










A CAD Procedure for Planning and Modelling Custom Surgical Guides for Tumor Resection

Michaela Servi¹ , Francesco Buonamici² , Lorenzo Guariento³ , Angela Busato⁴ , Yary Volpe⁵ , Maurizio Scorianz⁶  and Domenico Andrea Campanacci⁷ 

¹University of Florence, Italy, michaela.servi@unifi.it

²University of Florence, Italy, francesco.buonamici@unifi.it

³AOU Careggi, lorenzo.guariento@unifi.it

⁴AOU Careggi, busato.angela@gmail.com

⁵University of Florence, Italy, yary.volpe@unifi.it

⁶AOU Careggi, maurizio@scorianz

⁷AOU Careggi, domenicoandrea.campanacci@unifi.it

Corresponding author: Yary Volpe, yary.volpe@unifi.it

Abstract. The use of customized surgical guides for the removal of bone tumors is an increasingly common practice in advanced hospitals. These devices are modeled directly on the 3D model of the patient's anatomy and produced using additive techniques; therefore, they have the ability to perfectly anchor to the bone anatomy and allow the tissue of the tumor mass to be removed according to cutting paths predetermined by the physician during preoperative planning. This work proposes a semi-automatic framework for modeling custom surgical guides composed of two main modules: 1) a web application, called Precise, that allows the surgeon to visualize and manipulate 3D anatomical models and enter a set of cutting planes that determine the correct tumor resection paths and other useful information for modeling the surgical guide; 2) a fully automatic procedure that based on this information creates the guide's STL model. The entire procedure aims to create a specific type of cutting guide designed to guide an oscillating saw in a series of planar cuts, thus isolating the tumor. The cutting guides produced are composed of three main elements: a base, the blade slot, and pin's holes.

Keywords: Surgical Planning, Rapid Prototyping, Medical Application, Medical 3D Model

DOI: <https://doi.org/10.14733/cadaps.2024.487-498>

1 INTRODUCTION

In recent years, clinical practice is facing a profound transformation characterized by the patient-centered definition of surgical and nonsurgical treatments. Modern technologies such as Machine Learning (ML), Reverse Engineering (RE) and Additive Manufacturing (AM) techniques are central to this phase of evolution that increasingly involves treatment tailored to the patient's unique characteristics [9,7]. Prominent in this scenario is surgical planning for the removal of bone tumors,

for which it is becoming more and more common to create customized Surgical Cutting Guides (SCG) that perfectly match the patient's anatomy [11,4,3,8]. In fact, thanks to the evolution of 3D modeling and 3D printing tools, it is possible to create SCG in a short time frame, appropriate to the clinical procedure, with biocompatible materials and proven robustness. The procedure for making these devices requires close collaboration between surgeons and engineers/technicians in charge of controlling the engineering tools required for cutting guides production process. Across the entire process, the definition of the geometry of the SCG is known to be time-consuming; usually, it requires several iterations before reaching a clinically satisfactory result [5]. Different paths can be pursued to reach the final result, and the most common involves the CAD modelers following some guidelines initially provided by the surgeon to propose different solutions that can be accepted or rejected by the physician [6]. An alternative and faster resolution strategy, on the other hand, involves the physician indicating the cutting path directly on the 3D model of the anatomy with tumor, through, for example, the insertion of planes in a CAD modeling software (e.g. [10]) and, from these inputs, carrying out the SCG design. In this context, it becomes crucial to focus on two aspects: 1) for the physician to interface with simple and intuitive 3D visualization and manipulation software applications; and 2) the possibility of systematizing and automating modeling procedures to streamline the process and deliver the device as soon as possible. To reach such goals, this work proposes an architecture based on two modules: (i) an automated procedure for modeling SCG for bone tumors from simple inputs, which was developed within nTopology - a CAD software based upon implicit modeling [12] - and (ii) a new application - called Precise - that allows the clinician to autonomously define such design inputs by interacting with anatomical 3D models in a simplified environment. Figure 1 shows the workflow considered in the present work to produce the final SCG model.

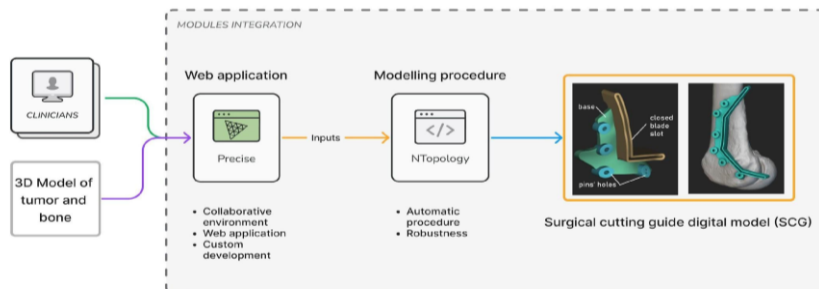


Figure 1: Implemented modules for custom tumor resection planning.

