Contact Modeling and Collision Detection in Human Joints

THÈSE N^{O} 4421 (2009)

PRÉSENTÉE LE 3 JUIN 2009
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE RÉALITÉ VIRTUELLE
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Ehsan ARBABI

acceptée sur proposition du jury:

Prof. M. Hasler, président du jury
Prof. D. Thalmann, directeur de thèse
Prof. H. Bleuler, rapporteur
Dr H. Delingette, rapporteur
Dr S. J. Ferguson, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2009

To my parents
Azim & Marzieh

Acknowledgments

First, I would like to express my gratitude to Professor Daniel Thalmann for providing me the opportunity of working in his laboratory, Dr Ronan Boulic for his supervision and scientific supports, and Professor Nadia Magnenat-Thalmann for her project leadership.

I would like to thank all my colleagues in VRLab. Special thanks to Schubert Ribeiro de Carvalho for his continuous friendship and helps, to Helena Grillon for her kind helps in providing the *résumé*, to Dr Anderson Maciel for his initial scientific supports, to Olivier Renault for his technical helps, and Josiane Bottarelli for her constant aids.

Also thanks to all my colleagues in the Co-Me project (MIRALab, ARTORG and the other places), especially Salman Chegini, Caecilia Charbonnier, Lazhari Assassi, Jerome Schmid, Dr Benjamin Gilles, Dr MyungJin Kang, Dr Pascal Volino, Dr Gwenael Guillard, Dr Laurent Moccozet, Dr Stephen Ferguson, and Dr Moritz Tannast. I also thank Swiss National Science Foundation for the financial support during Co-Me project.

I am grateful to all my friends from different nationalities, with whom I enjoyed my stay in Switzerland.

Finally, my very deep thanks to my mother *Marzieh*, my father *Azim*, and my brothers *Amin* and *Aryan*, who have been always supporting me in different ways.

Abstract

Collision detection among virtual objects is one of the main concerns in virtual reality and computer graphics. Usually the methods developed for collision detection are for either very general cases or very specific applications. The first main goal of this thesis is to propose accurate methods for collision detection in computer graphics for rotating or sliding objects. The methods take advantage of the limitation imposed on the rotating/sliding objects in order to ignore unnecessary calculations of the general methods and speed up the processing. In addition to finding the collision, the methods can also return penetration depths in either radial or cylindrical direction, which can be useful for further applications.

The second main goal is to apply the proposed collision detection methods in biomedical research related to human hip joints. In fact, during the past few years, femoroacetabular impingement (FAI) was recognized as the leading pathomechanism contributing to a significant number of so-called “primary” hip osteoarthritis. Thus, having medical simulation of hip joint can help both physicians and surgeons for better diagnosis and surgical planning. For diagnosing some of the human joint diseases, it is important to obtain the joint’s range of motion. By modifying the pre-processing stage of one of the collision detection methods, a new fast method for finding maximum range of motion in human joint was proposed and tested. The method is working without doing any collision detection tests and its accuracy does not depend on the rotational steps.

We also suggested a novel fast strategy for diagnosing hip diseases based on hip contact penetration depths. In this strategy, the contact penetration depths during hip movement are calculated for diagnosing hip impingements, by using the proposed collision detection methods. The strategy has been tested on pathological hip models during a daily activity. The results were found correlated with the contact stresses estimated by finite element method (FEM). By evaluating the results, the strategy proved to be capable for distinguishing among different hip pathologies (e.g. cam and pincer impingements).

In orthopedic simulations, the behavior of the bones and the related tissues are usually investigated during their movements about an estimated center of rotation. We also evaluated the importance of the hip joint center of rotation in medical simulations. For this reason, different centers of rotation calculated by five different methods were applied for hip movements about different medical axes of rotation. By calculating the hip contact penetration depths of ten patients during hip movements (using the proposed collision detection methods), the sensitivity of hip simulations to hip center of rotation has been evaluated.

Hip contact pressure has been a notable parameter to evaluate the physical conditions inside the hip joint. Many computational approaches estimate the pressure and contact pressures via finite element methods (FEM) by using 3D meshes of the tissues. Although this type of simulation can provide a good evaluation of hip problems, the process may be very time consuming. Also, these mechanical methods strongly depend on the movement details. We proposed and tested a fast statistical model for estimating hip contact pressures during its movement, without performing mechanical simulations and without any need for movement details. The estimation is done by evaluating geometric features extracted from 3D meshes of hip tissues, in order to link an unknown target hip model to some already mechanically evaluated training hip models.

Keywords: Computer graphics; collision detection; penetration depth; rotating and sliding objects; human joints; hip; femoroacetabular impingement (FAI); geometric features.

Résumé

La détection de collisions entre objets virtuels est essentielle en réalité virtuelle et dans le domaine de l'infographie. Généralement, les méthodes développées pour la détection de collisions adressent soit des cas très généraux soit des applications très spécifiques. Le premier objectif majeur de cette thèse est de proposer des méthodes précises de détection de collisions pour faire tourner ou glisser des objets. Ces méthodes tirent avantage des limitations imposées sur l'objet en rotation/glissement afin d'ignorer les calculs superflus des méthodes générales, permettant ainsi d'accélérer le traitement. En plus de détecter les collisions, ces méthodes peuvent aussi déterminer les profondeurs de pénétration selon les directions radiale ou cylindrique, ce qui peut être d'un très grand intérêt pour certaines applications futures.

Le deuxième objectif majeur est d'appliquer les méthodes de détection de collisions proposées dans le domaine de la recherche biomédicale sur l'articulation de la hanche humaine. Ces dernières années, le conflit femoro acétabulaire (FAI) a été reconnu comme étant le pathomécanisme principal contribuant à un nombre non-négligeable d'osteoarthrites dites «primaires» de la hanche. Afin de diagnostiquer certaines maladies articulaires humaines, il est important d'obtenir la portée du mouvement de l'articulation. En modifiant l'étape de prétraitement d'une des méthodes de détection de collisions proposées, une nouvelle méthode rapide permettant de trouver la portée maximale de mouvement d'une articulation humaine a été énoncée et testée. Cette méthode fonctionne sans avoir besoin de faire de tests de détection de collisions et sa précision ne dépend pas des pas de rotation.

Nous avons également proposé une stratégie novatrice et rapide, basée sur le contact de la hanche et les profondeurs de pénétration, afin de diagnostiquer certaines maladies de la hanche. Dans cette stratégie, les profondeurs de pénétration de contact pendant un mouvement de la hanche sont calculées afin de diagnostiquer les conflits de la hanche en utilisant les méthodes de détection de collisions proposées. Cette stratégie a été testée sur des modèles de hanche pathologiques durant des activités journalières. Nous avons découvert que les résultats étaient corrélés avec les stress de contact estimés par méthode d'éléments finis (FEM). En évaluant les résultats, cette stratégie s'est avérée être capable de distinguer entre différentes pathologies (ex: conflits came et tenaille).

Nous avons également évalué l'importance du centre de rotation de la hanche dans les simulations médicales. Pour cette raison, différents centres de rotation, calculés par cinq méthodes distinctes ont été appliqués pour les mouvements de la hanche autour de différents axes de rotation médicaux. En calculant les profondeurs de pénétration de contact de la hanche en mouvement de dix patients (en utilisant les méthodes de détection de collisions proposées), la sensibilité des simulations de la hanche au centre de rotation de la hanche a été évaluée.

La pression de contact de la hanche a été un paramètre notable afin d'évaluer les conditions physiques dans l'articulation de la hanche. Nous avons proposé et testé une méthode statistique rapide afin d'estimer les pressions de contact de la hanche lors de son mouvement sans devoir faire de simulation mécanique et sans avoir besoin des détails de mouvement. Cette estimation est faite en évaluant des spécificités géométriques extraites des maillages 3D des tissus de la hanche afin de pouvoir relier un modèle ciblé de hanche inconnu à des modèles de hanche utilises dans une phase d'apprentissage.

Mots clé: Infographie; détection de collisions; profondeur de pénétration; objets en rotation et glissement; articulations humaines; hanche; conflit femoro acétabulaire (FAI); spécificités géométriques.

Contents

Chapter 1 Introduction 8

1.1 Collision detection in computer graphics 8

1.2 Medical simulations for hip joint 9

1.3 Objective of the research and the approach 10

1.4 Contributions 11

1.4.1 Accurate and fast collision detection methods for rotating or sliding objects 11

1.4.2 Accurate and fast joint range finder method 11

1.4.3 New fast strategy for diagnosis hip pathologies 11

1.4.4 Evaluating the sensitivity of hip simulations to the hip center of rotation 11

1.4.5 Fast statistical method for estimating hip contact pressures from hip 3D models ... 12

1.5 Organization of the document 12

Chapter 2 State of the art 13

2.1 Collision detection in computer graphics 13

2.2 Hip joint simulation 18

Chapter 3 Collision detection for rotating or sliding objects 19

3.1 Introduction 19

3.2 Cylindrical segmenting collision detection 20

3.2.1 Overview: a 2D example 20

3.2.2 Method 22

3.2.3 Pre-processing 27

3.2.4 Discretizing resolution 27

3.3 Radial collision 28

3.3.1 Previous methods 28

3.3.2 Weakness of the previous methods 28

3.3.3 Method 30

3.3.4 Pre-processing 37

3.3.5 Table size 38

3.4 Testing and comparison 38

3.4.1 Reference methods for the comparisons 38

3.4.2 Comparison scenarios 39

3.5 Results 41

3.6 Discussion 49

3.6.1 Striking case 50

3.6.2 Sliding case 50

3.6.3 Striking vs. Sliding 52

3.6.4 Complexity 54
3.6.5 Penetration depths 54
3.7 Applications 55
3.7.1 Biomedical 55
3.7.2 Others 58
3.8 Conclusion 61
Chapter 4 Medical research and simulations 62
4.1 Introduction 62
4.2 Joint range finder 63
4.2.1 Introduction 63
4.2.2 Method 64
4.2.3 Biomedical experiment 66
4.2.4 Discussion and conclusion 68
4.3 Using contact penetration for hip medical diagnosis 69
4.3.1 Introduction 69
4.3.2 Method 70
4.3.3 Results 74
4.3.4 Discussion 75
4.3.5 Conclusion 78
4.4 Evaluating importance of the hip joint center of rotation in medical simulations 79
4.4.1 Introduction 79
4.4.2 Method 80
4.4.3 Medical experiment 83
4.4.4 Results 85
4.4.5 Discussion 87
4.4.6 Conclusion 90
4.5 Estimating hip joint contact pressure from hip mesh geometric features 91
4.5.1 Introduction 91
4.5.2 Method 91
4.5.3 Testing the method on different hip models 94
4.5.4 Discussion 95
4.5.5 Conclusion 98
Chapter 5 Conclusion 99
5.1 Summary 99
5.2 Contributions 100
5.2.1 Collision detection for rotating or sliding objects 100
5.2.2 Joint range finder 100
5.2.3 Using contact penetration for hip medical diagnosis 100
5.2.4 Evaluating importance of the hip joint center of rotation in medical simulations . 101

5.2.5 Estimating hip joint contact pressure from hip mesh geometric features 101
5.3 Future works 101
Appendices 103
Appendix A: finding the cylindrical bounding volume of a triangle 103
Appendix B: checking a vertex inside 2D representation of a triangle..................................... 104
Appendix C: calculating the angular distance between a vertex and a polygon 105
Appendix D: checking whether a vertex is penetrating an object in angular direction 106
Appendix E: mapping function 3 - quasi uniform radial segmentation 108
Appendix F: checking whether a vertex is penetrating an object in radial direction 109
Bibliography 111

Chapter 1 Introduction

1.1 Collision detection in computer graphics

One of the challenging topics in virtual reality and computer graphics is collision detection among virtual objects. It is due to the fact that objects of our world undergo diverse reactions when submitted to conditions toward the violation of the non-penetration law (two bodies cannot occupy the same place in space within the same time interval) (Maciel, Boulic and Thalmann 2007). Numerous approaches have been investigated to detect interfering objects in applications such as robotics, computational biology, games, surgery simulation, and cloth simulation. The important point in collision detection is to have both accurate results and high speed of computation. Several methods have been developed for collision detection in order to improve the speed of computation without having any negative effect on the accuracy. But the computational speed is still one of the main concerns, especially in high resolution cases or when the objects are deformable. While many of the original collision detection methods primarily address the problem of rigid bodies, recent approaches have started focusing on deformable objects. Collision detection for deformable objects is an essential component in interactive physically-based simulation and animation which is a rapidly growing research area with an increasing number of interesting applications (Teschner, Kimmerle, et al. 2005).

Usually the methods developed for collision detection are for either very general cases or very specific applications. Although the general methods can be used for all the cases, but due to their generality, many calculations which are not always necessary, must be done that may slow down the total process. On the other hand, the methods for very specific applications are only useful for a limited number of cases. Therefore having some methods designed for certain common cases is demanding in fast computer graphics based applications.

In many applications movements are only or partly rotational, without any major translation. There is a wide range of applications dealing with rotating or sliding objects such as mechanical simulation, joint simulation, robotics, medical applications, etc. Therefore, having a method specifically designed for rotating or sliding objects is useful for a wide range of applications. In such cases, we can take advantage of the limitations imposed due the type of movements, i.e. rotation or sliding, in order to develop a much faster collision detection method compared to the general methods.

1.2 Medical simulations for hip joint

In the past few years, femoroacetabular impingement (FAI) was recognized as the leading pathomechanism contributing to a significant number of so-called “primary” hip osteoarthritis (Ganz, et al. 2003), (Murphy, et al. 2004). It is defined by an early pathological contact between primary osseous prominences of the acetabulum and/or the femoral head-neck junction. Depending on the underlying pathomorphology and its related pathomechanism, two different types of FAI are distinguished: “pincer” and “cam” impingement. “Pincer impingement” describes a linear contact between the acetabular rim and the femoral head-neck junction. The maximum impact force is tangential to the articular surface. This is typically due to focal (e.g. acetabular retroversion) or general overcoverage of the femoral head. A so-called “cam impingement” occurs when the femoral head-neck junction has an abnormally large radius resulting in insufficient offset (Figure 1).

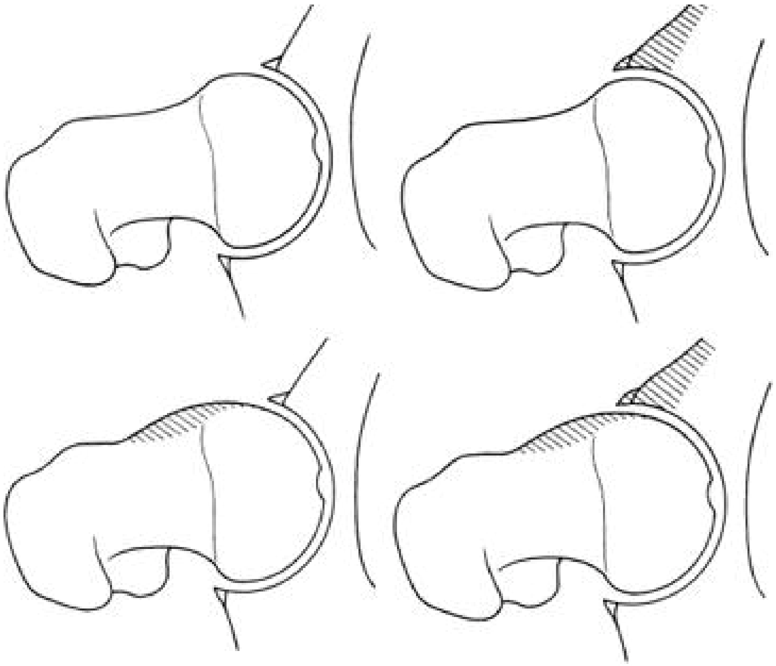

Figure 1: Upper-left: Normal hip. Upper-right: Pincer impingement. Bottom-left: Cam impingement. Bottom-right: Combination of cam and pincer impingement (Lavigne, et al. 2004).

Available options for impingement treatment include trimming of the overcovering portion of the acetabular rim in pincer FAI and surgical resection of the aspherical portion of the femoral head neck junction in cam type of FAI (Mardones, et al. 2005), (Lavigne, et al.

2004). Thus, having medical simulation of hip joint can help both physicians and surgeons for better diagnosis and surgical planning.

1.3 Objective of the research and the approach

Most of the methods for collision detection are for general/semi-general purposes and therefore none of them may be considered as an optimum method. This generality can be problematic in case we need higher speed and still accurate results, especially when the other time consuming computations may be done in parallel to the collision detection. In fact, we may have much faster methods if we could consider the conditions and limitations imposed on the objects.

The first main goal of this thesis is to propose different methods for collision detection among rotating or sliding objects, by considering the objects’ topology, geometrical relations and limitations in order to speed up the process while keeping the accuracy. The methods are compared with some other general methods in different scenarios to find their weakness and strength for different simulations.

The methods should be able to be used in the computer graphics applications where both accuracy and speed are important. For example, the accuracy and speed of the methods should allow them to be used in construction of visual medical applications which aids surgeons and physicians to diagnose a specific set of pathologies in human joints.

By exploiting the fact that the relative location of different entities vary within a limited domain, faster (and still accurate) methods for collision detection can be proposed. These geometrical limitations can be imposed on range of motion and/or type of motion of different objects. Since our goal is to propose collision detection methods for rotating or sliding objects, the limitation related to such kinds of movement should be considered. For instance, the relative motion of femoral capsule in the hip joint is quasi-rotational inside the pelvis; therefore this pre-knowledge of the geometrical relation between femoral capsule and the tissues attached to the pelvis can be used for designing the faster collision detection between them. In such quasi-rotational cases, the methods can be based on radial or cylindrical segmentation of the space for storing the necessary information. In fact, such segmentations can aid the methods to neglect unnecessary examinations and consequently increase the speed of performance while keeping the accuracy.

The topology and the shape of the obejcts, and also the way that they are attached to each other can also provide us some information about the possible and the impossible colliding regions. This knowledge can be again helpful for speeding up the collision. In fact, some objects always keep a limited range of distance from each other and cannot be either closer or farther than this limitation; therefore it is not essential to perform collision detection for some regions of the objects which are always attached or unattached due to their principles.

By combining all of these strategies which are based on movement and geometric limitations of the objects, accurate methods for faster collision detection among rotating and sliding objects can be obtained.

Although the methods can be applied for a wide range of applications in computer graphics, we are mainly going to use them for detecting collision among major deformable or rigid tissues integrated around the hip joints. In fact, our other major goal in this thesis is to exploit the proposed methods for examining contacts among the hip tissues during hip movement. Different biomedical research concerning hip joints is done in this thesis, where the contacts among virtual 3D models of the tissues are evaluated by using the proposed collision detection methods. In addition to the informative side of such biomedical research for helping future medical developments and research, they lead to propose a new strategy for diagnosing some of the hip pathologies.

1.4 Contributions

1.4.1 Accurate and fast collision detection methods for rotating or sliding objects

By taking advantage of the limitations imposed on the rotating and sliding objects, we propose two accurate and fast collision detection methods for rotating or sliding objects. In addition to detecting the collisions faster than general methods, they also return penetration depths in either radial or cylindrical direction. In general, the methods are capable of detecting the collisions and returning the penetration depths in the applications that the movement is modeled as either rotation or sliding. For instance, they can be applied in robotic research or inside human joints to evaluate the contact among deformable and/or rigid parts.

1.4.2 Accurate and fast joint range finder method

By modifying the pre-processing stage of one of the proposed collision detection methods, we propose a fast and accurate method for finding maximum range of motion in joints (e.g. human joints). The proposed joint range finder works without applying successive collision detection algorithms (vs. traditional methods). It needs to be performed only once per simulation to find both anti clockwise and clockwise range of motion. Due to its fast processing, the method can be used effectively in reverse engineering or biomedical applications.

1.4.3 New fast strategy for diagnosis hip pathologies

By applying both of the proposed collision detection methods for hip 3D models, we propose a novel fast strategy for evaluating hip pathologies. The strategy is suitable for real time medical hip simulations and allows an independent differentiation between the subtypes of hip impingement. The method can give some diagnostic information, especially in cases with combined impingement where the major component has to be defined and treated.

1.4.4 Evaluating the sensitivity of hip simulations to the hip center of rotation

We investigate the sensitivity of the penetration depths of hip tissues to the methods applied for estimating hip joint center of rotation (HJC). Different centers of rotation calculated by five methods were applied during hip movement of ten patients. The results of this investigation highlight the importance of the HJC estimation methods because of their influence on computer-aided medical research and diagnosis.

1.4.5 Fast statistical method for estimating hip contact pressures from hip 3D models

We propose a fast statistical model for estimating hip contact pressures during its movement. This model works without performing mechanical simulations and without any need to the movement details. In addition to its fast processing (compared to the slow mechanical methods), its independency from the movement information can be useful when the movement details are either missing or difficult to mathematically characterize in a simulation.

1.5 Organization of the document

The structure of this document is as follows. Chapter 1 gives an introduction to state of the art related to collision detection in computer graphics and also hip simulations. In Chapter 3, two collision detection methods for rotating or sliding objects (cylindrical segmenting collision detection and radial segmenting collision detection) are proposed and explained in details. The methods are also tested and compared with some previous collision detection methods. Some discussions about their usability and applications are done at the end of this chapter. Different medical research and simulations by direct or indirect exploitation of the proposed collision detection methods are done and explained in Chapter 4. The chapter starts with proposing a fast method for finding joint range of motion. Then a new strategy for hip medical diagnosis by using contact penetration is proposed and tested. The third section of this chapter is dedicated to evaluating importance of hip joint center of rotation in medical simulations. In the last section of the chapter a statistical model for estimating hip joint contact pressure from hip mesh geometric features is proposed, tested and evaluated. Chapter 5 provides the conclusion of this thesis by giving the summary of the research and highlighting the contributions. Finally, some appendices at the end of the thesis provide more mathematical details related to the collision detection methods proposed in Chapter 3.

Chapter 2 State of the art

2.1 Collision detection in computer graphics

There are many methods and strategies proposed for collision detection in computer graphics. One of the most famous strategies used for collision detection is applying “Bounding Volume Hierarchies” (BVH). Usually, a BVH is constructed for each object in a pre-processing step. The idea is to partition the set of object primitives (e.g. polygons) recursively until some leaf criterion is met (Figure 2).

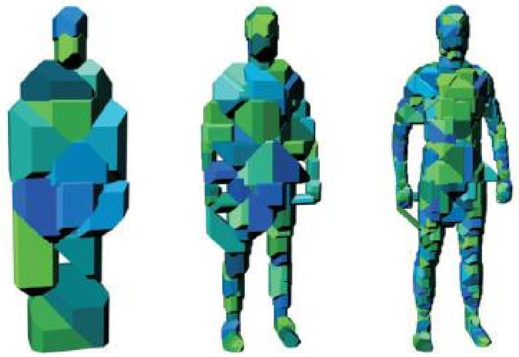

Figure 2: Bounding Volume Hierarchies (Teschner, Kimmerle, et al. 2005).

Different kinds of bounding volume such as AABB[1] (Van Den Bergen 1997), DOP[2] (Zachmann 1998), OBB[3] (Gottschalk, Lin and Manocha 1996), sphere (Palmer and

---

[1] Axis Aligned Bounding Box

[2] Discrete Oriented Polygon

[3] Oriented Bounding Box

Grimsdale 1995), spherical shell (Krishnan, et al. 1998) have been proposed (Figure 3). For collision test, the BVHs are traversed top-down and pairs of tree nodes are recursively tested for overlap. If the overlapping nodes are leaves then the enclosed primitives are tested for intersection (Teschner, Kimmerle, et al. 2005).

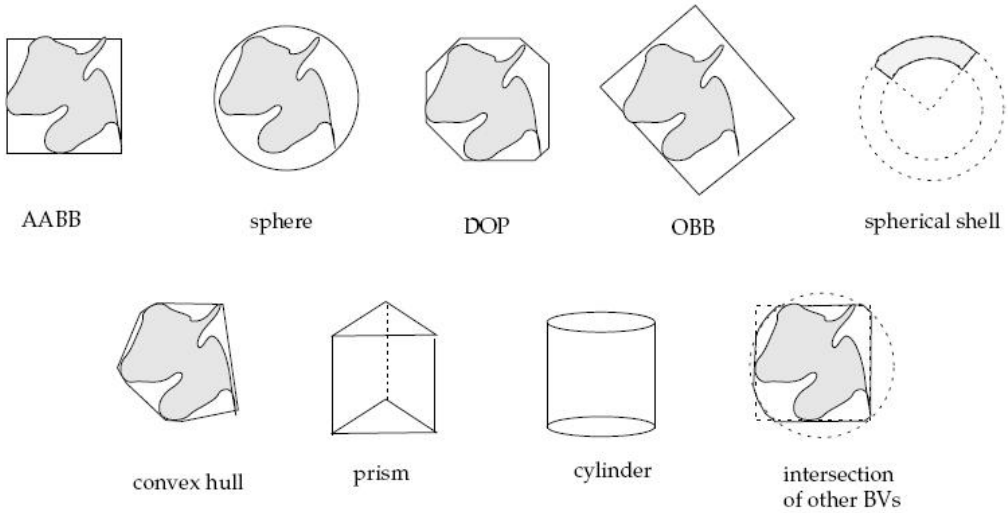

Figure 3: Different Bounding Volumes (Teschner, Kimmerle, et al. 2005).

Figure 4 shows the partitioning and collision testing parts for a hierarchical method based on spherical distance field. In this method the spherical shell bounding volumes at each level of the hierarchical spherical distance field discard the parts in certain sphere sectors from further consideration (Funfzig, Ullrich and Fellner 2006).

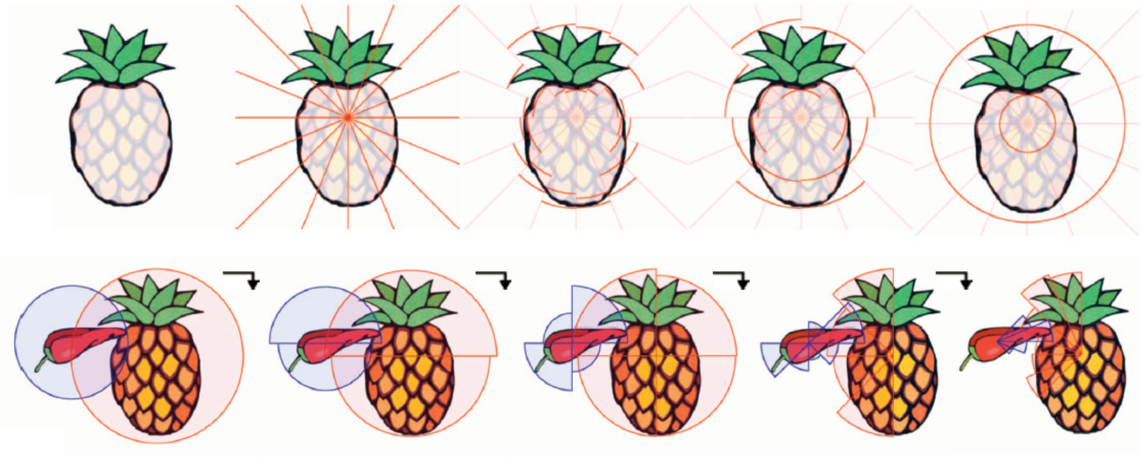

Figure 4: Upper: spherical partitioning of an object. Bottom: testing the collision (Funfzig, Ullrich and Fellner 2006).

There has been some other research related to the methods used for construction and updating BVH. In some cases the bounding volumes are not only covering object primitives at a certain time step. Instead, the volume is described by the linear movement of a primitive within two successive time steps. Such methods can lead to accelerate continuous collision detection. Also BVs are used for detecting self collisions, (Moore and Wilhelms 1998), (Larsson and Akenine-Möller 2001), (Zachmann and Langetepe, Geometric Data Structures for Computer Graphics 2003), (Teschner, Kimmerle, et al. 2005). Applying both BVH and Curvatures tests are used for having exact methods of collision and self-collision detection and avoiding unnecessary self-intersections tests, (Volino and Magnenat-Thalmann 1994).

In some other cases, it is the space which is going to be checked. That means the space is divided into some subspaces and then each subspace is checked for containing two colliding sets of points. There are various approaches proposing spatial subdivision for collision detection. These algorithms employ uniform grids (Turk 1990), octrees (Bandi and Thalmann 1995) or BSP[4] trees (Melax 2000), (Teschner, Kimmerle, et al. 2005). For example, (Teschner, Heidelberger, et al. 2003) employed spatial hashing with a uniform grid for the detection of collisions and self collisions for deformable tetrahedral meshes, (Figure 5).

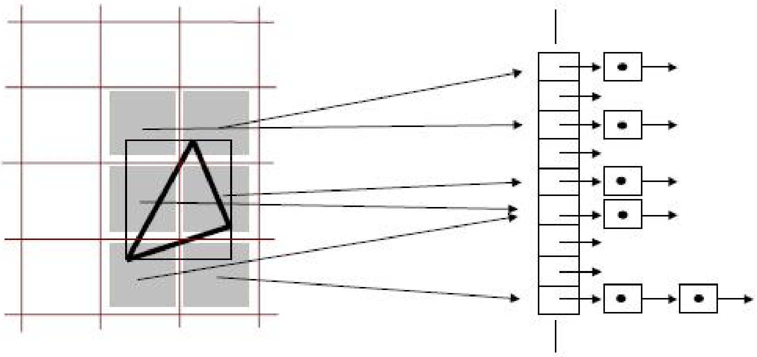

Figure 5: Spatial hashing with a uniform grid; mapping from 3D cell location to1D index by using hash function (Teschner, Heidelberger, et al. 2003).

Distance fields have been used for having faster contact detection too. Distance fields specify the minimum distance to a closed surface for all points in the field (Figure 6). Representing a closed surface by a distance field is advantageous because there are no restrictions about topology. Further, the evaluation of distances and normals needed for collision detection and response is extremely fast and independent of the geometric complexity of the object, (Teschner, Kimmerle, et al. 2005).

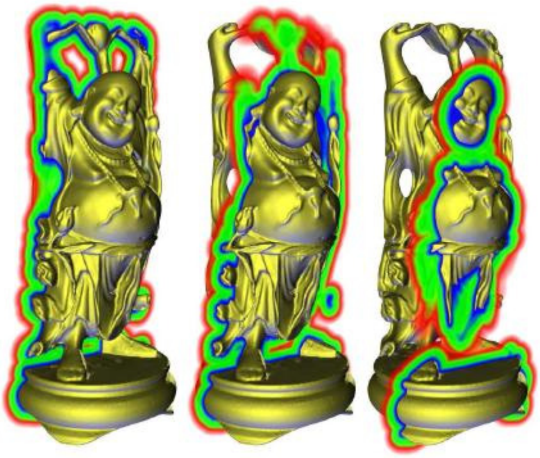

Figure 6: Distance field (Teschner, Kimmerle, et al. 2005).

---

[4] Binary space partitioning

Image-space techniques have been also proposed for collision detection. These approaches commonly process projections of the objects to accelerate collision queries (Figure 7). One advantage in using such methods is that, image-space techniques can be commonly implemented by using graphic hardware, (Baciu, Wong and Sun 1999), (Heidelberger, Teschner and Gross 2003).

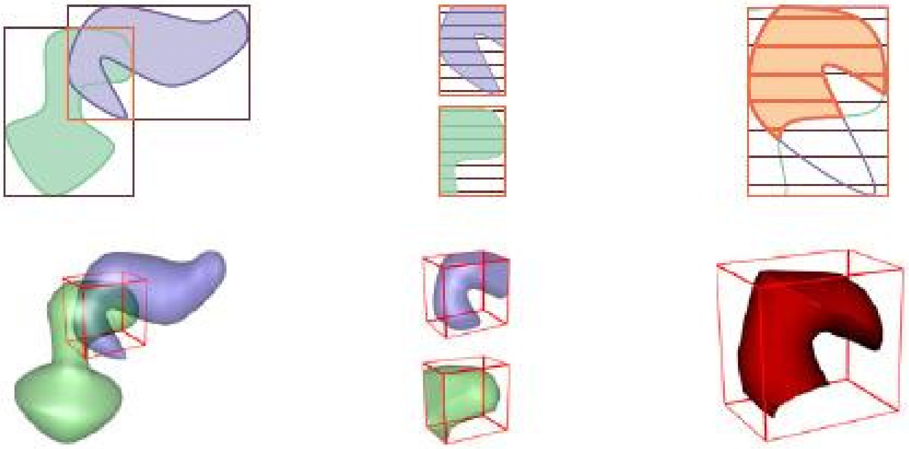

Figure 7: Image-space technique (Heidelberger, Teschner and Gross 2003)

Inexact methods have become a focus in some collision detection research. This idea is motivated by several observations. First, polygonal models are just an approximation of the true geometry. Second, the perceived quality of most interactive 3D applications does not depend on exact simulation, but rather on real-time response to collisions. At the same time, humans cannot distinguish between physically-correct and physically-plausible behavior of objects. Therefore, it can be tolerated to improve the performance of collision detection, while degrading its precision. One of the inexact methods is based on average-case approach. Conceptually, the main idea of this algorithm is to consider sets of polygons at inner nodes of the BVH. During traversal, pairs of these sets of polygons are checked. However, pairs of polygons are never explicitly checked. Therefore, there is no polygon information stored with the nodes of the BVH. Instead, the probability of the existence of a pair of intersecting polygons is estimated. The other inexact method is an algorithm based on randomly selected primitives. It consists of selecting random pairs of colliding features as an initial guess of the potential intersecting regions. The solution for identifying the colliding regions when the object moves or deforms, is to consider temporal coherence. If a pair of features is close enough at a time step, it may still be interesting in the next one. This allows tracking colliding regions over subsequent time steps as the objects are animated, (Lin and Canny 1992), (Klein and Zachmann 2003).

Most of the mentioned methods are for general/semi-general purposes and therefore none of them may be considered as an optimum method for collision detection. In other words, we may have much faster methods if we consider the condition and limitations related to our objects. For example for laparoscopic surgical simulation a method was proposed for collision/self collision detection in intestine which was based on approximating the intestine with a chain of spheres where these spheres, of the same radius, are uniformly distributed along the curve (Figure 8). Therefore, collision detection could be performed by calculating distance between the spheres and/or the other objects (France, et al. 2005). Although such methods can work fast for their related cases, but the problem of such very specific methods is that their usage may be limited to a small number of applications.

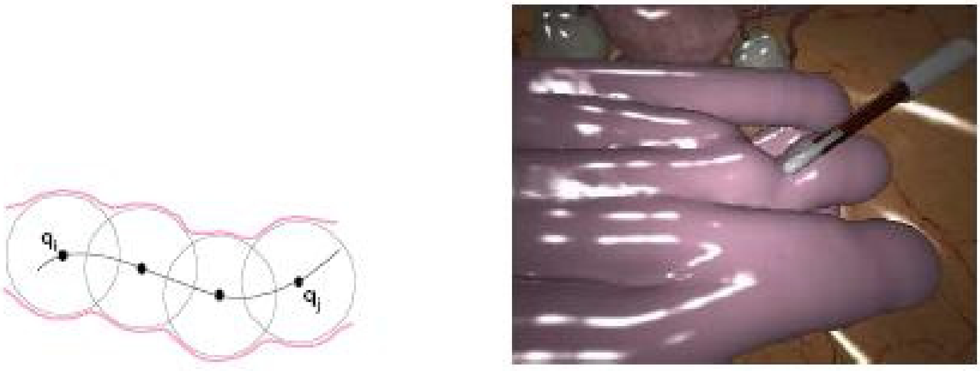

Figure 8: Modeling intestine with spheres for fast collision detection (France, et al. 2005).

On the other hand, we may have some limited conditions which can be valid in many cases; therefore finding an optimum solution for collision detection in such cases can speed up collision detection in many other similar cases too. One of such cases is when the objects mostly have rotational motions. Recently, a method of collision detection was proposed for dealing with situations in which soft structures are in constant but dynamic contact, which is typical of some 3D biological elements. The method proceeds in two stages. First, in a preprocessing stage, a mesh is chosen under certain conditions as a reference mesh and is spherically sampled (Figure 9). In the collision detection stage, the resulting table is exploited for each vertex of the other mesh to obtain, in constant time, its signed distance to the fixed mesh (Maciel, Boulic and Thalmann 2007). This method deals with the permanent contact of deformable objects which leads to a collision detection problem for some applications such as biomedical simulation of the human joint capsules. Although this methods has shown a fast performance in its main processing stage, its results is obtained by an approximation. Also, in case the objects deform in a non radial direction, the pre-processing stage must be repeated. The preprocessing is too slow compared to the main process, due to the algorithms used for creating the tables.

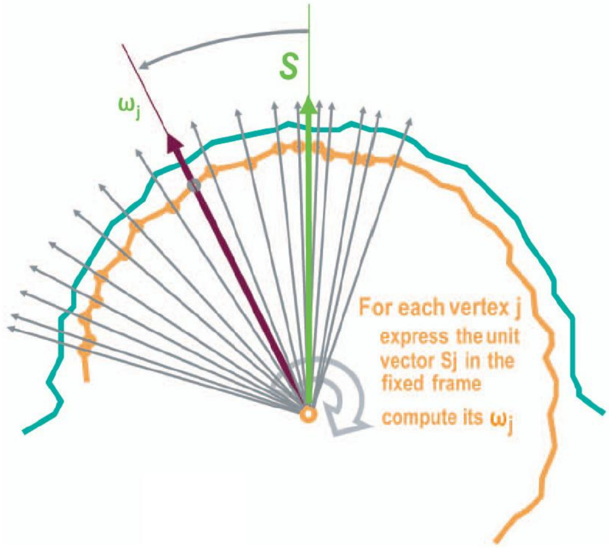

Figure 9: Ray-based sampling method (Maciel, Boulic and Thalmann 2007).

2.2 Hip joint simulation

Using computer aided simulations for investigating the joints behavior in normal and pathological cases (Martin 2005) could be proven to assist physicians to diagnose the illness faster and more accurately, and also to achieve a more precise surgical plan, (Tannast, Kubiak-Langer, et al. 2007), (Arbabi, Boulic and Thalmann 2007), (Chegini, Beck and Ferguson 2006), (Teran, et al. 2005), (Armand, et al. 2004), (Kang, Sadri, et al. 2003), (Scifert, et al. 1998), (Genda, et al. 1995). Human joint simulations usually starts by reconstructing three dimensional meshes of the joint tissues (bones, cartilages, etc) from CT or MR Images (Gilles, Moccozet and Magnenat-Thalmann 2006) and estimating the center of rotation such as for the hip (Kang, Sadri, et al. 2003), (Camomilla, et al. 2006). Once the three dimensional model is constructed, the critical task to handle can be the precise detection of collisions between virtual tissues so that the stresses in the colliding areas are faithfully evaluated (Chegini, Beck and Ferguson 2006), (Scifert, et al. 1998), (Genda, et al. 1995), (Rapperport, Carter and Schurman 1985) or the range of motion in a specific orientation is correctly estimated (Kang, Sadri, et al. 2003), (Armand, et al. 2004), (Arbabi, Boulic and Thalmann 2007).

