HHS Public Access

Author manuscript

Comput Methods Biomech Biomed Eng Imaging Vis. Author manuscript; available in PMC 2019 February 07.

Published in final edited form as:

Comput Methods Biomech Biomed Eng Imaging Vis. 2018 ; 6(5): 491–498. doi: 10.1080/21681163.2016.1172346.

Efficient Computation of Cartilage Contact Pressures within Dynamic Simulations of Movement

Colin R Smith[a], **Kwang Won Choi**[a], **Dan Negrut**[a], and **Darryl G Thelen**[a,b,c,*]

[a]Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, USA

[b]Department of Biomedical Engineering, University of Wisconsin-Madison, Madison, USA

[c]Department of Orthopedics and Rehabilitation, University of Wisconsin-Madison, Madison, USA

Abstract

The objective of this study was to assess the use of an advanced collision detection algorithm to simulate cartilage contact pressure patterns within dynamic musculoskeletal simulations of movement. We created a knee model that included articular cartilage contact for the tibiofemoral and patellofemoral joints. Knee mechanics were then predicted within the context of a dynamic gait simulation. At each time step of a simulation, ray-casting was used in conjunction with hierarchical oriented bounding boxes (OBB) to rapidly identify regions of overlap between articulating cartilage surfaces. Local cartilage contact pressure was then computed using an elastic foundation model. Collision detection implemented in parallel on a GPU provided up to a 10× speed increase when using high resolution mesh densities that had >10 triangles/mm^2. However, pressure magnitudes converged at considerably lower mesh densities (2.6 triangles/mm^2) where CPU and GPU implementations of collision detection exhibited equivalent performance. Simulated tibiofemoral contact locations were comparable to prior experimental measurements, while pressure magnitudes were similar to those predicted by finite element models. We conclude the use of ray-casting with hierarchical OBB for collision detection is a viable method for simulating joint contact mechanics in human movement.

Keywords

joint contact; collision detection; elastic foundation; cartilage contact pressure

1. Introduction

The loading of articular cartilage during functional movement is important to consider when investigating cartilage health and pathology. Because cartilage loading cannot be directly measured *in vivo*, computational modelling is a valuable tool to investigate dynamic cartilage loading and provide insight into surgical treatments and rehabilitation protocols. However, the prediction of cartilage loading within multibody movement simulations presents a complex computational problem. While finite element analyses (FEA) are conventionally used to estimate cartilage tissue stress, they remain too computationally

*Corresponding author: dgthelen@wisc.edu.

expensive to solve within the context of a movement simulation (Halloran et al. 2009; Halloran et al. 2010). **Traditional multibody musculoskeletal models resort to simplified kinematic joints to reduce complexity (Delp et al. 1990; Edith M. Arnold et al. 2010), but these intrinsically ignore the load dependent behaviour of the joint and cannot provide estimates of cartilage loading. As a result, elastic foundation models have been introduced to efficiently model joint contact within whole body simulations of movement** (Bei & Fregly 2004; Guess et al. 2014; Thelen et al. 2014).

In elastic foundation models, cartilage is considered an elastic tissue bonded to a rigid bone substrate. The articulating cartilage geometries are represented by surface meshes which can interpenetrate. Pressure on each element in contact is computed independently of neighbouring elements based on the local overlap distance and the thickness of the cartilage tissue. The resulting model of the cartilage tissue layer is mechanically equivalent to a bed of independent nonlinear compressive springs distributed over rigid bones (Bei & Fregly 2004). Elastic foundation approaches have been used to study contact mechanics in various diarthrodial joints including the hip (Abraham et al. 2013), knee (Elias et al. 2004; Bei & Fregly 2004; Elias & Cosgarea 2007; Anderson et al. 2010), and ankle (Haraguchi et al. 2009; Anderson et al. 2010).

Although elastic foundation models are substantially faster than FEA, in practice the detection of overlapping faces of high resolution cartilage meshes remains a computational bottleneck. This task becomes remarkably burdensome within a dynamic simulation of movement, where collision detection and penetration depth calculations are repeated a minimum of once per time step (Thelen et al. 2014; Smith et al. 2015). A brute force approach to collision detection evaluates penetration between every pair of elements of two surfaces, resulting in an $O(n^2)$ complexity. Prior biomechanical studies have reduced the computational demand by only searching physically plausible contact regions (Bei & Fregly 2004; Arbabi et al. 2009). Meanwhile, the computer graphics community has introduced a number of general purpose algorithms to accelerate collision detection of objects represented by polygon meshes (Lin & Gottschalk 1998). **One approach uses ray-casting in conjunction with hierarchical oriented bounding boxes (OBBs) to efficiently identify overlapping regions of two polygon meshes (Arvo & Kirk 1989). The ray-casting OBB algorithm provides local overlap depth estimations making it well suited for joint contact. Additionally, it can be implemented in parallel, and thus in theory made substantially faster when run on a parallelized graphics processor unit (GPU) (Lauterbach et al. 2010). However, it remains unclear what mesh resolution is required to obtain reliable cartilage contact pressures and whether the increase in computational speed resulting from a GPU implementation outweighs the memory transfer overhead.**

The objective of this study was to assess the feasibility of calculating cartilage contact pressures within a multibody dynamic simulation of movement using ray-casting with OBBs for collision detection. As a test case, knee cartilage contact pressures were simulated during walking using elastic foundation contact models. We investigate the computational efficiency of CPU and GPU implementations of collision detection and also assess the effects of mesh density on pressure magnitudes and contact areas.

2. Methods

2.1. Knee model

A multibody knee model was developed from magnetic resonance (MR) images of the right knee of a healthy young adult female (age = 23 years, height = 1.65 m, mass=61 kg). Development and validation of the knee model are detailed elsewhere (Lenhart et al. 2015). Briefly, bone and cartilage surface geometries of the tibia, femur and patella were manually segmented from the MR images and converted to triangulated surface meshes (MIMICS, Materialise Group, Leuven, Belgium). The origins, insertions and paths of 14 ligaments were also segmented and represented as bundles of nonlinear springs. The knee model allowed for six degree of freedom (DOF) tibiofemoral and patellofemoral motion. The knee was integrated into a generic lower extremity musculoskeletal model (Edith M Arnold et al. 2010), which included 44 muscles acting about the hip, knee and ankle joints (Figure 1). The full model was implemented in SIMM (Delp & Loan 1995) with the Dynamics Pipeline (Musculographics Inc., Santa Rosa, CA) and SD/Fast (Parametric Technology Corp., Needham, MA) used to generate the code describing muscle-tendon dynamics and the multibody equations of motion.

