XVIII IMEKO WORLD CONGRESS
Metrology for a Sustainable Development
September, 17 – 22, 2006, Rio de Janeiro, Brazil

3D LUNGS SURFACE RECONSTRUCTION FROM COMPUTED TOMOGRAPHY IMAGES

Tomasz Koszmider [1], *Krzysztof Strzecha* [2]

[1] Computer Engineering Department at Technical University of Lodz, Poland, tkoszmi@kis.p.lodz.pl
[2] Computer Engineering Department at Technical University of Lodz, Poland, strzecha@kis.p.lodz.pl

Abstract: New 3D medical images reconstruction algorithm is presented in this paper. It concerns utilization of digital processing and analysis of Computed Tomography images in process of 3D lungs reconstruction.

Keywords: Computed Tomography, 3D reconstruction.

1. INTRODUCTION

Computed Tomography (CT) is a diagnostic imaging procedure that uses x-rays and computer technology to produce detailed cross-sectional images of the body part being studied [1] (Fig. 1). These images can be used to three-dimensional reconstruction which is very popular non-invasive diagnostic method in medicine.

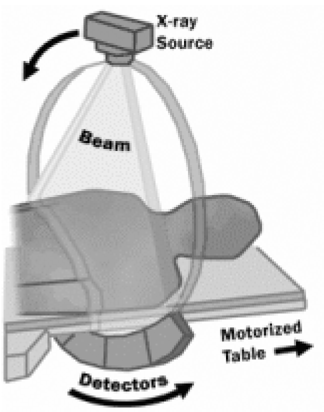

Fig. 1 CT data acquisition.

Usually in Computed Tomography scans of cross-sections are 12 or 16 bit grayscale image with pixel intensity being proportional to absorption characteristics of the tissue being studied (Fig. 2). X-ray absorption is measured in Hounsfield units (HU) (Table 1) [1] and it is called density.

3D reconstruction algorithm described in this paper was developed to assist physicians in lungs surgery planning. Although several other algorithms already exist, most of them use approximating methods of edge detection what results in less detailed reconstruction and sometimes in non-predictable artifacts existence.

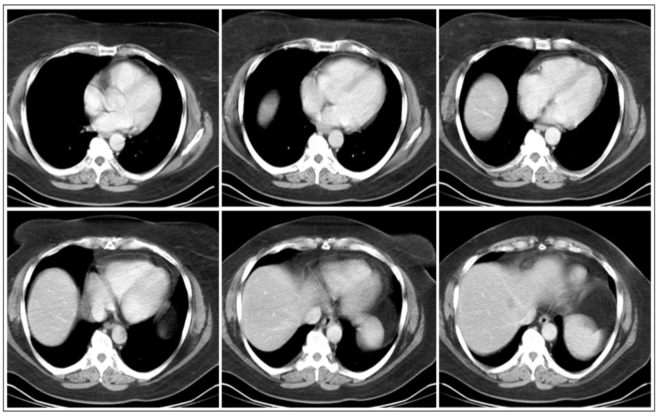

Fig. 2 CT lungs scans.

Table 1. Sizes (in points) and font styles.

Air	-1024 [HU]
Water	0 [HU]
Lungs	-900 ÷ -300 [HU]
Internal tissues	-200 ÷ 200 [HU]
Bones	200 ÷ 1000 [HU]

2. PURPOSE SPECIFICATION

Presented reconstruction algorithm was developed with idea of non-approximating technique for edge detection and fast process of 3D reconstructions in mind. The latter requirement was legitimate in view of possible re-usage of the separate modules in application for different operating system.

Proposed 3D reconstruction algorithm is described in two steps. Firstly process of identification of areas occupied by lungs is presented. Second part describes process of 3D reconstructions of identified outlines.

3. RECONSTRUCTION PROCESS

3.1. Identification

Identification of lungs areas generally was made in three steps:

- preliminary processing,
- outline tracing,
- recognition of geometric features of the image outline.

Preliminary processing

The goal of preliminary processing is to remove edge information not related to the boundaries of interest and to reduce their complexity [2].
This process has been done with basic image segmentation methods applied to thresholded binary image image (Eq. 1):

$$g(x,y)=\begin{cases}1 \ for \ L_S(x,y)\geq T \\ 0 \ for \ L_S(x,y)<T\end{cases} \quad (1)$$

where:
$L_S(x, y)$- image (two dimensional intensity function),
$g(x, y)$ - binary image (result of thresholding),
T - threshold value determined on base of histogram analysis.

