

# Reconstruction & Interaction with 3D Simplified Bone Models

R. Pulido  
University of Jaén  
Campus Las Lagunillas  
23071, Jaén, Spain  
rpulido@ujaen.es

F. Paulano  
University of Jaén  
Campus Las Lagunillas  
23071, Jaén, Spain  
fpaulano@ujaen.es

Juan J. Jiménez  
University of Jaén  
Campus Las Lagunillas  
23071, Jaén, Spain  
juanjo@ujaen.es

## ABSTRACT

Visualization and interaction with 3D bone models reconstructed from medical images are fundamental for bio-medical applications. They are useful for surgeons in order to diagnose and plan surgical operations. Although traditional reconstruction techniques based on isosurfacing are mainly used for visualization, it is desirable to obtain labelled models without outliers in order to improve the interaction. This paper presents the integration of digital image processing and computer graphics techniques to enable not only the reconstruction of simplified 3D bone models, but also the interaction with them. To that end, the segmentation of CT images is performed in order to obtain different bone regions and to label them. This approach is divided into three main parts: segmentation, reconstruction and interaction. The goal of the segmentation is to extract closed contours and to generate labelled regions which represent the bone structures to be reconstructed. Then, 3D bone models are obtained from these regions by isosurfacing. Finally, a detailed collision detection can be calculated between the 3D models in the scene in order to provide visual aid when the user is interacting with them. This interaction includes the calculation of distances, nearest points and overlapping triangles. Moreover, an immersive experience is provided by integrating the Leonar3Do stereo system.

## Keywords

medical image segmentation; bone labelling; 3D bone reconstruction; bone simplification; interaction; collision detection.

## 1 INTRODUCTION

Visualization, reconstruction and interaction with 3D bone models are useful for bio-medical applications. They can be used in medical tasks that involve diagnosis, surgical planning, etc.

The visualization of bone structures is usually performed from Computed Tomography (CT) images. CT images are mainly used to locate high density values and therefore to distinguish dense cortical bone from soft tissue. This type of images can also be processed in order to extract features and structures [Sac04]. CT images are usually acquired as a stack of slices and the data set is represented as a regular volumetric grid. This representation allows to visualize bone tissue by using volume rendering. To that end, there exist toolkits that enable the visualization of large volumetric data sets [BD02]. This type of visualization does not

require the extraction of 3D models. Therefore, the interactivity is limited to common camera operations, such as panning, tilting, and zooming.

Other works use the regular grid representation to perform a 3D reconstruction by isosurfacing. Isosurfacing enables the use of fast and simple methods for rendering. On the other hand, several drawbacks have to be considered due to the high resolution of the isosurfaces extracted. First, the generation of large polygonal datasets requires high computational cost and memory resources. Second, undesirable triangles can be generated due to the presence of artefacts and noise.

Furthermore, post-processing tasks, such as simplification, segmentation or labelling, have to be performed as long as we need to enable interaction and collision detection with the generated models [vdB04].

The main aim of this work is to present the reconstruction of 3D simplified bone models in order to enable interaction with them. For that, we present a set of techniques to perform the segmentation of bone regions from CT images. It allows to simplify, label and classify the bone models. The labelling has also been used to enable interaction with the reconstructed models. Finally, the Leonar3Do system [Leo12] has been integrated to let the user have an immersive experience

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

through stereo visualization. Thus, users can interact with a virtual 3D bone individually for different purposes such as education, training and surgical planning.

## 2 BACKGROUND AND OVERVIEW

There exist many segmentation techniques that can be applied to medical images [PXP00]. These approaches can be categorized as intensity-based, edge-based, region-based and deformable models. However, there is neither a standard segmentation technique that works well for all medical images nor all methods are equally good for the particular type of CT images.

Moreover, automatic segmentation methods usually require constraints, and difficulties are encountered due to different factors such as variations in the shape of the bone and heterogeneous structures [SRS\*10].

Concerning 3D reconstruction from medical images, different approaches have been presented in the literature. As commented above, isosurfacing is the technique commonly used. This approach was initially presented by Wyvill et al. [WMW86] but it was popularized by Lorensen and Cline through the Marching Cubes algorithm [LC87]. In recent years, some techniques have been adapted from popular methods in computational geometry such as Delaunay triangulation and Voronoi diagram [LYG\*09]. Other techniques are based on Hierarchical Space Subdivision Schemes [BHGS06]. Sharf et al. [SLS\*07] presented a method to interactively reconstruct surfaces by using a priori distribution. On the other hand, the extraction of contours, using a hierarchical spatial decomposition, can be complemented with a table of patterns to triangulate these contours [PJP12].

