Anatomical Mesh-Based Virtual Fixtures for Surgical Robots*

Zhaoshuo Li[1], Alex Gordon[2], Thomas Looi[2], James Drake[2], Christopher Forrest[2], Russell H. Taylor[1]

Abstract— **This paper presents a dynamic constraint formulation to provide protective virtual fixtures of 3D anatomical structures from polygon mesh representations. The proposed approach can anisotropically limit the tool motion of surgical robots without any assumption of the local anatomical shape close to the tool. Using a bounded search strategy and Principle Directed tree, the proposed system can run efficiently at 180 Hz for a mesh object containing 989,376 triangles and 493,460 vertices. The proposed algorithm has been validated in both simulation and skull cutting experiments. The skull cutting experiment setup uses a novel piezoelectric bone cutting tool designed for the da Vinci research kit. The result shows that the virtual fixture assisted teleoperation has statistically significant improvements in the cutting path accuracy and penetration depth control.**

I. INTRODUCTION

Craniosynostosis occurs when one or more cranial sutures prematurely fuse. The fused cranial sutures do not allow for proper expansion of the skull as the child grows. It is estimated that the birth incidence of craniosynostosis is 0.4 per 1,000 births [1]. To avoid serious cognitive and cosmetic complications, these cranial sutures must be reopened by surgical intervention. This operation typically lasts for a long time (around 8 hours) and requires delicate tool movement to avoid incision-related complications [2].

Surgical robots, such as the da Vinci Surgical System (Intuitive Surgical, US), can mitigate the challenges by extending human capabilities. Virtual fixture (VF), i.e. software motion constraints, can further reduce the operational difficulties by allowing the surgeon and robot to work together to complete the surgical task with improved stability, reliability and precision [3]. VF can stop the surgeon from making sudden motions and accidentally damaging critical anatomies. Examples of VF in robot-assisted surgeries where high precision is required include neurosurgery [4] and intraocular surgery [5]. To the best of our knowledge, the use of the da Vinci system for craniofacial surgery has never been attempted.

Prior work has constructed complex virtual fixtures from simpler primitives in parametric forms. For example, Prada *et al.* [6] use a set of point VF to assist path following tasks. Part *et al.* [7] defines a hyper-plane aligned with an artery to assist dissection. Marinho *et al.* [8] select points on the patient's anatomy to construct VF and the corresponding vector-field inequalities are created to ensure that the surgical tool stays sufficiently far from the anatomy. The process of selecting and defining primitives can be extremely tedious and impractical if done manually before the surgery, due to the geometric complexity of anatomical structures.

Ren *et al.* [9] propose a VF system for cardiac surgery to protect the exterior of the patient's heart without manually defining any primitives. The proposed system first extracts a mesh representation of the heart using the Marching Cubes algorithm on the MR/CT images and then approximates the surface of the heart using a set of B-spline basis functions. While this approach works for modeling the exterior shape of the patient's heart, the accuracy of the B-spline approximation can degrade severely for other anatomical structures as the complexity grows. Li *et al.* [10] overcome the previous limitation by generating VF from an anatomical mesh directly at 30 Hz in the robot control loop. One limitation of this approach, however, is the assumption that the anatomical surfaces near the tool at any given time can always produce a set of convex linear constraints. Although this assumption is often valid, it breaks down in cases where the surgical tool must work in proximity to highly complex anatomical structures.

In this paper, we propose a new dynamic constraint formulation for the online generation of virtual fixtures from polygon mesh representations of complex anatomical structures in 3D. The proposed algorithm does not make any assumptions of the local concavity/convexity of the model relative to the tool. These models can be obtained preoperatively such as from CT [11], or intra-operatively such as from 3D ultrasound [12]. The proposed formulation can be used as a high-level control algorithm to limit the tool's motion within a "safe" region. It can be also used to provide haptic feedback in a teleoperated control setting.

The contributions of the paper can be summarized as follows:

- a dynamic constraint formulation for virtual fixtures of anatomies modeled by polygon meshes, without the assumption of convexity/concavity of the model in proximity to the tool,
- validation experiment of the proposed approach in simulation with various shapes,
- run time experiment of the proposed approach,
- VF assisted skull cutting experiments using a novel piezoelectric cutter designed for craniotomy.

The rest of this paper is organized as follows: Section II provides the necessary background of VF and Section III

*This work was supported in part by a research contract from Galen Robotics, in part by Johns Hopkins University internal funds and in part by Division of Plastic Surgery and Neurosurgery at Hospital for Sick Children. **Disclosures**: Russel H. Taylor is a paid consultant to Galen Robotics and has an equity interest in that company. These arrangements have been reviewed and approved by JHU in accordance with its conflict of interest policy.

[1] Authors with Laboratory for Computational Sensing and Robotics, Johns Hopkins University, Baltimore, Maryland, USA

[2] Authors with the Center for Image Guided Innovation and Therapeutic Intervention lab at Hospital for Sick Children, Toronto, Ontario, Canada

presents the proposed approach. Section IV presents the validation experiment in simulation and the run time experiment. Section V presents the skull cutting experiments. Section VI concludes the paper.

II. BACKGROUND

A. *Constraint Optimization*

Constraint optimization approaches are well-established methods to implement virtual fixtures. In this work, we formulate robot kinematic motion control as a quadratic optimization problem with linear constraints [13]. The objective function solves for the desired motion as

$$\begin{aligned} &\arg\min_{\Delta \boldsymbol{q}} \|\Delta \boldsymbol{x} - \Delta \boldsymbol{x_d}\|_2, \\ &\text{subject to } \boldsymbol{A}\Delta \boldsymbol{x} \geq \boldsymbol{b} \\ &\qquad \Delta \boldsymbol{x} = \boldsymbol{J}\Delta \boldsymbol{q} \end{aligned} \tag{1}$$

where $\Delta \boldsymbol{x}$ and $\Delta \boldsymbol{x_d}$ are the computed and desired incremental Cartesian positions, and $\Delta \boldsymbol{q}$ is the incremental joint position. $\boldsymbol{J}$ is the Jacobian matrix relating the joint space to the Cartesian space. $\boldsymbol{A}$ and $\boldsymbol{b}$ are matrix and vectors necessary to describe the linear constraints.

In a telemanipulated control environment, $\Delta \boldsymbol{x_d}$ in the objective function is computed from the motion of the master manipulator. Additional objectives may be added to express additional desired behaviors.

Inequality constraints can be used to impose motion constraints. For example, a virtual forbidden wall for tool tip $\boldsymbol{x}$ can be defined by a hyper-plane with normal $\boldsymbol{n}$ and point $\boldsymbol{p}$, i.e. tool tip can only move on the positive side of the hyperplane as shown in Fig. 1. This can be achieved by forcing the signed distance d_t from the tool tip to plane at any time t to be positive, i.e.,

$$\begin{aligned} d_{t-1} &= \boldsymbol{n}^T(\boldsymbol{x} - \boldsymbol{p}) \\ \Delta d &= \boldsymbol{n}^T \Delta \boldsymbol{x} \\ d_t &= d_{t-1} + \Delta d \geq 0 \\ \boldsymbol{n}^T \Delta \boldsymbol{x} &\geq -\boldsymbol{n}^T(\boldsymbol{x} - \boldsymbol{p}) \end{aligned} \tag{2}$$

where Δd is the change of the signed distance. The constrained motion can be realized by setting $\boldsymbol{A} = \boldsymbol{n}^T$ and $\boldsymbol{b} = -\boldsymbol{n}^T(\boldsymbol{x} - \boldsymbol{p})$. The constrained least-squares problem may then be solved to produce the desired motion. Additional terms may be added to further constrain the tool motion.

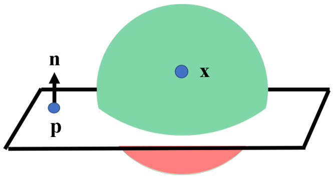


Fig. 1: Illustration of plane constraint. The radius of the sphere is the maximum motion capable of the robot in one control iteration. Green zones denote the allowable region and red zones denote the forbidden region.

B. *Polygon Mesh*

In this work, polygon meshes consisting of triangles are used to represent anatomical surfaces. A locally **concave surface** (Fig. 2a) produces a **convex set of linear constraints**. It is safe to include all triangles as plane constraints (Fig. 2b). A locally **convex surface** (Fig. 2c) produces a **non-convex set of linear constraints**. Naively adding all triangles as plane constraints will rule out many allowable regions (Fig. 2d). This necessitates an approach to dynamically activate and deactivate the constraints based on the local convexity and concavity of the anatomical surfaces.

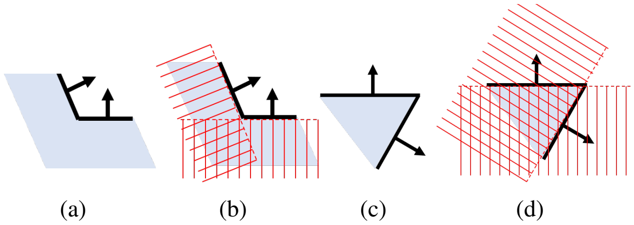


Fig. 2: 2D illustration of (a-b) a locally **concave surface**, and (c-d) a locally **convex surface**. Red crosshatched zones denote forbidden region. Note in (d) many allowable regions are ruled out. Arrows denote the face normals. Light blue zones denote the patient's anatomy.

III. VIRTUAL FIXTURE FOR POLYGON MESH

A. *Polygon Mesh Constraint Algorithm*

Instead of considering the whole mesh object, a *motion sphere* is built around the current position with the radius defined by the maximum motion capable of the robot in one control iteration (shown as the sphere in Fig. 1). The triangles intersected by the motion sphere are the only necessary ones to be considered in the current iteration. To enable efficient geometric-search intersection, the anatomical mesh is stored as a Principle Direction Tree (PD-Tree) [14]. The PD-Tree is similar to the KD-Tree, but with nodes split along the maximum distributive direction of the data. This provides an additional boost in search efficiency especially when triangles are not uniformly distributed. During the PD-Tree construction, the adjacency information of the triangles is stored. Using the PD-Tree of the anatomy and the motion sphere of the tool position, the corresponding closest points $\mathcal{CP}$ and face normals $\mathcal{N}$ of each intersected triangle are returned.

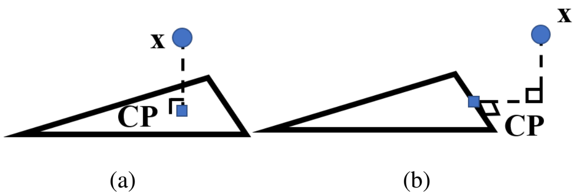


Fig. 3: $\mathcal{CP}$ location, (a) *in-triangle*, (b) *on-edge*. The blue rectangle denotes the $\mathcal{CP}$. The right-angle sign denotes perpendicularity.

The closest point location on a triangle is found using [15], which can land in two places - *in-triangle* or *on-edge* (Fig. 3). The closest points on each triangle are necessary to determine the local convexity/concavity and approximate local geometries using plane constraints. Based on the $\mathcal{CP}$ location, there are only three cases for a given triangle $\mathcal{T}_i$ to be added to the list of active constraints $\mathcal{L}$. The algorithm is shown in Algorithm 1.

Algorithm 1: Polygon Mesh Constraint

```
Input: PD-Tree, Current Position $x$
Result: List of Active Plane Constraints $\mathcal{L}$
Find intersected triangles $\mathcal{T}$, corresponding closest
  points $\mathcal{CP}$ and face normals $\mathcal{N}$ ;
for triangle $\mathcal{T}_i \in \mathcal{T}$ do
    if $\mathcal{CP}_i$ in-triangle & $\mathcal{N}_i^T(x - \mathcal{CP}_i) \geq 0$ then
        add $\{\mathcal{N}_i, \mathcal{CP}_i\}$ to $\mathcal{L}$ ;
    else if $\mathcal{CP}_i$ on-edge then
        Find adjacent triangle(s) $\mathcal{T}_{i,a}$ ;
        if $\mathcal{CP}_i == \mathcal{CP}_{i,a}$ & locally convex then
            add $\{x - \mathcal{CP}_i, \mathcal{CP}_i\}$ to $\mathcal{L}$ ;
        else if $\mathcal{N}_i^T(x - \mathcal{CP}_i) \geq 0$ & locally concave
          then
            add $\{\mathcal{N}_i, \mathcal{CP}_i\}$ to $\mathcal{L}$ ;
end
```

Condition 1 ($\mathcal{CP}_i$ *in-triangle* & $\mathcal{N}_i^T(x - \mathcal{CP}_i) \geq 0$): If $\mathcal{CP}_i$ is in-triangle and the current position x is on the positive side of the face normal ($\mathcal{N}_i^T(x - \mathcal{CP}_i) \geq 0$), then the triangle is added as a constraint. The face normal $\mathcal{N}_i$ and closest point $\mathcal{CP}_i$ will be used in Equation 2.

If the closest point is on-edge, then the information of the adjacent triangle(s) $\mathcal{T}_{i,a}$ is retrieved, where the subscript a denotes "adjacent". For an edge, there is one adjacent triangle. For a vertex, although in theory there could be infinite triangles intersecting the same vertex, it is sufficient to consider the two adjacent triangles whose edges intersect at the given vertex to determine the local geometry of the current triangle. When one of the adjacent triangles meets the following conditions, an active plane constraint is added. The local convexity can be found using

$$convexity = \begin{cases} 1, & \text{if } \mathcal{N}_{i,a}^T v > 0 \\ 0, & \text{if } \mathcal{N}_{i,a}^T v < 0 \end{cases} \tag{3}$$

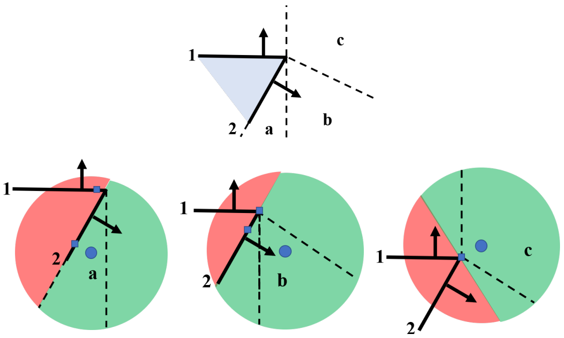
where v is the one of the non-shared edges.

Condition 2 ($\mathcal{CP}_i$ *on-edge* & $\mathcal{CP}_i == \mathcal{CP}_{i,a}$ & *locally convex*): If the closest point of the adjacent triangle(s) $\mathcal{CP}_{i,a}$ is equal to the closest point of the current triangle $\mathcal{CP}_i$, then the two points are on the same shared edge. When this occurs for the locally convex surface (Fig. 2c), the edge can be approximated by a plane with the normal equal to the vector $x - \mathcal{CP}_i$, which points from the closest point $\mathcal{CP}_i$ to the current position x. The plane also passes the closest point $\mathcal{CP}_i$.

Condition 3 ($\mathcal{CP}_i$ *on-edge* & $\mathcal{N}_i^T(x - \mathcal{CP}_i)$ & *locally concave*): If none of the above cases apply, and the current position x is on the positive side of the face normal of the current triangle ($\mathcal{N}_i^T(x - \mathcal{CP}_i) \geq 0$) and the local surface is concave (Fig. 2a), the face normal $\mathcal{N}_i$ and closest point $\mathcal{CP}_i$ are used to define the plane constraint.

B. Completeness of Algorithm

In this section, we will enumerate the different local cases that can arise and explain how the algorithm handles each case. Note that the figures use a 2D representation, where line segments denote triangles, arrows denote face normals, blue squares denote closest points, and blue circles denote the current tool tip position. The motion sphere is represented by the circle containing green (allowable), red (forbidden) and gray (undecided) regions.



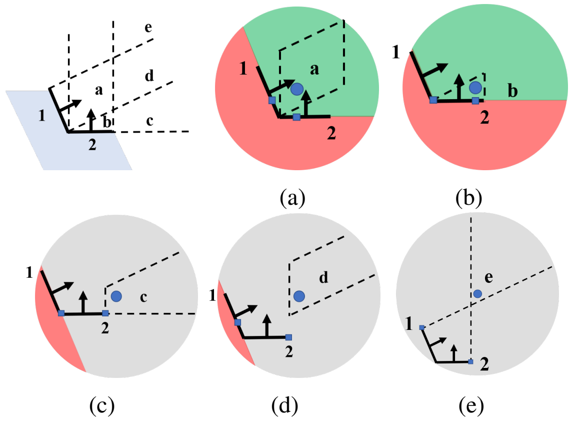
(a) (b) (c)

Fig. 4: Enumeration of a **locally convex** shape.

A locally **convex shape** formed by two triangles is illustrated in Fig. 4. There are three cases to be considered, and the rest will be the same due to symmetry. In case 4(a), $\mathcal{T}_2$ is activated (Condition 1); $\mathcal{T}_1$ fails to activate since $\mathcal{CP}$ is in-triangle while the current position is on the negative side of the face normal. In case 4(b), $\mathcal{T}_2$ is activated (Condition 1); $\mathcal{T}_1$ fails to activate since $\mathcal{CP}_1$ is on-edge while the adjacent triangle's closest point $\mathcal{CP}_2$ is not on the same edge, nor is the current position on the positive side of both normals. In case 4(c), a plane with normal equal to the vector pointing from $\mathcal{CP}$ to the current position is activated (Condition 2), since $\mathcal{CP}$ of both triangles land on the same edge and form a locally convex shape.

The case where two triangles forming a locally **concave shape** is illustrated in Fig. 5. In case 5(a), both triangles are activated (Condition 1). In case 5(b), $\mathcal{T}_2$ is activated (Condition 1) and $\mathcal{T}_1$ is activated due to local geometric concavity (Condition 3). In case 5(c), $\mathcal{T}_1$ is activated due to local geometric concavity (Condition 3); the activation of $\mathcal{T}_2$ is undecided due to the missing adjacency information of the other edge. If adjacency information is given, it will be one of the previously discussed cases. In case 5(d), $\mathcal{T}_1$ is activated (Condition 1); the activation of $\mathcal{T}_2$ is undecided due to the missing adjacency information. In case 5(e), both the activation of both triangles is undecided due to the missing adjacency information.

For both locally concave and convex shapes, the above cases cover all possible situations independent of the angle



(a) (b)

(c) (d) (e)

Fig. 5: Enumeration of a **locally concave** shape.

between the two triangles. Table I summarizes the result.

TABLE I: Activated constraints of the enumerated local cases. Brackets contain the normal and point used in Equation 2.

		Condition 1	Condition 2	Condition 3
Concave Fig. 5	a	$\{\mathcal{N}_2, \mathcal{CP}_2\}$		
	b	$\{\mathcal{N}_2, \mathcal{CP}_2\}$		
	c		$\{$x-$\mathcal{CP}_2, \mathcal{CP}_2\}$	
Convex Fig. 4	a	$\{\mathcal{N}_1, \mathcal{CP}_1\}$, $\{\mathcal{N}_2, \mathcal{CP}_2\}$		
	b	$\{\mathcal{N}_2, \mathcal{CP}_2\}$		$\{\mathcal{N}_1, \mathcal{CP}_1\}$
	c			$\{\mathcal{N}_1, \mathcal{CP}_1\}$
	d	$\{\mathcal{N}_1, \mathcal{CP}_1\}$		
	e			

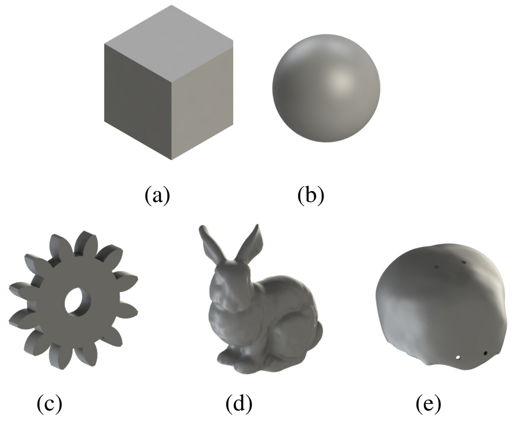
IV. SIMULATION EXPERIMENT

A. Simulation Validation

The proposed algorithm is evaluated in simulation with various shapes. In the experiment, there are 11 shapes tested, including simple geometries, such as cube and sphere, and complex geometries such as gear, Stanford bunny, and pediatric skull. Some examples of the tested mesh objects are shown in Fig. 6. The pediatric skull is generated from a CT scan, and later used to 3D print the skull phantoms for the cutting experiments. The proposed algorithm is successful at providing the virtual forbidden area for all the mesh objects. The full result can be found in the video supplementary material.

B. Run Time Experiment

A run time experiment is conducted in simulation to evaluate the performance of the proposed approach. The period of the run time loop, including the time of finding intersected triangles, detecting activation of triangles and solving for optimization solution is recorded. The time is then converted to frequencies with respect to the number of triangles and vertices in the mesh objects. The algorithm is implemented in C++ using the CISST library developed at Johns Hopkins University. The system used is Ubuntu 18.04 with Intel Core i7-8700. The object tested is a high-resolution Stanford bunny as shown in Fig. 6d. As the number of triangles increases from 51,978 to 989,376 and the number of vertices increases from 25,991 to 493,460, the run time loop frequency decreases from 7.73 kHz to 180 Hz. The performance of the algorithm exceeds the minimal requirement for a high-level teleoperated control loop of 1 kHz when the mesh object contains less than approximately 200,000 triangles. The full result is plotted in Fig. 7.


(a) (b)

(c) (d) (e)

Fig. 6: Examples of the objects tested in simulation. (a) cube, (b) sphere, (c) gear, (d) Stanford bunny, (e) pediatric skull.

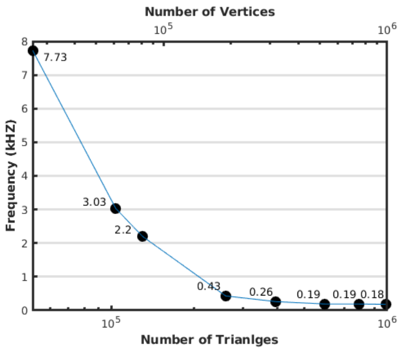


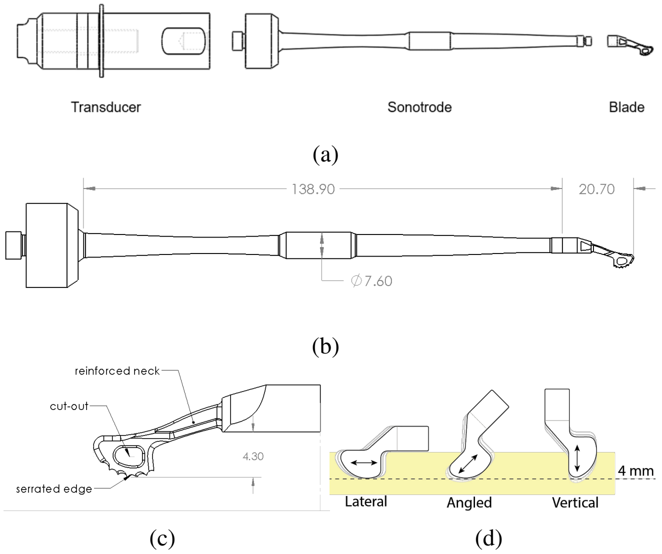
Fig. 7: Run time result of Frequency (kHz) versus Number of Vertices (top, log scale) and Number of Triangles (bottom, log scale). The frequency is labeled beside each data point.

V. SKULL CUTTING EXPERIMENT

The previously discussed VF algorithm is integrated into the da Vinci Research Kit (dVRK) [16], and two experiments are conducted to validate the effectiveness of the VF assistance. The experiment setup can be found in Section V-A. During the experiments, the performance using two control modes are compared - teleoperation and VF assisted teleoperation. In the path tracing experiment (Section V-B), a pre-cut skull phantom is traced along a planned path. The smoothness and penetration depth of the tool motion are compared. In the cutting experiment (Section V-C), four skull phantoms are cut using a novel piezoelectric cutter. The cutting accuracy, penetration depth and completion time of the tool motion are compared.

A. Experiment Setup

The piezoelectric cutter used in this experiment is an improved version from [17]. The cutter operates in the 20-30 kHz frequency range, with displacements of 15-350 microns. It has three components, the transducer, the sonotrode, and the blade. The blade is designed such that it can cut along various angles for up to 4 mm depth. The design details of the piezoelectric cutter can be found in Fig. 8.



(a)

(b)

(c) (d)

Fig. 8: The piezoelectric cutter used in the experiments, (a) components, (b) dimension, (c) blade design, (d) illustration of the blade cutting angles.

The skull phantoms used in the experiments are 3D printed with VisiJet PXL + ColorBond (3D Systems, US). The material has a flexural modulus of 7.16 GPa, which is similar to human skulls of 7.46 GPa [18]. The skull has a uniform thickness of 2 mm. A 30 mm planned cutting path is inked on the skull on the Sagittal plane. There are six fiducial markers placed on the skull for registration.

