Initial Results of a Method for the Generation of Triangle Meshes Representing Bone Fragments using a Spatial Decomposition

Félix Paulano-Godino and Juan J. Jiménez-Delgado
Department of Computer Science, University of Jaén, Campus Las Lagunillas s/n, Jaén, Spain

Keywords: Bone Fragments, 3D Reconstruction, Spatial Decomposition, Triangle Mesh.

Abstract: The generation of a virtual representation of the bones and fragments is an artificial step required in order to obtain helpful models to work with in a simulation. Nowadays, the Marching Cubes algorithm is a de facto standard for the generation of geometric models from medical images. However, bone fragments models generated by Marching Cubes are huge and contain many unconnected geometric elements inside the bone due to the trabecular tissue. The development of new methods to generate geometrically simple 3D models from CT image stacks that preserve the original information extracted from them would be of great interest. In order to achieve that, a preliminary study for the development of a new method to generate triangle meshes from segmented medical images is presented. The method does not modify the points extracted from CT images, and avoid generating triangles inside the bone. The aim of this initial study is to analyse if a spatial decomposition may help in the process of generating a triangle mesh by using a divide-and-conquer approach. The method is under development and therefore this paper only presents some initial results and exposes the detected issues to be improved.

1 INTRODUCTION AND BACKGROUND

Computer-assisted fracture reduction procedures usually require the generation of 3D models representing bone fragments. Although the generated models could be point clouds or volumes, most of the times these models are meshes, since computer-assisted techniques require geometric algorithms (Jiménez et al., 2016). In particular, interactive environments usually require triangle meshes in order to apply collision detection strategies. Computed Tomography (CT) is the most appropriate and accessible data acquisition technique for distinguishing bone from other tissues and thus, for generating 3D bone fragment models (Egol et al., 2010). This type of image stores at each pixel the radio-density obtained during the CT scan as intensity values in the image. The morphology of bone fragments makes the generation of triangle meshes difficult, since the shape of the segmented regions is very irregular because trabecular tissue appears in the outer part of the bone fragment due to the fracture. These features influence the choice of a method for generating meshes.

Nowadays, the Marching Cubes (MC) algorithm (Lorensen and Cline, 1987) is a de facto standard for generating triangle meshes representing bone fragments to be used in computer-assisted procedures (Jiménez et al., 2016). The MC algorithm has been utilized to generate meshes that represent acetabular fractures (Fornaro et al., 2008)(Fornaro et al., 2010a), fractured pelvic bones (Fornaro et al., 2010b) (Lee et al., 2012) and proximal humerus fractures (Fürnstahl et al., 2012). This algorithm has also been used to generate fragment models to represent highly fragmented bone fractures (Willis et al., 2007) and to provide models for a virtual orthopaedic surgery simulator (Tsai et al., 2001). The size of the models generated by these methods depends on the resolution of the medical images and the complexity of the segmented bone structures; hence models generated by MC are usually huge and contain unconnected geom-

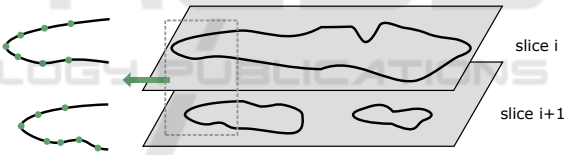

Figure 1: Simplified representation of the case in which contours from two consecutive slices do not have a one-to-one correspondence. Left, zoomed area showing the difficulty of deciding which points should be sewed together in 3D.

Paulano-Godino F. and JimÃľnez-Delgado J.
Initial Results of a Method for the Generation of Triangle Meshes Representing Bone Fragments using a Spatial Decomposition.
DOI: 10.5220/0006266302980305
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 298-305
ISBN: 978-989-758-224-0

etry inside the bone due to the morphology of trabecular tissue (Paulano et al., 2015). Therefore, generated meshes commonly need to be post-processed to improve their features. An alternative approach consists of generating a 3D model by triangulating external contours of the bone fragments extracted from medical images. However, this approach involves the resolution of some complex problems, such us deciding which points should be sewn together in 3D, or managing sewing when the contours are subdivided in consecutive slices (Figure 1). The use of spatial decompositions (Samet, 2010) allows dividing the point cloud into small sets of points, reducing the mesh generation problem to smaller sub-problems.