In computer-based simulations, some models have been primarily concerned with collision detection for osseous joint components during motion (Tannast, Goricki, et al. 2008), (Tannast, Kubiak-Langer, et al. 2007), (Langlotz, et al. 1998). Although collision detection between the bony parts can give useful information about the range of motion (Kubiak-Langer, et al. 2007), it does not take into account the important role which soft tissues play in the joint. Different types of femoroacetabular impingement (FAI) could be shown to greatly deform the acetabular labrum and the adjacent articular cartilage, resulting in different types of labral tears and degenerative chondral damage (Ito, Leunig and Ganz 2004).

High stresses are shown to be in association with soft tissue damage within the hip joint. High stresses may tear the fibers of the soft tissues (particularly the acetabular labrum and the articular cartilage), influence their biological environment, and therefore lead to osteoarthritis. In-vivo or in-vitro measurements of contact pressure within the hip joint have been performed parallel to the development of predictive tools (Michaeli, Murphy and Hipp 1997), (Hodge, et al. 1986). Other computational approaches estimate the stress and contact pressures via finite element methods (Russell, et al. 2006).

Estimating stresses within the soft tissues of the hip joint during the loads and motions of daily activities, using e.g. finite element analysis is a common approach to investigate hip joint pathologies, mainly related to developmental dysplasia of the hip (Chegini, Beck and Ferguson 2008), (Chegini, Beck and Ferguson 2006), (Russell, et al. 2006), (Hipp, et al. 1999), (Michaeli, Murphy and Hipp 1997). Although this type of simulation can provide a good evaluation of static hip problems, having a better estimation of stresses for a dynamic pathomechanism such as FAI is very time-consuming and cannot be applied for real time medical hip simulations. Thus, hip stress estimation is more likely suitable for research purposes rather than direct application in routine real time medical planning. There is a lack of a real time method evaluating penetration depth of the colliding soft and bony tissues in FAI, and which has been correlated with available biomechanical simulation data.

Chapter 3 Collision detection for rotating or sliding objects

3.1 Introduction

In computer graphics, many methods have been proposed to speed up the processing time for collision detection among virtual objects. Usually the methods developed for collision detection are for either very general cases or very specific applications. In this chapter we propose two new methods for collision detection based on finding the penetrating vertices (not edges) when the objects are rotating or sliding, which can be used for a wide range of applications. The methods takes advantage of the limitation imposed on the rotating/sliding objects in order to ignore unnecessary calculations of the general methods and speed up the processing. The main strategy applied in these methods is spatial segmentation in the angular or radial orientations. These kinds of segmentation help the methods to be more adjusted to the style of movement (rotation or sliding), and consequently improve the efficiency of the collision detection for the rotational cases. The methods are tested in different scenarios and compared with some of the previous methods (including general and specific ones), where the comparisons show the efficiency of the proposed ones.

When an object is rotating, two kinds of collision may happen: 1- sliding or 2- striking. In the sliding case the colliding area is almost parallel to the rotational trajectory and the penetration is usually considered in the radial direction. On the other hand, in the striking case, the objects' colliding area is almost perpendicular to the rotational trajectory and the penetration is usually considered in the angular direction (Figure 10). In this chapter the goal is to take advantage of the limitations imposed on the rotating objects, and propose faster collision detection methods for all the rotational cases. Therefore both of the striking and sliding cases are considered and two specific methods for both cases are proposed (cylindrical segmenting and radial segmenting collision detection methods, respectively).

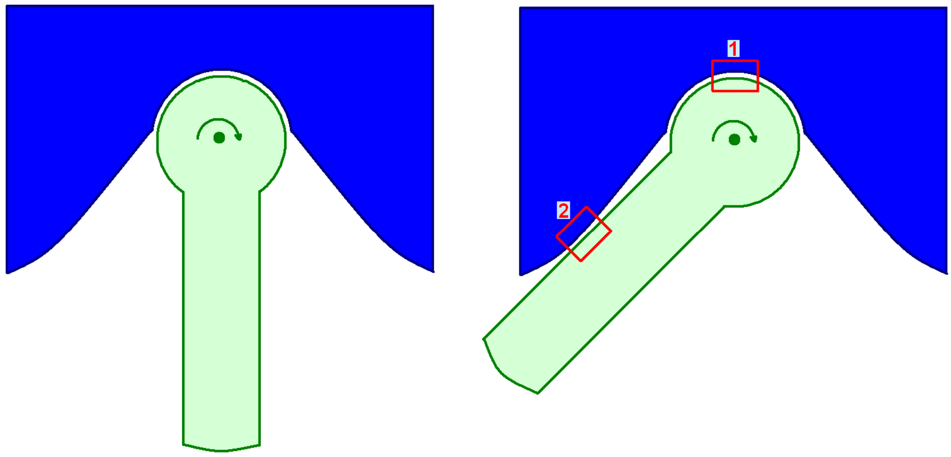

Figure 10: Sliding and striking collision during rotation. The light green object is rotating about its center and can collide with the dark blue object in two different ways: 1-sliding, 2-striking.

3.2 Cylindrical segmenting collision detection

For striking cases there is no specific method and the general collision detection methods are used for finding the collision during rotational strike. Therefore having a specialized method for detecting such kind of collision is missing. We propose a novel fast method suitable for rotational strike based on discretizing the space in the cylindrical orientation. This kind of segmentation helps the method to be more adjusted to the style of movement (rotation), and consequently improve the efficiency of the collision detection for the rotational cases. The cylindrical segmentations are done in the same orientation of rotation which does not only increase the speed of rigid collision detection but also increases the speed of updating for deformable collision detection (in angular direction). The method is not only returning the penetrating mobile vertices, but it also returns the corresponding penetrating fixed triangles in the angular direction, which consequently provides the curvilinear penetration depth of the vertices without applying any additional computations.

3.2.1 Overview: a 2D example

As an example we consider the objects in the xy plane and explain the method in 2D (without losing the generality). The white mobile object rotates around the origin of ‘O’ and may collide with the gray fixed object (see Figure 11.a). At the first step we neglect non colliding parts by calculating distance between the origin and different parts of the objects. Therefore according to Figure 11.b all parts of the objects which are either outside the big red circle or inside the small red circle cannot have any collision and are neglected. In the next step the space is polar segmented (see Figure 11.c). One table is filled for the fixed object where a table cell represents a complete polar segment (i.e. a ring). A table cell stores list of all the edges occupying the corresponding ring (polygons in 3D). Then for each vertex of the mobile object, we find the associated ring and consequently the corresponding table cell for the fixed object is found. We check whether the mobile vertex has any chance to meet any of the fixed edges stored in the found corresponding table-cell.

Among all the edges giving the positive answer for a vertex, we find the one with smallest angular distance to the mobile vertex. Then by using the angular-tangential element of the normal of the found fixed edge (polygons in 3D), it is checked whether the mobile vertex is penetrating the fixed object or not. For example in Figure 11.d, an arbitrary ring is chosen, where the distance between all the green parts in the fixed object with a vertex in the red part of the mobile object must be calculated. The green edge with smallest distance to the red vertex is checked further for penetration. This method is done for all the mobile vertices to find all the colliding pairs.

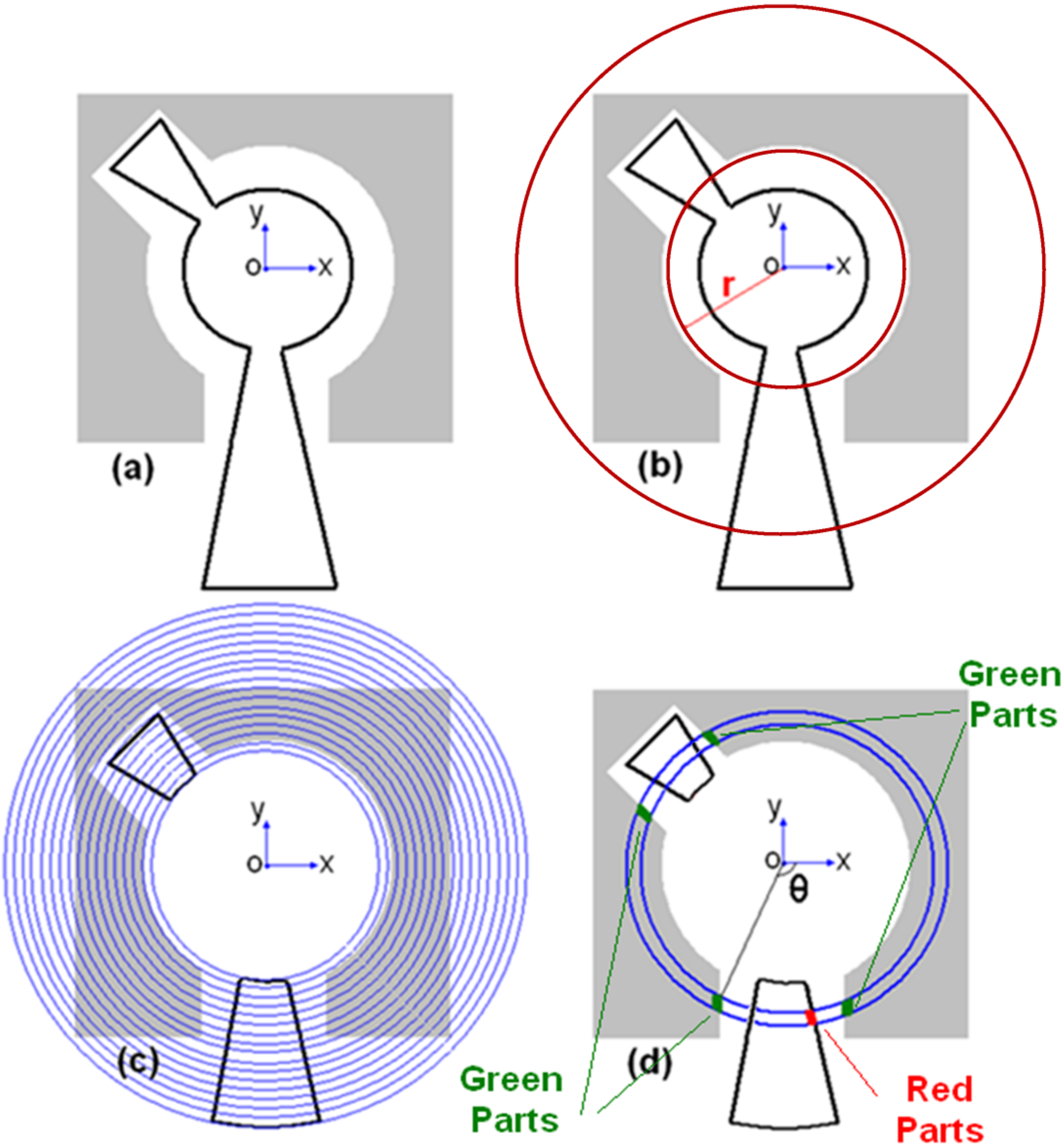

Figure 11: Different steps of the cylindrical collision detection method (striking) for a 2D case; a) two possible colliding objects, b) ignoring non colliding parts, c) polar segmenting the space, d) collision tests.

3.2.2 Method

The method works by segmenting the object spatial occupancy based on cylindrical coordinates. The corresponding ring-shaped segments (see Figure 15) are optimal for conducting collision detection for the associated axis of rotation vs. cube shaped cells (e.g. (Teschner, Heidelberger, et al. 2003)). It should be noticed that this kind of cylindrical segments are not behaving like bounding volumes of the objects (e.g. spherical shells in (Krishnan, et al. 1998)) and they are just spatial segments. For clarity, we consider one of the objects fixed and call it the fixed object (and the other one is called the mobile object). After segmentation, one table is created for the fixed object, and list of the fixed polygons is stored. By comparing the position of each mobile vertex with the fixed polygons stored in the corresponding cell of table, the penetrating vertices are found. Figure 12 shows the block diagram for the main steps of the algorithm for a single axis of rotation. The details are explained in the following sections.

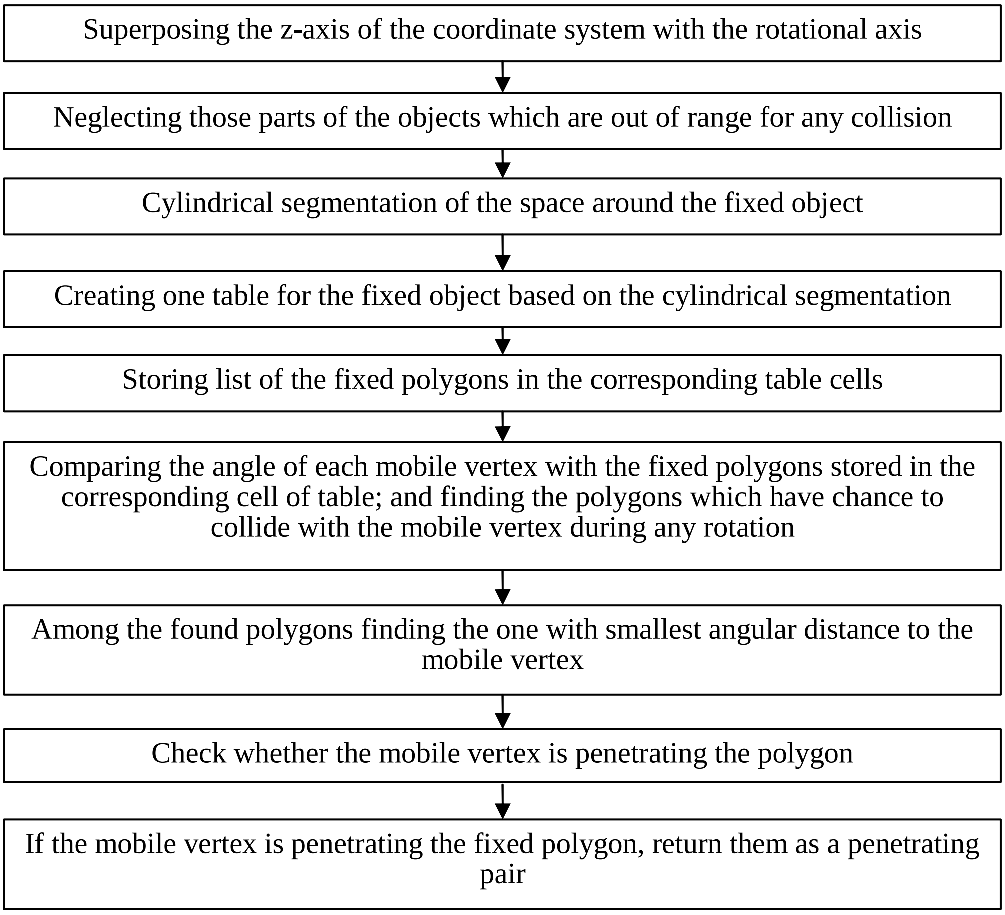
Superposing the z-axis of the coordinate system with the rotational axis
Neglecting those parts of the objects which are out of range for any collision
Cylindrical segmentation of the space around the fixed object
Creating one table for the fixed object based on the cylindrical segmentation
Storing list of the fixed polygons in the corresponding table cells
Comparing the angle of each mobile vertex with the fixed polygons stored in the corresponding cell of table; and finding the polygons which have chance to collide with the mobile vertex during any rotation
Among the found polygons finding the one with smallest angular distance to the mobile vertex
Check whether the mobile vertex is penetrating the polygon
If the mobile vertex is penetrating the fixed polygon, return them as a penetrating pair

Figure 12: Block diagram of the cylindrical segmenting collision detection algorithm, for a single axis of rotation.

3.2.2.1 Axis-aligned coordinate system

For making our calculations simpler, we build a coordinate system so that the mobile object's rotational axis is the z-axis. Therefore, at the beginning, we transform both objects in this new coordinate system (see Figure 13). This kind of transformation is very simple and fast; thus can be repeated in case the rotating axis changes.

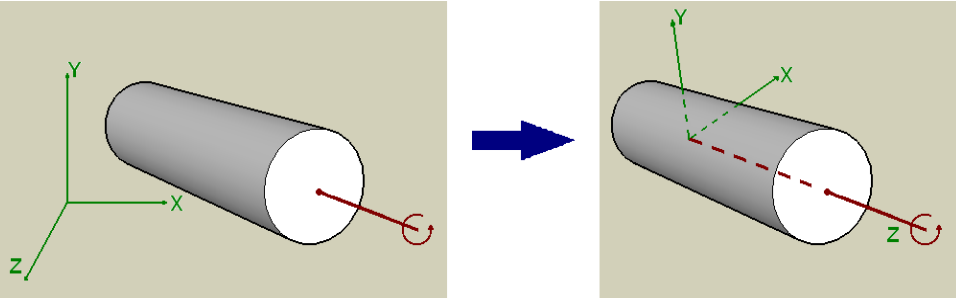

Figure 13: Axis-alignment of the coordinate system.

3.2.2.2 Cylindrical segmentation of the space around the objects and filling the table

Each vertex of both mobile and fixed object is converted from its Cartesian coordinates (X, Y, Z) to cylindrical coordinates (r, θ, z):

$$\text{if } (X\quad 0) \text{ then } \theta = \mathrm{ArcTan}(Y/X)\ ;\quad \text{else } \theta = \mathrm{ArcTan}(Y/X)+\pi\ . \qquad \text{Eq. 1}$$

$$\text{if } (\theta < 0) \text{ then } \theta = \theta + 2\pi\quad ;\quad \theta \in [0, 2\pi);$$

$$r = \sqrt{X^2 + Y^2}\quad ;\quad z = Z\ .$$

In order to avoid unnecessary calculations, we first calculate only the radius, ‘r’, for all the vertices. Then we find the maximum/minimum of ‘r’ and ‘z’ for both moving and fixed objects separately. These intersection defines the search intervals for r $[r_{G\_Min}, r_{G\_Max}]$ and z $[z_{G\_Min}, z_{G\_Max}]$:

$$r_{G\_Min} = \mathrm{Max}\ (r_{f\_Min}, r_{m\_Min})\ , \qquad \text{Eq. 2}$$

$$r_{G\_Max} = \mathrm{Min}\ (r_{f\_Max}, r_{m\_Max})\ ,$$

$$z_{G\_Min} = \mathrm{Max}\ (z_{f\_Min}, z_{m\_Min})\ ,$$

$$z_{G\_Max} = \mathrm{Min}\ (z_{f\_Max}, z_{m\_Max}),$$

where f (resp. m) designates the fixed object (resp. mobile object). It is clear that any point for which its ‘z’ is out of $[z_{G\_Min}, z_{G\_Max}]$ or its ‘r’ is out of $[r_{G\_Min}, r_{G\_Max}]$ does not collide with any other point (see Figure 14). It should be noticed that we can find and avoid more non-colliding points by introducing other parameters which are independent from ‘θ’ (i.e. independent from rotation) (such as $L = \sqrt{X^2 + Y^2 + Z^2}$, which was used in our experimental tests).

We prepare one table with size of $Z_{max} * R_{max}$, where Z_{max} and R_{max} are function of the search interval and of the resolution (resp. along z and along r):

$$Z_{max} = [\ (z_{G\_Max} - z_{G\_Min}) / Res_z\ ], \qquad \text{Eq. 3}$$

$$R_{max} = [\ (r_{G\_Max} - r_{G\_Min}) / Res_r\ ],$$

where Res_z and Res_r are the resolutions decided for z and r, respectively. Also, [x] means the integer ceiling of ‘x’. Each cell of the table corresponds to one ring segment of the space and the indices of the fixed polygons occupying that ring are stored in the corresponding table cell (see Figure 15).

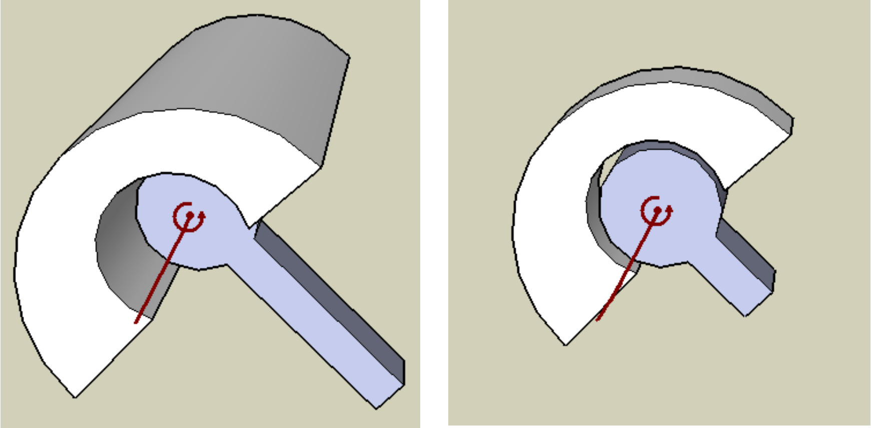

Figure 14: Neglecting non-colliding parts from further calculations. Left: original objects. Right: objects after neglecting some non-colliding parts (i.e. corresponding to the search intervals for r and z).

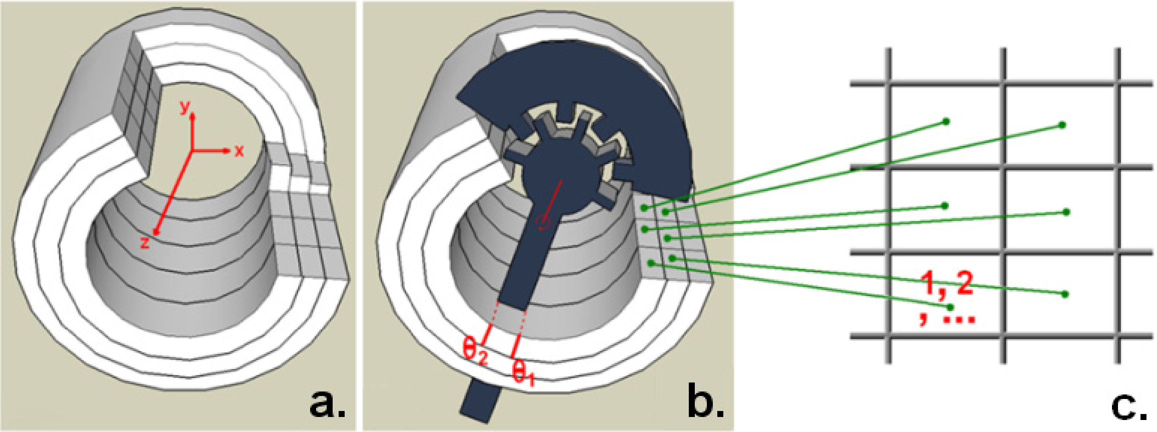

Figure 15: a) Cylindrical segmentation of the space without objects, b) with rotating objects, c) the corresponding table cells for storing list of the fixed polygons in each ring segment.

For each vertex of the objects, if ‘r’ and ‘z’ are inside their search interval, we discretize ‘r’ and ‘z’ according to our resolution and also calculate ‘θ’ for the vertex:

$$r_d = r_{discretized} = [\ (r - r_{G\_Min}) / Res_r\ ], \qquad \text{Eq. 4}$$

$$z_d = z_{discretized} = [\ (z - z_{G\_Min}) / Res_z\ ]\ .$$

Now, for each polygon of the fixed object's surface, the vertices are checked. If ‘r’ and ‘z’ of at least one of the polygon’s vertices are inside the valid ranges of $[r_{G\_Min}, r_{G\_Max}]$ and $[z_{G\_Min}, z_{G\_Max}]$ respectively, the polygon is considered for further tests, otherwise the polygon is neglected. To be sure that no polygon is missing, we can consider the valid range (search interval) wider than the estimated one. That means we should also consider the vertices which are about the average length of polygon's edge out of the valid range. After being assured that the polygon is inside the test range, the minimum and the maximum of 'r_d', 'z_d' of the box bounding the polygon are found (let's call them: r_{d_min},r_{d_max}; z_{d_min},z_{d_max}) (for the case that polygons are triangles, see Appendix A at the end of this thesis for more details). Then, the index of the polygon is stored in all the table cells which their row and column index are inside [z_{d_min},z_{d_max}] and [r_{d_min},r_{d_max}] respectively. This process is done for all the polygons in the fixed object and the polygon's indices are stored in the corresponding table cells. Therefore each cell contains series of indices of the fixed polygons whose range of 'z' and 'r' include row and column index of this cell.

3.2.2.3 Detecting the collision

In order to detect the collision and find the penetrating vertices, for each vertex in the mobile object, we find the corresponding ring segment (by using the discretized coordinates of the vertex, obtained from Eq. 4). Since the mobile object is rotating about the z axis, a mobile vertex can only collide with the fixed polygons stored in the corresponding table cell of the found segment ring. Therefore, we just need to check the angular distance between the mobile vertex and the fixed polygons in the corresponding table cell. For making our tests accurate, we should consider that it is possible to have some fixed polygons in the same ring segment of the mobile vertex, but they may never collide with the mobile vertex during any rotation. In other words, for a mobile vertex, not all the fixed polygons stored in the corresponding table cell can collide with it.

For finding and ignoring such non colliding fixed polygons we check whether a 2D representation of the vertex (based on r and z) is inside a 2D representation of the polygon (based on r and z); (for the case that polygons are triangles, see Appendix B at the end of this thesis for more details). For all the polygons with the positive answer, we calculate the angular distance between the mobile vertex and the fixed polygon (i.e. the amount of angle that the mobile vertex needs to be rotated in order to collide with the corresponding polygons) (see Appendix C at the end of this thesis for more details).

'The polygon returning the smallest angular distance to the vertex' and 'the rotational projection of the vertex on this polygon' are found. By using the polygon's normal vector (pointing out of the mesh), it is checked whether the mobile vertex is penetrating the fixed object or not. For this reason, the scalar product between 'the component of the polygon's normal vector which is tangential to the rotational trajectory' and 'the vector connecting the rotational projection of the vertex on the polygon to the vertex (itself)' is calculated (see Appendix D at the end of this thesis for more details). Since the polygon has the smallest angular distance to the vertex, the vertex is inside the fixed object if the angle between two vectors is larger than $\pi/2$. Therefore, their scalar product must be positive when the mobile vertex is not penetrating the fixed object (see Figure 16). This process is done for all the mobile vertices and finally all the penetrating mobile vertices and their corresponding fixed polygons are found. The algorithm is shown in Figure 17.

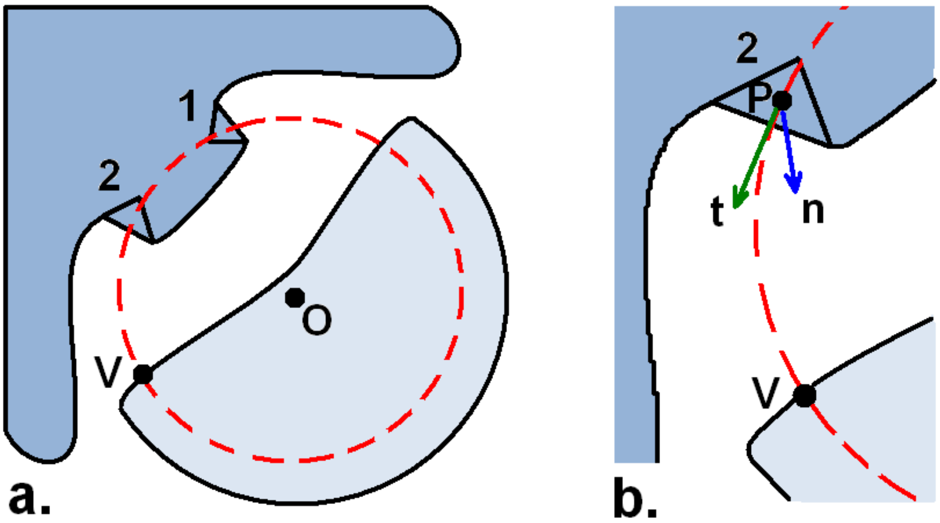

Figure 16: a) The light blue object is rotating about O and can collide with the dark blue object. The mobile vertex **V** is in the same cylindrical segment of the fixed triangles 1 and 2. **V** has the smallest angular distance to triangle 2; thus we only need to check whether **V** is penetrating triangle 2 or not. b) Vertex **P** is the rotational projection of **V** on triangle 2. By using the normal of triangle 2 (blue vector of **n**, pointing outward the fixed object) we find vector **t** (green vector) which is tangential to the rotational trajectory of vertex **V** (red dashed curve) and is also pointing outward the fixed object. By examining the angle between **t** and the vector connecting **P** to **V**, we can find out if **V** is inside the fixed object or not (see Appendix D at the end of this thesis for more details).

```
BEGIN
  k = 0
  FOR all the mobile vertices
    i = index of vertex
    [$z_d$, $r_d$] = cylindrical partition containing “vertex[i]”
    $delta_{min}$ = $2\pi$
    $j_{min}$ = -1
    FOR all the fixed polygons stored in table cell[$z_d$,$r_d$]
      j = index of polygon
      IF (Circular arc of “vertex[i]” intersects “polygon[j]”) THEN
        delta = arc angle between “vertex[i]” and “polygon[j]”
        IF (delta < $delta_{min}$) THEN
          $delta_{min}$ = delta
          $j_{min}$ = j
    IF ($j_{min}$   -1) THEN
      P = circular projection of “vertex[i]” on “polygon[$j_{min}$]”
      t = tangent vector to rotational trajectory at “P”
      n = normal of “polygon[$j_{min}$]”
      a = vertex[i] - P
      IF (t.n < 0) THEN
        t = -t
      IF (t.a < 0) THEN
        Colliding Pairs[k] = (vertex[i], polygon[$j_{min}$])
        Penetration Depth[k] = $delta_{min}$
        k = k+1;
  Return(k ‘number of collision pairs’)
END
```

Figure 17: The algorithm for collision detection based on cylindrical segmentation.

3.2.3 Pre-processing

Some parts of the process can be done in the pre-processing level in order to speed up the main collision detection process and avoid repeating them during the simulation.

3.2.3.1 Fixed object

Since the fixed object is not changing its position, the table built for the fixed object is valid for all the simulations and there is no need to re-build it during the simulation. The only concern can happen when the fixed object is getting deformed after the collision. Since the table is storing the fixed polygons' information based on their z and r (not their θ); therefore any deformation in the angular direction does not affect the table. As the mobile object is only rotating, it can be a true assumption to consider that the majority of the deformations are happening in the angular direction, which only affect the θ, while keeping z and r unchanged. So, in general we may not need to update the table in most of the deformations. Even for the deformations in the non-angular direction, just some of the polygons may be shifted to the neighboring segments. Therefore, instead of updated the whole table, we only need to update the table cells related to the polygons which are deformed in the non-angular direction. In case the deformation is small, by knowing the direction of deformation, we can simply move the information of the deformed polygon from its previous table cells to the neighboring table cells. Still we can completely avoid updating the table, and instead of that consider checking the polygons stored in the neighboring cells of the mobile vertices (in addition to the corresponding cell), during the collision detection. Thus, in case a polygon is shifted to another neighboring cell due to deformation, we will not miss checking it.

3.2.3.2 Mobile object

Although there is no table for the mobile object to be updated, but we still can do some pre-calculation in order to speed up the main collision detection process. In fact we can find the angular projection of each mobile vertex on its corresponding polygons (occupying the same ring), in the pre-processing level. This information does not change during the rotation as the angular projection of the mobile vertex is only depending on the z and r elements of the mobile vertices (not θ). Again, after any deformation, if the deformation of the mobile object is happening in the angular direction, these values will not get affected and there is no need to re-calculate them. But in case the fixed object is deformed or the mobile object is deformed in a non-angular direction, this information should be calculated again. For this reason, same as updating the fixed table, instead of updating the information related to all of the mobile vertices, we only need to update the information about the mobile vertices which are deformed in the non-angular direction or the mobile vertices occupying the same spatial segment of the deformed fixed polygons.

3.2.4 Discretizing resolution

If we choose a resolution smaller than the average length of polygons then we have to perform more unnecessary comparisons between table cells. The reason is that the information stored about one polygon is the same in all of the cells covering it. So having more cells does not provide more information but it may slow down the process for filling the table and consume a larger amount of memory. On the other hand, if we choose a resolution larger than the average length of polygons, then a large number of neighboring polygons can occupy one cell; therefore the larger number of possible colliding polygons is detected incorrectly and consequently more intersections tests are needed to be done in each ring, which may slow down the process. Thus, the best value for resolution of ‘z’ and ‘r’ is the average length of polygons. This choice of resolution is also similar to the resolution of cubic cells chosen by (Teschner, Heidelberger, et al. 2003), for detecting collision based on spatial cubic discretizing.

3.3 Radial collision

3.3.1 Previous methods

As it was explained before, most of the collision detection research and investigations are either for general cases, or for very specific applications. However, for detecting collision among rotating or sliding objects, there is a ray-based sampling method proposed by (Maciel, Boulic and Thalmann 2007). In this method, in a preprocessing stage, a mesh is chosen under certain conditions as a reference mesh and is spherically sampled by using spherical coordinate system. To create the table, every table cell is visited. For each of them, a pair of spherical coordinates is calculated based on the indices’ value [row; column]. Then, a ray is traced in the direction determined by the spherical coordinates, and the triangle of the fixed mesh intersecting that ray is assigned to that table cell. The table cells not assigned to a triangle are marked as empty. In the collision detection stage, the resulting table is exploited for each vertex of the other mesh to obtain its signed distance to the fixed mesh. The signed distance is finally compared to the one of the facing vertex from the mobile mesh for making decisions regarding collision. The table does not need to be updated at simulation time if we assume that the fixed mesh is rigid or deforms in radial direction.

This method returns the pairs of penetrated mobile vertices and their corresponding penetrated fixed triangles. That means for all the penetrating vertices of the mobile object, the corresponding penetrating triangle of the fixed object, which is in the vertex’s radial direction, is returned. Knowing the corresponding triangles of each penetrating vertex is useful for estimating the penetration depth of each vertex in radial direction. Therefore, this method not only returns the colliding vertices but also provides the radial penetration depth without performing any additional computations. The fact that the returned penetration depth is in radial direction, can be appropriate as the method is used for sliding cases.

Although the method can return all the penetrated vertices of the mobile object, it may not return all the penetrating triangles of the fixed object (as it only returns the penetrated fixed triangles which corresponds to the penetrated mobile vertices), but in case we need all the colliding vertices of all the objects, the method can be performed two times, where in the second time the mobile and the fixed objects should change their role (by noticing the movement is relatively defined).

3.3.2 Weakness of the previous methods

In the general methods, the fact that the methods are designed for the wide range of arbitrary cases, does not allow to take advantage of the rotation/sliding limitation. Therefore the general methods are not optimum in the performance and some unnecessary calculation may slow down the process. In ray-based sampling method proposed by

(Maciel, Boulic and Thalmann 2007), in each table cell, the triangle which is intersected by the ray corresponding to the table cell is stored. Each table cell is corresponding to a ray (vector) connecting the origin to the center of spatial spherical cell grid. In fact, each table cell is corresponding to a spatial spherical cell, where the discretized spherical coordinates of the central point of the cell's grid (indexing vertex) is forming the corresponding table cell's index. Therefore if a cell is occupied by more than one polygon, only the polygon which is intersected by the ray passing through the indexing vertex of the call grid is stored in the corresponding table cell and the other occupying polygons are ignored (Figure 18).

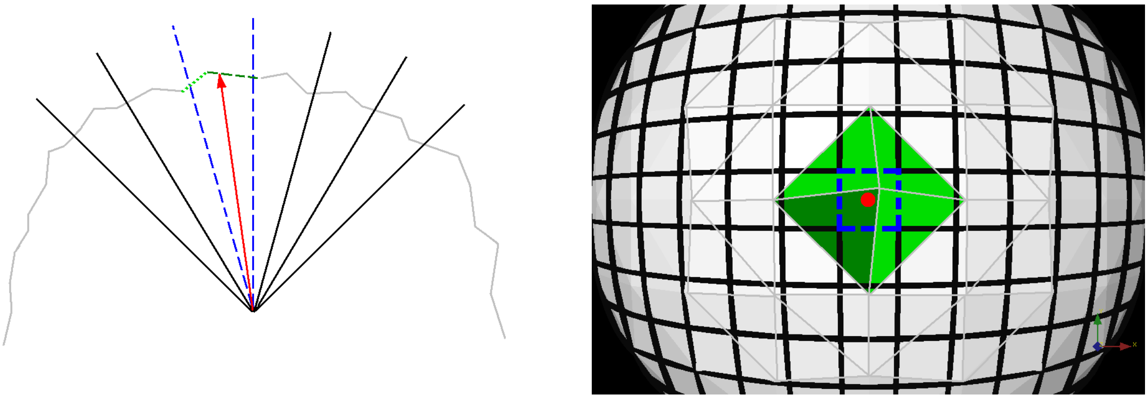

Figure 18: A spatial cell occupied by more than one triangle (left: 2D, right: 3D). The triangles are shown by gray lines. The black lines represent the spatial spherical grids (of the segments). The red vector (shown by a red circle in the right figure) is the indexing vector of the blue spatial cell (shown by dashed blue lines among the black lines). All the green triangles (shown by dashed and dotted lines in the left figure) are inside this cell but only the dark green one (shown by a dashed line in the left figure) is stored in the corresponding table cell. The reason is that the red indexing vector is not intersecting the light green triangles.

Since only one polygon can be stored in each table call, some colliding vertices may be neglected incorrectly. In fact, during the collision detection all the mobile vertices that are inside a spatial spherical cell are checked against the polygon stored in the corresponding table cell. Since each table cell contains the information of only one polygon, there will be a risk that some of the mobile vertices penetrate the polygons which are in the spatial cell but are not stored in the corresponding table cell (Figure 19). For increasing the accuracy, one has to increase the resolution of the space grid cells and sample the space more precisely. Such precise sampling can improve the accuracy, but beside taking large amount of memory and reducing the pre-processing speed, it cannot grant the 100% of accuracy for any shape of objects (the problem can be severe when the fixed object has some holes).

As described in (Maciel, Boulic and Thalmann 2007), if the fixed object's deformation is only in the radial direction, we do not need to re-create the tables. But in case the fixed object gets some deformations in non-radial directions, the pre-processing must be done again and the table must be updated. But the strategy used for creating and filling the tables is depending on the table size and can be very slow, especially when we use a high resolution space grid size (which is needed for improving the accuracy of the results).

