

SoftCut: a virtual planning tool for soft tissue resection on CT images

Ludovic Blache¹, Fredrik Nysjö¹, Filip Malmberg¹, Andreas Thor², Andrés Rodríguez Lorenzo², and Ingela Nyström¹

¹ Centre for Image Analysis, Uppsala University, Sweden

`ludovic.blache@it.uu.se`

² Plastic & Oral and Maxillofacial surgery, Department of Surgical Sciences, Uppsala University, Sweden

Abstract. With the increasing use of three-dimensional (3D) models and Computer Aided Design (CAD) in the medical domain, virtual surgical planning is now frequently used. Most of the current solutions focus on bone surgical operations. However, for head and neck oncologic resection, soft tissue ablation and reconstruction are common operations. In this paper, we propose a method to provide a fast and efficient estimation of shape and dimensions of soft tissue resections. Our approach takes advantage of a simple sketch-based interface which allows the user to paint the contour of the resection on a patient specific 3D model reconstructed from a computed tomography (CT) scan. The volume is then virtually cut and carved following this pattern. From the outline of the resection defined on the skin surface as a closed curve, we can identify which areas of the skin are inside or outside this shape. We then use distance transforms to identify the soft tissue voxels which are closer from the inside of this shape. Thus, we can propagate the shape of the resection inside the soft tissue layers of the volume. We demonstrate the usefulness of the method on patient specific CT data.

1 Introduction

Resection of soft tissue is commonly performed in head and neck cancer surgery. By removing the tumor, a defect in the face is created consisting of different tissue layers (*e.g.*, skin, fat, muscle, or bone). To reconstruct this defect it is usually needed to transplant vascularized tissue from other parts of the body, typically the thigh. To enhance the quality and precision of the reconstruction, it is necessary to have an accurate estimation of the shape, dimension and volume of the defect after the resection.

In this context, we aim at developing a surgical planning software that can help the surgeons estimate the size and shape of a soft tissue resection, based on patient specific computed tomography (CT) data. Our goal is to propose an intuitive and interactive system that allows the user to sketch the pattern of the resection directly on a volumetric image of the patient's head, taken from a CT scan. After specifying the thickness of the tissue layers to remove, the

2

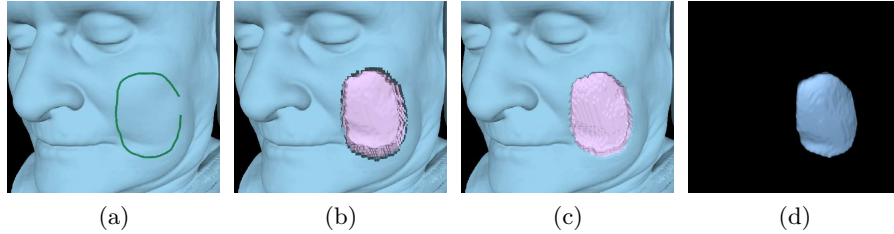


Fig. 1: **Resection workflow.** (a) Free-hand sketching. (b) Pre-visualization of the resection and depth fixing, where the skin is depicted in blue, internal soft tissues in purple, and bones in white. (c) Final volume carving. (d) Resected volume.

resection shape should then be automatically constructed by propagating this pattern into the soft tissue volume to appropriate depth. Rather than a realistic surgery simulation, our aim has been to develop a practical tool which allows to easily estimate the size and shape of a resection which can then be used as guide for flap harvesting.

The contributions presented in this article are the following:

1. an interactive tool to draw and set the depth of the resection,
2. an automatic volume carving method, based on distance transforms (DT),
3. a practical estimation of the dimensions of the resected volume.

