

Collision detection and response approaches for computer muscle modelling

Ondrej Havlicek
*Department of Computer Science
 and Engineering*
*Faculty of Applied Sciences,
 University of West Bohemia*
 Pilsen, Czech Republic
 0000-0002-6944-7084

Martin Cervenka
*Department of Computer Science
 and Engineering*
*Faculty of Applied Sciences,
 University of West Bohemia*
 Pilsen, Czech Republic
 0000-0001-9625-1872

Josef Kohout
*NTIS - New Technologies
 for the Information Society*
*Faculty of Applied Sciences,
 University of West Bohemia*
 Pilsen, Czech Republic
 0000-0002-3231-2573

Abstract—Computer muscle modelling is used for many purposes, from injury recovery and treatment of chronic diseases to disease prediction. These predictions often involve computing the muscle’s internal forces to determine further how fast something may happen (e.g. how quickly the muscle joint wears out). During the simulation of such a model, collisions of soft and rigid bodies inevitably occur. This paper tests various state-of-the-art collision handling methods: voxelisation, one using Signed Distance Fields and one based on Bounding Volume Hierarchies. These methods are tested in the context of muscle modelling with the previously proposed position-based dynamics approach. Compared to the other options, using the Discregrid library for Signed Distance Field generation shows the best results, mainly due to its accuracy to the speed of execution ratio. In contrast to the current system, visually pleasant improvements are significant.

Index Terms—Collision detection, Discregrid, Signed Distance Fields, Fast Collision Library, Voxelization, Muscle modelling, Position-based dynamics

I. INTRODUCTION

With diseases such as osteoporosis (prevalence up to 34.3% for females 50 years old or more in the USA, and about 10% among the average population [1]), where bone density decreases or osteoarthritis disease, where there may even emerge the need for joint replacement due to bone structure degeneration, the desire for a realistic musculoskeletal model arises. Such a model could be used to estimate various forces acting around the muscles and bones, which then may be used for prediction and prevention of the named diseases and many more.

Some state-of-the-art models (e.g. [2], [3]) use a generic model (from cadaveric studies or measured on completely different patients and edited with lengthy and exhaustive manual labour). Realistically, the human body varies greatly; with this diversity, the need for patient-specific models becomes increasingly apparent. This presumption leads to a new method, a statistical model. An example would be a statistical model built from 26 patients using Principal Component Analysis created

The authors would like to thank their colleagues and students at the University of West Bohemia for their discussions and suggestions. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic, project SGS-2022-015 New Methods for Medical, Spatial and Communication Data.

by [4]. Another personalized model based on Position-Based Dynamics (PBD) was presented by [5]. This particular model exhibits promising results in terms of model simplicity, speed and accuracy for biomechanical studies, in comparison with musculoskeletal models used commonly in practice. However, in some cases, e.g. extreme flexion around the hip joint, the muscles in this model behave unrealistically. The authors suggest it might be due to the used collision handling system.

The aim of our research, therefore, was to propose a new collision detection (CD) and response (CR) mechanism that would behave adequately even in various extreme scenarios.

In this paper, we present the results of our analysis of the current CD & CR mechanism used in the PBD muscle modelling by [5], and propose two principally different approaches for CD and a couple of minor improvements for CR. The proposed method surpasses the former approach (based on voxelisation) in accuracy, mainly around the problematic hip joint area where muscles no longer get unrealistically stuck.

II. POSITION BASED DYNAMICS APPROACH

In the PBD framework presented by [5], a scene exists consisting of a set of bones, each of which is represented by a triangular mesh surface and has an associated time-dependent transformation describing its movement, and a muscle, also represented by a triangular mesh surface. Each vertex of the muscle is interpreted as a PBD primitive node, having associated mass and velocity, and a set of constraints restricting the freedom of the movement of these points during the simulation. The constraints represent external forces acting on the muscle, including gravitation and fixation to a bone attachment area, as well as internal forces, including local shape and volume preservation.

