Title:
Computationally Efficient Software Tool for Virtual Hand Repositioning

Authors:
Adam Gorski, agorski3@uwo.ca, Western University, Canada
O. Remus Tutunea-Fatan, rtutunea@eng.uwo.ca, Western University, Canada
Louis M. Ferreira, lferreir@uwo.ca, Western University, Canada

Keywords:
Hand Kinematics, Splint Generation, Finite Element Method, GPU-Based Parallelization

DOI: 10.14733/cadconfP.2024.238-242

Introduction:
As the utilization of 3D printed splints advances in physiotherapeutic contexts, there are still obstacles hindering their widespread acceptance. A significant challenge revolves around the absence of the ability to adjust the position of limbs and fingers, a capability present in conventional hand splinting methods. Typically, the process involves scanning, constructing a model, building the splint, and finally, 3D printing. This contrasts with the traditional manufacturing approaches, where a thermoplastic with a low glass transition temperature is bent around patient's hand. During this process, a physiotherapist can manipulate patient's fingers to ensure that the splint hardens in a specific shape. Unfortunately, the replication of this dynamic adjustment is not feasible with 3D printed splints since patients may have difficulties in attempting to independently maintain their fingers in the prescribed recovery positions whereas the use of supports during in a non-ionizing photogrammetry scanner might significantly obstruct the geometry of the hand.

To tackle these obstacles, the primary goal of this research was to devise a post-scanning technique capable of adjusting the finger positions in a 3D hand model using a finite element method. This numerical approach is intended to provide greater physical accuracy, making it more realistic than counterpart methods relying solely on the rotation of finger geometry. As demonstrated later in this study - despite making use of rather limited anatomical data - the developed software tool proved to be efficient, rapid, and interactive, making it potentially attractive to a wide range of medical practitioners specializing in hand therapy,

Processing Workflow:
The software was designed with a linear step-by-step approach that guides the user through out the joint repositioning process as shown in Fig.1. The initial phase involves importing an STL file representing a hand scan that requires repositioning. The user is then tasked with adjusting the file to the accurate dimensions. Following that, the subsequent step entails placing individual joints and finger tips onto the model. This is achieved by selecting the target joint from the left pane or an already positioned joint, and then clicking on its target location on the 3D model. The software intersects the model twice, ensuring that the joints end up within the 3D model rather than just on its outer surface. Alternatively, the user can utilize a translational tool within the software to move the joints. In the final step, the user must

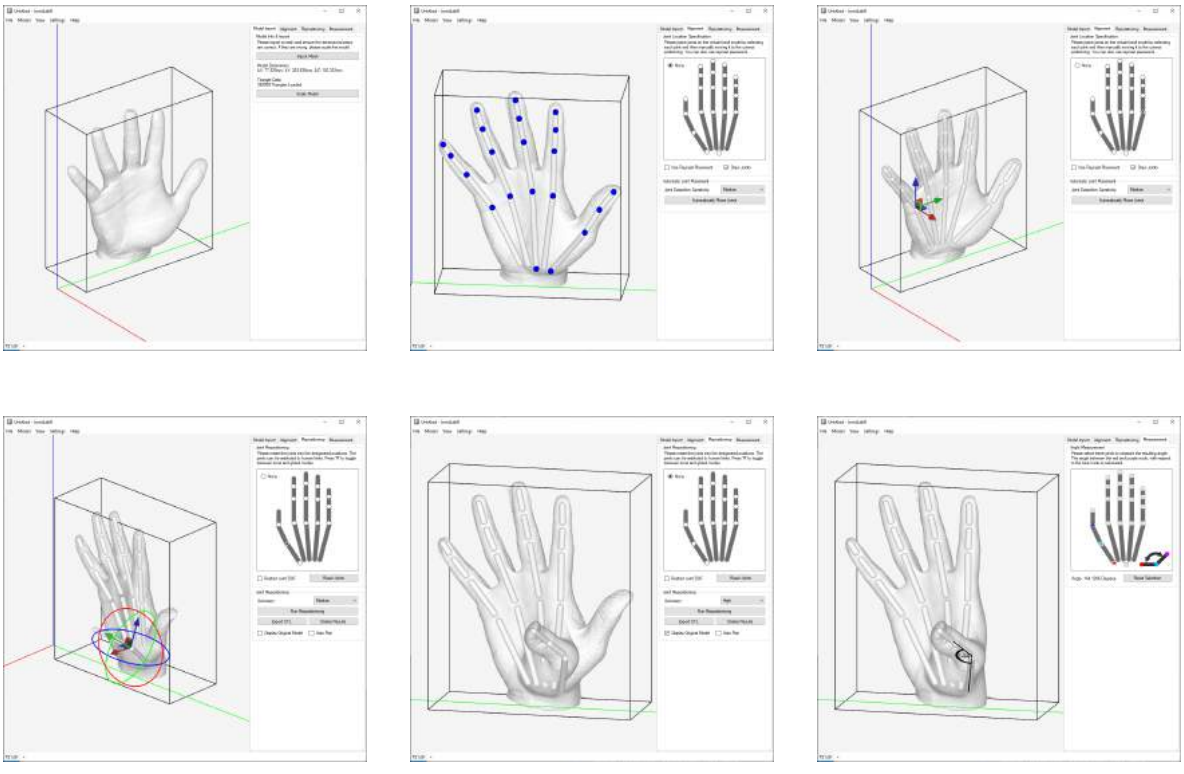

Fig. 1: Workflow phases.

rotate each joint to the desired orientation using a custom-built rotational tool. The real-time view is based on the *simplified* bone structure of the hand, and if desired, the user can display a real-time view of the deforming mesh at reduced quality. The user also has the option to enable/disable human physical restrictions, allowing only anatomically correct bone rotations. Additionally, there is an optional step that allows the display of angles between finger bones. This feature enables medical professionals to position the fingers in predetermined recovery positions and angles based on the type of injuries.

Software Implementation:

The complete software tool was coded in C#, utilizing OpenCL or CUDA for solving the finite element matrix. The software operates independently of external libraries, as everything has been incorporated into a standalone executable file. WinForms was employed for the user interface, and OpenGL was utilized for rendering the 3D model. The joint control system relies on matrices, with quaternions employed to interpolate between the initial and final bone targets during each load step. The interpolation steps and rotational tool are illustrated in Fig.2. The software also includes an automated joint placement feature, utilizing high and low curvature regions calculated in OpenCL to expedite the calculation process.

Finite Element Mesher and Solver:

Since significant displacements occur during limb repositioning, multiple load steps are necessary. This introduces challenges when the stiffness of each element is recalculated. In numerous instances, the internal elements undergo excessive distortion and this leads to a negative Jacobian and subsequent failure

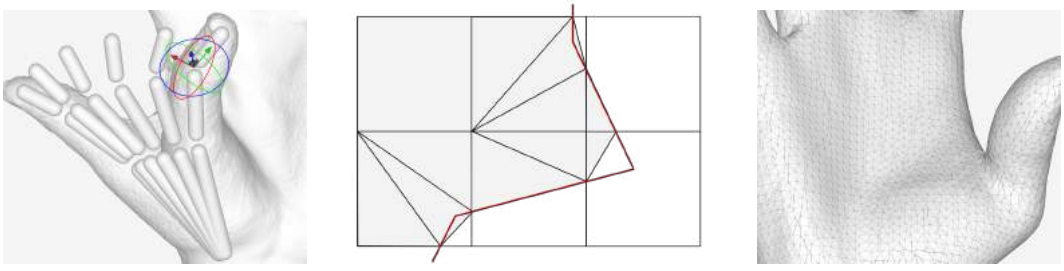

Fig. 2: Left: spherical Interpolation; Center: FE mesh generation; Right: Resulting FE mesh.

of the isoparametric mapping. To address this issue, the mesh was reconstructed from the beginning during each load step. Furthermore, to facilitate real-time calculation, a specialized mesher had to be used to accelerate the construction of the mesh, particularly since meshing could require a considerable amount of time. While Delaunay approaches alongside with external libraries (such as those detailed in [1]) were explored initially, the solution incorporated in the software relies on a hexahedron-based voxel mesh coupled with interpolated tetrahedrons. This simple approach is based on the marching-cubes algorithm [2] but with interpolated vertices (Fig.2).