The cutting tool is mounted on the dVRK Patient Side Manipulator (PSM). The skull phantom is placed directly underneath the PSM, centered at the RCM point. A pair of Blackfly S cameras (FLIR Systems, Canada) is used for stereo vision. The experiment setup is shown in Fig. 9.

In the VF assisted mode, haptic feedback is provided on the Master Tool Manipulator (MTM). The force exerted by the MTM on the surgeon's hand is proportional to the positional difference between the constrained PSM position and proxy position (i.e. the position without any VF constraint). The PSM is constrained such that the cutter can deviate from the planned path up to 1 mm perpendicular to the Sagittal plane on each side.

B. Path Tracing

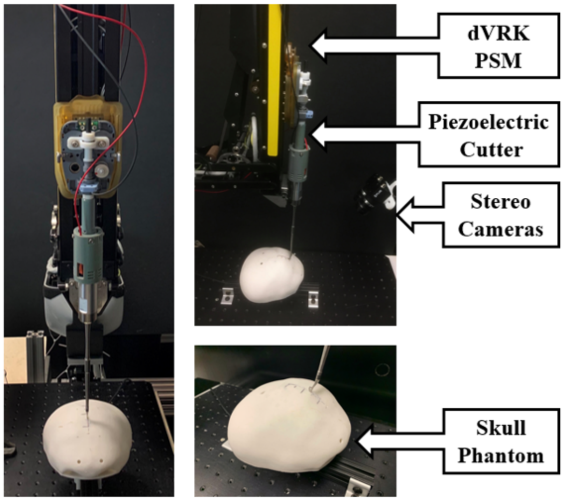


Fig. 9: Experiment setup for skull cutting experiments.

In this experiment, the exterior of a pre-cut skull is traced using the piezoelectric cutter along a planned path. The goal is to compare the penetration depth control between relying on only stereo vision and relying on stereo vision and VF assistance. The path is pre-cut to avoid excessive contact force between the cutter and the skull. The experiment is repeated five times for each mode. The overlays of the trajectories on the cross-section of the skull are shown in Fig. 10. In the VF assisted mode, the path can be traced by constantly pushing the PSM down since the VF will stop the cutter at the exterior surface of the skull. Even though the proxy trajectory has some deviation, the VF effectively constrains the motion. The trajectory of the tool without VF assistance has more variation and deviation due to the challenges in depth perception. The VF assisted trajectory is visually smoother and aligns with the exterior of the skull more accurately.

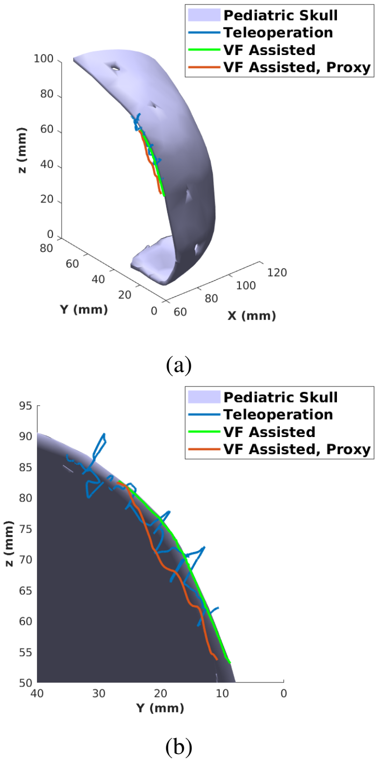
During surgery, it is desirable to have smooth trajectories without sudden adjustments. To qualitatively evaluate the motion smoothness of the trajectories, the spectral arc length (SAL) [19] is computed:

$$
\begin{aligned}
SAL &= -\int_0^{\omega_c} \sqrt{(\frac{1}{\omega_c})^2 + (\frac{d\hat{V}(\omega)}{d\omega})^2} d\omega \\
d\hat{V}(\omega) &\triangleq \frac{V(\omega)}{V(0)}
\end{aligned}
\tag{4}
$$

where ω_c is the cut-off frequency of 20 Hz, and V is the velocity of the cutter. The intuition of SAL is that in the Fourier domain, the complexity of the velocity profile in the frequency domain will be retained and amplified. Due to the negative sign in front of the integral, a larger SAL will result in a smaller arc length, thus indicating a smoother motion. The SAL scores are -20.98 ± 0.79 and -10.02 ± 3.7 respectively for the trajectories of teleoperation and VF assisted teleoperation modes. The improvement of the trajectory smoothness with VF assistance is statistically significant (p-value=1.9E-4).