Since the ray-based sampling method uses spherical coordinate system, the space cell grids are not distributed uniformly, which can vary the number of triangles occupying each space cell in different spatial regions (i.e. some regions (e.g. polar areas) may be sampled much more precisely than some other regions (e.g. equatorial areas)). Also the function transforming Cartesian coordinates to the spherical coordinates is a trigonometrical function which may be called for several times during the simulation. This several calling of trigonometrical functions can be time consuming specially for the high resolution objects.

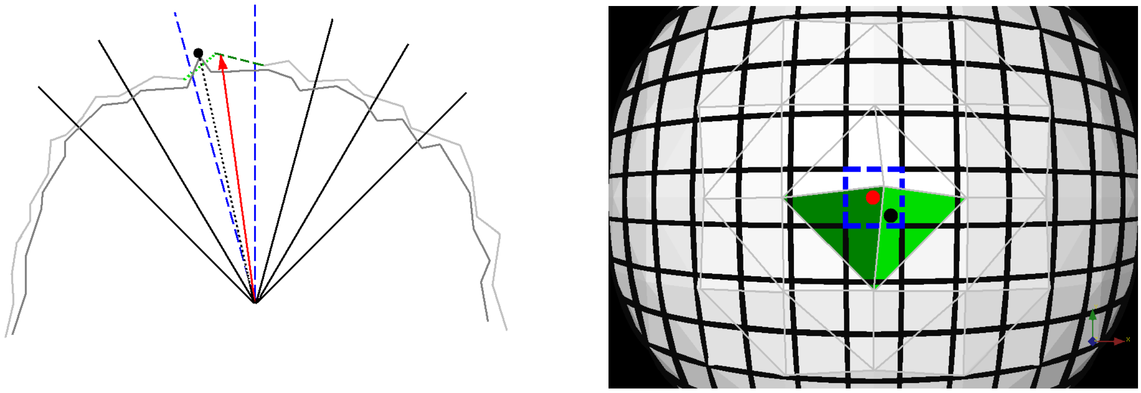

Figure 19: A mobile vertex penetrating a fixed polygon which is not stored in the corresponding table cell (left: 2D, right: 3D). The outer and inner surface in the left figure, are considered as the fixed and mobile object, respectively. The fixed triangles are shown by gray lines in the right figure. The black lines represent the spatial spherical grids (of the segments). The red vector (shown by a small circle inside the dark green triangle in the right figure) is the indexing vector of the blue spatial cell (shown by two long dashed lines). The dark green triangle (shown by a dashed line in the left figure) is stored in the corresponding table cell (because it intersects the red vector), but the light green triangle (shown by a dotted short line in the left figure) is not stored in that table cell while it is inside the blue spatial cell too. A vector is connecting the origin to the mobile vertex which penetrates the fixed object inside the blue spatial cell (shown by a small black circle inside the light green triangle in the right figure and by a dotted black line in the left figure). Since the vector related to penetrating vertex is intersecting the light green triangle, and not the dark green one; the algorithm may not detect the vertex as a penetrating one.

In this section we solve the problems of the previous method (accuracy and slow table creation), by proposing a new method using another strategy for creating and filling the tables. We also investigated the effect of uniformity of the space grid cells and also the effect of the simplicity of transforming function (by proposing another coordinate representation) on the performance of our new methods to decide for the best one.

3.3.3 Method

The method works by radial segmentation of the object's spatial occupancy. In general aspect, the method is basically working by comparing the radial distance between the vertices and the polygons which is inspired from the work done in (Maciel, Boulic and Thalmann 2007). The major difference with ray-based sampling method is in the strategy used for creating and filling the table and avoiding sampling the object mesh. The proposed new strategy not only makes the method return accurate collision answers (vs. the approximated answer of (Maciel, Boulic and Thalmann 2007)), but also increases the speed of table updating, significantly. We also investigated the effect of the mapping function (to map from Cartesian coordinates to radial segments) on the processing time to check the importance of simplicity of the function vs. uniformity of its output's distribution. Figure 20 shows the block diagram for the main steps of the algorithm.

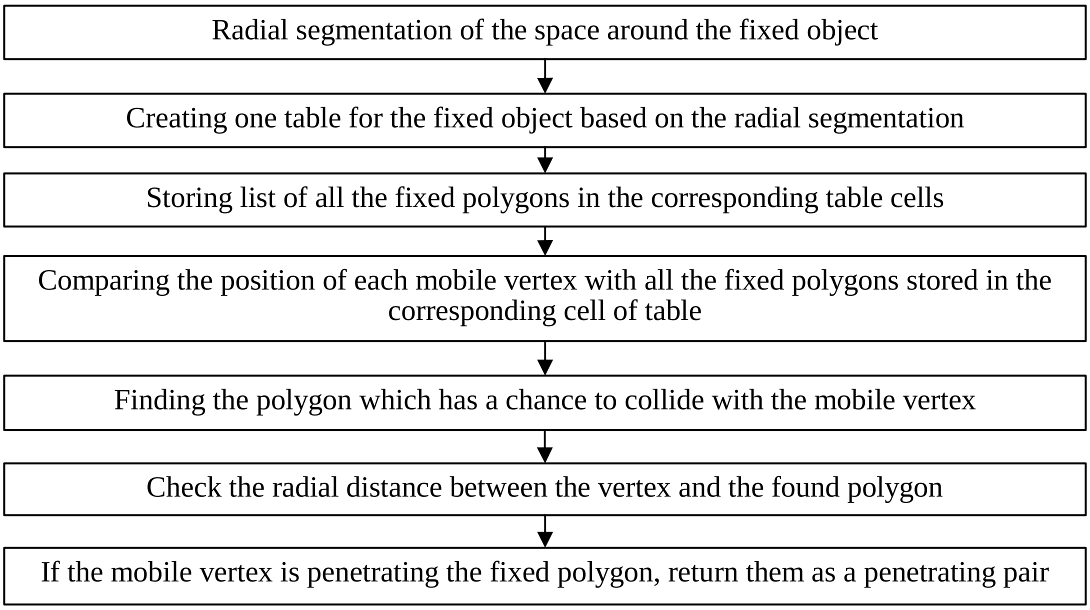
Radial segmentation of the space around the fixed object
Creating one table for the fixed object based on the radial segmentation
Storing list of all the fixed polygons in the corresponding table cells
Comparing the position of each mobile vertex with all the fixed polygons stored in the corresponding cell of table
Finding the polygon which has a chance to collide with the mobile vertex
Check the radial distance between the vertex and the found polygon
If the mobile vertex is penetrating the fixed polygon, return them as a penetrating pair

Figure 20: Block diagram of the radial segmenting collision detection algorithm.

3.3.3.1 Radial segmentation of the space around the objects and filling the table

Radial segmentation should be done in the way that all the points having the same orientation from the origin (but different distance) are classified in one segment. (Maciel, Boulic and Thalmann 2007) used θ and φ elements of the spherical coordinate system for radial segmenting the space. We used three types of functions. The first type is similar to what (Maciel, Boulic and Thalmann 2007) used. The second type is using a simpler function compared to the trigonometrical functions used for transforming to spherical coordinates. The third function gives spherically more uniform distribution of the outputs compared to the spherical coordinate system.

3.3.3.1.1 Mapping function type 1

As it was said, the first type is simply the function that transforms from Cartesian coordinate system to the spherical coordinate system:

$(a_1, a_2, R) = \mathrm{FunctionType1}(x, y, z),$ Eq. 5

$R = \sqrt{x^2 + y^2 + z^2},$

if $(x \quad 0)$ *then* $a_1 = \theta = \mathrm{ArcTan}(y/x)$;

if $(x<0 \ \& \ y \quad 0)$ *then* $a_1 = \theta = \mathrm{ArcTan}(y/x)+\pi$;

if $(x<0 \ \& \ y<0)$ *then* $a_1 = \theta = \mathrm{ArcTan}(y/x)-\pi$;

$a_1 \in (-\pi, \pi]$;

$a_2 = \varphi = \mathrm{ArcCos}(z/R)$,

$a_2 \in [0, \pi]$.

Where x, y and z are the elements of the Cartesian coordinate system. Based on a_1 and a_2, the space is radially segmented (see Figure 21).

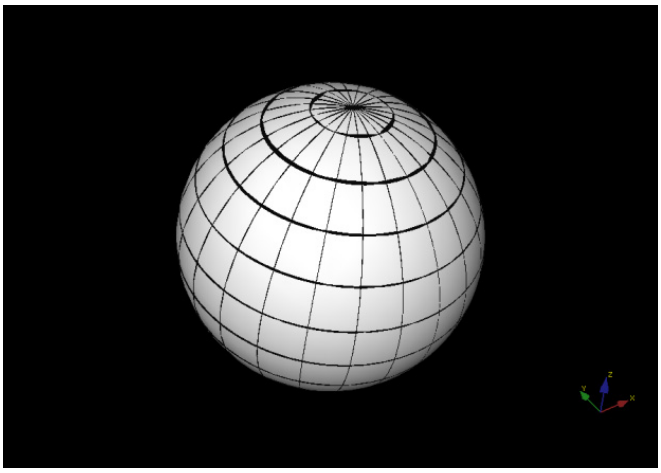

Figure 21: Radial segmentation based on the first type function (based on spherical coordinate system).

3.3.3.1.2 Mapping function type 2

In this mapping function, we used normalized Cartesian coordinates in order to avoid using trigonometrical transformations and make the function as simple as possible:

$(a_1, a_2, a_3, R) = \mathrm{FunctionType2}(x, y, z)$ Eq. 6

$R = \sqrt{x^2 + y^2 + z^2}$,

$a_1 = x/R$,

$a_2 = y/R$,

$\mathit{if}\ (z \quad 0)\ \ \mathit{then}\ \ a_3 = 1\ \ \mathit{else}\ \ a_3 = 0$.

Based on a_1, a_2 and a_3, the space is radially segmented (see Figure 22).

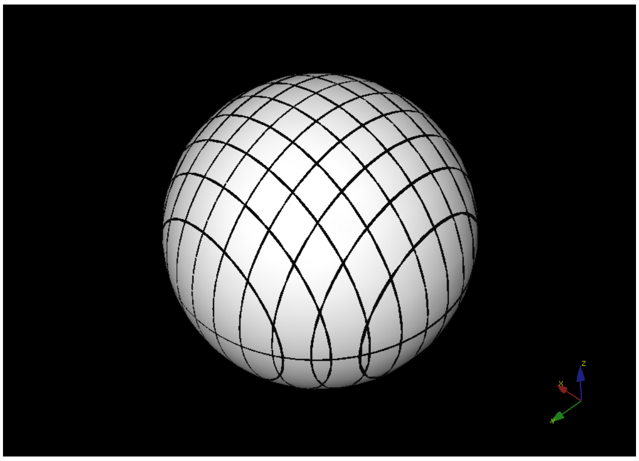

Figure 22: Radial segmentation based on the second type function (based on normalized Cartesian coordinates).

3.3.3.1.3 Mapping function type 3

In this mapping function, we tried to have more uniform distributed radial segments in the space. Different ways such as using recursive methods (Ritter 1999) or helix (Nishio, et al. 2006) can be applied for having uniform sampling of a sphere. However, having a uniformly sampled sphere does not necessarily provide the desired simple radial segments for classifying the polygons. The method that we used for this kind of segmentation is basically similar to the methods explained in (Yershova and LaValle 2004):

$$(a_1, a_2, a_3, R) = \text{FunctionType3}(x, y, z) \qquad \text{Eq. 7}$$

$$R = \sqrt{x^2 + y^2 + z^2},$$

Based on a_1, a_2 and a_3, the space is radially segmented in a quasi-uniform way (see Figure 23). The details about the calculation of a_1, a_2 and a_3 can be found in Appendix E.

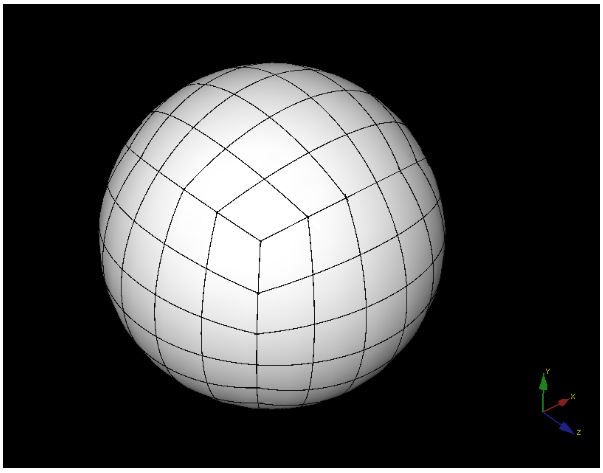

Figure 23: Radial segmentation based on the third type function (based on quasi-uniform radial segmentation).

3.3.3.1.4 Creating and filling the table

Based on a given table size, the table dimension is defined and an empty table is built:

using function type1: Eq. 8

$$\text{Dimension} = \left[\sqrt{\text{table size}}\right] \quad ; \quad \text{Table}(\text{Dimension} \times \text{Dimension})$$

using function type2:

$$\text{Dimension} = \left[\sqrt{\frac{\text{table size}}{2}}\right] \quad ; \quad \text{Table}(\text{Dimension} \times \text{Dimension} \times 2)$$

using function type3:

$$\text{Dimension} = \left[\sqrt{\frac{\text{table size}}{6}}\right] \quad ; \quad \text{Table}(\text{Dimension} \times \text{Dimension} \times 6)$$

where [x] means the larger closest integer value to x. So, as it can be seen in Eq. 8, for all types of mapping function the table's size is kept the same, although their dimension may change.

Based on the chosen mapping function, for all the vertices in the fixed object, we calculate a_1, a_2 (and a_3 for type 2 and 3). If we are using mapping function type 1 or 2, then among all the calculated a_1 and a_2 the maximum and the minimum of them are found:

$$a_{1\_Min} = \mathrm{Min}|_{a1}$$
$$a_{1\_Max} = \mathrm{Max}|_{a1} \ ,$$
$$a_{2\_Min} = \mathrm{Min}|_{a2}$$
$$a_{2\_Max} = \mathrm{Max}|_{a2} \ .$$
Eq. 9

For type 3 we simply define:

$$a_{1\_Min} = a_{2\_Min} = 0 \ ,$$
$$a_{1\_Max} = a_{2\_Max} = \pi/2 \ .$$
Eq. 10

Now for each fixed vertex we discretize its a_1 and a_2 based on the table's Dimension, defined in Eq. 8:

$$Res_{a1} = (a_{1\_Max} - a_{1\_Min}) / (\mathrm{Dimension}-1) \ ,$$
$$Res_{a2} = (a_{2\_Max} - a_{2\_Min}) / (\mathrm{Dimension}-1) \ ,$$
$$a_{1d} = (a_1 - a_{1\_Min}) / Res_{a1} \ ,$$
$$a_{2d} = (a_2 - a_{2\_Min}) / Res_{a2} \ .$$
$$a_{3d} = a_3 \ \{\text{for mapping function type 2 and 3}\}$$
Eq. 11

a_{1d}, a_{2d} (and a_{3d} for type 2 and 3) of a vertex are in fact the address of a table cell corresponding to the vertex. After discretizing all the a_1 and a_2 of the fixed vertices, now we start filling the table by finding the occupancy of all the fixed polygons. In other words, for each polygon of the fixed object, by using a_{1d}, a_{2d} (and a_{3d} for type 2 and 3) of its vertices we found all the table cells which cover it. It should be noticed that depending on the used mapping function the segments borders may have a curved shape rather than a linear shape. Thus, in some cases, for finding the segments containing the polygon, extra points on the polygons edges have to be considered in addition to the polygon vertices (see Figure 24).

After finding all the covering cells of a fixed polygon, the index of the polygon is stored in all of those covering cells. Since this method is done for all the fixed polygons, it ensures that no polygon is missing from being classified inside the table and also it is possible that some table cells contain more than one polygon's index. The algorithm for filling the table is shown in Figure 25.

The strategy of filling the table in ray-based sampling method proposed by (Maciel, Boulic and Thalmann 2007) is working by designating one radial vector to each table cell. Then for each table cell, all the fixed polygons are checked to see whether the radial vector of the table cell intersects any polygon or not. If a polygon is intersected then the polygon's index is stored in the corresponding table cell. As it can be seen the ray-based sampling method is working by searching among all the polygons for each table cell, while in the new proposed method, no searching algorithm is performed, which makes it much faster. Also, in the strategy used for filling the table in ray-based sampling method, a table cell is filled based on a radial vector associated to it. Therefore, for each table cell, only one polygon can be found, even if the associated segment to that table cell be occupied by more than one polygon. It is in contrary to the new proposed method which allows any polygons occupying a table cell be found and consequently no information is missed.

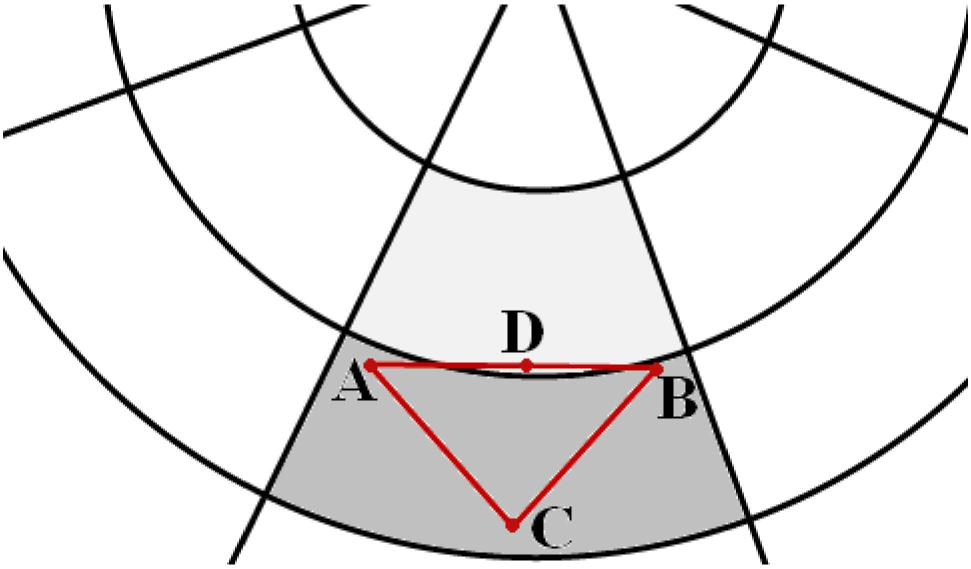

Figure 24: Finding the occupancy of the fixed polygons by considering extra points on the polygons edges, in addition to their vertices. Red triangle (polygon) is occupying both dark and light gray segments. However, due to curved shape of the segments borders (depending on the exploited mapping function), considering the triangle vertices (A, B and C) may only lead to find the dark segment. Thus, another point such as D should be considered on the triangle edge to find the light gray segment too.

```
BEGIN
  FOR all the fixed polygons
    i = index of polygon;
    [$a_{1d\_min}$,$a_{1d\_max}$] [$a_{2d\_min}$,$a_{2d\_max}$] = bounding range of "polygon[i]"
    FOR j = $a_{1d\_min}$ TO $a_{1d\_max}$
      FOR k = $a_{2d\_min}$ TO $a_{2d\_max}$
        store value "i" in cell[j][k] of the table
END
```

Figure 25: The algorithm for filling the table in the radial segmenting collision detection method.

3.3.3.2 Detecting the collision

After filling the table, it is time to find all the colliding pairs. For this reason, for each mobile vertex, the index of the table cell corresponding to the spatial radial segment containing the vertex is found (i.e. a_{1d}, a_{2d} (and a_{3d} for type 2 and 3) of the vertex). By knowing the corresponding table cell of the mobile vertex, we just need to check whether the vertex penetrates any fixed polygons whose index is stored in that table cell (let’s call them candidate polygons) or not. First of all, we discard the candidate polygons whose planes are intersected (if the fixed object is inside the mobile object) or not intersected (if the mobile object is sliding inside the fixed object) by the line connecting the origin to the mobile vertex. After that, among the remaining candidate polygons, we search for a polygon which is in the radial direction of the vertex, by checking whether the vector connecting the origin to the mobile vertex (regardless to its size) is intersecting the polygon (not its plane) or not. The found polygon (if any) and the mobile vertex are considered as one of the penetration pairs and the radial distance between them is the penetration depth of the vertex (see Figure 26) (see Appendix F at the end of this thesis for more details, in case the polygons are triangles). This process is done for all the mobile vertices and the penetrating mobile vertices with their corresponding fixed polygons are stored as penetrating pairs. The algorithm is shown in Figure 27 for the case that the mobile object is sliding inside the fixed object. When the fixed object is inside the mobile one, the algorithm is the same, but all the inequality brackets (> or <) are replaced by the opposite ones (< or >).

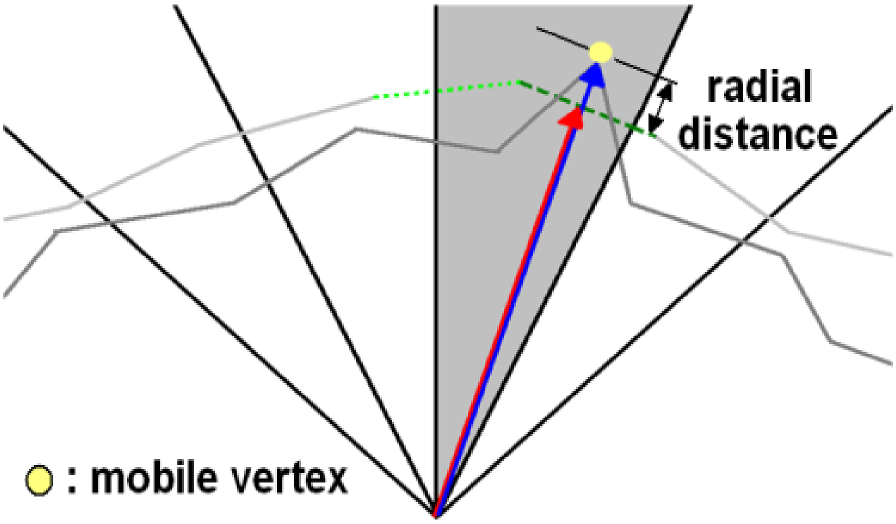

Figure 26: Checking whether the mobile vertex is penetrating the fixed polygon. Both of the green fixed triangles (dark dashed line and light dotted line) are inside the gray spatial cell and therefore they are the candidates to be checked against the mobile vertex (shown by a light yellow circle inside the gray spatial cell). Only the dark green triangle (dashed line) is intersected by the vector connecting the origin to the mobile vertex; thus, the light green triangle (dotted line) is ignored. The signed radial distance between the dark green triangle and the vertex is calculated by subtracting the red vector (short vector) from the blue one (long vector).

In case the polygons are triangles, the intersecting tests and the radial distance calculations can be done by using Barycentric coordinate (Schneider and Eberly 2003) (see Appendix F at the end of this thesis for more details). The same method can be also applied for the non triangular polygons (i.e. to consider each polygon as a combination of triangles and to do the ray-triangle test for each of them (Schneider and Eberly 2003)).

```
BEGIN
  k = 0
  FOR all the mobile vertices
    i = index of vertex
    ($a_{1\_d}$, $a_{2\_d}$, $a_{3\_d}$, R) = discretized radial coordinates of "vertex[i]"
    FOR fixed polygons stored in table cell[$a_{1\_d}$,$a_{2\_d}$,$a_{3\_d}$]
      j = index of polygon
      L = line passing through the origin and "vertex[i]"
      P = intersection point between the "polygon[j]"'s plane and "L"
      Diff = Dist2Origin(vertex[i])-Dist2Origin(P)
      IF (Diff   0) THEN
        IF "L" intersects "Polygon[j]" THEN
          Colliding pairs[k] = (vertex[i], polygon[j])
          Penetration depth[k] = Diff
          k = k+1
  Return(k 'number of collision pairs are found')
END
```

Figure 27: The algorithm for collision detection based on radial segmentation (mobile object is moving inside the fixed object). Dist2Origin(vertex) returns the Euclidian distance between the vertex and the origin.

3.3.4 Pre-processing

Similar to the cylindrical segmenting collision detection, some parts of the process can be done in the pre-processing level in order to speed up the main collision detection process and avoid repeating them during the simulation.

3.3.4.1 Fixed object

Since the fixed object is not changing its position, the table built for the fixed object is valid for all the simulations and there is no need to re-build it during the simulation. The only concern can happen when the fixed object is deformed after the collision. Since the table is storing the fixed polygons' information based on their radial information a_{1d}, a_{2d} (and a_{3d} for type 2 and 3) (not their R); therefore any deformation in the radial direction only changes the value of R and does not affect the table's information. As the mobile object is sliding on or under the fixed object, it can be a true assumption to consider that the majority of the deformations are happening in the radial direction (Maciel, Boulic and Thalmann 2007), which only affect the R, while keeping a_{1d}, a_{2d} and a_{3d} unchanged. So, in general we need to update the tables only if the fixed object is deformed in a non radial direction. Even in case the deformation is in non radial direction we only have to update the table information about the deformed polygons and not all.

Same as cylindrical segmenting method, we can also completely avoid updating the tables in deforming situation. In this case, instead of updating the table we expand the number of radial segments checked per each mobile vertex. In fact, we consider checking the polygons stored in the neighboring cells of the mobile vertices too (in addition to the corresponding cell), during the collision detection. Thus, in case a polygon is shifted to another neighboring cell due to deformation, we will not miss checking it.

There is no table for the mobile object and all the information related to the mobile object is calculated during the collision detection step. So, there is no pre-processing regarding to the mobile object and it can be either deformable or rigid without any restriction.

3.3.5 Table size

The table size decides the segmentation resolution (see Eq. 8). Similar to the cylindrical segmenting collision detection method, if we choose a too precise resolution that causes a polygon occupying several spatial segments, then we just consume more amount of memory. Because the information stored about one polygon is the same in all of the cells covering it, and having more cells does not provide more information but it may slow down the process for filling the table and consume a larger amount of memory. On the other hand, if we choose a too large resolution that causes several polygons to occupy one table cell, then we will have a larger number of candidate polygons for each mobile vertex and consequently more tests are needed to be done for each mobile vertex, which may slow down the process. Thus, the resolution should be chosen about the average size of polygons. Based on the chosen resolution, the table size can be decided too. In the ray-based sampling collision detection method proposed in (Maciel, Boulic and Thalmann 2007), in the first glance it may be assumed that the decision on the table size must be followed as the same rule as above. But since the results of ray-based sampling method are approximated, it is always better to have a larger size for the table, in order to reduce the proportion of wrong results. On the other hand, as it was explained, having a large size of table can consume large amount of memory and slow down the process for filling the table.

3.4 Testing and comparison

To evaluate the proposed methods for collision detection during rotation (striking and sliding), we compared the performance of the proposed methods with the performance of the previous methods in different scenarios. There are some benchmarks for comparing different collision detection methods; however their targets are usually the methods with general applications (Trenkel, Weller and Zachmann 2007). Therefore, we defined our scenarios which are more related to rotating/sliding cases.

3.4.1 Reference methods for the comparisons

For the sliding case, (Maciel, Boulic and Thalmann 2007) shows that their method performs faster than the other collision detection methods (e.g. proximities). Therefore, we used the method proposed in (Maciel, Boulic and Thalmann 2007) as one of our main reference methods in the sliding cases. Also we considered oriented bounding volume hierarchy method (OBVH) (Hudson, et al. 1997) and the spatial hashing method proposed in (Teschner, Heidelberger, et al. 2003) for all of the comparisons (in both striking and sliding cases). It should be noticed that the applied OBVH method only returns whether there is a collision or not, without giving the details about the colliding elements. Although, spatial hashing method returns the details about the penetrating vertices, it is not able to return them as pair in order to find penetrating depth without any additional computation (vs. ray-based sampling method and our methods which return the colliding elements as pairs). As it is explained in (Teschner, Heidelberger, et al. 2003), the spatial hashing method works by checking all the vertices (including internal vertices) vs. tetrahedra. Since we only need to find the colliding vertices of the surface, we increased its processing speed by making it only check the mesh vertices (and not internal vertices).

3.4.2 Comparison scenarios

In total, fifteen scenarios were tested for comparing the collision detection methods in a PC with CPU: Xeon-3.4 GHz & RAM: 2GB. There were two groups of scenarios: one for the striking collision and the other one for sliding collision. In each scenario, the experiment was performed three times, and the average value of the computational time was recorded.

3.4.2.1 First group of scenarios: cylindrical segmenting collision detection

For the first group (first four scenarios), cylindrical segmenting method was used and collision between right femur and pelvis in the human hip was detected during the femur's rotation (see Figure 28). All of the tests were done for four different resolutions of the objects (see Table 1). The objects were made up of 3D triangular meshes obtained by segmenting MR Images taken from a patient by the method described in (Gilles, Moccozet and Magnenat-Thalmann 2006).[5] Pelvis center of rotation was also found based on the scanned 3D models (Kang, Sadri, et al. 2003). Since in the spatial hashing method, the objects' volume should be built up of tetrahedra, we discretized the objects' volume for generating tetrahedra by using TetGen library (Si, TetGen 2005) which works based on the method explained in (Si and Gärtner 2004) (the other tested collision detection methods only need to have the information related to the objects' mesh).

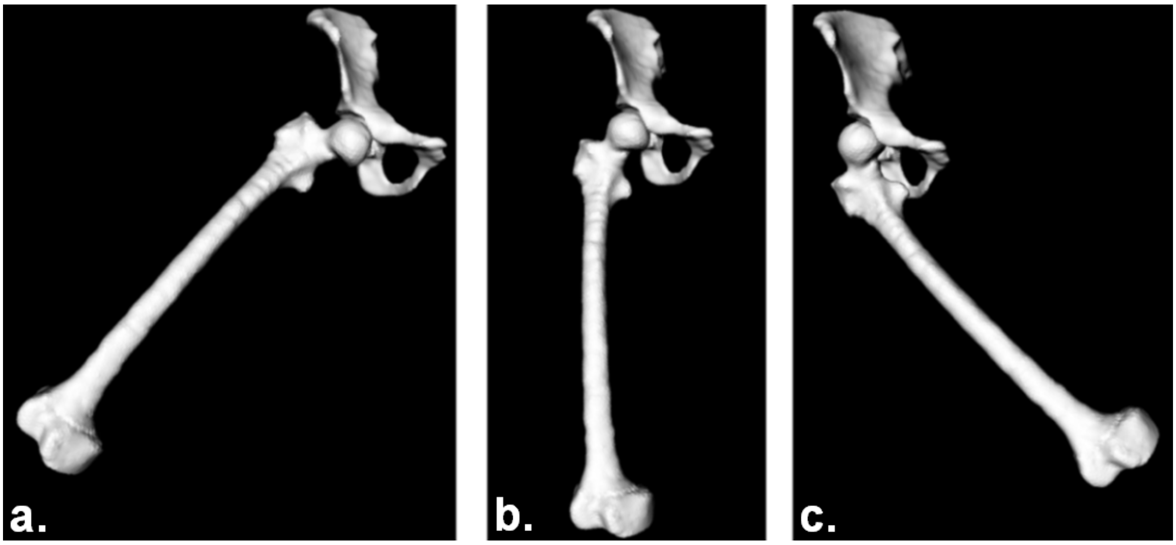

Figure 28: The first group of scenarios: using cylindrical segmenting method to find collision between femur and pelvis (since the cartilages are not considered, no sliding is happening on the femur ball).

---

[5] The 3D Meshes were prepared by Benjamin Gilles at MIRALab, University of Geneva.

Table 1: Different resolution of the objects in the first group of scenarios.

	Scen.1	Scen.2	Scen.3	Scen.4
Number of vertices in the femur’s mesh	773	3 086	12 338	49 346
Number of vertices in the pelvis’es mesh	1 221	4 884	19 536	78 144
Total number of vertices (mesh)	**1 994**	**7 970**	**31 874**	**127 490**
Number of triangles in the femur’s mesh	1 542	6 168	24 672	98 688
Number of triangles in the pelvis’es mesh	2 442	9 768	39 072	156 288
Total number of triangles (mesh)	**3 984**	**15 936**	**63 744**	**254 976**
Number of tetrahedra in the femur’s volume	8 232	28 257	104 839	368 839
Number of tetrahedra in the pelvis’es volume	9 954	31 410	100 550	344 486
Total number of tetrahedra (volume)	**18 186**	**59 667**	**205 389**	**713 325**

3.4.2.2 Second group of scenarios: radial segmenting collision detection

In the second group (fifth to fifteenth scenarios), two different cases were tested. The first case was used for a very general test in order to find the best mapping function (in addition to comparing to the other methods). In this case, radial collision between a sphere and a cube covering about 17% of the sphere’s upper part was detected during the sphere’s rotation (see Figure 29). All of the tests were done for seven different resolutions of the objects (see Table 2) and all three different types of mapping functions were considered separately.

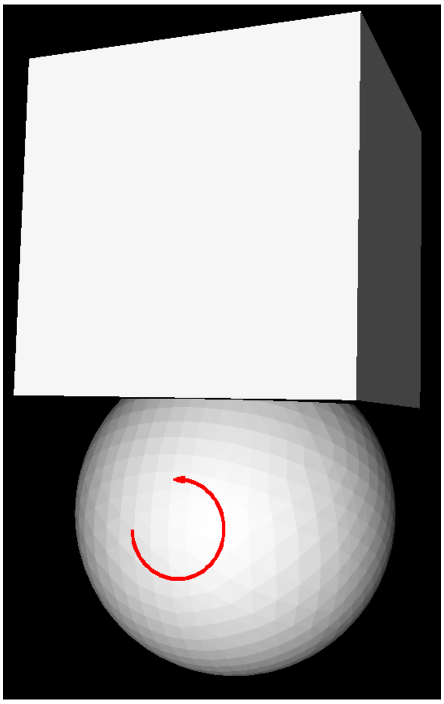

Figure 29: The second group of scenarios, first case: radial collision between sphere and cube.

In the second case, we tried to compare the methods in more practical scenarios. We tested the methods for the hip meshes created in a work done by (Chegini, Beck and Ferguson 2006).[6] The collision between femoral cartilage and the combination of pelvis cartilage and labrum was detected during the femur’s rotation and sliding (see Figure 30). All of the tests were done for four different resolutions of the objects (see Table 3), by applying the best mapping function found during testing the first case of the second scenario.

---

[6] The 3D Meshes were prepared by Salman Chegini at ARTORG, University of Bern.

Table 2: Different resolution of the objects in the second group of scenarios, first case.

	Scen.5	Scen.6	Scen.7	Scen.8	Scen.9	Scen.10	Scen.11
Number of vertices in the sphere's mesh	66	258	1 026	4 098	16 386	65 538	262 146
Number of vertices in the cube's mesh	98	386	1 538	6 146	24 578	24 578	24 578
Total number of vertices (mesh)	**164**	**644**	**2 564**	**10 244**	**40 964**	**90 116**	**286 724**
Number of triangles in the sphere's mesh	128	512	2 048	8 192	32 768	131 072	524 288
Number of triangles in the cube's mesh	192	768	3 072	12 288	49 152	49 152	49 152
Total number of triangles (mesh)	**320**	**1 280**	**5 120**	**20 480**	**81 920**	**180 224**	**573 440**
Number of tetrahedra in the sphere's volume	128	512	2 048	8 617	35 361	136 265	550 150
Number of tetrahedra in the cube's volume	412	1 909	7 807	30 021	116 199	116 199	116 199
Total number of tetrahedra (volume)	**540**	**2 421**	**9 855**	**38 638**	**151 560**	**252 464**	**666 349**

Same as the first group, the objects (for both cases) were made up of 3D triangular meshes; and for testing the spatial hashing method, objects' volumes were tetrahedralized. In each scenario, for the proposed radial segmenting method and also the ray-based sampling method, the table sizes were chosen the same (for all types of mapping functions).

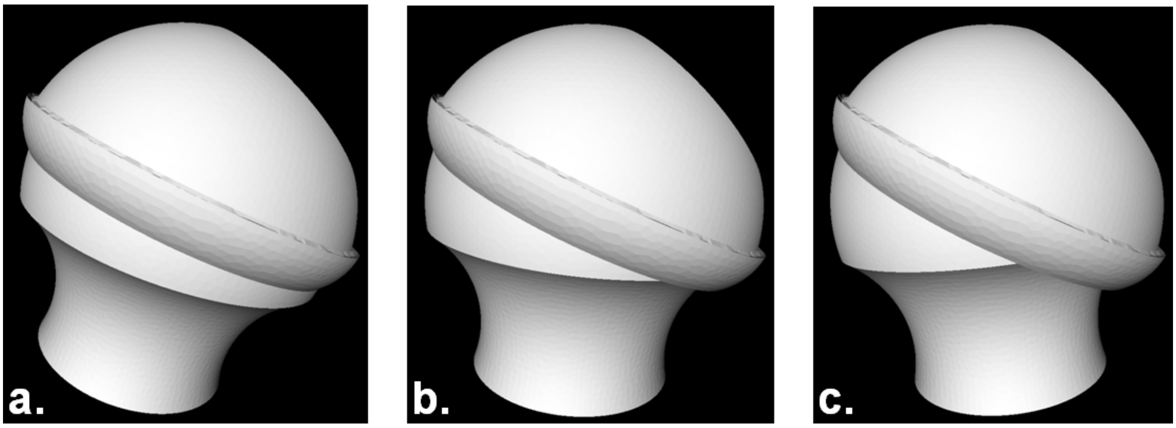

Figure 30: The second group of scenarios, second case: radial collision between 'femoral cartilage' and 'the combination of pelvis cartilage and labrum'.

3.5 Results

The processing time for each collision detection method in each scenario was measured. These measurements were done for the pre-processing and the main processing of the methods separately, in order to provide a better comparison. The pre-processing stage for the cylindrical segmenting method and the radial segmenting method is considered similar to what has been explained in section 3.2.3 and section 3.3.4, respectively. For ray-based sampling method and the spatial hashing method, the pre-processing is considered as the stage used for creating the related tables. As it was discussed before, the pre-processing steps may need to be repeated in case the objects are deformed in any arbitrary direction. Thus, the total computation times should be also compared with each other. Since the applied OBVH method returns the collision answer without any details, we only compared its total time of processing.

Table 3: Different resolution of the objects in the second group of scenarios, second case.

	Scen.12	Scen.13	Scen.14	Scen.15
Number of vertices in the femur cartilage's mesh	1 414	2 171	3 869	8 649
Number of vertices in the pelvis cartilage and labrum's mesh	1 438	2 111	3 935	7 412
Total number of vertices (mesh)	**2 852**	**4 282**	**7 804**	**16 061**
Number of triangles in the femur cartilage's mesh	2 824	4 338	7 734	17 294
Number of triangles in the pelvis cartilage and labrum's mesh	2 876	4 222	7 870	14 824
Total number of triangles (mesh)	**5 700**	**8 560**	**15 604**	**32 118**
Number of tetrahedra in the femur cartilage's mesh	35 269	53 824	78 895	147 094
Number of tetrahedra in the pelvis cartilage and labrum's mesh	36 291	42 420	84 855	109 272
Total number of tetrahedra (volume)	**71 560**	**96 244**	**163 750**	**256 366**

In the first group of scenarios OBVH, spatial hashing and the proposed cylindrical segmenting method were tested in each scenario and the results are demonstrated in Figure 31, Figure 32, and Figure 33. In all of these figures, the bars are normalized based on the processing time related to cylindrical segmenting method.