Edges complexity is well reduced by utilizing morphological operations (with the same structuring element):

- dilation (Eq. 2) [2, 3],
- erosion (Eq. 3) [2, 3].

$$A \oplus B = \{x, y : B_{xy} \cap A \neq \emptyset\} \quad (2)$$

$$A \otimes B = \{x, y : B_{xy} \subseteq A\} \quad (3)$$

where:
$A, B \subset R^2$
$\oplus$ - dilation,
$\otimes$ - erosion,
A - image,
B - structuring element.

Careful choice of morphological filter sequence with it's structuring element ensures that original shape of the specimen will not be deformed.

Extraction of edges information from morphologically filtered binary image has been done by convolving 3x3 point detection mask with image (Fig. 3) [4].

-1	-1	-1
-1	8	-1
-1	-1	-1

Fig. 3 Point mask.

Thinning edges to a single pixel in width and removing each edge pixel's p-neighbors allows intersections to be identified as points with more then two n-neighbors.

Outline tracing algorithm (Fig. 4)

Outline tracing algorithm has following aims:

- remove redundant edges, this is all those which are not part of a closed loop,
- separate edges which form closed loop, in order to each consist of pixels with two and only n-neighbors.

Algorithm starts with the first element of array containing the edges. It checks every succeeding element's value until it reaches last element of array or it finds an edge (represented by value of 1). In the latter case, algorithm assumes that this element is part of outline and it starts outline tracing procedure.

Firstly, image and point detection mask correlation is determined. It allows calculating number of n-neighbors of point being considered. Depending on this number one of four possible actions is taken:

1) If point has two n-neighbors, its position is stored in a temporary array. One of two n-neighbors, which were not considered earlier, becomes currently considered.
2) If point has more than two n-neighbors, the intersection occurs and current point position is stored in temporary array with '-' sign indicating intersection. First clockwise n-neighbor (begins with upper one) becomes currently considered.
3) If point does not have n-neighbors, it cannot be part of a closed loop. Tracing procedure terminates with empty temporary array.
4) If point has one n-neighbor, current edge is redundant. Algorithm removes last element of temporary array until first element of array is meet (tracing procedure terminates with empty temporary array), or negative value is meet (number after '-' sign indicates currently considered point position).

When action 1 or 2 performs and currently considered point's position already exist in temporary array, tracing procedure terminates. All points from temporary array stored after first occurrence of currently considered point's position indicates successfully traced outline.

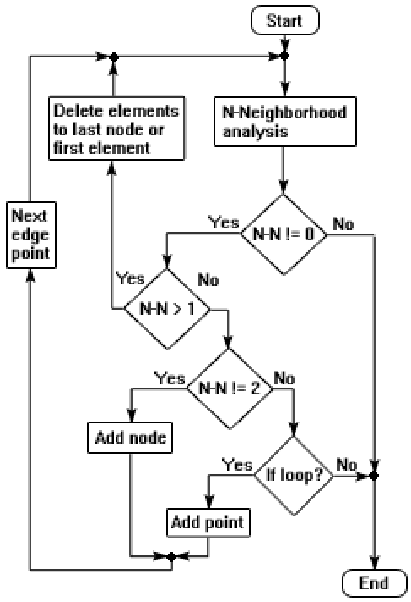

Fig. 4 Outline tracing algorithm.

Geometric features of traced outlines (Fig. 5)

This algorithm complements identification process. It determines area of traced outlines. With this information main program can easily calculate percentage of pixels with lungs densities and decide whether the outline bounds area occupied by lungs or not. Result of whole identification process is shown on Fig. 6.

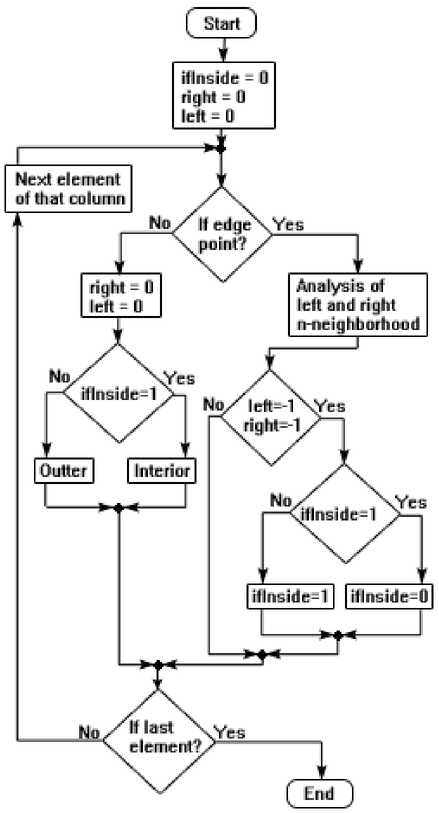

Fig. 5 Outline's area calculating algorithm.

