly classified into seven groups (1) Edge based (2) Region growing (3) Histogram thresholding (4) Clustering (Fuzzy and Hard) (5) Physical model based (6) Fuzzy approaches (7) Neural Network and Genetic Algorithm based approaches.

Edge detectors have been an essential part of problems based on computer graphics and image processing. It is widely used for image segmentation. Region boundaries show a sharp variation in pixel characteristics. This property can be used to detect region boundaries. Canny J. [3] proposed an algorithm for edge detection in 1986. There are several other edge detection algorithms like Prewitt, Sobel Operator and Hough transform but Canny Edge Detection gives significantly better results [25], [10], [7] due to three performance criteria (1) Good detection (2) Good localization and (3) Only one response to a single edge. Olaya Grove et al.[8] presented a paper on use of NURBS for contour fitting of CT scan images. They have also used Canny edge detector. Detection of a closed boundary around the region of interest is the first step in image segmentation. It is difficult to form a closed contour around the object since the edges are disconnected due to noise persisting in the image. The image is pre-processed to remove such discrepancies. Pre-processing includes steps like normalization, increasing contrast, boundary modification and filtering that facilitate continuous edges and closed contours.

Moore neighborhood boundary tracing technique is applied to trace a closed boundary i.e. contour along the detected edge [9]. Moore neighborhood of a pixel P consists of 8 cells surrounding the pixel P (Fig. 1) that share either a common edge or a vertex. The pixel P lies on the detected edge and the algorithm extracts the contour by going along the pattern in a clockwise direction.

Thus edge detection along with Moore neighborhood tracing algorithm is used to segment out the region of interest from the image under analysis. The original image after pre-processing is subjected to Canny edge detection. The detected edge is traced by Moore neighborhood algorithm. In image segmentation, the points that lie inside the traced contour are detected by point in polygon algorithm and the pixel intensities of these points are extracted. The extracted data is displayed to show the results of image segmentation.

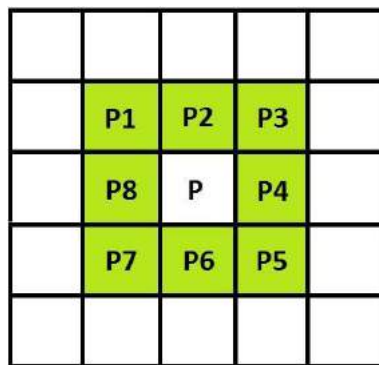


Fig. 1: Moore neighborhood of a pixel P.

Edge detection and image segmentation are important tools in image processing. These techniques have been used in applications like:

A. Security: Iris scans Fingerprint recognition.

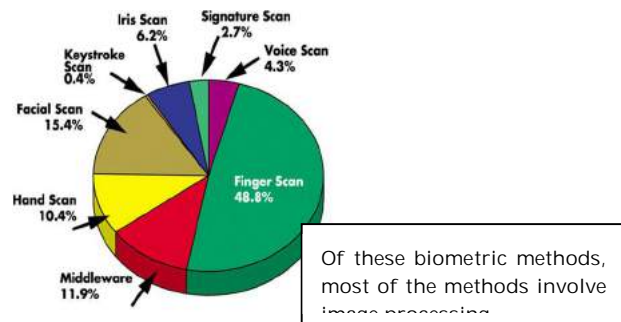


Fig. 2: Use of biometrics [1].

- B. Satellite imagery: Extraction of coastlines, road network extraction.
- C. Medical imaging: Generation of a 3d model of brain or human body from CT scan or MRI data, Digital X- Rays.
- D. Multimedia applications: Color image and video transmission over the internet, digital broadcasting, interactive TV, video-conferencing, distance education [12].
- E. Image compression: For removal of unnecessary data from images.

Hence detailed research is being carried out in the field of image processing to obtain accurate results by edge detection and image segmentation.

2 ALGORITHM

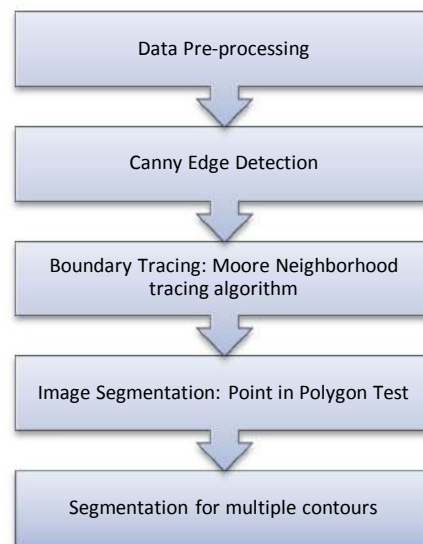


Fig. 3: Algorithm chart.

2.1 Data Pre-Processing:

Raw data (input data) preprocessing commonly comprises a series of sequential operations that enhance data and transform it into a format that will be easily, effectively and efficiently processed with more accuracy. The steps involved are

2.1.1 Conversion from RGB to grayscale image

Conversion of a color image to grayscale is not unique. One of the common methods is eliminating hue and saturation and retaining the luminance of the image at the same time. RGB to gray scale conversion facilitates easy processing based on grayscale values of pixels.