Each of these constraints can be interpreted as a cost function, resulting in a nonlinear system of differential equations, for which the PBD tries to find their global optimum in the sense of gradient descent using an iterative Gauss-Seidel solver [6]. In each iteration, every PBD node (i.e., a vertex of the triangular muscle surface) is moved from its original position \mathbf{x} to a position \mathbf{p} , satisfying the constraints. This new position may end up inside a bone, which means a collision has occur-

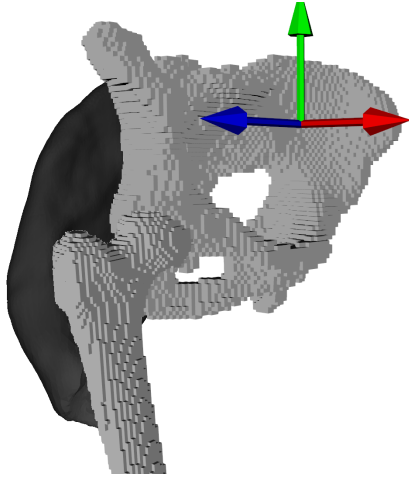


Fig. 1. An example of the voxelisation structure in the pelvic area [5].

red. These collisions need to be detected and resolved after each solver iteration, i.e. the new position \mathbf{p} of each node gets corrected to a noncolliding position \mathbf{p}' .

A. Collision handling

The collision handling in the work [5] was done by creating a voxelised model (a uniform grid) of the bones, as illustrated in Fig. 1. According to the authors, the reasons for choosing this approach were its simplicity and feasible time requirements.

The CD had been designed to further satisfy the run time and memory needs via leaving the bone collision models in their initial rest-pose, while the nodes to test for collisions after each solver iteration would get inversely transformed to the rest-pose coordinate system, then checked whether or not they are contained in any of the bone voxels. If so, then a collision would get detected and the CR process would begin. The result of the CR would then get transformed back to the visual model coordinates. Otherwise, no collision would occur.

The CR is a process where given voxel size a , the position \mathbf{p} of the colliding point is incrementally set by the distance of a towards the original position \mathbf{x} , until it is no longer contained in any of the bone voxels and therefore not colliding. The final position is the denoted position \mathbf{p}' . The process of CD & CR can be seen in Fig. 2.

III. PROBLEMS IDENTIFIED

A. Tunnelling effect

The main issue to consider is that the discrete bone movement may be too fast to simulate the behaviour properly. Consider a femur bone performing the flexion. When it rotates about just 2° (a typical step in simulations), the displacement of the distal part of this bone is nearly 3 cm (considering the average length of the bone being 41.61 cm [7]). Such a displacement might result in muscle penetrating the bone to the other side, as illustrated in Fig. 3.

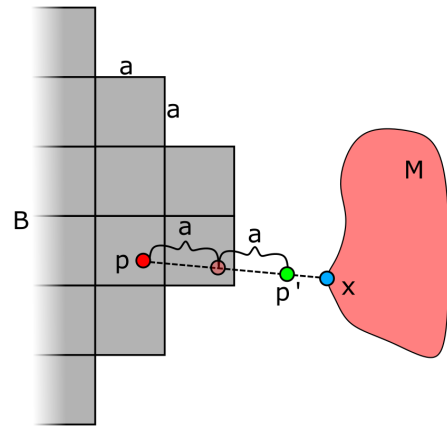


Fig. 2. Collision response mechanism of the voxelisation approach: the point is moving from the desired position \mathbf{p} towards its initial position \mathbf{x} using the constant step of the voxel size a until it reaches free space (position \mathbf{p}').

B. Unavoidable collisions

Due to modelling inaccuracies, the geometrical models of bones and muscles typically slightly penetrate even at the beginning of the simulation, most often at the places where the muscle is connected to the bone. In the original approach, therefore, when the point collides even in its initial position \mathbf{x} , its "noncolliding" position \mathbf{p}' is obtained by transforming \mathbf{x} using the same transformation that was applied to the bone with which the point collides.

The problem is that the voxelisation approach falsely identifies the muscle points closer to the bone surface than the voxel size a as colliding. As a result, these points are bound to the bone and move with it even though they should move freely. For example, the central part of the iliacus muscle is close to the femur head and, therefore, is bound to this bone. During hip flexion, the consequence is dire: a part of the muscle is wound into the free space in the hip joint.