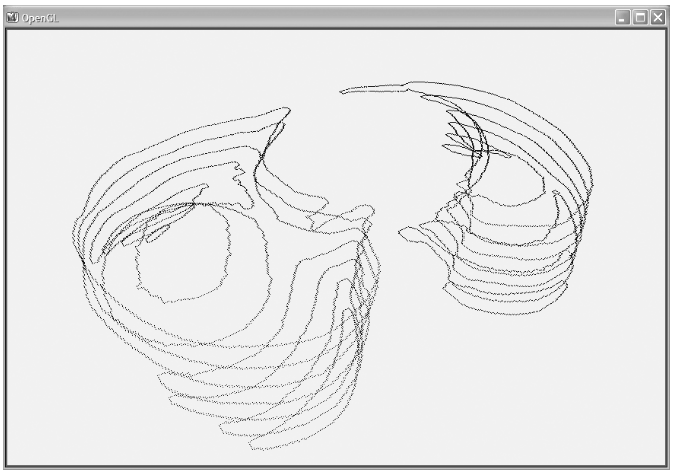

Fig. 6 Result of lungs identification process.

Algorithm uses four following indicators:

ifInside - indicate if non-edge points are outline's internal or background points;

left - n-neighbor of previous column;

right - n-neighbor of next column.

Each element of outline's array may have value 1 indicating edge elements, or value 0 indicating non-edge elements which may be either outline's internal or outside point. Algorithm checks every column of outline's array separately beginning with lowest index element. To decide whether currently considered point belongs to outline's interior or not, algorithm has to check if edge was intersected by previously considered points of that column and how many times it has happened (Fig. 7).

Detection of outline intersection is made by analysis of edge point's left and right n-neighborhood. Indicators 'left' and 'right' can be of following values:

- 0 if value of n-neighbor equals 0,
- 1 if value of n-neighbor equals 1,
- -1 if both left and right n-neighbors are of value 1.

Every time these indicators change theirs values to -1, indicator 'ifInside' become of value:

- 0 if number of these changes is even,
- 1 if number of these changes is odd.

All non-edge points which were considered while 'ifInside' indicator had value of 1 are outline's interior points.

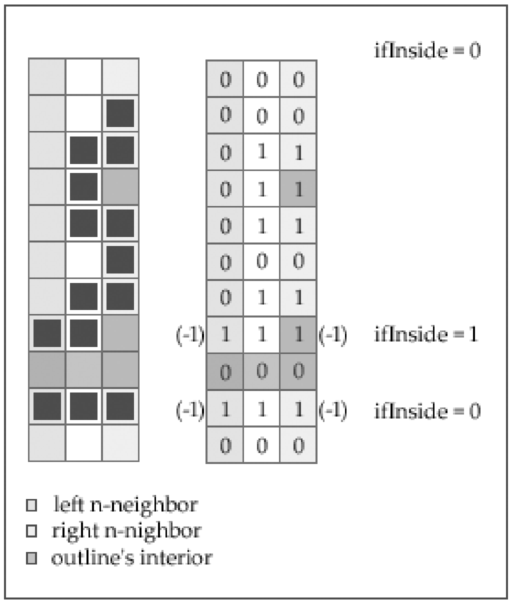

Fig. 7 Edge crossing detection.

3.2. 3D reconstruction of identified outlines

Due to specific lungs structure algorithm of 3D reconstruction was developed on this special case requirement. It creates triangle strips from points of all identified outlines between every two neighboring cross-sections (Fig. 8). Algorithm must take into consideration number of identified outlines in each pair of cross-sections. This information is most significant in choice of a way these outlines would be connected. Level of details depends only on CT scan's resolution.

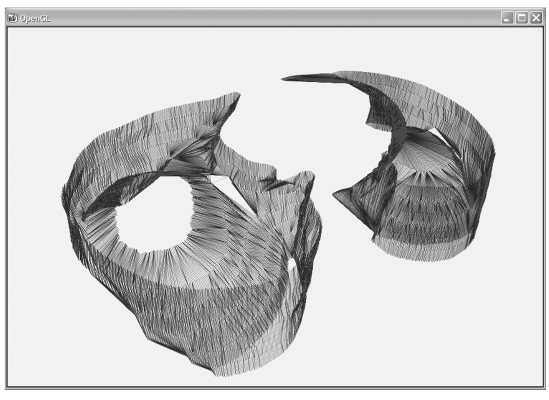

Fig. 8 Result of 3D mesh construction process.