2.1.2 Normalization

Normalization is the process of changing the range of pixel intensity values [26]. For a grayscale image, the pixel intensities vary from 0 to 255. Normalization changes the pixel intensity values to a range that is more common and easy for estimation/ manipulation/ understanding. In this case the range of normalization is 0 to 1.

2.1.3 Increase contrast/ reducing brightness

The brightness of original image (Fig. 4(a)) is reduced to decrease the effect of noise and facilitate easy detection of edges during Canny edge detection.

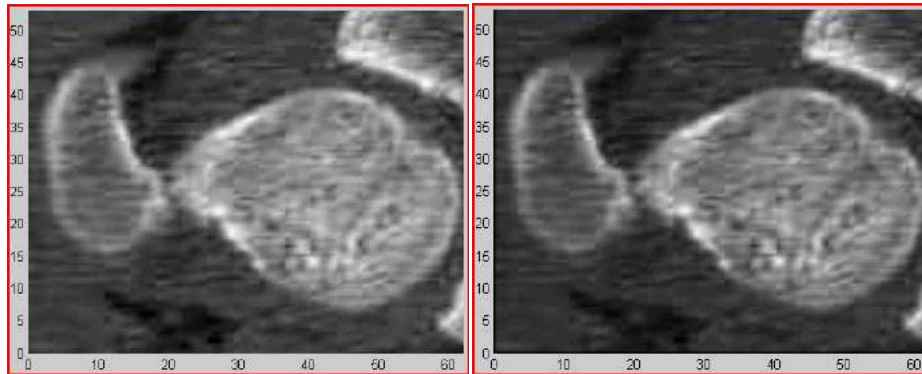


Fig. 4: Referred from left to right: (a) Original image and (b) after reducing brightness.

2.1.4 Boundary modification

Intensity values of points at the image boundary are modified so as to enable formation of closed contours when the object lies on the boundary of the image.

Boundary modification:

$f(x,y)$ \rightarrow pixel intensity at (x,y) co-ordinate

x_{min}/x_{max} \rightarrow min/max coordinate in x direction

y_{min}/y_{max} \rightarrow min/max coordinate in y direction

$$B_1 = \{f(x,y) : x \in \bar{X} = x_{\min} \& x = x_{\max}\} = 0 \quad \&$$

$$B_2 = \{f(x,y) : y \in \bar{Y} = y_{\min} \& y = y_{\max}\} = 0 \quad \text{i.e. pixel intensity of background}$$

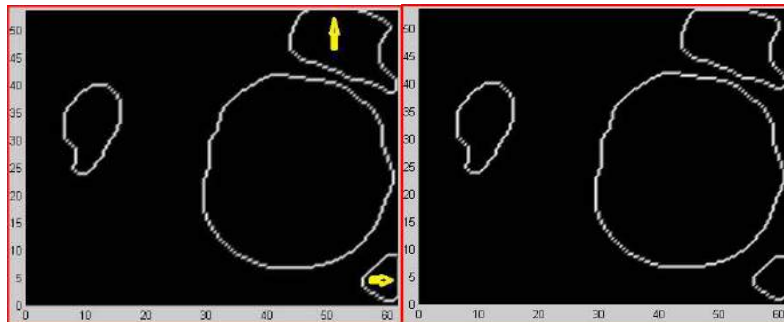


Fig. 5: Referred from left to right: (a) Before boundary modification and (b) after boundary modification.

If boundary is not modified, there will be discontinuous edges (i.e. open contours) (fig. 5(a)) that will result in errors at the boundary tracing stage, since closed contours are essential for contour tracing. We can also see the boundary modification does not result in significant loss of relevant information (fig. 5(b)).

2.1.5 Threshold

Threshold [15],[22] is used to segment an image by setting all pixels whose intensity values are above a threshold to a foreground value (i.e. the object) and all the remaining pixels to a background value. It basically converts intensity data into a logical array. Here the object pixel is given an intensity value of 1 while background pixel is given an intensity value of 0. An important parameter is the choice of Threshold value (T), on the basis of which the operation is done.

Threshold function

$$F1 = \{f(x,y) : f(x,y) > T\} = 1 \quad (\text{object})$$

$$F2 = \{f(x,y) : f(x,y) \leq T\} = 0 \quad (\text{background})$$

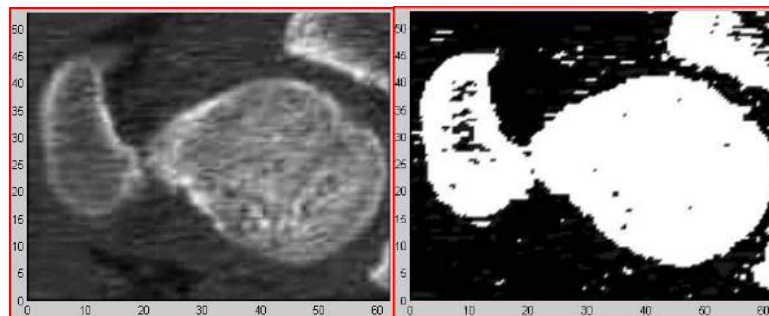


Fig. 6: Referred from left to right: (a) Before Threshold and (b) after Threshold.

