

Title:

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

Authors:

Michaela Servi, michaela.servi@unifi.it, University of Florence  
 Francesco Buonamici, francesco.buonamici@unifi.it, University of Florence  
 Lorenzo Guariento, lorenzo.guariento@unifi.it, University of Florence  
 Angela Busato, busato.angela@gmail.com, AOU Careggi  
 Yary Volpe, yary.volpe@unifi.it, University of Florence  
 Maurizio Scorianz, maurizio@scorianz.it, AOU Careggi  
 Domenico Andrea Campanacci, domenicoandrea.campanacci@unifi.it, AOU Careggi

Keywords:

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

DOI: 10.14733/cadconfP.2023.117-121

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 singular characteristics [4]. 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 [6]. 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 [2]. Different paths can be followed 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 [3]. 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. [5]) 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 inside nTopology - a CAD software based upon implicit modeling [7] - 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. Fig. 1 shows the workflow considered in the present work to produce the final SCG model.

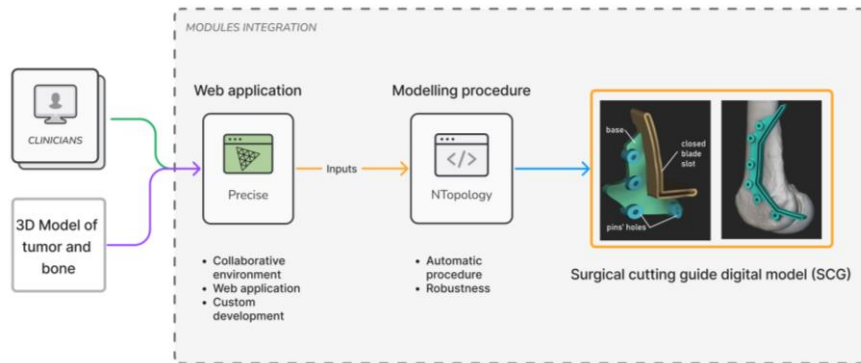


Fig. 1: Implemented modules for custom tumor resection planning.

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 Fig. 1): a) *Blade slot*: it is formed by an extrusion with a slot to guide the blade along the cutting surface 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.

#### 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.

Fig. 2 shows the architecture layers of the application, divided into two parts: frontend and backend, and the involved libraries.

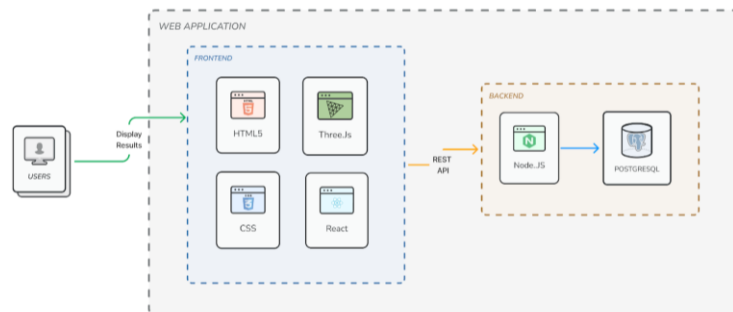


Fig. 2: Architecture of Precise web application.

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 processing.

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 draw on the model surface (useful in instances when the physician needs to mark anatomic areas that should not be touched by the SCG) [8] or taking measurements in space [9]. Finally, for some features it was necessary to define dedicated procedures, described below:

*Global alignment.* 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.* A dedicated routine was implemented for clipping with multiple planes. 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; 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.

*Offset.* The procedure creates the mesh of a selected STL model by offsetting faces and vertices by a user-specified value. 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. To define the inputs necessary for the modelling procedure, the user must initially import the

STL model of reference anatomy and tumor. Safety margins for resection can be taken into account by offsetting the tumor of an arbitrary value (yellow model in Fig. 3), then a series of planes can be placed and used to verify the cut with the clipping function either individually or all together (Fig. 3 shows an example of clipping with three planes). Once the clipping planes have been defined, the user proceeds by entering the points that indicate the path of the SGC base profile positioning them avoiding specific anatomical areas based on clinical considerations (points in red in Fig. 3). In order to model the guide and define the direction of extrusion of the *Base*, by clicking the dedicated button, these points are constrained to lie on a plane having the normal coincident with the current-view plane. Finally, the clinician can insert the pins in the desired position (grey elements in Fig. 3); 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.