In this paper, an initial study for the development of an alternative method to generate triangle meshes from medical images using a spatial decomposition is presented. In this study, we decided to use a spatial decomposition named tetra-tree (Jiménez et al., 2006), since conducted studies demonstrated that it presents several advantages for interactive environments with regards to other data structures based on rectangular cells such as the octree (Chen and Huang, 1988). Mainly, the tetra-tree makes the selection of related triangles easier, being able to select inaccessible parts without the use of complementary algorithms (Jiménez et al., 2011). Section 2 describes the main stages of the proposed method. Then Section 3 discusses the main advantages of using a tetra-tree to generate triangle meshes and analyses the issues to improve in the proposed method.

2 PROPOSED METHOD

2.1 Overview

The first stage of the proposed mesh generation method consists of extracting contour points from medical images. Then, a spatial decomposition named tetra-tree (Jiménez et al., 2006) recursively divides the space into tetrahedra with a shared origin named tetra-cones, so that these tetra-cones cover the whole space without overlapping. As the tetra-tree is built, the contour points are classified in the tetra-cones. After that, triangle patches are generated using the points classified in each tetra-cone. Finally, triangle patches are hierarchically sewn in order to obtain the reconstructed model. Figure 2 summarizes the stages of the proposed method.

With the aim of easing the understanding of the proposed algorithms, a simple model representing a patella (Figure 3) will be used to exemplify the process in the following subsections.

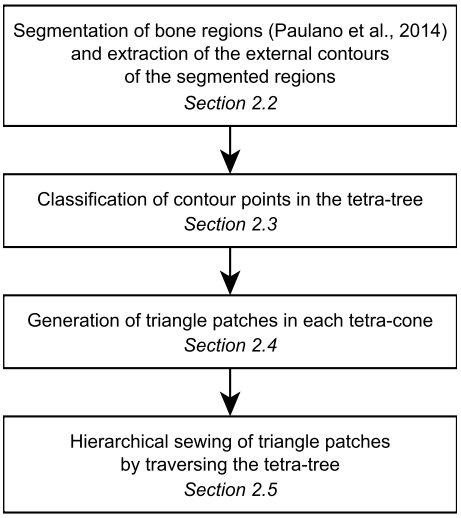

Figure 2: Overview of the proposed mesh generation method.

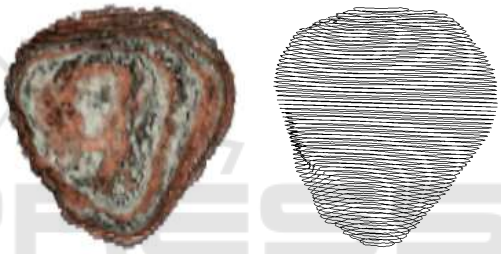

Figure 3: Left, volume visualization of a patella segmented from CT images. Right, external contours extracted from the segmentation results.

2.2 Segmentation and Extraction of External Contours

The input of our mesh generation method is a point cloud representing the outer part of bone fragments. These point clouds must be extracted from the information available in medical images.

In a first step, all the bone fragments are segmented from CT scans. For that purpose, the segmentation algorithm described in (Paulano et al., 2014) is used. The algorithm generates a region for each bone fragment in each slice (Figure 3, left). Unlike other approaches, this algorithm is able to separate wrongly joined bone fragments during the segmentation process. Then the external contour of each segmented region is extracted. With that aim, an approach similar to the proposed in (Pulido et al., 2014) is applied. First, the Marching Squares algorithm is utilized in order to generate contours from the segmented regions (Ho et al., 2005). This algorithm can be considered as a 2D adaptation of MC. The algo-

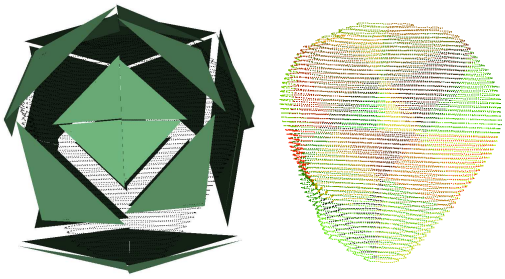

Figure 4: Left, a tetra-tree associated to the contour points. Right, points belonging to each tetra-cone are shown in a different colour.

rithm does not ensure that the generated contours are closed, but this is not relevant in this case since the final goal is to generate point clouds. Due to the noise present in the regions because of trabecular tissue, inner contours may be obtained. These inner contours are removed using the point-in-polygon algorithm by Feito and Torres (Feito and Torres, 1997); hence only the outer contour of each fragment is extracted in each slice. After removing internal contours, the vertices of each external contour are grouped in order to form the point cloud used as input by the mesh generation algorithm (Figure 3, right).