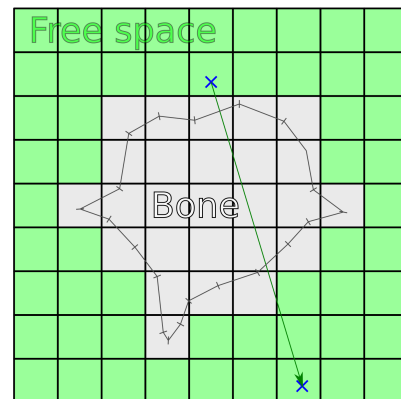


Fig. 3. The muscle vertex (in blue) may penetrate the bone all the way through, because of the big displacement due to the angular motion of the bone.

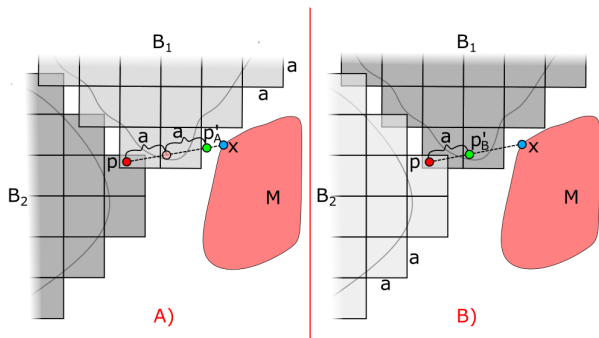


Fig. 4. Collision of a muscle with two different bones. The problem is the collision response as far as both bones force the vertex to move to a different position.

C. Voxelisation problems

Another unsolved problem is a collision with multiple bones simultaneously (see Fig. 4). One of the bones forces the muscle vertex to be in p'_A on left and the second one forces the vertex to the other position p'_B . Applying these CRs consecutively would then lead to only the latter one of them being resolved, which would sometimes leave the point in a collision.

The last problem with voxelisation is that the muscle vertex may get stuck in a gap in between the voxels. As shown in Fig. 5, the blue vertex may move only upwards, but the side-to-side motion is suppressed unless a much bigger force is applied, forcing the blue vertex out. The consequence of the issue is an unrealistic laggy movement and possibly the addition to the problem of muscle wounding in the joint.

IV. PROPOSED METHODS

Considering the simplicity of the voxelisation method, the urge for a more sophisticated collision handling system manifests itself through some of the described problems.

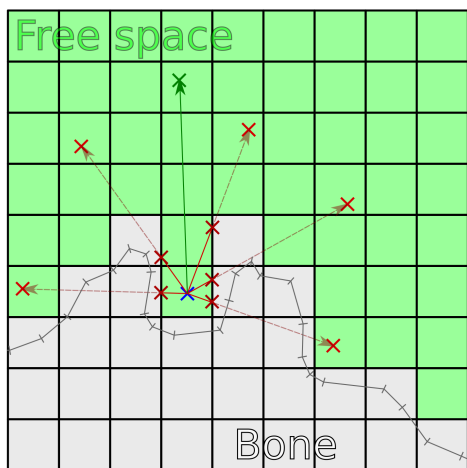


Fig. 5. The blue muscle vertex is stuck in between the voxels occupied by a bone. To get the vertex out sideways, a large force has to be applied.

The general problem seems to be that the voxelisation CR receives on the input only the information about the original and the colliding position of a given node and a rough approximation of the bone surface. Therefore, we propose two different CD & CR approaches, which both provide broader information about the detected collision state, allowing the CR to be more reflective of the underlying physical reality and represent the bone more accurately.

A. Discregrid

Discregrid library is a C++ library based on a Signed Distance Field generated for a bounded finite subspace, allowing to tell for any point in space (x, y, z) the shortest distance and direction to the given triangular surface residing in that subspace. Moreover, the sign of the result adds additional information if the point is inside or outside the surface.

For the method to work, the bounded finite subspace is firstly discretized [8] into a user-defined resolution cuboid grid where each cuboid is a Serendipity type with 32 nodes [9]. At each node, the shortest path to a given surface, as well as the sign, are computed as described by [10], who addresses the problem of discontinuity of a mesh at the edges and the vertices by defining an adequate pseudo-normal for them based on the surrounding angles.

Once the distance, direction and sign are known at every node of the grid, this set of values can be quickly estimated at any point inside any of the cuboids using interpolation by cubic Lagrange polynomials [9].

A collision is detected if the interpolation of a vertex position has a negative sign. This collision can then easily be resolved by pushing the vertex along the shortest distance direction to the surface by this distance instead of the direction it came from.