Applications that are usually used to work with medical images are limited to 3D visualization [BD02] and they usually only allow to rotate the camera and to remove outliers if it is necessary. There are several toolkits which enable the manipulation of models reconstructed from medical scans. ITK-SNAP [YPH\*06] is an application that allows to interactively segment structures from medical scans. MITK [WWV\*04] combines VTK and ITK to provide interaction with models reconstructed from medical scans. Most of these methods and tools do not allow to separate and classify different 3D bone models and interact with them. In order to achieve it, labelling is essential to identify the different bone structures and it is a non-trivial task that usually requires user interaction in complex cases. Previous works perform the labelling after isosurface extraction. Harders et al. [HBG\*07] process the manual separation of erroneously connected isosurfaces, being the artificial connections removed by the user. To this end, a cut line can be interactively drawn onto the bone isosurface. Yushkevich et al. [YPH\*06] use cut-plane

operations to relabel a segmentation into different structures. Consequently, these approaches require to post-process the isosurface generated.

In this paper, we present an approach for labelling the different bone structures before the reconstruction. To that end, CT images are processed to perform the bone labelling during the segmentation. This image processing is focused on the extraction of filled bone regions, which allows to reconstruct 3D simplified bone models. In addition, the bone labelling enables the interaction with these models. This interaction includes, among others, picking and collision detection methods. Finally, the Leonar3Do system lets the user have an immersive experience through the stereo visualization. For all the above process, this approach can be used for training and surgical planning that requires the interaction with virtual bones. It can help the user to reconstruct and place bones in their correct position from real medical images.

An overview of the process is shown in Figure 1. As depicted in Figure 1 the different techniques are mainly grouped into three main categories: "Segmentation & Labelling", "3D Bone Reconstruction" and "Interaction". During the segmentation, CT images are taken as input to be processed individually. In addition, the bone regions of interest are identified for each CT image. Then, these regions are used for extracting a set of 3D simplified bone isosurfaces. Finally, different techniques are integrated to enable stereo visualization and interaction with these 3D bone models.

The results of the different techniques are shown throughout the paper. For that, a CT scan of the knee has been used. It consists of 156 slices that were taken at 0.675 mm intervals at a resolution of 512 x 512 pixels.

## 3 SEGMENTATION AND LABELLING

The goal of this step is to obtain a set of labelled and filled bone regions for each CT image. Both features are essential for the classification and simplification of the bone structures. In order to achieve it, we use the image processing techniques described below.

### 3.1 Smoothing

Since CT images are very heterogeneous, specially in the trabecular bone areas, images are smoothed with a curvature flow filter [Set99]. This algorithm uses a level set formulation and performs edge-preserving smoothing (see Fig. 2). Thus, the pixels of a particular intensity form one level set. The level set function is then evolved under the control of a diffusion equation where the speed is proportional to the curvature of the contour. Thus, areas of high curvature are diffused faster than areas of low curvature. In addition, noise artefacts disappear quickly, while large scale interfaces will be slow