The meshing process starts by converting the mesh into voxels. This computational intensive operation is carried out by the graphics processing unit (GPU) . The intersection points along the edge of each scanline are used to create the interpolation data required to obtain a seamless surface. This data can be directly used by a solver that does not rely on a mesh or assembly [3]. Nevertheless, achieving a smooth surface becomes challenging due to the voxel-like nature of the mesh. Along these lines, although the use of a standard Laplacian smoothing could represent a potential solution, complications might arise due to the volume changes to accompany both voxelization and smoothing operations. The tests were performed by interpolating marching cube data and while the resulting mesh appeared visually plausible, the issues with volume persisted.

The subsequent step relies on the construction of the elements and nodes for the finite element mesh. Since this phase can be parallelized only in part, its execution was directed to the CPU. At this stage, the interpolation data can be processed to generate tetrahedrons that are essential for a seamless skin surface. The generated mesh can be fed into a matrix-less solver or it can be assembled into a sparse matrix and solved conventionally. Given that the computation of the mesh or the assembly of the matrix are not time consuming operations, the algorithm was set to default to the sparse matrix solver that is known to be the fastest of the three analyzed options. The three solving approaches described above are depicted in Fig.3.

Upon completion of the construction of the finite element mesh, the global stiffness matrix is organized into sparse 3x3 blocks to minimize both memory usage and bandwidth consumption. Subsequently, the employment of a custom block Jacobi preconditioned conjugate gradient solver enables a rapid solution. The developed solver can be executed on GPU via CUDA or OpenCL, as well as on CPU via AVX vector acceleration. This approach yields virtually instantaneous results. For each load step, the joint control tools (translational and rotational) undergo conversion to quaternions as well as spherical interpolation. This is required to ensure that joints meticulously adhere to a physically accurate trajectory toward their designated locations.

Material Properties:

For simplicity reasons, the entire model of the hand was assumed to consist of bone, muscle and joints only. While this representation might be regarded as simplified compared to others such as those presented by

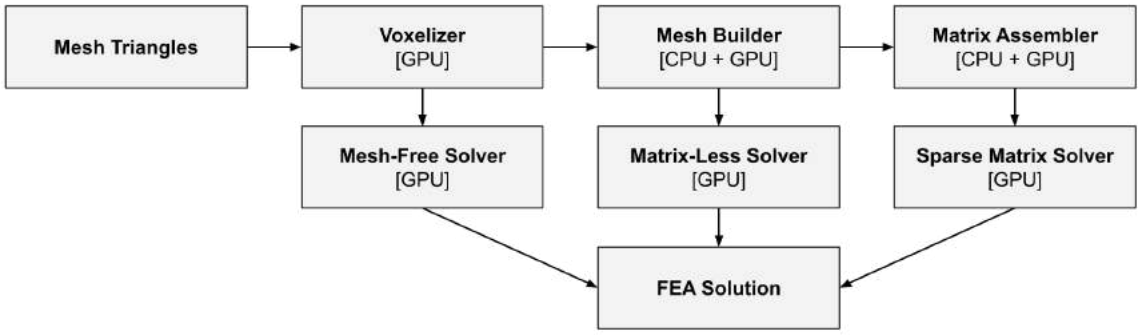

Fig. 3: FEA solving options.