B. Flexible Collision Library

Flexible Collision Library (FCL) is a CD library written in C++ programming language, providing multiple CD approaches, such as convex polytope-based CD, bounding volume hierarchy (BVH) CD, continuous CD, broad-phase CD, point cloud CD and parallel CD with proximity computation. For speedup, the Sweep and Prune approach over BVH has been used. The library can also handle eight basic shapes: general triangle meshes, convex triangle meshes, spheres, AABB cuboids, cones, cylinders, ellipsoids and capsules.

The library provides so-called managers, which are objects taking care of updating the built structures and detecting collisions among them. The collision information also contains the directional vector between the centroids of the colliding basic shapes, which can be further used for CR.

The dynamic AABB tree collision manager has been used for testing purposes. The bounding box parallel to the euclidean space axes encapsulates each primitive of the object (e.g. the triangle in triangular mesh), which can be checked much faster if it collides with another primitive. The hierarchical structure of these AABB boxes forms a tree, which allows for even faster CD.

C. Collision with multiple bones

Assuming the improved accuracy of the collision models with the proposed methods, collisions with multiple bones should be extremely rare as the bone collision models should not overlap in any way.

Still, in the case of Discregrid, it is possible to virtually add margin to the collision models in hopes to prevent even edge collisions, which could eventually result in this case. For that case, a naive approach of adding the given gradients together was implemented.

V. EXPERIMENTAL RESULTS

The CD approaches have been tested using an existing PBD library originating in [5] on the LHDH dataset [11]. This dataset was chosen because it contains the most refined surface triangle meshes of bones and muscles. The advantage is also its public availability. All nonmanifold edges, degenerate triangles and duplicated vertices and the smoothing of all muscle and bone models have also been done by [11] using MeshLab [12]. Dissection data from [13] are also included, containing muscle attachment areas and geometrical paths of superficial fibres.

For the sake of testing, four muscle models have been used: *adductor brevis*, *gluteus maximus*, *gluteus medius* and *iliacus*. Their vertices count are 502, 9878, 5313 and 6931, respectively.

The experiments were conducted on CPU Core-i5-7200U 2.5GHz, GPU NVIDIA GeForce 930MX, Windows 10, and were compiled in C++ release mode.

A. Visual comparison

The visual comparison between all three CD algorithms shows a clear difference. The Discregrid and FCL are superior, resolving the problem of incorrect muscle shape near joints, which is problematic for the voxelisation approach (see Sections III-B and III-C). To observe the difference, see Fig. 6. We note that the FCL provides (not shown in the figure) an even smoother surface than Discregrid, which is not surprising considering that it works directly with the triangular mesh of the bone.

B. Run time

The execution time shows that Discregrid and FCL are slower than the voxelisation approach since these methods are more complicated. As shown in Fig. 7, the FCL is performing poorly, requiring about one second to detect the collisions. However, we note that the FCL approach has been tested using a single thread only, and multi-threading could improve the performance.

The voxelisation has been originally proposed to ensure real-time simulation (at least 30 FPS on a common PC). The slowdown of the voxelisation means that in the same setup, it would read 6 FPS, which is not precisely real-time but can either be rendered directly with the visible delays or precalculated quickly and rendered as a video for other purposes. FCL is slower (approximately 1 frame per 7 seconds), but still applicable.