2.3 Classification in the Spatial Decomposition

A tetra-tree (Jiménez et al., 2006) is a spatial decomposition technique that recursively divides the space into tetra-cones with a common origin, so that these tetra-cones cover the entire space without overlapping among them (Figure 5, left). As the tetra-tree is built, the points are classified in the generated tetra-cones (Figure 4, right). Then the tetra-tree is adapted in order to fit the classified geometric elements. For that purpose, the enveloping tetrahedron associated with each tetra-cone is calculated. The enveloping tetrahedron of a given tetra-cone is the smallest tetrahedron that shares the origin and the lateral faces with that tetra-cone, and contains all the points that are classified into it (see Figure 5, right). Since a tetra-cone is defined by three planes that intersect in a point, the top cover of each tetrahedron is a triangle. This top cover is perpendicular to the segment that goes from the centroid of the model to its incircle. The distance from the centroid of the model to the top cover is established by the point classified in the tetra-cone which is furthest to the centroid. Figure 4 (left) shows an adapted tetra-tree associated with the point cloud representing the patella.

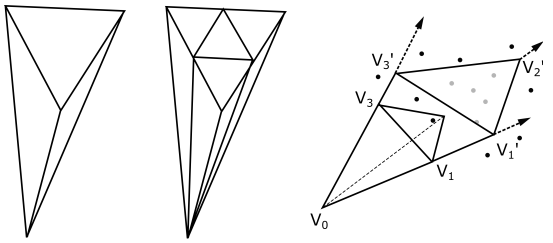

Figure 5: Left, a representation of the subdivision of a tetra-cone. Right, a schema that represents the bounding tetrahedra associated with a tetra-cone (Jiménez and Segura, 2008).

2.4 Generation of Patches

The classification of all the contour points in the tetra-tree enables the execution of a divide-and-conquer approach, so that the points classified in each tetra-cone are triangulated in order to generate patches. Nevertheless, not all the points classified in a tetra-cone must be triangulated together. In order to avoid generating incorrect triangles, the points are clustered before being triangulated. A Euclidean cluster extraction algorithm has been used to group the points that must be triangulated together. With this algorithm, the points are grouped whenever they can be connected by points closer than a pre-defined threshold. In the case of the patella that we use as example, only one cluster is generated for each tetra-cone.

Once the clusters have been generated, the points belonging to each cluster are used to generate patches. The proposed method triangulates points belonging to consecutive contours. In order to perform the triangulation, the connectivity information of the contours is used. First, the contour points are sorted and grouped by slices. Then the contours of each slice are divided into connected polylines. After that, the polylines belonging to consecutive slices are triangulated. Depending on the number of polylines in which each contour has been divided, three different triangulation cases are considered: one-to-one, one-to-two and two-to-two. Any other case is discarded and thus not triangulated during this stage of the algorithm. Depending on each case, a different triangulation algorithm is applied.

2.4.1 One-to-One Triangulation

The easiest case occurs when both contours are composed by only one polyline. In this case new triangles are generated, minimizing the number of triangles with large sides and small angles. Given the first two points of each polyline A_0, A_1, B_0, B_1, the algorithm first computes the distance from A_0 to B_1 and the distance from B_0 to A_1. The smaller of these two

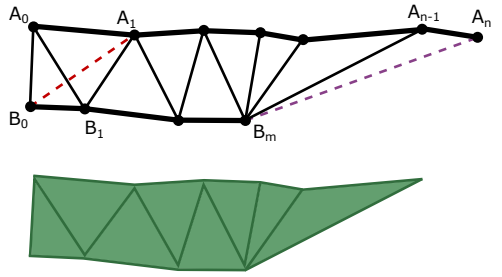

Figure 6: Top, example of a one-to-one triangulation of polylines using the proposed algorithm. Bottom, obtained triangles.

