

**COB-2023-0754**

## **AUTOMATING VIRTUAL ORAL AND MAXILLOFACIAL SURGICAL PLANNING: DEVELOPMENT OF AN INTERFACE WITH PYTHON IN OPEN-SOURCE SOFTWARE**

**Dayanna Axly Santiago Villantoy**

Universidade Estadual de Campinas, Cidade Universitária Zeferino Vaz, Campinas-SP, Brasil  
d169087@dac.unicamp.br

**Livia Helena Martineli Teixeira**

**Denize Vilela Novais**

**Pedro Yoshito Noritomi**

**Marcília Valéria Guimarães**

Centro de Tecnologia da Informação Renato Archer, Dom Pedro I Highway (SP-65), Campinas-SP, Brasil  
lteixeira@cti.gov.br, dnovais@cti.gov.br, pnoritom@cti.gov.br, mvguimaraes@cti.gov.br

**Abstract.** *The use of three-dimensional (3D) models for virtual surgical planning is becoming increasingly relevant in the healthcare field and offers benefits such as cost reduction and surgical time. Although commercial Computer-Aided Design (CAD) software can obtain and manipulate 3D medical images, their high cost limits their use in complex surgical planning, especially in public hospitals. In this context, this article aims to develop a plugin in Blender, an open-source software, that integrates a Python interpreter to automate and customize biomodels used in complex virtual oral and maxillofacial surgical planning. The plugin generates mirrored replicas of regions affected by craniofacial dysfunctions. To achieve this, a customized panel was developed to optimize and reduce the time and steps required for a specific edition of the STL mesh, following the anatomical reference points of the 3D model without modifying the object's proportions with the performed edits. The validation method compares the generated mesh after using the plugin with the mesh model generated by the commercial software. Preliminary tests of the plugin's functionality have shown relevant results for complex virtual surgical planning of maxillofacial reconstructions. Therefore, this study contributes to the development of more accessible technologies for complex virtual oral and maxillofacial surgical planning in hospitals, using open-source software as an economical alternative to commercial 3D modeling software.*

**Keywords:** *3D Modeling, Virtual Surgical Planning, Plugin, Open Source Software, Python.*

### **1. INTRODUCTION**

Virtual surgical planning is a non-invasive process that uses medical images for the diagnosis, procedure selection, and treatment planning (Singh and Singh, 2021). This approach brings significant advantages, such as safer preoperative planning, reduced surgical time, and decreased hospital costs (Efanov *et al.*, 2018). The combination of this technique with three-dimensional (3D) printing, additive manufacturing, enables the creation of a personalized biomedical model of the patient, allowing the medical team to determine the best approach to be adopted in the surgery (Jang *et al.*, 2019). Additionally, this strategy facilitates the understanding of the patient's family regarding the surgical procedure to be implemented.

Oral and maxillofacial surgery is the healthcare field that benefits the most from this type of planning, especially in the treatment of traumas and injuries in the jaw, maxilla, face, and skull region (Jang *et al.*, 2019). In virtual surgical procedures, such as tumor resection and reconstruction of the affected area, mirroring is an important tool. This technique allows the overlay of the mirrored image of the healthy region onto the side affected by the lesion, providing the surgeon with precise delineation of the areas to be resected, as well as enabling proper planning to position plates and bone grafts in the reconstructed region (Zoabi *et al.*, 2022).

Despite virtual surgical planning being a promising approach, its use is limited by the high cost of commercial software, which hinders the use of this technology in public medical centers. In this context, the objective of this work is to develop a plugin in a free and open-source 3D modeling software for virtual oral and maxillofacial surgical planning. The plugin aims to optimize and reduce the time required to perform specific edits on 3D biomodels, such as mirroring healthy regions in areas affected by injuries. This aims to offer an economical alternative to commercial 3D modeling software, making virtual surgical planning more accessible to various professionals.

## 2. METHODOLOGY

To obtain the necessary 3D biomodel for virtual surgical planning, a set of processes must be followed, culminating in the printing of the 3D model. An essential part of integrating the healthcare field with 3D technologies is understanding the surgical approach to be adopted by the physician in a specific case, in order to provide an appropriate 3D model. In this regard, the process begins with the surgeon's request and the receipt of medical data in the Digital Imaging and Communications in Medicine (DICOM) format, which is obtained through Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) scans. Using this data, image segmentation is performed using the open-source software InVesalius 3.1 to generate a model in the Standard Triangle Language (STL) format. Next, the model undergoes editing and adjustments according to the specified requirements. Subsequently, the biomodel requires approval before proceeding with the 3D printing.

