

A Procedure for the 3D Reconstruction of Biological Organs from 2D Image Sequences

Kirana Kumara P

*Centre for Product Design and Manufacturing
Indian Institute of Science
Bangalore, 560 012 India*

Ashitava Ghosal

*Centre for Product Design and Manufacturing
Indian Institute of Science
Bangalore, 560 012 India*

Abstract

In this work, a procedure is presented for the reconstruction of biological organs from image sequences obtained through CT-scan. Although commercial software, which can accomplish this task, are readily available, the procedure presented here needs only free software. The procedure has been applied to reconstruct a liver from the scan data available in literature. 3D biological organs obtained this way can be used for the finite element analysis of biological organs and this has been demonstrated by carrying out an FE analysis on the reconstructed liver.

Key Words: analysis, finite, element, liver, model.

1. Introduction

Finite element analysis has been widely practiced in the biomedical engineering field for the simulation of biological organs (e.g., for surgery planning, surgery simulation). Actual organ geometries are needed for these finite element analyses. But, anatomical dimensions and geometry of body organs differ from patient to patient due to various factors such as age, body size, and weight and due to the presence of pathologies such as cysts and cancer. Using scanning procedures like Computed Tomography (CT), one can get patient specific 2D image sequences corresponding to different organs. But, one needs to use commercial software (e.g., 3D Doctor [1], AMIRA® [2]) to extract 3D geometry of biological organs from these image sequences. Hence the goal of the present paper is to provide a free alternative to these paid software. As an alternative to the commercial software, the present paper gives a procedure to get patient specific organ geometries from patient specific scanned image sequences. The procedure makes use of multiple software but they are all free.

In this work, it is also demonstrated that the geometry obtained by following the procedure presented in this

work, can be used for a finite element analysis. Commercial software Rhinoceros [3] and ANSYS [4] are used for this purpose.

The paper is organized as follows. Section 2 describes the necessary software. Section 3 gives the general procedure to get 3D geometries from scanned image sequences. Section 4 applies this procedure for the 3D reconstruction of a liver. Section 5 uses the liver geometry thus obtained for a finite element analysis, demonstrating that an FE analysis can be performed on patient specific organ geometries obtained using the procedure presented in this work.

2. Necessary Software

The software that are needed to carry out the procedure (to be explained in the next section) are: ImageJ, ITK-SNAP, MeshLab. Very concise information on these software is given below. Also, the features that are most useful for the present work are taken a note of.

2.1. ImageJ

ImageJ [5, 6, 7] is an image processing software. This software is not subject to copyright protection and is in the public domain. It is written in Java and hence may be installed on any computer with pre-installed Java.

Present work uses ImageJ 1.42q (together with Java 1.6.0). Its role is to import an image stack from the file, getting image information, selecting the image portions with a rectangle and processing the selected portion only, removing selected portion of the image, subtracting a particular intensity value from each pixel in the image, multiplying each pixel intensity value by a particular number, applying the image processing procedures to a single image or to the whole image stack, reversing the order of images in an image stack, concatenating image stacks, saving the processed image stack in different file formats etc.

2.2. ITK-SNAP

ITK-SNAP [8, 9] is a software application used to segment (image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics) structures in 3D medical images. It provides semi-automatic segmentation using active contour methods, as well as manual delineation and image navigation. It can export a segmented image sequence as a 3D surface mesh. ITK-SNAP is free software, provided under the General Public License [10]. ITK-SNAP 2.0.0 is used in the present work.

ITK-SNAP is capable of importing image stacks in different file formats. The present work uses the software for 3D automatic (also called semi-automatic) segmentation.

Automatic segmentation first involves the selection of the region of interest. The region of interest applies to all the images in the stack. Snake initialization is the next step. Snake is a closed curve or collection of closed curves in 2D, and a closed surface or collection of closed surfaces in 3D. Snake can be initialized using a circular bubble. The snake is in fact the initial approximation for the segmented volume. In the next step, the snake is made to evolve or the initial segmentation represented by the spherical bubble changes its shape towards the shape of the final segmented volume that represents the surface of the biological organ to be extracted from the image stack. The final step in using ITK-SNAP is to export the 3D segmented volume as a 3D surface.

2.3. MeshLab

MeshLab [11, 12] is an open source system for the processing and editing of unstructured 3D triangular meshes. The system is aimed to help the processing of the typical not-so-small unstructured models arising in 3D scanning, providing a set of tools for editing, cleaning, healing, inspecting, rendering and converting this kind of meshes. It is licensed under the GNU General Public License. MeshLab v1.2.2 has been used in the present work. Some of the numerous useful features available in the software are explained next.