[4], it is unlikely that the model adopted will greatly affect the displacements that represent the main target of the present work. Materials were assumed to be linear isotropic. Bone stiffness was set to 20 GPa while that of the joints and muscles were set to 5 kPa and 100 kPa, respectively. Poisson's ratio was assumed to be 0.3. A simplified bone model consisting of "capsules" was used, whereas the outer geometry of the hand was acquired via a photogrammetry scanner.

Results:

In each test scenario, two scans were captured of the identical hand in distinct positions. Subsequently, these two scans were aligned to ensure the correspondence between the dorsal side of the hand. After that, virtual bones were introduced and adjusted such that the initial scan aligned with the second one that was obtained for a different pose of the hand having the thumb in a different position. At the last step, the repositioned hands were compared by means of the surface deviation map.

As Fig. 4 implies, the accuracy appears to be acceptable for small displacements. While the developed software tool seems to encounter difficulties for excessively large displacements and sharp corners, they are not regarded as important since the traditional procedure that relies on the fabrication of the splints is likely to be characterized by much higher error values. The tests performed were characterized by a maximum error of slightly under 6 mm, a mean absolute deviation of about 0.8 mm and root mean square error of around 1.4 mm. Since the virtually repositioned mesh/hand tends to underestimate the real geometry of the hand, the splints to be fabricated based on the virtually repositioned geometry will end up being slightly tighter than expected. While the split remains usable (hand tissue will redistribute to adjust to the geometry of the splint), additional work is presently underway to correct/reduce the discrepancy between the skin of the virtually repositioned hand and that of its physical counterpart.

Conclusions:

The main objective of the current study was to develop a software tool capable to virtually reposition the hand of a patient whose mobility is limited as a consequence of various pathological conditions. This virtual reposition is often necessary in order to enable the design of a hand splint to be fabricated for therapeutic reasons. To address the study objective, an innovative software tool - capable to mimic the kinematics of the hand - was developed from scratch. This software tool - encompassing a large finite element model - is essentially capable to estimate the deformation of the hand skin according to the kinematics of the bony structure. To reduce the computational time, advanced parallelization techniques were used for solving purposes. The preliminary tests presented above suggest that the accuracy of the software tool yields within an acceptable range.

Fig. 4: Left: original hand; Center: real hand; Right: repositioned hand.

The developed software tool empowers physiotherapists to generate 3D printed splints to be used for the treatment of hand injuries. Unlike the conventional methods, the software tool eliminates the need to move patients' hands into potentially uncomfortable positions that cannot be either acquired or sustained for prolonged periods of time due to the underlying pathological conditions. Future developments will attempt to increase the realisticity and the accuracy of the developed software tool.

Acknowledgement:
This research was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Adam Gorski, https://orcid.org/0009-0004-4429-8256
O. Remus Tutunea-Fatan, https://orcid.org/0000-0002-1016-5103
Louis M. Ferreira, https://orcid.org/0000-0001-9881-9177

References:

[1] Faraj, N.; Thiery, J.M.; Boubekeur, T.: Multi-material adaptive volume remesher. Computers Graphics, 2016, 58, 150-160. ISSN 0097-8493. https://doi.org/10.1016/j.cag.2016.05.019

[2] Lorensen, W.E.; Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. InSeminal graphics: pioneering efforts that shaped the field, 347-353, 1998. https://doi.org/10.1145/37402.37422

[3] Lopes, P.C.F.; Pereira, A.M.B.; Clua, E.W.G.; Leiderman, R.: A gpu implementation of the pcg method for large-scale image-based finite element analysis in heterogeneous periodic media. Computer Methods in Applied Mechanics and Engineering, 2022, 399, 115276 https://doi.org/10.1016/j.cma.2022.115276

[4] Harih, G.; Tada, M.: Chapter 21 - development of a feasible finite element digital human hand model. In S. Scataglini; G. Paul, eds., DHM and Posturography, 273-286. Academic Press, 2019. ISBN 978-0-12-816713-7. https://doi.org/10.1016/B978-0-12-816713-7.00021-0