The entire procedure aims at the creation of a specific type of SCG designed to guide an oscillating saw in a series of planar cuts thus isolating the tumor. The SCG is defined by three main elements (depicted in Figure 1): a) *Blade slot*: it is formed by an extrusion with a slot to guide the blade along the cutting path defined by the surgeon with a series of incident planes. b) *Base* – i.e. the part of SCG in direct contact with the bone; its shape determines the stability of the device and gives a reliable reference for correct positioning by fitting the bone anatomy. c) *Pins' holes*, to fix the device to the bone. The pin diameter depends on the specific application. Axes of the pins should not be parallel to each other to ensure stability.

In order to reach this result, the modelling procedure requires the following inputs: *input#a*) Polygonal mesh of the target bone *input#b*) the cutting path, defined by a series of planes surrounding the tumor (i.e. SCG planar cuts), in two representations: i) centers and normals; ii) STL format; *input#c*) a series of points on the bone surface defining the profile of the *Base* to define the exact contour; *input#d*) a plane to guide the *Base's* extrusion – i.e. the direction of extrusion of the base; *input#e*) pins' application points and vectors defining their orientations.

The *Precise* app is responsible for the generation of such information in the correct format (CSV files and STLs files) in order to streamline the entire workflow. To this purpose, the environment for digital surgical planning performed by the surgeon required a development from the ground up; following a thorough examination of the software available in the literature [1], it was decided to develop a dedicated application encapsulating the required features in a single product - i.e. simplify the interaction and manipulation of 3D models, providing tools accessible to physicians. *Precise* implements a collaborative framework between physicians and engineers and was developed as a web application (thus reachable anywhere and without the need for local installation). The two modules will be described in the following sections.

2 WEB APPLICATION

The implemented application described below promotes a collaborative framework for designing SCG, enabling the surgeons to perform the preoperative planning of a tumor resection.

The fundamental requirements of the application, considering the tasks to be carried out, were initially defined as follows:

- import models defined by tessellated geometries (e.g. STLs) with single or multiple bodies;
- possibility to add and manipulate planes;
- use the planes to visualize the tumor resection result (cut the 3D models);
- add 3D models of pins;
- perform measurements on the 3D model;
- draw on the model surface;
- offset an element (with specific reference to the tumor area, so that a safety margin could be considered in the design);
- add annotations;
- insert 3D points on the bone surface.

Figure 2 shows the architecture layers of the application, divided into two parts: frontend (the presentation layer) and backend (the data access layer), and the involved libraries.

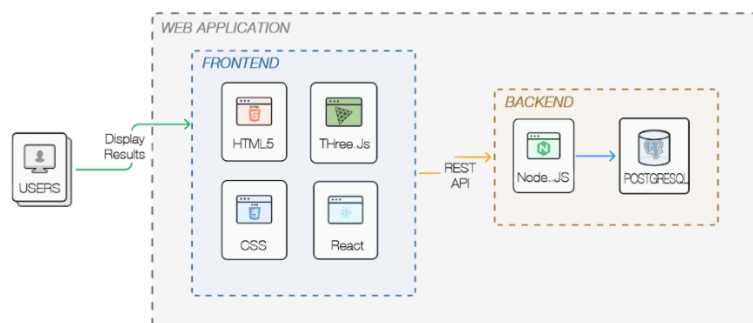


Figure 2: Architecture of Precise web application.

The involved technologies are, for the frontend:

- HTML and CSS for page definition and styles;
 - JavaScript for the business logic part, page rendering, page processing, interaction with the backend via API calls;
 - React as a library for web component development;
 - Three.js for 3D data manipulation.
- For the backend:
- Node.JS for interacting with the database and generating data to be made available to the frontend;

- PostgreSQL for creating the database, the entities needed by the application, such as the list of users who can access it and the projects that are created.

Once the application infrastructure was defined, some of the features listed above were implemented by directly exploiting features provided by the Three.js library, such as importing STL objects, intuitively moving objects in space, and inserting known objects (such as pins or points) into the scene. Open source APIs were integrated so as to add features such as drawing on the model surface (useful in instances when the physician needs to mark anatomic areas that should not be touched by the SCG) [13] or taking measurements in space [14]. On the other hand, for some features, it was necessary to define dedicated procedures, described below:

Global alignment. The alignment of all imported objects with the center of the global reference system is an important first control, essential for having a product that is easy to use, and rarely found in cutting-edge software. To accomplish this, the program saves the first imported object's center of gravity and displacement with respect to the global reference system in a centralized state. This displacement is then applied to all elements (including those imported at the same time) managed in the project, both in the current session and in future sessions, resulting in elements that are aligned and thus easily manageable.

Clipping. This is a critical feature because it is used to define the tumor resection paths that will be used to model the surgical guide. The clipping feature refers to either clipping the object with a single plane or clipping the object with multiple planes, and clipping must be negated in both cases. The corresponding functionality provided by the Three.js library was used to implement clipping with a single plane and its negation. On the other hand, to implement clipping with multiple planes, a dedicated routine was implemented. The issue of this procedure lies in the directions of the normals of the inserted planes, which may sometimes disagree and thus not realize a consistent clipping. To align the normals the implemented procedure performs two steps: 1) calculation of the mean barycenter of all the planes (denoted by c in Figure 3a); 2) check for each plane whether the normal faces the mean barycenter, and if not, reverse the normal, thus aligning all the normals. The user can choose which of the two set of trimmed bodies to show/hide (Figure 3b).

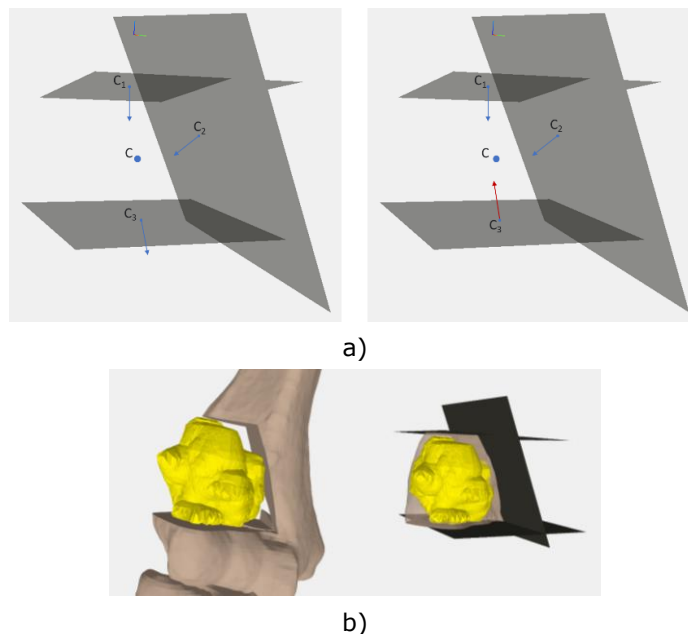


Figure 3: a) Normal alignment procedure for clipping with multiple clipping planes; b) example of multiplane cut before and after reversing the cutting direction.

Offset. A dedicated offset procedure was created in this work since it is currently missing from the Three.js library. The procedure creates the mesh of a selected STL model by offsetting faces and vertices by a user-specified value. To define the position of vertices and faces in the offsetted model, the original model's vertices are saved in a hash table (this allows optimizing the performance of the algorithm). To define the direction of displacement of the vertices, the normals of the faces insisting on a vertex are averaged, then the vertex is moved along the average direction by the amount entered by the user, and finally the faces and their normals are recalculated based on the displaced vertices. Given the program's intended use, it was deemed unnecessary to correct any intersecting faces.

The Precise application enables the visualization and manipulation of 3D data through a web browser. The editor page, accessed after logging in and opening a project, contains the primary features of the application, including all the tools specified in the requirement definition phase. Specifically, the editor page (Figure 4) is divided into four sections: 1. the header which contains a list of for file manipulation (such as import, export, and save), a button for sharing the project, and the user profile settings; 2. the left sidebar, which contains tools for managing the objects in the scene; 3. the center of the page which is devoted to 3D navigation, and 4. the right sidebar which is dedicated for annotation features.

As mentioned in the introductory section the inputs necessary for the modelling procedure are: input#a) Polygonal mesh of the target bone and tumor input#b) a series of planes defining the cutting path; input#c) a series of points on the bone to define the exact contour of the Base; input#d) a plane to guide the Base's extrusion; input#e) pins' application points and vectors defining their orientations. To define the inputs, the user must initially import the STL model of reference anatomy and tumor (*input#a*). Safety margins for resection can be taken into account by offsetting the tumor of an arbitrary value (yellow model in Figure 4), then a series of planes can be placed (*input#b*) and used to verify the cut with the clipping function either individually or all together (Figure 4 shows an example of clipping with three planes). Every plane is modeled as a limited planar surface using an STL file composed of 2 triangles defining a square. Once the clipping planes have been defined, the user proceeds by entering the points (*input#c*) that indicate the path of the SGC base profile positioning them avoiding specific anatomical areas based on clinical considerations (points in red in Figure 4). In order to model the guide and define the direction of extrusion of the Base, after orienting the 3D model in the direction accessed during surgery, by clicking the dedicated button the current-view is blocked such to constrain the *input#c* points to lie on a plane having the normal coincident with the current-view plane, this plane is saved as *input#d*. Finally, the clinician can insert the pins in the desired position (*input#e*) (grey elements in Figure 4); the pins are inserted as cylinders using a dedicated function and can be moved in the 3D environment. All these elements can easily be exported all together (CSV format for points and vectors, and STL format for 3D elements) by clicking on the dedicated export button.

3 AUTOMATIC MODELING PROCEDURE

nTopology offers a procedural environment where operations are represented by a series of function blocks that can be combined to create personalized workflows and automate repetitive operations reliably [6]. The SCG creation workflow consists of a series of Boolean operations and extrusions, able to deliver the result regardless of a large variation of the inputs [5]. To launch the SCG modelling procedure, the user is required to insert the desired values for the following variables: *pins' diameters*, *blade thickness*, *blade slot height*, *blade's clearance*, *base's height*, which may vary in each case.

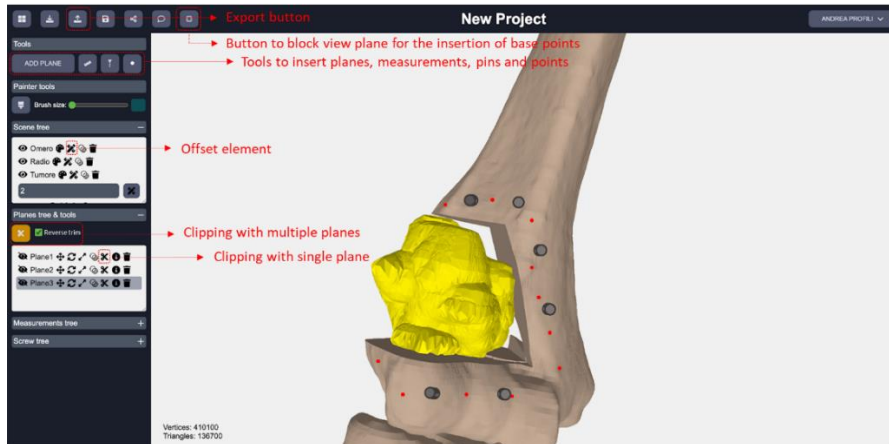


Figure 4: Usage example of Precise application.

In first instance the procedure models *Base*, *Pins' holes*, and *Blade slot*, separately:

- **Base** – the contour points – *input#c* – are used to extrude a polygonal line connecting the points. The profile is extruded perpendicular to the build plane – *input#d* to a standard oversized value (extrude profile block – Figure 4a). This rough shape will be subsequently brought to the final thickness with Boolean operations. The obtained solid is then smoothed (Smoothen body block).
- **Pins' holes** – pins application points and orientations – *input#e* – are used to create two sets of cylinders (cylinder block): internal and external; the diameter of the internal set is defined by the variable pins' diameter; the external set is obtained offsetting the first one with a standard value of 3mm.
- **Blade Slot** – two bodies are created (internal and external blade slots). The internal slot is modelled by thickening the STLs planar surfaces – *input#b* – (Thicken body block) with a value equal to blade's thickness plus blade clearance. The external blade slot is created by applying an offset to the internal blade slot; the thickness depends on the SCG material (e.g. for polymers, 5 -10mm).

The bodies are refined to their final thickness and shape by intersecting them with the bone surface itself (to create the contact surface with the anatomy) and with offsets of the target bone (*input#a*) – whose offset values varies for each element (e.g. Figure 5b for the base). The offset values used by the procedure are: *base's height* for the *base*; *blade slot height* for the *external blade slot*; a standard value of 10mm for the *external pins' holes*. The thickness of the base must ensure good resistance and stiffness while minimizing mass and material. For polymeric SCG, 5mm is usually a good tradeoff value. The *base*, *external pins' holes*, *external blade slot* are then merged together (*Boolean union* block – Figure 5c). Subsequently, the internal *pins' holes* and *internal blade slot* are subtracted from the resulting body (*Boolean subtraction* block – Figure 5d) to create the final SCG model (Figure 5e).

4 CASE STUDIES

In order to better understand the functioning of the procedure and the usefulness of the Precise web application for concurrent design, the proposed framework was applied to two case studies for the design of custom SCG. Both cases present patients with bone tumors which need a surgical resection intervention. Anatomical data were anonymized before being used for this study, which did not involve actual use of the proposed SGC devices on the patient.

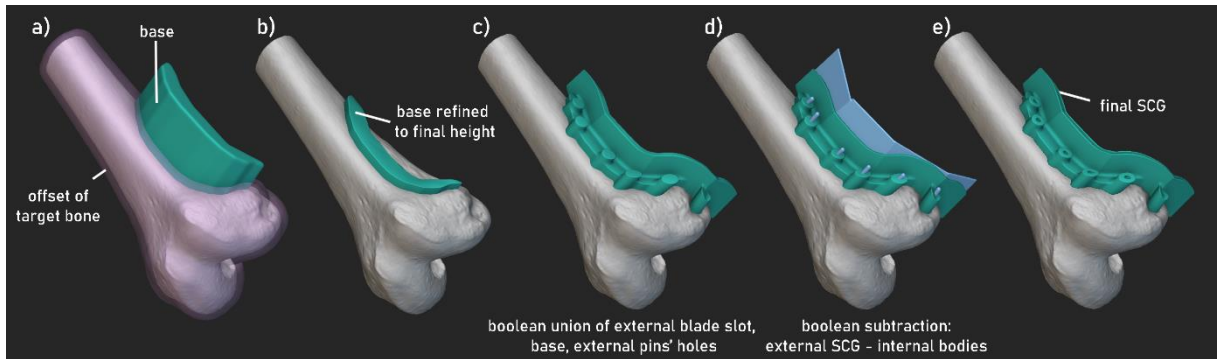


Figure 5: SCG modelling steps inside nTopology.