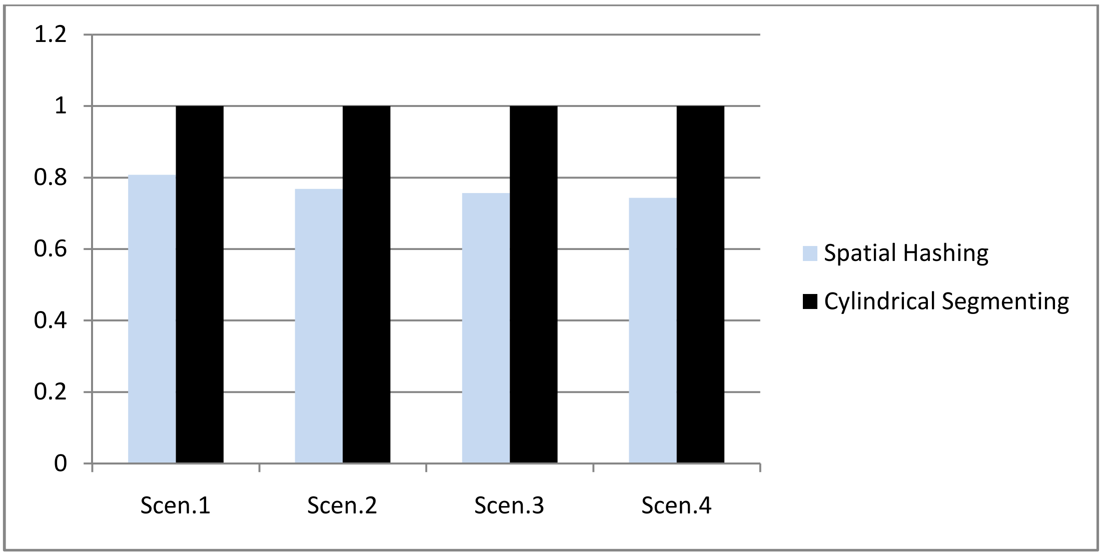

Figure 31: Pre-processing computational time for spatial hashing method and the proposed cylindrical segmenting method. The values are normalized by the values related to the cylindrical segmenting method. The pre- processing time corresponding to cylindrical segmenting method is 16.58, 58.72, 217.43 and 817.68 millisecond for Scenarios 1, 2, 3 and 4, respectively. Vertical and horizontal axes stand for 'pre-processing processing time (normalized)' and 'testing scenarios'.

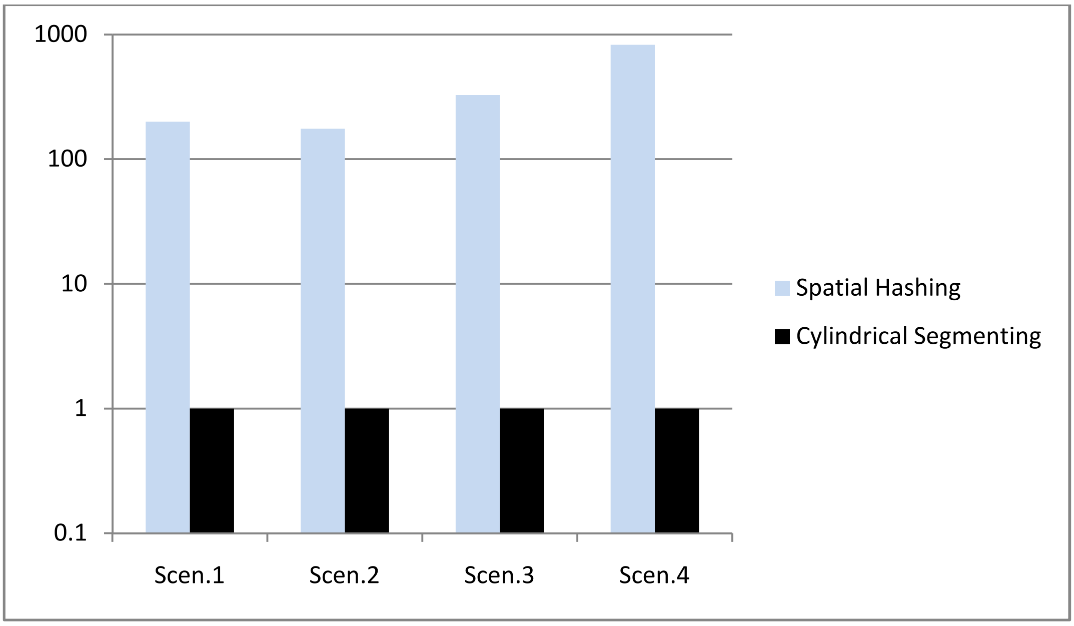

Figure 32: Main processing computational time for spatial hashing method and the proposed cylindrical segmenting method. The values are normalized by the values related to the cylindrical segmenting method and the bars are illustrated in logarithmic scale. The main processing time corresponding to cylindrical segmenting method is 0.16, 0.76, 2.85 and 10.60 millisecond for Scenarios 1, 2, 3 and 4, respectively. Vertical and horizontal axes stand for 'main processing time (normalized)' and 'testing scenarios'.

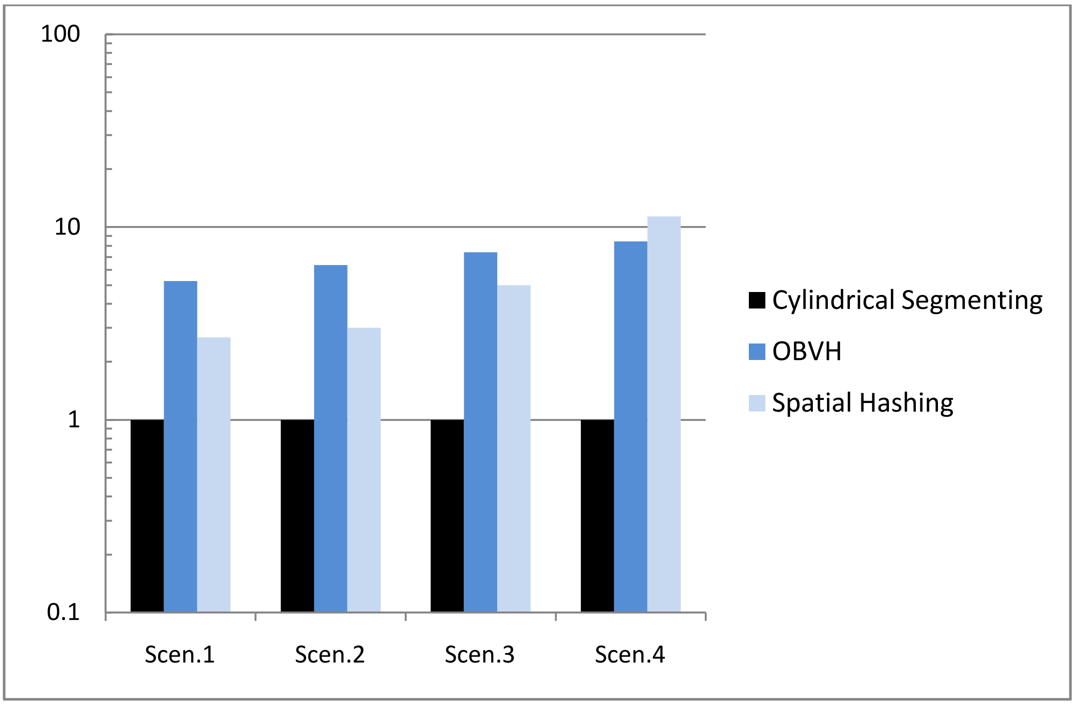

Figure 33: Total processing computational time for spatial hashing method, OBVH and the proposed cylindrical segmenting method. The values are normalized by the values related to the cylindrical segmenting method and the bars are illustrated in logarithmic scale. The total processing time corresponding to cylindrical segmenting method is 16.74, 59.48, 220.28 and 828.28 millisecond for Scenarios 1, 2, 3 and 4, respectively. Vertical and horizontal axes stand for 'total processing time (normalized)' and 'testing scenarios'.

In the second group, OBVH, spatial hashing, ray-based sampling method, and the proposed radial segmenting method (for all the three types of the mapping function) were tested in scenarios 5 to 11 (first case). The computational times are listed in Table 4, Table 5, and Table 6.

Table 4: Pre-processing computational time for spatial hashing method, ray-based sampling method and the proposed radial segmenting method (for 3 types of mapping function: type 1: based on spherical coordinates; type 2: based on normalized Cartesian coordinates; type 3: based on quasi-uniform radial segmentation); all the values are in millisecond.

	Scen.5	Scen.6	Scen.7	Scen.8	Scen.9	Scen.10	Scen.11
Spatial Hashing	0.30	0.91	3.64	16.88	71.62	243.3	976.4
Ray-based sampling	4.61	53.29	603.5	9 165	140 034	138 761	138 470
Radial Segmenting (mapping function type 1)	0.44	0.90	3.60	13.65	72.65	69.01	76.56
Radial Segmenting (mapping function type 2)	0.39	0.78	2.47	9.59	34.27	38.00	36.79
Radial segmenting (mapping function type 3)	0.93	2.87	11.14	38.92	161.6	168.0	153.8

Table 5: Main processing computational time for spatial hashing method, ray-based sampling method and the proposed radial segmenting method (for 3 types of mapping function: type 1: based on spherical coordinates; type 2: based on normalized Cartesian coordinates; type 3: based on quasi-uniform radial segmentation); all the values are in millisecond.

	Scen.5	Scen.6	Scen.7	Scen.8	Scen.9	Scen.10	Scen.11
Spatial Hashing	1.17	5.54	40.41	323.1	2 983	23 088	100 862
Ray-based sampling	0.031	0.10	0.34	1.35	6.58	20.10	69.99
Radial Segmenting (mapping function type 1)	0.055	0.18	1.04	7.48	50.52	155.6	559.5
Radial Segmenting (mapping function type 2)	0.060	0.16	0.66	5.24	21.29	61.52	167.9
Radial segmenting (mapping function type 3)	0.19	0.71	1.66	9.65	41.75	135.0	437.0

Table 6: Total processing computational time for spatial hashing method, OBVH (returning the collision result without any details about the colliding elements), ray-based sampling method and the proposed radial segmenting method (for 3 types of mapping function: type 1: based on spherical coordinates; type 2: based on normalized Cartesian coordinates; type 3: based on quasi-uniform radial segmentation); all the values are in millisecond

	Scen.5	Scen.6	Scen.7	Scen.8	Scen.9	Scen.10	Scen.11
Spatial Hashing	1.47	6.45	44.06	340.0	3 055	23 331	101 838
OBVH	5.63	22.36	103.5	426.1	1 962	4 611	15 941
Ray-based sampling	4.64	53.39	603.8	9 166	140 040	138 781	138 540
Radial Segmenting (mapping function type 1)	0.49	1.08	4.65	21.13	123.2	224.6	636.1
Radial Segmenting (mapping function type 2)	0.45	0.94	3.13	14.83	55.56	99.52	204.7
Radial segmenting (mapping function type 3)	1.12	3.58	12.80	48.58	203.4	303.0	590.8

For the seconds case (scenarios 12 to 15), the comparison among OBVH, spatial hashing, ray-based sampling method, and the proposed radial segmenting method by using the 2nd type of mapping function (normalized Cartesian) was done (see Figure 34, Figure 35, and Figure 36).

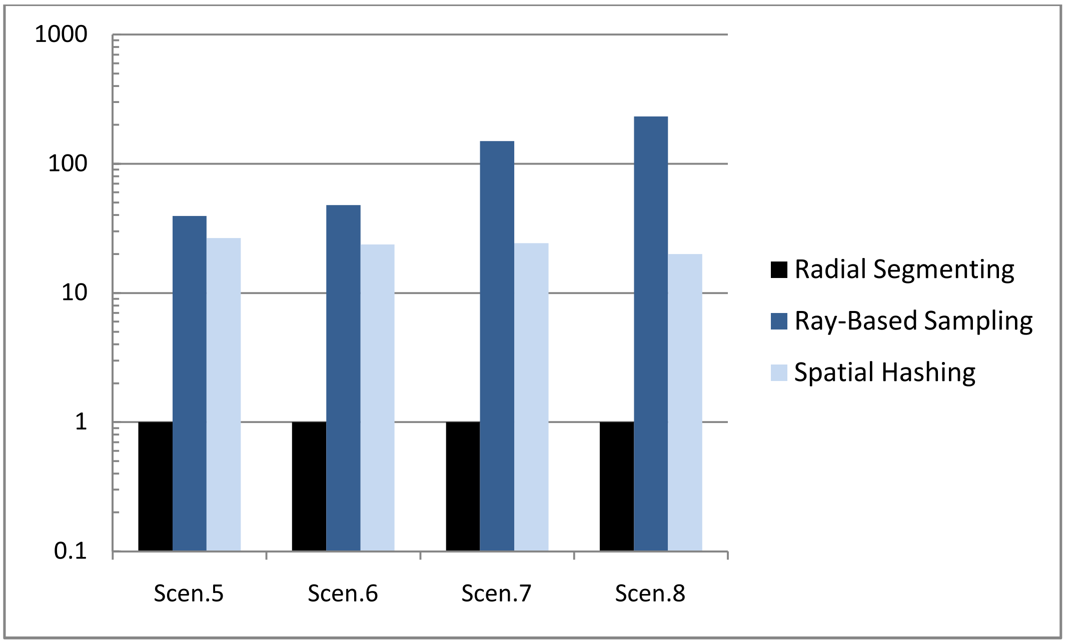

Figure 34: Pre-processing computational time for spatial hashing method, ray-based sampling method and the proposed radial segmenting method (using normalized Cartesian mapping function (2nd type)). The values are normalized by the values related to the radial segmenting method and the bars are illustrated in logarithmic scale. The pre-processing time corresponding to radial segmenting method is 1.83, 3.27, 4.86 and 10.61 millisecond for Scenarios 12, 13, 14, and 15, respectively. Vertical and horizontal axes stand for 'pre- processing time (normalized)' and 'testing scenarios'.

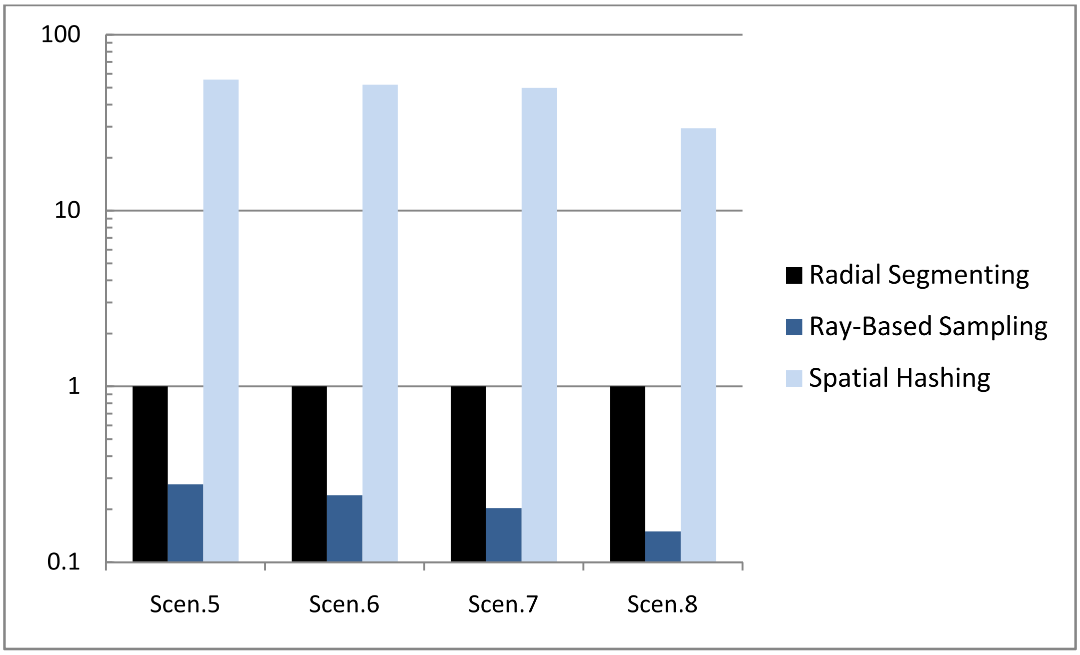

Figure 35: Main processing computational time for spatial hashing method, ray-based sampling method and the proposed radial segmenting method (using normalized Cartesian mapping function (2nd type)). The values are normalized by the values related to the radial segmenting method and the bars are illustrated in logarithmic scale. The total processing time corresponding to radial segmenting method is 1.88, 2.86, 6.15 and 17.57 millisecond for Scenarios 12, 13, 14, and 15, respectively. Vertical and horizontal axes stand for 'main processing time (normalized)' and 'testing scenarios'.

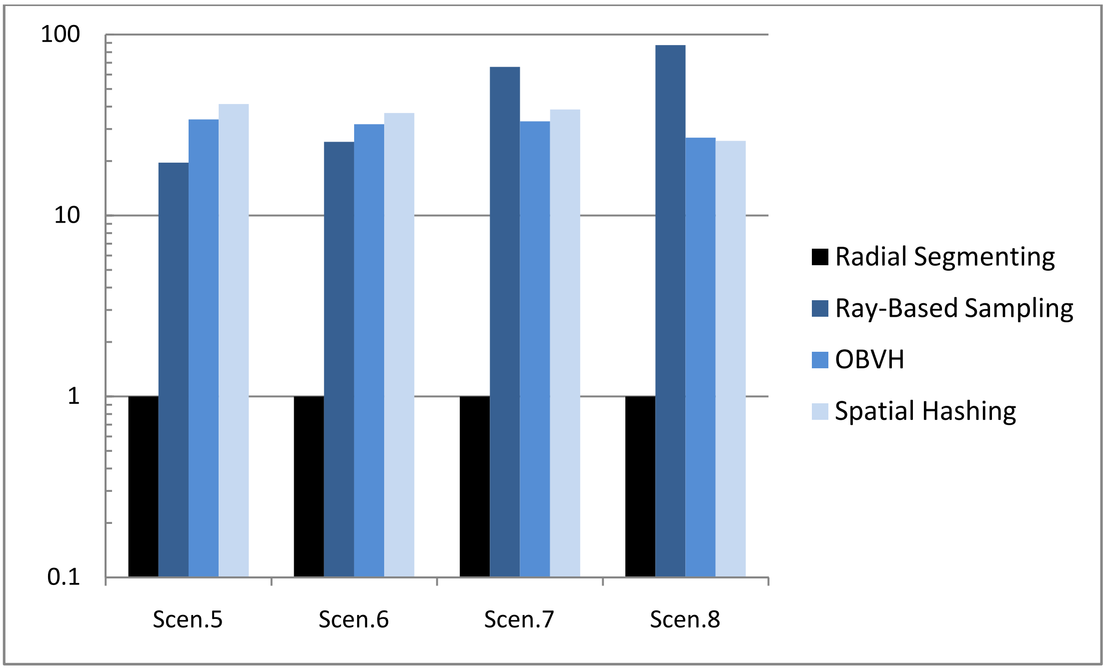

Figure 36: Total processing computational time for spatial hashing method, OBVH, ray-based sampling method and the proposed radial segmenting method (using normalized Cartesian mapping function (2nd type)). The values are normalized by the values related to the radial segmenting method and the bars are illustrated in logarithmic scale. The total processing time corresponding to radial segmenting method is 3.71, 6.14, 11.01 and 28.18 millisecond for Scenarios 12, 13, 14, and 15, respectively. Vertical and horizontal axes stand for 'total processing time (normalized)' and 'testing scenarios'.

For having a good comparison, we used the same value of the tables' size, in all of our proposed methods and ray-based sampling method. In ray-based sampling method, increasing the dimension of the table helps to store the information of the fixed object more precisely which can cause the method to return more accurate results, but it can slow down the pre-processing stage significantly. However, since the main processing stage is working based on comparing the position of each mobile vertex with only one fixed polygon stored in the corresponding table-cell, increasing the table's size does not have any special influence on the main processing time.

On the other hand, table's dimension does not affect the accuracy of the proposed radial segmenting methods (they always return the accurate results), but it can decrease the main processing time and increase the pre-processing time. The main processing time is decreased, because we will have smaller number of fixed polygons stored in each table-cell and consequently smaller number of comparison is needed to be done for each mobile vertex. The effect of table size on processing time of the methods based on space segmentation has been already investigated (Teschner, Heidelberger, et al. 2003). However for investigating the effect of table dimension on the proposed radial method for all three types of mapping function, we repeated the same scenarios of 5 to 11 (first case, second group) with about 6 times larger size of table comparing to the previous experiments. The results are shown in Figure 37, Figure 38, and Figure 39.

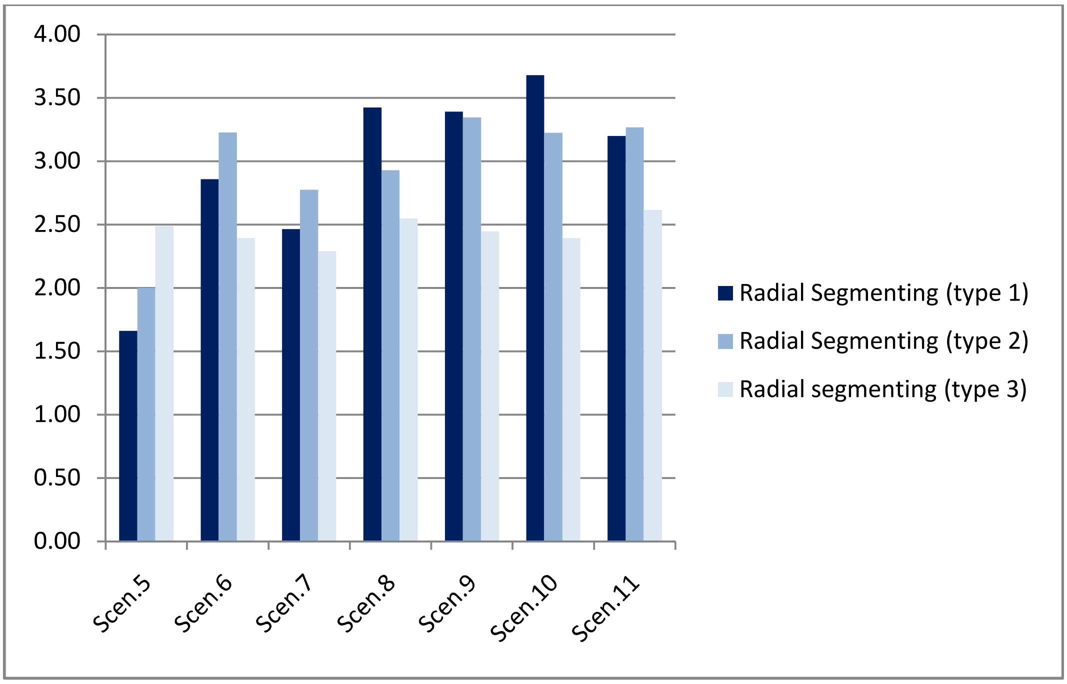

Figure 37: Pre-processing computational time for the proposed radial segmenting method with two different table's size (for 3 types of mapping function: type 1: based on spherical coordinates; type 2: based on normalized Cartesian coordinates; type 3: based on quasi-uniform radial segmentation). The bars are showing ratio of the "pre-processing time"s ('when the table is six times larger than its previous size' to 'when the table has its previous size').

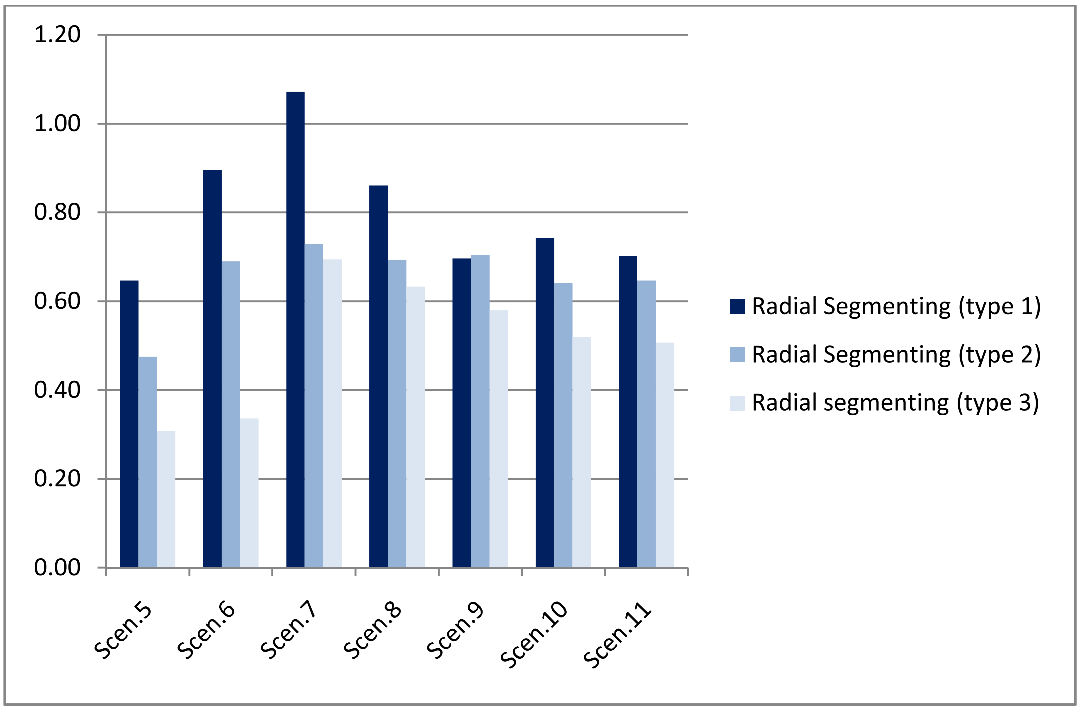

Figure 38: Main processing computational time for the proposed radial segmenting method with two different table's size (for 3 types of mapping function: type 1: based on spherical coordinates; type 2: based on normalized Cartesian coordinates; type 3: based on quasi-uniform radial segmentation). The bars are showing ratio of the "main processing time"s ('when the table is six times larger than its previous size' to 'when the table has its previous size').

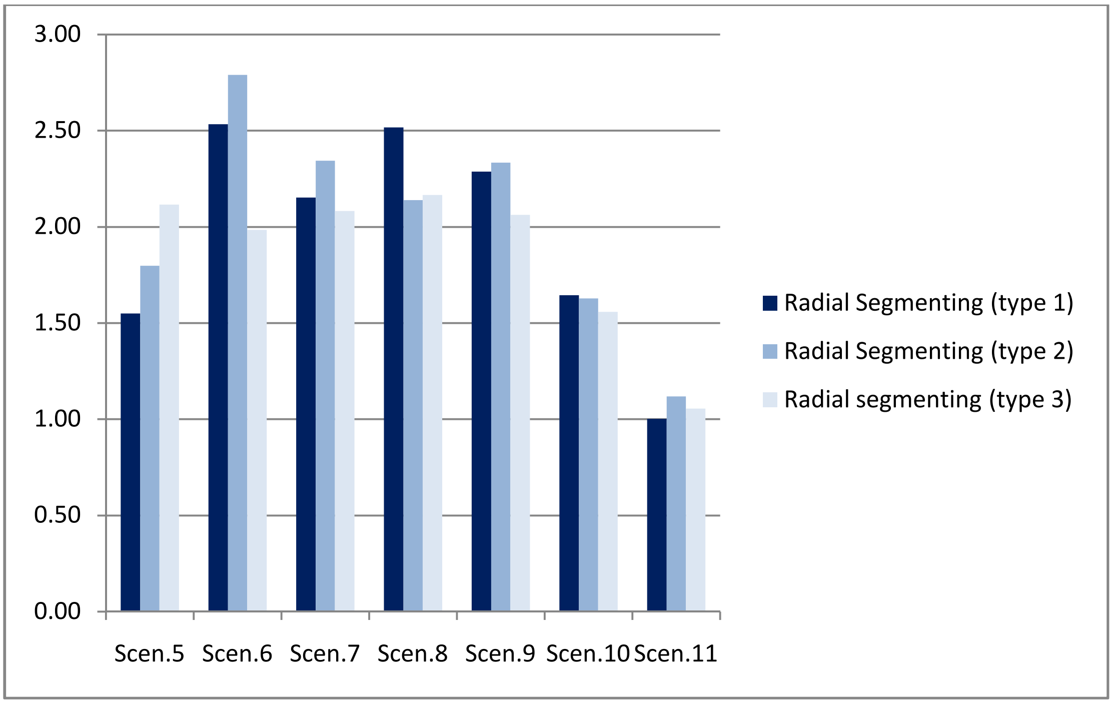

Figure 39: Total processing computational time for the proposed radial segmenting method with two different table's size (for 3 types of mapping function: type 1: based on spherical coordinates; type 2: based on normalized Cartesian coordinates; type 3: based on quasi-uniform radial segmentation). The bars are showing ratio of the "total processing time"s ('when the table is six times larger than its previous size' to 'when the table has its previous size').

3.6 Discussion

Based on the results, it could be seen that the proposed methods in both striking and sliding case are performing faster than the other methods. As it can be seen, the total processing time of the proposed cylindrical segmenting and radial segmenting methods can be respectively up to about one and two orders of magnitude faster than the other methods. It should be noticed that the applied OBVH method only returns whether there is a collision or not, without giving the details about colliding elements. Although cubic segmenting method (spatial hashing) returns the details about penetrating vertices, it is not able to return them as pair in order to find penetrating depth without any additional computation (vs. (Maciel, Boulic and Thalmann 2007) and our methods which return the colliding elements as pairs).

Both of the methods are able to be used accurately for different kinds of joint whose movement can be modeled as rotation (ball- socket joints such as hip joint and shoulder). In case the joint movement contains some non-negligible translations (e.g. knee), the radial segmenting collision detection method can still be used to return accurately the radial penetration depths.

The proposed radial segmenting collision detection method works by radial segmentation of the space around a fixed reference point in one coordinate system, without considering the type of movement. Thus, for rotating objects that reference point can be chosen as the joint center of rotation and for the sliding objects just choosing a point approximating a spherical center of the joint is enough. In order to find curvilinear penetration depth for the joints with non-negligible translations, the current cylindrical segmenting collision detection method has a potential to be extended. The extension should be done by checking the collision of a vertex in its nearby rings too (during negligible translation, only checking the vertex's containing ring is sufficient). In fact, based on the size of cylindrical partitions (i.e. ring) and the range of translation, we can decide how many neighboring cylindrical partitions of the partition containing a candidate vertex should be checked for discovering possible collision of the vertex.

3.6.1 Striking case

In striking case, the pre-processing time of the cylindrical segmenting method is about 1.3 times of the spatial hashing pre-processing time (almost the same order of magnitude) (see Figure 31). But in the main processing stage, the computation can be done more than two orders of magnitude faster than spatial hashing (hundreds of times faster) (see Figure 32). The reason is due to the fact that in the cylindrical segmenting method, the spatial segments are similar to cylindrical rings which are in the same orientation of the rotation. Therefore, it helps us to consider only the vertices and polygons which may collide with each other during the rotation.

In case the objects have no deformation or their deformation is in the angular direction, there is no need to perform the pre-processing stage for the cylindrical segmenting method, during the simulation. Otherwise, the pre-processing should be done for the deformed polygons and vertices. In the worst case when the all of the polygons of both of the objects are largely deformed, the comparison must be done based on the total computational time. By comparing the total computational time of the proposed cylindrical segmenting method with spatial hashing and OBVH (see Figure 33), it can be seen that the proposed method can be about one order of magnitude faster than spatial hashing and OBVH. It should be noticed again that the tested OBVH method, is only returning the collision results without any details about the colliding vertices or polygons, while both of the proposed methods and spatial hashing method returns the information about penetrating elements. In addition to the speed, the proposed method is also returning the penetrating elements as pairs, which can be very useful for having the penetrating depth (in angular direction), without any additional calculations. Also, while we need to repeat the pre-processing stage in the proposed method only if the objects are deformed in the non-angular direction, the pre-processing stage in spatial hashing method must be repeated for deformations in any direction.

In the proposed method, if the rotating axis changes, the pre-processing must be repeated again. By checking the results (see Figure 32 and Figure 33), it can be noticed that even the total processing time in the cylindrical segmenting method is up to ten times lower than the main processing time in the spatial hashing method.

3.6.2 Sliding case

In pre-processing stage, as the objects resolution increases the proposed radial segmenting method (for all three types of the mapping functions), take a smaller amount of computational time compared to the other methods (see Table 4). The pre-processing stage can be more than one order of magnitude faster in the proposed method compared to the spatial hashing method, and more than two orders of magnitude faster compared to the ray-based sampling method in high resolution cases (see Table 4 and Figure 34). Among different types of mapping function, using mapping function type 2 which is based on simple formulations (normalized Cartesian coordinates) shows a faster performance (see Table 4). It should be noticed that, while in the proposed radial segmenting method and the ray-based sampling method, the pre-processing time is depending only on the resolution of the fixed object; in the spatial hashing method the pre-processing is affected by the resolution of the mobile object too, (scenarios 9, 10 and 11).

In the main processing stage the spatial hashing method is about two orders of magnitude slower than the proposed radial segmenting method and the ray-based sampling method (see Table 5 and Figure 35). Among different types of the mapping functions, again using mapping function type 2 (simple formulations: normalized Cartesian coordinates), shows a faster performance compared to the other types (see Table 5). The proposed radial segmenting method with the second type mapping function is between 1.5 to 6.7 times slower than the ray-based sampling method (see Table 5 and Figure 35). This increase of speed in the main processing of the ray-based sampling method compared to the proposed radial segmenting method (type 2), is understandable when we notice that the ray-based sampling method is simplified in the main stage which makes it an approximated method that may not give the accurate results, while the proposed method is doing all the comparisons completely without any approximation in the main stage which causes the results to be totally accurate.

In case the fixed object has no deformation or its deformation is in the radial direction, there is no need to perform the pre-processing stage for the ray-based sampling method and the proposed radial segmenting method, during the simulation. Otherwise, the pre-processing stage must be considered as part of the main processing and repeated in each simulation step (the pre-processing stage in spatial hashing method must be repeated for deformations in any direction). In this case the comparison must be done based on the total computational time. By comparing the total computational time of the proposed radial segmenting methods with spatial hashing, ray-based sampling and OBVH, it can be seen (see Table 6 and Figure 36) that the proposed method can be performed up to more than two orders of magnitude faster than spatial hashing and ray-based sampling methods. Also the proposed radial segmenting method can be more than one order of magnitude faster than OBVH (see Table 6 and Figure 36), while OBVH method is only returning the collision results without any details about the colliding vertices or polygons. In addition to the speed, the proposed method and the ray-based sampling method are also returning the penetrating elements as pairs (to be used for estimating the penetration depth).

It was also seen, when we change the size of table, it affects the pre/main processing time in the proposed radial segmenting methods (see Figure 37 and Figure 38). As it could be seen in Figure 39, sometimes the total processing time is almost the same for different sizes of table. It is clear if we increase the size of table continuously, we will finally come to a limit where the main processing time remains constant and only the pre-processing time is increasing. It can happen when using the smaller size of grids cannot help to store less information of the fixed polygons in table cells any more. Based on the applications, one can decide what size of table is suitable. For example if pre-processing step is not going to be repeated too much during the simulation, one can use as large as possible size of table. In the results it could be also seen that increasing the size of table has stronger influence in improving the main processing speed of the method using the third type mapping function (uniform radial segmentation). It can be due to the uniformity of the third type function, which makes the method more dependent on the size of grids, while in the other types, due to their non-uniformity, some spatial cells are already too small so that increasing the table's size may not reduce the number of stored fixed polygons.

Usually we do not need to consider normals of the polygons when using radial segmenting collision detection method. However, in some cases when the fixed object is irregular in term of convexity (see Figure 40), the normals should be considered to avoid any miss-collision detection.

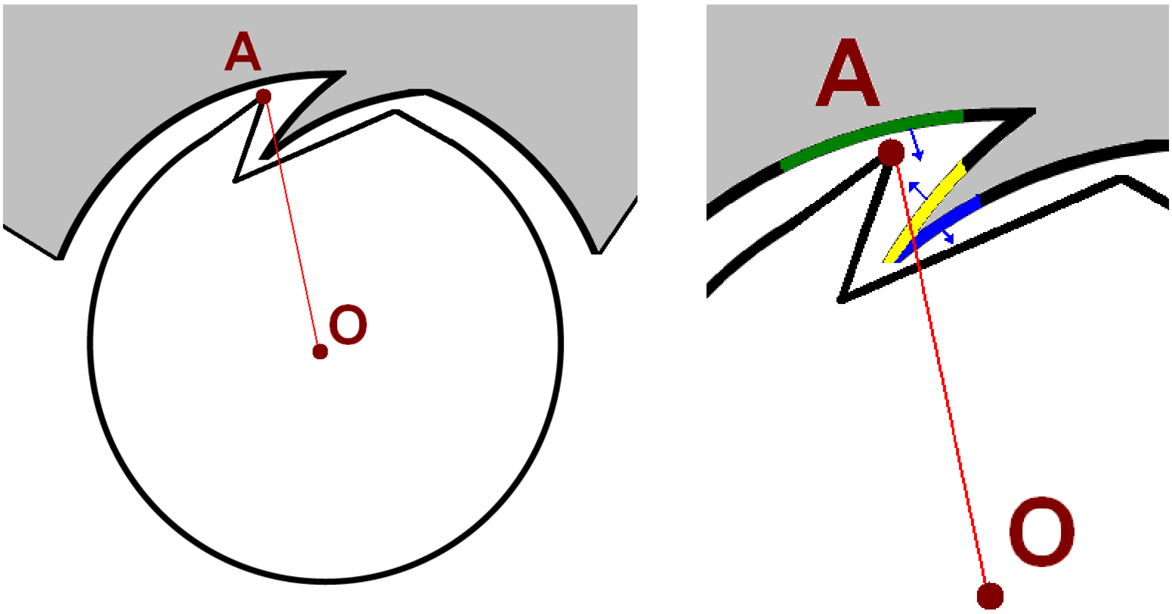

Figure 40: Irregular sliding case. Left: The white object is rotating about O and can have a collision with a gray fixed object. Vertex A of the mobile object is not penetrating the gray fixed object; however without considering the normals it may be returned as a penetrated vertex. Right: The area around vertex A is zoomed. The vector connecting center of rotation ('O') to the vertex A, passes through 3 polygons on the fixed object (shown by blue, yellow, and green lines). The thin blue arrows represent the normals of these polygons. Without considering the normals, vertex A will be found penetrating blue and yellow polygons, mistakenly. But, by considering the normals of the yellow and green polygons, it will be known that the vector A is out of the fixed object.