2.2 Contact Pressure

At each time step of a simulation, cartilage contact pressures were calculated using an elastic foundation model. Cartilage contact was determined by the overlap of cartilage surface meshes fixed to the femur, tibia and patella segments. These segments were positioned in accordance with the current system state. The contacting triangles of cartilage meshes were determined using an OBB collision detection algorithm (Section 2.3).

The contact pressure (p) on an individual triangle in each cartilage mesh was calculated according to elastic foundation theory developed for articular cartilage (Bei & Fregly 2004):

$$p = -\frac{(1-\nu)E}{(1+\nu)(1-2\nu)}\ln\left(1-\frac{d}{h}\right) \quad (1)$$

where E is the cartilage elastic modulus, ν is the cartilage Poisson's ratio, d is the local overlap depth and h is the local cartilage thickness. E and ν were assumed to be 5 MPa and 0.45 respectively (Blankevoort & Huiskes 1991; Caruntu & Hefzy 2004). The calculation of d is defined in the following section (Eq. 2). Local cartilage thickness (h) was computed by casting a ray from each triangle in the cartilage mesh towards the underlying bone mesh. The ray-triangle intersection was determined using the collision detection algorithm and Eq. 2 determined the local thickness.

2.3 Collision Detection

The pressure calculation required local cartilage overlap depth values for each contacting triangle in the surface meshes. Prior to the simulation, an OBB tree was constructed for the femoral cartilage geometry using the Proximity Query Package (PQP) software (Larsen et al. 2014). The OBB tree was constructed using a top down approach, where the parent box in the hierarchy encloses the entire mesh and is recursively subdivided to generate child

OBBs. The lowest level of the OBB tree is a leaf node, which consists of a bounding box fit around a single triangle. The OBBs were oriented to the principal vectors of the covariance matrix calculated from the positions of the enclosed triangle vertices to ensure a tight fit (Gottschalk et al. 1996).

For each pair of contacting cartilage surfaces, we defined *contact* (tibia/patella) and *target* (femur) surfaces. To check for contact, a normal ray was cast in both directions from the centre of each triangle in the *contact* meshes. A ray-OBB intersection test determined if the ray intersected the parent box of the OBB tree (Williams et al. 2005). If no intersection occurred, the triangle was not in contact and the test was terminated. If the ray intersected the parent OBB, then a ray-OBB intersection test was performed for each child OBB in the next sub-hierarchy. This process was repeated recursively until a leaf node was reached, resulting in a pair of potentially contacting triangles (Figure 2).

The depth of penetration (d) for each triangle pair was computed using:

$$d = \frac{\left(\overrightarrow{P_t} - \overrightarrow{C_c}\right) \cdot \widehat{n_t}}{\widehat{n_c} \cdot \widehat{n_t}} \quad (2)$$

where $\overrightarrow{P_t}$ is the intersection point on the *target* triangle (femur), $\overrightarrow{C_c}$ is the centre of the ray-casting triangle (*contact* body), $\widehat{n_t}$ is the *target* triangle unit normal vector and $\widehat{n_c}$ is the *contact* triangle unit normal vector (Figure 3). The intersection point on the *target* triangle was used for the distance calculation instead of the triangle centre to insure C0 continuity when the model pose changed and the ray intersected a neighbouring triangle. A positive value of d indicated the triangles were contacting and pressure was computed.

In practice, we included several constraints on the algorithm which accelerated its computational performance and robustness. A maximum distance threshold was defined to restrict ray intersections to a feasible region. In cases of extreme concavity where multiple OBBs were intersected, each path was traced through the OBB hierarchy. If this resulted in more than one intersected triangle, only the closest was used. Finally, to exploit the small changes in segment poses between time steps, a ray-intersection test was performed with the contacting triangle from the previous time step before progressing through the OBB tree.

2.4 GPU implementation

The collision detection algorithm was implemented both on a Central Processing Unit (CPU: AMD Phenom II X6 1055 T Processor 2.8GHz, with 8GB Main RAM) and a Graphics Processing Unit (GPU: NVIDIA GeForce GTX 560Ti with 1GB Graphics RAM). **The OBB tree was constructed on the GPU using the gProximity software (Lauterbach et al. 2014) which first groups the triangles into hierarchies, then fits OBBs to these groups in parallel using the same principal component definition as the CPU (Lauterbach et al. 2010).** The GPU implementation of the collision detection relied on Compute Unified Device Architecture (CUDA) (Nickolls et al. 2008) to perform all ray-OBB intersection tests **of a single hierarchy level** in parallel on 384 CUDA cores.

2.5 Neuromuscular simulations of walking

To assess the performance of the contact algorithm, we used the musculoskeletal model to simulate tibiofemoral cartilage contact pressures during walking. **The trajectories of reflective markers placed over bony landmarks, and ground reaction loads were recorded while the subject walked overground at a preferred speed in a motion analysis laboratory(Lenhart et al. 2015). At each frame of the gait cycle, a global optimization inverse kinematics routine determined pelvis translations, pelvis rotations, hip angles, knee flexion angle, and ankle angle that minimized the sum of squared differences between model marker locations and measured marker locations. At this stage, secondary tibiofemoral and all patellofemoral degrees of freedom were assumed to be a constrained function of tibiofemoral flexion, with these functions based on our simulated passive knee behaviour.**