Fill Hole checks whether the 3D surface is watertight. MeshLab should report that there are no holes in the mesh to edit.

Quadric Edge Collapse Decimation filter is used to convert the geometry described by very large number of triangle faces that is usually the case, to a geometry that is a very good approximation to the original one but described with lesser number of faces.

Triangle Quality filter displays the quality measures for each of the triangle surfaces using a colour code.

MLS Projection filter is used to improve the triangle quality of surface triangles. It accomplishes this by projecting the whole mesh onto the MLS surface defined by itself [13].

Laplacian Smooth filter is another filter used for improving triangle quality. Here, the smoothing is accomplished by calculating the average position for each vertex with nearest vertex.

Taubin Smooth filter is also found to be very useful for improving triangle quality and it makes two steps of smoothing, forth and back, for each iteration.

3. The General Procedure for Extracting 3D Organs from 2D Image Sequences

These steps make up the procedure to obtain 3D organs from 2D image sequences. These steps make use of the software (all free) explained in the previous section.

- (1) Open ImageJ, import the image file (e.g., *.tif), process the image stack with the goal of brightening the pixels which belong to the organ of interest but with the goal of making the intensity of the pixels which do not belong to the organ of interest zero. This makes the next step (segmentation) much easier. Many a times, to reduce size, image files give only the half of the information (the other half of the organ may be considered to be symmetric to the first half). In that case, one can reverse the order of images in the image stack and then concatenate this image stack with the original image stack to get the full image stack. One should save the processed image stack in a file format that can be read by ITK-SNAP (e.g., *.raw).
- (2) Open ITK-SNAP and open the file saved in the previous step. If the image file format does not include the image header information (like the total number of pixels in each direction, absolute distance between pixels in both the directions, total number of such images in the image stack considered, distance between the consecutive images in the stack, voxel representation scheme etc.), ITK-SNAP requests the user to enter this information. In this case, the information has to be obtained from the source that supplied the image file. Then, ITK-SNAP displays the image stack. Next, one has to complete segmentation (manual or automatic). Then, the segmented volume has to be exported as a 3D surface mesh using the menu item *Save as Mesh...* in a file format supported by MeshLab (e.g., *.stl).
- (3) Open MeshLab and open the file saved in the previous step. While opening, MeshLab asks whether it is all right to unify duplicate vertices

and one can select *OK*. Apply *Fill Hole* to ensure that the mesh does not contain any holes. Next, *Quadric Edge Collapse Decimation* filter is used to reduce the number of triangle faces to the target number of faces. *Triangle Quality* is used to colour code the triangles to identify ill shaped triangles and the filters *MLS projection*, *Laplacian Smooth* and *Taubin Smooth* are used to make those triangles well shaped. Finally, the processed mesh is saved. This mesh represents the 3D organ of interest.

4. Application of the Procedure for the 3D Reconstruction of a Liver

In this section, the procedure presented in the previous section has been followed to reconstruct a pig liver from the images obtained through CT-scan. A pig liver has been reconstructed here just because the CT-scan data was readily available for this case. But, the procedure is general in the sense that one can reconstruct any organ, animal or human, by following the procedure.

4.1. Obtaining the Image File and its Details

The image file has been obtained from [14]. The image file is a zipped .tif image stack that contains a pig liver in the undeformed state and it contains 147 images. The image sequence has the name: `LiverNoDeformation.tif` and has a size of around 37 MB. The image stack was obtained using CT-scan and one can refer to [15] for the details on the procurement and processing of the liver used for the scan, the scanning equipments used, scanning procedure employed, and further details on the image sequence etc. The things which are of interest to us are: 512 X 512 pixels make up each image in the stack, the distance between pixels (in both directions) = 0.29 mm, distance between consecutive images in the stack = 1 mm, each pixel needs 8 bits of memory to describe it and each pixel is described by an 8 bit unsigned integer, the images are gray scale images, since the images are gray scale images and since each pixel is represented by an 8 bit unsigned integer each pixel can have an intensity between 0 to 255 (totally 256 intensity levels).

4.2. Use ImageJ (Step 1 of Section 3)

Use *File* → *Import* → *Raw...*, enter image information (from Section 4.1) and the image stack `LiverNoDeformation.tif` will be displayed. The 50th and the 100th image are shown in Figure 1 and Figure 2 respectively.