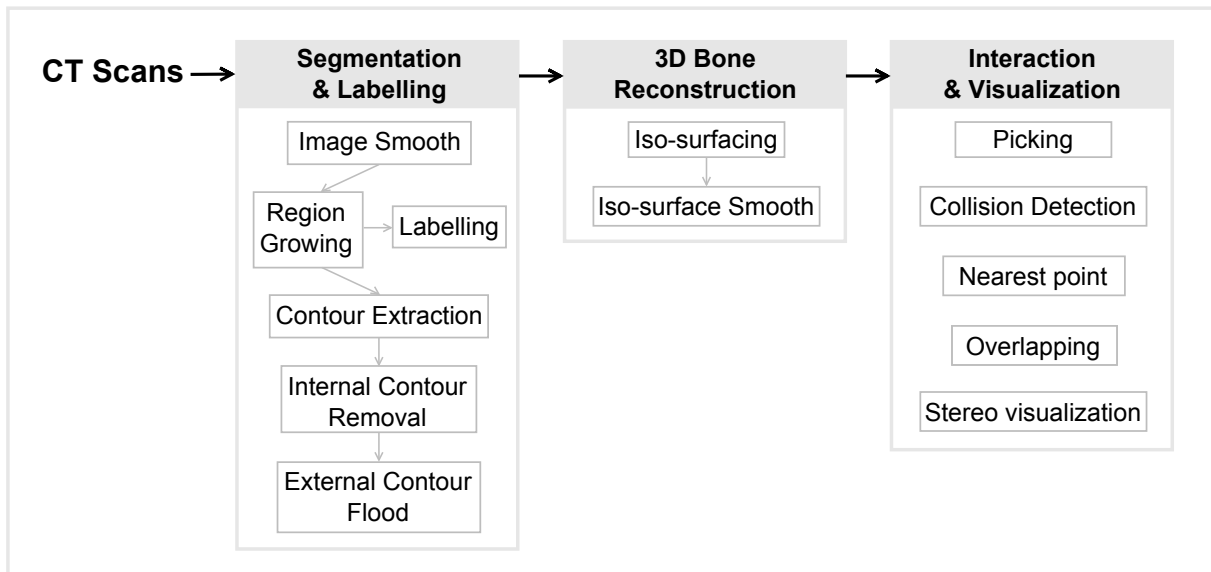


Figure 1: General overview of the process.

to evolve, thereby preserving sharp boundaries between objects.

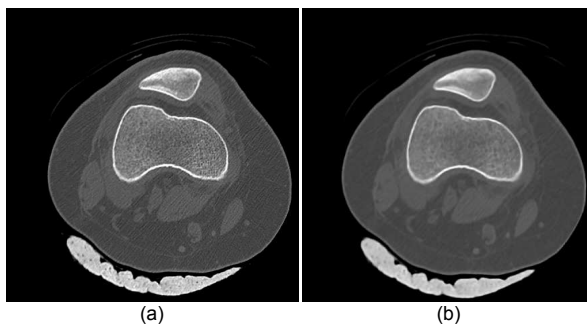


Figure 2: Smoothing a CT image. a) Original CT image. b) CT image obtained from smoothing.

### 3.2 Segmentation

We take advantage of these smoothed images to perform a particular region growing segmentation. This segmentation is based on setting a seed for expanding a region into neighbouring points. It is performed according to a threshold range of intensity values (see Algorithm 1). It also requires to know the location of a set of seeds, one for each bone regions of interest in the image (see Fig. 3a). Thus, a set of seeds is automatically established by seeking high density values, which correspond to cortical bone tissue, in the image (see Algorithm 2). Then, these seeds are finally classified by the user interactively, to validate or discard them. Furthermore, the user can interact with the image to select the seed location in order to achieve more accurate results.

Once the user has validated a seed, it is propagated along the image stack to identify the surrounding cortical tissue. This procedure reduces the amount of time

and effort required during the segmentation. The seed propagation consists in analysing the  $n \times n$  neighbouring values (being  $n$  a predefined neighbourhood size) in the adjacent slice to determine the possible location of new seeds.

**Data:** Set of slices and seed points.

**Result:** Set of Bone regions for each slice.

**THRESHOLD\_RANGE[2]**  $\leftarrow$  Minimum and maximum threshold values;

**for each Slice  $S_i$  do**

**for each seed  $s_j$  in Slice  $S_i$  do**

        Region  $R_j \leftarrow \emptyset$ ;

        CandidatePoints  $\leftarrow \emptyset$ ;

        Add seed  $s_j$  to CandidatePoints;

**while** CandidatePoints  $\neq \emptyset$  **do**

            Point  $p \leftarrow$  Next point in CandidatePoints;

            Add  $p$  to Region  $R_j$ ;

            // Neighbourhood Analysis

**for**  $k \leftarrow 1$  to 8 **do**

**if** neighbour value  $nv_k \in$

**THRESHOLD\_RANGE** **then**

**if** neighbour point  $np_k \notin R_j$  **then**

                            Add  $np_k$  to CandidatePoints;

**end**

**end**

**end**

            Remove  $p$  from CandidatePoints;

**end**

**end**

**end**

Return Set of Regions  $R$  for each Slice  $S$

**Algorithm 1:** Threshold Connected Region Growing Algorithm

**Data:** Slice.  
**Result:** Set of potential seeds.  
 THRESHOLD\_RANGE[2]  $\leftarrow$  Minimum and maximum threshold values;  
 $S \leftarrow \emptyset$ ; // Seeds  
 $I \leftarrow \text{SlicePointSet}$ ;  
 $A \leftarrow \emptyset$ ; // Auxiliary Point Set  
**for**  $i \leftarrow 1$  **to** *Slice Width* **do**  
     **for**  $j \leftarrow 1$  **to** *Slice Height* **do**  
         **if**  $\text{Point } P_{ij} \in \text{THRESHOLD\_RANGE}$  **then**  
             **if**  $P_{ij} \notin A$  **then**  
                 Add  $P_{ij}$  to  $S$ ;  
                  $A \leftarrow \text{AURegion Growing Segmentation}$   
                 from  $P_{ij}$ ;  
             **end**  
         **end**  
     **end**  
**end**  
 Return Set of Seeds  $S$

**Algorithm 2:** Automatic Search of Potential Seeds

### 3.3 Labelling

The bone labelling is performed at the same time that the user validates the seeds during the segmentation. This is a semi-automatic process which starts associating seeds to each single bone structure. The user can group more than one seed to identify branching bone structures. This 2D identification procedure allows to classify the bone structures avoiding the post-processing of isosurfaces. This post-processing usually involves a topological study for classifying groups of connected triangles.