The Threshold operation has converted the grayscale image (fig. 6(a)) to a logical array (Binary image) (fig. 6(b)) which would fetch precise results at the edge detection stage.

2.1.6 Median filtering

Filtering is the process of removing unwanted signal/ component from a signal/data. There are various filtering techniques like Median filtering, Averaging filter, Gaussian lowpass filter, Laplacian filter, Prewitt filter, and Sobel filter. Their effectiveness was tested by applying each of the filters to CT images (Fig. 7). Median filter provided the best results for each image.

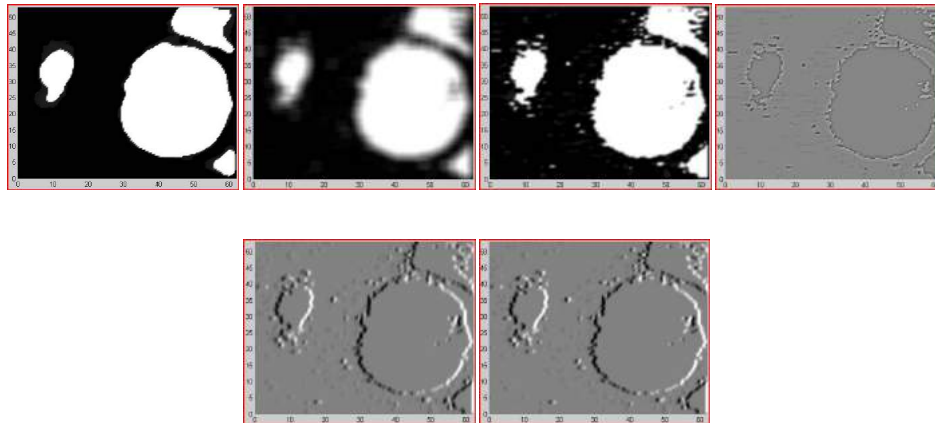


Fig. 7: Referred from left to right: (a) Median filtering, (b) Averaging filter, (c) Gaussian low-pass filter, (d) Laplacian filter, (e) Prewitt filter, and (f) Sobel filter.

Median filtering [11], [14] is a sort of non-linear filtering technique which is used to reduce Salt and Pepper noise (fig. 8(a)) in signal processing. A median filter is more effective when the goal is to simultaneously reduce noise and preserve edges.

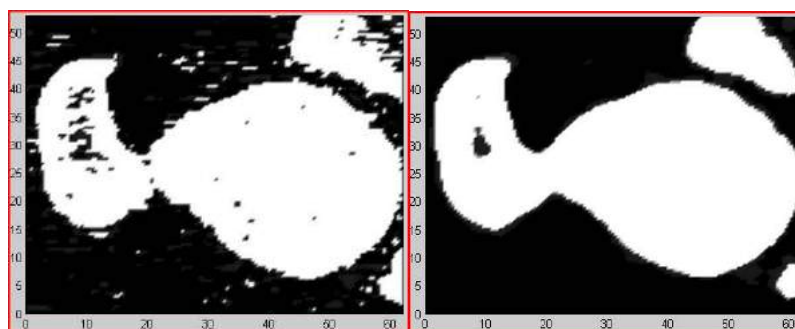


Fig. 8: Referred from left to right: (a) Before Median Filtering and (b) after Median Filtering.

The Median Filtering operation has successfully removed all the Salt and Pepper noise with preserved edges. The image is now ready for edge detection (fig. 8(b)).

2.2 Canny Edge Detection

After pre-processing, data is subjected to edge detection. Various edge detection techniques like Canny, Prewitt, Roberts and Sobel were applied to CT images and their results were analyzed (Fig. 9).

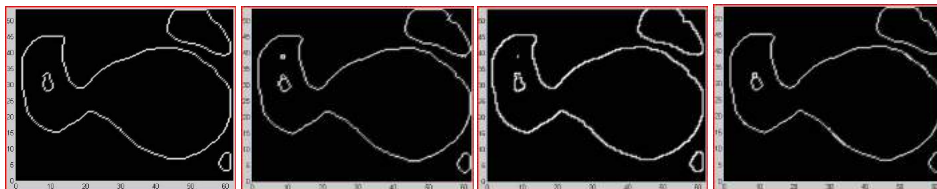


Fig. 9: Referred from left to right. Edge detection results: (a) Canny,(b) Prewitt,(c) Roberts, and(d) Sobel.

Canny edge detection algorithm [25] gives the best results, thus it is finally selected. The method takes pre-processed data/image as input (Fig. 10(a)) and gives connected curves that indicate the object boundaries as output. Algorithm uses the variation in pixel intensities as a sign of change in material properties to recognize object boundaries. The output is a logical set with pixel intensities of detected edges having a value 1 while intensities at all other pixels are 0 (Fig. 10(b)).

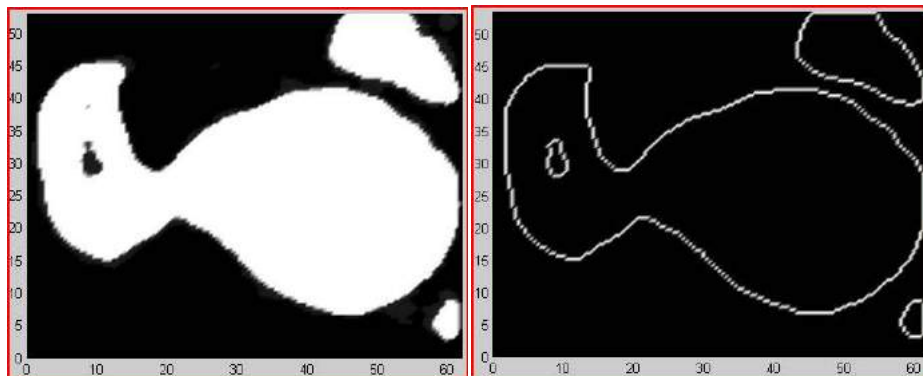


Fig. 10: Referred from left to right: (a) Input image and(b) Results after Canny edge detection.

Fairly accurate results are obtained from Canny edge detection algorithm.

2.3 Boundary Tracing

Tracing of detected edge is done using Moore neighborhood contour tracing algorithm [5]. Other tracing techniques include square tracing algorithm, radial sweep and Theo Pavlidis' algorithm but Moore neighborhood tracing produces the best results. Results of Canny edge detection consisting of connected curve are given as input and output is a set with co-ordinates of cells (pixels) that lie on the

Computer-Aided Design & Applications, 9(4), 2012, 501-516

© 2012 CAD Solutions, LLC, <http://www.cadanda.com>

curve. Once the pixels that lie on the edge (points having pixel intensity 1) around the object are detected, they are joined to form a closed contour (Fig. 11).

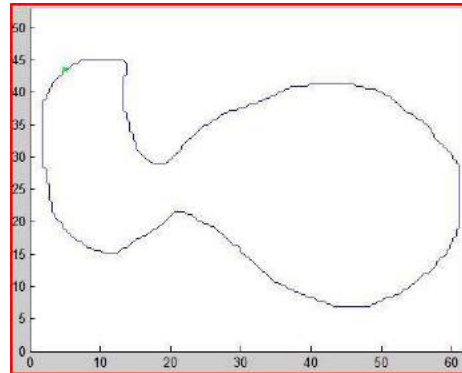


Fig. 11: Traced contour.

Special Case: object on the image boundary

In this case, contour formation is not possible without image pre-processing, since edge detection algorithm is not able to detect a closed contour. Boundary modification (image pre-processing step 5) causes a sudden variation in pixel intensities of the object at the image boundary resulting in their detection at the edge detection stage. Thus, closed contour is formed at the boundary tracing stage.

2.4 Image Segmentation

2.4.1 Segmentation of image with single contour

In this article, image segmentation [17] is based on edge detection and the type of segmentation is complete segmentation. Complete segmentation divides an image R into finite number (S) of regions (R1, R2, R3, ..., Rn.)

$$R = \{R_1, R_2, R_3, \dots, R_n\}$$

$$R_i \cap R_j = \emptyset \quad \text{where } i \neq j$$

Point in polygon algorithm [4],[9] is applied for image segmentation. In computational geometry, point in polygon algorithm is used for testing the location of a point with respect to a polygon i.e. find whether a point on a plane lies inside, outside or on the boundary of a given polygon. Contour traced by the Moore neighborhood tracing can be represented as a polygon with a very large number of small edges. This contour is the given polygon (Fig. 12).

Point in polygon algorithm:

Let (x,y) be the coordinates of the point that is under test relative to the polygon C

Let P be the point in polygon function

$P(x,y) = 1$, if (x,y) is inside or on the polygon C

$P(x,y) = 0$, if (x,y) is outside C

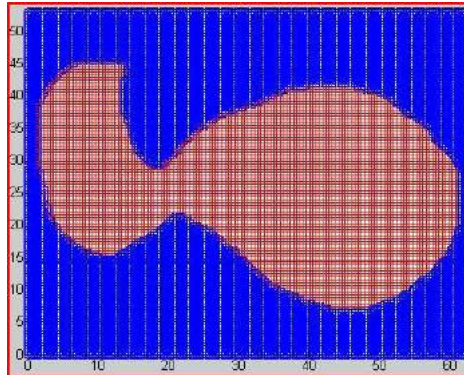


Fig. 12: Point in polygon test: Here, points inside the contour have been marked red while other points are marked blue.