In fact, in such irregular cases a mobile vertex can be located between two layers of the fixed object. Thus, pure distance calculation cannot always ensure whether the mobile vertex is penetrating the fixed object or not. In this case, the normals can help to know whether the vertex is inside the fixed object or it is just trapped between two different layers of it. When the mobile vertex is inside the fixed object, normals are pointing out of it. On the other hand, when the mobile vertex is just between two layers of the fixed object, the normals are pointing toward it (see Figure 40).

3.6.3 Striking vs. Sliding

The method proposed for striking case is able to return the collision information during sliding too. In fact the penetrating mobile vertices during the rotation can be found by using both of the methods for sliding and striking, but the fixed polygons corresponding to each penetrating mobile vertex are not necessarily the same in both of the methods (see Figure 41). So by this explanation, we need to know which method should be used for a certain case. Of course, based on the application, the user can decide which method is better to be applied, but sometimes the object is both striking and sliding during its rotation. That means some parts of the mobile object is striking during the rotation, while some other parts are sliding during the rotation. For such cases we can virtually divide the mobile object to two parts: 1-sliding part, 2-striking part. After such division, the method related to each part is applied only for the same part (i.e. cylindrical segmenting method for the striking part and radial segmenting method for the sliding part).

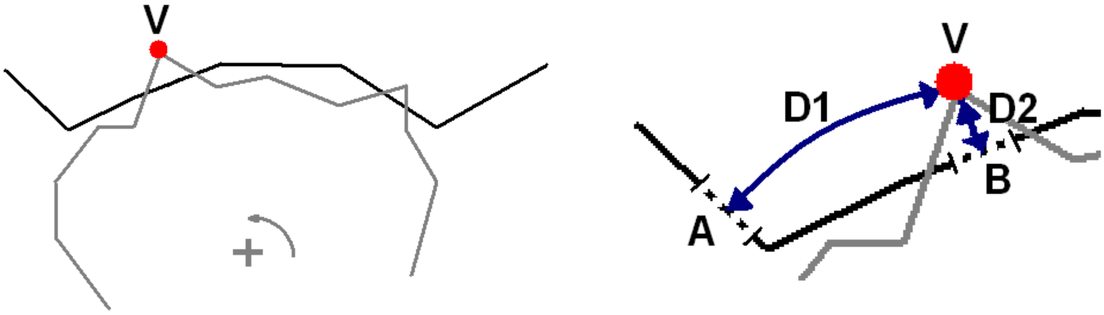

Figure 41: Sliding vs. Striking. Left: gray object is rotating about '+' and is colliding with the outer black object. Vertex V of the mobile object is penetrating the fixed object. Right: By using the cylindrical segmenting method, the pair of vertex V and polygon A (shown by a dotted line) with the penetration depth of D1 are found. By using radial segmenting method, the pair of vertex V and polygon B (shown by a dotted line) with the penetration depth of D2 are found.

Such division can be done by using the distance between the vertices to the center of rotation. Depending on the application we can define a threshold value for the distance. If the distance between a mobile vertex and the center of rotation is smaller than the threshold, the vertex is checked by the radial segmenting method for the collision. On the other hand if the distance between a mobile vertex and the center of rotation is larger than the threshold, the vertex is checked by cylindrical segmenting method for the collision (see Figure 42). Depending on the objects one may need to define multiple thresholds. It should be noticed that using both of the methods returns the correct penetrating vertices, and our reason for defining the threshold(s) is to obtain the appropriate penetrating depths during collision detection.

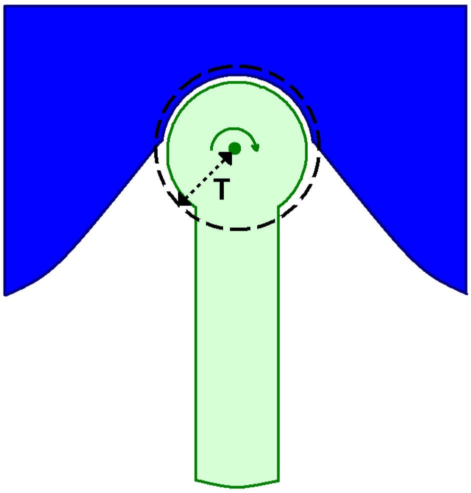

Figure 42: Division of the mobile object to sliding and striking parts. The dashed circle is used for such division as a threshold. The mobile vertices whose distance to the center of rotation is smaller than T are considered as the sliding vertices and the ones whose distance is larger than T are considered as the striking vertices.

Beside of obtaining an appropriate penetration depth by using the appropriate methods for different parts of mobile objects, the total speed for collision detection can be also increased. The reason is that each method is working efficiently for its related case. In other words, the method for sliding is working based on radial segmentation which is more appropriate for sliding cases in term of computational speed, and the method for striking is working based on cylindrical segmentation which is also more appropriate for striking cases.

3.6.4 Complexity

If v, p and t are the number of vertices, surface polygons and tetrahedra respectively, the complexity is O(p)+O(v) for both of the proposed methods. Since the complexity of the proposed methods is a linear function of the number of elements, the proposed methods are more efficient in term of complexity compared to OBVH (O(p×log(p))) and naive methods (O(p×p)). In other words, when the number of object elements increase, the number of calculations in the proposed methods increases linearly, but the number of calculations in OBVH and naive methods increases more significantly due to the nonlinear dependency of the number of calculations to the number of elements in these methods. In cubic segmenting method (spatial hashing), the complexity is O(v)+O(t). ‘t’ is either in the same order of ‘p’ (for very thin objects) or higher than it. That means the number of elements considered for calculations in cubic segmenting method (spatial hashing) is either equal or more than the number of elements needed for the calculations in the proposed methods. Thus, the proposed methods are faster than cubic segmenting method too. Of course, it should be considered again that OBVH and cubic segmenting method are designed for general proposes, and cubic segmenting method is able to returns internal penetrating vertices too, which may be useful in some graphical applications.

3.6.5 Penetration depths

During a collision, one or more vertices of an object may penetrate inside another object. Penetration depth of an object’s vertex is amount of displacement needed for pulling the vertex out of the other object, in order to resolve the collision. In many applications, after finding the penetrating vertices by a collision detection method, another computation is done in order to calculate the penetration depths of the vertices. The penetration depths are usually calculated by finding the smallest distance between the penetrated elements (e.g. vertices) and the surface of the other object (Schneider and Eberly 2003). The methods we propose are able to calculate the penetration depths at the collision detection stage, without performing any separate calculation. Considering the radial direction and circular arc for calculating the penetration depths, instead of finding the minimum distance, helps to prevent possible inconsistency of the calculated penetration depths. As an example, Figure 43 shows how the penetration depths calculated in the radial directions are more robust for rough surfaces. In addition to the robustness, calculating the penetration depths along radial direction and circular arc provides more meaningful interpretations of the penetration depths in some applications (e.g. human joints) compared to the penetration depths based on the smallest distance. For example, cylindrical segmenting collision detection method returns the penetration depth along the circular arcs induced by ongoing rotation axis, which represents the amount of curvilinear depth that a vertex has penetrated along its rotational trajectory. Regarding the radial segmenting collision detection, if the approximated center of rotation is considered as the fixed reference point, the method returns the amount of penetration, perpendicular to the rotational path. The advantage of directional penetration depth is explained in section 4.3 for a biomedical application.

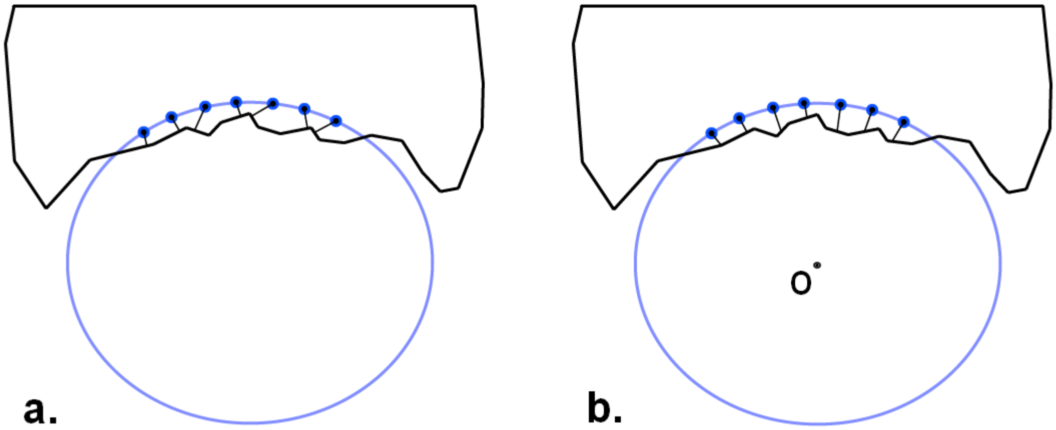

Figure 43: Computing the penetration depths of the vertices in the light blue object, based on (a) minimum distance (usually along the normals of the other surface) and (b) radial distance (the penetrated vertices of the light blue object are shown by small dots).

3.7 Applications

3.7.1 Biomedical

For diagnosing some of the human joint diseases, it is important to examine the joint according to some specific protocols. For instance, the flexion adduction internal rotation test is used to aid in the diagnosis of femoroacetabular impingement (Martin 2005). There can be different protocols for examining different kinds of pathology. For a joint simulation, the desired medical rotation can be interactively applied on the joint by rotating the virtual limbs about the joint estimated center of rotation (Kang, Sadri, et al. 2003). As a biomedical experiment, we tested our methods on 3D triangular hip meshes (femur and pelvis bones, cartilages and labrum) obtained by segmenting MR Images of a patient’s right hip (Gilles, Moccozet and Magnenat-Thalmann 2006), (Kang, Sadri, et al. 2003).[7] The proposed collision detection methods were continuously used while interactively rotating the virtual hip models about different medical axes of rotation. Radial segmenting method was used for calculating radial penetration depth when femur cartilage slid under labrum and pelvis cartilage. Cylindrical segmenting method was used for calculating curvilinear penetration depth when femur bone collided with labrum (Arbabi, Boulic and Thalmann 2009).

All the process for detecting collision and calculating penetration depth in the mentioned directions were done in real time by using the same PC of section 3.4. The results of the simulation during hip Abduction are illustrated in Figure 44, Figure 45, Figure 46, by coloring different areas of tissues based on their penetration depth.

---

[7] The 3D Meshes were prepared by Benjamin Gilles at MIRALab, University of Geneva.

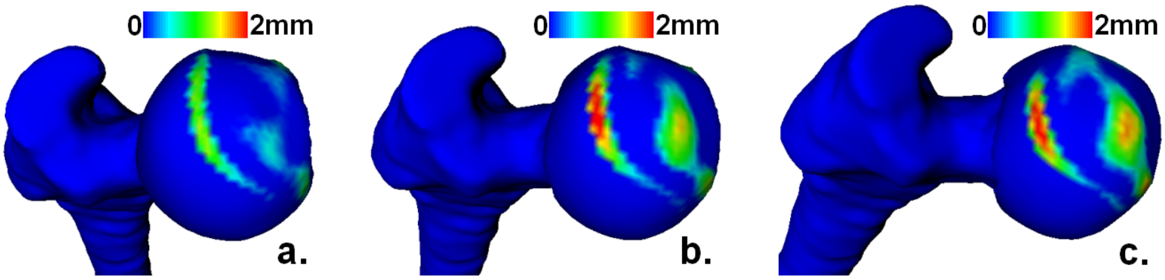

Figure 44: Coloring femur cartilage based on its radial penetration depth during Abduction, when it penetrates pelvis cartilage and labrum. Pelvis cartilage and labrum are transparent.

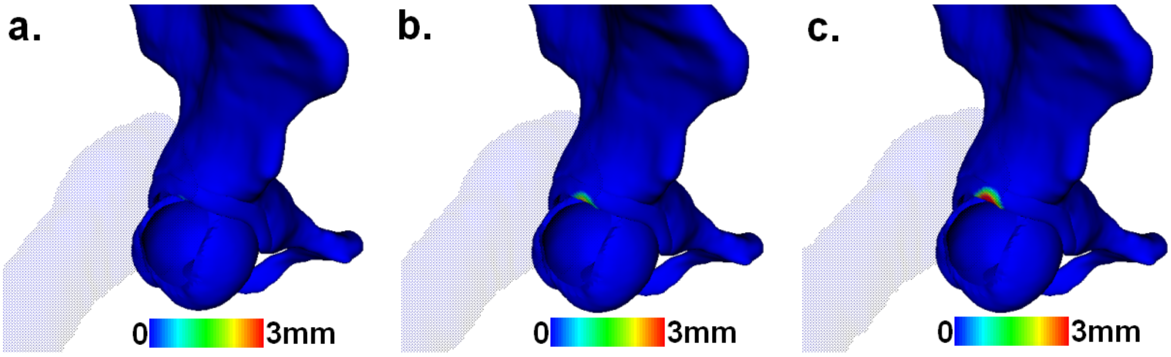

Figure 45: Coloring labrum based on its curvilinear penetration depth during Abduction, when femur bone collides with it. Femur bone is semi transparent.

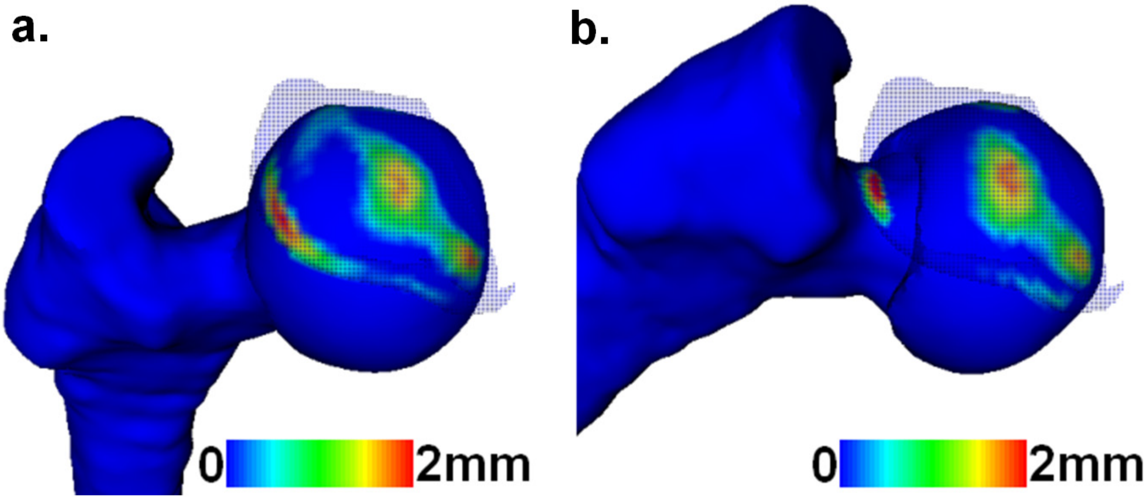

Figure 46: Curvilinear and radial penetration (together) during Abduction, (a) Femur cartilage penetrates pelvis cartilage and labrum, but femur bone is not colliding with labrum yet; (b) Femur cartilage still penetrates pelvis cartilage and labrum, while femur bone is also colliding with labrum. Pelvis cartilage and labrum are semi transparent.

Figure 47 and Figure 48 show the maximum values of penetration depth during Abduction and combined Abduction-External Rotation (external rotation after abduction), respectively.

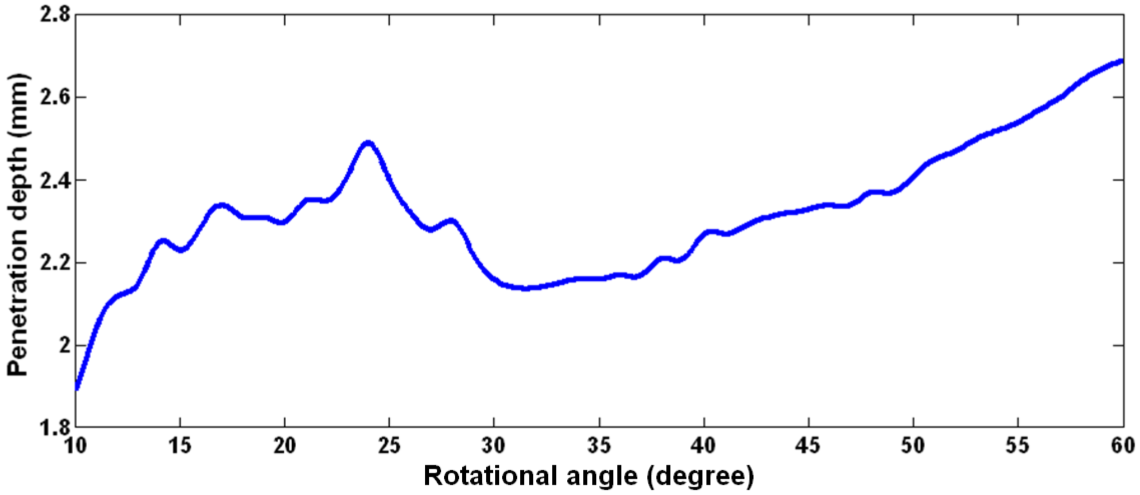

Figure 47: maximum radial penetration depth during Abduction, when femur cartilage penetrates pelvis cartilage and labrum.

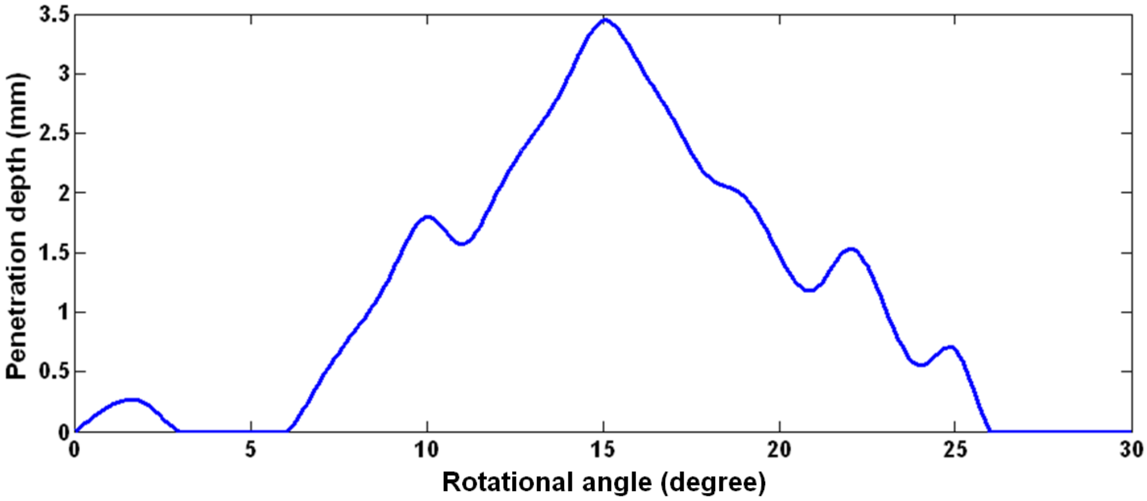

Figure 48: maximum curvilinear penetration depth during Abduction-External Rotation, when femur bone collides with labrum (the angles are related to external rotation, after abduction is done up to its limit).

The results of this biomedical test suggest that the proposed methods can be used for different kinds of medical applications. In general, one of the main applications is about helping the medical doctors to perform a better diagnosis and treatment, by checking the penetrations happening between tissues during joint movements. For example, if the penetration is high in some parts of a virtual tissue, it indicates that removing those parts in the real tissues can help to decrease the penetration and consequently reduce the stress.

The methods can be also applied for medical investigation of the joint contacts when the joint geometry changes due to abnormalities or replacement (by prostheses). For example, effects of tissues shapes in knee kinematics were analyzed by (Akalan, Ozkan and Temelli 2008). Also, (Lenaerts, et al. 2008) analyzed the effect of subject-specific modeling of hip geometry on hip contact forces during gait. Another analysis can be done by using the proposed methods to observe the changes in the penetration depths during joint motions too. In fact, while the force calculations can demonstrate the amount of stress that a joint is tolerating, it may not clearly find those parts of the joint tissues which hinder the motion and consequently cause the stress in the other parts. On the other hand, calculating the penetration depth, in addition to its faster process can highlight the parts of the joint tissues hindering the motion of the other parts (i.e. higher penetration indicates stronger hindering).

The proposed methods are able to address the geometrical calculations needed for the applications performing deformation and calculating stresses too. Iin different deformable models, usually before performing the mechanical calculations, the collision should be handled and the necessary information (such as penetrating vertices, penetration depth) should be provided as input to the model (Spillmann and Teschner 2005). Therefore, having the methods returning such information fast and accurately, improve the total speed of the calculations needed for deformation or stress calculations. Also, the returned penetration depths are either perpendicular (radial) or parallel (tangential) to the vertices rotational path. This helps to robustly estimate the forces along two locally orthogonal directions when the reconstructed surface models are noisy and/or rough due to pathologies (to compare with the penetration depths based on the minimal distance shown on Figure 43.a.).

More information about some of the potential biomedical applications is provided in the next chapter.

3.7.2 Others

The applications of proposed collision detection methods are not limited to medical cases, and they are capable to handle collision detection in some other computer graphics situations too. In this section we try to introduce their non-medical applications by providing some examples and testing the proposed methods on them.

In general, any graphical condition resembling rotation or sliding can take benefit from the proposed methods. Such conditions may be seen more frequently in mechanical simulations and robotics. For example Figure 49 illustrates a usability of cylindrical segmenting collision detection during moving a stone by a lever. Using cylindrical segmenting collision detection for this case not only aids to find collision rapidly but also retunes the amount of penetration happens by the lever inside the stone, in the same direction of its movement. Such, information can be useful for further deformations or other mechanical evaluations. Another example can be seen in Figure 50, when a wrench and a screw collide with each other. By using the cylindrical segmenting collision detection and calculating the penetrating parts, the mechanical evaluation of the regional stresses can be performed more easily to predict the possible responses of these instruments (such as the movement and regional deformations in the wrench and the screw).

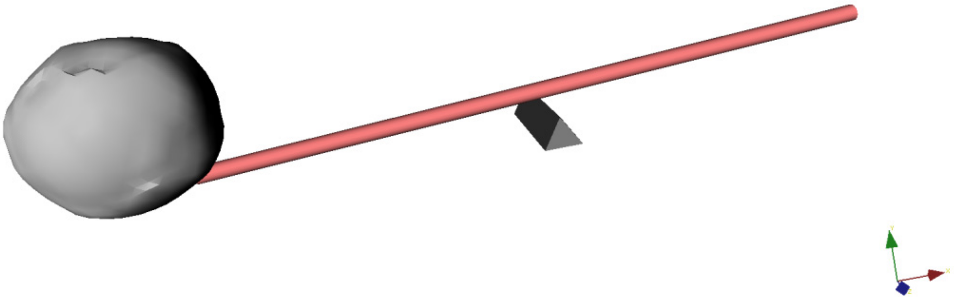

Figure 49: The red lever rotates about the gad and collides with the white stone (cylindrical segmenting collision detection). Number of polygons (lever, stone, total): 1056, 1280, 2336. Number of vertices (lever, stone, total): 530, 642, 1172. Pre-processing, main processing, and total processing time were 2.72, 0.024, and 2.75 milliseconds, respectively.

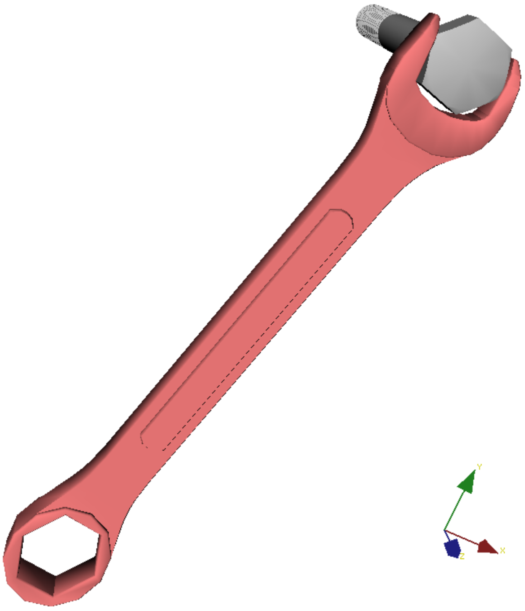

Figure 50: The wrench and screw collide during the rotation (cylindrical segmenting collision detection). Number of polygons (wrench, screw, total): 2878, 921, 3799. Number of vertices (wrench, screw, total): 1458, 461, 1919. Pre-processing, main processing, and total processing time were 35.48, 0.038, and 35.52 milliseconds, respectively.

Cylindrical segmenting collision detection method can be also applied in robotic applications when there are some levels of rotations (hierarchy). In such cases, the method must be used for each level of hierarchy separately. As an example, Figure 51 shows how the method can be applied for a robot designed for grabbing different objects.

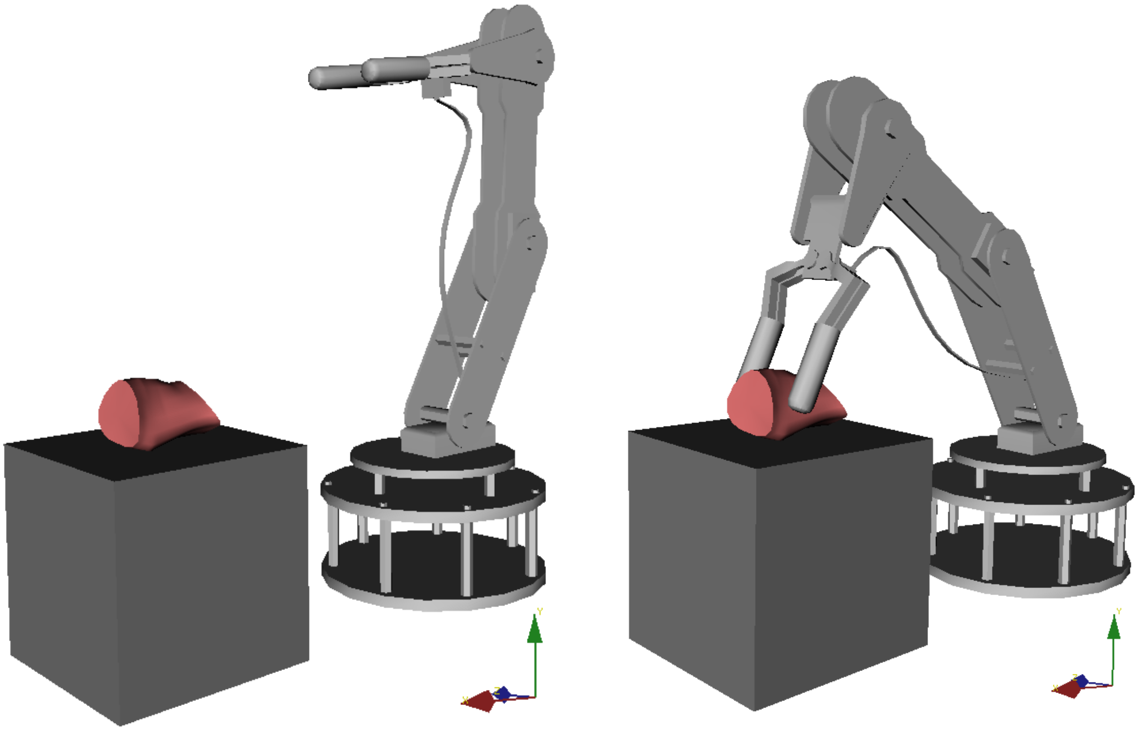

Figure 51: The robot arm can collide with the red object located on a box (Left: before collision; Right: after collision). Number of polygons (robot arm, red object, total): 1760, 1056, 2816. Number of mesh vertices (robot arm, red object, total): 1033, 530, 1563. By using cylindrical segmenting collision detection, pre-processing, main processing, and total processing time were 3.87, 0.035, and 3.91 milliseconds, respectively.

In addition to mechanical oriented simulations, the methods may be used in more general applications. For example, radial segmenting collision detection can be used for finding collision when an object (e.g. ball, torus or car) is moving on a rough surface. In such cases, the rough surface can be radially segmented around a fixed point above the surface to find the collision between it and the moving objects (see Figure 52).

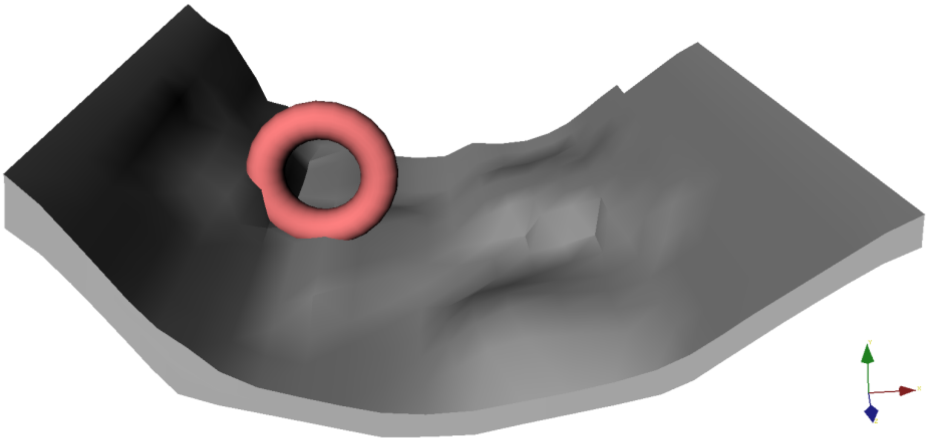

Figure 52: The red torus collides with the rough surface when moving on it (radial segmenting collision detection). Number of polygons (torus, rough surface, total): 1152, 4003, 5155. Number of vertices (torus, rough surface, total): 576, 1984, 2560. Pre-processing, main processing, and total processing time were 2.36, 0.76, and 3.12 milliseconds, respectively.

All these examples of non-medical applications (Figure 49, Figure 50, Figure 51 and Figure 52) were experienced by using PC with CPU: Xeon-3.4 GHz & RAM: 2GB.[8] The experiments were performed for three times per each example, and the average of obtained computational times was recorded. The details of the objects and the computational times can be found in the captions of the related figures.

3.8 Conclusion

In this chapter we presented two new fast collision detection methods, highly adapted to joint behavior. The methods take advantage of the relative proximity and the nature of the movement by cylindrical and radial segmentation of the space, to discard unnecessary calculations. They were compared with other collision detection methods in different scenarios. The comparison results showed that the methods can be up to about two orders of magnitude faster than previous methods. The penetration depths are accurately calculated by finding the exact distance between the penetrating vertices of one object and the polygonal-based surface of the other object, during the collision detection (without additional calculations). The proposed collision detection methods are able to be used accurately in different computer graphics applications, in which the movements can be modeled as rotation or sliding.

---

[8] Some objects were created by using Misfit Model 3D 1.3.4, and some others were downloaded from http://www.turbosquid.com/.

Chapter 4 Medical research and simulations

4.1 Introduction

In this chapter four different medical investigations and research are explained. These investigations and research are focusing on hip joint and the proposed collision detection methods in the previous chapter are applied in some of them.

At the beginning, based on the spatial partitioning algorithm proposed for cylindrical segmenting collision detection method in the previous chapters, a new method for finding the joints range of motion is proposed. The method is working without doing any collision detection tests, which makes it faster than traditional methods based on collision detection. In addition, unlike traditional methods its accuracy does not depend on the rotational steps.

In another work, the proposed collision detection methods are exploited for calculating contact penetration depths in the pathological hip joint during a daily activity. The calculated penetration depths are found correlated with to the FEM-based estimated contact stresses and the pathology of the hip joints. The results suggest a novel fast strategy for diagnosing hip diseases.

The third work is dedicated to evaluating the importance of the hip joint center of rotation in medical simulations. For ten patients, the hip contact radial and curvilinear penetration depths are estimated during hip movements about its medical axes. The movements and penetration calculations are done by using different centers of rotations estimated by different methods. Based on the results, the sensitivity of the hip contact penetration during its movement to the method used for estimation hip center of rotation is evaluated.

Finally, the last research is about estimating hip joint contact pressure from hip mesh geometric features. In this research a statistical model for estimating hip contact pressures during its movement is proposed without performing mechanical simulations and without any need for movement details. The estimation is done by using the geometric features extracted from the unknown target hip model and some already mechanically evaluated training hip models. The method is later tested on some pathological hip models and the results are compared with mechanically based estimated contact pressures.

4.2 Joint range finder

4.2.1 Introduction

For diagnosing some of the human joint diseases, it is important to obtain the joint's range of motion. For example loss of internal rotation in the hip is one of the first signs of internal hip pathology and can be related to diagnoses, such as arthritis. Also, increase in femoral or acetablular anteversion usually demonstrates an increase in the internal rotation. Clinical hip examinations are usually based on rotating the hip in the special orientations. For instance, the flexion adduction internal rotation test is used to aid in the diagnosis of femoroacetabular impingement (Martin 2005). Such methods are usually based on the patient's feedback during the examinations and therefore the diagnosis may not be easy and accurate. After diagnosing the hip disease, the treatment may be based on surgery. For example options for treatment of femoroacetabular impingement include trimming of the anterior aspect of the acetabular rim (Mardones, et al. 2005). Since the operation can be highly invasive, it is essential that surgeon has a good knowledge of the joint's range of motion before operation to know exactly about the surgery strategy and reduce the risk of miss-operation.

Having a computer aided method for finding the joint's range of motion can help us to have a more accurate diagnosis and/or surgery strategy. The computer aided methods for joint simulations usually exploit the reconstructed 3D model of the joint's tissues (e.g. hip) obtained from MRI or CT (Gilles, Moccozet and Magnenat-Thalmann 2006). Although there are several works done for human-joint graphical simulation, but they are mainly focusing on either graphical modeling of tissues, mechanical modeling of soft tissues or estimating location of hip joint center (Kang, Sadri, et al. 2003), (Maciel, Boulic and Thalmann 2003). There is not much works for finding joint's range of motion. The current methods used for finding range of motion in human joints are based on performing successive rotation increments as long as no collision occurs between the 3D meshes of the fixed part and the mobile part along the selected anatomical axis of rotation (Kang, Sadri, et al. 2003). This can be highly time-consuming due to the fact that we cannot know the number of rotation steps in advance.

For increasing the simulation speed sometimes the collision detection is restricted to certain area of the tissues. Imposing such restrictions needs defining different areas of the tissues for the simulation program (Kang, Sadri, et al. 2003), which are not easy to be done automatically. These traditional methods can take a large amount of computing time. Therefore, for speeding up the collision detection based methods, it is necessary to use faster algorithms of collision detections. A well known method is using bounding volume hierarchy (Larsson and Akenine-Möller 2001), (Teschner, Kimmerle, et al. 2005). There are many other methods proposed for collision detection, such as Distance fields (Teschner, Kimmerle, et al. 2005), Image-space techniques (Baciu, Wong and Sun 1999), spatial hashing (Teschner, Heidelberger, et al. 2003), average-case approach (Klein and Zachmann 2003), and random selection method (Lin and Canny 1992) (see section 2.1 for more details). They may speed up the simulations in certain cases, but they cannot change the fact that they still must be performed several times during the rotation until a collision happens. Also the accuracy of the estimated range is always depending on the rotation steps. Although the computational speed of the methods for finding range of motion were improved by either segmenting the tissues and/or applying faster collision detection algorithms, but there is no work for developing completely independent model from collision detection algorithms.

In this section, we propose a novel method for finding maximum range of motion for human joints with rotating movements (or any artificial rotating joints). This method is based on classifying the fixed part of the joint in a cylindrically segmented space (similar to the pre-processing step of cylindrical segmenting collision detection method, explained in section 3.2), without using any collision detection algorithm. It proved to be much faster than traditional ones, and needs to be performed only once per axis of orientation. This method is illustrated on the case of finding maximum range of motion in the human hip joint, which is not only useful for further hip simulation but also provides information about hip disease for surgeon. The results are also compared with the hip’s range of motion obtained from human cadavers.

4.2.2 Method

For clarity, we consider one of the objects fixed and call it the fixed object (and the other one is called the mobile object). We first cylindrically segment the fixed object’s spatial occupancy. After that, one table is created for the fixed object, and list of the fixed polygons is stored. By comparing the position of each mobile vertex against the fixed polygons stored in the corresponding cell of table, the range of motion can be found accurately and without performing any collision detection.

4.2.2.1 Table creation

This step is done similar to the pre-processing step of cylindrical segmenting collision detection method (section 3.2). At the beginning, we transform both objects in a new coordinate system so that the moving object’s rotational axis is the z-axis (see section 3.2.2.1 for more details). Each vertex of both mobile and fixed object is converted from the Cartesian coordinates (X, Y, Z) to cylindrical coordinates (r, θ, z). Then, in order to avoid unnecessary calculations, we find and avoid non-colliding points by introducing the parameters which are independent from 'θ' (e.g. r and z). We prepare one table, where its size is function of the search interval and of the resolution (resp. along z and along r). Each cell of the table corresponds to one ring-segment of the space and the indices of the fixed polygons occupying that ring are stored in the corresponding table-cell (see section 3.2.2.2 for more details). Figure 53 shows the main block diagram for creating the table.

4.2.2.2 Finding the range of motion

For every vertex of the mobile object, we find the corresponding ring-segment which contains the vertex (by using the discretized coordinates of the vertex). Since the mobile object is rotating around the z axis, a mobile vertex may only collide with the fixed polygons staying in the same ring-segment of the vertex. Thus, we just need to check the angular distance between the mobile vertex and the fixed polygons in the corresponding table-cell of the ring-segment. For making our tests accurate, we should consider that it is possible to have some fixed polygons in the same ring-segment of the mobile vertex, but they may never collide with the mobile vertex in any rotational angle. For omitting such non colliding fixed polygons, we check whether a 2D representation of the vertex (r_0, z_0) is inside a 2D representation of the polygon (see Appendix B at the end of Chapter 3 for more details). If the answer is positive, then we calculate the angular distance between the mobile vertex and the fixed polygon (i.e. the amount of angle that the mobile vertex needs to be rotated in order to collide with the corresponding polygon, see Appendix C at the end of Chapter 3 for more details). This process is done for all the mobile vertices and the smallest and the largest values are found, where they correspond to the range of motion in anti-clockwise and clockwise directions, respectively (see Figure 54).