### 3.4 Contour and region extraction

Contour extraction is performed to identify the boundaries of a bone region. Two methods are mainly used when dealing with medical images: Marching Squares and Edge Tracking. These methods are easy to implement and take into account an isovalue to determine the appearance of boundaries and generate contour lines.

On the one hand, Marching Squares is considered as a two-dimensional version of the Marching Cubes algorithm. It is based on a divide-and-conquer approach, studying every cell composed of four points in a slice to detect boundaries (see Fig. 3). On the other hand, Edge Tracking detects edge intersections. Once a boundary point has been detected, a contour line is generated moving across cell boundaries until the contour line closes back on itself [ZLZ08]. As opposed to Marching Squares, this algorithm always provides closed contour lines. Both the Marching Squares and the Edge Tracking algorithms use interpolation techniques to get more accurate results.

In this case, even though closed contours are required to delimit the different bone regions, Marching Squares

has been used because of its simplicity (see Fig. 5b). In fact, both of them can be used since the result of the contour lines are always closed due the features of the region growing segmentation as we described above.

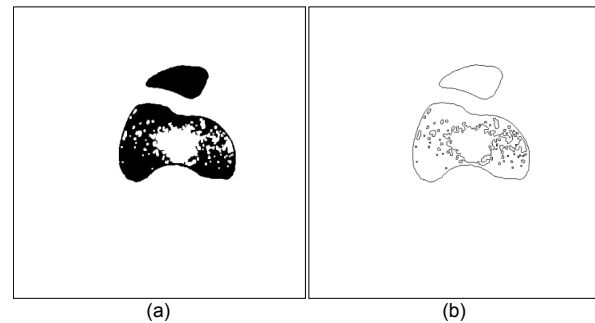


Figure 3: Smoothed CT image processing. a) Region growing segmentation result. b) Contour extraction result.

Finally, bone regions are generated by flooding the extracted contours. Since each region is represented by external and internal contours (see Fig. 3b), it is possible to reduce the processing time by using only the external contours. For that, we take advantage of the fact that contours generated during the segmentation are always closed. Therefore, the contours are treated as polygons and the Jordan Curve Theorem is used to detect the inner contour points (see Algorithm 3)(see Fig. 4a). This algorithm is also utilized to flood the external contours (see Fig. 4b). It is important to note that the flooding process is required to classify the inner pixels as bone tissue and to enable the 3D bone reconstruction.

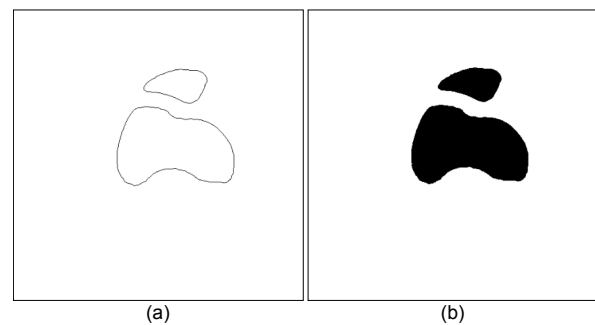


Figure 4: Segmentation from closed contours. a) Removal of internal contours. b) Flood of external contours.

## 4 3D BONE MODELS RECONSTRUCTION

Once the region segmentation is performed, three-dimensional bone models are generated by isosurfacing. Using the Marching Cubes algorithm, the 3D bone models are reconstructed. The results, which correspond to an area of the knee, are shown in Figure 6b. This comparative also shows the removal of non-desirable geometry. This geometry can appear