C. Cutting

In this experiment, four skulls are cut along the planned path using the piezoelectric cutter. The interior of the skull



(a)

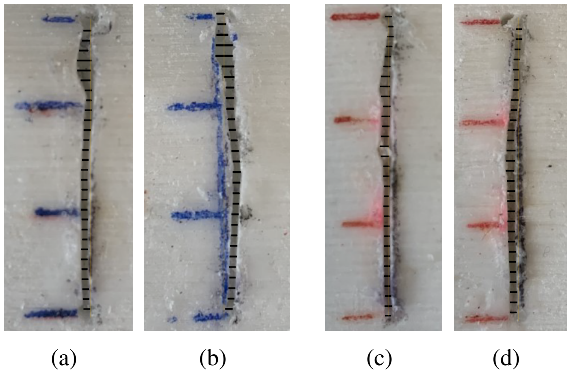
(b)

Fig. 10: Overlays of sample trajectories from the path tracing experiment on the pre-cut pediatric skull.

is covered by a layer of wax to evaluate the penetration depth after cutting. Two skulls are cut with VF assistance and two without. The following metrics are used to compare the performance of teleoperation and VF assisted teleoperation:

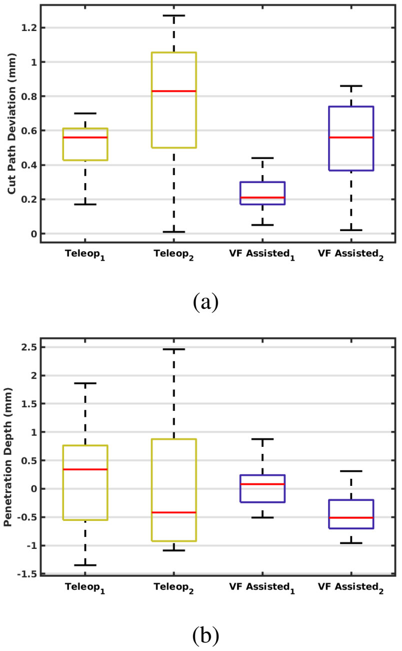
- **Cutting Path Deviation (CPD)**: the deviation of the cut path from the planned path. The minimal deviation is desired for the most accurate cutting path.
- **Penetration Depth (PD)**: the penetration depth from the interior skull surface. Minimal penetration is desired to avoid damages to the dura mater.
- **Duration**: the total time it takes to complete the cutting procedure.

The CPD is measured as the perpendicular distance from the furthest points of the cut path to the planned path at 1 mm intervals, totaling 30 measurements per cutting trial. To measure the CPD, the cutting path is imaged by a high-resolution camera. The pixel-to-mm conversion is obtained by dividing the pixel distance between the inked line segments by 10 mm, which is the actual distance. The CPD is measured in pixel first and converted to mm using the pixel-to-mm conversion. The result can be found in Fig. 12a and the cut paths of the four trials can be found in Fig. 11. The maximum CPD of the teleoperated mode is 1.27mm, while the maximum CPD of VF assisted mode is only 0.86mm. The mean±std CPD are 0.63 ± 0.30mm for the teleoperation mode and 0.37 ± 0.24mm for the VF assisted mode. The improvement in cutting path accuracy with VF assistance has been found statistically significant. (p-value=2.3E-6).



(a) (b) (c) (d)

Fig. 11: Cut path using (a-b) teleoperation, (c-e) VF assisted teleoperation. Marks are inked every 10 mm along the path. The distance between the black horizontal line segments (labeled during CPD analysis) inside the path is 1 mm.



(a)

(b)

Fig. 12: Experiment results for the four trials of skull cutting. (a) Cutting path deviation. (b) Penetration depth.

The PD is measured as the distance between the exterior surface of the skull and the top surface of the wax minus the 2 mm skull thickness at 1 mm intervals, totaling 30 measurements per trial. The apparatus used for PD measurement is shown in Fig. 13. An LSB200 miniature load cell with a resolution of 1 mN (FUTEK Advanced Sensor Technology, US) is attached on an XYZ linear stage with 1 µm accuracy (Newport Corporation, US). The skull is mounted on a vertical column. A hypodermic 304 stainless steel tubing of 0.016” outer diameter is attached to the load cell to sense contacts with surfaces. Contact positions at the exterior surface of the skull and the wax are recorded when the contact force exceeds 10 mN. The PD result is shown in Fig. 12b, where a positive PD value indicates an over-cut and a negative PD value indicates an under-cut. In the teleoperation mode, the maximum over-cut is 2.46 mm and the maximum under-cut is 1.35 mm. In the VF assisted mode, the maximum over-cut is 0.87 mm and the minimum under-cut is 0.96 mm. The mean±std absolute PD are 0.83 ± 0.51mm for the teleoperation mode and 0.39 ± 0.26mm for the VF assisted mode. The constrained motion enforced by the VF improves the PD result statistically significantly (p-value=7.9E-3). However, there still exist errors due to the registration offset and mechanical deflection in the system.

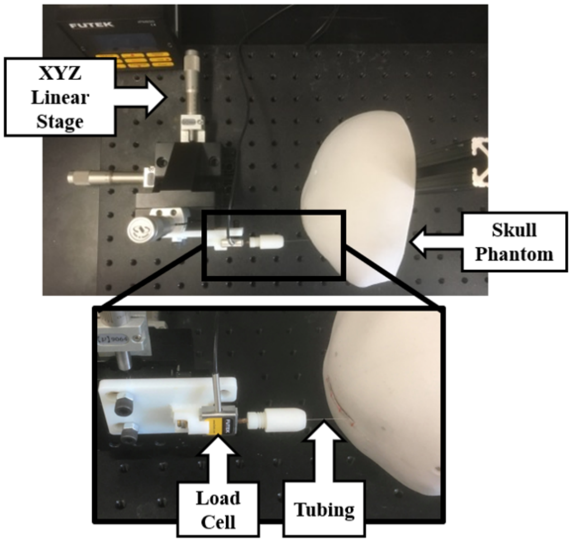


Fig. 13: Penetration depth measurement setup.