Data Extraction Algorithm:

Let, $B(b_1, b_2, b_3 \dots b_n)$ be the set of points that lie on or inside the polygon

$s(x, y)$ be the point under test

- Initially B is empty
- Loop1 increment in x direction
 - Loop2 increment in y direction
 - Test all points s from top to bottom and left to right for point in polygon test
 - If $P(s) = 1$
 - Add $s(x,y)$ to set B
 - Endif
 - Endloop2
- Endloop1
- Extract the pixel intensities of points having coordinates given by set B from the original image/data
- Display the segmented data to see results

Output- Segmented image

Successfully segmented analytical part of the original image (Fig. 13(a)) has been shown after segmentation process (Fig. 13(b)).

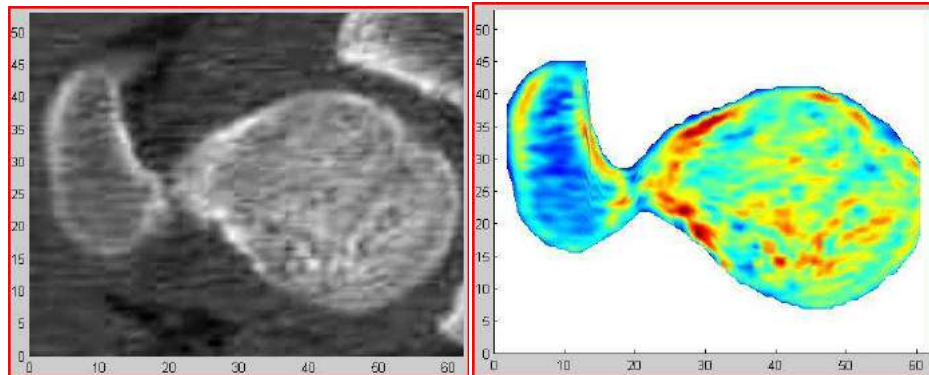


Fig. 13: Referred from left to right: (a) Original image and (b) Segmented image.

2.4.2 Segmentation of image containing multiple contours

The novelty of this article is that it can accurately do the segmentation of an image containing multiple objects. The following algorithm, when applied, not only does the segmentation of an image with multiple objects but also provides additional information like the number of contours which is essential for operations where this data is further used. Example is biomedical imaging, where this data can be further used to develop 3D models of body parts.

Algorithm of segmentation of image containing multiple objects:

Input: Image consisting of 2 or more objects

Let, N - Number of contours
 B - Set of points that lie on or inside contours

Algorithm:

Let, $B(b_1, b_2, b_3 \dots b_n)$ be the set of points that lie on or inside the polygon
 $s(x, y)$ be the point under test
 $edge$ = result from canny edge detection (1 if a point lies on the edge, else 0)
 tc = Traced contour
 N = Number of closed contours

- Initially $B = \emptyset, N = 0, edge, tc = \emptyset$

- *Image preprocessing*
- *Canny edge detection*
- *Loop1 increment in x direction*
 - Loop2 increment in y direction*
 - If edge(s) = 1*
 - Edge detected*
 - Break*
 - Endif*
 - Endloop2*
 - Trace contour and store in tc*
 - If tc ≠ ∅ and previous_tc ≠ tc*
 - Contour repeated*
 - Endif*
 - If non-repeated contour*
 - Loop3 increment in x direction*
 - Loop4 increment in y direction*
 - Apply point in polygon test*
 - If positive add s to B*
 - Endloop4*
 - Endloop3*
 - If first and last point of tc are same*
 - Closed contour detected*
 - N = N+1*
 - Endif*
- *Endloop1*
- *Extract the pixel intensities of points having coordinates given by set B from the original image/data.*
- *Display the segmented data to see results*

Output: Segmented image consisting of only the points/pixels that lie within the multiple contours

2.5 Results for Images with Multiple Contours

The algorithm of image segmentation was applied to the CT scan image of a proximal femur which consists of two contours. The large contour is of the head of the proximal femur, while the small contour is of greater trochanter.

2.5.1 Edge detection

Initially, the image is prepared for edge detection by pre-processing methods. Results of each step have been shown in Fig. 14. The original image is shown in Fig. 14 (a). It can be seen that the original image is very bright, thus its brightness is reduced to suppress the effect of noise and facilitate edge detection (Fig. 14 (b)). Thresholding operation is performed to convert the data into a logical array (Fig. 14 (c)). Noise is removed using median filtering. After pre-processing, image is subjected to edge detection (Fig. 14 (e)) and then contours are traced using Moore neighborhood boundary tracing technique (Fig. 14 (f)).