**Data:** Set of contours.  
**Result:** Set of external contours.  
 $EC \leftarrow \emptyset$ ; // External Contour Set  
 $I \leftarrow SlicePointSet$ ;  
**for each Contour**  $C_i$  **do**  
     $inside \leftarrow false$ ;  
     $numContour \leftarrow 1$ ;  
     $numContours \leftarrow$ Number of Contours;  
    **while**  $numContour < numContours$  **and**  $!inside$  **do**  
        **if**  $C_i \neq C_{numContour}$  **then**  
             $numPoint \leftarrow 1$ ;  
             $numPoints \leftarrow$ Number of Points of  $C_i$ ;  
            **while**  $numPoint < numPoints$  **and**  $!inside$  **do**  
                Point  $P_{x,y} \leftarrow Point_{numPoint}$  of  $C_i$ ;  
                 $crossings \leftarrow 0$ ;  
                **for each Line Segment**  $LS_j$  of  $C_{numContour}$  **do**  
                    **if** ray down from  $P_{x,y}$  crosses  $LS_j$  **then**  
                         $crossings \leftarrow crossings + 1$ ;  
                    **end**  
                **end**  
                **if**  $crossings$  is odd **then**  
                     $inside \leftarrow true$ ;  
                **end**  
            **end**  
        **end**  
        **if**  $!inside$  **then**  
            Add  $C_i$  to  $EC$ ;  
        **end**  
    **end**  
**end**  
Return Set of External Contours  $EC$   
**Algorithm 3:** Extraction of External Contours

due to the presence of internal noise (see Fig. 5 and artefacts 6a). In addition, a brief comparative about the resolution of the generated models is shown in Table 1. The 3D models generated are cleaner and have lower resolution. Moreover, an additional slice has been added above and below of the three-dimensional image in order to avoid the presence of holes and to obtain models totally closed (see Fig. 6b).

Finally, a Laplacian smoothing filter has been applied to the 3D bone models. The models are smoothed adjusting the point coordinates, making the cells better shaped and the vertices more evenly distributed. It is achieved analysing each vertex to determine which other vertices are connected with it. Then, the coordinates of each vertex are modified according to the average of the connected vertices. This process is repeated until the desired result is obtained. On the other hand, lower resolution models can be generated by simplifying the contours obtained during the segmentation step.

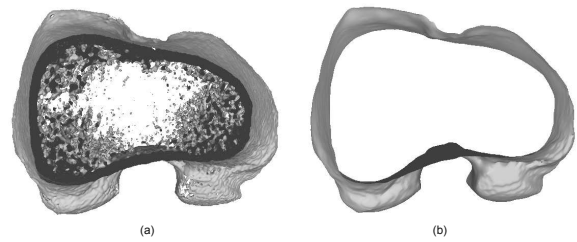


Figure 5: View from axial plane of femur isosurfaces: a) isosurface obtained directly from original CT images, b) isosurface obtained from region segmentation.

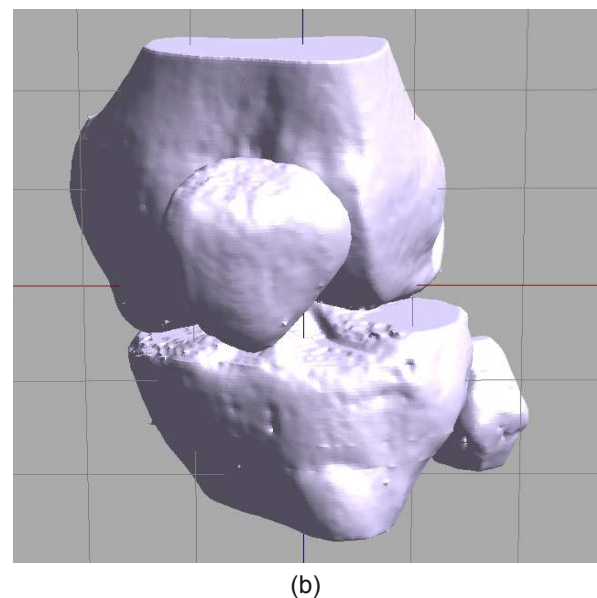
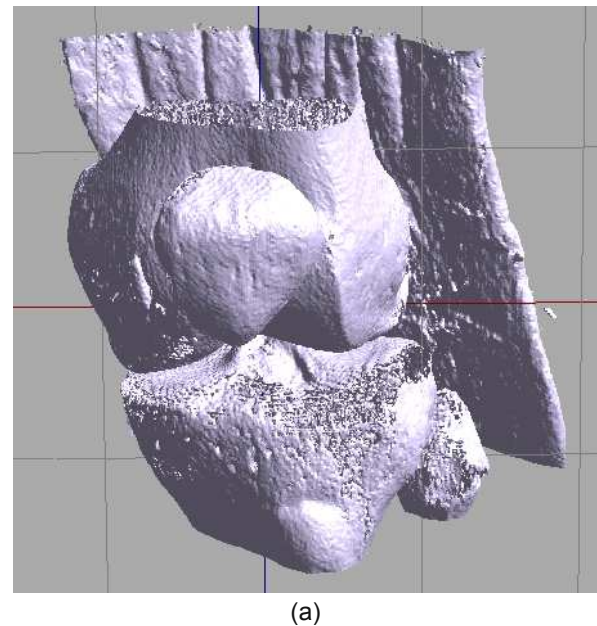


Figure 6: Results obtained by isosurfacing: a) isosurface obtained directly from original CT images, b) simplified and labelled isosurfaces obtained from region segmentation.