4. CONCLUSION

Size of human lungs and their internal construction allows saying that this organ is best object for 3D reconstruction and visual presentation. Lungs density is much bigger then surrounding tissues, therefore segmentation of these areas is very exact. Result of 3D reconstruction is astonishingly good and detailed (Fig. 9, 10). Hardware requirements are not high. PC with AMD Athlon 1,99 GHz and 256 MB of Random Access Memory need only 7 seconds to calculate 3D mesh from 7 cross-section images.

To compare results with existing methods, two popular 3D volumetric reconstruction algorithms (Marching Cubes [5] and Marching Tetrahedrons [6]) were implemented. Both methods gives less detailed result and also both have propensity to reconstruct no existing structures. Reconstruction from 7 cross-section images using Marching Tetrahedrons method takes also nearly 7 seconds (Table 2), which is good result in comparison to almost 15 seconds of reconstruction taken by the Marching Cubes method.

Table 2. Reconstruction times comparison.

Algorithm	Reconstruction time [s]
Described algorithm	7
Marching Cubes	15
Marching Tetrahedrons	7

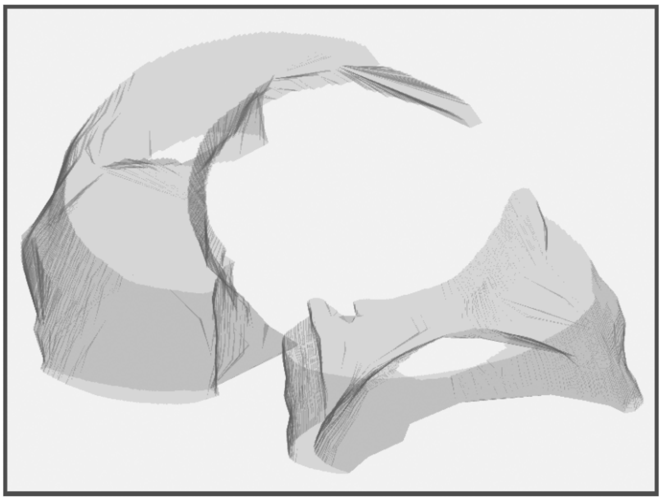

Fig. 9 Reconstructed lungs model (bottom view).

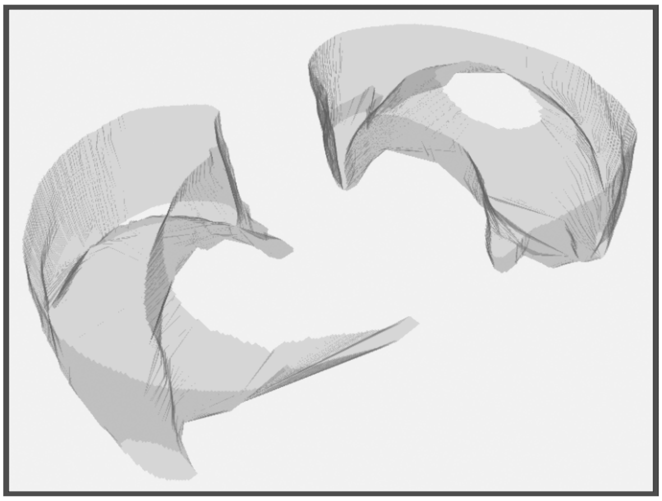

Fig. 10 Reconstructed lungs model (top view).

REFERENCES

[1] A. C. Kak, Malcolm Slaney *"Principles of Computerized Tomographic Imaging"*. IEEE Press, 1988.

[2] J. Teuber *„Digital image processing"*. Copenhagen University Observatory, 1993.

[3] R. C. Gonzales, P. Wintz *„Digital image processing"*. Addison-Wesley Publishing Co., Reading, Massachusetts, USA, 1987.

[4] Julian Kerr, University of New South Wales *"The 'Trace' Method for Segmentation of Lungs from Chest CT Images* by *Deterministic Edge Linking"*. http://www.cse.unsw.edu.au/~juliank/2000/trace/trace.pdf

[5] W.Lorensen, H.Cline *"Marching Cubes: A High Resolution 3D Surface Construction Algorithm"*. Computer Graphics, 21 (4): 163-169, July 1987.

[6] A.Doi, A.Koide *"An Efficient Method of Triangulating Equivalued Surfaces by using Tetrahedral Cells"*. IEICE Transactions Commun, Elec. Info. Syst, E74(1) 214-224, January 1991.