distances defines the segment which is used to generate the triangle. In Figure 6, the red coloured triangle $A_0B_0A_1$ is discarded since the distance between B_0 and A_1 is bigger than the distance from A_0 to B_1. This step is repeated until all the points belonging to one of the polylines have been triangulated. Finally, the last point of the already sewn polyline is used to generate triangles with the remaining points of the opposite polyline. Considering that all the points of B have been triangulated and A_j is the next point in A to be triangulated, the algorithm tries to generate new triangles until the distance from A_j to B_m is greater than twice the average distance between A and B. This constraint provides a better triangle distribution and avoids degenerated triangles. In Figure 6, the purple coloured triangle $A_{n-1}B_mA_n$ is discarded because the distance from B_m to A_n is bigger than twice the average distance between A and B.

2.4.2 One-to-Two Triangulation

If one of the contours is composed by two sub-polylines - A and B -, these must be sorted before being sewn. Then the sub-polylines are sequentially sewn to the polyline C of the opposite contour. Since the two sub-polylines are sewn using the same process, the algorithm to sew a single sub-polyline A to a polyline C is explained for simplicity. Firstly, the polyline point C_i closest to the first point of A is calculated. In Figure 7, the closest point to A_0 is coloured in purple. After that, the one-to-one algorithm is applied but starting from the closest point previously calculated. Using that algorithm, new triangles are generated until all the points of A have been triangulated. Then the last point of A is used to generate new triangles. The main difficulty consists of deciding when to stop sewing a sub-polyline and to start the next one. Being C_j the next point of the polyline to be triangulated, the algorithm stops sewing the sub-polyline A when the distance from the last point of A to C_j is greater than twice the average distance between A and C and it is also bigger than the distance from the first point of B to C_j. As in the one-to-one case, these constraints provide a better triangle distribution and avoid degenerate triangles. Once the first sub-polyline A has been sewn, the same procedure is repeated for the second sub-polyline B. In Figure 7, the closest point to B_0 is coloured in blue. Being C_k the next point of C to be triangulated, in this case new triangles are generated until the distance from the last point of B to C_k is larger than twice the average distance between B and C or all the points have been triangulated. In Figure 7, the red coloured triangle $C_{l-1}B_mC_l$ is discarded because the distance from B_m to C_l is bigger than twice the average distance between B and C.

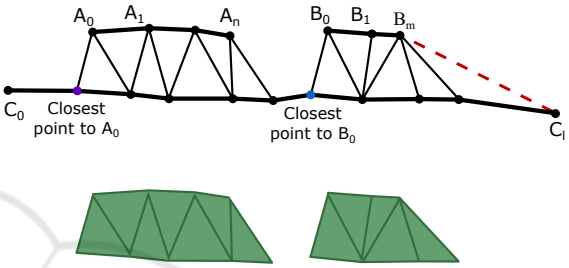

Figure 7: Top, example of a one-to-two triangulation of polylines using the proposed algorithm. Bottom, obtained triangles.

2.4.3 Two-to-Two Triangulation

The two-to-two case requires deciding which sub-polylines have to be sewn together. Two sub-polylines are sewn if the average distance between them is less than a pre-defined threshold. This value is manually set and mainly depends on the distance between slices in the source medical images. The order in which the sub-polylines are sewn is also determined by the distance between them. Firstly, the system tries to sew together the two closest sub-polylines, and then the two remaining sub-polylines. In both cases, the one-to-one algorithm (Figure 6) is used in order to sew the two sub-polylines.

The triangulation algorithm described above must be repeated in order to generate patches for each cluster. Figure 8 shows the triangle patches generated by the triangulation algorithm in the case of the patella.

2.5 Sewing Triangle Patches

The triangulation algorithm generates one or more triangle patches for each tetra-cone. In order to close the mesh, these patches need to be sewn together. For that purpose, the tetra-tree hierarchy is traversed in post-order. For each node of the hierarchy, the patches stored in its 4 children nodes that are closer than a

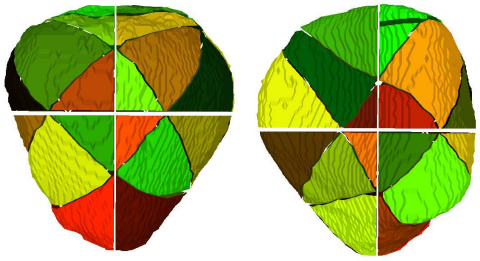

Figure 8: From left to right: front and back views of the triangle patches generated by applying the triangulation algorithm to each cluster in the leaf nodes of the hierarchy. Each patch is displayed in a different colour.