2 Previous work

Computer-assisted solutions have been widely used for surgical planning since three-dimensional (3D) models became available for preoperative work-up [1]. This technology allows the surgeon to visualize patient specific images and to prepare several steps of surgical interventions. Such systems are particularly important in oral and maxillofacial surgery [2]. Most current solutions however focus on bone surgery [3] [4] [5] [6], including resections, osteotomies, fracture repair, or patient specific orthopedic implants. Concerning soft-tissue, some approaches have been proposed to interactively segment various organs, *e.g.*, [7] [8] [9], that often rely on region-growing or graph-based algorithms. However, these types of methods cannot be applied to segment a fragment of facial soft-tissues for resection planning. In this case, the interactive demarcation of the volume of tissues to cut-out can be implemented with a virtual carving tool. Such *volume sculpting* approaches have been used, *e.g.*, in surgery simulation [10].

Chen *et al.* proposed an interactive *peeling* tool [11] to *open* the volume data, allowing the user to access internal organs. This method uses a *point radiation* emitted from a free-hand planar mask to remove a user-specified depth layer of voxels from the rendering process. However, this technique carves the volume uniformly, without considering the variation in voxel intensity and thus the

type of tissues they represent. Ning *et al.* [12] described an interactive tool to cut volumetric data following arbitrary curves drawn on the screen. Xu *et al.* [13] developed another free-form cutting tool for prosthesis construction. These methods rely on a two-dimensional (2D) closed shape defined in screen space which is then projected through the volume data.

Our approach follows a similar principle as the method by Xu *et al.* [13], but allows to draw a free-hand shape directly on the 3D model surface, which enables to define complex shapes without being limited by the curvature of the surface. We also take advantage of the 3D grid structure of the data by using DTs to easily propagate the resection pattern inside the volume and fix the depth of the carving while avoiding carving of bone tissues.

3 Methods

The input data are patient specific images acquired through CT scans, and stored using the DICOM standard. The set of slice images forms a discrete volume represented by a 3D grid of voxels. The volume image is visualized using a surface rendering, based on a *ray-casting* algorithm [14], where the isovalue can be interactively changed in a sliding window. The rendered isosurface usually corresponds to the skin density value (*i.e.*, Hounsfield unit $\in [-863; -1015]$).

The method we propose contains the following steps as illustrated in Figure 1:

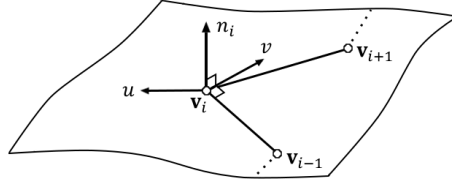
1. draw the shape of the resection on the skin's surface (see Section 3.1),
2. estimate which voxels of the volume will be inside the resection (see Sections 3.2 and 3.3),
3. fix the depth of the resection (see Sections 3.4 and 3.5),
4. compute the final shape of the resection (see Section 3.6).

3.1 Interactive drawing

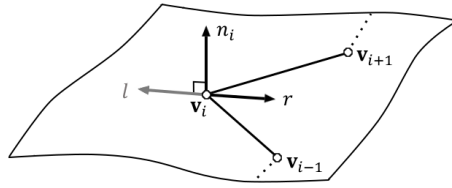
We propose to use a stencil-like tool to *paint* a free-hand shape on the isosurface. When the volume is rendered through the ray-casting operation, the rendering produces two output images: a shaded surface rendering image and a color-coded image. The surface rendering is sent to the framebuffer and displayed on the screen. In the second image, each point of the surface is rendered with a color where RGB values correspond to the XYZ position of the point in the 3D world space. This image is stored in a texture.

During the drawing process, the 2D position p of the mouse pointer on the screen is tracked and regularly recorded throughout time. At each time step t , we read the value of the texture pixel (texel) at position p_t in the stored texture. The given RGB values indicate the 3D position of the point of the surface hit by the ray cast from the camera viewpoint toward the mouse pointer. With this picking process, the 3D position of the projection of the mouse pointer on the surface is recorded (see Figure 1a). The mouse trajectory is thus transformed into a sequence of 3D vertices, indexed throughout time. Each pair of adjacent points

4



(a) Computation of vectors \mathbf{u} and \mathbf{v} . The local normal to the surface is noted \mathbf{n}_i .