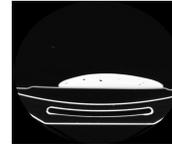


Figure 1. The 50th Image

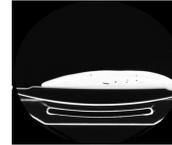


Figure 2. The 100th Image

It can be observed that the liver is located around the center of the images and the bottom portion of the images shows the part of the experimental set up supporting the liver. Our goal here is to make the pixels belonging to the liver very bright and making the intensity values of other pixels (which do not belong to liver) zero; we should do this for all the images in the image stack. This makes the next step (segmentation) that uses ITK-SNAP very trivial, since only the pixels that belong to the liver volume have some positive intensity values. This is accomplished as follows.

Use *Process* → *Math* → *Subtract...*, enter *Value* = 200, select *Yes* to accept the processing of all 147 images. Now select the bottom portion which does not belong to the liver in a yellow rectangular box and use *Process* → *Math* → *Multiply...*, then enter *Value* = 0, and process all 147 images. This will set the intensity values of the pixels selected in the rectangular box to zero. Follow the same procedure to eliminate the left side portion of the image stack that does not belong to the liver. Now only those pixels that belong to the liver volume have some positive intensity value. Now use *Process* → *Math* → *Multiply...*, then enter *Value* = 3 to multiply these intensity values by a factor of three. This brightens up the image of the liver. The 50th and 100th image, after undergoing processing with ImageJ are shown in Figure 3 and Figure 4.

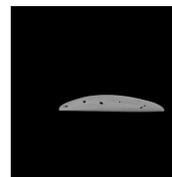


Figure 3. The 50th Image (After using ImageJ)

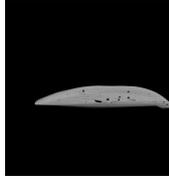


Figure 4. The 100th Image (After using ImageJ)

Now, *File* → *Save As* → *Raw Data...* has been used to save the image as a.raw. The original image stack represents only half of the complete image stack. Hence, use *Stack Reverser* to reverse the order of images in the image stack, save this new stack as b.raw, *Concatenate* a.raw and b.raw to get the full stack (with 294 images) 'LiverNoDeformation.raw'.

4.3. Use ITK-SNAP (Step 2 of Section 3)

Use *File* → *Open Grayscale Image...* to browse to the location of 'LiverNoDeformation.raw'. *Raw Binary File* should be selected as the image file format. Supply the image details (from 4.1) when asked. Image stack will be displayed. Now select *SNAP Tool* in the *IRIS Toolbox*. Make sure that the region selected for segmentation (through red dotted lines) includes the whole liver, and then press *Segment 3D*. Now select the *Intensity Regions* radio button, and press *Preprocess Image* and *Intensity Region Filter* window pops up. Select the proper settings here, proceed to the next step and *Add Bubble*. Now *Iterate Continuously*. Now we can see how bubble slowly changes its shape to the shape of the image on the screen. When shape of the bubble assumes the shape of the image on the screen (in all the windows), segmentation is complete and we should press *Finish*. All the images in the image stack have been segmented by now. Now use *Segmentation* → *Save As Mesh...* to save the segmented volume that represents the liver as a surface mesh made of triangle faces. Here, the surface mesh file is named as 'liver.stl'. Size of the file is 86.0 MB.

4.4. Use MeshLab (Step 3 of Section 3)

Open 'liver.stl' in MeshLab. The liver as seen in MeshLab is shown in Figure 5. Here, the liver is represented by 288190 vertices or 576376 triangle faces. We use *Quadric Edge Collapse Decimation* filter to reduce the total number of faces representing the organ. Now, the same liver represented by 1002 vertices (2000 faces) is shown in Figure 6 and represented by 102 vertices (200 faces) is shown in Figure 7. We use *MLS Projection*, *Laplacian Smooth* and *Taubin Smooth* filters to improve triangle quality. The liver represented by 102 vertices (or 200 faces) only but with well-shaped triangles

is shown in Figure 8. Now, we will save this model (liver represented by 102 vertices or 200 faces, all triangles being well shaped) as 'liver102vprocessed.stl'. The size of the file is around 60 KB.

Now we have obtained the 3D model of the liver (shown in Figure 5 to Figure 8).