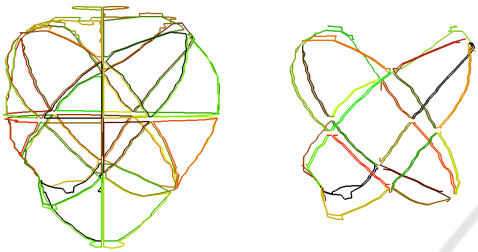

Figure 9: Left, external contours of the triangle patches generated in each leaf tetra-cone. The external contour of each patch is displayed in a different colour. Right, pairs of polylines that are triangulated together in the last level of the tetra-tree. Each pair is displayed in the same colour.

previously defined threshold are sewn. As occurs in the patch generation algorithm, this threshold value is manually chosen and depends on the point distribution, and thus on the distance between slices and the resolution of the source medical images. The order in which the patches are sewn is defined by the distance between patches and closer patches have priority.

The algorithm proposed to sew two patch contours starts with the identification of the points to be sewn. For that purpose, the external edges of each patch are calculated (Figure 9, left) as the edges that only belong to one triangle. Then these edges are projected onto the plane defined by the top cover of the tetra-cone containing them. In order to avoid overlapping of the projection of the two patches, the patches are separated by the displacing them the distance between the centroids of both patches, in the direction defined by both centroids. In 2D, the edges to be sewn are defined as those whose points can be connected to a point belonging to the other patch without intersecting any of the two patches (Figure 10). Figure 9 (right) shows the polylines to be sewn between every two patches in the last level of the tetra-tree for the case of the patella. Finally, new triangles are generated using the one-to-one method to sew two polylines already described in Section 2.4.

This sewing procedure is repeated in each node of

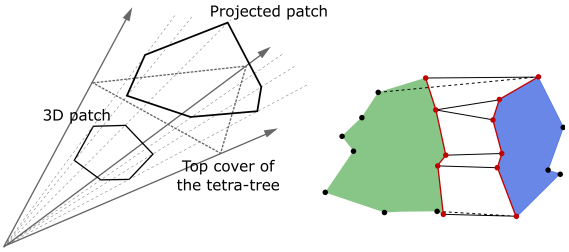

Figure 10: Left, schema that represents the linear projection of the external contours of a triangle patch onto the plane defined by the top cover of its associated tetra-cone. Right, schema representing the selection of the edges to be sewn between two patches. Solid lines connect points of edges that belong to the group of edges to be sewn group since they do not intersect with any patch. Dotted lines intersect with the patch, and therefore their associated edges are discarded.

the hierarchy. Once all the nodes of a level have been processed, the generated patches are transferred to the parent node and the sewing algorithm is repeated at the upper level. The process is repeated until reaching the root node of the tetra-tree. The top image of Figure 11 shows the patches generated in each of the tetra-cones belonging to the top level of the tetra-tree.

At the top level, the parent node receives the patches generated in its eight children nodes. These patches have to be sewn together in order to close the mesh. In this step of the algorithm, patches are sewn two by two in a sequential process. Figure 11 summarizes the three stages of this process.

At the first two stages of the process (Figure 11, top and middle), the criteria used to select the points to be sewn is that each point must be sewn to its closest patch. Given a pair of patches, P_a and P_b, the points to be sewn in P_a are those whose closest patch is P_b. The points to be sewn in P_b are calculated using the same procedure. Then patches are sewn using the one-to-one algorithm previously described to sew polylines. Due to the holes associated with the first and the last slices of the stack, two sub-polylines may be obtained when calculating the edges to be sewn in some cases. To deal with the sewing in these cases, the two-to-two algorithm is applied.

At the last stage of the process (Figure 11, bottom), only two patches are remaining. At this stage, all the external edges of both patches belong to the polylines to be sewn, and therefore, the algorithm to extract external contours is applied to get them. Since the polylines representing the external contours are closed, the one-to-one sewing algorithm requires being adapted to properly join the two patches. As mentioned above, the generated model contains two holes associated with the contours in the first and the last slices of the stack. Nevertheless, these two holes

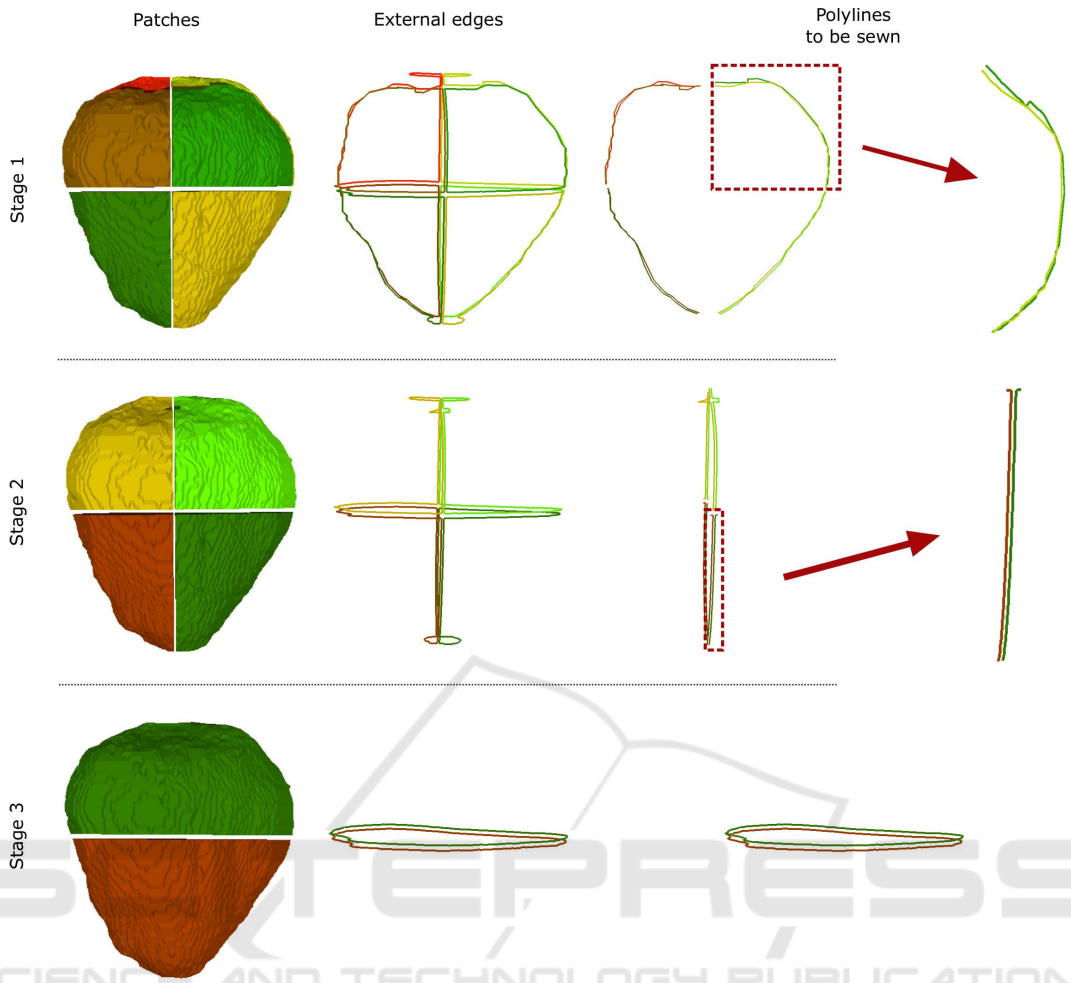

Figure 11: Left, triangle patches received as input in each of the three stages carried out at the top level of the tetra-tree. Centre, external edges of those patches. Right, calculated polylines to be sewn at each of the three stages.

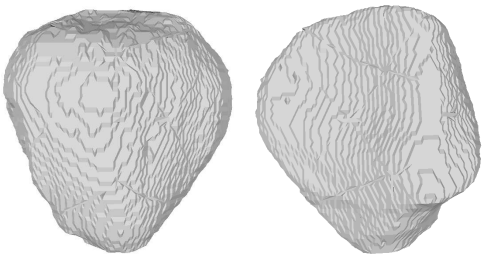

Figure 12: Front and back views of a mesh generated using the proposed procedure in the case of the patella.

do not affect to the application of the algorithm, and can be filled at the end. Figure 12 shows the results obtained by applying the method in the case of the patella.