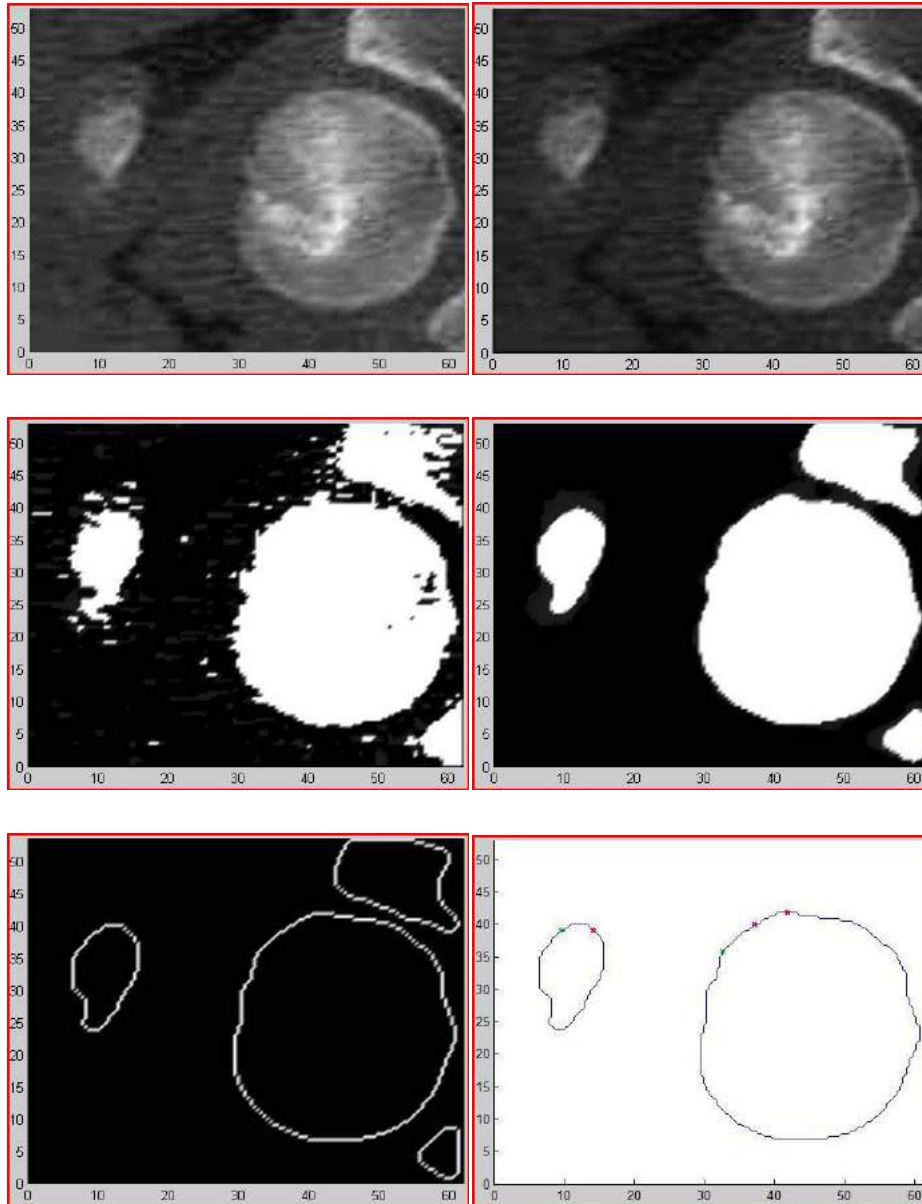


Fig. 14: Referred from left to right and top to bottom:(a) Original image,(b) Brightness reduced,(c) Threshold & converted to logical,(d) after Median Filtering,(e) Edge detected, and(f) both contours traced.

Note: The red and green crosshair in fig. 14(f) represents the points where contour tracing begins. The green one represents successful tracing of contour and red one denoting that the contour was detected again but it was prevented from being retraced as proposed in the algorithm, thus avoiding repetition.

2.5.2 Segmentation of images with multiple contours

The algorithm for segmentation of contours is effective in extracting the points inside multiple contours and storing set of points inside each contour as a separate entity. This section elaborates on the segmentation of multiple contours and the steps in segmentation of multiple contours have been explained. The sequence of operations is as follows.

- First contour

The image is subjected to Canny edge detection after pre-processing. Fig. 14 (e) shows the results of edge detection with multiple contours. According to the algorithm, the next step is Moore neighborhood contour tracing. This algorithm first detects a point on the first contour. This point is marked as green crosshair. The algorithm then starts tracing the points on the contour in the clockwise direction and stores them in the matrix (tc). If the first and the last point in the matrix (tc) are same, then it detects the completion of contour and remembers the contour with the variable name $N=1$. After the detection of contour, point in polygon test is used to detect the co-ordinates of points inside the contour. Fig. 15 (b) shows the results of point in polygon test. Points inside the first contour are marked red while all other points on the image are marked blue. Pixel intensities of points that lie inside the contour are extracted from the original image and displayed to show the segmented contour (Fig. 15 (c)).