A computed muscle control (CMC) algorithm was used to compute the muscle excitations needed to drive the model to track the measured hip, knee and ankle kinematics, while the pelvis motion was prescribed to the measured coordinates (Thelen et al. 2014) (Figure 4). **CMC is a feedforward-feedback controller that uses the error between the simulated and measured kinematics at the current time step to compute muscle excitations required to generate the measured the joint angle trajectories. At each time step, the current pose of the femur, tibia and patella were used to calculate the ligament forces and cartilage contact pressures. Measured ground reactions were applied directly to the feet. The computed muscle excitations were applied to the model and the equations of motion integrated to the next time step where the process was repeated. All tibiofemoral kinematics except flexion and all patellofemoral kinematics were allowed to evolve naturally as a result of the calculated muscle forces, ligament forces and cartilage contact pressures.**

2.6 Performance tests

We performed a series of tests to assess the influence of the cartilage mesh density and processor implementation on the contact pressure computation time and contact pressure patterns. Computation times needed to calculate the contact pressures were evaluated for the CPU and GPU implementations. These timed computations were performed at a single frame of the walking simulations, with the femur and tibia positions set to correspond to the second peak of the tibiofemoral loading during stance. Computation times were repeated for cartilage surface mesh resolutions varying from 0.02 triangles/mm^2 to 144 triangles/mm^2. Re-meshing was performed using a proprietary algorithm which ensures uniform tessellation (Geomagic, 3D Systems, Rockhill, SC). Reported GPU times included the transfer time between the main memory in the host and the global memory in the GPU. Additionally, we compared mean pressure, contact area and centre of pressure (COP) to assess the mesh density for which the cartilage surfaces generated converged values.

3. Results

3.1. Neuromuscular simulation of walking

The nominal muscle-actuated gait simulation closely tracked the measured kinematics (RMS error <1.0^{o} for all joints). Simulation of one gait cycle took 120 minutes to generate using the CPU contact detection implementation and a mesh density of 2.6 triangles/mm^2. Net tibiofemoral contact force patterns exhibited the characteristic double-peak during stance, with the majority of the force passing through the medial compartment. The mean pressure on the medial tibial plateau showed a similar trend with a double peak during stance (1st peak = 6.7 MPa, 2nd peak = 6.5 MPa), and reduced contact pressure during swing. The mean pressure on the lateral tibial plateau peaked at heel strike (5.0 MPa) and then remained relatively constant through the rest of the gait cycle (Figure 5). Contact on the medial compartment moved to the posterior portion of the plateau at the first peak, then to the anterior region of the plateau for the second peak. The medial centre of pressure (COP) translated more than the lateral COP in the anterior-posterior direction over the gait cycle (Range: medial = 14.7 mm, lateral = 10.8mm).

3.2. Sensitivity to Mesh Density

The mean pressure, contact area and centre of pressure metrics all converged to consistent values as mesh density was increased. A mesh density of at least 2.6 triangles/mm^2 was required for these metrics to fall within 1% of the converged values (Figure 6).

3.3. Computation Times

Solution time for contact detection at a single pose increased with increasing mesh density for both the CPU and GPU implementations. The GPU implementation of the contact detection algorithm resulted in similar computation times to the CPU implementation at mesh densities up to approximately 10 triangles/mm^2. Thereafter, the rate of increase in computation time was greater for the CPU implementation (Figure 7). At the highest tested mesh density of (144 triangles/mm^2), the GPU implementation was 10× faster than the CPU implementation.

4. Discussion

Computing cartilage contact pressures within a dynamic simulation requires that collision detection is performed at every time step. However, collision detection is computationally demanding when high resolution cartilage surface meshes are used. In this study, we demonstrate that the use of ray casting with oriented bounding boxes can accelerate collision detection, allowing for the simulation of tibiofemoral and patellofemoral cartilage contact pressures within a dynamic simulation of gait. The new computational approach allows for investigations of causal relationships between ligament properties, contact pressures and muscle coordination to be performed (Smith et al. 2015), which have great importance to both orthopaedic and rehabilitative medicine.

Our gait simulation predicted the characteristic bimodal loading of the tibiofemoral joint during the stance phase of walking. The medial contact pressures were higher than on the

lateral side, and the medial centre of pressure progressed anteriorly from the first peak of tibiofemoral loading to the second peak during stance. *In vivo* tibiofemoral contact pressures during gait have not been measured, thus direct validation is not feasible. However, the simulated pressure patterns agree favourably with image-based measures of tibiofemoral contact patterns during normal gait. Liu et al. measured cartilage deformations of 7 to 23% during the stance phase of gait, with larger anterior-posterior excursions and contact areas on the medial tibial plateau than on the lateral plateau (Liu et al. 2010). Our gait simulations show similar trends; our medial contact area estimates area are close to Liu et al.'s measurements, however our lateral contact area tend to be ~50% lower.

Collision detection has been extensively studied in the computer graphics literature, yet the existing algorithms have not been leveraged in biomechanical modelling due in part to their general purpose formulation (Arbabi et al. 2009). We implemented a popular computer graphics approach that combines ray-casting and a hierarchical structure of oriented bounding boxes to identify overlapping faces of two cartilage surfaces. While alternate collision detection algorithms exist that rely on hierarchical structures of various bounding volumes, the OBB is one of the best suited for the joint contact application. Other bounding volumes such as spheres and axis-aligned bounding boxes (AABB) perform fast intersection tests and do well in "rejection tests" when meshes are far apart. However, these algorithms produce looser fitting bounding boxes which result in more "false positives", where a ray intersects the bounding volume, but none of the included triangles. For joint contact applications where congruent meshes are in close proximity, the OBB excels because it fits tightly around the collection of triangles, yet still allows for fast intersection tests. (Hahn 1988; Gottschalk et al. 1996). For a mesh density of 2.6 triangles/mm^2, we found the OBB approach was over three orders of magnitude faster than a brute force approach that checks for contact between all face pairs between two surfaces.