The total time to finish the cutting task for the two teleoperation trials are 232s and 237s respectively. For the two VF assisted trials, the total time to finish are 212s and 196s respectively. No statistically significant difference is found between the teleoperated (234 ± 3.8s) and VF assisted (204 ± 11s) modes, with p-value=0.069. Additional trials are needed to further verify the difference of duration between the two modes.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a method for the direct generation of avoidance virtual fixtures from anatomical mesh models. The proposed approach can provide a virtual forbidden area for patient anatomies during robot-assisted surgeries. The method does not assume the convexity/concavity of the anatomy near the tool. The method has been validated in simulation, and the run time experiment is presented. An initial piezoelectric cutting experiment on skull phantoms in a telesurgical environment using the dVRK is reported. Statistically significant improvements have been found in cutting path accuracy and penetration depth control compared to conventional teleoperation. In the future, we plan to demonstrate our method in the presence of anatomical movement and deformation. We also plan to conduct systematic user studies with multiple clinician users.

REFERENCES

[1] A. G. Hunter and N. L. Rudd, “Craniosynostosis. i. sagittal synostosis; its genetics and associated clinical findings in 214 patients who lacked involvement of the coronal suture (s),” *Teratology*, vol. 14, no. 2, pp. 185–193, 1976.

[2] S. A. Tatum and W. D. Losquadro, “Advances in craniofacial surgery,” *Archives of facial plastic surgery*, vol. 10, no. 6, pp. 376–380, 2008.

[3] S. A. Bowyer, B. L. Davies, and F. R. y Baena, “Active constraints/virtual fixtures: A survey,” *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2013.

[4] H. Ueda, R. Suzuki, A. Nakazawa, Y. Kurose, M. M. Marinho, N. Shono, H. Nakatomi, N. Saito, E. Watanabe, A. Morita *et al.*, “Toward autonomous collision avoidance for robotic neurosurgery in deep and narrow spaces in the brain,” *Procedia CIRP*, vol. 65, pp. 110–114, 2017.

[5] Z. Li, M. Shahbazi, N. Patel, E. O’Sullivan, H. Zhang, K. Vyas, P. Chalasani, P. L. Gehlbach, I. Iordachita, G.-Z. Yang *et al.*, “A novel semi-autonomous control framework for retina confocal endomicroscopy scanning,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7083–7090.

[6] R. Prada and S. Payandeh, “A study on design and analysis of virtual fixtures for cutting in training environments,” in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. IEEE, 2005, pp. 375–380.

[7] S. Park, R. D. Howe, and D. F. Torchiana, “Virtual fixtures for robotic cardiac surgery,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2001, pp. 1419–1420.

[8] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, “Dynamic active constraints for surgical robots using vector-field inequalities,” *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, 2019.

[9] J. Ren, R. V. Patel, K. A. McIsaac, G. Guiraudon, and T. M. Peters, “Dynamic 3-d virtual fixtures for minimally invasive beating heart procedures,” *IEEE transactions on medical imaging*, vol. 27, no. 8, pp. 1061–1070, 2008.

[10] M. Li, M. Ishii, and R. H. Taylor, “Spatial motion constraints using virtual fixtures generated by anatomy,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 4–19, 2007.

[11] K. Santosh, S. Antani, D. S. Guru, and N. Dey, *Medical Imaging: Artificial Intelligence, Image Recognition, and Machine Learning Techniques*. CRC Press, 2019.

[12] T. Furuhata, I. Song, H. Zhang, Y. Rabin, and K. Shimada, “Interactive prostate shape reconstruction from 3d trus images,” *Journal of computational design and engineering*, vol. 1, no. 4, pp. 272–288, 2014.

[13] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, “Constrained cartesian motion control for teleoperated surgical robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 1996.

[14] J. Williams, R. Taylor, and L. Wolff, “Augmented kd techniques for accelerated registration and distance measurement of surfaces,” *Computer Aided Surgery: Computer-Integrated Surgery of the Head and Spine*, pp. 1–21, 1997.

[15] M. W. Jones, “3d distance from a point to a triangle,” *Department of Computer Science, University of Wales Swansea Technical Report CSR-5*, 1995.

[16] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, “An open-source research kit for the da vinci® surgical system,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[17] A. Gordon, T. Looi, J. Drake, and C. R. Forrest, “An ultrasonic bone cutting tool for the da vinci research kit,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6645–6650.

[18] J. A. Motherway, P. Verschueren, G. Van der Perre, J. Vander Sloten, and M. D. Gilchrist, “The mechanical properties of cranial bone: the effect of loading rate and cranial sampling position,” *Journal of biomechanics*, vol. 42, no. 13, pp. 2129–2135, 2009.

[19] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, “A robust and sensitive metric for quantifying movement smoothness,” *IEEE transactions on biomedical engineering*, vol. 59, no. 8, pp. 2126–2136, 2011.