IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Parallel Haptic Rendering for Orthopedic Surgery Simulators

Mohammadreza Faieghi[1], S. Farokh Atashzar[2], O. Remus Tutunea-Fatan[3], and Roy Eagleson[4]

***Abstract*— This study introduces a haptic rendering algorithm for simulating surgical bone machining operations. The proposed algorithm is a new variant of the voxmap point-shell method, where the bone and surgical tool geometries are represented by voxels and points, respectively. The algorithm encompasses computationally efficient methods in a data-parallel framework to rapidly query intersecting voxel-point pairs, remove intersected bone voxels to replicate bone removal and compute elemental cutting forces. A new force model is adopted from the composite machining literature to calculate the elemental forces with higher accuracy. The integration of the algorithm with graphics rendering for visuo-haptic simulations is also outlined. The algorithm is benchmarked against state-of-the-art methods and is validated against prior experimental data collected during bone drilling and glenoid reaming trials. The results indicate improvements in computational efficiency and the force/torque prediction accuracy compared to the existing methods, which can be ultimately translated into higher realism in simulating orthopedic procedures.**

I. Introduction

Haptics-enabled surgery simulation is an emerging technology in surgeon training to train surgical motor skills in mixed reality environments. This technology reduces the need for training on patients, cadavers, and animals, and enables objective evaluation of the skill acquisition process in a systematic manner [1]. Although this technology has attracted a great deal of interest in several surgical disciplines, its use in orthopedics continues to remain minimal, primarily due to the technical challenges in rendering haptic interactions with ossesous tissues, which is quintessential for simulating bone machining operations that are an integral part of various orthopedic surgeries. One important challenge in this regard is the need for haptic rendering algorithms that are computationally efficient for real-time implementation, and at the same time, present accurate tool-tissue interaction modeling for realistic simulation experience [2].

The inherent material removal associated with machining operations creates the need for workpiece geometry reconstruction at each computing frame. However, since real-time geometry reconstruction is computationally tedious, customized haptic rendering algorithms are required to maintain the minimum 1 kHz update rate, which is needed for guaranteeing stable haptic rendering, in particular for interacting with rigid tissues (such as bone). The well-know voxmap-point shell (VPS) method that uses discretized geometries for simplified haptic computations [3] has been used in the literature as a potential solution to address some of the mentioned challenges [4]–[8]. In this method, the surgical tool is generally modeled using a point cloud (point-shell) whereas bone is represented by a voxelized model (voxmap) that is often established on a 3D grid. Therefore, material removal can be easily modeled by switching the occupancy status of voxels in the grid, and force and torque can be computed as a function of the voxels removed. Although this intuitive approach has shown good potential for matching the physics-based bone machining models [8]–[10], the corresponding performance is heavily dependent on the size of sampling resolutions imposed on objects. When elements are too coarse, the haptic feedback is prone to abrupt changes in force/torque magnitudes and in turn, stability and realism of simulation are degraded drastically. Finer-sized elements could potentially correct this issue; however, it leads to significant increase in the computational load. Given the stringent 1 kHz timing constraint, the maximum attainable resolution is thus severely constrained.

Previous studies have used hierarchical data structures such as bounding volume hierarchy [5] and octree [6], [11] to accelerate haptic rendering via tree-based computations. Nonetheless, the tree traverse time and especially the memory consumption in hierarchical structures are highly sensitive to the sampling resolution and this, in turn, limits the improvements that can be achieved by means of these methods. In the general context of haptic rendering, precomputed distance fields have been proved effective for efficient VPS implementation [12]. However, their applications in bone machining are associated with an added overhead for recomputing distance fields in the runtime. This could become problematic, particularly in case of severe intersections in which a large portion of the distance field needs to be continuously updated. In [13] virtual objects were presented in polygonal meshes. In this case, the intersection region of contacting meshes was discretized by means of a graphics processing unit (GPU)-based voxelization routine and the resulting voxels were further used in haptic computations. Unfortunately, voxelization runtime relies largely on the number of polygons and their topology [14]. Therefore, the performance of this algorithm remains rather limited in the case of complex tools and workpiece geometries.

Another limitation in bone machining simulation is the difficulty to accurately predict forces and torques gener-

This work was supported by Natural Sciences and Engineering Research Council (NSERC) of Canada and Canadian Institutes of Health Research (CIHR) under the Collaborative Health Research Projects program.

[1]M. Faieghi is with the Department of Aerospace Engineering, Ryerson University, Toronto, Canada reza.faieghi@ryerson.ca

[2]S. F. Atashzar is with the Tandon School of Engineering, New York University, USA. He is also with NYU WIRELESS f.atashzar@nyu.edu

[3]O. R. Tutunea-Fatan is with the Department of Mechanical and Materials Engineering, Western University, London, Canada email: rtutunea@eng.uwo.ca

[4]R. Eagleson is with the Department of Electrical and Computer Engineering, Western University, London, Canada eagleson@uwo.ca

ated during bone removal. This stems from the geometric complexity and material anisotropy of the bone [15] that essentially makes the development of closed analytical solutions virtually impossible. For this reason, empirical and mechanistic approaches tend to be favoured compared to their analytical counterparts. In this context, regression-based models building on data gathered during bone machining trials have been used to predict axial thrust and torque as a function of machining parameters e.g. feedrate, spindle speed, cutting depth, etc. [16]. However, these data-driven models do not explicitly account for variable tool geometries and are specific for the utilized tool. Thereby the corresponding accuracy can be significantly degraded for simulating a drilling operation with a tool geometry different than the one used for data collection.

To overcome the above-mentioned challenge, mechanistic models have been developed using cutting mechanics. One of the most common approaches in this regard relies on the principle of oblique cutting in which derived equations are to be calibrated against empirical data in order to make them account for bone anisotropy. The theory of oblique cutting was originally developed for metal cutting [17]. After some early attempts to extend to composite materials, notably by Chandrasekharan et al.'s [18], it was also applied to bone removal operations [9], [10]. Past studies have outlined that the use of this method for bone cutting leads to a rather limited accuracy for torque predictions [9], [10]. In the context of haptic rendering, this model is often simplified, primarily to reduce the computational load, which will negatively impact the prediction accuracy, and hence deteriorates the realism of the simulation [6], [11].

Newer studies focused on the machining of composites have improved the Chandrasekharan et al.'s method in terms of force and torque prediction accuracy [19], [20]. However, the potential benefit of such models for bone machining has not been investigated so far despite its high potentials. This is one of the focuses of the present study.

This paper is motivated by a clear need for the development of a new modeling technique capable of simulating the removal of bone with high resolution, and precision and fast update rate demanded by haptic algorithms for realistic haptics-enabled orthopedic surgical simulators. To address this, the present study proposes a new computational framework to allow a fast and accurate haptics rendering to be used during simulation of various bone machining operations such as drilling, reaming, burring, sawing. The bases of the bone drilling model used in the present work build upon the technique proposed in [19] for machining of composite materials. Also, the collision detection scheme presented in this study is an extension of our earlier collision detection algorithm [21]. The fast GPU-based computation of our recent work [21] and its low sensitivity to voxmap resolutions, makes it amenable to extend attainable resolutions involving the VPS method. Although [21] focuses only on collision detection, the present study will includes the whole haptics rendering loop in an attempt to effectively integrate the accurate collision detection with generalizable bone machining models in order to perform fast and accurate force/torque computations for haptic rendering.

In addition to the above, the coupling between haptics and graphics rendering algorithms is crucial for the development of accurate visual replication of material removal. As such, the present study outlines how to integrate the haptic rendering computations with graphic rendering methods and enable realistic visuo-haptic simulations.

The proposed technique will be evaluated in two case studies: (1) bone drilling, a common, yet complex bone machining operation, and (2) glenoid reaming, a challenging bone removal operation associated with shoulder joint replacement surgery [22], which was not simulated in the past for haptics rendering due to the complex geometry of the surgical tools. To verify the effectiveness of the proposed algorithm, the performance will be compared with the method presented in [11]. One important feature of the present study is presenting an in-depth analysis of the prediction accuracy of bone machining models used in haptic simulations and the effects of sampling resolutions on the performance of haptic/graphic rendering loops.

The primary contribution of this study is represented by the developed haptic rendering algorithm. The proposed solution encompasses several elements of novelty in terms computational framework and force model. These elements are meant to address some of the current challenges related to the simulation of the bone removal, particularly with respect to efficiency and prediction accuracy. All considerations in this study were made in the context of the glenoid resurfacing as encountered in surgical orthopaedics and this represents one other important element of novelty of the current work.

II. Algorithm

Figure 1 depicts the overall schematic of the proposed haptic rendering algorithm. The voxmap and point shell data structures are created during an initialization phase and are stored in GPU memory to be used in the runtime. At each haptic computing frame, the position and orientation of the point shell are updated based on user interactions. A collision detection method queries intersection between point shell and voxmap, and identifies the intersection point-voxel pairs at the event of a collision. At this point, the intersecting voxels are removed from the voxmap structure to account for material removal. Next, elemental cutting forces are computed for each intersecting elements. For this, the elemental forces are calculated in a local coordinate frame, and then transferred to a global tool coordinate frame. Once all elemental forces are expressed in one coordinate frame, they are summed to generate axial thrust and torque which are conveyed to the haptic device. All these operations must be computed within less than 1 ms to guarantee the update rate of 1 kHz required stable haptic rendering. Graphic simulation is often an inseparable part of haptic simulators. Thus, in parallel with the haptic rendering loop, the geometry reconstruction routine takes the updated voxmap structure and builds the resurfaced bone geometry. The geometry is then passed to a graphics rendering engine for visualization

purposes. The maximum allowable computing time for the graphic computations is usually set to 33.33 ms to ensure a minimum of 30 frames per second (FPS) necessary for smooth graphic rendering.

In the following, the details of each component of the proposed haptic rendering algorithm are presented. In order to explain force/torque computations clearly, a twist drill bit is used as a sample tool merely for graphical representations in the paper. Using the proposed approach the mathematical derivations can be generated for different tool geometries (if certain conditions are met). This has been discussed in details and the corresponding calculations are conducted for the glenoid reaming procedure, for the first time, in Section III-B.

A. Voxmap Data Structure

The bone voxmap model is derived from Computed Tomography (CT) images and is stored in a uniform 3D grid to allow accessing voxels via integer grid coordinates. The grid is stored in a 1D contiguous array coupled with an invertible 3D to 1D mapping to map grid cells to the array elements. The value stored in each array element represents the spatial density of the bone that is derived using Hounsfield unit (HU) values in the CT image. Material removal can be replicated by the iterative reduction of the density values until they correspond to a void space. This method of implementing voxmap requires allocating memory for the whole grid (including empty grid cells); however, it still consumes less memory compared to conventional implementations of octrees or other hierarchical data structures. In addition, as it is based on contiguous arrays and integer operations, it guarantees fast access to voxels. The details of processing bone CT images and constructing efficient voxmaps can be found in our recent work [23].

B. Point Shell Data Structure

To construct a point shell, one simple approach is to apply surface voxelization [14] on a triangle mesh model of tool and extract the centroids of output voxels. In this manner, since the extracted points belong to a voxel grid, they can be accessed by integer grid coordinates similar to voxmap. However, due to the particular design of our algorithm, there is no need to allocate space to empty cells for storing the point shell. As such, point shell can be implemented using a smaller array where each element contains the integer grid coordinates of an occupied grid cell. An additional parameter is used per element to identify whether the element is on the cutting edge or not. To further save computing memory, the grid coordinates and the cutting edge identifier of each element are compressed into a single integer using Morton encoding. The details for Morton encoding is given in [24].