Knee bones	Direct isosurfacing	Isosurfacing after segmentation	Reduction (%)
Points	1.609.968	205.895	87,21%
Triangles	3.180.573	408.176	87,17%

Table 1: Comparative of the geometry generated by isosurfacing from previous segmentation

## 5 INTERACTION & VISUALIZATION

Once the 3D bone models have been reconstructed, the interaction is enabled. For that reason, it has been necessary to implement picking and collision detection methods. In addition, due to the size and the topology of the reconstructed models, those methods must be fast and robust. 3D bone models can also be selected, translated and rotated to be placed correctly in the scene. To simplify this step, the calculation of nearest points and the overlapping triangles between every two bone models is performed. These methods have been implemented in C++ using PQP - A Proximity Query Package [LGLM99] to calculate a detailed collision detection. Furthermore, the Leonar3Do system [Leo12] has been utilized in order to improve the interaction and visualization. This is a virtual reality system that enables a stereo visualization and interaction by using active glasses.

### 5.1 Collision detection

In order to implement the interaction with the different 3D bone models, we have not only to calculate the collision but also to generate a realistic response. Since the reconstructed models are isosurfaces, it is necessary to use algorithms that can work with that type of models. Distances and nearest points between two models can help the user to place models that are not correctly positioned (see Fig. 7). Some algorithms have been tested [PJPO12] to choose the best suited to this problem. In this work, algorithms have been tested with models with a complexity of up to one million triangles. The results show that the PQP library [LGLM99] is quite fast and robust. This library uses swept sphere hierarchies to perform collision detections. Moreover, PQP is able to calculate different collision detection parameters, such as distances, overlapping triangles (see Fig. 8), or closest points.

### 5.2 3D picking

The set of 3D bone models can be selected by picking. Each selected model can be rotated and translated by using the mouse. These transformations are performed independently of the other models in the scene. In order to pick bone models, we have implemented a method that consists of casting a ray from the observer that passes through the mouse position and the calculation of the collision with all the objects in the scene by

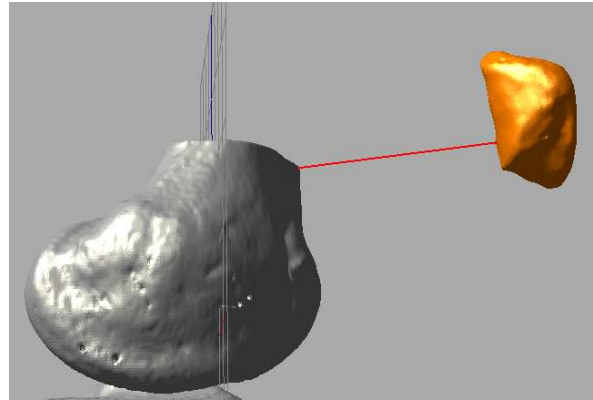


Figure 7: Calculation of the nearest points between 3D models.

using the PQP library. In this way, the data structures previously constructed are used. This method is robust since it is based on PQP.

### 5.3 Stereo Visualization and Immersion

The Leonar3Do system mainly consists of a spatial input device (called "bird"), 3D glasses and monitor-mounted sensors. Both the bird and the 3D glasses operate in six degrees of freedom and sensors can track both the bird and the glasses. To this end, the Leonar3Do system uses a technology based on infrared sensors. The bird has two buttons, one big and one small, which can be programmed. In addition, the big button is sensitive to pressure. On the other hand, 3D glasses enable stereoscopic vision.

This visualization system allows the user to better understand the scene and makes easy to select objects by using the spatial input device. In addition, the stereo visualization increases the feeling of immersion when manipulating the 3D bone models (see Fig. 9).