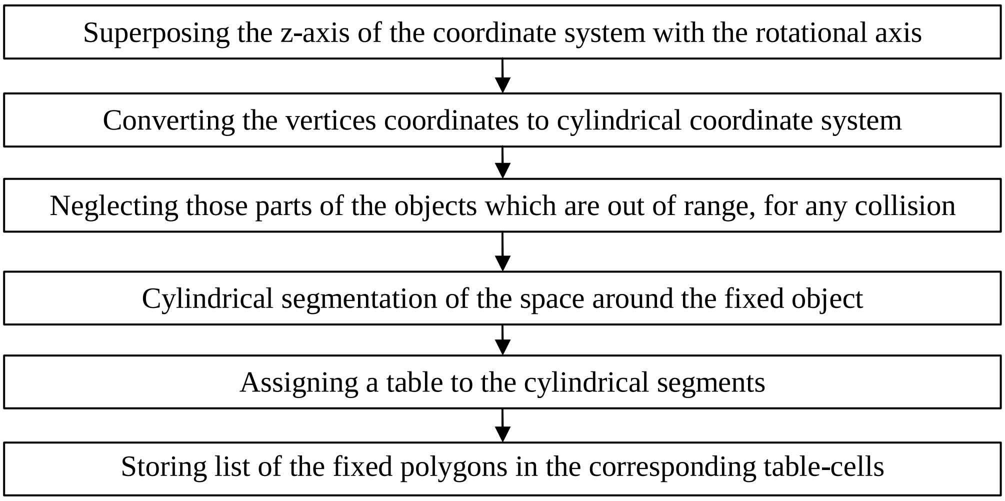
Superposing the z-axis of the coordinate system with the rotational axis
Converting the vertices coordinates to cylindrical coordinate system
Neglecting those parts of the objects which are out of range, for any collision
Cylindrical segmentation of the space around the fixed object
Assigning a table to the cylindrical segments
Storing list of the fixed polygons in the corresponding table-cells

Figure 53: Block diagram for creating the table in the range finder method.

```
BEGIN
  $delta_{max} = 0$
  $delta_{min} = 2\pi$
  FOR all the mobile vertices
    i = index of vertex;
    [$z_d$, $r_d$] = cylindrical partition containing "vertex[i]"
    FOR all the fixed polygons stored in table cell[$z_d$, $r_d$]
      j = index of polygon
      IF (Circular arc of "vertex[i]" intersects "polygon[j]") THEN
        delta = arc angle between "vertex[i]" and "polygon[j]"
        IF (delta < $delta_{min}$)   THEN   $delta_{min}$=delta   ENDIF
        IF (delta > $delta_{max}$)   THEN   $delta_{max}$=delta   ENDIF
  Range of motion in anticlockwise direction = $delta_{min}$
  Range of motion in clockwise direction = $2\pi$ - $delta_{max}$
END
```

Figure 54: Algorithm for finding the range of motion.

4.2.2.3 Resolution

For discretizing resolution, the best value is the average size of the fixed polygons. The reason is that if we chose a larger value than the average size of polygons, then we will have more number of polygons in each ring-segment and consequently more tests must be done for each mobile vertex. It decreases the processing speed. On the other hand, if we chose a smaller value, then we have not changed the number of polygons stored in the rings, but we have increased number of table-cells. It will slow down the creation of the tables and also more amount of memory will be occupied.

4.2.3 Biomedical experiment

We applied the proposed method for finding the range of motion in human hip joint. Our tests were based on finding the bone to bone range of motion; where femoral bone was considered as a mobile part limited by pelvis (see Figure 55).

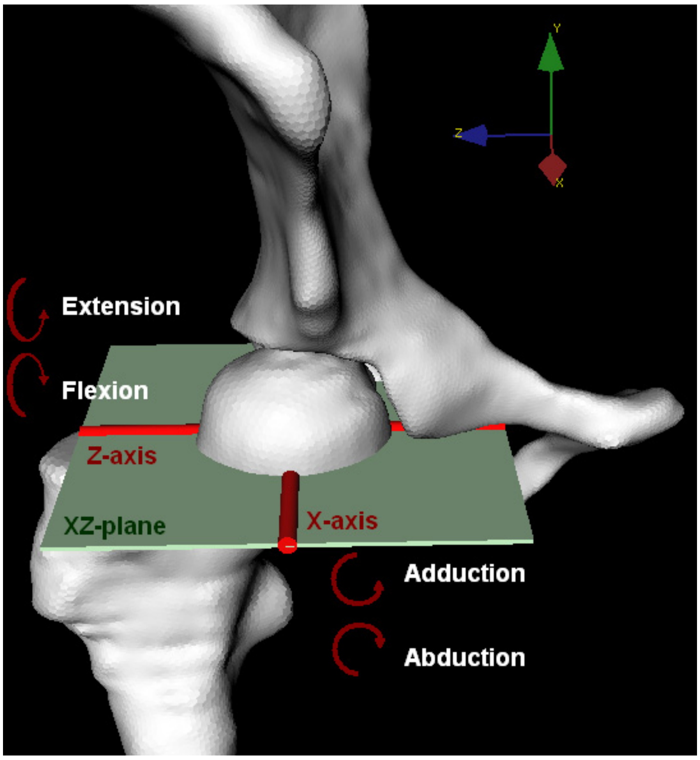

Figure 55: Finding the range of motion in the human hip joint based on hip graphical model

We used 3D triangular meshes obtained by segmenting MR Images taken from a patient by the method described in (Gilles, Moccozet and Magnenat-Thalmann 2006). Center of rotation was also found based on the scanned 3D models (Kang, Sadri, et al. 2003).[9] The method was tested for 4 different mesh resolutions (1: 254’976, 2: 63’744, 3: 15’936 and 4: 3’984 triangles). The tests were done by a PC with CPU: Xeon-3.4 GHz & RAM: 1GB.

In order to have a general comparison between the proposed method and the traditional collision based methods, we performed the same scenario for finding the range of motion around an axis (anatomical x-axis for finding adduction/abduction range), using repetitive collision detection method. We used two different methods for collision detection. One

---

[9] The 3D Meshes were prepared by Benjamin Gilles at MIRALab, University of Geneva.

was based on oriented bounding volume hierarchy (Hudson, et al. 1997) and the other one was based on spatial hashing method (Teschner, Heidelberger, et al. 2003). In the spatial hashing method we classified the mesh triangles instead of tetrahedral and also the algorithm was modified in the way to terminate after finding the first collision. Such modification could increase the speed of computation and consequently provide a better comparison. The rotating step used for the collision detection based methods was 1° (it should be noticed that the proposed method does not need any specification for step, and it find the exact range of motion). The results are shown in Table 7.

Table 7: Processing time for finding abduction and adduction range in the hip joint (normalized based on the values of the proposed method). The processing time of the proposed method was 156, 31, 16 and 4 milliseconds for mesh resolutions of 1, 2, 3 and 4, respectively.

	Proposed Method	Spatial Hashing	OBVH
Resolution 1	1	10.6	46.1
Resolution 2	1	18.6	61.5
Resolution 3	1	17.6	37.1
Resolution 4	1	19.5	66.3

We found range of motion for the other anatomical axis (y: internal/external rotation; z: flexion/extension), based on ISB International Society of Biomechanics Standardization Document (Wu, et al. 2002), too. We applied the method for finding range of motion for different axes inside the horizontal plane (the plane built by x-axis (adduction-abduction) and z-axis (flexion, extension)). The plane is shown inside Figure 55 by green color. The results are illustrated in Figure 56. Upper curve shows the range of motion, when the femur is rotating around different axes in xz-plane, in anti-clockwise direction; and the lower curve is showing the same rotations in the clockwise direction.

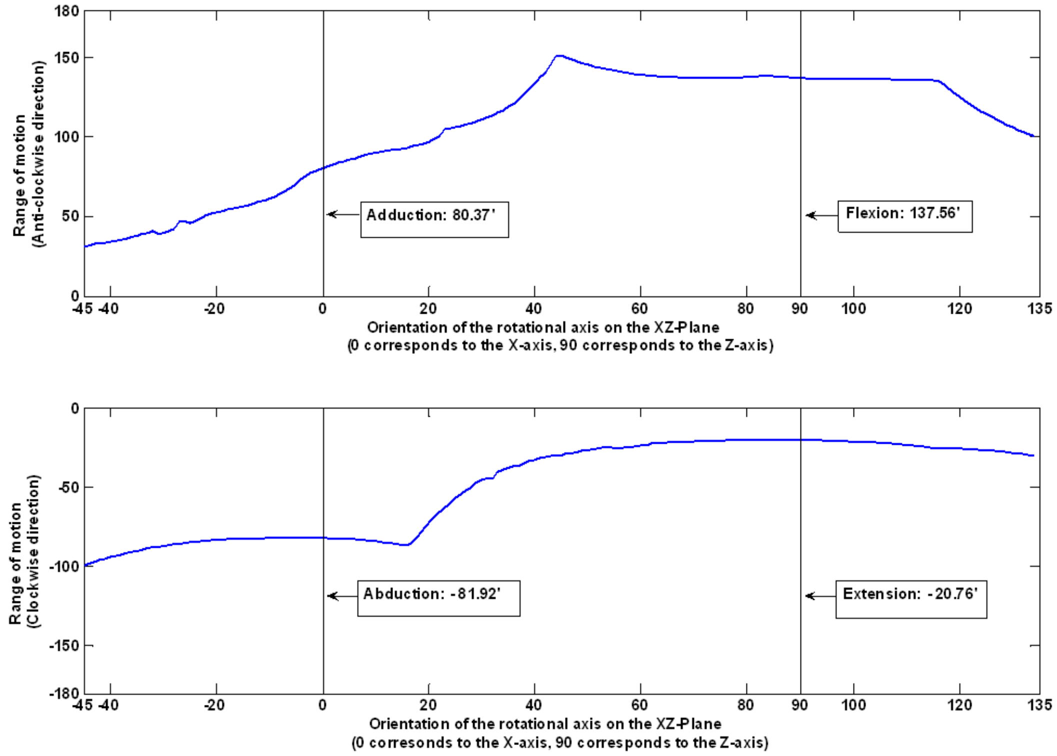

Figure 56: Range of motion for different axes inside the horizontal plane

The results were compared with the full hip capsule's range of motion obtained from human cadavers (Genoud, et al. 2000). The values provided by (Genoud, et al. 2000), are obtained by considering full hip capsule and the stresses applied on the hip soft tissues. Therefore, the valid bone-to-bone range of motion must be equal or larger than the full hip capsule's range of motion. Our results were validated by confirming that the found range of motion is never less than the full capsule's range of motion. Also it was observed that for flexion/extension our bone-to-bone range of motion is almost the same of full hip capsule's range of motion.

4.2.4 Discussion and conclusion

In this section we proposed a novel method for finding the range of motion in joints based on cylindrical segmentation and without using any collision detection algorithm. The method needs to be performed only once per simulation to find both anti clockwise and clockwise range of motion. The method was tested on the case of finding the range of motion in the human hip joint. For this reason we applied the algorithm on 3D triangular meshes exploited by segmenting MR Images of the hip joint. We compared the computational speed with two collision detection based method.

These experiments showed that the proposed algorithm is significantly faster than the previous methods (10 to 66 times faster) and can be used for the biomedical applications such as musculo-skeletal simulation as it is difficult to infer precisely the relative range of motion from scanned or reconstructed 3D models. The estimated range of motion was also an accurate value, which is another advantage comparing to the collision detection based methods, where we have to define rotation steps. Our results were also validated by confirming that the found range of motion is never less than the full capsule's range of motion, and it is even almost the same of full hip capsule's range of motion for flexion/extension.

4.3 Using contact penetration for hip medical diagnosis

4.3.1 Introduction

In the past few years, femoroacetabular impingement (FAI) was recognized as the leading pathomechanism leading to a significant number of so-called ‘primary’ hip osteoarthritis (Murphy, et al. 2004), (Ganz, et al. 2003). It is defined by an early pathological contact between primary osseous prominences of the acetabulum and/or the femoral head-neck junction. Depending on the underlying pathomorphology and its related pathomechanism, two different types of FAI are distinguished: ‘pincer’ and ‘cam’ impingement. The pincer type of FAI describes a linear contact between the acetabular rim and the femoral head-neck junction. The maximum impact force is tangential to the articular surface. A cam impingement occurs when the femoral head-neck junction has an abnormally large radius. This eccentric portion slides into the acetabulum and induces compression and shear stress at the junction between the labrum and the cartilage and at the subchondral landmark. The maximum impact force is perpendicular to the articular surface.

Diagnosis of FAI can prove to be difficult, particularly in cases with subtle pathomorphologies and/or combined hip dysplasia and FAI occurring in the same hip (Günther, et al. 2008). Several computer aided simulations for three-dimensional analysis of hip pathologies have been presented before (Kubiak-Langer, et al. 2007), (Tannast, Kubiak-Langer, et al. 2007), (Arbabi, Boulic and Thalmann 2007), (Kang, Sadri, et al. 2003), (Chegini, Beck and Ferguson 2008), (Russell, et al. 2006), (Michaeli, Murphy and Hipp 1997). Most of these methods imply substantial drawbacks for use in daily clinical practice or for research purposes. These limitations include mainly the neglect of soft tissue structures (labrum, cartilage) or the inability to achieve a real-time simulation. Approaches based on finite element (FE) analysis can provide an excellent evaluation of static hip problems such as developmental dysplasia of the hip (Chegini, Beck and Ferguson 2008), (Russell, et al. 2006), (Michaeli, Murphy and Hipp 1997), (Hipp, et al. 1999). However, having a better estimation of stresses for a dynamic pathomechanism such as FAI is very time-consuming and not applicable for real time medical hip simulations (Chegini, Beck and Ferguson 2008). Finally, there is a lack of a fast method evaluating penetration depth of the colliding soft and bony tissues in FAI, and which has been correlated with available biomechanical simulation data.

We introduce a new computer-assisted method for real-time evaluation of FAI based on virtual penetration depth for the colliding tissues including soft tissue structures. The aims of this study were (1) to validate this new method with existing FE models, and (2) to determine potential differences between cam, pincer and combined types of FAI and hip dysplasia. We hypothesized that (1) there is a good correlation between virtual penetration depths of the colliding tissues and predicted stresses, and (2) this method can be used for differentiation of subgroups of FAI and dysplasia (Arbabi, Chegini, et al. 2009).

4.3.2 Method

4.3.2.1 The penetration depth method

Generally, two virtually rotating objects collide and potentially deform depending on the structural properties. The penetration depth method assumes that the objects are rigid bodies that can virtually penetrate each other. Applying this principle to the impinging hips, two types of penetration can be estimated when the femur penetrates with the acetabular chondrolabral complex during rotation: (a) curvilinear or (b) radial penetration (Figure 57).

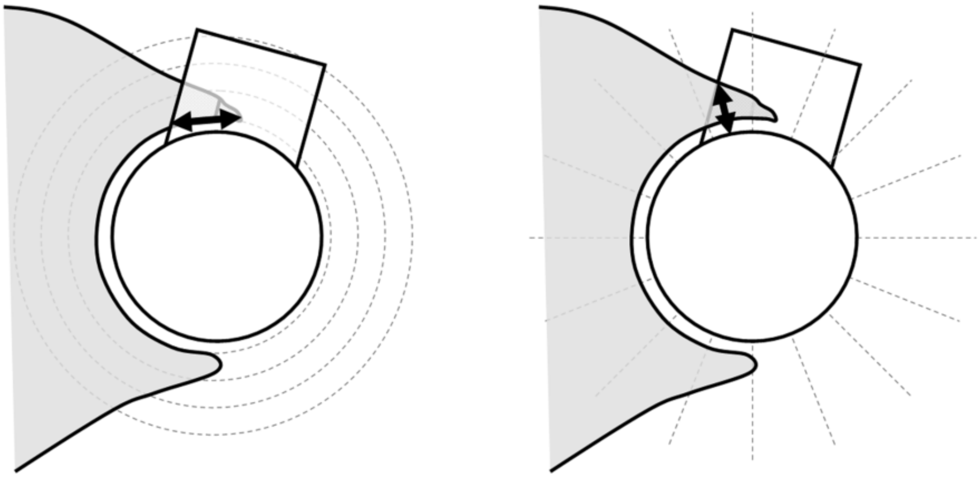

Figure 57: Left: Cylindrical segmentation of the space around the hip joint. Curvilinear penetration depth obtained by cylindrical segmenting method. Right: Radial segmentation of the space around the hip joint. Radial penetration depth obtained by radial segmenting method (Arbabi, Chegini, et al. 2009); (Image courtesy of Moritz Tannast, Dept. Orthopaedic Surgery, Inselspital, University of Bern).

Analogous to the pincer type of FAI, the 'curvilinear' penetration occurs in the angular direction, which is tangential to the rotational trajectory. By applying cylindrical segmenting collision detection method proposed in this thesis (see Chapter 3), the space around the femoral head was cylindrically segmented based on the given point of rotation. For each labrum surface polygon, its index was stored in table cells corresponding to the cylindrical segments intersecting the polygon. The penetrating femur vertices were detected by determining the table cell that femur vertices belong to and checking potential collision along a circular trajectory with the labrum surface polygons stored in the cell. Finally, the penetration depth of a femur vertex was found by calculating the smallest amount of distance that the femur must to be moved away on its rotational trajectory in order to prevent collision with the labrum (Figure 57: left). In order to have smoother results, the method was also repeated by exchanging the role of femur and labrum (i.e. finding the penetrated labrum vertices inside the femur).

Analogous to the cam type of FAI, the 'radial' penetration takes place in the radial direction. By using the proposed radial partitioning collision detection method (see Chapter

3), the space around acetabular soft tissues (acetabular cartilage, labrum and chondrolabral transition zone) was radially segmented, and the acetabular soft tissues surface polygon indices were stored in corresponding table cell(s). Collision was detected by determining the table cells that femur vertices belong to and calculating the radial distance between femur vertices and acetabular polygons occupying the same cell (radial segment). If the radial distance of a femur vertex to the center was more than the radial distance of its corresponding acetabular soft tissues polygon, the femur vertex was returned as a penetrated vertex. The difference between these two radial distances was considered as the radial penetration depth (Figure 57: right). This method was also repeated by exchanging the role of femur and acetabular soft tissues (i.e. finding the penetrated acetabular vertices inside the femur) to obtain smoother results.

4.3.2.2 Hip models

The morphology of the human hip can be described by various selected anatomical and radiographical parameters (Tannast, Siebenrock and Anderson 2007). For simplicity and comparability, only two important parameters were chosen to quantify acetabular and femoral pathomorphologies: the lateral center-edge (CE) angle of Wiberg (Wiberg 1939), and the α angle of Nötzli (Nötzli, et al. 2002), respectively. The CE angle (Figure 58: left), as an indicator for pincer impingement, is defined as the angle formed by the perpendicular to the inter-tear drop line and the line passing from the center of the femoral head to the lateral edge of the acetabulum (Wiberg 1939). The α angle (Figure 58: right), as an indicator for cam impingement, is formed by the femoral neck axis and a line connecting the center of the femoral head with the point of beginning a sphericity (Nötzli, et al. 2002).

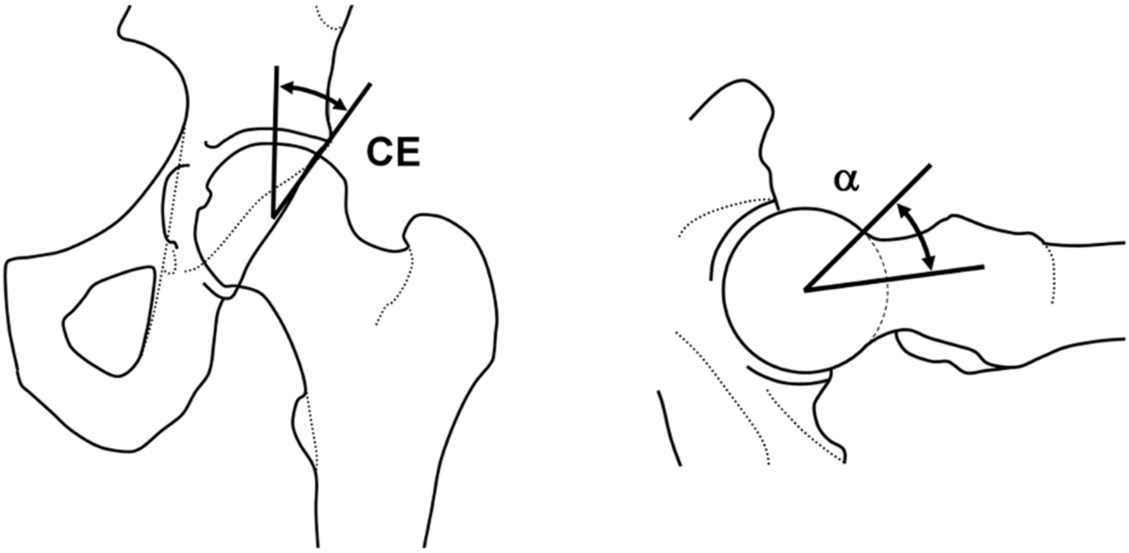

Figure 58: Definition of CE angle (left) and α angle (right) in a hip joint (Arbabi, Chegini, et al. 2009); (Image courtesy of Moritz Tannast, Dept. Orthopaedic Surgery, Inselspital, University of Bern).

The hip models were prepared by Salman Chegini at ARTORG, University of Bern, by using CAD software[10]. These models included acetabular and femoral bone, articular

---

[10] Solidworks 2005, Solidworks Corp., Boston, MA, USA

cartilage, the labrum and the chondrolabral transition zone. The femoral and acetabular cartilage surfaces are known to be spherical with deviations from sphericity on the order of <100 µm (Macirowski, Tepic and Mann 1994). Hence, the articulating joint surfaces were modeled as portions of spherical surfaces with a common 25-mm radius. The acetabular cartilage was horseshoe-shaped and had a constant thickness of 2 mm. The femoral head cartilage was assumed to be 2 mm at its thickest point, gradually reducing to zero toward the lateral edge (Eckstein, et al. 1997). The labrum was defined based on morphological studies, such that it covered the femoral head cartilage in the unloaded condition with a triangular cross-section and a height of 7 mm from the acetabular rim to the tip (Won, et al. 2003). In order to create a wide range of hip geometries, a consecutive series of α and CE angles were chosen for evaluation, covering normal and pathological joint morphologies. The CE angle values ranged from 0° – 40°, α angles ranged from 40° – 80°. Increments of 10° were selected for both parameters, resulting in a total of 25 different joints for evaluation, e.g. normal (CE=20°, α=40°), cam (CE=20°, α=80°), pincer (CE=40°, α=40°), combined FAI (CE=40°, α=80°), dysplastic (CE=0°, α=40°), or combined dysplastic and impinging morphologies (CE=0°, α=80°) (Figure 59) (Chegini, Beck and Ferguson 2008).

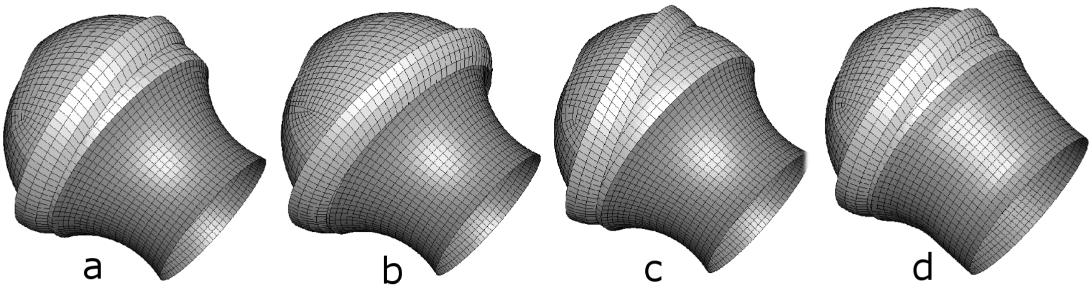

Figure 59: a: normal joint, b: pincer joint c: dysplastic joint. d: cam type joint (Chegini, Beck and Ferguson 2008); (Image courtesy of Salman Chegini, ARTORG, University of Bern).

4.3.2.3 Validation

The proposed penetration depth method was validated in relation to previously described FE simulations for FAI (Chegini, Beck and Ferguson 2008), which were considered as a gold standard. FE simulations were done by Salman Chegini at ARTORG, University of Bern. The created models were imported to FE software[11], and were meshed by quadratic elements for FE simulation and surface S3 elements for use in the fast collision detection algorithm. For FE simulations, specific motions described later were applied as a prescribed rotational kinematic about the femoral head center. The translation was kept unconstrained and the corresponding joint reaction force vector was simultaneously applied. The applied force on the femoral head center was balanced by the integrated vectoral sum of all contact pressures, including the impinging reaction at the outer margins of the acetabulum. The femoral head could relocate itself within the acetabulum, adopting a more preferential distribution of opposing impingement contact stresses. Articular cartilage was modeled as a linear elastic material with an elastic modulus of E=12 MPa, and Poisson ration of υ=0.45 (Moglo and Shirazi-Adl 2003). The labrum's material properties were E=20 MPa and υ=0.4 (Ferguson, Bryant and Ito 2001), and the bony components were assumed to be rigid in comparison to the soft tissues. As an output

---

[11] ABAQUS 6.6, ABAQUS Germany Ltd., Aachen, Germany

parameter of the simulation, von Mises stress, which is a measure of distortion energy in the material (Chegini, Beck and Ferguson 2008), was reported.

Deep flexion (particularly when combined with adduction and internal rotation) could be shown as the motion pattern with the highest femoro-acetabular impact (Tannast, Goricki, et al. 2008). Consequently, the known average in-vivo load and motion data for standing-to-sitting were used for evaluation (Bergmann, et al. 2001). These were described in defined coordinate system (O center of the femur head; OX connecting the center of the left femur head to the right one, OY toward front and OZ upward). The rotations of the femur head around these axes were defined as flexion, adduction and internal rotation respectively. The angle of rotation which was associated with largest contact stress values (using the FE method) was (59.6°,13.6°, 10.9°) in the defined coordinate system and was used to evaluate our penetration depth algorithm. The total motion for standing-to-sitting was divided into 30 equal and consecutive sub-motions. All of the penetrating vertices were found, and both curvilinear and radial penetration depths as well as the Mises stresses were calculated in each sub-motion (see Figure 60) (the full movement could not be applied for some extreme pathological hip models due to large deformation (Chegini, Beck and Ferguson 2008)). Due to the fast processing time of the applied collision detection methods, the calculations of the penetration depth method could be done in real time during hip motion. Because of the computational nature of the evaluated parameters and lack of observer-dependent variables, no reliability or reproducibility analysis was necessary.

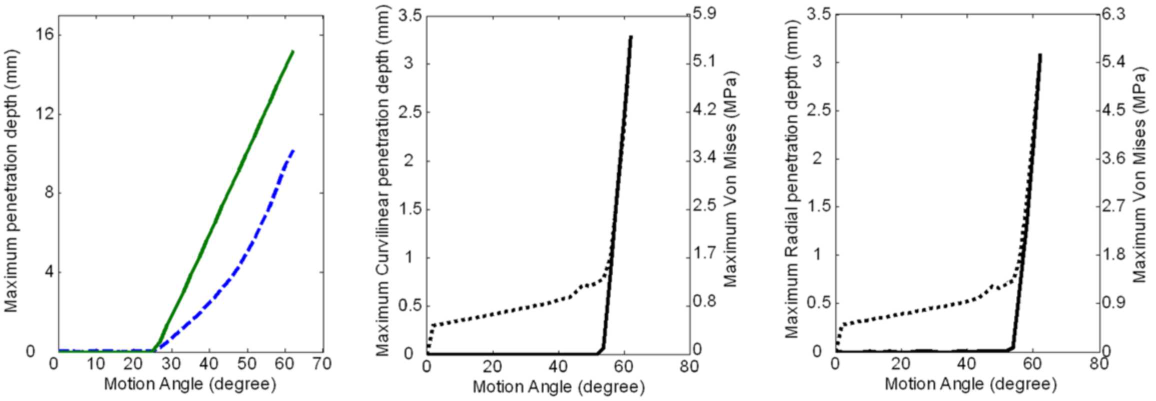

Figure 60: Left: maximum curvilinear penetration (solid curve) and radial penetration (dashed curve), during standing to sitting (for case: α=60˚, CE=40˚). Right, middle: maximum curvilinear (middle) and radial (right) penetration depth during standing to sitting (solid curves), (for case: α=CE=40˚). Maximum von Mises stress (dotted curve) during the same movement is illustrated by dotted curves.

4.3.2.4 Relation between penetration depths and pathologies

For each hip model, a curve describing the maximum curvilinear and radial penetration depths for different motion angles was created. For quantitative evaluation, the maximum of the curve values (peak) was extracted from each curve. In order to detect patterns for the curvilinear and the radial penetration for specific hip pathomorphologies, the difference of the normalized maximum penetration depths was calculated. This normalization was performed for radial and curvilinear penetration depths, separately. The calculated differences could show which type of penetration is stronger for the corresponding hip pathologies (if the difference is positive (negative), the curvilinear (radial) penetration depth is stronger).

4.3.2.5 Statistical evaluation

Correlations between the penetration depths and the von Mises stresses were analyzed using a simple linear regression model. Pearson's correlation coefficient *r* was interpreted as "poor" below 0.3, "fair" from 0.31–0.5, "moderate" from 0.51–0.6, "moderately strong" from 0.61–0.8, and "very strong" from 0.81–1.0 (Chan 2003). Significance was defined as $p < 0.05$.

4.3.3 Results

4.3.3.1 Validation

The mean correlation coefficient was 0.84 (standard deviation 0.18, range 0.31 – 0.97) for all penetration depths, 0.91 (standard deviation 0.04, range 0.83 – 0.97) for the curvilinear penetration, and 0.80 (standard deviation 0.22, range 0.31 – 0.97) for the radial penetration. A strong correlation was found for pure cam, pure pincer, combined cam-pincer and combined dysplasia-impingement pathologies both for the curvilinear and the radial penetration. The penetration depth could not be calculated for hips with pure hip dysplasia or a normal hip since no penetration was detected (Table 8 and Table 9).

Table 8: Correlation between the maximum curvilinear penetration depths and von Mises stress (left) and the related p-values (right), during standing-to-sitting. "n.p." stands for no penetration.

	α=40˚	α=50˚	α=60˚	α=70˚	α=80˚
CE=0˚	n.p.	n.p.	n.p.	n.p.	0.85, $2.14*10^{-9}$
CE=10˚	n.p.	n.p.	n.p.	0.83, $5.75*10^{-9}$	0.89, $2.01*10^{-11}$
CE=20˚	n.p.	n.p.	0.89, $2.51*10^{-11}$	0.91, $1.43*10^{-12}$	0.89, $3.99*10^{-10}$
CE=30˚	n.p.	0.90, $9.50*10^{-12}$	0.95, $1.04*10^{-16}$	0.90, $1.01*10^{-11}$	0.91, $5.31*10^{-9}$
CE=40˚	0.97, $3.73*10^{-19}$	0.96, $3.82*10^{-18}$	0.93, $4.15*10^{-12}$	0.88, $4.39*10^{-9}$	0.95, $3.42*10^{-8}$

Table 9: Correlation between the maximum radial penetration depths and von Mises stress (left) and the related p-values (right), during standing-to-sitting. "n.p." stands for no penetration.

	α=40˚	α=50˚	α=60˚	α=70˚	α=80˚
CE=0˚	n.p.	n.p.	n.p.	n.p.	0.83, $7.38*10^{-9}$
CE=10˚	0.31, 0.092	0.34, 0.064	0.39, 0.031	0.83, $6.42*10^{-9}$	0.90, $3.96*10^{-12}$
CE=20˚	0.60, $3.88*10^{-4}$	0.61, $2.70*10^{-4}$	0.89, $2.75*10^{-11}$	0.95, $4.93*10^{-16}$	0.91, $1.58*10^{-11}$
CE=30˚	0.58, $6.20*10^{-4}$	0.90, $9.89*10^{-12}$	0.97, $7.90*10^{-20}$	0.97, $3.32*10^{-20}$	0.93, $4.27*10^{-10}$
CE=40˚	0.97, $7.17*10^{-20}$	0.97, $4.56*10^{-20}$	0.96, $4.98*10^{-16}$	0.97, $3.93*10^{-16}$	0.96, $1.34*10^{-8}$

4.3.3.2 Relation between penetration depths and pathologies

Each of the 25 evaluated hip models resulted in a characteristic pattern of curvilinear and radial penetration depth. For dysplastic hips and hips with a normal morphology, no penetration depths could be calculated because of lack of a collision. The maximum curvilinear penetration depth was found for a combined cam-pincer pathomorphology. The maximum radial penetration depth was found for pure cam deformities. The maximum normalized difference between curvilinear and radial penetration depths was found for combined cam-pincer hips. The minimum normalized distance was seen for pure cam impingement.

4.3.4 Discussion

In contrast to range finder methods which neglect soft tissue structures (cartilages and labrum) (Kubiak-Langer, et al. 2007), (Tannast, Kubiak-Langer, et al. 2007), our method take into account the presence of soft tissues along with bony parts. Although FEM models consider soft tissues (Chegini, Beck and Ferguson 2008), they have a slow processing speed which avoids them to be used in real time applications. In addition, FE models, which include finite-sliding contact between the soft tissues, are sensitive to the smoothness of the surfaces. In fact, the patient specific finite element models from MRI (in order to include soft tissues), face difficulties due to the resolution of the models created from MRI (Gilles, Moccozet and Magnenat-Thalmann 2006). The introduced surface irregularities invalidate the contact solution and require manual correction of the model geometry. On the other hand, the penetration depth model is not sensitive to small irregularity of the surfaces, and has the potential to be applied to patient specific models with less manual intervention.

There are limitations for our approach. The penetration depth method is not valid in evaluating normal hips or hips with pure hip dysplasia. In these cases, there is no collision (and therefore no penetration) of the two rotating objects. Nevertheless, this can represent important clinical information. The absence of penetration depths allows the exclusion of relevant FAI. Another limitation is the fact that our calculations are strictly based on virtually created geometrical models. We did not evaluate individual patient morphologies. But since our models could be proven to behave very similar to the actual joint kinematics and damage patterns (Chegini, Beck and Ferguson 2008), it is conceivable, that the penetration depth method proves a similar utility in analyzing individual hip morphologies.

The von Mises stress rises in impinging joints due to large deformations introduced into the labrum by the bony part of the femur head. The potential penetration translates into either displacement or deformation in the real deformable soft tissues. The areas with large local deformation experience the highest distortion energy, rather than the displaced areas. In the impinging hip joint, the lower zone of the labrum experiences the largest displacement and consequently the upper zone adjacent to the acetabular rim is under high distortion. Therefore, although the maximum penetration depth and maximum stress are not occurring in the same zone (Figure 61), there is a cause and effect relationship between them that can be well quantified. In fact, the calculated correlation coefficients shows that the estimated penetration depths and the von Mises stresses are highly correlated when the hip joint is suffering from cam, pincer or combined pathologies.

To date, the differentiation of the individual subtypes was based on an observer-based judgment, which might be subject to substantial misinterpretation and underestimation of pathologies. The penetration depth method allows a more independent differentiation between the subtypes of impingement. Our results (Figure 62) show how the maximum of the curve (related to maximum curvilinear and radial penetration depths) can distinguish between different kinds of hip models. Having less intersection and more distance between individual curves demonstrate how the parameter extracted from penetration depth varies with the pathology. As it can be seen, the curves are more separated for cam and pincer cases (CE 20˚).

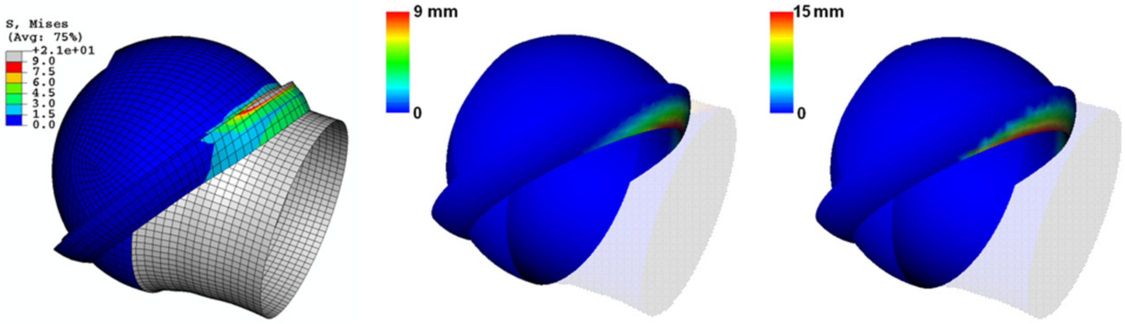

Figure 61: Left: Von Mises stress in a deformed FEM model (α=60˚, CE=40˚). Center and Right: Labrum is colored based on its Radial and Curvilinear Penetration depths in the same model, respectively (femur bone is semi-transparent) (Arbabi, Chegini, et al. 2009); (Left image courtesy of Salman Chegini, ARTORG, University of Bern).

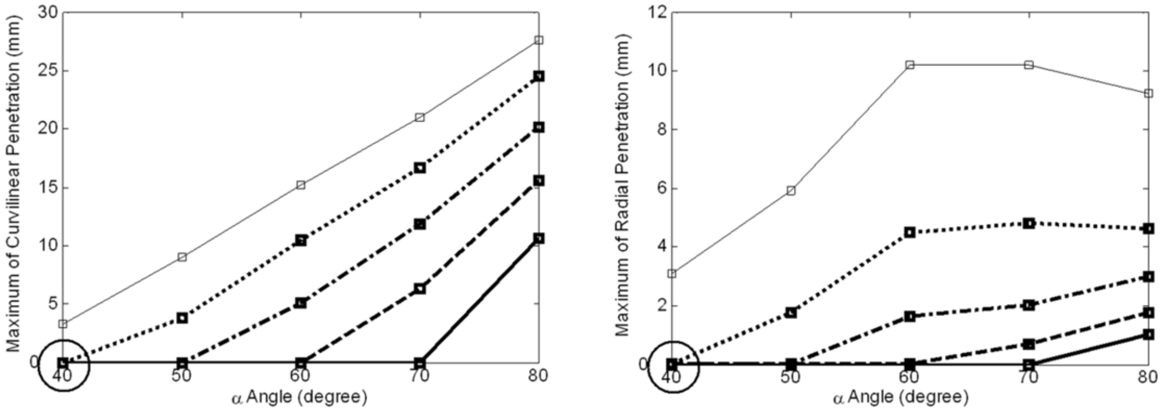

Figure 62: Changes in the maximum curvilinear (left) and radial (right) penetration (maximum during all the movement) by α angle. CE = 0˚, 10˚, 20˚, 30˚ and 40˚ are presented by thick-solid, dashed, dashed-dotted, dotted and thin-solid curves, respectively. The normal case (CE=20˚, α=40˚) is indicated by a circle.