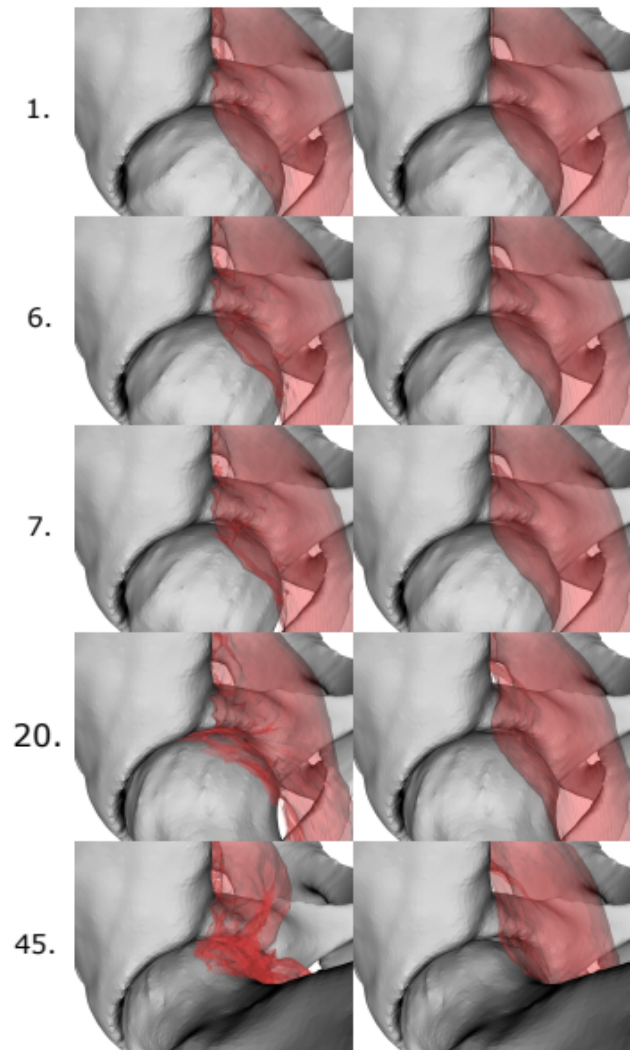


Fig. 6. Iliacus muscle deformation in detail. The voxelisation approach (on the left) shreds the surrounding muscle tissue into the hip joint. Discregrid (on the right) has only a slightly rough surface near the joint. The number on the left denotes the number of the time frame in the simulation of flexion.

One of the main differences between the methods is that in the voxelisation and Discregrid approaches, no update of the collision model has to be made. The collision models are also built only for the rigid bones which can then be queried for arbitrary points whether it collides or not and even the collision information at a constant time.

With the FCL library, BVH collision models have to be built for every object in the scene (bones and muscles). In contrast with the prior methods, these collision models need to be updated each iteration - both for the bones in case of rigid translation and for the muscles due to likely geometry deformation. Furthermore, no arbitrary point CD seems to be supported by the management system. As a result, the collision information has to be post-processed to become usable for CR, which hinders this method's time effectiveness.

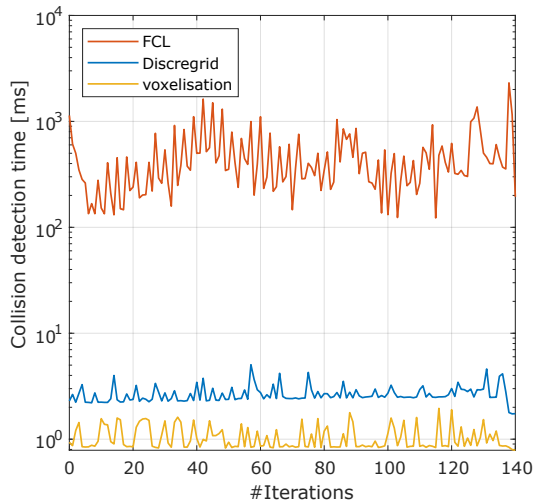


Fig. 7. Run time of collision detection algorithms on the iliacus muscle. The voxelisation is the fastest to execute in general. The Discregrid is approximately 3x slower and the FCL is more than 200x slower on average.

C. Memory consumption

The memory consumption while using the iliacus muscle is shown in Fig. 8. The smallest amount of memory has been used by the original voxelisation algorithm, followed by the Discregrid and FCL algorithms.

Table I shows the results for each tested muscle. All of the consumed memory has been measured using the Visual Studio 2019 16.11.7 Performance Profiler before the first PBD iteration (after the CD structure allocation) and the last PBD iteration (after the whole simulation).

As can be seen in Fig. 8, the different approaches mainly differ in the amount of memory allocated for the collision structures in frame 1. The rise of the memory in the graph reflects the memory consumed by the deformation process while the different CD & CR algorithms used do not seem to have a great impact on the memory used during the course

TABLE I
THE MEMORY USED FOR DIFFERENT CD & CR ALGORITHMS WHILE THE MUSCLE IS DEFORMING. ALL MEASUREMENTS ARE IN MB.