Since this work is being developed at the Centro de Tecnologia da Informação Renato Archer (CTI), a case provided by a surgeon of a patient with ameloblastoma, a benign tumor in the left facial region, was used to apply the interface tools. The flowchart shown in Figure 1 represents the modeling steps that will be followed until 3D printing. However, the crucial part of this process and the focus of this work lies in virtual surgical planning, as this step ensures that the three-dimensional model becomes an appropriate prototype to guide the surgery.

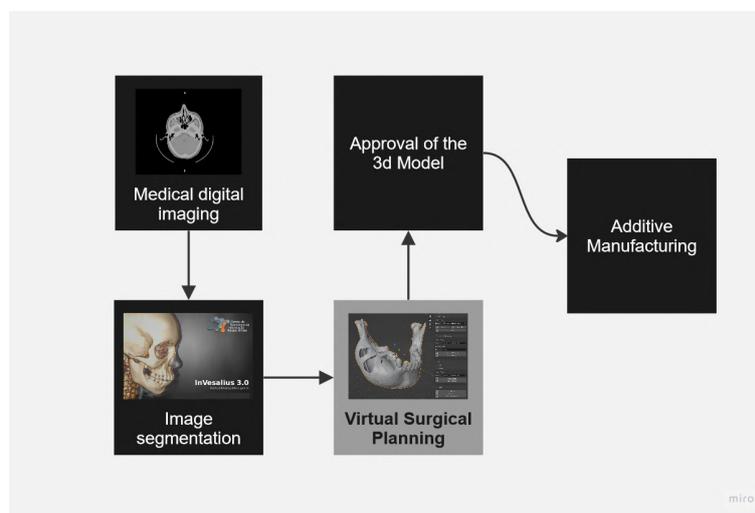


Figure 1. Process to obtain a 3D biomodel for use in surgical planning

### 2.1 Virtual Surgical Planning (VSP)

The image segmentation process is employed to transform 2D images into a 3D surface. Often, these images contain noise or require corrections, making it essential to manipulate the 2D images to delineate the region of interest.

The software used for medical image segmentation was Invesalius (Amorim *et al.*, 2011; Sannomiya *et al.*, 2008), free software developed by CTI, which generates the data for the 3D virtual model. To generate the biomodel, image segmentation was performed by setting a specific contrast threshold for each type of tissue or material. This allows for selecting only the pixels in the image that fall within this threshold, for example, using a threshold of 300 to identify the bone tissue. Considering that the region of interest is the mandible, only the pixels corresponding to that region were selected, and using the available tools in Invesalius, it was possible to select and separate the disconnected regions, as shown in Fig. 2.

After making the necessary adjustments to the images, we exported the 3D surface data in the STL format. This format contains information about the model's geometry and converts the surface into a set of triangles that form a mesh. Additionally, the STL format is widely used in 3D printing.

To provide a 3D model that meets the surgeon's needs, it is important to use 3D modeling software for more complex editing and adjustments. However, commercial software available in the market is often expensive, making it challenging for public healthcare centers to acquire them.

As a solution to this issue, a technological solution was developed to automate virtual craniofacial surgical planning using open-source software. Blender, selected as the software for developing this planning tool, is a free and open-source 3D creation suite that includes a Python interpreter through the Application Programming Interface (API). With the help of this Python interpreter, a customized user interface was developed, which provides tailored functionalities for importing and exporting the .stl file of the biomodel, as well as combining modifiers and tools.

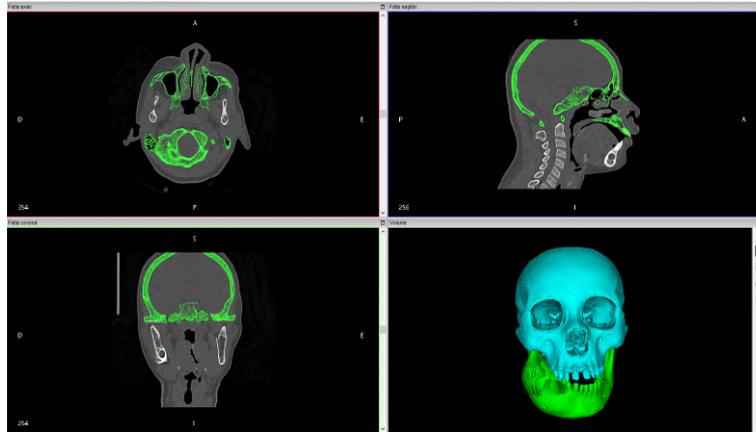


Figure 2. Image segmentation in InVesalius, with the mandible highlighted in green.

## 2.2 Functionalities

The custom interface developed for surgical planning applications currently includes the following features: Importing STL files, Mesh preparation for editing, Mirroring, Mesh corrections, Inserting identification, Inserting support. These features were chosen because they are suitable for meeting the needs of the cases sent to the virtual surgical planning CTI. Figure 3 illustrates the mentioned interface, providing a visualization of the available features that will be further detailed in the following sections of the paper. It is important to emphasize that these features are not limited, allowing for the inclusion of others according to the specific needs of the applications.

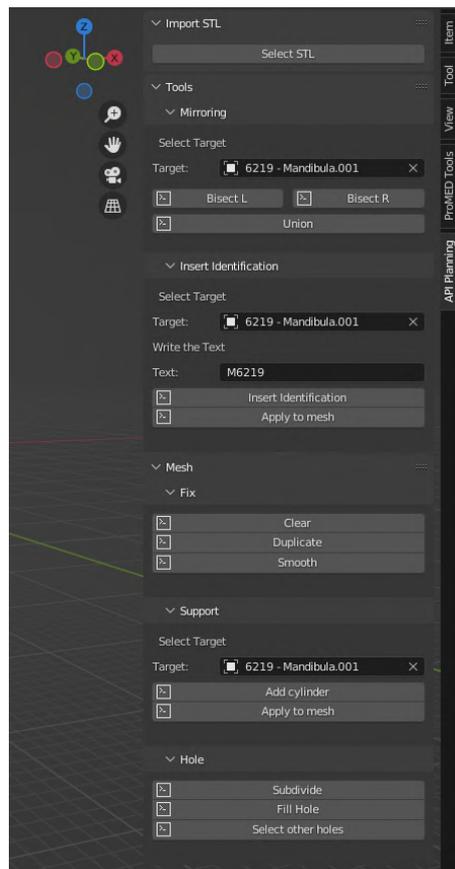


Figure 3. API interface 3D view Blender displaying a custom panel with functions developed for surgical planning.

Furthermore, the development of the custom API is based on the Python programming language, which is widely known and used by various developers. This facilitates the improvement and editing of the functions already created, as seen in Figure 4, which shows a portion of the script that will make the functions and classes created available in the Blender API, thus appearing in the custom panel. Another advantage of developing the API in Python within an open-

source software is that it allows for expanding the number of tools visible in the panel and creating others with the aim of meeting specific demands required for the virtual surgical planning process, making the implementation more agile and flexible.

```
563 ##### Register and Unregister #####
564
565 from bpy.utils import register_class
566
567 def register():
568     #section menu
569     bpy.utils.register_class(Import_Stl)
570     bpy.utils.register_class(Ferramentas)
571     bpy.utils.register_class(Espelhar)
572     bpy.utils.register_class(Add_Text)
573     bpy.utils.register_class(Mesh01)
574     bpy.utils.register_class(Corrigir)
575     bpy.utils.register_class(Suporte)
576     bpy.utils.register_class(Furo)
577
578
579     #operation
580     bpy.utils.register_class(MESH_OP_ImportStl)
581     bpy.utils.register_class(SimpleOperator)
582     bpy.utils.register_class(Duplicar)
583     bpy.utils.register_class(BisectL)
584     bpy.utils.register_class(BisectR)
585     bpy.utils.register_class(Mirror)
586     bpy.utils.register_class(Union)
587     bpy.utils.register_class(Texto)
588     bpy.utils.register_class(Smooth)
589     bpy.utils.register_class(Preencher)
590     bpy.utils.register_class(Preencher_grade)
591     bpy.utils.register_class(Subdividir)
592     bpy.utils.register_class>SelectHole)
593     bpy.utils.register_class(Texto_01)
594     bpy.utils.register_class(Marcador)
595     bpy.utils.register_class(Suporte_Malha)
596
597 # Register Functions
598 types.Scene.Target = bpy.props.PointerProperty(
599     type= types.Object
600     poll= object_select_poll
601 )
602
603
604 #bpy.types.Object.my_tool = PointerProperty(type=VIEW3D_OT_Properties)
605
File: C:\Users\dayannav\Desktop\Text_API.py
```

Figure 4. Script snippet to register the classes containing the panel functions.

### 2.2.1 Import STL file

This functionality allows the user to import the STL file in a targeted and convenient manner, obtained previously by Invesalius. When using it, the software will open the computer's file explorer, where the user can choose the desired STL file. After opening the file, the interface redefines the origin of the 3D mesh, setting it to the center of mass and aligning it with the center of the Blender workspace axes. Figure 5 shows the Blender screen after selecting the option to import the STL model file.

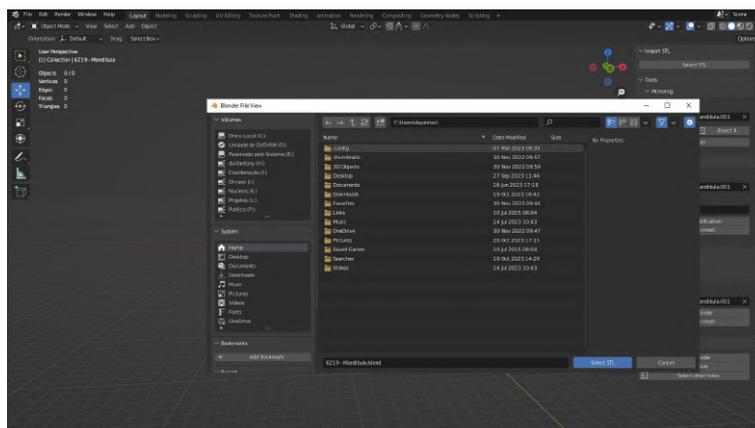


Figure 5. Open window to browse for .stl file through the panel.

### 2.2.2 Mesh

This part of the interface provides tools for editing the mesh without compromising its original dimensions, aiming to ensure the integrity of the mesh and avoid difficulties when printing the 3D model. These tools are: Fix, Hole, and Support.

The Fix tool has three applications. The first one is "Clean", which is useful for deleting any other external elements in Blender's object mode. In edit mode, it deletes disconnected edges, faces, or vertices from the mesh, which could

cause issues during printing. The second application is "Smooth," which makes the mesh more uniform and improves the model's visual appearance. The amount of smoothing can be adjusted by the user since it varies for each 3D model. Lastly, the "Duplicate" application creates a copy of the current mesh, allowing for editing tests without modifying the original mesh or comparing it with the mesh after modifications.

The Hole tool consists of three applications used to close holes in the mesh. "Select other holes" requires the edges forming the hole to be selected in Blender's edit mode. This tool then allows the selection of all other holes in the mesh. After that, the "Fill Hole" application creates a new smooth surface composed of a set of polygons that matches the shape of the model. It converts the polygons into triangles and reorganizes them to improve the mesh's geometry, as shown in Fig. 6. In some cases, the new surface created to fill the hole may have larger polygons than the other polygons in the mesh. In such cases, the "Subdivide" application can be used to resolve this issue.

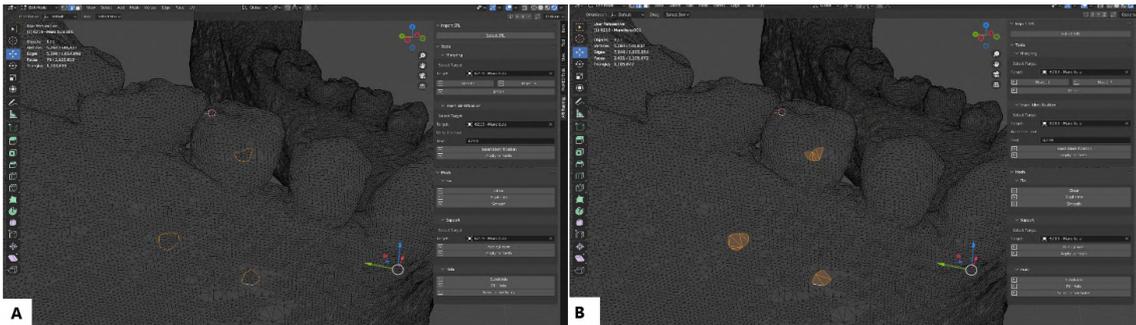


Figure 6. (A) Biomodel with all the holes selected. (B) Biomodel after using fermentation to seal the custom panel's holes.

Finally, the "Support" tool, which is essential in some cases where the biomodel has fragile or very thin regions. To improve stability when printing the model and prevent these fragile regions from breaking easily after the part's manufacturing, a rod is used to connect the fragile region to a more robust region of the 3D model. This support tool is essentially composed of a cylinder that connects two ends, and the tool developed for the panel simplifies this process by providing an option to add the cylinder.

The user can define the size and position, and then join the cylinder to the model to create a single 3D body. This can be seen in Figure 7, which shows a cylinder already placed on the model, connecting the two regions of the condyle because they have thinner walls.

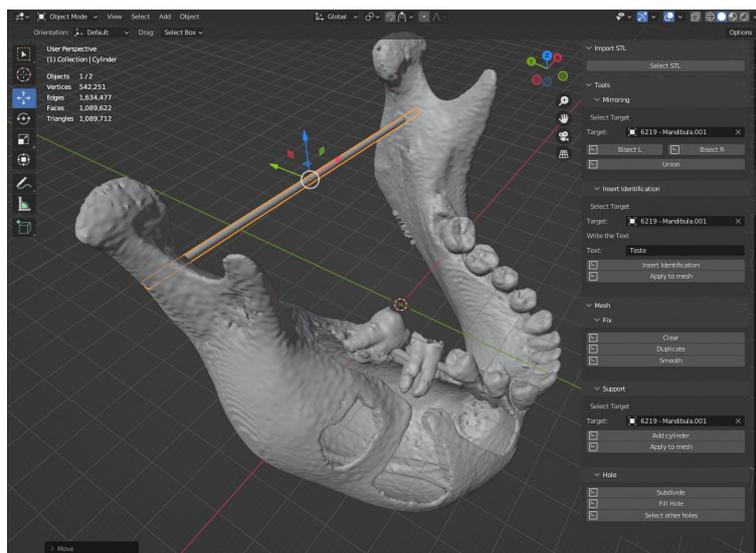


Figure 7. Biomodel after using the custom panel support tool.

### 2.2.3 Tools

These adaptations optimize and reduce the time required to perform specific edits on an STL mesh, providing efficiency to the VSP process. Thus, this API, in addition to facilitating the model editing process, includes a tool focused on the mirroring technique in regions affected by craniofacial dysfunctions. This function allows for obtaining a model that

replicates the healthy region in the affected area, which is essential for complex VSP of maxillofacial reconstructions. This enables visualization and preoperative simulation of the surgical outcome, as shown in Fig. 8.

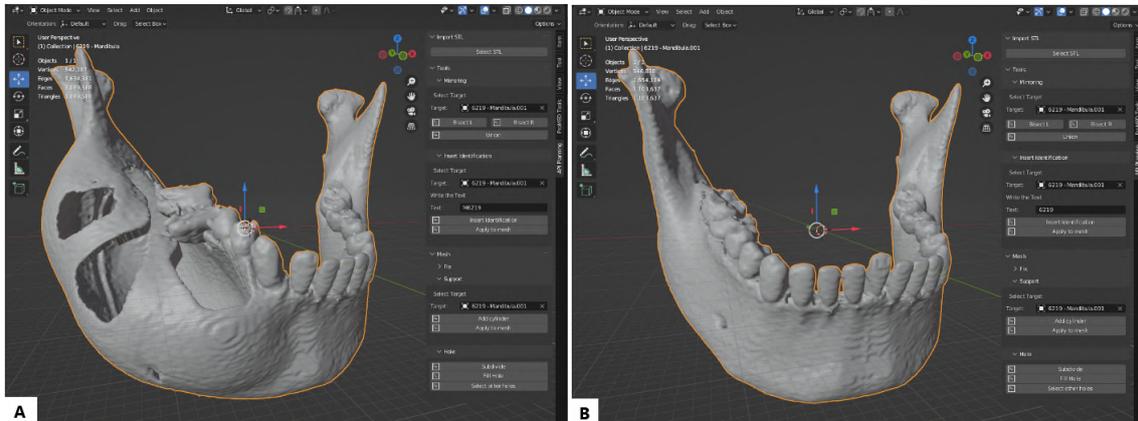


Figure 8. (A) Blender interface with the custom panel showing the jaw with the tumor, (B) Jaw without the tumor after using the mirroring tool from the custom panel.

This section of the interface contains more specific and complex tools, such as "Mirroring" and "Insert Identification". The Mirroring tool generates a new 3D model by mirroring the healthy region onto the affected region, allowing for comparison and visualization of how the region would look after surgical reconstruction. First, the mesh is selected as the target, and then the user can choose between mirroring the left region using "Mirror L" or the right region using "Mirror R". After that, a new model of the mirrored and healthy region is created, and the user can make any necessary adjustments before finalizing the process, as shown in Fig. 9. Finally, the "Union" option is used to merge the model of the healthy region with the mirrored region.

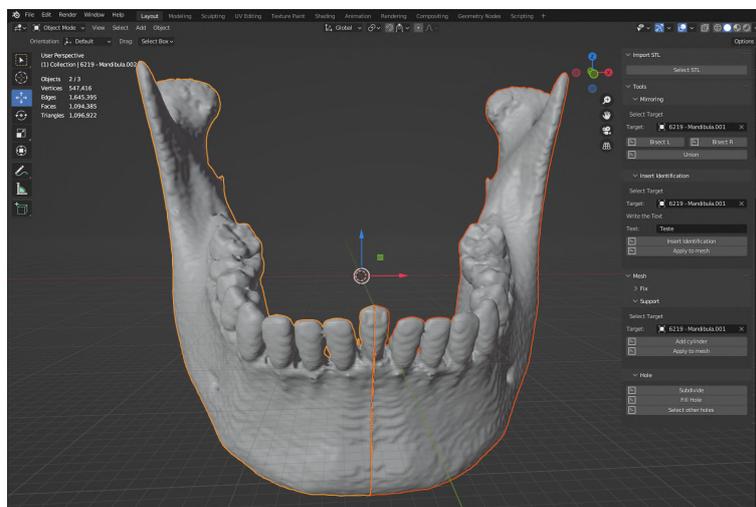


Figure 9. New 3D models created from the main model for mirroring. Mirrored region with yellow outline and normal region with orange outline.

Furthermore, this area of the interface also includes the "Insert Identification" tool. It allows for the inclusion of customized raised text on the mesh, serving as an identification for the 3D model, as shown in Fig. 10. In this function, the desired text is entered into the text box in the interface, and the text is converted into a mesh and then merged with the selected target mesh.

In the customized interface, first, the mesh where the text will be applied is selected as the target. Then, the desired identification text is entered into the text box, and the "Insert Text" option is selected. Finally, the "Apply to Mesh" option is used to transform the text into a 3D model and merge it with the selected target mesh.

### 2.3 Comparison between meshes for model validation

In this step, specific reference areas were selected for the comparison between the mesh models generated by the plugin, referred to as MMP (Mirrored Mandible Plugin), and the mesh models generated by the commercial software, referred to as MCS (Mirrored Mandible Commercial Software). These areas correspond to specific anatomical landmarks

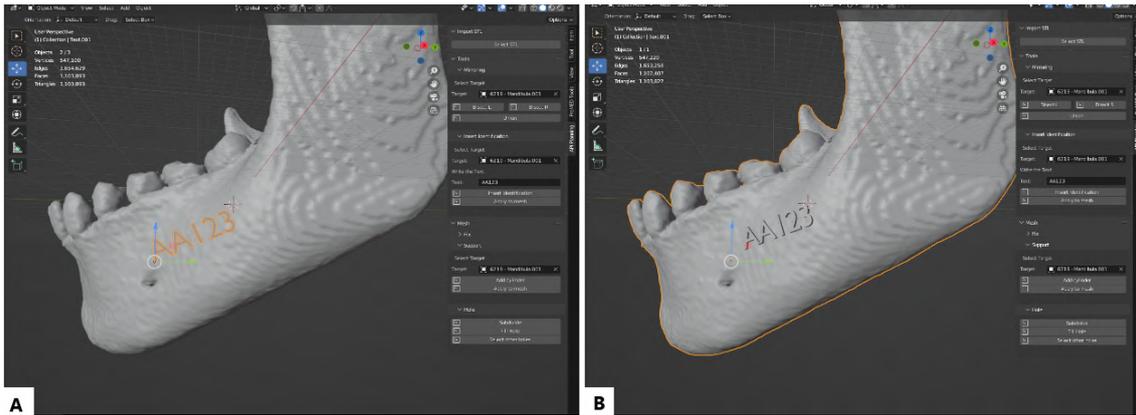


Figure 10. (A) 3D biomodel and 2D text, (B) New 3D model created that combines the 3D text and the previous 3D model.

of the mandibular region, including the right and left condyles, right mandibular angle, body, and ramus of the mandible (Frank and Netter, 2019), as depicted in Figure 11. The comparison was performed using the mandible mesh with the tumor based on the patient's tomographic images analyzed in this paper.

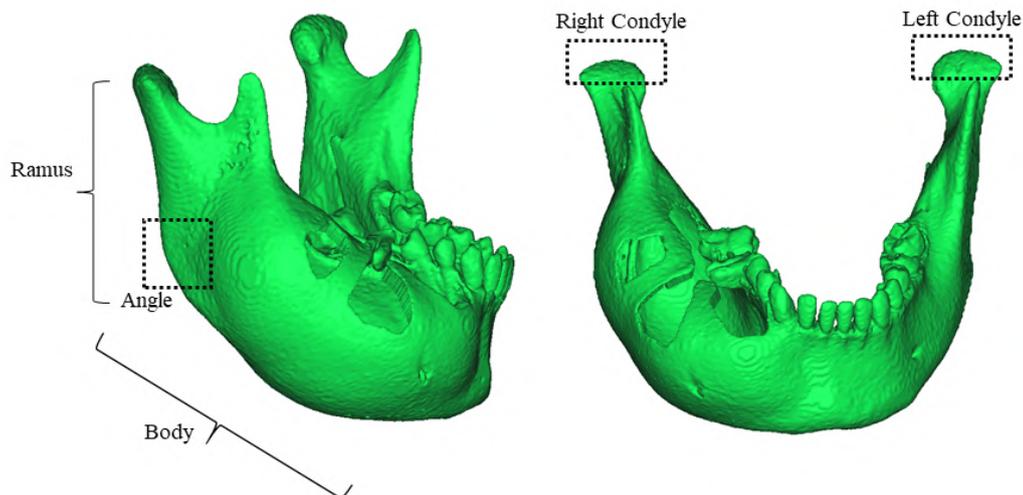


Figure 11. Views of the mandibles with the evaluated areas.

Once the comparison areas were defined, the validation of the plugin was performed using the open-source software CloudCompare. In this step, the MMP mesh was compared to the MCS mesh, which had already been approved by the responsible surgeon for the case, in order to determine possible differences between the models within the areas shown in Figure 11.

After importing the meshes into CloudCompare, the MMP and MCS mirrored mandibles were aligned according to Figure 12, with the MCS mesh serving as the reference for alignment with the MMP mesh. Alignment ensures that the meshes are properly positioned spatially for accurate comparison. Subsequently, the software performed a detailed comparison by analyzing the geometric attributes of each mesh, such as vertex positions and surface normals. In this way, the differences between the meshes were calculated, enabling the identification of regions with deviations.

### 3. RESULTS AND DISCUSSIONS

The purpose of this work is to develop a plugin for oral and maxillofacial virtual planning in free software. This tool serves as an alternative to commercial solutions, aiming to optimize planning time where procedures like mirroring the healthy region in place of the affected region are needed.

Based on the performed comparisons, it was possible to verify that the plugin generated a model with high similarity compared to the model generated in the commercial software, based on a qualitative analysis. For this case of ameloblastoma tumor, the usual procedure reported in the literature (Turek *et al.*, 2021; Davies *et al.*, 2019; Sannomiya *et al.*, 2008) is the tumor resection with a safe margin. After the resection, mandibular reconstruction is generally performed using pre-bent metal plates based on a mirrored 3D-printed biomodel from the virtual planning. The basis for mirroring is to maintain

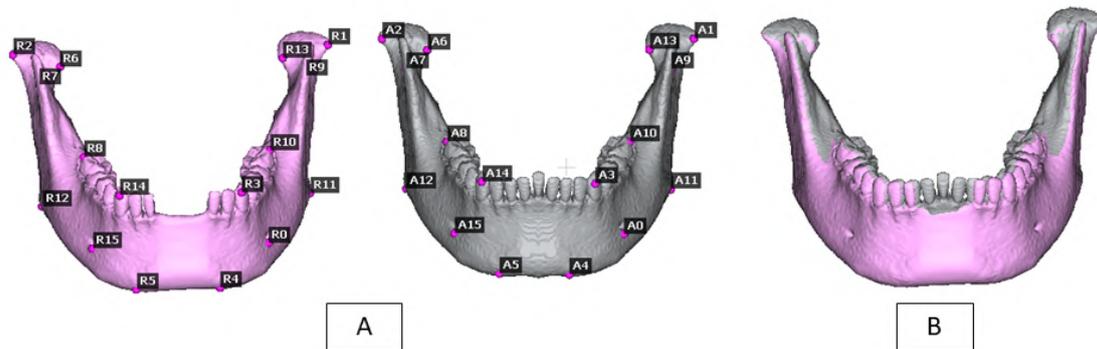


Figure 12. (A) Alignment points, (B) aligned meshes.

the condyle position and the proximity of the mandibular body and ramus with the original model.

The two compared models were within the same scale, with small deviations. In Figure 13, it was observed that the regions of the mandibular angle and body on both sides were preserved. In the condylar regions, there was a small deviation of the MMP compared to the MCS, more prominent on the right side of the mirrored tumor. The largest deviation, as shown in Figure 13, was in the anterior teeth region, but this region was not necessary for planning in this case.

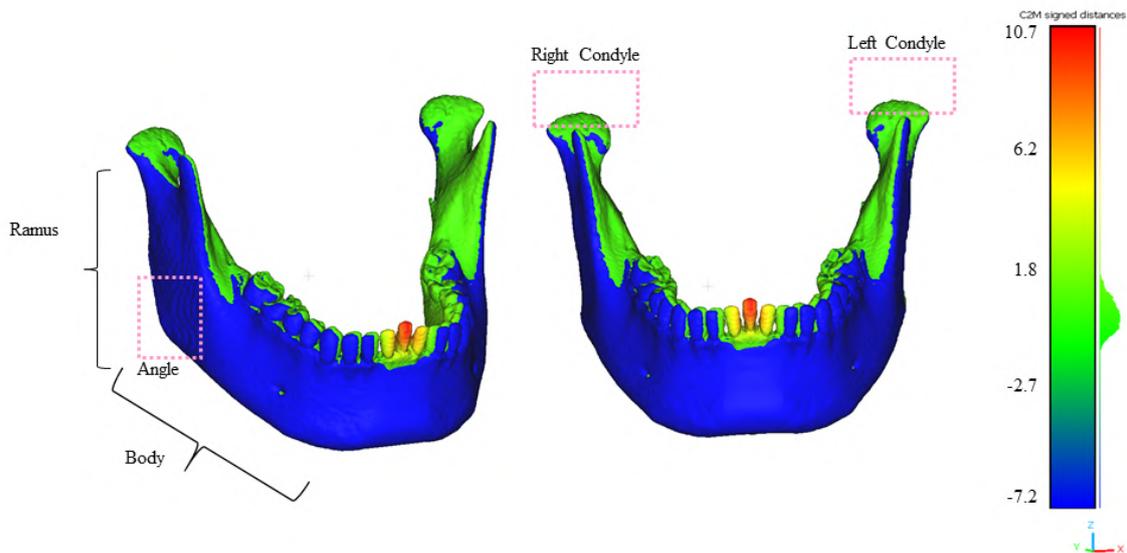


Figure 13. Views of the jaws with the evaluated areas with the color scale unit in millimeters (mm).

#### 4. CONCLUSION

In this study, the development stages of a plugin for VSP of oral and maxillofacial lesions, which require the use of a mirroring technique, in a free 3D modeling software with an integrated Python platform were presented. The preliminary results, qualitatively evaluated on a mandible with ameloblastoma, were promising. A satisfactory similarity was observed between the mirroring model generated by the plugin and the model generated by expensive commercial software. Based on this, it can be concluded that the developed plugin may represent a promising alternative to commercial tools for VSP. Furthermore, being developed in free software, it offers an accessible solution for public healthcare centers. The next steps of this work include validating the plugin in different cases of mandibular lesions and expanding it to areas such as the maxilla and orbit.

## 5. ACKNOWLEDGEMENTS

The authors would like to thank CNPq and CTI Renato Archer for the funding and technical support provided for this study and our parents. They would also like to express their gratitude to Prof. Dr. José Wilson Magalhães Bassani for the financial support to attend the conference, their colleagues from LAPRINT - Thiago Palhares, Leonardo Machado, Deise Mara Gouvea, Isabelle Miki Ikuno, Marina Soler Donaire, and Otávio Henrique Junqueira Amorim - for their collaboration in this study.

## 6. REFERENCES

- Amorim, P.H., de Moraes, T.F., Azevedo, F.d.S. and da Silva, J.V., 2011. “Invesalius: Software livre de imagens médicas”. *Centro de Tecnologia da Informação Renato Archer-CTI, campinas/SP-2011-CSBC2011*.
- Davies, J.C., Chan, H.H., Jozaghi, Y., Goldstein, D.P. and Irish, J.C., 2019. “Analysis of simulated mandibular reconstruction using a segmental mirroring technique”. *Journal of Cranio-Maxillofacial Surgery*, Vol. 47, No. 3, pp. 468–472.
- Efanov, J.I., Roy, A.A., Huang, K.N. and Borsuk, D.E., 2018. “Virtual surgical planning: the pearls and pitfalls”. *Plastic and reconstructive surgery Global open*, Vol. 6, No. 1.
- Frank, H. and Netter, M., 2019. *Atlas of human anatomy*. Elsevier Inc.
- Jang, W.H., Lee, J.M., Jang, S., Kim, H.D., Ahn, K.M. and Lee, J.H., 2019. “Mirror image based three-dimensional virtual surgical planning and three-dimensional printing guide system for the reconstruction of wide maxilla defect using the deep circumflex iliac artery free flap”. *Journal of Craniofacial Surgery*, Vol. 30, No. 6, pp. 1829–1832.
- Sannomiya, E.K., Silva, J.V.L., Brito, A.A., Saez, D.M., Angelieri, F. and da Silva Dalben, G., 2008. “Surgical planning for resection of an ameloblastoma and reconstruction of the mandible using a selective laser sintering 3d biomodel”. *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology*, Vol. 106, No. 1, pp. e36–e40.
- Singh, G.D. and Singh, M., 2021. “Virtual surgical planning: modeling from the present to the future”. *Journal of clinical medicine*, Vol. 10, No. 23, p. 5655.
- Turek, P., Pakla, P., Budzik, G., Lewandowski, B., Przeszlowski, Ł., Dziubek, T., Wolski, S. and Frańczak, J., 2021. “Procedure increasing the accuracy of modelling and the manufacturing of surgical templates with the use of 3d printing techniques, applied in planning the procedures of reconstruction of the mandible”. *Journal of Clinical Medicine*, Vol. 10, No. 23, p. 5525.
- Zoabi, A., Redenski, I., Oren, D., Kasem, A., Zigran, A., Daoud, S., Moskovich, L., Kablan, F. and Srouji, S., 2022. “3d printing and virtual surgical planning in oral and maxillofacial surgery”. *Journal of Clinical Medicine*, Vol. 11, No. 9, p. 2385.

## 7. RESPONSIBILITY NOTICE

The following text, properly adapted to the number of authors, must be included in the last section of the paper:  
The author(s) is (are) solely responsible for the printed material included in this paper.