We compared both of the estimated penetration depths with each other, for different hip models, and also discussed with a medical doctor about the results (Dr. Moritz Tannast at Dept. Orthopaedic Surgery, Inselspital, University of Bern). It was observed that, depending on the pathology, a different type of penetration depth is more significant. In Figure 63: bottom-left, the values below (above) zero indicate that the radial penetration depth is more (less) significant than curvilinear penetration depth, for the corresponding models. By checking the results, it is seen that when the hip model is closer to pure cam morphology, radial penetration depth is more significant in comparison to curvilinear penetration depth. In fact, pure cam morphology shows the smallest negative values compared to any other model. In contrast, the models closer to a pure pincer morphology have more curvilinear penetration depths compared to radial penetration depth. This could give some diagnostic information, especially in cases with combined impingement, where the major component has to be defined and treated. In brief, in Figure 63: bottom-left, the red region, the blue region, and the green region are corresponding to pincer cases, cam cases, and dysplasia/normal cases, respectively. Thus, by estimating curvilinear and radial penetration depths of a hip model, we can find the pathological category that hip joint belongs to. Table 10 allows an easy assignment to one of the subgroups.

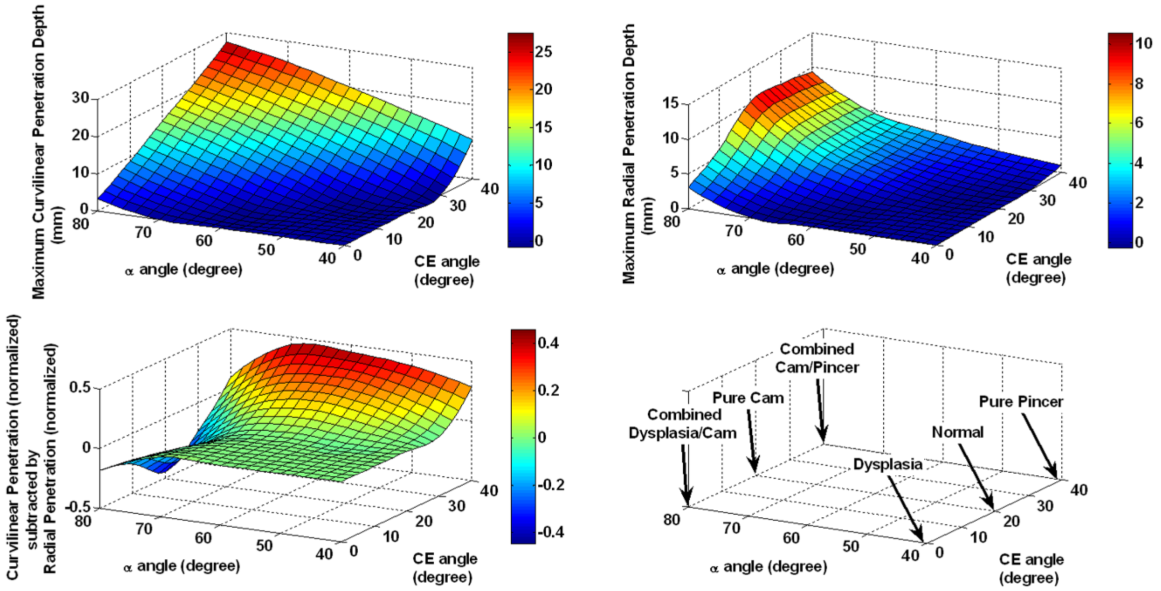

Figure 63: Upper images demonstrate the maximum of curvilinear (left) and radial (right) penetration depths for different combinations of α and CE angles. Bottom-left demonstrates difference between normalized values of these two kinds of penetration depths for different combinations of α and CE angles (curvilinear penetration values subtracted by radial penetration values (normalized)). Bottom-right shows hip pathologies corresponding to different combinations of α and CE angles.

Table 10: Overview on the maximum curvilinear and radial penetration depths and their differences for different characteristic joint morphologies. Curvilinear penetration depths were judged as follows: ‘very low’ between 0 – 5mm, ‘low’ 5 – 10mm, ‘intermediate’ 10 – 15mm, ‘high’ 15 – 20mm, and ‘very high’ 20 – 25mm. Radial penetration was judged as follows: ‘very low’ between 0 – 2mm, ‘low’ 2 – 4mm, ‘intermediate’ 4 – 6mm, ‘high’ 6 – 8mm, and ‘very high’ 8 – 10mm. Differences between both methods were judged as follows: ‘very low’ between (-0.4) – (-0.2) mm, ‘low’ (-0.2) – 0mm, ‘intermediate’ 0 – 0.2mm, and ‘high’ 0.2 – 0.4mm. See Figure 63.

Clinical interpretation of hip morphology	Lateral center edge angle (degrees)	α angle (degrees)	Maximum curvilinear penetration (CP)	Maximum radial penetration (RP)	Differences between both methods (CP – RP)
Normal hip	20	40	Very Low	Very low	intermediate
Pure pincer impingement	40	40	Low	Very low	High
Pure cam impingement	20	80	High	Very high	Very Low
Combined cam-pincer impingement	40	80	Very high	Very high	Intermediate
Hip dysplasia	0	40	Very low	Very low	Intermediate
Combined dysplasia and impingement	0	80	Very low	Low	Low

Finally, this research suggests a novel fast way for evaluating hip diseases, especially hip impingement. The proposed method is much faster than stress estimation methods and can be used for real time medical applications. The results of this method have a considerable correlation with hip stresses and are also able to distinguish among different hip pathologies. The last distinguishing ability is not only useful for diagnosing hip diseases, but also can help surgeons to test and validate their surgical plans on the virtual hip joint, by tracking the changes in the penetration depths before and after the surgery (Arbabi, Chegini, et al. 2009).

In summary, the penetration depth method is a valuable tool for evaluating FAI, but not hip dysplasia. It allows a fast, validate quantification of the impingement and depending on the individual features of the two penetration depths and their interrelationship an easy, observer-independent assignment to one of the subgroups of FAI. In future, the method must be evaluated based on individual patient morphologies. Furthermore, it can potentially be tested on other human joints or new hip models by considering more tissues (e.g. capsule). Also, the method may be improved by introducing new parameters to be extracted from penetration depths or by imposing some slight but effective changes in the algorithms for estimating penetration depths.

4.3.5 Conclusion

We proposed and validated a new fast strategy based on estimating penetration depths for evaluating hip pathology, suitable for real time medical hip simulations. Different models of pathological hip joints were prepared based on different CE and α angles. Stresses (von Mises) and penetration depths (curvilinear and radial) were validated separately for each hip model, using the FE method and a collision detection strategy respectively. The proposed method could be shown to correlate strongly when compared to von Mises stresses of the FE analysis. It could be shown that the penetration depth method allows a differentiation between characteristic pathomorphotypes related to femoroacetabular impingement, and it can be used for real time medical applications, which are desired by physicians.

4.4 Evaluating importance of the hip joint center of rotation in medical simulations

4.4.1 Introduction

In orthopedic simulations the behavior of bones and related tissues such as cartilages are investigated during their movements (i.e. rotation about an estimated center). In general, hip simulations include investigating contacts happening among soft or bony tissues during hip rotation. During hip rotation, contacting tissues apply pressure on each other which can cause damage if the pressure exceeds the normal amount. For medical simulations, three dimensional virtual models of real tissues are created based on CT or MR images. These virtual tissues can collide and penetrate each other during the computer based simulation.

Since these kinds of hip simulations are based on rotation, it is needed to have an estimation of hip joint center of rotation (HJC) in advance. There are several methods for estimating HJC, and different simulations may apply different methods for obtaining HJC. Therefore, it is very important to know how the results of simulation may vary based on the methods used for estimating HJC.

Many different methods of HJC estimation have been proposed that can be classified in predictive and functional approaches. The predictive (static) approach relies upon the location of anatomical landmarks (Bell, Petersen and Brand 1990), (Boudriot, Hilgert and Hinrichs 2006), (Kirkwood, Culham and Costigan 1999), (Seidel, et al. 1995). The functional (dynamic) approach estimates the HJC from recorded motion (Camomilla, et al. 2006), (Cappozzo 1984), (Chang and Pollard 2007), (Piazza, Okita and Cavanagh 2001), (Siston and Delp 2006) or simulated (Gilles 2007), (Kang, Sadri, et al. 2003), (Kang 2004). It has been reported that the functional approach is more accurate since joint dynamics are taken into account (Wu, et al. 2002). Nevertheless, the quality of the functional approaches depends on various factors, such as the type, the amplitude, and the number of movements. For instance, (Piazza, Okita and Cavanagh 2001) reported a significant increase of the HJC location error when the range of hip motion was reduced from 30° to 15°. As a result, when the movements' amplitude is limited the predictive approaches can be the only applicable methods (e.g. (Bush and Gutowski 2003)). The usage of simulated motions can avoid some of these shortcomings. By using 3D models reconstructed from medical imaging data of subjects in supine position, (Gilles 2007) showed that a fast functional approach can outperform some predictive approaches.

In many hip computer-based simulations, it is the contact among hip tissues that should be investigated. Therefore, three dimensional virtual models of real tissues are created based on CT or MR images, and we simulate the virtual tissues that penetrate one another during the movement. In reality stresses corresponding to the amount of virtual penetration occur to avoid such inter-penetration of real tissues. Therefore having an estimation of penetration depth can give an appropriate measure for investigating hip contacts during the simulation. For estimating penetration depth among virtual objects, collision detection methods are used. In fact, collision detection methods return colliding elements (i.e. vertices and polygons building virtual tissues). There are many methods proposed for collision detection. In Chapter 3, we proposed two methods for detecting collision among rotating objects. In these two methods, the penetration depths are calculated in either radial direction (originating from center of rotation) or angular direction (along the circular arc induced by on-going rotation axis). Because these methods are specialized for rotating objects, we use them for detecting collision and estimating penetration depth among contacting tissues of the hip joint.

In this section we investigated the sensitivity of hip simulation to the estimated center of rotation. The HJCs were estimated by five different methods for ten patients. For each patient and each estimated HJC[12] we rotated hip about different medical axes and estimated changes in the penetration depth of hip tissues during the rotation (Arbabi, Schmid, et al. 2009).

4.4.2 Method

4.4.2.1 MRI acquisition and models reconstruction

The MRI protocol and segmentation method, explained below, were used by Jerome Schmid at MIRALab, University of Geneva for reconstruction of the hip 3D models.

4.4.2.1.1 MRI protocol

In close collaboration with physicians, a protocol is defined to acquire images carrying sufficient information in a reasonable time. A 1.5T Siemens system is used for the acquisition. Due to clinical and technical constraints, high resolution imaging of the complete bones is not applicable. However, the coverage of the full bones is essential to define anatomical coordinate systems (Wu, et al. 2002). To tackle this issue, fast (3 min approx.) but low resolution acquisitions covering femur and hip bones are first performed. Then high resolution acquisitions, exclusively focused on the joint area, are performed (4 min per hip). This combination of various datasets provides enough information to the segmentation procedure, and avoids excessive acquisition times (Gilles 2007) (Arbabi, Schmid, et al. 2009).

4.4.2.1.2 MRI Segmentation

The MRI datasets obtained by the described protocol are segmented to get models of the bones (femur and hip bone) and hip cartilages (femoral cartilage, acetabular cartilage, and labrum). The segmentation method based on (Gilles, Moccozet and Magnenat-Thalmann 2006), (Gilles 2007), and (Schmid and Magnenat-Thalmann 2008) is briefly presented as follows.

Generic models of the bones and cartilages were built once using an interactive segmentation tool. When a new MRI volume needs to be segmented, the generic model is coarsely positioned into the MRI volume. A registration approach, which consists in deforming the generic models to match patient-specific anatomical boundaries, is then adopted. Generic models vertices are considered as lumped mass particles evolving under the Newtonian law of motion. A particle is subjected to internal and external forces in a multi-resolution scheme. Internal forces ensure smoothness and exploit prior knowledge of the models to create constraints (e.g. volume preservation, medial surfaces constraints, and shape priors). External forces use topological (e.g. cartilage-bone attachments), image (maximization of intensity based similarity measures) and non-penetration (hierarchical collision detection (Teschner, Kimmerle, et al. 2005) and response (Volino and Magnenat-

---

[12] The 3D Meshes and HJCs were prepared by Jerome Schmid at MIRALab, University of Geneva.

Thalmann 2000)) constraints. The first-order differential equation system relating forces to particle state (position and velocity) is resolved by a stable implicit scheme (Baraff and Witkin 1998), (Volino and Magnenat-Thalmann 2000). The segmentation is validated in experiments (in this research, a mean accuracy of 1.5 mm was reported for the bones segmentation), and the quality of the cartilages segmentation are visually validated by medical experts. Figure 64 shows some examples of the segmented 3D models.

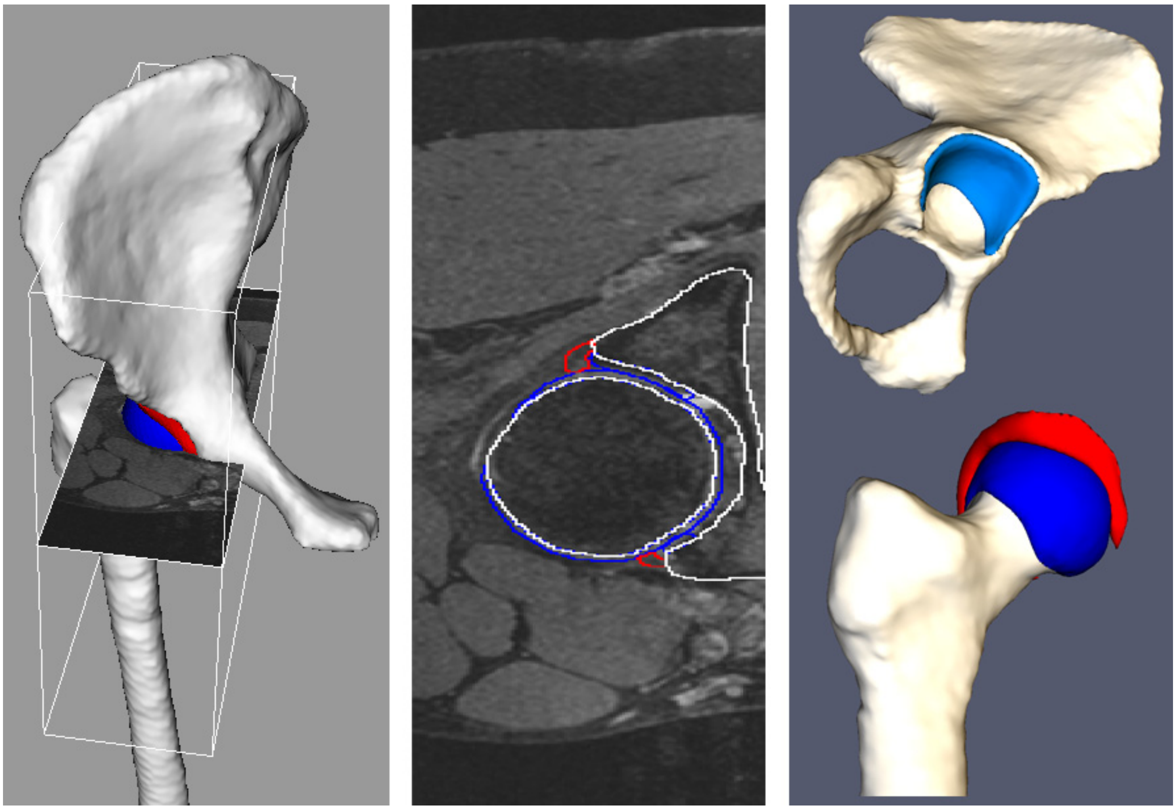

Figure 64: Left and Middle: Example of the segmentation overlay in a high resolution dataset focused on the hip joint. Blue color indicates femoral and acetabular cartilages, while red is used for labrum. Right: Example of reconstructed model (right femur and hip bone with femoral cartilage, acetabular cartilage and labrum). The method is based on (Gilles, Moccozet and Magnenat-Thalmann 2006), (Gilles 2007) and (Schmid and Magnenat-Thalmann 2008); (Image courtesy of MIRALab, University of Geneva).

4.4.2.2 HJC estimation methods

In (Gilles 2007) three predictive and two functional approaches have been depicted. These approaches were used by Jerome Schmid at MIRALab, University of Geneva for estimating HJCs. They are described here briefly.

4.4.2.2.1 Predictive approaches

The HJC is estimated as the center of the sphere that approximates the best the femoral head or the acetabulum. The approximation is thus a least square fitting, which aims to find the center (and radius) of the fitted sphere to the reconstructed data. These two methods are denoted as *femoralheadsphere* and *acetabulumsphere* methods. They do not lead to the same result due to an inhomogeneous inter-articular distance. The *doublesphere* approach considers the joint as a perfect ball and socket joint where inter-articular distance is constant. It aims at finding the common center of the femoral and acetabulum spheres (both spheres are centered on the same point). The fitting process for a simple sphere is then extended in this case.

4.4.2.2.2 Functional approaches

The main idea in the functional approach is to enforce a certain inter-articular distance: given each vertex P_i of the acetabulum, the difference $|d_i - d_i^{ref}|$ is minimized, where d_i (d_i^{ref}) denotes the (reference) distance between the vertex P_i and the femoral head. For a given joint transform, a minimization process is used to minimize the differences $|d_i - d_i^{ref}|$ through hip bone infinitesimal translations. This process requires an initialization. Two different approaches are thus explored: (i) the *dconst* approach that uses a *constant* reference distance for *all* the vertices equal to the radii difference between the fitted femoral and acetabulum spheres, (ii) the *dref* approach that uses the initial distances acquired in the reconstructed position.

In a simulated motion, each joint transform can be hence optimized in terms of shifts to seek reference inter-articular distances. The HJC is then considered as the point of the femur which moves the less in the hip bone frame during the optimized motion. A global minimization technique is thus applied to identify the HJC. The chosen simulated motion is a circumduction pattern (Kang, Sadri, et al. 2003) with elevation set to 20°.

4.4.2.3 Contact management

As it was explained in Chapter 3, when two objects collide with each other during rotation, two kinds of penetration may occur: (i) tangential or (ii) radial. The curvilinear penetration happens in the angular direction that is tangential to the rotational trajectory (such as when femur bone collides with labrum). On the other hand, the radial penetration usually happens among the surfaces that are sliding on each other during rotation (such as when femur cartilage slides under acetabular cartilage). Therefore, two kinds of penetration are investigated:

4.4.2.3.1 Curvilinear penetration between femur and labrum

By using the proposed cylindrical segmenting method (see section 3.2), femur bone is cylindrically segmented and polygon indices are stored in corresponding table cell(s). Then the penetrating labrum vertices are detected by determining the table cell they belong to and checking potential collision with the femur polygons stored in the cell, along a circular trajectory. The penetration depth of a labrum vertex is estimated by calculating the smallest distance that the femur bone needs to be moved away in its rotational trajectory in order to leave the penetrated vertex of labrum out (Figure 65).

4.4.2.3.2 Radial penetration between femoral cartilage and the acetabular soft tissues (consisting of acetabular cartilage and labrum)

The penetration among femur and acetabular soft tissues are evaluated by using radial segmenting method proposed in section 3.3. In this method, instead of applying cylindrical segmentation, the acetabular soft tissues are segmented radially (originating from center of rotation). Therefore, the returning penetration depths are calculated by finding the radial distance between each femur cartilage vertex and the acetabular polygons occupying the same radial segment of the vertex. The method is also repeated by exchanging the role of femur cartilage and acetabular soft tissues (Figure 66).

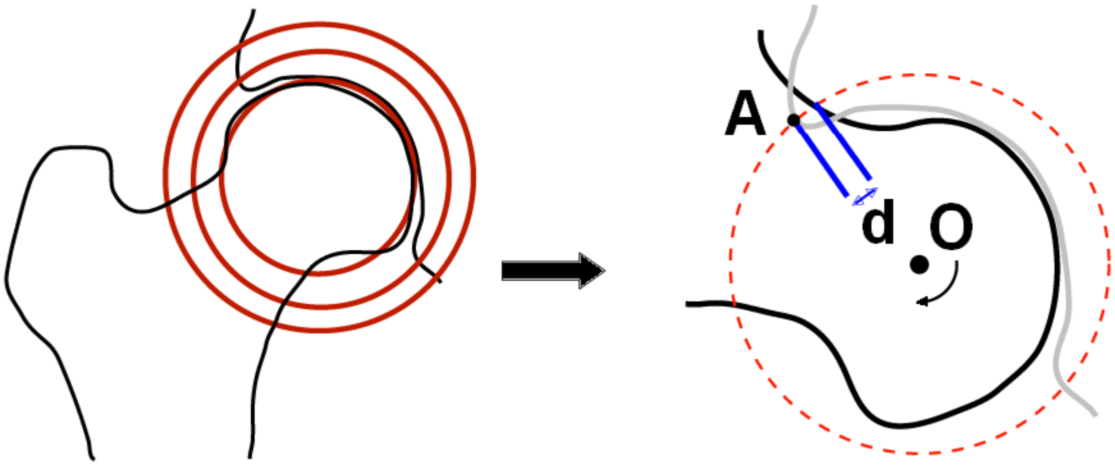

Figure 65: Left: Cylindrical segmentation of the space around the hip joint. Right: Curvilinear penetration depth obtained by cylindrical segmenting method (‘d’ represents the amount of curvilinear penetration depth of vertex ‘A’ located on the labrum (light colored) inside femur bone (dark colored), when the bone is rotating about ‘O’).

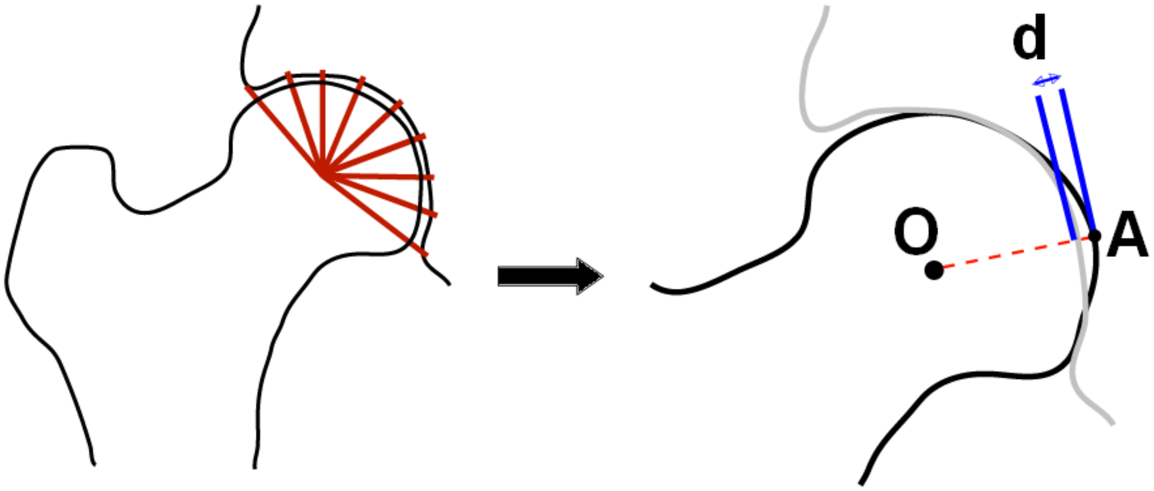

Figure 66: Left: Radial segmentation of the space around the hip joint. Right: Radial penetration depth obtained by radial segmenting method. ‘d’ represents the amount of radial penetration depth of vertex ‘A’ located on the femur cartilage (dark colored) inside acetabular cartilage (light colored), when the bone is rotating about ‘O’.

4.4.3 Medical experiment

Ten healthy female subjects (average age: 24) were selected by MIRALab, University of Geneva for the following study. All the experiments were approved by subjects and the ethical committees. The 3D meshes of the patients hip tissues based on MR Images and their centers of rotation (estimated based on 5 different approaches) were prepared by Jerome Schmid at MIRALab, University of Geneva. We simulated a patient’s hip meshes by rotating the hip about 5 different HJCs, and calculating tissues contact penetration depths (curvilinear and radial). We did these rotations and calculations for all the patients (totally 50 simulations). For simplicity we name each simulation based on the method used for estimating HJC and the number of patients. For example *SIM(doublesphere, 3)* stands for the simulation done for the third patient by considering the HJC estimated by doublesphere. In each simulation, we rotated the femur bone about its medical axes in order to perform Internal Rotation, External Rotation, Abduction, Adduction, Extension, and Flexion, based on the standards explained in (Wu, et al. 2002). The maximum amount of rotation was derived from the investigation done in (Genoud, et al. 2000) about hip range of motion. For each degree of rotation we estimated the maximum value of both curvilinear penetration depth (when femur bone collides with labrum) and radial penetration depth (when femoral cartilage slides under the acetabular cartilage and labrum). Therefore, for each patient we obtained the curves showing how the maximum penetration depth changes during rotation about each medical axis, by considering each of estimated HJC for the corresponding patient (see Figure 67 as an example).

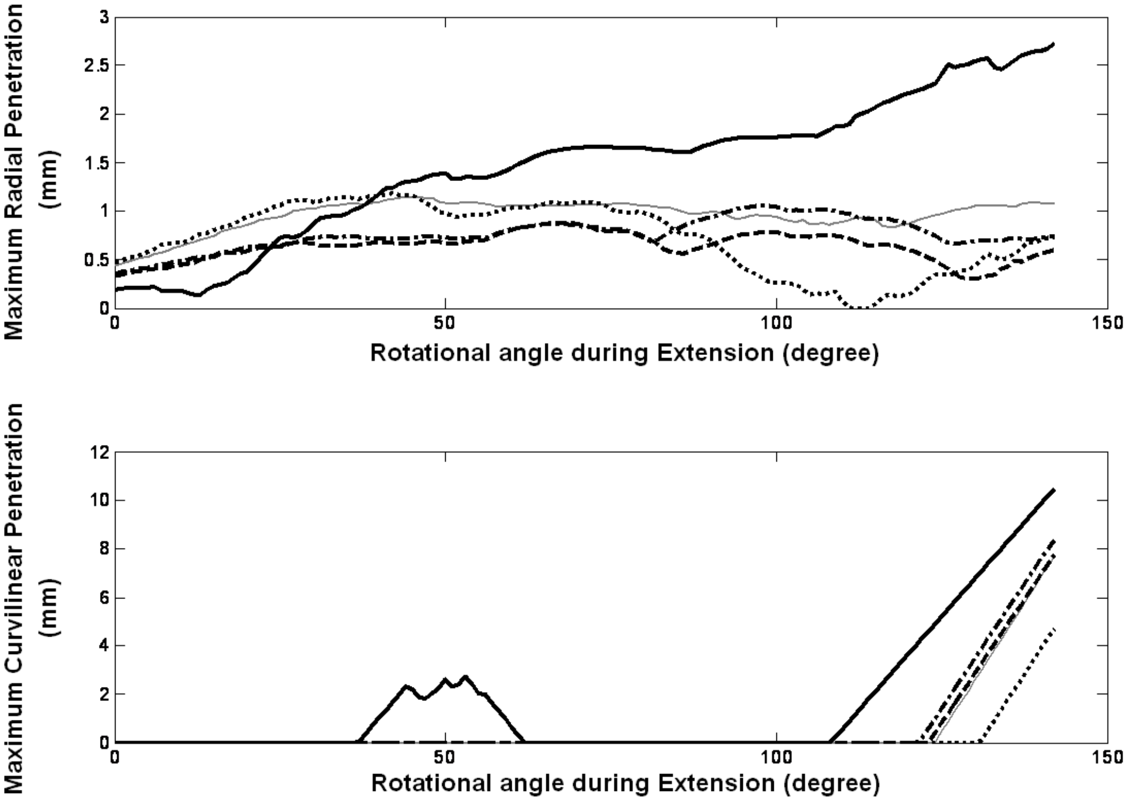

Figure 67: Maximum amount of radial (upper) and curvilinear (bottom) penetration depths during hip Extension of one of the patients. The methods used for estimating HJC are dref (thin solid curve), acetabulumsphere (dotted curve), doublesphere (dashed curve), femoralheadsphere (dot-dashed curve), and dconst (thick solid curve).

For simplicity, we represent the obtained results of each simulation by following arrays:

$$\mathrm{Array}^{ANG} = [FL^{ANG}, EX^{ANG}, IR^{ANG}, ER^{ANG}, AD^{ANG}, AB^{ANG}], \qquad \text{Eq. 12}$$
$$\mathrm{Array}^{RAD} = [FL^{RAD}, EX^{RAD}, IR^{RAD}, ER^{RAD}, AD^{RAD}, AB^{RAD}],$$

where $FL^{ANG/RAD}$, $EX^{ANG/RAD}$, $IR^{ANG/RAD}$, $ER^{ANG/RAD}$, $AD^{ANG/RAD}$ and $AB^{ANG/RAD}$ correspond to the array of maximum Angular(curvilinear)/Radial penetration depth during Flexion, Extension, Internal Rotation, External Rotation, Adduction and Abduction respectively.

4.4.4 Results

4.4.4.1 Results series 1: per patient

Because a real hip joint does not have any known center of rotation, in order to investigate the effect of HJC in penetration depth, we considered one of the methods for estimating HJC as a reference method and compared the other methods with it. In (Gilles 2007) dref has been presented as a method for estimating HJC with smaller error; therefore we considered the results of SIM(dref, X) as our reference values for patient number X. It should be noticed that choosing a reference method is just for simplifying the comparison, and having different reference methods returns the same conclusion.

To have a general evaluation, we first considered all the rotations together. Then for each simulation we estimated array of absolute differences in penetration depth (compared to the reference simulation):

$\text{DeltaArray}^{ANG}_{SIM(i,\,j)} = |\,\text{Array}^{ANG}_{SIM(i,\,j)} - \text{Array}^{ANG}_{SIM(dref,\,j)}\,|$, Eq. 13

$\text{DeltaArray}^{RAD}_{SIM(i,\,j)} = |\,\text{Array}^{RAD}_{SIM(i,\,j)} - \text{Array}^{RAD}_{SIM(dref,\,j)}\,|$,

i = {acetabulumsphere, doublesphere, femoralheadsphere, dconst},

j = {0 to 10},

where $\text{DeltaArray}^{ANG}_{SIM(i,\,j)}$ and $\text{DeltaArray}^{RAD}_{SIM(i,\,j)}$ demonstrate the absolute difference between the maximum penetration depths of method i compared to dref, for patient j. In order to have percentage of differences instead of absolute differences, we estimated another array ($\text{PercentageDeltaArray}^{RAD}_{SIM(i,\,j)}$) by dividing each value of $\text{DeltaArray}^{RAD}_{SIM(i,\,j)}$ to its corresponding value in $\text{Array}^{RAD}_{SIM(dref,\,j)}$. Because curvilinear penetration only happens when the femoral bone collides with labrum, its array also included zero (no penetration). Thus, the percentage of differences could not be calculated for curvilinear penetration depths. Finally, mean and standard deviation of each series of $\text{DeltaArray}^{ANG}_{SIM(i,\,j)}$ and $\text{PercentageDeltaArray}^{RAD}_{SIM(i,\,j)}$ were calculated (Figure 68 and Figure 69).

4.4.4.2 Results series 2: per medical axis of rotation

In addition to investigating changes in the penetration depth during all the rotations (together), we also investigated the changes in the penetration depth for rotating about each medical axis separately. In this case, instead of focusing on separate patients, we considered all the patients together and focused on different medical rotations.

By considering Eq. 12, we calculated:

$\text{DeltaMeanXY}^{ANG}_{SIM(i,\,j)} = \text{MEAN}(\,|\text{XY}^{ANG}_{SIM(i,\,j)} - \text{XY}^{ANG}_{SIM(dref,\,j)}\,|\,)$, Eq. 14

XY = { FL, EX, IR, ER, AD, AB },

i = {acetabulumsphere, doublesphere, femoralheadsphere, dconst},

j = {0 to 10}.

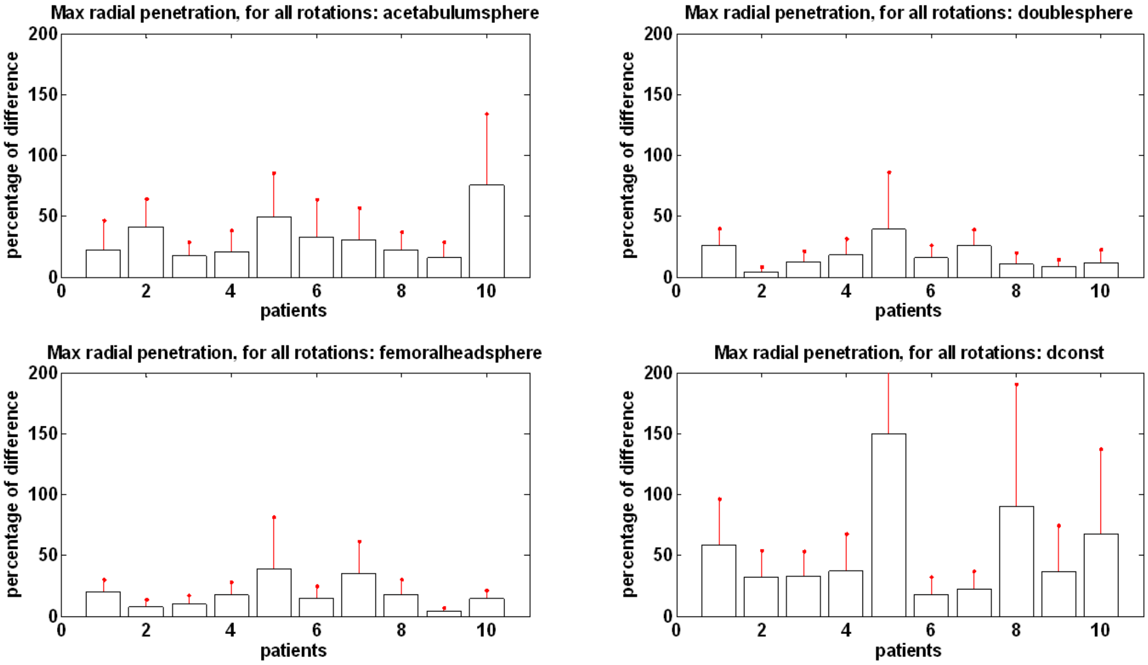

Figure 68: Maximum difference between radial penetration depths based on dref HJC estimation and radial penetration depths based on other estimation methods (acetabulumsphere (upper-left), doublesphere (upper-right), femoralheadsphere (bottom-left), and dconst (bottom-right)) are calculated during all 6 types of medical rotations (all together). For having the percentage of these differences, the values are divided by the maximum penetration depths based on dref HJC estimation. The mean values of these percentages are shown as tick bars, where each bar represents one of the ten patients. The thin line above each bar depicts the standard deviation.

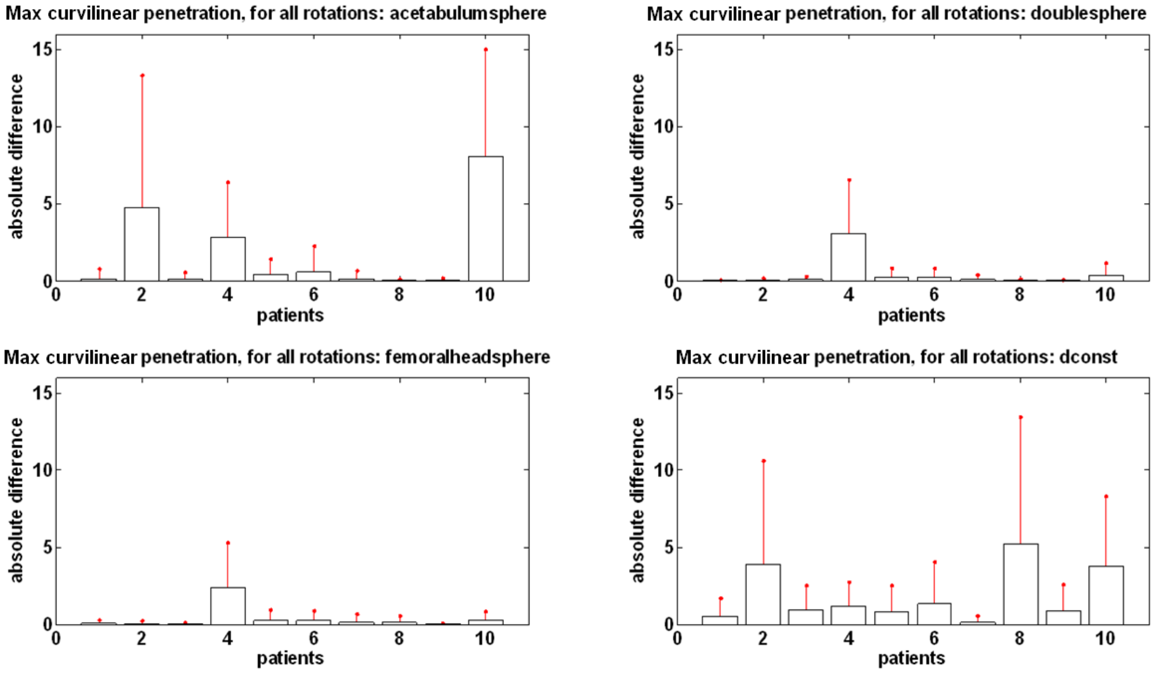

Figure 69: Maximum difference between curvilinear penetration depths based on dref HJC estimation and curvilinear penetration depths based on other estimation methods (acetabulumsphere (upper-left), doublesphere (upper-right), femoralheadsphere (bottom-left), and dconst (bottom-right)) are calculated during all 6 types of medical rotations (all together). The mean values of these data are shown as tick bars (in mm), where each bar represents one of the ten patients. The thin line above each bar is showing the standard deviation of the data.

$\text{DeltaMeanXY}^{\text{ANG}}_{\text{SIM(i, j)}}$ stands for the mean value of the difference between *maximum penetration by using type i of HJC* and *maximum penetration by using dref type of HJC*, during XY type of medical rotation for the j^th^ patient. By having $\text{DeltaMeanXY}^{\text{ANG}}_{\text{SIM(i, j)}}$ for each patient we created a new array including all the patients together:

$$\text{DeltaMeanArrayXY}^{\text{ANG}}_{\text{SIM(i)}} = \qquad \text{Eq. 15}$$

$$[\text{DeltaMeanXY}^{\text{ANG}}_{\text{SIM(i, 0)}};\text{DeltaMeanXY}^{\text{ANG}}_{\text{SIM(i, 1)}};\ldots;\text{DeltaMeanXY}^{\text{ANG}}_{\text{SIM(i, 10)}}]$$

i = {acetabulumsphere, doublesphere, femoralheadsphere, dconst}.

The same computations were also done for Radial type of penetration. However, instead of calculating absolute difference, we calculated percentage of differences by dividing the absolute differences to the corresponding values when dref method is used. These calculations led us to have $\text{PercentageDeltaMeanArrayXY}^{\text{RAD}}_{\text{SIM(i)}}$.

$\text{DeltaMeanArrayXY}^{\text{ANG}}_{\text{SIM(i)}}$ and $\text{PercentageDeltaMeanArrayXY}^{\text{RAD}}_{\text{SIM(i)}}$ demonstrate how penetration depth differs for method i of HJC estimation compared to the dref method, for a specific medical rotation (XL), by considering all the patients. Thus, we calculated mean, maximum and minimum values of $\text{DeltaMeanArrayXY}^{\text{ANG}}_{\text{SIM(i)}}$ and $\text{PercentageDeltaMeanArrayXY}^{\text{RAD}}_{\text{SIM(i)}}$ (Figure 70 and Figure 71).

4.4.5 Discussion

When the penetration depth for different patients are investigated separately (results series 1), it can be seen that for almost all of the estimated HJCs, radial penetration depth differs considerably (about 30% in average) compared to when HJC is estimated based on dref. Yet, these differences in estimated penetration depth are not the same for all of HJCs. The changes in the penetration depth are smaller when doublesphere and femoralheadsphere methods are used for estimating HJC (less than 19% in radial case and less than 0.42 mm in angular case, in average). The differences increase to about 55% in radial case and 1.87 mm in angular case (in average) for all the patients when the dconst method is used. Such amount of differences highlights the fact that the hip simulations can give different results and conclusions when the HJC estimation method shifts to one another, (especially from dref to dconst). Thus, the medical results of different hip simulations are not comparable if the hip joint center is estimated by different or unknown methods.

These results also show that the difference in penetration depth is patient oriented. For example, results of series 1 show that patient 5 usually has the highest difference in radial penetration depth, when the HJC estimation method changes. Also, patient 4 has the highest difference in curvilinear penetration depth compared to the other patients, when doublesphere and femoralheadsphere methods are applied for estimating HJC. In addition, it demonstrates that radial penetration and curvilinear penetration depths may vary in different manners when the method for estimating HJC changes.

These results suggest that the dconst and acetabulumsphere present the highest variability with respect to the femoralheadsphere and doublesphere methods. This can be explained by various factors. First of all, the dconst method was depicted in (Gilles 2007) as the worst method for computing the joint center with respect to the other approaches. Secondly, the performance of the acetabulumsphere can be explained by errors in the reconstruction of the acetabulum. Indeed, it is more difficult to segment the acetabulum than the femoral head. As a result the sphere fitting will be affected by these errors. This also explains why the doublesphere method seems to be less sensitive than the acetabulumsphere method, as errors of the acetabulum segmentation are somehow compensated by the higher accuracy in the femoral head reconstruction.

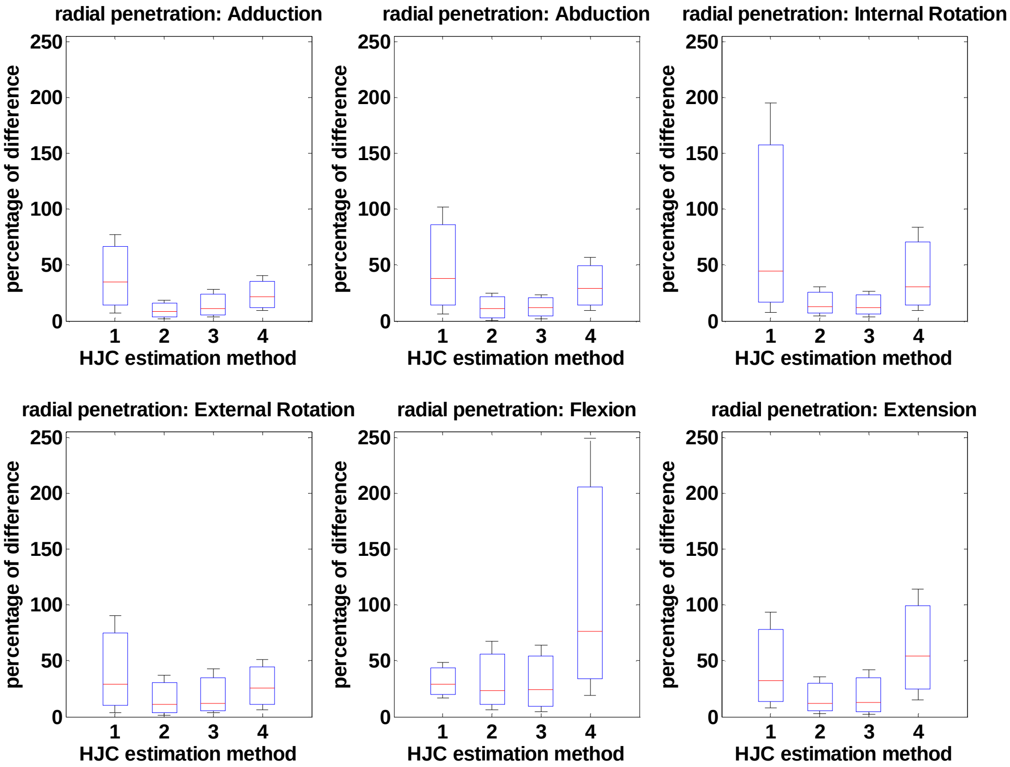

Figure 70: Maximum difference between radial penetration depths based on dref HJC estimation and radial penetration depths based on the other estimation methods are calculated for all the patients together and during each medical rotation separately (Adduction (upper-left), Abduction (upper-middle), Internal Rotation (upper-right), External Rotation (bottom-left), Flexion (bottom-middle), and Extension (bottom-right)). For having the percentage of these differences, the values are divided by the maximum penetration depths based on dref HJC estimation. The mean values of these percentages are shown as a median line of the tick bars, where each bar represents difference between dref method with one of the other methods for estimating HJC (acetabulumsphere (bar #1), doublesphere (bar #2), femoralheadsphere (bar #3), and dconst (bar #4)). The T-shape end limit on each bar represents the range of these percentages (maximum and minimum), (the tick bars are expanded up to ¾ of their range).

Cartilages are still more difficult to segment compared to bones due to their thickness and their poor contrast in some images. Radiologists often use contrast agent injection to increase the visibility of the cartilages. This operation was not used in our experiments as it remains an invasive procedure. Cartilage segmentation remains hence difficult and resulting errors can as well influence the results. This can explain why some discrepancies exist among the various patients as the segmentation quality can vary from one dataset to another.

When the investigation is focusing on different axis of rotation (results series 2), in almost all of the cases using acetabulumsphere and dconst methods for estimating HJC cause the highest change in both radial and curvilinear penetration. This is consistent with the previous remarks. However, radial and curvilinear penetration depth during Adduction and External Rotation shows to be less sensitive to the methods used for estimating HJC (compared to the other types of rotation). On the other hand, Flexion is among the most sensitive types of rotation to the HJC estimation methods. Therefore, hip simulation results (obtained based on different or unknown HJC estimation method) are more robust and comparable when the hip movement is closer to Adduction or External Rotation, rather than Flexion.

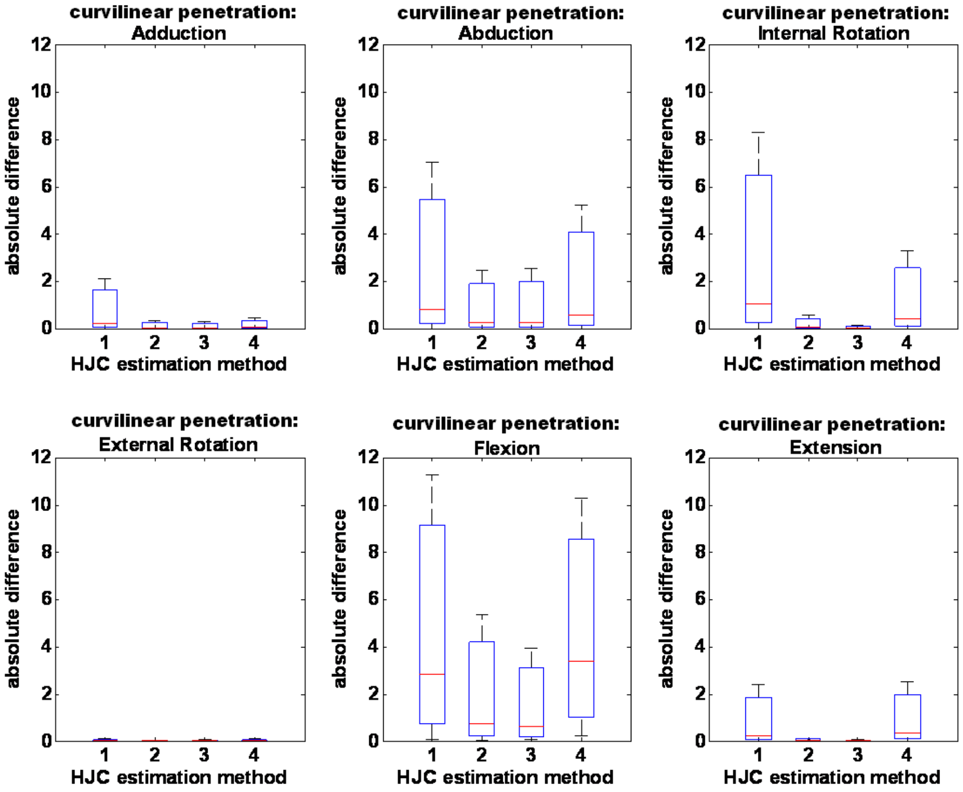

Figure 71: Maximum difference between curvilinear penetration depths based on dref HJC estimation and curvilinear penetration depths based on the other estimation methods are calculated for all the patients together and during each medical rotation separately (Adduction (upper-left), Abduction (upper-middle), Internal Rotation (upper-right), External Rotation (bottom-left), Flexion (bottom-middle), and Extension (bottom-right)). The mean values of these data (in mm) are shown as a median line of the tick bars, where each bar represents difference between dref method with one of the other methods for estimating HJC (acetabulumsphere (bar #1), doublesphere (bar #2), femoralheadsphere (bar #3), and dconst (bar #4)). The T-shape end limit on each bar represents the range of these data (maximum and minimum in mm), (the tick bars are expanded up to ¾ of their range).

As a summary, the results indicate that hip medical investigations are not robust when the HJC estimation method changes. In fact, researchers should be careful in choosing the methods of HJC estimation, before providing any conclusion from their medical research. In any case, the HJC estimation methods that should be considered are those that are reported to be more accurate. For example, the dconst method should be avoided. In addition, special attention should be paid in assessing or considering the errors made in the reconstruction as they have an impact on the results. Presented HJC estimation methods are indeed affected by these segmentation errors. Also, the meshes quality and their resolution can affect the accuracy of the penetration depth computation.

Finally, in order to reduce the negative influence of such sensitivity on the hip joint research, and to have comparable simulation results, a standard method for estimating HJC should be decided and used in all the future research. For finding the standard method, different kinds of measurement can be used. However, because HJC is affecting the simulations, we suggest that the tissue contact penetration depths during the real hip movement (non-idealized) be calculated by gradually recording the relative position of the hip tissues, when the patients move their hip in the medical directions. The results should be then compared with the contact penetration depths calculated during idealized hip movement (when the hip joints are rotating about their estimated HJC). A HJC estimation method, which returns the smallest difference in penetration depths for a statistically significant number of patients (compared to the real movement), can be recommended as the standard method.

4.4.6 Conclusion

In this section, we investigated the sensitivity of the penetration depth of hip tissues to the methods applied for estimating hip joint center of rotation. Two different kinds of penetration depths were considered (radial and angular) and the investigations were done for 10 different patients and for 6 types of medical rotation. This investigation highlights the importance of the HJC estimation methods because of their influence on computer-aided medical research and diagnosis, and suggests finding a standard method for estimating HJC.

4.5 Estimating hip joint contact pressure from hip mesh geometric features

4.5.1 Introduction

The mechanical contributions of the cartilage layers inside a joint include articulation and load transfer, during which two cartilage layers exert contact pressure on each other. This pressure has been a notable parameter to evaluate the physical conditions inside the hip joint. High pressures are shown to be in association with soft tissue damage within the hip joint. Therefore, in-vivo or in-vitro measurements of contact pressure within the hip joint have been performed parallel to the development of predictive tools (Hodge, et al. 1986).

Many computational approaches estimate the pressure and contact pressures via finite element methods (FEM) (Russell, et al. 2006), during the loads and motions of daily activities, by using 3D meshes of the tissues (Chegini, Beck and Ferguson 2006). Although this type of simulation can provide a good evaluation of hip problems, the process may be very time-consuming and unsuitable for fast medical hip simulations. Also, lack of the details related to the joint movement can completely prevent the simulation to be performed.

In this section, we propose a statistical model for estimating hip contact pressures during its movement. The estimation is done by extracting and evaluating the geometrical information of the target hip and some other hip models which are already mechanically investigated. The method does not require any details about the joint movement, and having 3D meshes of the joints is sufficient. The method has been also examined on 25 different hip models, and the estimated pressures were compared to the FEM-based calculated pressures.

4.5.2 Method

Some of our hip models have been already evaluated mechanically (training models), and we want to evaluate an unknown hip model (target model) based on these training models. The contact pressures of the training models during a specific movement are available. The goal is to estimate the pressures of the target model during the same movement, by using the key geometrical features of the training models. The main steps of the method can be listed as:

1- Geometrical feature extraction and reduction, from the training and target models;
2- Finding the closest training models to the target model, based on the features;
3- Pressure estimation based on the closest training models.

For simplicity, let's consider that we have one target hip model, and the number of training hip models is '**m**' (>0). For each training hip model, maximum contact pressures in '**p**' steps of hip movement are available (i.e. step of rotation during hip movement):

$$\{\text{Pressure values of the } i^{th} \text{ Training Set}\} = \{\{Pr_j\}_i\}, \qquad \text{Eq. 16}$$

$$\text{Pressure values of the Target Set} = \{Pr_j\}_{target}, \quad i=1 \text{ to } \mathbf{m} \;\&\; j=1 \text{ to } \mathbf{p}.$$

For estimating the pressure of the target model, we extract ‘**f**’ number of features from all the hip models (training and target sets):

Training Features = {Feature array of the i^{th} Training Set}={ $\{F_l\}_i$ } , Eq. 17

Target Features = Feature array of Target Set = $\{F_l\}_{target}$, i=1 to **m** & l=1 to **f**.

4.5.2.1 Geometrical feature extraction

Since hip 3D meshes may have different resolutions, we cannot use the raw 3D coordinates of the hip vertices as input features. Instead, we sample the tissue meshes with constant sampling steps for all the hip models. Due to spherical shape of hip joint, the sampling is performed in the spherical coordinate system, rather than Cartesian coordinate system. For each sample point the distance between the sample point and the joint center (R) is stored as a feature.

Tissue sampling provides a large number of R values as feature. Processing this large number of data can slow down the total pressure estimation process and consequently make the process less efficient. The principal component analysis (PCA) discriminates directions with the largest variance in a data set for identifying the most representative features (Malhi and Gao 2004). We apply PCA on the normalized features (R’s) and keep ‘**e**’ number of the most effective output features.

{Effective Features of the i^{th} Training Set} = { $\{EF_k\}_i$ }, i=1 to **m** & k=1 to **e**. Eq. 18

The most effective features of the target set is also found by using the same PCA parameters calculated for training sets.

Target Effective Features = $\{EF_k\}_{target}$, k=1 to **e**. Eq. 19

It is possible that a geometric feature is considered important by PCA, however it does not have much effects on the contact pressure. Thus, in order to find the pressure-correlated features in each step of hip movement, we calculate the statistical correlation coefficient between each effective feature and the contact pressure at a specific hip position (i.e. step of rotation), for all the training hip models. Based on these correlation coefficients, the features can be ranked and **e'** number of the highest ranked ones are considered for later computations. Since the pressures are related to a range of motion rather than just a specific hip position, these correlation coefficients calculations are done for all the hip movement steps:

Correlation Coefficients = Eq. 20

{Correlation coefficient between “pressure at the j^{th} step of hip movement” and “the k^{th} Effective feature”, for all training sets} =

{Correlation coefficient between $Pr_{j,i=1 \text{ to } \mathbf{m}}$ and $EF_{k,i=1 \text{ to } \mathbf{m}}$} = { $\{CC_k\}_j$ },

k=1 to **e** & j=1 to **p**.

We also define HCC(**e'**,j) as a set containing the indices of **e'** (**e**) number of the effective features with the highest correlation coefficient at the j^{th} step of hip movement.

HCC(**e'**,j) = Eq. 21

{k | CC_{kj} is among the **e'** number of the highest correlation coefficients}

4.5.2.2 Finding the closest set per feature

Among all the training sets we search for the set which has the smallest distance to the target set. It is possible that a training hip model be close to the target hip model only based on one geometric feature. If this geometric feature is affecting the pressure more than any other features (in a step of movement), we can estimate the contact pressure of the target model by using the training model which is close to the target model only based on this feature. Thus, the distances are calculated by using the extracted features. For each feature (k), we find the training set (i) which returns the smallest distance between $EF_{k,target}$ and all the $EF_{k,i}$.

Closest Set Corresponding to All the Effective Features = Eq. 22

{Closest Set for Feature k} = {CS_k} = {$MIN|_i$(DIST<$EF_{k,target}$,$EF_{k,i}$>)},

i=1 to **m** & k=1 to **e.**

There are different methods for finding distance between two sets (Theodoridis and Koutroumbas 2003). For finding the smallest distance between $EF_{k,target}$ and all the $EF_{k,i}$ ’s, two kinds of Euclidean distances are considered: DIST1 and DIST2. $DIST1(B,A)_k$ is Euclidean distance between array A and array B, when the k^{th} elements in the both arrays are ignored, which demonstrates how much the difference between sets of A and B is depending on the k^{th} element. $DIST2(B,A)_k$ is the absolute difference between the k^{th} elements in array A and array B. It demonstrates how the k^{th} element of sets of A and B are close to each other. For each feature, we first find 4 training sets with the smallest DIST1 to the target set and then among these 4 training sets, we selected the one with smallest DIST2 (4 was chosen by trial). At the end, ‘**e**’ number of training sets, corresponding to ‘**e**’ number of effective features, is found (with possible repetition).

4.5.2.3 Estimating pressures

The pressure of the target hip at the j^{th} step of movement is estimated as the weighted average pressures of the **e'** number of closest training sets to the target set (i.e. CS), corresponding to **e'** number of high correlated effective features. The weights are the correlation coefficients of each effective feature ($\{CC_k\}_j$):

$$Pr_{j,target} = \quad ^{k \in HCC(\mathbf{e'},j)} (CC_{k,j} * Pr_{j,i}) \;/\; ^{k \in HCC(\mathbf{e'},y)} (CC_{k,j}) \;,\; \text{where } i = CS_k . \qquad \text{Eq. 23}$$

4.5.3 Testing the method on different hip models

4.5.3.1 Hip models

As it was explained before (see section 4.3), α and CE angles are two parameters which can characterize some hip joint pathologies. The CE angle (Figure 58: right) is defined as the angle formed by the perpendicular to the inter-tear drop line and the line passing from the center of the femoral head to the lateral edge of the acetabulum (Wiberg 1939). The α angle is formed by the femoral neck axis and a line connecting the center of the femoral head with the point of beginning a sphericity (Nötzli, et al. 2002) (Figure 58: left).

We used the same hip models explained in section 4.3.2.2, prepared by Salman Chegini at ARTORG, University of Bern, by using CAD software[13]. They were created with variety of α and CE angles to cover a wide range of hip geometries. The CE angle values were 0°, 10°, 20°, 30°, 40° degrees and α angles were 40°, 50°, 60°, 70°, and 80°. Combination of all possibilities of the α and CE angles produced 25 joints varying from normal (CE=20°, α=40°) to a dysplastic (CE=0°, α=40°), cam (CE=20°, α=80°), and pincer joint (CE=40°, α=40°) (see Figure 59) (Chegini, Beck and Ferguson 2008).

Since α and CE angles are geometrical parameters, they could be also included in the feature lists. In fact, having α and CE angles can increase the accuracy of the estimation as they are correlated to the hip joint pathology. However, calculating α and CE angles in real human hip joint may not be straightforward. Especially for α angle, the hip head has to be modeled as an ideal sphere in order to find the point where femoral head diverges from spherical (Nötzli, et al. 2002). Thus, such values can be highly depending on the methods used for their calculations. In order to have a more robust method, we avoided using α and CE angles among our features.

4.5.3.2 Hip movement

We wanted to test our method during a medically meaningful hip movement. It has been shown that hip impingement is a kinematical problem associated with large range of motion and is most often observed in young, active patients (Miller, et al. 1975), (Bizzini, Notzli and Maffiuletti 2007). Therefore, the motion data for standing-to-sitting were chosen as representative of the frequent daily activities with a relatively large range of motion for comparison (Bergmann, et al. 2001). We divided the total movement to 50 continuous steps, and estimated the pressure for them.

4.5.3.3 Pressure calculation by FEM

The FE simulations of the models were done by Salman Chegini at ARTORG, University of Bern. Similar to section 4.3.2.3, these models were imported to finite element software[14], and the motion data was applied as a prescribed rotational kinematic about the femoral head center. The translation was kept unconstrained and the corresponding joint reaction force vector was simultaneously applied (Chegini, Beck and Ferguson 2008). Cartilage was modeled as a linear elastic material with an elastic modulus of E=12 MPa,

---

[13] Solidworks 2005, Solidworks Corp., Boston, MA, USA

[14] ABAQUS 6.6, ABAQUS Germany Ltd., Aachen, Germany

and Poisson ration of υ=0.45 (Moglo and Shirazi-Adl 2003). The labrum's material properties were E=20 MPa and υ=0.4 (Ferguson, Bryant and Ito 2001), and the bony components were assumed to be rigid in comparison to the soft tissues. As outputs of the simulation, contact pressures were reported. In five pathological models, due to the large deformation, the rotation was stopped before being fully accomplished. These models and the percentage of their accomplished rotation are listed in Table 11.

Table 11: The models without full range of rotation.

α angle	80˚	80˚	80˚	70˚	60˚
CE angle	40˚	30˚	20˚	40˚	40˚
Percentage of the accomplished rotation	50 %	71 %	91 %	83 %	87 %

4.5.3.4 Pressure estimation by using the proposed method

If we consider the total number of hip models as '**m**+1', we trained and tested the method '**m**+1' times. Each time, one of the models was chosen as a target model and the rest (**m**) were used as the training models. The estimated pressures during the rotation were compared with the FEM-based calculated pressures, to evaluate the error. Since five models were not examined during the full rotation (see Table 11), we performed the test for two cases. In the first case, half of the rotation was considered so that all 25 models could be tested. In the second case, the full rotation was considered so that 5 models had to be excluded and 20 models were tested. Totally 1944 features were extracted from femur bone, cartilage and acetabulum soft tissues (**f**=1944), and 10 of them were considered as effective features after applying PCA (**e**=10), which means more than 99% of the features were discarded. We considered **e'** number of the most effective features among these 10 features, based on their correlation coefficients with contact pressures (CC). In order to evaluate the effect of the number of extracted features on the estimated pressures, we considered different values for **e'** (1 to 10) and estimated the pressures for each value of **e'**, separately.

4.5.3.5 Results

For each test, we computed the error by calculating the difference between the estimated pressure and the FEM-based calculated pressure. The difference was divided by the FEM-based calculated pressure, in order to have the percentage of error (see Figure 72). Since the percentage of error was calculated for different steps of rotation, its mean value was calculated during the rotation too. The pressure estimation average error of different hip models (in percentage), when **e'**=10, for the first and the second cases are listed in Table 12 and Table 13.

For overall evaluation of the method, we calculated the mean value of these average percentages of the error among all the models, for a certain value of **e'** (the number of effective features used for the contact pressure estimation). The overall average errors, related to the different estimations done by considering different values of **e'**, are shown in Figure 73 and Figure 74.

4.5.4 Discussion

The results show that the average error of contact pressure estimation is 5% for both cases (Figure 73). However depending on the joint model, the estimation error may vary. For example, as it can be seen in Table 12, when the target joint is chosen from the bordering joint models (hip models with the maximum/minimum α and CE angles), the estimation error is usually more than the error of the others models. The reason can be due to the lack of enough training models geometrically close to these bordering target models. For example, the estimated contact pressure for the target joint model with α = 70° and CE = 30° is 1.11% in the first case, when the model is in the middle of the training models. But, in the second case the same target model is a bordering model, which causes the estimation error increases to 8.06%. In fact, if we just consider the non-bordering models as our target models, the estimation error and its standard deviation decrease significantly. For example in the first case, for **e'**=10, the estimation error is in average 1.62% with a standard deviation of 0.79%, when only the non-bordering target models are considered (the average error is 5.14% with a standard deviation of 9.1%, when the bordering models are also included). This indicates that for having a better estimation it is important to have a wide range of training models in order to ensure that our target models are not bordering.

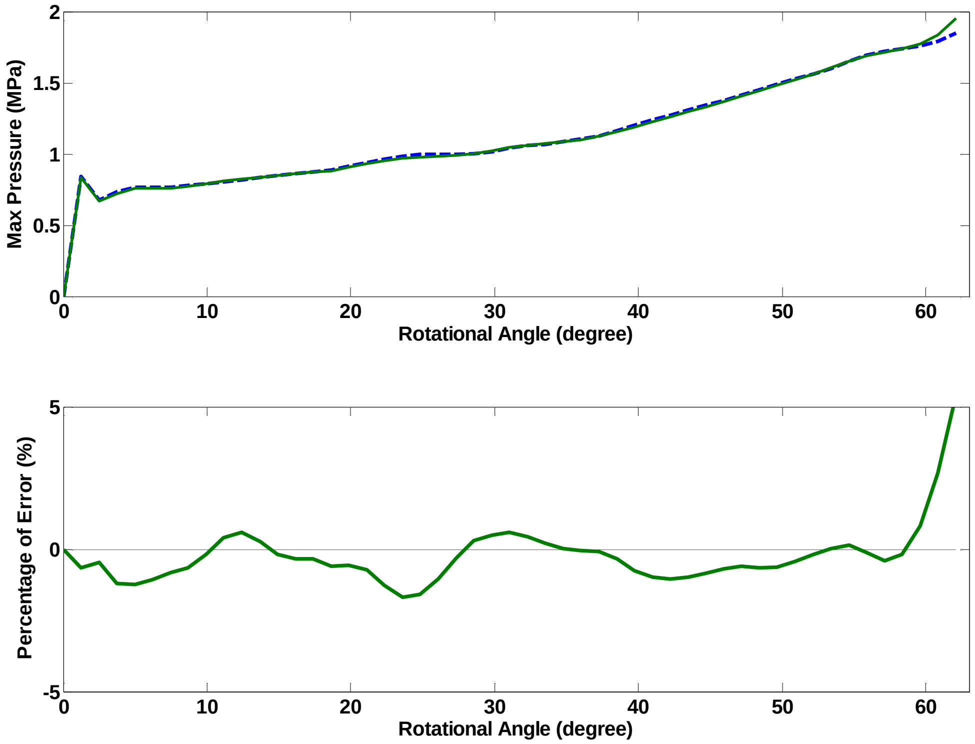

Figure 72: Above: Estimated maximum contact pressure (solid) and FEM-based calculated maximum contact pressure (dashed) (α=60°, CE=20°, **e'**=10); Bottom: Percentage of error (signed).

Table 12: Overall pressure estimation error (percentage) of the hip models for the first case (**e'**=10).

CE angle α angle	0°	10°	20°	30°	40°
40°	4.64	3.75	1.58	1.78	1.85
50°	4.52	2.79	1.84	1.21	43.59
60°	4.93	1.22	0.78	1.00	1.69
70°	4.42	3.00	1.65	1.11	5.34
80°	8.66	1.71	2.41	0.90	22.17

Table 13: Overall pressure estimation error (percentage) of the hip models for the second case (**e'**=10).

CE angle α angle	0˚	10˚	20˚	30˚	40˚
40˚	5.48	1.66	1.63	2.05	4.99
50˚	5.52	2.57	3.21	6.93	17.66
60˚	4.98	1.82	0.72	6.16	-
70˚	5.06	5.74	8.42	8.06	-
80˚	1.24	3.15	-	-	-

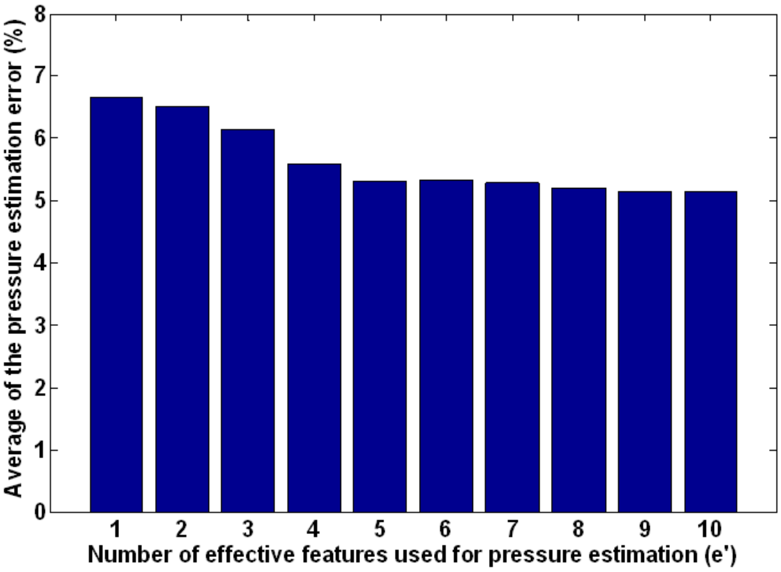

Figure 73: Average of the pressure estimation error (percentages), when **e'** number of effective features is used, for the first case.

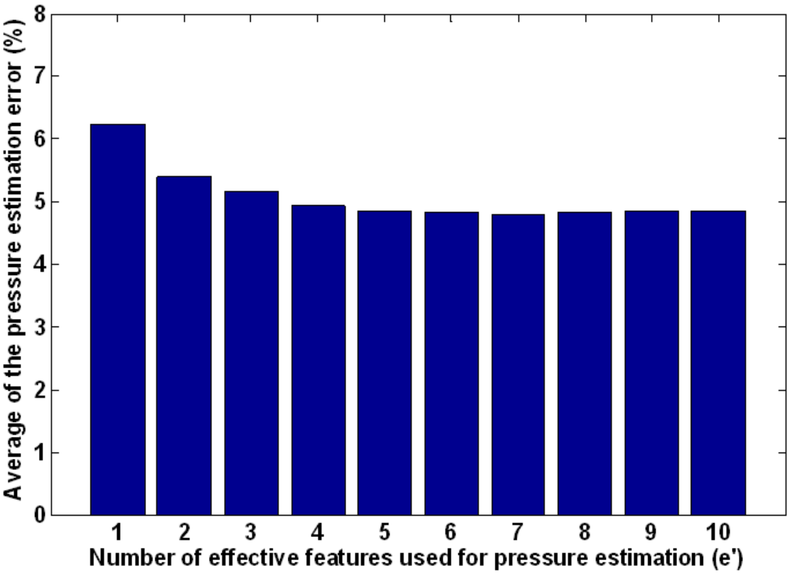

Figure 74: Average of the pressure estimation error (percentages), when **e'** number of effective features is used, for the second case.

In Figure 73, it can be seen that the estimation error decreases more than 1% when the number of used effective features (**e'**) increases from 1 to 5. That shows having more features can help to have better estimation. However, when the number of used effective features increases more (5 to 10), the error changes less than ~0.2%. So, in general in both cases, using 5 numbers of effective features can be enough for getting close to the best estimation. Comparing the first case with the second case shows that the average accuracy of the estimated contact pressures is almost the same. In fact, the average accuracy in the first case is just about 0.6% less than the accuracy in the second case (Arbabi, Chegini, et al. 2008).

4.5.5 Conclusion

In this section we proposed a method for estimating contact pressures of the human joints during their movement. The estimation was done based on training a system by the geometrical features of some joint models, which are already mechanically investigated. The geometrical features were extracted from 3D meshes of the joints at their default posture. The method was later tested on different pathological hip joints during standing-to-sitting (as an example), and the estimated contact pressure was compared with FEM-based contact pressures. The comparison showed that the accuracy of the method was in average about than 95%.

The proposed method suggest a new faster strategy for estimating the contact pressure needed in different biomedical applications, compared to the mechanical models such as FEM (the total processing time in the proposed method was less than 2 seconds even when all the 10 features were used, by using CPU: Xeon-3.4 GHz & RAM: 2GB). In addition to the speed, the method does not need the movement details. Such independency from the movement information can be useful when the movement details are either missing or difficult to mathematically characterize in a simulation.

Chapter 5 Conclusion

5.1 Summary

The main goal of this thesis was to propose different methods for collision detection in computer graphics for rotating or sliding objects. The methods were decided to be mainly used inside human joints, to detect collision among major deformable and/or rigid tissues. Therefore, we first proposed and tested two fast and accurate methods of collision detection for rotating/sliding objects (cylindrical and radial). In addition to finding the collision, the methods could also return penetration depths in either radial or cylindrical direction, which can be useful for further applications.

After proposing the collision detection methods, some biomedical investigations and research were carried on by exploiting the proposed methods. As a first biomedical research, by modifying the pre-processing stage of the cylindrical segmenting collision detection method, a new fast method for finding maximum range of motion in human joint was proposed and tested.

In another work, we used both of the collision detection methods for evaluating hip pathologies. The contact penetration depths of pathological hip models were calculated during hip movement. The results were compared with contact stresses and hip pathologies to find the potential correlation.

By calculating the hip contact penetration depths for ten patients, the sensitivity of hip simulations to hip center of rotation was also evaluated. For this reason, different center of rotations calculated by five methods were applied during hip movement, and both radial and curvilinear contact penetration depths were calculated.

Finally, we proposed and tested a fast statistical model for estimating hip contact pressures during its movement, without performing mechanical simulations and without any need for movement details.

5.2 Contributions

5.2.1 Collision detection for rotating or sliding objects

The two proposed methods for rotating and sliding objects are suitable for different computer graphics applications. Based on the results, it could be seen that the proposed methods are performing faster than the other methods. The methods work accurately without any kind of approximation, and they return details about colliding elements and the penetration depth without any additional computations. Such details can be either used in future simulations (e.g. deformation and collision response) or evaluated directly in some research (e.g. medical evaluations).

In general, the methods can be used in any applications where the objects are rotating or sliding. From medical point of view, both of the methods are able to be used accurately for different kinds of human joint whose movement can be modeled as rotation (ball-and-socket joints such as hip joint and shoulder). In case the joint movement contains some non-negligible translations (e.g. knee), the radial segmenting collision detection method can still be used to return accurately the radial penetration depths.

5.2.2 Joint range finder

The proposed joint range finder method works based on cylindrical segmentation and without using any collision detection algorithm. The method needs to be performed only once per simulation to find both anti clockwise and clockwise range of motion. The experiments showed that the proposed algorithm is faster than the previous methods and can be used for the reverse engineering applications or the biomedical applications such as musculo-skeletal simulation as it is difficult to infer precisely the relative range of motion from scanned or reconstructed 3D models. The estimated range of motion was also an accurate value, which is another advantage comparing to the collision detection based methods, where we have to define a value for rotation steps.

5.2.3 Using contact penetration for hip medical diagnosis

We proposed and validated a new fast strategy based on estimating penetration depths for evaluating hip pathology, suitable for real time medical hip simulations. The proposed method could be shown to be strongly correlated to von Mises stresses of the FE analysis. To date, the differentiation of the individual subtypes was based on a somewhat observer-based judgment which might be subject of substantial misinterpretation and underestimation of pathologies. The penetration depth method allows a more independent differentiation between the subtypes of impingement. The results show how the maximum of the curve related to maximum curvilinear and radial penetration depths can distinguish between different kinds of hip models. In fact, when the hip model is getting closer to pure cam, radial penetration depth is more significant in comparison to curvilinear penetration depth. In contrast, the models closer to pure pincer have more curvilinear penetration depths comparing to radial penetration depth. This could give some diagnostic information, especially in cases with combined impingement where the major component has to be defined and treated.

5.2.4 Evaluating importance of the hip joint center of rotation in medical simulations

We investigated the sensitivity of the penetration depth of hip tissues to the methods applied for estimating hip joint center of rotation. This investigation highlights the importance of the HJC estimation methods because of their influence on computer-aided medical research and diagnosis. The results show that the medical results of different hip simulations are not comparable if the hip joint center is estimated by different or unknown methods. In other words, hip medical investigations are not robust when the HJC estimation method changes. In any case, the HJC estimation methods that should be considered are those that are reported to be more accurate. In order to reduce the negative influence of such sensitivity on the hip joint research, and to have comparable simulation results, a standard method for estimating HJC should be decided and used in all the future research.

5.2.5 Estimating hip joint contact pressure from hip mesh geometric features

The proposed statistical method suggest a new faster strategy for estimating the contact pressure needed in different biomedical applications, compared to the mechanical models such as FEM. The estimation is done based on training a system by the geometrical features of some joint models which are already mechanically investigated. The experimental results show that the average accuracy for the estimated contact pressures is 95%. In addition to the speed, the method does not need the movement details (e.g. axes of rotation and range of movement). Such independency from the movement information can be useful when the movement details are either missing or difficult to mathematically characterize in a simulation.

5.3 Future works

As a future work for collision detection, other types of bounded constrained movement can be explored. By taking advantage of other kinds of constraint, faster but accurate collision detection methods can be proposed for specific applications. For example, the cylindrical segmenting method can be improved in order to handle the cases with non-ideal rotation. Such improvement can adapt the method for non-ideal joints, such as knee. For performing such improvements, we should also check the collision of a vertex in its nearby rings (during negligible translation, only checking the vertex’s containing ring is sufficient). The number of these neighboring rings (cylindrical segments) depends on the size of cylindrical partitions (i.e. ring) and the range of translation.

The proposed strategy for real-time evaluation of FAI works by applying both radial segmenting and cylindrical segmenting collision detection methods. The strategy was tested on CAD hip models created based on consecutive series of α and CE angles. The results were highly promising and could suggest a real time computer-based method for diagnosing and differentiating FAI. However, the strategy still needs some realistic tests and improvements before being applied in medical practices. The method should be improved and adapted for pathological hip models created from real patients data (instead of CAD models). In fact, the difference among real hip models is not only depending on α and CE angles and the other geometrical features may affect it too. