COMPUTER-AIDED DESIGN & APPLICATIONS, 2018
VOL. 15, NO. 2, 203–210
https://doi.org/10.1080/16864360.2017.1375670

Taylor & Francis
Taylor & Francis Group

Extending VTK library to dynamically modify polygonal meshes in medical applications

Andrea Vitali [a], Daniele Regazzoni [a], Caterina Rizzi [a] and Giorgio Colombo [b]

[a]Università degli studi di Bergamo, Italy; [b]Politecnico di Milano, Italy

ABSTRACT

Rising reverse engineering applications require the use of 3D models based on polygonal meshes and a set of modeling operations to shape the final product. Usually, 3D modeling tools do not exploit existing software development kits for developing custom virtual applications. This approach can be very useful for reverse engineering process in the medical field. The paper describes the software development of local mesh modeling algorithms as an extension of the open source library VTK with the aim of shaping 3D triangulated meshes. Then, a case study has been considered to create virtual modeling tools supporting the design process of custom-fit products and, in particular, of lower limb prosthetic socket for amputees. The application has been tested to design a socket for a patient with an above knee amputation. Results reached so far provide a positive feedback on the quality of the designed process and outcome, mainly thanks to the use of the virtual tools based on the developed software modules for 3D mesh modeling.

KEYWORDS

Polygonal 3D modeling; medical devices; VTK; prosthesis design

1. Introduction

In the last years, emerging technologies, such as mixed reality and additive manufacturing, have fostered the improvement of the quality of interaction with 3D models based on polygonal meshes. Meshed models are generally preferred because data are more frequently gathered through reverse engineering (RE) techniques [6], [8]. The demand of reverse engineering is growing constantly in several fields (industrial and not) as well as the request of high quality polygonal mesh. For example in medical field, several software applications allow creating a 3D model of a human body district by a Computer Tomography (CT) [3], [14], [16], [26]. This means extrapolating by a number of images a tessellated mesh on which the physician can define a medical procedure or treatment. Another kind of RE application implies the use of a 3D scanned model of an anatomic district to design custom-fit product, i.e. limb prostheses [2],[4],[17],[18],[20]. Finally, there are applications exploiting 3D models of the entire human body, or a part of it, as a map to show data and easing the medical evaluation of patient's condition.

The most important features of a high-quality mesh are [7], [13], [24]:

- The initial point cloud has to be ordered as much as possible in the 3D space, i.e., the points have to be equidistant among them and the faces have a good allocation/distribution along the object surfaces.
- The polygonal mesh has to be manifold (Fig. 1(a)). The manifold conditions are: each edge is incident to one or two faces and the faces incident to a vertex form a closed or an open fan. Manifoldness is very important to allow the correct creation of objects by using additive manufacturing technology (Fig. 1).

Many algorithms have been developed to manage polygonal meshes for industrial design [13], [21], [24]. Available operations on polygonal meshes can be subdivided in two main categories:

- Full mesh modeling. Some algorithms are available to improve the quality of the whole 3D mesh. They permit to reposition, decimate and subdivide all faces of the 3D polygonal model in order to get a high quality polygonal model. These algorithms are usually applied before starting the modeling phase.
- Local mesh modeling. The algorithms of this category permit to locally change the polygonal mesh to design the requested final object (Fig. 2). These algorithms allow a real-time adding/removing of points and faces in specific zones of the model according to the user's needs. A manifold mesh is necessary for correctly executing local modeling [23].

CONTACT Andrea Vitali andrea.vitali1@unibg.it; Caterina Rizzi caterina.rizzi@unibg.it; Giorgio Colombo giorgio.colombo@polimi.it

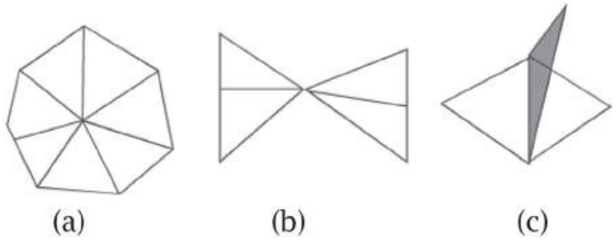

Figure 1. A manifold mesh (a) and two examples of non-manifold mesh (b) and (c).

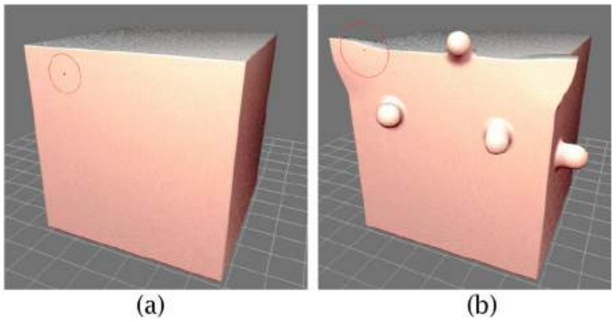

Figure 2. A polygonal cube (a) modeled in several zones by using both dragging and inflating virtual tools (b) of SculptGL application [22].

Several software applications have been developed to test the potentiality and the performance level of these algorithms applied on polygonal meshes [5], [22]. However, free and commercial software libraries for 3D sculpting present several limits that can be summarized as follows:

- Each 3D modeling application is based on its own 3D data structure and only a few commercial software libraries can be easily exploited to develop customized CAD applications for mesh modeling. Usually, available local modeling tools have different data structure to describe mesh topology. Therefore, the software algorithm for local mesh modeling has to be part of a wider software development kit to develop complete virtual applications (e.g., CAD, CAM and CAE tools).
- During a reverse engineering process, a virtual application has to include other functions to properly manage the mesh during the virtual product design process. Available solutions do not permit to automatize a sequence of simple 3D mesh modeling operations in order to emulate manual operations.

Thus, the main aim of this research work is to develop a software for local mesh modeling as an extension of Visualization ToolKit (VTK) library [25] to shape triangle meshes. VTK is an open-source software development kit (SDK) usually adopted for scientific applications [1], [19]. The implemented algorithms permit to create virtual modeling tools that emulate the tasks executed during the conventional manual manufacturing process (e.g. using hands or manual tools, such as a rasp). They have been applied to the design of a medical device, the socket of lower limb prosthesis that requires a high level of customization.

The first part of the paper introduces the basic 3D modeling tools for triangulated meshes. Then, the software development is described focusing on the algorithms related to the local mesh modeling based on VTK topology. A case study shows the use of the developed modules for creating virtual tools inside a knowledge-guided CAD application for socket design. Finally, the testing of the new tools is shown through the case study and results as well as future developments are presented and discussed.

2. Software development

Local mesh modeling operations are conceived to modify both the geometry and the topology of the triangulated manifold mesh through the execution of a software module. In this section, the algorithms are introduced for each basic modeling tool as well as the software architecture for local mesh modeling.

2.1. Basic 3D modeling tools

The developed modules encapsulate known algorithms described in [13], [24]. They allow to locally model manifold mesh in real-time by starting from both operations of decimation and subdivision. Two parameters have to be defined before modeling: the minimum and the maximum length of the edges of the 3D mesh. These values are used by modeling algorithms for evaluating if a selected surface can be either subdivided or decimated according to the goal of the modeling operation. In addition, the optimal choice of these parameters has an impact on the real-time performance of the modeling operations because recursive routines for subdivision and decimation execute their software instructions until one of the two parameters has been achieved. If either minimum length is too small or maximum length is too big the number of iterations quickly increases and the execution time of each operation does not permit a real-time interaction.

Algorithms of local decimation allow reducing the number of triangles in the selected zone of the 3D mesh. This algorithm removes the edges until their length reaches the defined maximum length. Thus, the topology of the polygonal mesh is updated (i.e., edges are removed and their points are deleted) and points and edges are repositioned in order to obtain new triangles with the same size without affecting the original shape.

The algorithm of subdivision permits to generate smaller triangles by starting from the selected surface. The software module can subdivide selected edges according to users' need until their lengths are greater than the specified threshold length. In case the length of a subdivided edge is less than the threshold length, the algorithm does not consider it anymore and then, the routine evaluates another edge for subdivision. This iterative procedure is done for all edges in the selected surface. Also in this case, new triangles are repositioned with the same size along the surface. The repositioning of triangles is based on the mesh Laplacian smoothing [15] calculating the curvature along the selected surface through the Laplace operator.

Subdivision and decimation are always applied before each modeling operation occurs to emulate the final behavior of the manual operation, such as inflating/deflating the mesh, or flattening a selected part. All developed algorithms ensure high quality mesh and the modeling operations keep it manifold.