3 DISCUSSION AND CONCLUSIONS

In this paper, a preliminary study for the development of a new method to generate triangle meshes representing bone fragments from medical images has been presented. This method uses a spatial decomposition to implement a divide-and-conquer approach that allows simplifying the problems that arise when triangulating the contours extracted from consecutive slices. Since the presented method is under development, only initial results have been shown. Next paragraphs discuss the main advantages of using a tetra-tree instead of other spatial decompositions, as well as the main aspects that should be improved in order to apply the proposed method in the generation of more complex bone fragment models.

The use of a tetra-tree provides additional properties to the generated model that encourage its uti-

lization in interactive environments (Jiménez et al., 2011). This spatial decomposition enables the visualization of the triangles that either take part or are close to taking part in an interaction, enabling the accessibility to far or non-visible parts of the generated model without having to use additional algorithms such as ray-triangle or ray-box intersection tests. Additionally, the tetra-tree provides a smooth transition when changing the level of detail during interaction. In contrast to other widely used spatial decompositions like the octree (Chen and Huang, 1988), the tetra-tree is object-centred; hence the classification of the geometry is invariant to rotations. Moreover, the construction cost is lower and the interaction times are faster than using an octree (Jiménez et al., 2011).

Although the developed algorithm produces promising results in the generation of bone fragment models, some cases in which the algorithm does not perform well have been detected. Firstly, the classification of points in the tetra-tree is not always optimal, and hinders the generation of patches in the more irregular areas, especially in the case of long bones. In addition, the classification of points often generates triangulation cases that cannot be solved using the algorithms described in Section 2.4. Finally, in the case of models with several concavities, the linear projection proposed in Section 2.5 in order to find the edges to be triangulated for sewing two patches may not be the best option.

Regarding the classification of the point cloud, the tetra-tree does not provide a homogeneous classification of the points in the case of long bone fractures. In these cases, the more detailed area, which is usually the fracture zone, is located at the ends of each bone fragment; hence the tetra-tree should be further subdivided in those areas. The research of a new spatial decomposition that fits better the shape of any bone fragment would be profitable. For that purpose, the new spatial decomposition could use additional information such as the intensity value or the estimated curvature at each point.

During the generation of patches from contours, we detected that two special triangulation cases are not resolved. On the one hand, some isolated points can be classified into one tetra-cone because of the location of the centroid and the orientation of the tetra-tree. These points divide polylines into two subpolylines, and thus a hole is obtained during triangulation. Furthermore, isolated points may cause that the triangulation algorithm described in Section 2.4 generates incorrect triangles. In order to solve this case, two different approaches could be used: developing a new triangulation algorithm that is not affected by isolated points, or reclassifying the points of the affected tetra-cone avoiding isolated points.

On the other hand, a closed small contour can be completely classified in a single tetra-cone. Therefore, that closed contour must be triangulated with the polyline extracted from the next slice. The one-to-one triangulation algorithm described in Section 2.4 is not able to properly generate triangles in that case. To address this case, two different strategies could be utilized: reclassifying the points in the tetra-tree avoiding the classification of entire contours in a single tetra-cone, or developing a new triangulation algorithm with the purpose of dealing with this case.

Once patches have been generated in the leaf tetra-cones, these are hierarchically sewn as described in Section 2.5. In order to calculate the edges to be triangulated, patches are linearly projected onto the top plane of their associated tetra-cone. This projection obtains good results in some of the testes cases. However, the linear projection of the patches onto the top plane of the tetra-tree does not work well if the bone fragments have various concavities, such as the case of femoral condyles. In order to deal with these cases, a different type of projection should be used.

In the future, new methods will be developed in order to solve the special cases detected when sewing consecutive contours. In addition, the method could be extended to work with alternative spatial decompositions that fit better the shape of any bone fragment. For that purpose, these spatial decompositions could use not only the position of the points, but also the intensity and the estimated curvature at each point, in order to adjust them better to the bone fragment in the more detailed areas like the fracture zone. Finally, a new strategy to correctly calculate edges to be sewn when a linear projection does not perform well will be developed

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 DPI2015-65123-R.