C. Collision Detection

The collision detection task is massively parallelized by launching GPU threads per each point shell elements. Prior to the start of threads, the collision between the bounding boxes of tool and bone is checked in order to predict the potential

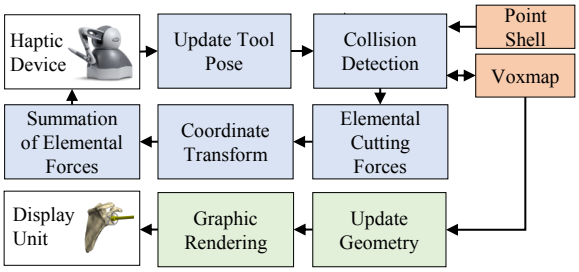

Fig. 1: Main components in the proposed haptic rendering algorithm

intersecting region(s). This allows for an early determination of the non-intersecting condition for points that are outside of the region. As such, the threads corresponding to these elements are culled, resulting in further reduction of the computational load. In each thread, the coordinates of the assigned element are transformed into voxmap coordinates and the element's location in the voxmap array is calculated. If the corresponding element in the array is occupied, then the point shell is in an intersecting condition. This simple sets of computations along with the early bounding box collision check result in one of the fastest collision detection algorithms that is suitable for VPS-based haptic rendering. The details of the algorithm can be found in [21].

D. Elemental Cutting Forces

The elemental cutting forces are generated for intersecting point shell elements that are lying on the cutting edges of the tool. These forces are calculated per elements, in parallel, within the same GPU threads that were initially launched for collision detection.

According to the oblique cutting theory, the two dominant elemental cutting forces are normal and tangential to the rake face, which are denoted by df_n and df_t, respectively, in Fig. 2. The magnitude of these forces can be calculated by

$$df_n = k_c A_u, \quad \text{and} \quad df_t = k_f k_c A_u, \tag{1}$$

where k_c is the specific cutting coefficient, representing the energy consumed per unit volume of material removed and k_f is the friction coefficient corresponding to the friction of chip travel on the rake face. A_u denotes the uncut chip area in a plane perpendicular to the velocity vector V and can be calculated as $A_u = \frac{f\delta}{N_c}\cos\mu$ where N_c is the number of cutting lips on a given tool, δ is the element size, and μ is known as the feed angle defined as the angle between the V and the projection of V on the xy plane (Fig. 3). Denoting f as the feedrate, μ at point p can be calculated as $\mu = \arctan\left(\frac{f}{2\pi r}\right)$, where r is the radial distance of the element centered at p from the tool rotation axis [19].

An important step in modeling the elemental forces is to determine k_c and k_f. These coefficients are usually predicted via regression models based on workpiece material properties and machining parameters. To predict k_c and k_f, Chandrasekharan et al. [18] have suggested a power-law function of A_u, V and normal rake angle γ_n defined as the angle between the rake face at a point p and the normal to both the cutting edge and V (Fig. 3). In the realm of

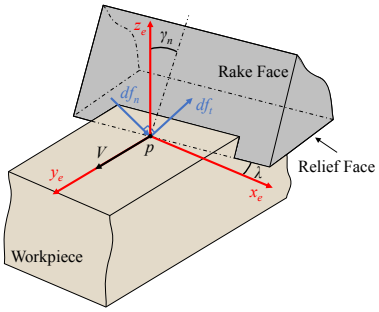

Fig. 2: Elemental cutting forces and elemental coordinates frame in oblique cutting model

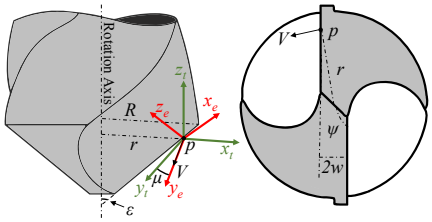

Fig. 3: Twist drill bit geometry and its corresponding elemental and tool coordinate frames

haptic rendering the power-law equation is replaced by a reciprocal function of V, the underlying intention being to reduce computational load [6]. However, experimentation in [20] has revealed that k_c and k_f are primarily functions of γ_n, and should be modeled as a function of that. As such, in this study, k_c and k_f are predicted via regression models of γ_n; however, these models can be easily replaced by any other prediction method as more theoretical knowledge becomes available in this context.

E. Coordinate Transformations

To transform df_n and df_t to haptic device task space, the elemental forces are first represented in a local Cartesian coordinates frame $\mathcal{E} : x_e y_e z_e$ shown in both Figs. 3 and 3. $\mathcal{E}$ is defined such that y_e is aligned with V, x_e is perpendicular to y_e in the plane defined by y_e and cutting edge, and z_e is perpendicular to the $x_e y_e$ plane. Considering the inclination angle λ as the angle between the cutting edge and the normal to V in the plane containing both the velocity and the cutting edge, the normal rake angle γ_n as the angle between the rake face at a point p on the cutting edge and the normal to both V and z_e, measured in a plane perpendicular to the cutting edge at point p, and the chip flow direction η_n which is usually set equal to λ according to the chip flow law of Stabler [17], df_n and df_t can be represented in $\mathcal{E}$ using the following expression [19]

$$\begin{bmatrix} df_{x_e} \\ df_{y_e} \\ df_{z_e} \end{bmatrix} = \mathbf{T} \times \begin{bmatrix} df_t \\ df_n \end{bmatrix}, \tag{2}$$

where

$$\mathbf{T} = \begin{bmatrix} -\cos\gamma_n \sin\lambda & \sin\eta_c \cos\lambda - \cos\eta_c \sin\gamma_n \sin\lambda \\ -\cos\gamma_n \cos\lambda & \cos\eta_c \sin\gamma_n \cos\lambda + \sin\eta_c \sin\lambda \\ -\sin\gamma_n & \cos\eta_c \cos\gamma_n \end{bmatrix}. \tag{3}$$