In the first case the anatomical region of interest is the pelvis, in the second one is the tibia. The steps that were followed in both cases are the ones described in the previous sections. In the following paragraphs, a step-by-step description of the procedure followed to obtain the final SCG model is presented. In both cases, the STLs files required to start the design activity were obtained through the segmentation of the tissues of interest (bone, tumor) from a CT image acquisition. For the pelvis case, separate STL models were created for each bone composing the region of interest. The generated models were then imported into the precise application, where suitable and recognizable materials/colors were assigned to them.

4.1 Detailed Description of the Test Cases

4.1.1 Precise application

The pelvis case studied is characterized by a tumoral mass positioned near the left iliac crest, at the intersection with the sacroiliac joint. The tumoral mass is highlighted in red in Figure 6a and Figure 6b (frontal and dorsal/lateral views). The Tibia case presents a significant tumoral mass near the knee articulation (orange 3D model in Figures 7a, 7b and 7c). Using the Precise app and considering the shape of both the bone and the tumor, the clinician has decided on the surgical access to perform the resection operation: this operation is eased by the possibility of interacting with the 3D models of the anatomy. In the Pelvis case the surgical access is posterior, with the point of view proposed in Figure 6c. For the Tibia, the surgical access is lateral, with the direction depicted in Figure 7a. Once the access way is defined, the clinician can start to define the region of the bone portion that will be removed, considering the need for pull-out constraint in placing the cutting planes delimiting the salvaged portion from the rest of the bone. The access way influences also the direction of the cutting planes, as the cutting operation will need to be performed accessing the area with the surgical saw from the identified direction. To determine the exact contour of the portion of the bone to be removed, the first operation that is performed is the generation of an offset volume of the tumor (Offset element button in Figure 4), to define a tolerance region around the tumoral area; offsets of the models are depicted in yellow in Figure 6c and Figure 7d. Planes are then inserted into the environment clicking on the "ADD PLANE" button visible in Figure 4. The final placement of each plane will determine the contour used to cut the portion of the bone to be removed; the position of the planar surfaces are defined by the user who is able to interact with a handle that offers commands to rotate, translate and scale the models in all directions (Figure 6c and Figure 7d). Figures 6d and Figure 6e show the configurations approved as final results for the cutting surfaces. Subsequently, the user needs to indicate a series of points (*input#*) that indicate the contour of the SCG. This is done by orienting the 3D models in the access direction, locking the view using the

dedicated button (Figure 4) and projecting a series of points on the surface of the healthy bone through a point-and-click interaction of the user. The screen plane direction is then saved by the Precise application to be used as *input#d*. The contour points creation result and plane direction extracted is depicted in Figure 6d using the pelvis test case as example. Finally, the user needs to place the pins to fix the SCG on the healthy bone. This is done by adding cylinders of the desired diameter in the 3D environment (“insert geometry” area highlighted in red in Figure 4) and positioning them in the desired position/orientation thanks to the same 3D tool used for the planes (visible in Figure 6c and 7e). When the pins are positioned and the final solution is confirmed by the clinicians, the *inputs#a/e* can be exported to be used in the following modelling phase.

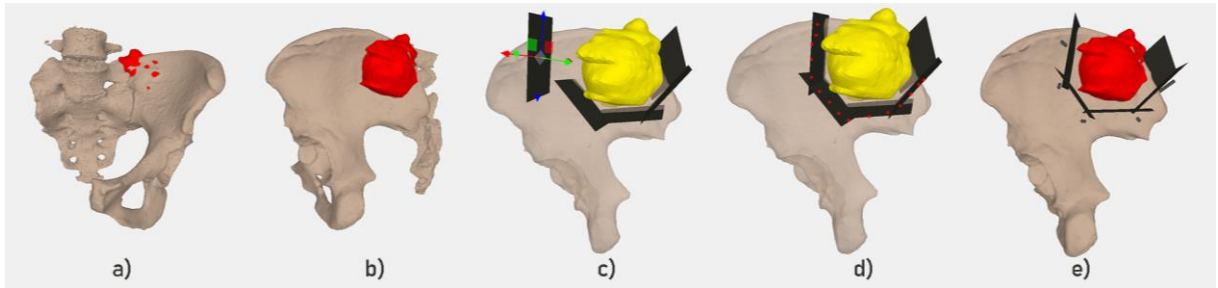


Figure 6: Steps performed in the Precise application on the Pelvis test case.

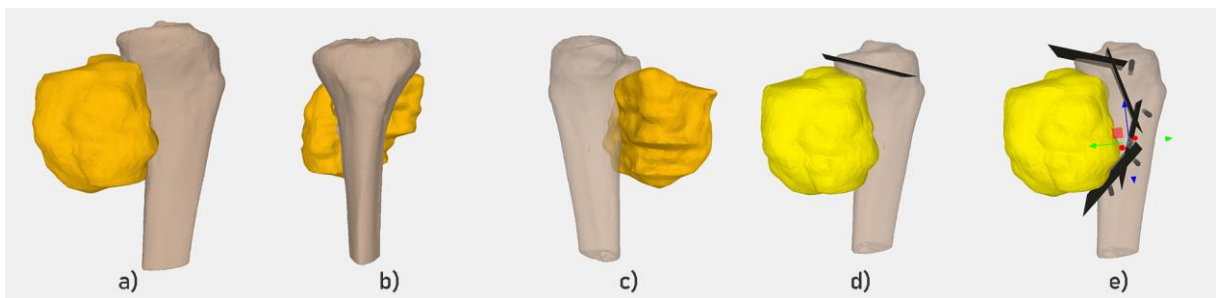


Figure 7: Steps performed in the Precise application on the Tibia test case.

4.1.2 *nTopology* procedure

Once the *nTopology* project file designed in this study is opened by the user, the first operation to be performed is the import of the series of input geometrical information generated by the Precise application. To guide the user and to minimize the probability of accidental mistakes, the graphical user interface designed in the study presents a series of import blocks to the user on the left of the screen, in a dedicated section of the *nTopology* Notebook called “Inputs”, visible in Figure 8. For each input required by the procedure, the user needs to select the file exported in the correct format from the Precise application. Moreover, the user needs to import the bone STL model as well.

The actual modeling performed automatically by *nTopology* starts with the introduction of the desired user variables within the *nTopology* user interface; Table 1 reports the specific values used for the two test cases.

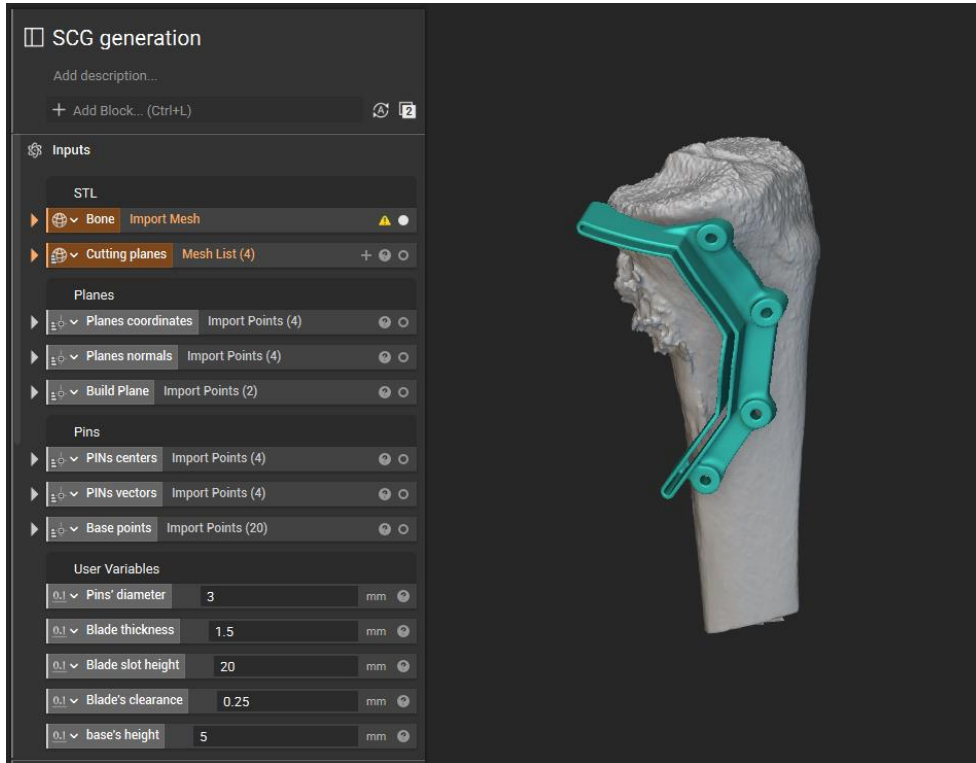


Figure 8: Inputs section of the nTopology procedure.

Variable	Value - Pelvis	Value - Tibia
Pins' diameter	3mm	3mm
Blade thickness	1.5mm	1.5mm
Blade slot height	20mm	20mm
Blade's clearance	0.25mm	0.25mm
Base's height	5mm	5mm

Table 1: User design variables.

These are the only operations that the user is required to actively perform in order to generate the final result. Indeed, as soon as nTopology has all the required input fields filled in with the correct information, the entire modeling workflow described earlier is applied automatically from start to finish. Accordingly, after a short time, depending on the complexity of the STL bone model and the performances of the PC running nTopology, the user can observe the final result and export the SCG STL file. As an example, with a Pc with i7 processor, GTX 1050 video card and 16 GB ram the average time is 2 minutes.

In detail, nTopology performs the series of steps previously described; some of these are depicted in Figure 9 for the Pelvis test case and Figure 10 for the Tibia. As previously described, the entire workflow is based on a series of boolean operations performed on the separate elements composing the SCG: Base, Pins' Holes and Blade Slot. Each element is separately modelled, often relying on an intersection with a surface obtained by offsetting the bone surface to determine the thickness of each part of the SCG with respect to the bone surface. As an example, Figure 9f depicts in orange the offset surface obtained from the Pelvis bone STL model to define the thickness of the

Blade Slot part of the SCG. Steps a through e in Figure 9 and Figure 10 presents the main milestones of the modeling operation, earlier described in Section 3, for the Pelvis and Tibia test cases. The bodies depicted in cyan in Figure 9d and Figure 10d have been used to subtract volumes to create holes and slot in the SCG, thanks to boolean subtractions. The procedure run without errors in both case studies, and the final SCG model was produced without the need for user interventions. The final SCG models are depicted in Figure 11.

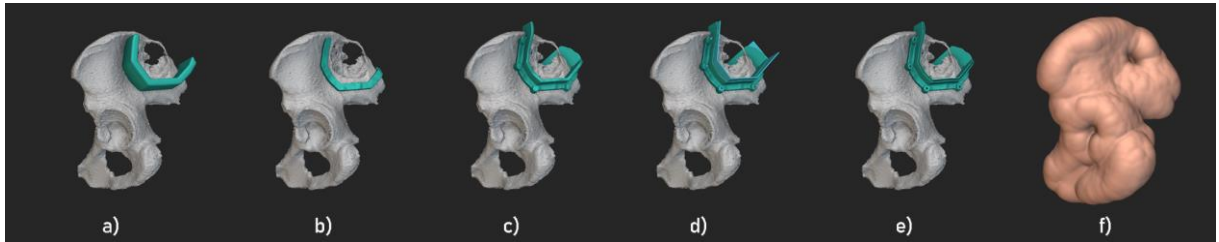


Figure 9: nTopology steps performed on the Pelvis test case.

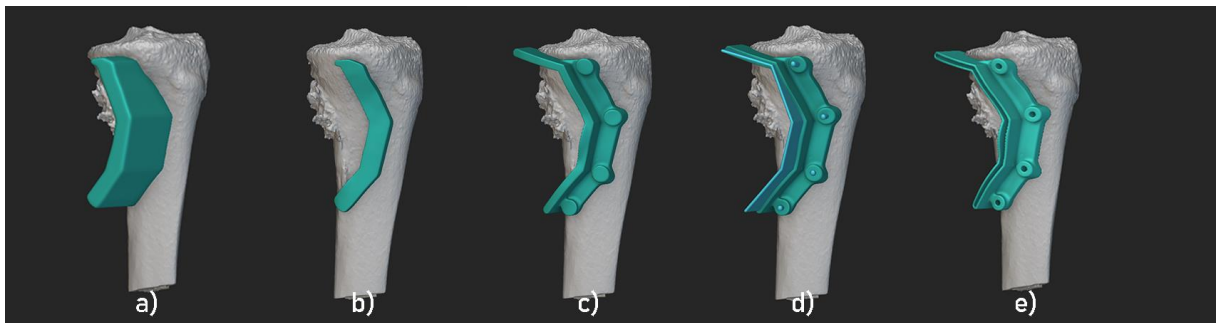


Figure 10: nTopology steps performed on the Tibia test case.

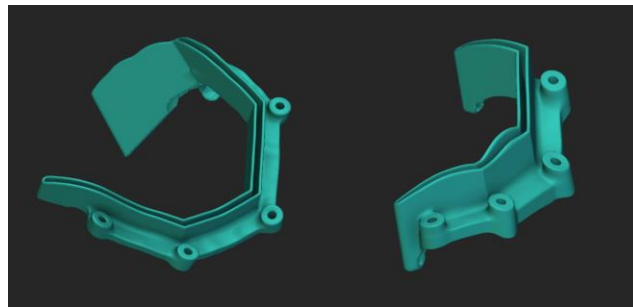


Figure 11: Final SCG 3D models for the Tibia (right) and Pelvis (left) test cases.

5 CONCLUSIONS

Innovative technologies such as Reverse Engineering and Additive Manufacturing tools has largely demonstrated to provide precise and accurate surgical instruments which help surgeons achieve better outcomes for their patients. In the orthopedic surgery field, the use of cutting guides directly modelled on the patient's anatomy, can for example reduce surgical time and enhance surgical efficiency, resulting in shorter hospital stays and faster recovery times, among other advantages. In

this context, this work proposed a new collaborative workflow for the design of such customized surgical tools. The need to implement such tool and its contextualization to the current medical software landscape is thoroughly analyzed and explained in [2]. Specifically, two modules were presented, i) a web application in which the surgeon can introduce key elements for the construction of guides directly on the 3D model in a simple and intuitive way and ii) a procedure automated within nTopology software to systematize the CAD modeling of surgical guides. The two modules are interconnected as they were developed with the idea of cooperating in the creation of a specific custom guide consisting of a slot to guide the cutting tool and holes for fastening to the bone. In fact, the *Precise* application was developed based on two important properties, namely, the ability to create an environment for navigating and manipulating 3D objects that is easily accessible even to non-expert users and the ability to enter all the necessary inputs to the automatic procedure.

The entire modeling procedure, from the initial anatomical data to the modeling of the personalized guide, was tested on two significantly different bone tumor cases: a Pelvis tumor and a Tibia tumor. In both cases, no difficulties were encountered by the surgeon in entering all the necessary elements into the *Precise* software and in using the program features. Furthermore, in both cases the modelling procedure automated in nTopology encountered no problems in generating the file.

This work thus lays the foundation for a systematized yet personalized vision of care in which the manufacturing processes of surgical medical devices can be greatly streamlined and accelerated, skipping steps to date compelled by the need for expert CAD modelers in the manufacturing process and the infeasibility for many physicians to manipulate three-dimensional objects.

6 ACKNOWLEDGMENTS

The authors wish to acknowledge the Tuscany Region (Italy) for co-funding the PRECISE project (BANDO RICERCA SALUTE 2018), which originated and made possible this research.

Michaela Servi, <http://orcid.org/0000-0002-4071-6615>
 Francesco Buonamici, <http://orcid.org/0000-0001-5186-9724>
 Lorenzo Guariento, <http://orcid.org/0000-0001-6130-5834>
 Angela Busato, <http://orcid.org/0000-0003-4777-0689>
 Yary Volpe, <http://orcid.org/0000-0002-5668-1912>
 Maurizio Scorianz, <http://orcid.org/0000-0002-7164-065X>
 Domenico Andrea Campanacci, <https://orcid.org/0000-0002-3912-3943>

REFERENCES

- [1] Buonamici, F.; Guariento, L.; Volpe, Y.: 3D Digital Surgical Planning: An Investigation of Low-Cost Software Tools for Concurrent Design, Lecture Notes in Mechanical Engineering, 2020, 765–775. https://doi.org/10.1007/978-3-030-31154-4_65/TABLES/6
- [2] Busato, A.; Servi, M.; Volpe, Y.; Furferi, R.; Scorianz, M.: Precise: A web-based 3D visualization and manipulation application for surgical planning of tumour resection, SoftwareX, 22, 2023, 101398. <https://doi.org/10.1016/j.softx.2023.101398>
- [3] Caiti, G.; Dobbe, J.G.G.; Strijkers, G.J.; Strackee, S.D.; Streekstra, G.J.: Positioning error of custom 3D-printed surgical guides for the radius: influence of fitting location and guide design, International Journal of Computer Assisted Radiology and Surgery, 13, 2018, 507–518. <https://doi.org/10.1007/S11548-017-1682-6/TABLES/6>
- [4] Ghai, S.; Sharma, Y.; Jain, N.; Satpathy, M.; Pillai, A.K.: Use of 3-D printing technologies in craniomaxillofacial surgery: a review, Oral and Maxillofacial Surgery, 22, 2018, 249–259. <https://doi.org/10.1007/S10006-018-0704-Z/FIGURES/5>
- [5] Guariento, L.; Buonamici, F.; Marzola, A.; Furferi, R.; Volpe, Y.: A Semi-Automatic CAD

- Procedure to Design Custom-made Surgical Cutting Guides, 2021, 123–127. <https://doi.org/10.14733/CADCONF.2021.123-127>
- [6] Guariento, L.; Marzola, A.; Scorianz, M.; Ospedaliera Careggi, A.; Yary Volpe, F.: Design Automation of Lattice-based Customized Orthopedic for Load-bearing Implants, 2023, 308–312. <https://doi.org/10.14733/cadconf.2022.308-312>
- [7] Javaid, M.; Haleem, A.: Additive manufacturing applications in medical cases: A literature based review, Alexandria Journal of Medicine, 54, 2019, 411–422. <https://doi.org/10.4314/bafm.v54i4>.
- [8] Louvrier, A.; Marty, P.; Barrabé, A.; Euvrard, E.; Chatelain, B.; Weber, E.; Meyer, C.: How useful is 3D printing in maxillofacial surgery?, Journal of Stomatology, Oral and Maxillofacial Surgery, 118, 2017, 206–212. <https://doi.org/10.1016/J.JORMAS.2017.07.002>
- [9] Petzold, R.; Zeilhofer, H.F.; Kalender, W.A.: Rapid prototyping technology in medicine—basics and applications, Computerized Medical Imaging and Graphics, 23, 1999, 277–284. [https://doi.org/10.1016/S0895-6111\(99\)00025-7](https://doi.org/10.1016/S0895-6111(99)00025-7)
- [10] Tran, K.L.; Kwon, J.Y.; Gui, X.Y.; Wang, E.; Yang, D.; Durham, J.S.; et al.: Virtual surgical planning for maxillary reconstruction with the scapular free flap: An evaluation of a simple cutting guide design, Head & Neck, 45, 2023, 115–125. <https://doi.org/10.1002/HED.27214>
- [11] Park, J.W.; Kang, H.G.; Lim, K.M.; Park, D.W.; Kim, J.H.; Kim, H.S.: Bone tumor resection guide using three-dimensional printing for limb salvage surgery, Journal of surgical oncology, 118(6), 2018, 898-905. <https://doi.org/10.1002/jso.25236>
- [12] nTopology, <https://ntopology.com/>, Next-Generation Engineering Design Software.
- [13] gkjohnson/three-mesh-bvh, <https://github.com/gkjohnson/three-mesh-bvh>, A BVH implementation to speed up raycasting and enable spatial queries against three.js meshes.
- [14] Three.js Tutorials, <https://sbcodes.net/threejs/measurements/>, Raycaster Measurements.