REFERENCES

Chen, H. H. and Huang, T. S. (1988). A survey of construction and manipulation of octrees. *Computer Vision, Graphics, and Image Processing*, 43(1):112–113.

Egol, K. A., Koval, K. J., and Zuckerman, J. D. (2010). *Handbook of Fractures*. Lippincott Williams & Wilkins.

Feito, F. R. and Torres, J. C. (1997). Inclusion test for general polyhedra. *Computers & Graphics*, 21(1):23–30.

Fornaro, J., Harders, M., Keel, M., Marincek, B., Trentz, O., Szekely, G., and Frauenfelder, T. (2008). Interactive visuo-haptic surgical planning tool for pelvic and acetabular fractures. *Studies in health technology and informatics*, 132(Figure 1):123–5.

Fornaro, J., Keel, M., Harders, M., Marincek, B., Székely, G., and Frauenfelder, T. (2010a). An interactive surgical planning tool for acetabular fractures: initial results. *Journal of orthopaedic surgery and research*, 5(50):1–8.

Fornaro, J., Székely, G., and Harders, M. (2010b). Semi-automatic segmentation of fractured pelvic bones for surgical planning. *Biomedical Simulation*, 5958:82–89.

Fürnstahl, P., Székely, G., Gerber, C., Hodler, J., Snedeker, J. G., and Harders, M. (2012). Computer assisted reconstruction of complex proximal humerus fractures for preoperative planning. *Medical image analysis*, 16(3):704–20.

Ho, C.-c., Fu-che, W., Bing-yu, C., and Ouhyoung, M. (2005). Cubical marching squares: Adaptive feature preserving surface extraction from volume data. *Computer Graphics Forum*, 24:2005.

Jiménez, J. J., Feito, F. R., and Segura, R. J. (2011). Tetra-trees properties in graphic interaction. *Graphical Models*, 73(5):182–201.

Jiménez, J. J., Feito, F. R., Segura, R. J., and Ogáyar, C. J. (2006). Particle Oriented Collision Detection using Simplicial Coverings and Tetra-Trees. *Computer Graphics Forum*, 25(1):53–68.

Jiménez, J. J., Paulano, F., Pulido, R., and Jiménez, J. (2016). Computer assisted preoperative planning of bone fracture reduction: simulation techniques and new trends. *Medical Image Analysis*, 30:30–45.

Jiménez, J. J. and Segura, R. J. (2008). Collision detection between complex polyhedra. *Computers & Graphics*, 32(4):402–411.

Lee, P.-Y., Lai, J.-Y., Hu, Y.-S., Huang, C.-Y., Tsai, Y.-C., and Ueng, W.-D. (2012). Virtual 3D planning of pelvic fracture reduction and implant placement. *Biomedical Engineering: Applications, Basis and Communications*, 24(03):245–262.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169.

Paulano, F., Jiménez, J. J., and Jiménez, J. (2015). Surface reconstruction of bone fragments: A comparative study. In Tavares, J. M. R. S. and Jorge, R. N., editors, *Computational Vision and Medical Image Processing V*, chapter 51, pages 321–326. CRC Press.

Paulano, F., Jiménez, J. J., and Pulido, R. (2014). 3D segmentation and labeling of fractured bone from CT images. *The Visual Computer*, 30(6-8):939–948.

Pulido, R., Paulano, F., and Jiménez, J. J. (2014). Reconstruction & Interaction with 3D Simplified Bone Models. In *WSCG 2014 Communication Papers Proceedings*, pages 321–327.

Samet, H. (2010). Sorting in space: multidimensional, spatial, and metric data structures for computer graphics applications. In *SA '10 ACM SIGGRAPH ASIA 2010 Courses*, pages 1–52, Seoul, Republic of Korea. ACM.

Tsai, M.-D., Hsieh, M.-S., and Jou, S.-B. (2001). Virtual reality orthopedic surgery simulator. *Computers in Biology and Medicine*, 31(5):333–351.

Willis, A., Anderson, D., Thomas, T., Brown, T., and Marsh, J. L. (2007). 3D reconstruction of highly fragmented bone fractures. In *Medical Imaging 2007: Image Processing. Proceedings of the SPIE*, page 65121.