The Leonar3Do API (LeoAPI) is divided into two main parts: the tracking API and the stereoscopic API. The first one allows to know where both the spatial input device and the glasses are situated. The spatial input device has been used to manage a 3D cursor that enables the selection of bone models in the scene and to manipulate them. On the other hand, the position of the glasses is used to manage the camera, hence when the user moves the head, the camera moves accordingly. To do this, both the translation and the rotation of the user head are considered. This increases the feeling of immersion and makes the application easy to use. It is

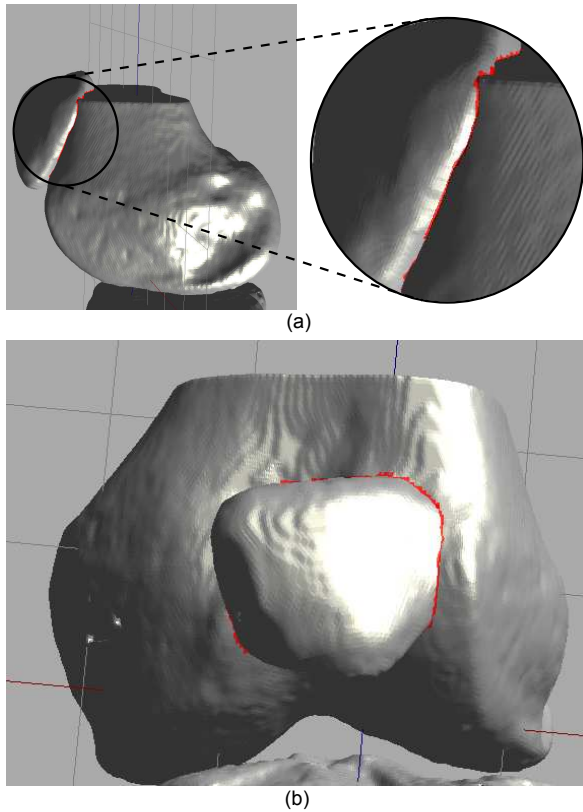


Figure 8: Overlapping triangles are displayed in red. a) Overlapping triangles between the patella and femur models. b) Overlapping triangles in (a) from a different perspective.

important to consider that the camera position is modified by the LeoAPI, hence that it is placed at the focal point. The stereoscopic API enables the implementation of the stereo view and it is responsible for generating the final frame of the application, whether or not the stereo mode is activated. Because of this responsibility, it is necessary to disable the auto-buffer swapping because the LeoAPI already does it. Moreover, the LeoAPI fills a projection matrix which have to be provided to OpenGL before rendering the canvas. However, this projection matrix can be modified before passing it to the OpenGL pipeline.

## 6 CONCLUSIONS AND FUTURE WORK

An approach to reconstruct 3D simplified bone models and to enable interaction with them has been presented. A semi-automatic approach has been used to identify bone tissue from CT images. The proposed segmentation method requires extra processing time but in return it allows to label the different bone structures. The 3D reconstruction has been performed by isosurfacing. Comparing our results with traditional isosurfacing approaches, the obtained 3D models are less complex and contain no noise due to the previous segmentation.

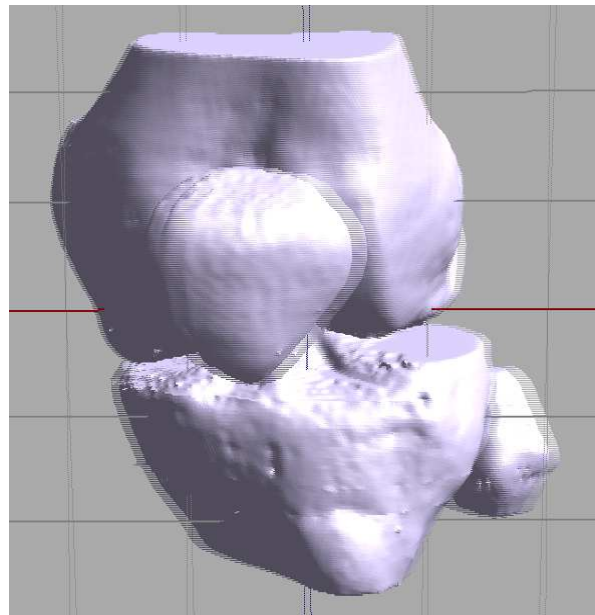


Figure 9: Stereo visualization with Leonar3Do system.

Consequently, artefacts are also avoided during the reconstruction.

Furthermore, the stereo Leonar3Do system and computer graphic techniques have been integrated to enable interaction with the reconstructed 3D models in terms of geometry. This interaction includes a detailed collision detection that helps to provide visual feedback to the user. Moreover, the Leonar3Do system is used to provide an immersive experience that enables a stereo visualization. As a result, our proposal allows users to interact with individual 3D bones for different applications such as education and surgical planning.