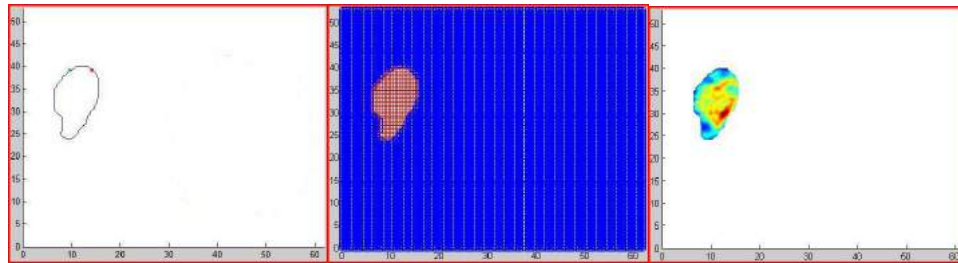


Fig. 15: Referred from left to right. First contour: (a) Contour traced, (b) after point in polygon test, and (c) after segmentation.

- Second contour

After segmentation of the first contour, the loop of contour tracing begins again. If the first contour is detected again, its first point is marked by a red crosshair and the algorithm prevents it from being retraced. The tracing algorithm then detects the second contour, traces it and remembers it by the variable name $N = 2$. Interior points are segmented using the same method and stored as a separate matrix. Fig. 16 (c) shows the second contour that was successfully segmented. If more contours exist, they are segmented in a similar manner.

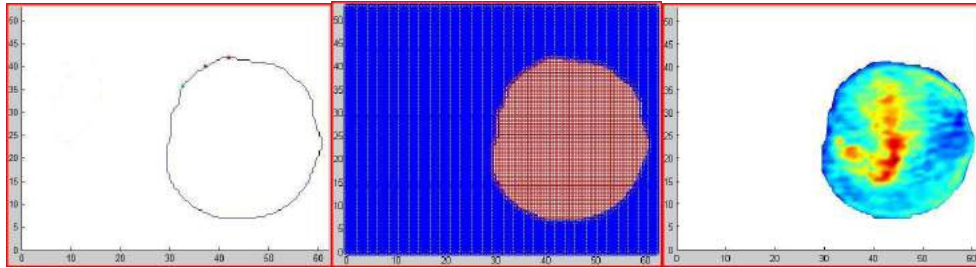


Fig. 16: Referred from left to right. Second contour: (a) Contour traced, (b) after point in polygon test and(c) after Segmentation.

2.5.3 Segmented contours

Successfully segmented regions of the original image (Fig. 14(a)) are shown in Fig. 17.

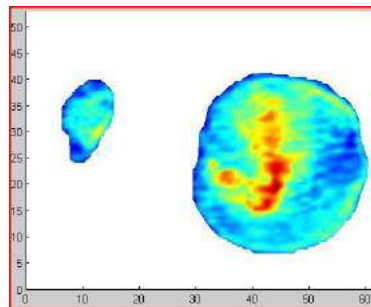


Fig. 17: Final image with segmentation of multiple contours.

3 CONCLUSION

Image segmentation of grayscale CT image based on median filtering, Canny Edge detection and Moore neighborhood boundary tracing is presented and the results at each step have been displayed. A novel algorithm for segmentation of multiple contours on an image has been developed. CT scan images possess tremendous noise, and the shapes of human organs are very complex. To yield better results at the edge detection stage, the image was first subjected to pre-processing. Pre-processing involves RGB to grayscale conversion, normalization, boundary modification, thresholding and filtering. Before selecting the filtering method, we compared various filtering techniques like Median filtering, Averaging filter, Gaussian low-pass filter, Laplacian filter, Prewitt filter, and Sobel filter to remove noise. Median filtering was adopted since it achieved better results for noise removal.

Different Edge detection algorithms were also compared to select the best possible results. Canny and Sobel edge detection returned good results, while Prewitt and Roberts returned weak edges. We have selected Canny edge detection method because it first smoothens the edges to eliminate noise [7], the gradient is calculated using the derivative of a Gaussian filter. It uses two thresholds to detect strong and weak edges. Weak edges are included in the output only if they are connected to strong edges. Segmentation of multiple contours from image has been successfully achieved.

The application of above algorithm can be extended to medical imaging, where 3D reconstruction of body parts, from CT scan images is carried out by eliminating unnecessary data to segment out the region of interest.