(b) \mathbf{r} is obtained by projecting \mathbf{v} on \mathbf{u}_{\parallel} .

Fig. 2: Orthogonal vectors associated to a vertex v_i of the free-hand line strip defined on the isosurface. The final vectors \mathbf{l} and \mathbf{r} are oriented towards the left and right hand positions of the line strip, respectively (see text).

are connected by an edge and the complete set of vertices forms what we define as a *line strip*. This line strip is composed by n vertices noted $v_i, i \in [0, n - 1]$. When the drawing is completed, the strip is automatically closed by connecting the last vertex v_{n-1} with the first vertex v_0 .

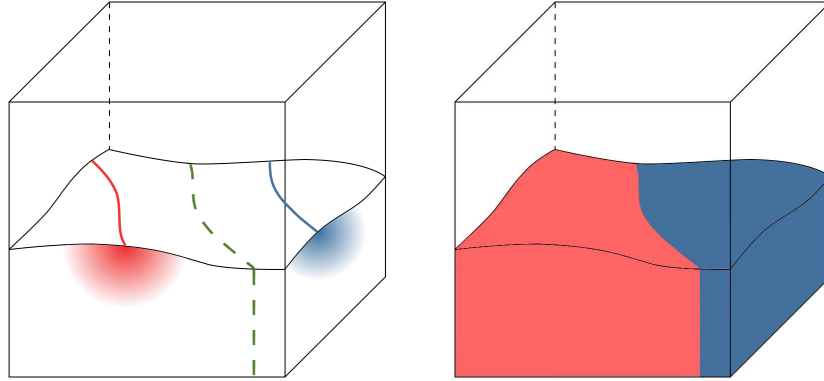
3.2 Defect pattern

This sequence of indexed vertices defines a polygon which is contained in the non-flat isosurface. To be able to identify which voxels are inside this shape, we compute an *external* and an *internal* line strip. These two strips define two shapes that corresponds to a dilation and an erosion, respectively, of the original line strip in the domain of the isosurface. We can then compare for each voxel its distance towards the external and internal line strip to find out whether it is inside or outside the shape (*i.e.*, closer to internal or external).

To compute these internal and external strips, we locally compute, for each vertex v_i of the initial line strip, two vectors as following:

$$\mathbf{u} = \frac{\mathbf{a} + \mathbf{b}}{\|\mathbf{a} + \mathbf{b}\|}$$

$$\mathbf{v} = \mathbf{a} \wedge \mathbf{n}_i$$



(a) Distance transforms toward external (red) and internal (blue) line strips (b) Voxels outside (red) and inside (blue) the resection

Fig. 3: Left: each voxel inside the object (*i.e.*, below isosurface) gets a Euclidean distance toward internal and external strips. The inside of the shape to carve is defined by the voxels which Euclidean distance toward internal line strip is smaller than distance to the external one. The limit of the resection (green) thereby follows the shape of the line strips. Right: internal voxels segmented according to their closest line strip.

where \mathbf{a} and \mathbf{b} are normalized vectors along the incoming edges, defined by $\mathbf{v}_i - \mathbf{v}_{i-1}$ and $\mathbf{v}_i - \mathbf{v}_{i+1}$, respectively. The vector \mathbf{n}_i is the local normal to the surface. In our case, we use as normal approximation the inverse gradient vector of the image intensity $-\nabla I_i$ at position of \mathbf{v}_i . The vector \mathbf{v} is therefore orthogonal to \mathbf{n}_i and \mathbf{a} and is oriented toward the *right-hand* direction of \mathbf{a} in the local tangent plane, as depicted in Figure 2a. The vector \mathbf{u}_{\parallel} is then computed by removing the normal component from \mathbf{u} :

$$\mathbf{u}_{\parallel} = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n}_i)\mathbf{n}_i$$

We then compute a new vector \mathbf{r} by projecting \mathbf{v} on \mathbf{u}_{\parallel} . Vector \mathbf{l} is finally defined as the opposite to \mathbf{r} , as illustrated in Figure 2b. The position $\mathbf{v}_i + \mathbf{r}$ corresponds to the right-hand point of \mathbf{v}_i and is set on the local tangent plane of the surface. The left-hand point is given by $\mathbf{v}_i + \mathbf{l}$. Note that \mathbf{u}_{\parallel} can be computed using either the local normal \mathbf{n}_i or an average normal $\bar{\mathbf{n}}$ which is the mean of all local normals. This choice induces a different carving effect as discussed in Section 4.

This operation is repeated for each vertex and we then obtain two new sequences of n vertices that we name right-hand and left-hand strips. We then compare the length of both line strips. Considering that the resected areas are approximately convex and defined by a single closed curve (cannot contain holes), the longest line strip is defined as the external and the shortest is internal. This

assumption forces the user to avoid drawing self-intersecting line strips, which would however be inconsistent from a surgery perspective.

3.3 Shape filling

Once the internal and external line strips are computed, the next step is to determine which voxels of the volume are closer to the internal strip than to the external strip. This way, we can propagate the contour of the resection, defined on the isosurface, inside the volume and compute the 3D shape of the defect.

We first compute a bounding box in which we will perform the next computations. We then repeat the following operations for each line strip (internal and external):

- Create a binary voxel grid with the same voxel spacing as the original data. For each pair of consecutive vertices v_i and v_{i+1} in the path, we set to 1 all the voxels along the segment $[v_i, v_{i+1}]$. The other voxels are set to 0.
- Compute a Euclidean distance transform (EDT) on this binary 3D image.

For each voxel, we then have access to two values which represent the Euclidean distance between the position of the voxel and the closest point of each strip (see Figure 3a). The shape of the resection is defined as the set of voxels which distance to internal strip is shorter than distance to external strip (see Figure 3b). We implemented an unsigned Euclidean DT as described by Saito *et al.* [15], rather than a Chamfer DT [16]. The reason is that we need to get as much precision as possible even in large distances to efficiently propagate the shape inside the volume, and that we might have to deal with anisotropic voxels.

We use a flood fill algorithm to segment all the voxels inside the resection shape. We only incorporate the voxels which intensity values in the 3D image correspond to soft tissues. The voxels which density values correspond to air (*i.e.*, outside of the isosurface) and bone are ignored and thus act as boundaries (see Figure 4). Note that any other density values can be maintained outside the shape (*e.g.*, blood vessel). We can then restrict the carving to a reduced range of density values instead of carving all the voxels uniformly as in the method by Chen *et al.* [11].

3.4 Mask

Once the shape of the resection is fixed, the user must set the *depth of cut* of the defect. This depth represents the thickness of the tissue layer that we wish to remove. We allow the user to interactively modify this depth using a 3D mask. We first setup a new binary volume in the bounding box. All the voxels which are closer from the internal strip and which intensity value is lower than isosurface (*i.e.*, outside the object) are set to 1. This filling is also achieved with a flood fill process, with an isosurface point as seed, to ignore the potential cavities inside the data. We then compute a new EDT on this binary image. We create a binary 3D mask of the data that represents the voxels contained in the resection (see

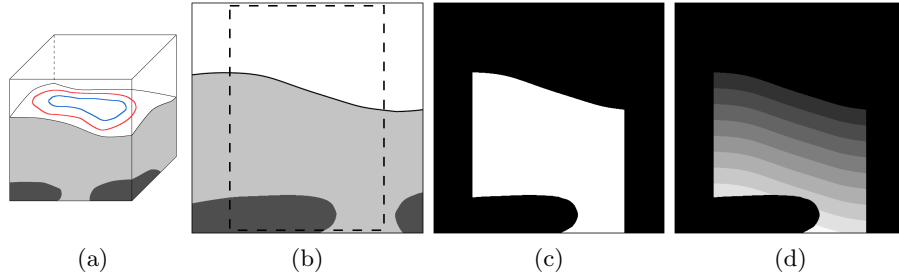


Fig. 4: (a) External (red) and internal (blue) line strips drawn on the isosurface of the volume which contains soft tissues (light grey) and bone (dark grey) intensity values. (b) A slice of this volume, showing the area of the defect (dashed box) deduced from distances toward both line strips. (c) Binary mask of the resection. (d) Soft tissue voxels are filled with distance toward isosurface.

Figures 4b and 4c). We then multiply this mask by the EDT. As a result, we obtain for each voxel inside the resection its distance toward the isosurface (see Figure 4d). By applying a threshold on this distance values, we can easily define a resection with a uniform thickness.

3.5 Interactive carving

As the depth of cut value can be interactively modified by the user, the defect created on the patient's face can then be previsualized on the image volume by thresholding the mask values. We use a direct volume rendering through GPU-based ray-casting [14]. The volume data and the mask are both stored in 3D textures. For a given voxel, we can then read its intensity value and its mask value from the fragment shader that performs the ray-casting. When a ray, cast from a pixel on the screen, hits a voxel which intensity value is below the isovalue, we check if it is located inside the area depicted by the mask. If so, the distance value read from the mask is compared with the depth of cut fixed by the user. If the distance associated with the voxel is below this depth, then the voxel is ignored and the ray-casting continues until we hit a voxel which distance is greater than threshold, or which is outside the mask (which means we hit the border of the resection). This allows the user to modify the depth of cut while instantaneously previsualizing the result of the resection (see Figure 1b).

When the shape and the depth of cut are satisfactory to the user, the volume can be carved. All the voxels contained in the defect shape and which mask value is below the threshold are transformed into void voxels (see Figures 1c and 5).

3.6 Resection estimation

After carving the defect into the volume, the shape of the resection itself can also be estimated by creating a new binary volume where only the voxels contained

8

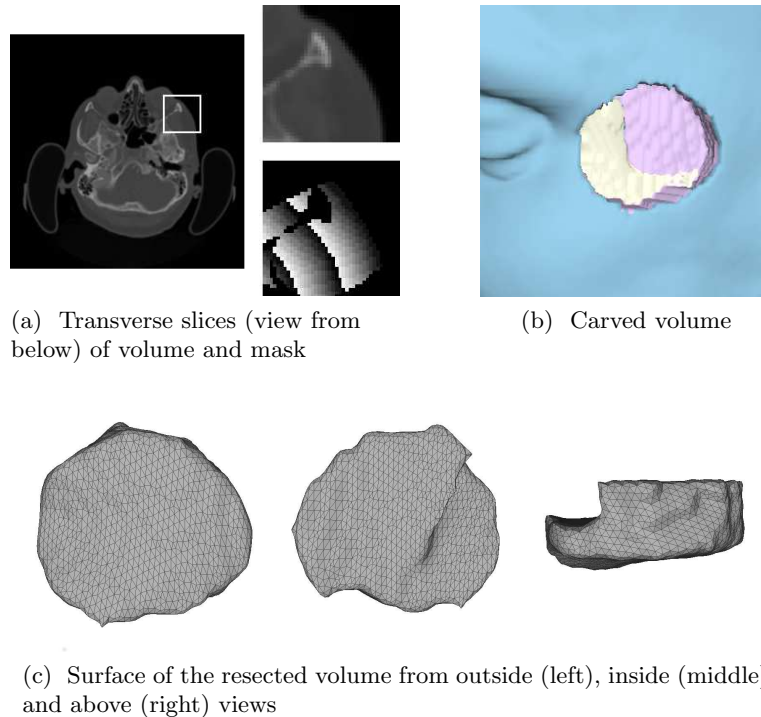


Fig. 5: **Example of a deep resection.** In (a), the transverse slice of a volume (top) and the associated mask (bottom) show that the distance function is not propagated in bone areas. The carved data (b) shows that the bone voxels (white) are not carved and thereby leave a print in the resected volume (c).

in the mask, and below the depth of cut, are set to a non-void value. The surface of this resection volume is then extracted using a marching cubes algorithm [17] to obtain a triangle mesh (see Figure 1d). This marching cubes reconstruction of a binary volume naturally leads to sharp edges on the surface. To avoid this, we apply a Laplacian smoothing algorithm [18] on the mesh. The 3D object we finally obtain can be visualized and measured by the surgeons to estimate the defect dimensions. It can also be physically reconstructed with a 3D printer. This will allow the surgeon to harvest a soft tissue flap that closely fits the defect created by the resection.

4 Results and discussions

We tested our method on several datasets, including the MANIX and PHENIX images from the Osirix DICOM image library ³. Both images were acquired

³ <http://www.osirix-viewer.com/resources/dicom-image-library>

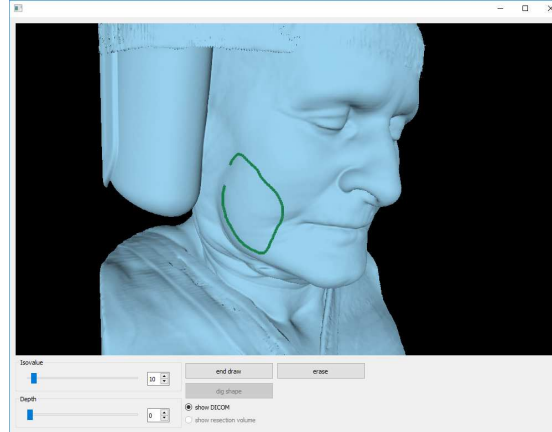


Fig. 6: Graphical user interface of our software

through CT scan. We used two versions of the MANIX image: a high-resolution of $512 \times 512 \times 460$ voxels and a low-resolution of $256 \times 256 \times 230$ voxels. The PHENIX image have a resolution of $512 \times 512 \times 361$ voxels. These volumes are similar to the data that our software is designed to handle and can then be used as proof of concept. Our experiments were performed on an Intel Core i7-6700HQ 2.6 GHz CPU with 16.0 GB of RAM and an Nvidia Quadro M1000M GPU.

The GPU implementation of the ray-casting [14] allows real-time rendering of the data. The previsualization of the resection (see Section 3.5) is achieved by thresholding the voxels following their mask values and is processed in the same fragment shader as the ray-casting. We can then maintain an average frame rate of 60 fps during the rendering of the volume. The EDTs used in our method (see Sections 3.3 and 3.4) are computed in a reduced bounding box, thus avoiding unnecessary computations on the whole volume. The computational time for the construction of the resection depends on the resolution of the volume and the size of the painted shape.

We recorded the computing time after drawing very large resections. Building and filling the shape, as described in Sections 3.2 and 3.3, take a maximum of 300 ms for the low-resolution volume. The building of the mask detailed in Section 3.4 is then computed in 400 ms. For high-resolution volumes, a large resection shape is computed in 4 s and the mask is built in 3.5 s. Noticeable computing times may occur with higher resolution volumes (*i.e.*, more than 512^3 voxels) during the EDT computations and the final carving (see Section 3.6), especially in the case of very large resection shapes. However, it would be limited to isolated operations and would not affect interactive visualization of the data. The user's satisfaction is high from the surgeons' point of view. Thanks to an intuitive user interface (see Figure 6), a resection mesh can be sketched and build in less than five minutes by an inexperienced user⁴.

⁴ http://www.cb.uu.se/research/haptics_applications/softcut

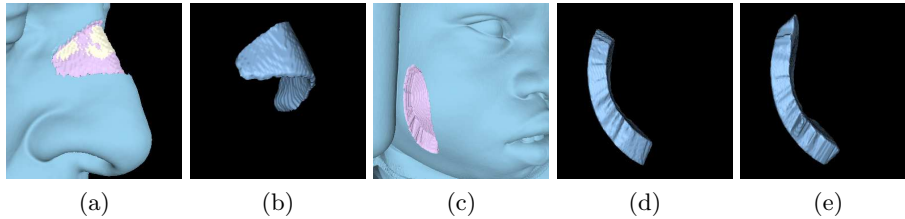


Fig. 7: Left: a 2.25 mm-thick resection on the nose of MANIX volume. Carved volume (a) and resection shape (b) show the high curvature of the surface. Right: large resection on the PHENIX volume (c) showing the different carving effect depending on computation using the local normal (d) or a unique average normal vector (e).