Rather than employ general purpose algorithms, prior biomechanical modelling studies have done well to exploit knowledge of the application, such as physically plausible joint behaviour to improve collision detection. Bei and Fregly restricted collision detection to neighbours of previous contact patches, since relatively small changes in pose occur between simulation time steps (Bei & Fregly 2004). Arbabi and colleagues pre-processed cartilage surfaces into spatial bins, which rely on the relative proximity and nature of the movement between cartilage surfaces to reduce the number of computations (Arbabi et al. 2009). We also found that knowledge of small joint motion between time steps could be used to speed up our algorithm. In particular, considerable speedup was achieved by first performing a ray-intersection test with the contacting triangle from the previous time step before repeating the OBB checks. Additionally, although the algorithm could work with closed meshes, we used only the contacting surface of the cartilage to reduce the number of triangles of potential triangle contacts. When implemented in this way, CMC generated gait simulations in ~120 minutes using the minimum mesh density required to achieve converged pressure metrics. While considerably longer than using simplified joints (Thelen & Anderson 2006), the simulations provide considerably more biomechanical information such as ligament loading, ligament stretch, and cartilage contact pressure patterns (Smith et al. 2015).

Elastic foundation models treat each face of a cartilage surface mesh independently, which allows them to be easily parallelized on a GPU. We found that a GPU collision detection algorithm (Lauterbach et al. 2010) was 10× faster than a serial CPU implementation when using very high resolution meshes (144 triangles/mm^2). However, contact pressure and area converged at lower resolution meshes (2.6 triangles/mm^2), where the GPU and CPU implementations exhibited comparable computation times. The limited computational performance gains at these mesh densities is likely due to the overhead costs associated with data transfer that arises in GPU scientific computing. The GPU implementation may provide benefit for applications which require meshes with large number of faces such as skin-prosthesis interfaces (Faustini et al. 2006; Lee & Zhang 2007) or foot-ground contact (Neptune et al. 2000).

There are several limitations to consider in this work. First, we did not include a meniscus in our knee model, which is well recognized to distribute pressure in the tibiofemoral joint. Recent studies have introduced discretized meniscus models(Guess et al. 2010) that would be well suited to incorporate into our multi-body knee model in the future and will further increase the need for fast contact detection. Additionally, our pressure calculations assumed linearly elastic cartilage tissue behaviour, which clearly ignores viscoelastic effects. The effects of viscoelasticity may be negligible when considered in context of the assumptions required for the elastic foundation model. **While the elastic foundation model assumes that the deformation of each element is independent of neighbouring elements, it's predictions of cartilage contact pressures have shown good agreement with FEA models (Guess et al. 2013; Abraham et al. 2013) and experiments (Anderson et al. 2010).** However, if more complex constitutive representations of cartilage are required, it may be preferable to use gait simulation outputs as boundary conditions on a conventional finite element model.

We conclude that the use of an elastic foundation model with a ray-casting OBB contact detection algorithm is a viable approach for simulating articular contact within the context of dynamic full body movement simulations. The computational speed achieved allows for musculoskeletal simulations involving joint contact to be performed more readily, permitting the use of probabilistic approaches to look at how injury and intervention-induced changes in knee structure may affect *in vivo* knee mechanics and function.

Acknowledgements

The authors gratefully acknowledge the support of National Science Foundation (NSF) Grant 0966535 and National Institutes of Health (NIH) Grant EB015410. Additionally, we thank Rachel Lenhart, Jarred Kaiser and Anne Schmitz for their contributions to the development of the knee model and Radu Serban for his constructive feedback.

6. References

Abraham CL, Maas S a, Weiss J a, Ellis BJ, Peters CL, Anderson AE. 2013 A new discrete element analysis method for predicting hip joint contact stresses. J Biomech [Internet]. [cited 2014 Nov 26]; 46:1121–7. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3623562&tool=pmcentrez&rendertype=abstract

Anderson DD, Iyer KS, Segal NA, Lynch JA, Brown TD. 2010 Implementation of Discrete Element Analysis for Subject-Specific, Population-Wide Investigations of Habitual Contact Stress Exposure. J Appl Biomech. 26:215–223. [PubMed: 20498493]

Arbabi E, Boulic R, Thalmann D. 2009 Fast collision detection methods for joint surfaces. J Biomech. 42:91–99. [PubMed: 19062019]

Arnold EM, Ward SR, Lieber RL, Delp SL. 2010 A Model of the Lower Limb for Analysis of Human Movement. Ann Biomed Eng [Internet]. 38:269–279. Available from: http://link.springer.com/10.1007/s10439-009-9852-5

Arnold EM, Ward SR, Lieber RL, Delp SL. 2010 A Model of the Lower Limb for Analysis of Human Movement. Ann Biomed Eng. 38:269–279. [PubMed: 19957039]

Arvo J, Kirk D. 1989 A survey of ray tracing acceleration techniques In: An Introd to Ray Tracing. [place unknown]; p. 201–262.

Bei Y, Fregly BJ. 2004 Multibody dynamic simulation of knee contact mechanics. Med Eng Phys [Internet]. [cited 2014 Jan 31]; 26:777–89. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1680082&tool=pmcentrez&rendertype=abstract

Blankevoort L, Huiskes R. 1991 Ligament-Bone Interaction in a Three-Dimensional Model of the Knee. J Biomech Eng. 113:263–269. [PubMed: 1921352]

Caruntu DI, Hefzy MS. 2004 3-D Anatomically Based Dynamic Modeling of the Human Knee to Include Tibio-Femoral and Patello-Femoral Joints. J Biomech Eng [Internet]. [cited 2015 Jan 14]; 126:44 Available from: http://biomechanical.asmedigitalcollection.asme.org/article.aspx?articleid=1411236

Delp S, Loan J, Hoy M. 1990 An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. Biomed … [Internet]. [cited 2012 Oct 20]; 37:757–767. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=102791

Delp S, Loan J. 1995 A graphics-based software system to develop and analyze models of musculoskeletal structures. Comput Biol Med [Internet]. [cited 2013 Jan 10]; 25:21–34. Available from: http://www.sciencedirect.com/science/article/pii/001048259598882E

Elias JJ, Cosgarea AJ. 2007 Computational modeling: an alternative approach for investigating patellofemoral mechanics. Sports Med Arthrosc. 15:89–94. [PubMed: 17505324]

Elias JJ, Wilson DR, Adamson R, Cosgarea AJ. 2004 Evaluation of a computational model used to predict the patellofemoral contact pressure distribution. J Biomech [Internet]. [cited 2015 Feb 20]; 37:295–302. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0021929003003063

Faustini MC, Neptune RR, Crawford RH. 2006 The quasi-static response of compliant prosthetic sockets for transtibial amputees using finite element methods. Med Eng Phys. 28:114–121. [PubMed: 15941666]