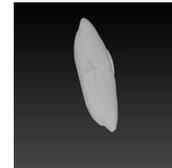


Figure 5. The Liver Displayed in MeshLab (288190 Vertices or 576376 Faces)

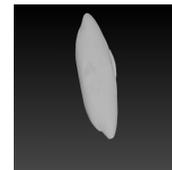


Figure 6. The Liver Displayed in MeshLab (1002 Vertices or 2000 Faces)

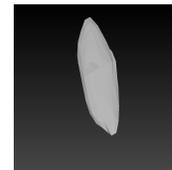


Figure 7. The Liver Displayed in MeshLab (102 Vertices or 200 Faces)

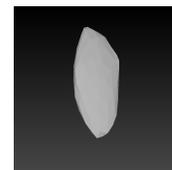


Figure 8. The Liver Displayed in MeshLab (102 Vertices or 200 Faces; Well Shaped Triangles)

5. Using the Reconstructed Liver in a Finite Element Analysis

3D model of the liver obtained in the previous section is a surface mesh made of triangles. Here, the commercial software Rhinoceros is used to convert this 3D surface into 3D solid.

Open 'liver102vprocessed.stl' in Rhinoceros. Use the command *MeshToNurb*, and the resulting solid may be saved in ACIS (.sat) format using *Save As....* Here, the file is named as 'liver102vprocessed.sat'. Size of the file is around 500 KB.

Now the liver 'liver102vprocessed.sat' is imported into ANSYS and used in a finite element analysis. Here, the purpose of carrying out the FE analysis is just to show that 3D models of the organs obtained through the procedure presented in this paper do not pose serious problems during meshing. A test load, boundary condition and material are assumed. Figure 9 shows the deformed shape of the liver superimposed over the undeformed liver, as seen in ANSYS.

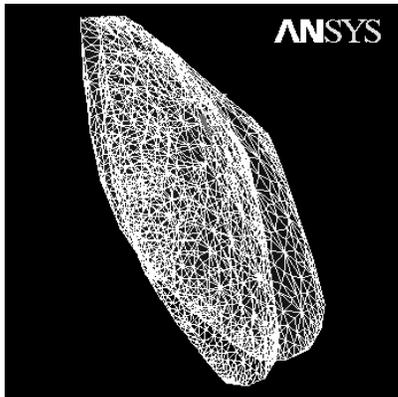


Figure 9. The Liver (Deformed + Undeformed)

While the above analysis does not represent a typical biomedical engineering analysis, it clearly shows that the surface mesh (which represents the reconstructed liver) generated by the present procedure is of good quality.

6. Closing Remarks

A procedure for obtaining 3D solid models of biological organs from image sequences that are obtained through CT-scanning has been presented in this work. The procedure has been verified by successfully reconstructing a liver. The 3D models of the organs thus obtained can be used in a finite element analysis.

The procedure presented here requires much free software. When compared with the commercial software

that can do the same task, one can observe that it is easier and more efficient to use commercial software; usually single commercial software can generate the 3D model from an image sequence. But the procedure presented here will be useful for budget conscious individuals, organizations and applications and where one does not need to obtain organs from images very often.

Since all the software required by the present procedure are in public domain, it would be nice if these software are clubbed together to produce a software comparable to commercial software, and which would be available free for all.

7. References

- [1] <http://www.3d-doctor.com>
- [2] <http://www2.amira.com>
- [3] <http://www.rhino3d.com>
- [4] <http://www.ansys.com/>
- [5] <http://rsbweb.nih.gov/ij/>
- [6] Rasband, W.S., ImageJ, U. S. National Institutes of Health, Bethesda, Maryland, USA, <http://rsb.info.nih.gov/ij/>, 1997-2009.
- [7] Abramoff, M.D., Magelhaes, P.J., Ram, S.J., "Image Processing with ImageJ", *Biophotonics International*, Volume 11, issue 7, pp. 36-42, 2004.
- [8] <http://www.itksnap.org>
- [9] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability", *Neuroimage*, 2006 Jul 1;31(3):1116-28.
- [10] <http://www.gnu.org/licenses/gpl.html>
- [11] <http://meshlab.sourceforge.net/>
- [12] MeshLab, Visual Computing Lab - ISTI - CNR
- [13] Oztireli, Guennebaud and Gross, "Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression", Eurographics 2009
- [14] <http://biorobotics.harvard.edu>
- [15] Amy Elizabeth Kerdok, *Characterizing the Nonlinear Mechanical Response of Liver to Surgical Manipulation*, Ph.D Thesis, The Division of Engineering and Applied Sciences, Harvard University, 2006.