Our method was tested to draw and estimate various resection shapes. Note that the sketching of the shape processed by mouse picking can be interrupted and completed in several steps. This allows the user to move the 3D model and then continue painting from a better visualization viewpoint, which is useful to paint on high curvature parts of the surface (*e.g.*, nose of chin) as depicted in Figure 7 (left). As explained in Section 3.2, the normal vector used to compute the vectors pointing to right and left-hand positions can be a local normal estimation or an average normal computed on the complete resection shape. In the first case, the volume will be carved by propagating the sketched pattern along the local normals, which is adapted to high curvature areas. The second option will propagate the pattern toward a constant direction, which is adapted for flatter surfaces, as illustrated in Figure 7 (right).

The final 3D shape of the resected volume is exported as a 3D triangle mesh (see Section 3.6) using the STL file format. The resulting file can then be natively read by 3D printing tools. The usefulness of our method may be substantial as it would lower morbidity for patients, saving time in the operating room and making the choice of flap and availability for the soft tissue transfer better and more safely planned. Figure 8 shows an example of a resection shape manufactured with a Formlabs stereolithographic (SLA) 3D printer. The printing process was achieved in less than 2 hours. This printed object can then be used as a guide for skin flap harvesting during the reconstructive surgery. The surgeon can thus cut a soft tissue free flap with the right shape, size and thickness to closely fit the defect.

5 Conclusion and future work

In this paper, we have proposed a new approach for interactive drawing and cutting of soft tissue layers in volumetric medical images. This method can be used for surgery planning to estimate the shape and size of oncologic resection. We provide a painting tool to draw a free-hand closed curve on a patient specific

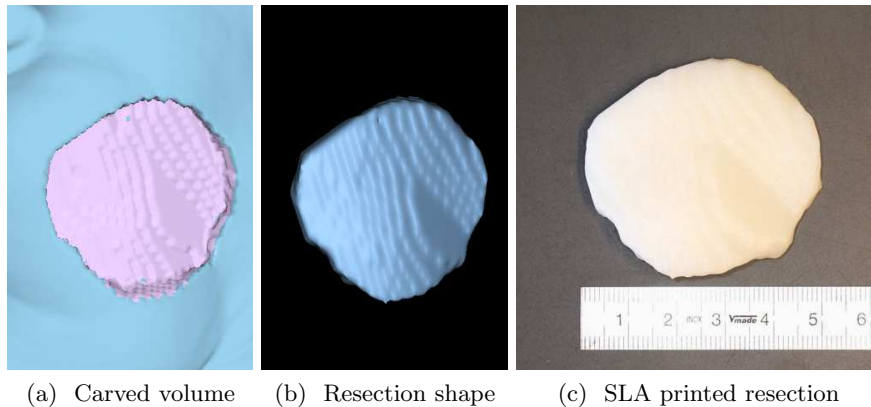


Fig. 8: Example of a 6.7 mm-thick resection planning on the cheek of MANIX volume.

face reconstruction. We then compute an eroded and dilated version of this contour which respectively correspond to the inside and the outside of the shape. Using Euclidean DT, we can then propagate the shape inside the volume and segment the voxels which belong to the inside of the resection. We also provide an interactive tool to modify the depth of the carving. Once the shape and depth are fixed, the volume can be virtually carved. We can then provide a 3D model of the resection to the surgeon.

Considering the high frame rate of the visualization, it is feasible to add more desired features. A planned future development is to include stereo-rendering combined with, *e.g.*, a haptics interface, to provide a virtual reality solution. Numerical methods, such as finite element methods (FEM), could also be used to develop a realistic simulation of soft tissue behaviour through resecting and flap harvesting.