The operations allowed on the mesh are subdivided as follows:

- **Inflate/Deflate**: These algorithms translate the selected points along their normal. The more the point is near to the selected point on the mesh, the more the translation will be great (Fig. 3).
- **Dragging:** adds new mesh portions by moving the cursor in any desired direction. The algorithm executes a translation as for inflate and deflate operations, but the direction of the translation is given by mouse pointer motion (Fig. 4.).

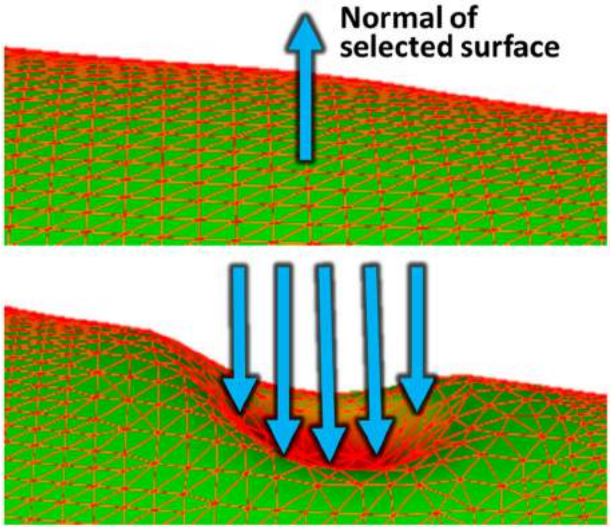

Figure 3. Example of deflate operation.

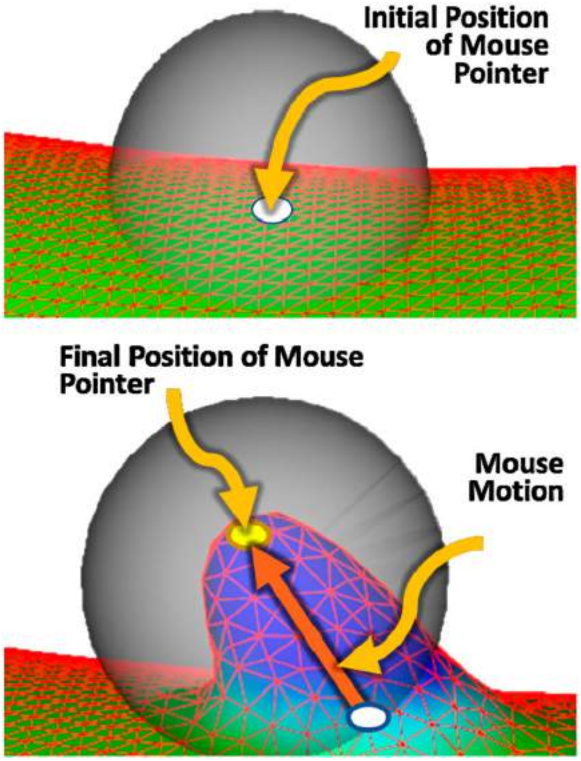

Figure 4. Basic behavior of dragging operations with generation of new triangles according to the direction of the mouse motion.

- **Smoothing**: Given a selected surface, this algorithm uses the Laplacian filter to evaluate the average curvature of the given surface. The algorithm of smoothing calculates a translation for each point of triangles having a different curvature from the average value of the selected surface. In this way, smoothing allows removing useless details along the 3D mesh (Fig. 5).

Even if these algorithms are already known in scientific literature, there is not a simple way to interface the memory structure for topology management inside existent open-source software library. Thus, an extension of VTK has been developed to make available these algorithms. The developed software architecture allows extending modules with others 3D local modeling operations according to the final purpose of the application as well as the basic modeling operations can be used to emulate more complex operations according to predefined procedures.

2.2. Software architecture

The whole software platform has been developed in C++ following the object-oriented paradigm shown in Fig. 6. As abovementioned, the developed modules exploit and

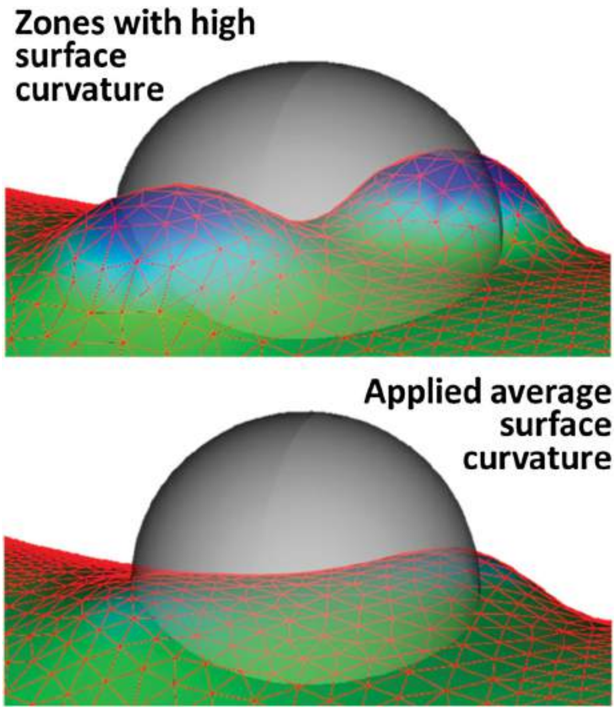
Zones with high surface curvature
Applied average surface curvature

Figure 5. An example of smoothing operation along a surface.

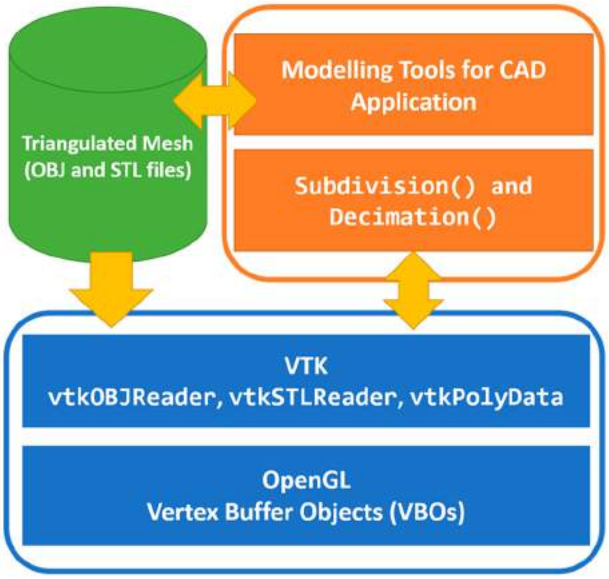
Triangulated Mesh (OBJ and STL files)
Modelling Tools for CAD Application
Subdivision() and Decimation()
VTK
vtkOBJReader, vtkSTLReader, vtkPolyData
OpenGL
Vertex Buffer Objects (VBOs)

Figure 6. Software architecture: developed modules of the new platform (orange); basic library used (blue).

extend VTK. At present, there are important applications developed starting from VTK in the medical field and for industrial 3D simulations. VTK makes available file reader in order to load the mesh topology inside a memory structure, which can be used to get needed information (e.g., points, faces and connected edges) through a single programming class named vtkPolyData. VTK uses OpenGL library for 3D rendering.

The system links a polygonal mesh to a set of virtual tools that aim at designing the product 3D model by local mesh modeling. Furthermore, an event listener has been developed to exploit the execution pipeline of VTK, which automatically detects interaction events, such as event of mouse and keyboard. Thus, the software developer can associate an interaction event for each modeling operation the final application requires.

An acquired 3D triangle model can be loaded into a vtkPolyData object using either vtkObjReader or vtkStlReader, which can respectively read OBJ and STL files in application based on VTK. As mentioned before, the class vtkPolyData contains the topology information of the associated 3D mesh previously loaded. This class makes available a set of methods allowing to dynamically manage the lists of points and triangles and the relations between them.

The modules use these methods to add, remove or modify both points and faces and their relationships. The method BuildLinks() has to be called before starting the 3D modeling. This method creates the topology of the actual mesh and gives the possibility to dynamically update the topology through a set of methods for adding during both subdivision and decimation algorithms. In particular, there are a set of methods for replacing and deleting both point and cells.

After the execution of the algorithm relative to either subdivision or decimation, these methods update the topology of the triangle mesh. Once the algorithm has been executed and the topology modified, the method Modify() automatically updates the OpenGL Vertex Buffer Objects for rendering the mesh according to the new topology.

Finally, a memory structure has been developed in order to deliver undoing and redoing operations after mesh modeling. These operations can be easily adopted inside a custom application through the methods Undo() and Redo(), which are developed according to available modeling operations. The memory structure of these features can be used also to map other operations that are not 3D modeling procedures and thus, several extensions of Undo() e Redo() methods can create the whole undo/redo management system of the developed application.

3. Case study

The system has been adopted to develop a knowledge-guided CAD system, named Socket Modeling Assistant (SMA). SMA is part of a virtual design environment for lower limb prosthesis and permits to create the 3D model of the socket according to patient's characteristics (e.g., lifestyle) and residual limb morphology [9], [10].

The 3D socket model is created by applying virtual modeling tools on a polygonal mesh acquired either using Magnetic Resonance Imaging (MRI) technique or laser scanning. SMA is also interfaced with a Finite Elements Analysis (FEA) system to analyze the pressure distribution at the socket-residual limb interface and, once reached the final shape, exports the polygonal mesh for 3D printing with data-driven multi-material option [11].

Traditionally the prosthetist modifies locally the shape of a positive chalk; therefore, we had to develop virtual modeling tools that emulate the traditional design tasks interacting with dynamic meshes (the geometrical models of the residual limb and the socket) and locally modify the mesh either in automatic or interactive mode.

The developed platform has been exploited by setting the minimum length at 2 millimeters and the maximum length to 10 millimeters of each edge of the acquired 3D model of the residual limb. These two values have been chosen according to the quality of the models initially acquired either by 3D scanner or by 3D automatic reconstruction from MRI volume. During this step, the acquired point cloud is treated to obtain a manifold mesh. The 3D scanner acquisition is done with the low-cost application Skanect [23], which automatically cleans and filters the obtained point cloud and allows exporting a manifold mesh in STL file format. The 3D reconstruction from MRI volume uses an ordered point cloud fitted to a NURBS surface, which is used for reshaping a triangulated manifold mesh by VTK modules [12].

Furthermore, the undoing and redoing operations of SMA are totally based on the software modules developed for 3D mesh modeling. Inside SMA, two virtual tools have been developed, which exploit developed platform for local mesh modeling such as the sculpting tool and the lower zone definition tool.

Using the sculpting tool, the user can proceed with an interactive shape manipulation on both the socket and the amputated lower limb. The abovementioned basic modeling operations have been exploited inside SMA to emulate the usual operations done by the orthopedic technicians by using tools and hands to shape the positive chalk model of the socket.

The modeling operations are used on the mesh inside the sculpting tool of SMA as shown in the following:

- **Inflate/Deflate → adding/removing materials**: the system permits to emulate the operations of adding or removing material along the selected surface by applying respectively inflate or deflate operation. If the user selects the inflate operation the algorithm adds virtual material towards the outside of the virtual model. Deflate operation executes the same operations toward the interior of the virtual object (Fig. 7.).

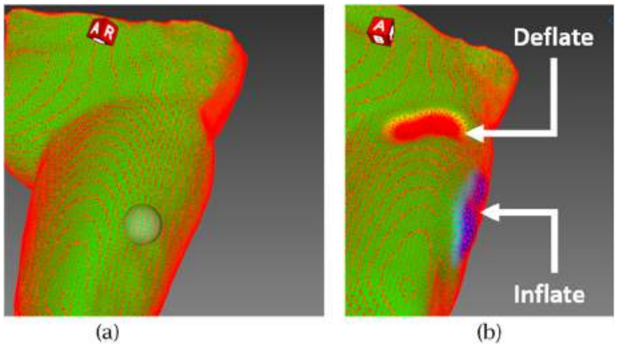

Figure 7. The initial 3D model of the residual lower limb (a) shaped using (b) deflating and inflating operation in two different zones.

- **Flattening**: this operation allows flattening the selected surface (Fig. 8.a.). Flattening operation can be applied to the initial model of virtual residual limb in order to remove a particular deformation (.e.g., to create the lower part of the socket) and obtain a flat surface (Fig. 8.b.).
- **Dragging**: inside SMA, dragging operation is used to add a new part of the mesh along the ischiatic area in order to prepare the internal side of the final socket (Fig. 9).
- **Smoothing**: this operation smooths and brushes the 3D mesh to remove details relative to the initial 3D model of the amputated lower limb, such as scars (Fig. 10).

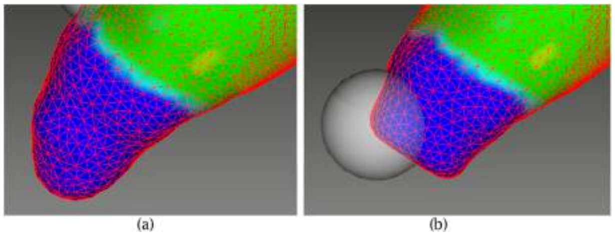

Figure 8. Flattening operation exploited inside SMA.

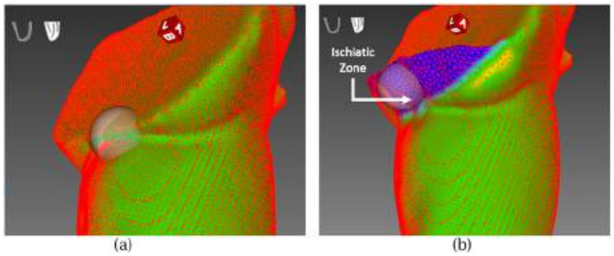

Figure 9. Dragging operation (b) applied on a 3D scanned model of a residual lower limb (a).

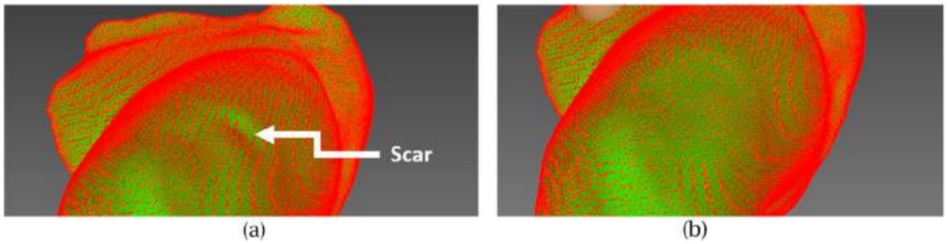

Figure 10. Using the virtual smoothing tool on the virtual residual limb (a) to remove a scar (b).

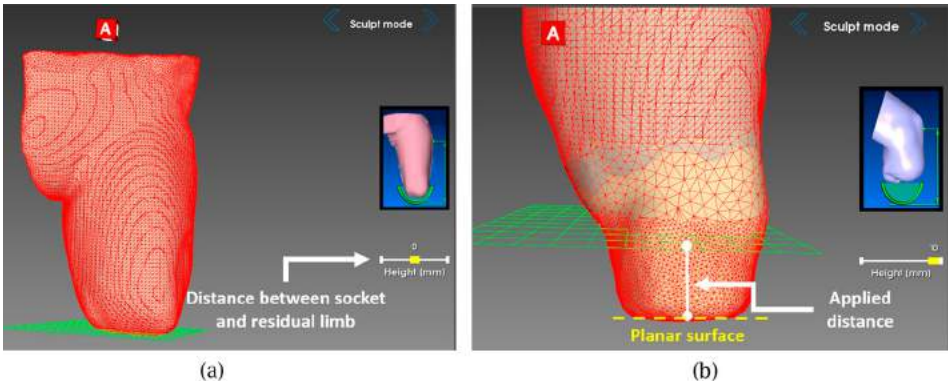

Figure 11. Definition of the lowest part by using local modeling operations in automatic way.

Developed platform has also been exploited to automatically design the lower part of the socket. The *lower zone definition* tool automatically combines dragging and flattening. It creates the final shape of the lower zone starting from the 3D model of the residual limb. The user has only to define the final distance between the lowest point of the residual lower limb and the lowest point of designed socket using the proper slider (Fig. 11.a.). According to this distance, SMA automatically applies the dragging operation until the mesh has reached the required shape. Finally, flattening is applied in order to create a planar surface and simplify the adding of the link with the 3D socket model (Fig. 11.b.).

SMA also exploits undoing/redoing systems developed for local mesh modeling. Each virtual tool has customized `Undo()` and `Redo()` methods permitting to undo/redo operations done during the use.

4. Test

The new version of SMA has been tested with the aim of executing the whole design procedure to create a socket for a real lower limb prosthesis. During this procedure, the sculpting tool and the lower zone definition tool have been used to obtain the final socket shape.

In order to execute the test procedure, a male patient has been involved. He is 53 years old and has an above knee amputation. His residual limb has been 3D scanned using a Microsoft Kinect device v1 and the low-cost application Skanect [23]. The 3D scanned model has been loaded inside SMA to start the design procedure of the socket.

Virtual tools of SMA including those for local mesh modeling have been used by engineering students to create the final model of the socket. Then, the designed socket has been realized by using additive manufacturing technology, in particular a material extrusion equipment working with PLA material. Finally, the 3D printed socket has been worn by the patient to evaluate if the socket was properly fitting. The patient was able to wear the socket, but there were zones too tight for permitting to load the entire body weight and walk.

After the conclusion of the preliminary test phase, the use of the virtual tools has been considered very easy and intuitive. In addition, the modeling operations allow executing detailed tasks on the mesh, which cannot be performed by the automatic procedures provided by the other virtual tools of SMA. By the way, the sculpting tool presents some limits relative to the selection of the local zone the user needs to modify. The sphere selection does not permit to execute modeling operation to obtain sharp edges and chamfers that may be useful, for instance, to link the socket with the rest of the prosthesis.

Future tests have been planned with an orthopedic laboratory in Italy to assess the new application by qualified technical personnel. The aim will be the use of SMA with local mesh modeling to design several sockets for lower limb prosthesis and thus, evaluating the effectiveness of the proposed method.

5. Conclusion

The paper describes a software platform for local mesh modeling based on VTK. This research work aims at going beyond the limits of existing software solutions for mesh modeling, which are not easily re-usable to create CAD application in the medical fields.

The developed platform makes available several software modules for creating virtual modeling tools based on both subdivision and decimation of a triangulated mesh. The structures of the topology of the mesh are totally based on those defined in VTK and thus, this approach allows the developers to exploit all other functionalities that are available in VTK.

The platform has been tested with SMA, a knowledge guided CAD application to design the socket of lower limb prosthesis. Two virtual tools have been developed that exploit new local mesh modeling tools. The *sculpting tool* uses the local mesh modeling on the 3D mesh of the acquired residual lower limb to execute several operations for customizing the final socket shape. The lower zone definition tool exploits developed modules in automatic way. Indeed, the user does not directly use the mesh modeling tools, but he/she exploits them through the slider that manage the distance between lower part of the residual limb and the lowest part of the designed socket. This approach totally hides the local mesh modeling to the user and the automatic procedure will be used also in other virtual tools of SMA in order to maintain a very high quality of the mesh.

Furthermore, the automatic procedure of local mesh modeling will be proposed also in other medical contexts, such as the design of maxillo-facial and dental prosthesis.

Finally, future developments have been planned in order to test the developed modules in a mixed reality application to recreate the orthopedic laboratory where the orthopedic usually operates.

ORCID

Andrea Vitali http://orcid.org/0000-0001-9261-4357
Daniele Regazzoni http://orcid.org/0000-0001-5533-7047
Caterina Rizzi http://orcid.org/0000-0002-1779-5183
Giorgio Colombo http://orcid.org/0000-0002-9999-8960

References

[1] 3DSlicer, https://www.slicer.org/.

[2] Abouel Nasr, E.; Al-Ahmari, A. M.; Moiduddin, K.; Al Kindi, M.; Kamrani, A. K.: A digital design methodology for surgical planning and fabrication of customized mandible implants. Rapid Prototyping Journal, 23(1), 2017. https://doi.org/10.1108/RPJ-11–2014–0157

[3] Ameddah, H.; Assas, M.: Bio-CAD reverse engineering of free-form surfaces by planar contours. Computer-Aided Design and Applications, 8(1), 2011, 37–42. http://doi.org/10.1080/16864360.2010.10738809

[4] Bartesaghi, S.; Colombo, G.; Morone, S.: Spatial Augmented Reality and Simulations to Improve Abdominal Aortic Aneurysm Diagnosis and Monitoring. Computer-Aided Design and Applications, 12(6), 2015, 803–810. http://doi.org/10.1080/16864360.2015.1033347

[5] Blender, https://www.blender.org/.

[6] Carulli, M.; Bordegoni, M.; Cugini, U.: Integrating Scents Simulation in Virtual Reality Multisensory Environment for Industrial Products Evaluation. Computer-Aided Design & Applications, 13(3), 2016, 320–328.

[7] Calabrese, C.; Salvati, G.; Tarini, M; Pellacini, F.: cSculpt: a system for collaborative sculpting. ACM Transactions on Graphics (TOG), 35(4), 2016. https://doi.org/10.1145/2897824.2925956

[8] Colombo, G.; Comotti, C.; Redaelli, D. F.; Regazzoni, D.; Rizzi, C.; Vitali, A.: A Method to Improve Prosthesis Leg Design Based on Pressure Analysis at the Socket-Residual Limb Interface. ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2016. https://doi.org/10.1115/detc2016–60131

[9] Colombo, G.; Facoetti, G.; Rizzi, C: A digital patient for computer-aided prosthesis design. Interface focus, 3(2), 2013. https://doi.org/10.1098/rsfs.2012.0082

[10] Colombo, G.; Facoetti, G.; Rizzi, C.; Vitali, A: Mixed reality to design lower limb prosthesis. Computer-Aided Design and Applications, 13(6), 2016, 799–807. https://doi.org/10.1080/16864360.2016.1168223

[11] Comotti, C.; Regazzoni, D.; Rizzi, C.; Vitali, A.: Additive Manufacturing to Advance Functional Design: an Application in the Medical Field, Journal of Computing and Information Science in Engineering, 2016. https://doi.org/10.1115/1.4033994

[12] Colombo, G.; Rizzi, C.; Facoetti, G.; Vitali, A.: SimplyNURBS: A Software Library to Model NURBS for Medical Applications, Computer-Aided Design and Applications, 12(6), 2015, 794–802. http://doi.org/10.1080/16864360.2015.1033346

[13] Dunyach, M.; Vanderhaeghe, D.; Barthe, L.; Botsch, M.: Adaptive Remeshing for Real-Time Mesh Deformation. In Eurographics, 2013, 29–32.

[14] Ebert, L. C.; Schweitzer, W.; Gascho, D.; Ruder, T. D.; Flach, P. M.; Thali, M. J.; Ampanozi, G.: Forensic 3D visualization of CT data using cinematic volume rendering: a preliminary study. American Journal of Roentgenology, 208(2), 2017, 233–240. https://doi.org/10.2214/AJR.16.16499

[15] Herrmann, L. R.: Laplacian-isoparametric grid generation scheme. Journal of the Engineering Mechanics Division, 102(5), 1976, 749–907.

[16] Liu, H.; Zhang, C.; Zhang, C.: Visualization and surface rendering based on medical image. Computer-Aided Design and Applications, 9(1), 2012, 79–86.

[17] Nayak, C.; Singh, A.; Chaudhary, H.; Tripathi, A: A novel approach for customized prosthetic socket design. Biomedical Engineering: Applications, Basis and Communications, 28(03), 2016, 1650022. https://doi.org/10.4015/s1016237216500228

[18] Munsch, M.: Laser additive manufacturing of customized prosthetics and implants for biomedical applications. Laser Additive Manufacturing: Materials, Design, Technologies, and Applications, 2016, 399.

[19] Paraview, http://www.paraview.org/.

[20] Pathak, V. K.; Nayak, C.; Singh, A. K.; Chaudhary, H.: A Virtual Reverse Engineering Methodology For Accuracy Control Of Transtibial Prosthetic Socket. Biomedical Engineering: Applications, Basis and Communications, 28(05), 2016, 1650037. https://doi.org/10.4015/s101623721650037x

[21] Rajic, A.; Desnica, E.; Stojadinovic, S.; Lazic-Vulicevic, L.; Eric, S.: Reverse engineering and CAD inspection of knee implant using the Noomeo Optinum 3D Scanner. Annals of the Faculty of Engineering Hunedoara, 13(2), 2016, 39.

[22] SculptGL, https://stephaneginier.com/sculptgl/.

[23] Skanect, http://skanect.occipital.com/

[24] Stanculescu, L.; Chaine, R.; Cani, M. P.: Freestyle: Sculpting meshes with self-adaptive topology. Computers & Graphics, 35(3), 2011, 614–622. https://doi.org/10.1016/j.cag.2011.03.033

[25] Visualization ToolKit, http://www.vtk.org/.

[26] Zhu, L.; Li, Y.; Yu, Y.; Zhang, B.; Wang, L.: 3D reconstruction for soft tissue of the human body. In Mechatronics and Automation (ICMA), 2016 IEEE International Conference, 2016, 1721–1725.