Muscle	First iteration	Last iteration
Voxelization		
Adductor brevis	389	534
Gluteus maximus	526	735
Gluteus medius	458	660
Iliacus	537	688
Discregrid		
Adductor brevis	540	740
Gluteus maximus	677	888
Gluteus medius	609	810
Iliacus	632	839
FCL		
Adductor brevis	685	885
Gluteus maximus	835	1048
Gluteus medius	761	962
Iliacus	786	991

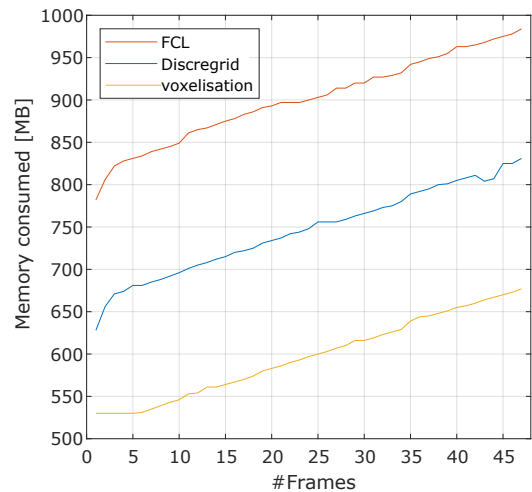


Fig. 8. Memory consumption of all of the algorithms when *iliacus* muscle is used. The voxelisation required the least amount of data, but the Discregrid and FCL seem to be in the same memory complexity level (shifted up only by a constant).

of the simulation. The small deviations in fact represent the number of collisions detected and resolved.

VI. DISCUSSION

The mentioned CD and CR approaches vary significantly in the underlining data structure. The grid approaches (Discregrid and voxelisation) can be simply set to the desired accuracy (changing the grid resolution) to increase the computational speed. This can be done similarly in FCL by allowing various approximations or modifying the narrow-phase solvers, which is also a place for run time improvement of the FCL method in the future since, in this paper, only the default FCL setup has been used.

One big advantage of the FCL is that its hierarchies are designed to be updated during runtime, allowing for CD and CR even among muscles, which could be deformed at any time during the simulation. This is generally hard to achieve using the grid approaches with respect to run time requirements, making them feasible for handling collisions against rigid bones only. This is where the FCL approach could be taken advantage of in the future, e.g. only handling the collisions among muscles. This would, however, result in a heterogeneous CD and CR, increasing the intricacy of the solution, which may be harder to maintain or extend.

On the other hand, Discregrid seems a good successor to voxelisation since it provides feasible accuracy and more information about collision state can be used for CR. CR in the direction of the bone surface results in a more realistic behaviour, resembling the effect of the muscle sliding against the bone surface. This does not change the fact that bone is discretized, which is mitigated to some degree by the Signed Distance Field interpolation, providing better accuracy than voxelisation while running relatively fast compared to the single-threaded FCL implementation.

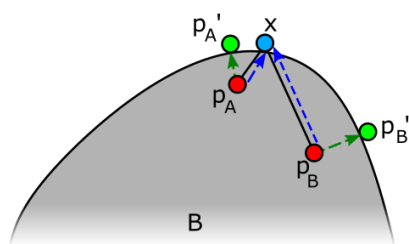


Fig. 9. Simplified illustration of the Discregrid approach respecting the PBD constraints more than the voxelisation approach.

In comparison to the voxelisation approach, where CR tries to revert the colliding point back to the presumably noncolliding original position, the Discregrid approach could be argued to respect the PBD constraints more as illustrated in Fig. 9, where from position x , PBD may move the point either to position p_A or p_B (in red). Voxelisation CR (in blue) tries to return the point toward the original position, reducing the two different PBD constraint results into one point. Discregrid CR (in green) adjust the PBD positions more respectfully (points p'_A and p'_B), merely projecting them onto the bone surface.

Additionally, as previously pointed out, FCL can run multi-threaded, which is currently not viable in our PBD implementation. This is also a possibility for future research.

A. Tunnelling effect

From the Fig. 9, it is easy to imagine that the CR could cause the point to travel to the other side of the bone since the shortest path to the bone surface would reside in that direction. This would cause the same situation as illustrated in Fig. 3. This problem is not directly addressed in this paper, but one possible solution could be to use Discregrid once again and get the gradients of both the former and final positions x and p' , respectively. Then it could be stated that if the angle between them is bigger than some set angle e.g. 135° , the point p' ended up on the other side of the bone and thus the tunnelling occurs. If that happens, the final position could be set to an intersection of the bone surface with the path defined by the points x and p .