References

1. Girod, S., Keeve, E., Girod, B.: Advances in interactive craniofacial surgery planning by 3D simulation and visualization. *International Journal of Oral and Maxillofacial Surgery* **24** (1995) 120–125
2. Sarti, A., Lamberti, C., Gori, R., Erbacci, G., Bassani, L., Bianchi, A., Marchetti, C.: Virtual planning of facial reconstructions. *Imaging Decisions MRI* **11** (2007) 29–38
3. Carrillo, F., Vlachopoulos, L., Schweizer, A., Nagy, L., Snedeker, J., Fürnstahl, P.: A time saver: Optimization approach for the fully automatic 3D planning of forearm osteotomies. In: *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI) Part II*. Volume 10434 of *Lecture Note in Computer Science (LNCS)*, Springer International Publishing (2017) 488–496

4. Ozdemir, F., Karani, N., Fürnstahl, P., Goksel, O.: Interactive segmentation in MRI for orthopedic surgery planning: bone tissue. *International Journal of Computer Assisted Radiology and Surgery* **12** (2017) 1031–1039
5. Zeng, C., Xing, W., Wu, Z., Huang, H., Huang, W.: A combination of three-dimensional printing and computer-assisted virtual surgical procedure for preoperative planning of acetabular fracture reduction. *Injury* **47** (2016) 2223–2227
6. Olsson, P., Nysjö, F., Hirsch, J., Carlbom, I.: A haptics-assisted cranio-maxillofacial surgery planning system for restoring skeletal anatomy in complex trauma cases. *International Journal of Computer Assisted Radiology and Surgery* **8** (2013) 887–894
7. Cordes, J., Dornheim, J., Preim, B., Hertel, I., Strauss, G.: Pre-operative segmentation of neck CT datasets for the planning of neck dissections. In: *Proceedings of SPIE Medical Imaging*. (2006)
8. Wang, G., Zuluaga, M.A., Pratt, R., Aertsen, M., Doel, T., Klusmann, M., David, A.L., Deprest, J., Vercauteren, T., Ourselin, S.: Slic-seg: A minimally interactive segmentation of the placenta from sparse and motion-corrupted fetal MRI in multiple views. *Medical Image Analysis* **34** (2016) 137–147
9. Smektala, T., Nysjö, J., Thor, A., Homik, A., Sporniak-Tutak, K., Safranow, K., Dowgierd, K., Olszewski, R.: Three-dimensional eyeball and orbit volume modification after LeFort III midface distraction. *The Journal of Craniofacial Surgery* **26** (2015) 1652–1655
10. Petersik, A., Pflessner, B., Tiede, U., Höhne, K.H., Leuwer, R.: Realistic haptic interaction in volume sculpting for surgery simulation. In: *Proceedings of International Symposium on Surgery Simulation and Soft Tissue Modeling (IS4TM)*. Volume 2673 of *Lecture Note in Computer Science (LNCS)*, Springer Berlin Heidelberg (2003) 194–202
11. Chen, H.L.J., Samavati, F.F., Sousa, M.C.: GPU-based point radiation for interactive volume sculpting and segmentation. *The Visual Computer* **24** (2008) 689–698
12. Ning, H., Yang, R., Ma, A., Wu, X.: Interactive 3D medical data cutting using closed curve with arbitrary shape. *Computerized Medical Imaging and Graphics* **40** (2015) 120–127
13. Xu, C., Li, S., Wang, L., Xiao, C.: CT image based 3D orbital prosthesis model construction method. In: *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. (2016) 204–208
14. Kruger, J., Westermann, R.: Acceleration techniques for GPU-based volume rendering. In: *Proceedings of the 14th IEEE Visualization. VIS '03* (2003) 287–292
15. Saito, T., Toriwaki, J.I.: New algorithms for Euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern Recognition* **27** (1994) 1551–1565
16. Borgefors, G.: Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing* **27** (1984) 321–345
17. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics* **21** (1987) 163–169
18. Taubin, G.: A signal processing approach to fair surface design. In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '95* (1995) 351–358