Next, let us define the global tool coordinates frame $\mathcal{T} : x_t y_t z_t$ such that x_t is along the radial direction, z_t is along the tool rotation axis, and y_t is perpendicular to the $x_t z_t$ plane (Fig. 3). Considering the point angle ε as the angle between cutting edge at point p and z_t, and web angle β as the angle between x_t and the projection of the cutting edge on $x_t y_t$ plane at point p, the rotation matrix from $\mathcal{E}$ to $\mathcal{T}$ can be represented by [19]

$$\mathbf{R} = \begin{bmatrix} \cos\tau & 0 & -\sin\tau \\ \sin\tau\sin\mu & \cos\mu & \cos\tau\sin\mu \\ \sin\tau\cos\mu & -\sin\mu & \cos\tau\cos\mu \end{bmatrix}, \tag{4}$$

where $\tau = \arccos\left(\frac{\sin\varepsilon\cos\beta}{\cos\lambda}\right)$. Therefore, df_n and df_t can be described in $\mathcal{T}$ using

$$\begin{bmatrix} df_{x_t} \\ df_{y_t} \\ df_{z_t} \end{bmatrix} = \mathbf{R} \times \mathbf{T} \times \begin{bmatrix} df_n \\ df_t \end{bmatrix}. \tag{5}$$

The main difference between the above formulation and the previous work in the context of bone machining modeling and haptic simulation is the incorporation of the feed angle μ throughout the formulations. To simplify computations, previous work has commonly neglected the effects of μ for elements on cutting lips with the assumption that on cutting lips r becomes sufficiently large such that $\mu \to 0$. However, this is not necessarily correct for cutting lip elements in the vicinity of chisel edge. Therefore, μ should be taken into account in order to study the distribution of forces along the cutting lips which is critically important in simulation of the early stages of bone drilling stage, before all point shell elements enter voxmap.

It should be highlighted that in derivation of the above expressions, no particular tool geometry is presumed. Therefore, the above formulation can be generalized and applied for any machining tool as long as all the above-mentioned cutting angles are defined accordingly. These angles reflect the geometrical traits of a given tool and should be calculated for all elements distributed along cutting edges. For example, in the case of a twist drill bit, ε is provided by manufacturer. On cutting lips, $\beta = \arcsin\left(\frac{w}{r}\right)$, where w is the half-width of the chisel edge (Fig. 3), $\lambda = \arcsin\left(\sin\beta\cos\mu\sin\varepsilon + \sin\mu\cos\varepsilon\right)$, and γ_n can be calculated using the following expression

$$\begin{aligned} \gamma_n = &\arctan\left(\frac{\tan\theta_l\cos\beta}{\sin\varepsilon - \cos p\tan\theta_l\sin\beta}\right) \\ &-\arctan\left(\frac{\sin\beta\cos\varepsilon - \sin\varepsilon\tan\mu}{\cos\beta}\right), \end{aligned} \tag{6}$$

where θ_l is the local helix angle. Considering θ as the helix angle defined at the outermost point on the cutting lip and ψ as the chisel edge angle (Fig. 3), θ_l can be calculated as $\theta_l = \tan^{-1}\left(\frac{r}{R}\tan\theta\right)$, where $R_c = \frac{w}{\sin\psi}$ [19].

As the chisel edge poses a different geometry compared to the drill bit cutting lips, it has different expressions for the above angles. For many drill bits, the chisel edge is perpendicular to the drill axis in which case $\varepsilon = 90°$, and thus $\lambda, \beta, \tau \equiv 0$. In addition, μ becomes significant because of small values of r. Furthermore, γ_n can be calculated as

$\gamma_n = -\arctan\left(\tan\varepsilon_l \cos\left(90^\circ - \psi\right)\right)$, where ε_l is the point angle on cutting lips in the vicinity of chisel edge.

F. Summation of Elemental Forces

Once the elemental forces are transferred to $\mathcal{T}$, axial thrust F_{z_t} and torque M_{z_t} are calculated as follows

$$F_{z_t} = N_c \sum_{k=1}^{N_e} df_{z_t,k}, \quad \text{and} \quad M_{z_t} = N_c \sum_{k=1}^{N_e} r_k df_{y_t,k}, \quad (7)$$

where the subscript k corresponds to one element and N_e is the total number of elements imposed on a cutting lip. It is worth mentioning that in machining operations the lateral forces of elements often tend to cancel each other, and thus $\sum_{k=1}^{N_e} df_{x_t,k} \simeq 0$.

In the previous GPU-based haptic rendering algorithms [6], [11], [25], the summations in (7) are implemented by storing elemental forces in separate data containers and then, launching a GPU-based reduction algorithm to perform the summations in (7). However, the use of additional data containers for elemental forces and launching separate GPU threads for the reduction algorithm leads to significant memory and running time overheads. To circumvent this, in this study, the summations are performed using GPU atomic operations. These instructions enable add operations across parallel threads but are limited to integer data types. As such, elemental forces are first multiplied by an arbitrary large number N, transformed into integers, and passed to two atomic adders (one for thrust and one for torque). Once atomic operations are completed, the resulting values are divided by N and converted back to floating numbers. The computed axial thrust and torque are ultimately transferred into haptic device task space based on standard 3D homogeneous coordinates manipulations using the pose of the device end-effector.

G. Geometry Reconstruction and Graphic Rendering

Since the proposed haptics rendering algorithm stores voxmap data structure in GPU memory, visual replication of material removal can be handled efficiently by passing the ownership of voxmap data from the haptics rendering loop to the graphics rendering loop, without any need for time-consuming data transfers. Then, various standard graphics rendering techniques can be used to draw the updated geometry of bone in parallel with haptic rendering, and thus provide visuo-haptic simulation experience. To demonstrate this, in this study, a GPU-based implementation of the marching cubes algorithm [26] was used for the bone geometry reconstruction. In this setup, the marching cubes algorithm computes a new isosurface of bone geometry from the updated voxmap array in real-time. The generated isosurface is then passed to a standard OpenGL program to render the bone geometry. The choice of marching cubes and the OpenGL program is merely because of their simplicity, and they can be readily replaced by various surface and volume rendering techniques.

III. Results and Discussion

The performance of the proposed algorithm was evaluated by means of two case studies: (1) bone drilling and (2) glenoid reaming. While drilling represents a common machining operation and hence will be used herein as a standard comparison ground, glenoid reaming represents a challenging orthopaedic procedure that was not analyzed so far.

The haptic rendering algorithm was implemented in OpenCL v1.2, whereas graphic rendering was performed by OpenGL v4.5. The test hardware included an Intel Core i7-8750H CPU, 16 GB of RAM and NVIDIA GeForce RTX 2080 GPU with 8 GB of video memory. The tests were performed using a Touch® haptic device. The force feedback on this device is limited to 3 DOFs with a maximum force of 3.3 N in each direction. As such, all force magnitudes were scaled down to keep them within the operating range of the device. Nonetheless, this has not interfered with the haptic performance evaluation to be detailed in the upcoming sections.

A. Case Study 1: Bone Drilling

The experimental data used to calibrate k_c and k_f are derived from the data depicted in Fig. 7 in [9]. This data correspond to axial thrust and torque as measured in full factorial drilling trials of bovine tibia involving feed rates of $f = 0.01$ and 0.02 $\frac{\text{mm}}{\text{rev}}$ and spindle speeds of $n = 500$ and 3000 rpm, using a drill bit with the following parameters: $R = 1.25$ mm, $\varepsilon = 59^\circ$, $\theta = 20^\circ$ and $\psi = 116^\circ$.

To compare the technique developed in this study with prior methods, the haptic rendering algorithm outlined by Zheng et al. [6] was implemented. This algorithm is one of the latest haptic rendering methods reported in the literature. It is also a GPU-based variant of the VPS technique, making it a suitable candidate for benchmarking purposes. As explained in section II-D, Zheng et al.'s method predicts k_c and k_f as a reciprocal function of V and its formulation is based on the work of Chandrasekharan et al. [18].

By contrast, the proposed algorithm relies on a more elaborated model that determines k_c and k_f by means of a regression model of $1 - \sin\gamma_n$, as suggested by [19]. To compare the modeling accuracy of both methods and to determine how they can be extrapolated to a non-tested machining scenario, a leave-one-out cross-validation study was carried out. More specifically, trials from three different machining scenarios were used to calibrate k_c and k_f, whereas the remaining scenario was used for testing. This process was repeated until all possible combinations of machining parameters were used for testing. Calibrations were performed in the starting phase of the trial when the bit gradually enters into the bone. As tool advances into the bone, more cutting lip elements engage with bone elements such that both axial thrust and torque will increase. Therefore, by knowing the drilling depth at each time point as well as the associated thrust/torque magnitude, the contribution of each element can be calculated along with their specific/empirical cutting force/torque coefficients.

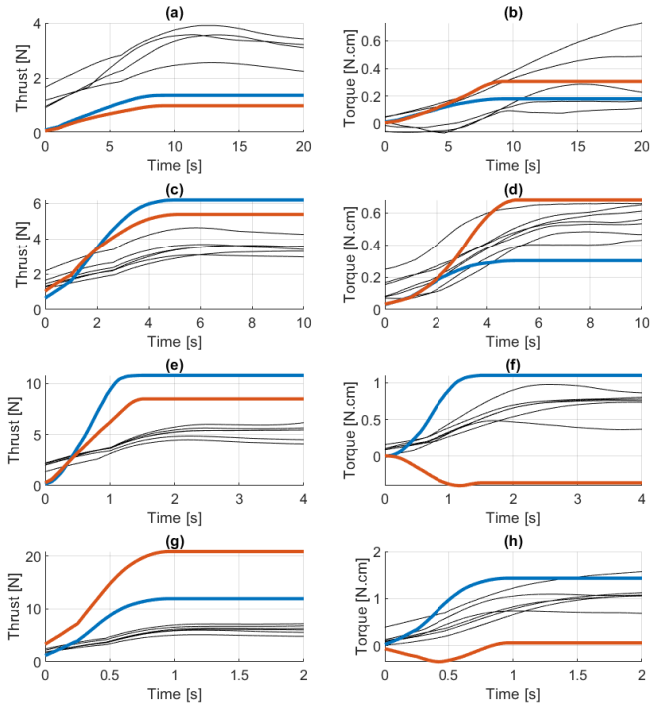

Fig. 4: Cross-validation thrust and torque prediction results for our algorithm (blue) and Zheng et al.'s method (coral) compared with experimental data (black). (a)-(b): $f = 0.01\ \frac{\text{mm}}{\text{rev}}$, $n = 500$ rpm, (c)-(d): $f = 0.02\ \frac{\text{mm}}{\text{rev}}$, $n = 500$ rpm, (e)-(f): $f = 0.01\ \frac{\text{mm}}{\text{rev}}$, $n = 3000$ rpm, and (g)-(h): $f = 0.02\ \frac{\text{mm}}{\text{rev}}$, $n = 3000$ rpm.

Since the modeling of the cutting forces developed on chisel edges is similar in both algorithms, only cutting forces developed on cutting lip elements will be analyzed further.

Figure 4 illustrates the results of the cross-validation analysis. First off, a high degree of variation between bone drilling trials is evident. This is likely caused by the inherent heterogeneity of bone properties that will in turn lead to different levels of thrust and torque even if machining conditions used were practically identical. Because of this, it will always be difficult to predict with accuracy the characteristics of a bone removal operation for a specific osseous specimen. However, the general trend of variation can be captured with a reasonable accuracy. The side-to-side comparison of both algorithms reveals that the prediction accuracy remains relatively low in both cases. This being said, the proposed algorithm seems to yield relatively more accurate torque predictions. Zheng et al.'s method remains characterized by a rather low prediction accuracy and in two cases yields completely inaccurate negative torque values. While it is reasonable to expect that prediction results might improve if more experimental results would be available for calibration, these results imply that the proposed algorithm performs overall better than the prior method. Evidently, the implementation of the proposed algorithm would translate in a more accurate haptic feedback associated with bone drilling operations.

To assess the running time, both algorithms were used to conduct visuo-haptic simulations of bone drilling trials by means of the Touch® haptic device. Trials were performed

TABLE I: Average (standard deviation) of running time in milliseconds for our method in different resolutions

Point Shell Resolution	**Voxmap Resolution**			
	128^3	256^3	512^3	1024^3
128^3	0.23 (0.04)	0.22 (0.04)	0.22 (0.04)	0.15 (0.04)
256^3	0.22 (0.03)	0.23 (0.04)	0.22 (0.04)	0.15 (0.04)
512^3	0.21 (0.05)	0.20 (0.05)	0.19 (0.05)	0.16 (0.04)
1024^3	0.20 (0.04)	0.18 (0.03)	0.20 (0.04)	0.21 (0.07)
2048^3	0.40 (0.07)	0.40 (0.07)	0.43 (0.08)	0.47 (0.14)

for different resolutions of voxmap and point shell structures. The running time of haptic and graphic rendering loops was measured in nanoseconds by means of the `chrono` timer available in the C++ standard library.

Tables I and II tabulate mean and standard deviations of running time measured for bone drilling simulation trials performed for different resolutions, each measured over multiple runs. These results indicate a significant difference in the running time of the algorithms. More specifically, while Zheng et al.'s method exceeds the haptic threshold of 1 ms running time for high resolutions, the running time of the proposed algorithm remains around 0.5 ms even in the case of the highest resolution tested. Interestingly, even at the highest resolution, the proposed algorithm was capable to maintain a 45 FPS graphic rendering framerate (50% more than the targeted 30 FPS). This is significantly better than Zheng et al.'s method that could not achieve more than 22 FPS graphic rendering refresh rate for 2048^3 voxmap, primarily due to the GPU being overloaded by less efficient computations.

It could be emphasized here that the running time of our algorithm shows low sensitivity to voxmap resolutions. This is a consequence of the implicit definition of voxmap as well as the fact that the voxmap resolution does not affect the number of threads to be launched by the GPU. Therefore, it could be speculated that the proposed algorithm could handle even higher resolutions as long as computing memory is available to store the required data structures.

The low running time of the proposed algorithm has two significant advantages: (1) Makes possible significant improvements of the haptic/graphic refresh rate. This results in a higher transparency of interactions with virtual environment and leads to improved simulation realism. (2) Enables the use of higher sampling resolutions for point shell and voxmap. This leads to smoother force/torque trajectories and increases the stability of the haptic rendering with positive effects on the haptic feedback quality.

B. Case Study 2: Glenoid Reaming

Glenoid reaming represents a challenging bone machining operation performed in shoulder joint replacement surgeries [22]. Because of its inherent complexity, glenoid reaming constitutes a suitable target for haptic-augmented surgery simulators. However, the mechanics of this bone removal operation has not been analyzed and hence no virtual replicas were devised so far. However, in the current context,

TABLE II: Average (standard deviation) of running time in milliseconds for Zheng et al.'s method in different resolutions

Point Shell Resolution	**Voxmap Resolution**			
	128^3	256^3	512^3	1024^3
128^3	0.27 (0.10)	0.29 (0.10)	0.35 (0.70)	1.98 (3.86)
256^3	0.28 (0.10)	0.30 (0.10)	0.71 (0.35)	3.90 (1.98)
512^3	0.41 (0.09)	0.35 (0.11)	0.76 (0.36)	3.97 (2.03)
1024^3	0.41 (0.08)	0.46 (0.10)	0.95 (0.35)	4.16 (2.00)
2048^3	1.18 (0.16)	1.28 (0.17)	1.80 (0.46)	5.02 (2.02)

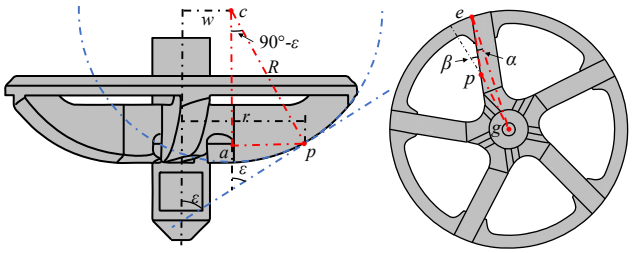

Fig. 5: Reamer geometry

glenoid reaming will be used to illustrate how the theoretical framework presented in Section II can be used to simulate the haptic-augmented interaction between tool and bone.

The first step for the determination of the elemental cutting forces in glenoid reaming is the evaluation of the geometrical characteristics of the surgical tool (also called reamer). For this purpose, the geometry of a clinical-grade standard nipple-guided spherical reamer (Fig. 5) was tessellated by means of a triangular mesh. The reamer under consideration contains five curved cutting edges, each corresponding to an arc of a circle with the radius of R and the center c which is at a distance of w from the tool axis. The central feature of the reamer does not participate in the bone removal process, but plays a guiding role for the tool. Due to the curvature of cutting lips, ε defined in Section II varies along the cutting lips. As such, by considering an element at a point p with radial distance of r, ε can be obtained by means of the triangle Δpac as $\varepsilon = \arccos\left(\frac{r-w}{R}\right)$. The web angle equivalent for reamer, shown in Fig. 5, can be obtained by applying the law of sines for the triangle Δepg which yields $\beta = \arcsin\left(\frac{R\sin\alpha}{r}\right)$, where α is the angle between the cutting edge and the radial direction at the periphery of cutting lips. The geometrical model of the reamer suggests that $\gamma_n \equiv 0$ along the cutting lips. Also, the expressions for μ and λ remain identical to those associated with the drilling tool. The rest of the reamer parameters were assessed as $R = 16.22$ mm, $w = 6.43$ mm, and $\alpha = 9.46°$.

To calibrate k_c and k_f, experimental data from robot-driven glenoid reaming of denuded cadaveric human scapulae were used. A robotic manipulator was used to conduct glenoid reaming on five freshly frozen cadaveric human scapulae, and the axial thrust and torque during each trials were measured using a 6 DOF load cell. All trials were conducted under supervision of an orthopedic surgeon to ensure they conform with the standard clinical practice. In addition, the use of the robot allowed high repeatability across all trials to ensure reliability of the experimental data.

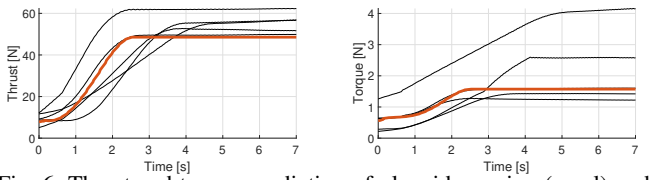

Fig. 6: Thrust and torque prediction of glenoid reaming (coral) and experimental data (black)

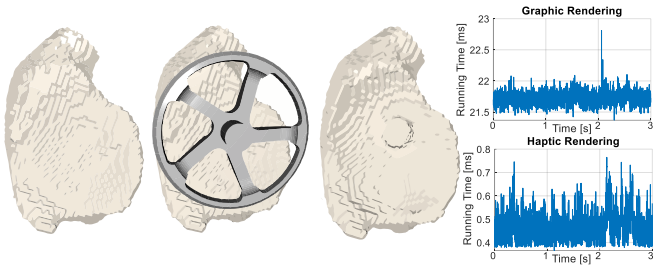

Fig. 7: Visuo-haptic simulation of glenoid reaming. From left to right: glenoid surface prior to the operation, simulation of glenoid reaming, replication of material removal, and running time of haptic and graphic rendering loops for voxmap resolution of 1024^3 and point shell resolution of 2048^3

The details of the experiment design and data collection can be found in [22], [27]. Since $\gamma_n \equiv 0$, k_c and k_f were modeled as a function of A_u and V. Figure 6 shows axial thrust and torque predictions and experimental data. The experimental glenoid reaming trials were conducted under identical machining parameters. As such, since no other similar data exists in the literature, the accuracy of the predictions could not be adequately verified.

Visuo-haptic glenoid reaming simulation trials were performed by means of different point shell and voxmap resolutions similar to Section III-A. The voxmap model was created from a clinical CT scan of human scapula, segmented, and upsampled to a desired resolution by means of the method described in [14].

Since the running time of VPS techniques are determined by the sampling resolution of voxmap and point shell structures, running times similar to Tab. I was expected. In fact, it was possible to run simulations with point shell resolution of 2048^3 and voxmap resolution of 1024^3 and reliably maintain the target haptic and graphic rendering refresh rates. Figure 7 depicts a graphical representation visuo-haptic simulation of glenoid reaming and its associated running times.

The main difference between bone drilling and glenoid reaming simulations is that the reamer is significantly larger than the drill bit (36.5 mm vs 2.5 mm diameter, respectively), such that the intersection volume is also correspondingly larger for the former. Consequently, the elements generated on tool and bone in glenoid reaming are coarser and this will likely lead to more abrupt changes in force/torque and lower quality visual replication. This mandates the use of computationally-efficient algorithms in order to increase the simulation realism. The algorithm proposed in this study enables haptic computations in resolutions as fine as 1024^3 and beyond. The glenoid used in the study measured $50 \times 37 \times 32$ mm^3 such that at a 1024^3 resolution, the voxel size will be in micron range level. As such, our algorithm allows

incorporating high-fidelity micro resolution bone models in haptic simulation of bone machining operations. To the authors' best knowledge, this is the first time to enable micro resolutions for bone machining simulation, which opens doors to incorporate micro-CT bone images throughout simulations, and thus use the inherent advantages that micro-CT images offers over their clinical CT counterparts to allow for more accurate modeling of bone structural properties throughout haptic computations.

IV. Conclusion

This study developed a new haptic rendering algorithm for simulation of bone machining operations. The proposed algorithm was established upon of efficient computational methods, mainly in a data parallel framework, and a more elaborated force model drawn upon latest models in the context of composite machining. Comparing the algorithm with the existing approaches revealed that improved force/torque prediction accuracy and significant computational efficiency advantages can be achieved, enabling micro-resolution geometrical representations of bone, while maintaining reliable haptic and graphic simulation of bone machining operations.

One of the limitations of this study was that the haptic device used for testings had a limited output force range and was essentially unable to render torques. To address this, future extensions of this work will focus on the implementation of the proposed algorithm on a 6 DOF haptic device with extended output force/torque ranges. The enhanced version of the haptic-augmented simulator will be used to improve the efficacy of surgical training in orthopaedics.