B. Unavoidable collisions

To overcome the roughness of voxelisation discretisation, FCL can be utilised since it does not omit any information and does CD at the level of the actual bone triangles and not their grid approximation. Nevertheless, due to the impractical running times of the currently implemented FCL solution, it cannot be used. Instead, the Discregrid results display enough accuracy in the iliacus muscle case and therefore seem a reasonable compromise overall.

Because of the better CD accuracy and the fact, that the method provides a gradient to the bone surface, there is no longer a need for the colliding points to follow the bone transform in case the bone moves into the muscle, as this transform is already encompassed by the gradient.

C. Voxelisation problems

Furthermore, the problem of the muscle being stuck exposed in Fig. 5 has not been fully solved in this paper but is to an acceptable extent mitigated by the greater accuracy of the proposed methods.

VII. CONCLUSION

This paper presented the analysis of CD & CR system used by [5] and proposed one new CD approach based on Signed Distance Fields and another one based on Bounding Volume Hierarchies to overcome the shortcomings of a former method in the context of a state-of-the-art PBD muscle modelling approach. Moreover, slight modifications to CR were made to introduce more realistic muscle sliding behaviour. The Discregrid approach, based on the Signed Distance Fields, proved to be a suitable trade-off between improved accuracy and time requirements.

REFERENCES

- [1] S. W. Wade, C. Strader, L. A. Fitzpatrick, M. S. Anthony, and C. D. O'Malley, "Estimating prevalence of osteoporosis: examples from industrialized countries," *Archives of Osteoporosis*, vol. 9, no. 1, p. 182, May 2014. [Online]. Available: <https://doi.org/10.1007/s11657-014-0182-3>
- [2] S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp, and J. M. Rosen, "An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 8, pp. 757–767, Aug 1990.
- [3] E. Arnold, S. Ward, R. Lieber, and S. Delp, "A model of the lower limb for analysis of human movement," *Annals of biomedical engineering*, vol. 38, pp. 269–79, 12 2009.
- [4] J. Zhang, J. Fernandez, J. Hislop-Jambrich, and T. F. Besier, "Lower limb estimation from sparse landmarks using an articulated shape model," *Journal of Biomechanics*, vol. 49, no. 16, pp. 3875 – 3881, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021929016311186>
- [5] J. Kohout and M. Červenka, "Muscle deformation using position based dynamics," in *Biomedical Engineering Systems and Technologies*, X. Ye, F. Soares, E. De Maria, P. Gómez Vilda, F. Cabitza, A. Fred, and H. Gamboa, Eds. Cham: Springer International Publishing, 2021, pp. 486–509.
- [6] J. Bender, M. Müller, and M. Macklin, "A Survey on Position Based Dynamics," in *EG 2017 - Tutorials*, A. Bousseau and D. Gutierrez, Eds. The Eurographics Association, 2017.
- [7] A. Vasilopoulos, G. Tsoucalas, E. Panagouli, G. Trypsianis, V. Thomaidis, and A. Fiska, "Odontoid process and femur: A novel bond in anatomy," *Cureus*, vol. 12, 03 2020.
- [8] B. Angles, D. Rebain, M. Macklin, B. Wyvill, L. Barthe, J. Lewis, J. V. D. Pahlen, S. Izadi, J. Valentin, S. Bouaziz, and A. Tagliasacchi, "VIPER," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 2, no. 2, pp. 1–26, jul 2019.
- [9] D. Koschier and J. Bender, "Density maps for improved sph boundary handling," in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, ser. SCA '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: <https://doi.org/10.1145/3099564.3099565>
- [10] A. Bærentzen and H. Aanæs, "Generating signed distance fields from triangle meshes," *Technical University of Denmark*, 01 2002.
- [11] M. Viceconti, G. Clapworthy, and S. V. S. Jan, "The virtual physiological human - a european initiative for in silico human modelling," *The Journal of Physiological Sciences*, vol. 58, no. 7, pp. 441–446, 2008.
- [12] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: an open-source mesh processing tool," *Computing*, vol. 1, pp. 129–136, 01 2008.
- [13] S. Van Sint Jan, "Introducing anatomical and physiological accuracy in computerized anthropometry for increasing the clinical usefulness of modeling systems," *Critical Reviews in Physical and Rehabilitation Medicine*, vol. 17, pp. 149–174, 01 2005.