4 REFERENCES

- [1] Advanced Biometric Control, <http://www.advancedbiometriccontrol.com/company.php>
- [2] Astola J.; Kuosmanen P.: Fundamentals of nonlinear digital filtering, CRC, 1997, Boca Raton, FL.
- [3] Canny, J. F.: A computational approach to edge detection, IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-8, November, 1986, 679-697.
[doi:10.1109/TPAMI.1986.4767851](https://doi.org/10.1109/TPAMI.1986.4767851)
- [4] Franklin R.: Point Inclusion in Polygon Test, <http://www.ecse.rpi.edu/Homepages/wrf/geom/pnpoly.html>
- [5] Ghuneim, A. G.: Moore Neighborhood Contour tracing, 2010. http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/moore.html
- [6] Giraldi G. A.: Image Enhancement, LNCC-National Laboratory for Scientific Computing- A.V. Getulio Vargas, 333, 25651-070, Petropolis, RJ, Brazil.
- [7] Gonzalez R.C.; Woods R.E.: Digital Image Processing, 2nd ed. Prentice Hall, 2002.
- [8] Grove, O.; Rajab, K.; Piegler, L. A.; Susana Lai-Yuen: From CT to NURBS: Contour Fitting with B-spline Curves, Computer-Aided Design & Applications, 8(1), 2011, 3-21.
- [9] Haines, E.: Point in polygon strategies. In: Heckbert, P.: Editor, Graphic Gems IV, Academic Press, Boston, MA, 1994, 241-46.
- [10] Heath M.; Sarkar S.; Sanocki T.; Bowyer K.W.: Comparison of Edge Detectors: A Methodology and Initial Study, Computer Vision and Image Understanding, 69, 1998, 38-54.
[doi:10.1006/cviu.1997.0587](https://doi.org/10.1006/cviu.1997.0587)
- [11] Huang T.S.: Two dimensional digital signal processing II: Transforms and Median Filters, Springer-Verlag, New York, 1981.
- [12] Ikonomakis, N.; Plataniotis, K. N.; Venetsanopoulos, A. N.: Color Image Segmentation for Multimedia Applications, Journal of Intelligent and Robotic System, 28, 2000, 51-20.
[doi:10.1023/A:1008163913937](https://doi.org/10.1023/A:1008163913937)
- [13] Jain A.: Fundamentals of Digital Image Processing, Prentice-Hall, 1986, 408.
- [14] Justusson, B.J.: Median filtering: statistical properties. In: Huang, T.S.: Editor, Two Dimensional Digital Signal Processing II, Springer, Berlin, 1981, 161-196.
[doi:10.1007/BFb0057597](https://doi.org/10.1007/BFb0057597)
- [15] Lee S.U.; Chung S.Y.; Park R.H.: A comparative performance of several global thresholding techniques for segmentation, Computer Vision, Graphics, and Image Processing, 1990, 171-190.
[doi:10.1016/0734-189X\(90\)90053-X](https://doi.org/10.1016/0734-189X(90)90053-X)
- [16] Pavlidis T.: Algorithms for graphics and image processing, Computer Science Press, Rockville, MD, 1982.
- [17] Pham, Dzong L.; Xu, Chenyang; Jerry, L.: Current Methods in Medical Image Segmentation, Annual Review of Biomedical Engineering, 2, 2000, 315-337.
[doi:10.1146/annurev.bioeng.2.1.315](https://doi.org/10.1146/annurev.bioeng.2.1.315)

Computer-Aided Design & Applications, 9(4), 2012, 501-516

© 2012 CAD Solutions, LLC, <http://www.cadanda.com>

- [18] Pradhan, Ratika; Kumar, Shikhar; Agarwal, Ruchika; Mohan, P.; Pradhan; Ghose M. K.: Contour Line Tracing Algorithm for Digital Topographic Maps, International Journal of Image Processing (IJIP), 2010.
- [19] Preparata, F. P.; Shamos, M.I.: Computational Geometry: An Introduction. Springer-Verlag, New York, NY, 1985.
- [20] Rosenfeld, A.: SSDigital Topology, American Mathematical Monthly, vol. 86, 1979, 621-630. [doi:10.2307/2321290](https://doi.org/10.2307/2321290)
- [21] Rosenfeld, A.; Melter, R. A.: SSDigital Geometry, The Mathematical Intelligencer, 11(3), 1989, 69-72. [doi:10.1007/BF03025195](https://doi.org/10.1007/BF03025195)
- [22] Sahco P.K.; Soltani S.; Wong A.K.; Chen Y.C.: A Survey of Thresholding Techniques, Comp. Vision, Graphics Image Process, 41, 1988, 233-260. [doi:10.1016/0734-189X\(88\)90022-9](https://doi.org/10.1016/0734-189X(88)90022-9)
- [23] Senthilkumaran, N.; Rajesh, R.: Edge Detection Techniques for Image Segmentation, International Journal of Recent Trends in Engineering, 1, May 2009, 250-253.
- [24] Shapiro, Linda G.; Stockman, George C.: Computer Vision, 2001, 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3.
- [25] Shin, M.C.; Goldgof, D.; Bowyer, K.W.: Comparison of Edge Detector Performance through Use in an Object Recognition Task, Computer Vision and Image Understanding, 84, October, 2001, 160-178. [doi:10.1006/cviu.2001.0932](https://doi.org/10.1006/cviu.2001.0932)
- [26] Struc, Vitomir; Pavesic, Nikola: Image Normalization Techniques for Robust Face Recognition, Proceedings of the 8th WSEAS International Conference on SIGNAL PROCESSING, ROBOTICS and AUTOMATION, 2009.