Gottschalk S, Lin MC, Manocha D. 1996 OBBTree: A Hierarchical Structure for Rapid Interference Detection. Proc 23rd Annu Conf Comput Graph Interact Tech [Internet]:171–180. Available from: http://portal.acm.org/citation.cfm?doid=237170.237244

Guess TM, Liu H, Bhashyam S, Thiagarajan G. 2013 A multibody knee model with discrete cartilage prediction of tibio-femoral contact mechanics. Comput Methods Biomech Biomed Engin [Internet]. [cited 2014 Jan 28]; 16:256–70. Available from: http://www.ncbi.nlm.nih.gov/pubmed/21970765

Guess TM, Stylianou AP, Kia M. 2014 Concurrent prediction of muscle and tibiofemoral contact forces during treadmill gait. J Biomech Eng [Internet]. 136:021032 Available from: http://www.ncbi.nlm.nih.gov/pubmed/24389997

Guess TM, Thiagarajan G, Kia M, Mishra M. 2010 A subject specific multibody model of the knee with menisci. Med Eng Phys [Internet]. [cited 2014 Jan 30]; 32:505–15. Available from: http://www.ncbi.nlm.nih.gov/pubmed/20359933

Hahn JK. 1988 Realistic animation of rigid bodies. ACM SIGGRAPH Comput Graph [Internet]. 22:299–308. Available from: http://portal.acm.org/citation.cfm?doid=378456.378530

Halloran JP, Ackermann M, Erdemir A, van den Bogert AJ. 2010 Concurrent musculoskeletal dynamics and finite element analysis predicts altered gait patterns to reduce foot tissue loading. J Biomech [Internet]. 43:2810–5. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2946980&tool=pmcentrez&rendertype=abstract

Halloran JP, Erdemir A, van den Bogert AJ. 2009 Adaptive Surrogate Modeling for Efficient Coupling of Musculoskeletal Control and Tissue Deformation Models. J Biomech Eng. 131:011014. [PubMed: 19045930]

Haraguchi N, Armiger RS, Myerson MS, Campbell JT, Chao EYS. 2009 Prediction of three-dimensional contact stress and ligament tension in the ankle during stance determined from computational modeling. Foot ankle Int. 30:177–185. [PubMed: 19254515]

Larsen E, Gottchalk S, Lin M, Manocha D. 2014 PQP - A Proximity Query Package. :http://gamma.cs.unc.edu/SSV/.

Lauterbach C, Mo Q, Manocha D. 2010 GProximity: Hierarchical GPU-based operations for collision and distance queries. In: Comput Graph Forum. Vol. 29 [place unknown]; p. 419–428.

Lauterbach C, Mo Q, Manocha D. 2014 gProximity. :http://gamma.cs.unc.edu/GPUCOL/.

Lee WCC, Zhang M. 2007 Using computational simulation to aid in the prediction of socket fit: A preliminary study. Med Eng Phys [Internet]. 29:923–929. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1350453306001925

Lenhart RL, Kaiser J, Smith CR, Thelen DG. 2015 Prediction and Validation of Load-Dependent Behavior of the Tibiofemoral and Patellofemoral Joints During Movement. Ann Biomed Eng [Internet]. Available from: http://link.springer.com/10.1007/s10439-015-1326-3

Lin M, Gottschalk S. 1998 Collision detection between geometric models: A survey. Proc IMA Conf Math Surfaces [Internet].:1–20. Available from: http://users.soe.ucsc.edu/~pang/161/w06/notes/cms98.pdf

Liu F, Kozanek M, Hosseini A, Van de Velde SK, Gill TJ, Rubash HE, Li G. 2010 In vivo tibiofemoral cartilage deformation during the stance phase of gait. J Biomech [Internet]. [cited 2014 Feb 10]; 43:658–65. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2823844&tool=pmcentrez&rendertype=abstract

Neptune R, Wright I, van den Bogert AJ. 2000 A method for numerical simulation of single limb ground contact events: application to heel-toe running. Comput Methods Biomech Biomed Engin [Internet]. [cited 2012 Jun 19]; 3:321–334. Available from: http://www.me.utexas.edu/~neptune/Papers/cmbbe3(4).pdf

Nickolls J, Buck I, Garland M, Skadron K. 2008 Scalable parallel programming with CUDA. Queue. 6:40.

Smith CR, Lenhart RL, Thelen DG, Kaiser J, Vignos MF. 2015 Influence of Ligament Properties on Tibiofemoral Mechanics in Walking. 1.

Thelen DG, Anderson FC. 2006 Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. J Biomech [Internet]. [cited 2012 Oct 4]; 39:1107–15. Available from: http://www.ncbi.nlm.nih.gov/pubmed/16023125

Thelen DG, Choi KW, Schmitz AM. 2014 Co-Simulation of Neuromuscular Dynamics and Knee Mechanics during Human Walking. J Biomech Eng [Internet]. [cited 2014 Feb 24]; 136:1–8. Available from: http://www.ncbi.nlm.nih.gov/pubmed/24390129

Williams A, Barrus S, Morley RK, Shirley P. 2005 An Efficient and Robust Ray-Box Intersection Algorithm. ACM SIGGRAPH 2005 Courses.:9.

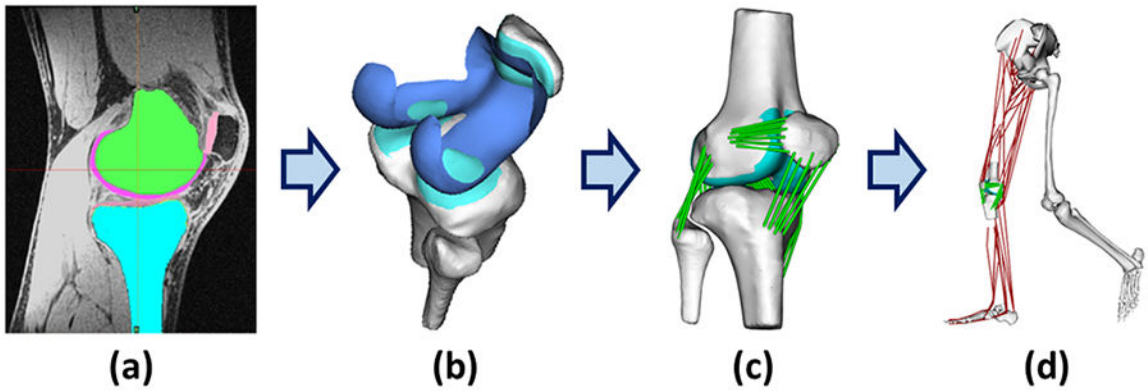

Figure 1.

Model development: a) Cartilage, bone and ligament geometries were manually segmented from MR images b) Bone and cartilage geometries were converted to triangulated surface meshes c) Ligaments were represented as bundles of nonlinear springs spanning from origin to insertion d) The knee model was integrated into a generic lower extremity model(Edith M Arnold et al. 2010).

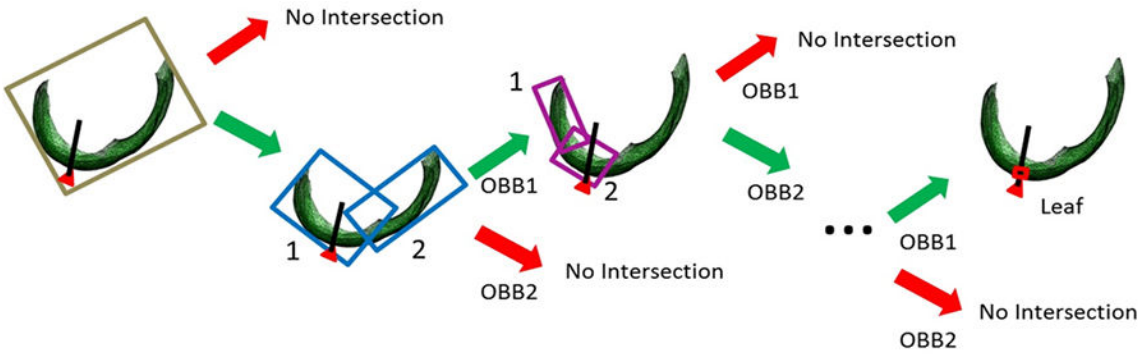

Figure 2.

Collision Detection: A ray is cast normal to each triangle in the *contact* mesh. A ray-intersection test is performed for each level in the oriented bounding box (OBB) hierarchy until a leaf node is reached and the contacting triangle in the *target* mesh is determined

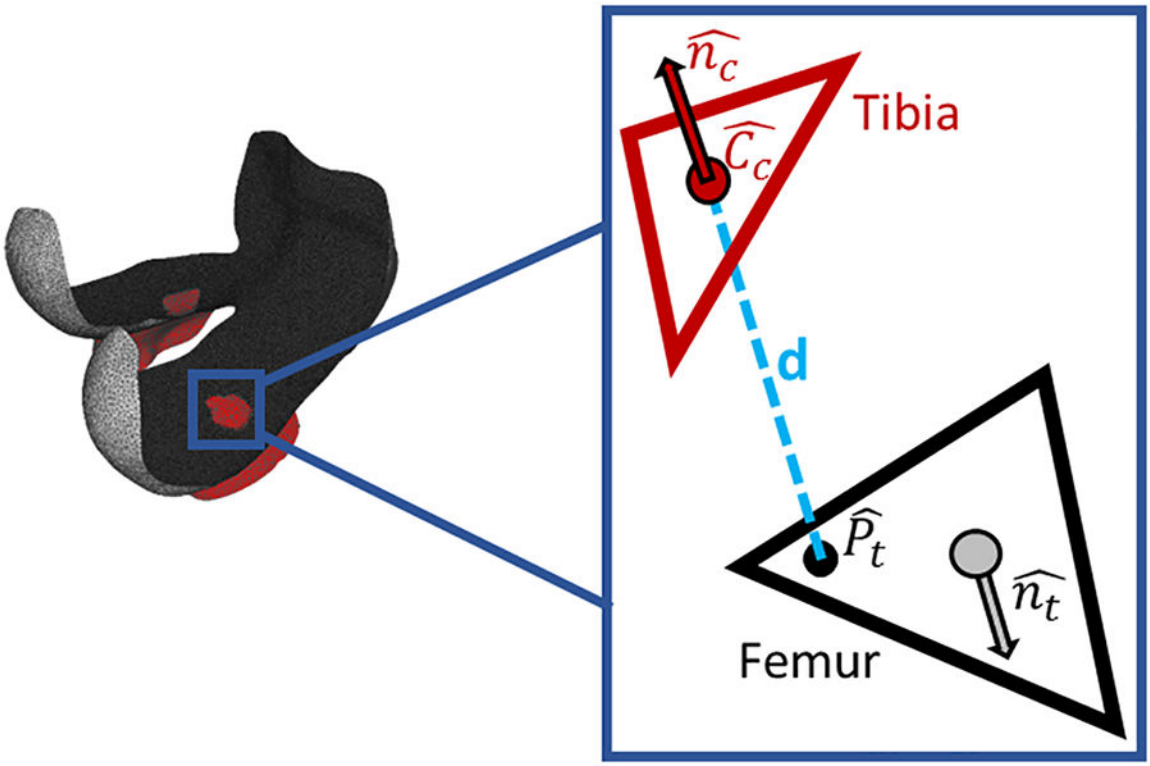

Figure 3.

Penetration Depth: The local depth of penetration (d) is defined as the normal distance from the center of a *contact* mesh face to the point of intersection with the *target* mesh.

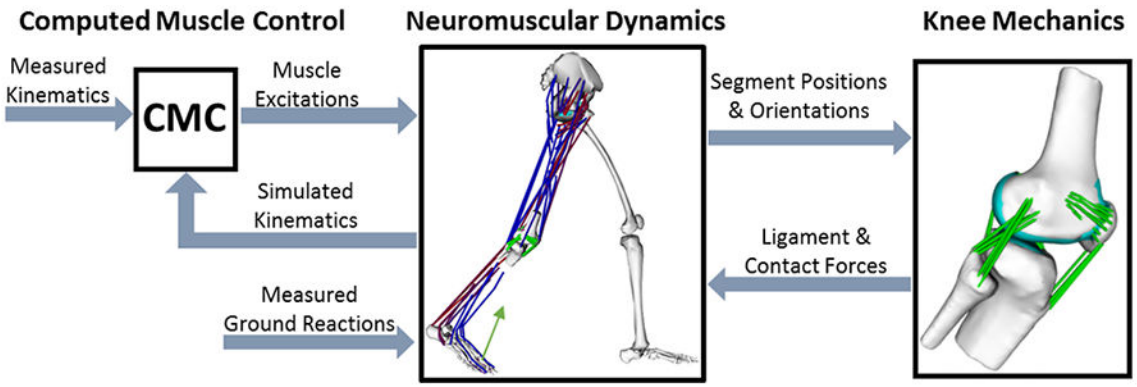

Figure 4.

Neuromuscular Simulation: A computed muscle control (CMC) algorithm was used to modulate the lower limb muscle excitations such that the simulation closely tracked the measured hip, knee, and ankle angles. At every time step, the tibia, patella, and femur positions were used to ascertain the tibiofemoral and patellofemoral contact and ligament forces. These forces were then applied within the forward dynamic simulation of the neuromusculoskeletal model.

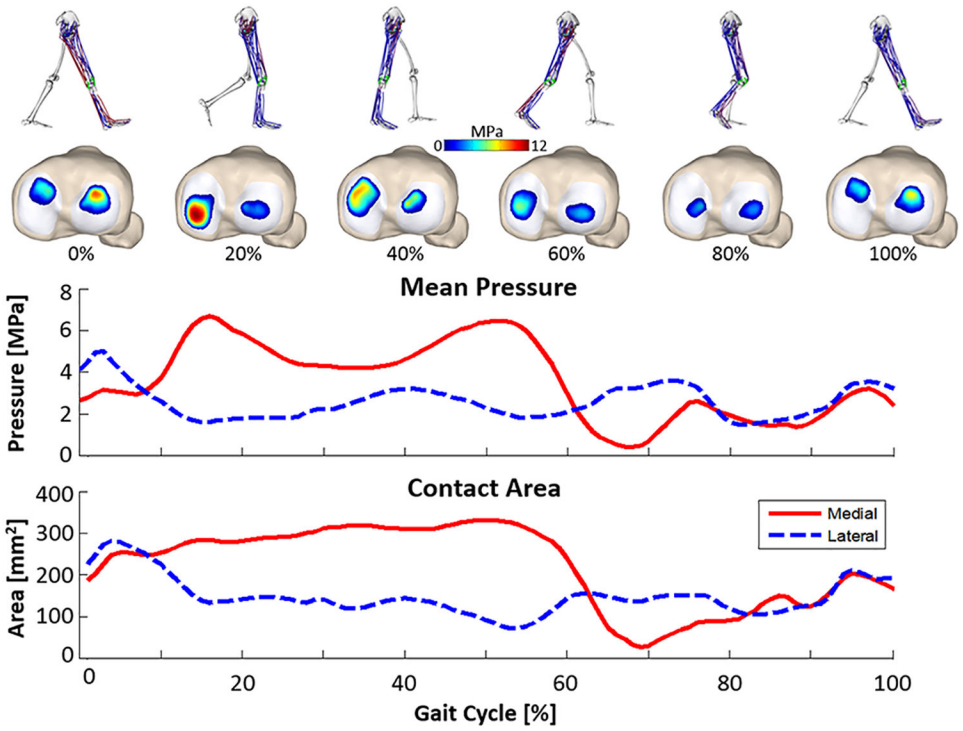

Figure 5.

Tibiofemoral Contact: Simulated cartilage contact pressure and area on the medial and lateral compartments of the tibial plateau over a gait cycle.

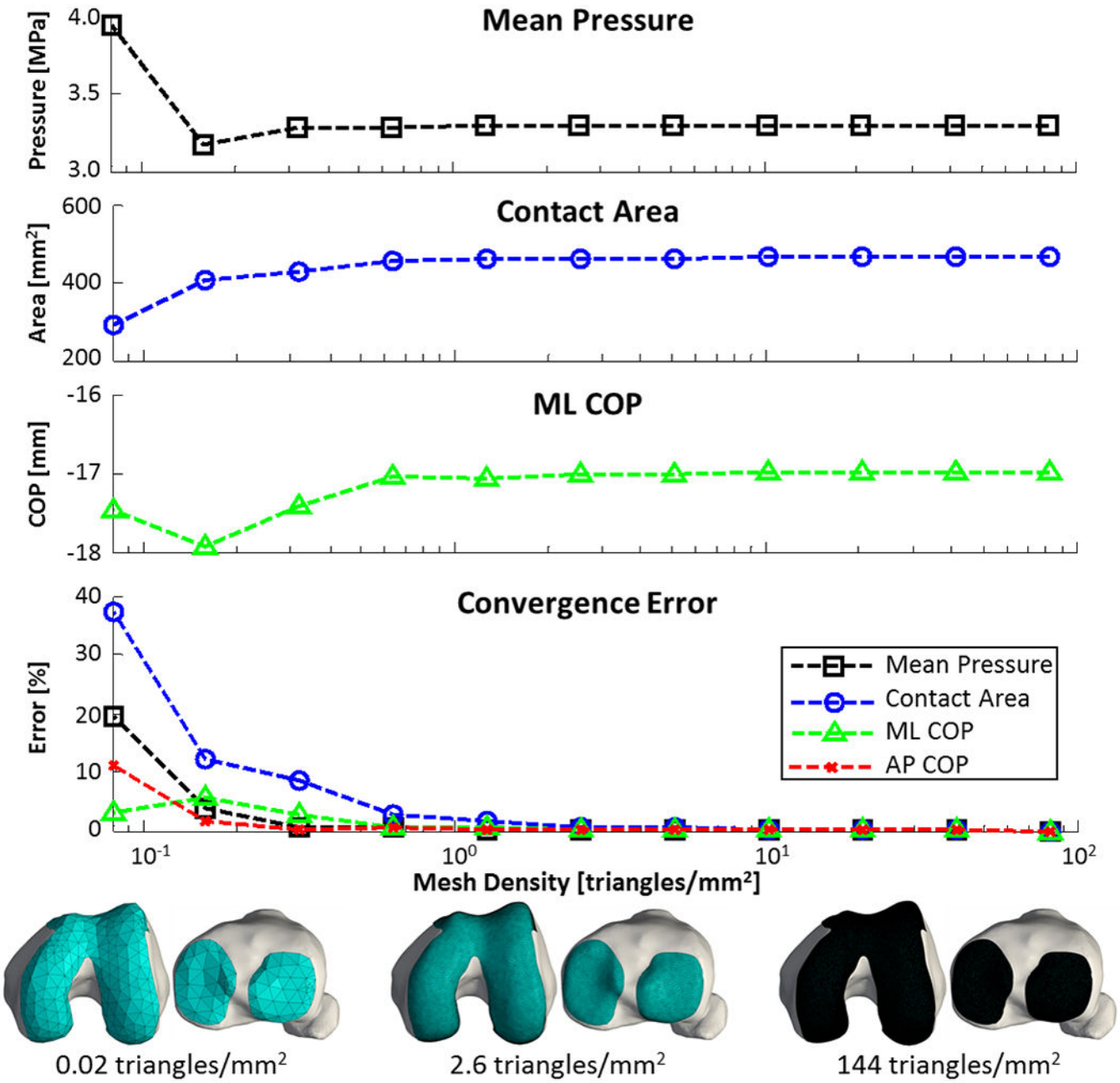

Figure 6.

Mesh Convergence: Tibiofemoral pressure calculations were repeated at a single pose with mesh densities varying from 0.02 to 144 triangles/mm^2. Predicted pressure, area and centre of pressure (COP) values were within 1% of the converged value at a mesh density of 2.6 triangles/mm^2.

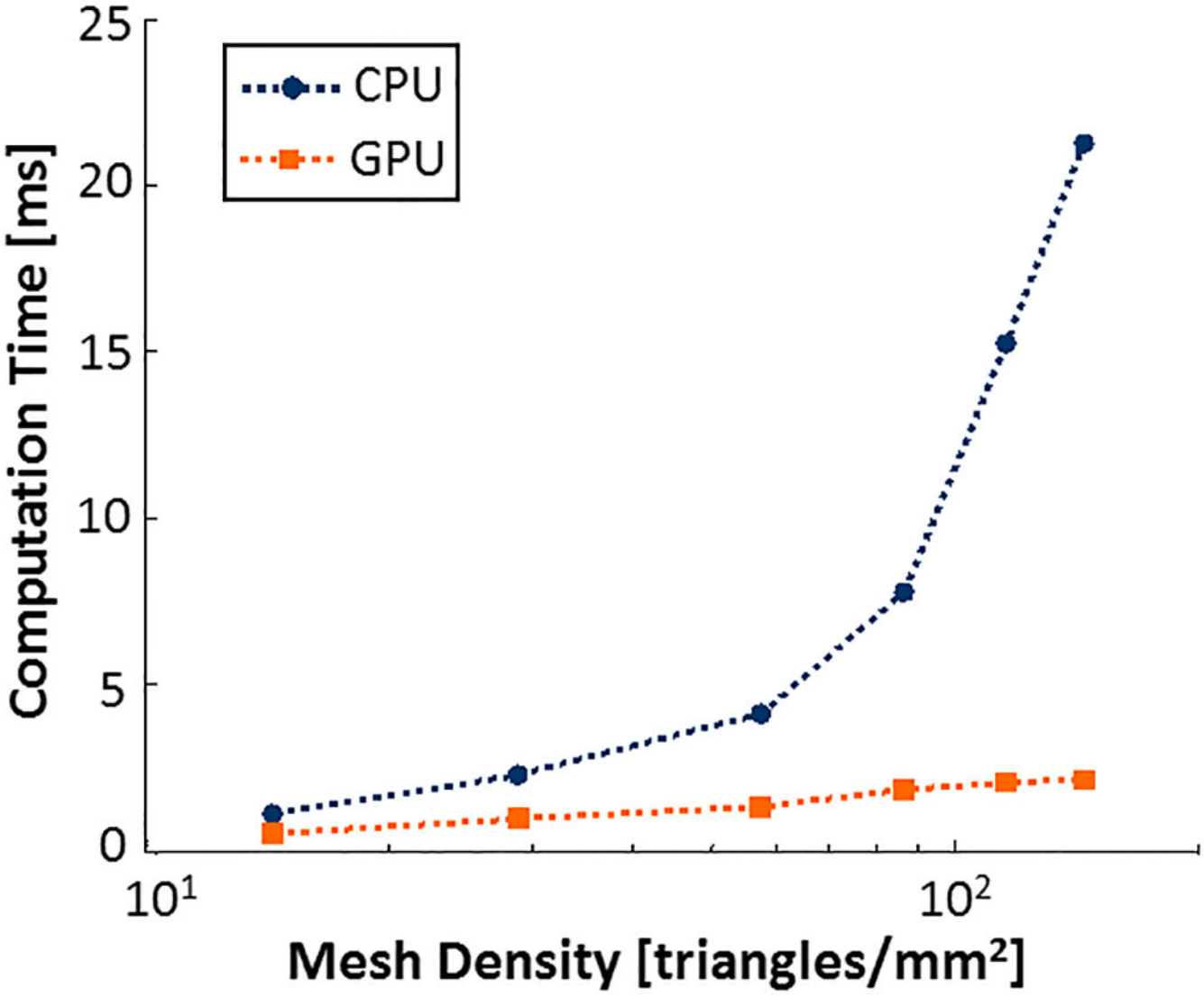

Figure 7.

Computation Time: A comparison of computation times for tibiofemoral pressure calculation at a single pose (2nd peak of tibiofemoral loading) for the CPU and GPU implementations of the collision detection algorithm. GPU times include the transfer time between the main memory in the host and the global memory in the GPU. CPU and GPU computation times were similar for cartilage mesh densities lower than ~10 triangles/mm^2.