The algorithm presented in this study can be expanded in several directions. While the comparison results in Fig. 4 were in favor of the proposed force model, more in-depth analysis is required for the evaluation of the proposed force model. Of note, the high variability of bone material properties (both intra- and inter-specimen) combined with the wide range of machining parameters to be tested will require extensive and thereby expensive trials (owed primarily to the high cost of the human humeral specimens). Also, the investigation of the effect of bone material properties (density, elasticity, etc.) to be extracted from CT images on k_c and k_f coefficients could be another interesting line of inquiry in this regard.

While the proposed algorithm was implemented in a direct haptic rendering setup, future improvements could involve integrating the algorithm with virtual coupling, projection-based force reflection or other similar methods. Finally, the collision detection could be also augmented with a bisection algorithm in order to enable the determination of the time of contact and therefore provide a smoother haptic rendering in a continuous collision detection framework.

References

[1] K. Rangarajan *et al.*, "Systematic review of virtual haptics in surgical simulation: A valid educational tool?" *Journal of Surgical Education*, 2019.

[2] G. W. Thomas *et al.*, "A review of the role of simulation in developing and assessing orthopaedic surgical skills," *The Iowa orthopaedic journal*, vol. 34, p. 181, 2014.

[3] W. A. McNeely *et al.*, "Six degree-of-freedom haptic rendering using voxel sampling," in *ACM SIGGRAPH 2005 Courses*. ACM, 2005, p. 42.

[4] M.-D. Tsai and M.-S. Hsieh, "Accurate visual and haptic burring surgery simulation based on a volumetric model," *Journal of X-ray Science and Technology*, vol. 18, no. 1, pp. 69–85, 2010.

[5] M. Vankipuram *et al.*, "A virtual reality simulator for orthopedic basic skills: a design and validation study," *Journal of biomedical informatics*, vol. 43, no. 5, pp. 661–668, 2010.

[6] F. Zheng *et al.*, "An analytical drilling force model and GPU-accelerated haptics-based simulation framework of the pilot drilling procedure for micro-implants surgery training," *Computer methods and programs in biomedicine*, vol. 108, no. 3, pp. 1170–1184, 2012.

[7] S. Chan *et al.*, "High-fidelity haptic and visual rendering for patient-specific simulation of temporal bone surgery," *Computer Assisted Surgery*, vol. 21, no. 1, pp. 85–101, 2016.

[8] M. Arbabtafti *et al.*, "Physics-based haptic simulation of bone machining," *IEEE Transactions on Haptics*, vol. 4, no. 1, pp. 39–50, 2010.

[9] J. Lee *et al.*, "Modeling and experimentation of bone drilling forces," *Journal of biomechanics*, vol. 45, no. 6, pp. 1076–1083, 2012.

[10] Y. Lin *et al.*, "A predictive bone drilling force model for haptic rendering with experimental validation using fresh cadaveric bone," *International journal of computer assisted radiology and surgery*, vol. 12, no. 1, pp. 91–98, 2017.

[11] F. Zheng *et al.*, "Graphic processing units (gpus)-based haptic simulator for dental implant surgery," *Journal of computing and information science in engineering*, vol. 13, no. 4, 2013.

[12] H. Xu and J. Barbič, "6-dof haptic rendering using continuous collision detection between points and signed distance fields," *IEEE transactions on haptics*, vol. 10, no. 2, pp. 151–161, 2016.

[13] B. Heidelberger *et al.*, "Detection of collisions and self-collisions using image-space techniques," 2004.

[14] M. Faieghi *et al.*, "Fast and cross-vendor opencl-based implementation for voxelization of triangular mesh models," *Computer-Aided Design and Applications*, vol. 15, no. 6, pp. 852–862, 2018.

[15] M. Marco *et al.*, "A review on recent advances in numerical modelling of bone cutting," *Journal of the mechanical behavior of biomedical materials*, vol. 44, pp. 179–201, 2015.

[16] L. Yanping *et al.*, "Simulation and evaluation of a bone sawing procedure for orthognathic surgery based on an experimental force model," *Journal of biomechanical engineering*, vol. 136, no. 3, 2014.

[17] Y. Altintas, *Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design*. Cambridge University Press, 2012.

[18] V. Chandrasekharan *et al.*, "A mechanistic approach to predicting the cutting forces in drilling: with application to fiber-reinforced composite materials," *Journal of Engineering for Industry*, vol. 117, no. 4, pp. 559–570, 1995.

[19] M.-B. Lazar and P. Xirouchakis, "Mechanical load distribution along the main cutting edges in drilling," *Journal of Materials Processing Technology*, vol. 213, no. 2, pp. 245–260, 2013.

[20] M.-B. Lazar and P. Xirouchakis, "Experimental analysis of drilling fiber reinforced composites," *International Journal of Machine Tools and Manufacture*, vol. 51, no. 12, pp. 937–946, 2011.

[21] M. Faieghi *et al.*, "Parallelized collision detection with applications in virtual bone machining," *Computer Methods and Programs in Biomedicine*, vol. 188, p. 105263, 2020.

[22] M. Faieghi *et al.*, "Vibration analysis in robot-driven glenoid reaming procedure," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2020.

[23] M. Faieghi *et al.*, "Fast generation of cartesian meshes from micro-computed tomography data," *Comput. Aided Des. Appl.*, vol. 16, no. 1, pp. 161–171, 2019.

[24] J. Baert *et al.*, "Out-of-core construction of sparse voxel octrees," in *Proceedings of the 5th high-performance graphics conference*, 2013, pp. 27–32.

[25] M. Razavi *et al.*, "A gpu-implemented physics-based haptic simulator of tooth drilling," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 11, no. 4, pp. 476–485, 2015.

[26] C. Dyken and G. Ziegler, "GPU-accelerated data expansion for the marching cubes algorithm," in *Proc. PGU Technology Conf*, 2010, pp. 115–123.

[27] M. Sharma, "Experimental Determination of Motion Paramters and Path Forces of Robot-Driven Glenoid Reaming," Master's thesis, The University of Western Ontario, 2018.