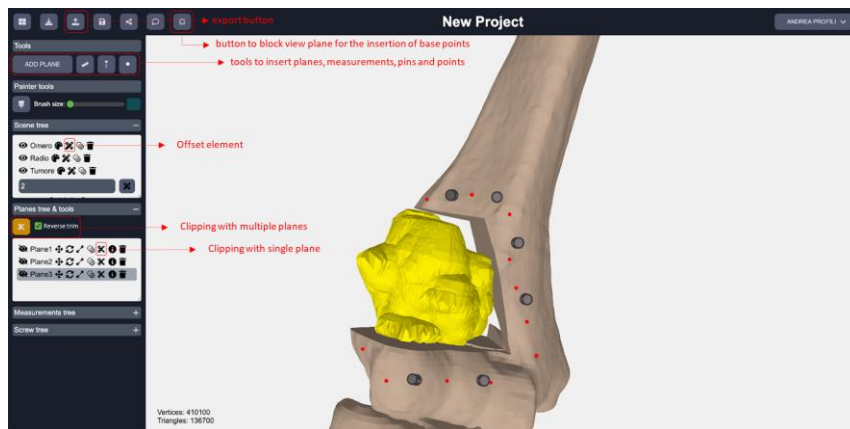


Fig. 3: Usage example of *Precise* application.

#### 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 [3]. 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 [2]. 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.

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 - Fig. 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. Fig. 4b 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 - Fig. 4c). Subsequently, the *internal pins' holes* and *internal blade slot* are subtracted from the resulting body (*Boolean subtraction* block - Fig. 4d) to create the final SCG model (Fig. 4e).

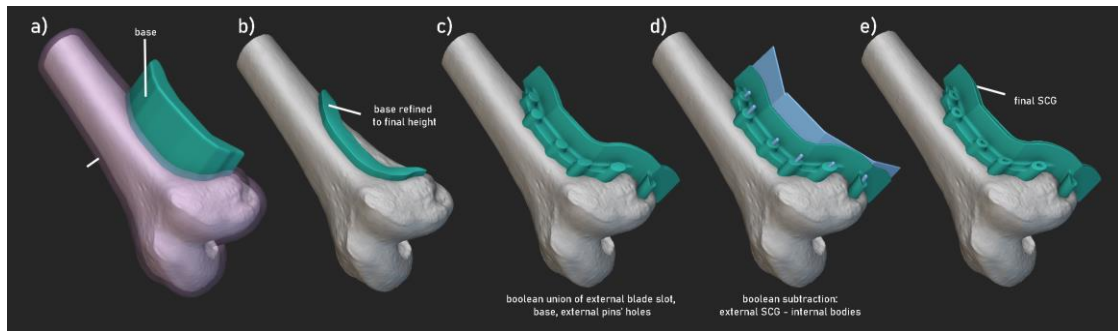


Fig. 4: SCG modelling steps inside nTopology.

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.

#### 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](https://doi.org/10.1007/978-3-030-31154-4_65/TABLES/6)
- [2] 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>
- [3] Guariento, L.; Marzola, A.; Scorianz, M.; Volpe, Y.: Design Automation of Lattice-based Customized Orthopedic for Load-bearing Implants, 2022, 308-312. <https://doi.org/10.14733/cadconf.2022.308-312>
- [4] 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)
- [5] Tran, K.L.; Kwon, J.Y.; Gui, X.Y.; Wang, E.; Yang, D.; Durham, J.S.; Prisman, E.: 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>
- [6] 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 Background and Objectives: The three-dimensional (3D)-printed bone tumor, 118(6), 2018, 898-905. <https://doi.org/10.1002/jso.25236>
- [7] Next-Generation Engineering Design Software | nTopology. <https://ntopology.com/> (accessed: 01/09/2023)
- [8] GitHub - gkjohnson/three-mesh-bvh: A BVH implementation to speed up raycasting and enable spatial queries against three.js meshes. <https://github.com/gkjohnson/three-mesh-bvh> (accessed: 01/09/2023)
- [9] Raycaster Measurements - Three.js Tutorials. <https://sbcode.net/threejs/measurements/> (accessed: 01/09/2023)