As future work, we plan to apply this approach in the reconstruction of bones and fragments involved in a fracture. The reconstructed models can be used for surgery planning and virtual fracture reduction. For that purpose, the segmentation algorithm should be extended to deal with fractured bones. Moreover, the variation of bone intensity level between adjacent medical images can lead to changes in the shape of bone regions that provokes stepped boundaries during the reconstruction. These sharp boundaries are quite difficult to be smoothed even though the vertices of the models can be adjusted as we described in Section 4. Therefore, other approaches should be studied in order to perform the triangulation between contours.

## ACKNOWLEDGEMENTS

This work has been partially supported by the Ministerio de Economía y Competitividad and the European Union (via ERDF funds) through the research project TIN2011-25259.

## 7 REFERENCES

- [BD02] BHANIRANTKA P., DEMANGE Y.: Opengl volumizer: a toolkit for high quality volume rendering of large data sets. In *IEEE/ACM SIGGRAPH Symposium on Volume Visualization and Graphics* (2002), pp. 45–53.
- [BHGS06] BOUBEKEUR T., HEIDRICH W., GRANIER X., SCHLICK C.: Volume-surface trees. *Computer Graphics Forum (Proc. of EUROGRAPHICS 2006)* 25, 3 (2006), 399–406.
- [HBG\*07] HARDERS M., BARLIT A., GERBER C., HODLER J., SZÉKELY G.: An optimized surgical planning environment for complex proximal humerus fractures. In *MIC-CAI Workshop on Interaction in Medical Image Analysis and Visualization* (2007).
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics 21*, 4 (1987), 163–169.
- [Leo12] LEONAR3DO:, 2012. <http://www.leonar3do.com> [Online accessed 31-March-2014].
- [LGLM99] LARSEN E., GOTTSCHALK S., LIN M., MANOCHA D.: *Fast Proximity Queries with Swept Sphere Volumes*. Tech. rep., Dep. of Computer Science, UNC Chapel Hill, 1999.
- [LYG\*09] LV S., YANG X., GU L., XING X., PAN L., FANG M.: Delaunay mesh reconstruction from 3d medical images based on centroidal voronoi tessellations. In *Int. Conference on Computational Intelligence and Software Engineering* (2009), pp. 1–4.
- [PJP12] PULIDO R., JIMÉNEZ J. J., PAULANO F.: Surface reconstruction from 3d medical images based on tri-tree contouring. In *Proc. of the International Conference on Computer Graphics Theory and Applications* (2012), pp. 175–181.
- [PJPO12] PAULANO F., JIMÉNEZ J. J., PULIDO R., OGAYAR C. J.: A comparative study of implemented collision detection strategies. In *Proc. of the International Conference on Computer Graphics Theory and Applications* (2012), pp. 485–490.
- [PXP00] PHAM D. L., XU C., PRINCE J. L.: Current methods in medical image segmentation1. *Annual Review of Biomedical Engineering* 2, 1 (2000), 315–337.
- [Sac04] SACHSE F.: 5. digital image processing. In *Computational Cardiology*, vol. 2966 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, pp. 91–118.
- [Set99] SETHIAN J.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1999.
- [SLS\*07] SHARF A., LEWINER T., SHKLARSKI G., TOLEDO S., COHEN-OR D.: Interactive topology-aware surface reconstruction. *ACM Trans. Graph.* 26, 3 (July 2007).
- [SRS\*10] SHARMA N., RAY A. K., SHUKLA K. K., SHARMA S., PRADHAN S., SRIVASTVA A., AGGARWAL L.: Automated medical image segmentation techniques. *Journal of medical physics* 35, 1 (2010), 3–14.
- [vdB04] VAN DEN BERGEN G.: *Collision Detection in Interactive 3D Environments: Text*. Collision Detection in Interactive 3D Environments. Morgan Kaufmann Publishers, 2004.
- [WMW86] WYVILL G., MCPHEETERS C., WYVILL B.: Data structure for soft objects. *The Visual Computer* 2, 4 (1986), 227–234.
- [WVW\*04] WOLF I., VETTER M., WEGNER I., NOLDEN M., BOTTGER T., HASTEN-TEUFEL M., SCHOBINGER M., KUNERT T., MEINZER H.-P.: The medical imaging interaction toolkit (mitk) a toolkit facilitating the creation of interactive software by extending vtk and itk. In *Medical Imaging 2004* (May 2004), International Society for Optics and Photonics, pp. 16–27.
- [YPH\*06] YUSHKEVICH P. A., PIVEN J., HAZLETT C. H. C., SMITH C. R. G., HO C. S., GEE J. C., B G. G.: User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage* 31 (2006), 1116–1128.
- [ZLZ08] ZHENG L., LI G., ZHU H.: The generation algorithm of tissue contour lines in medical image. In *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on* (2008), pp. 2631–2634.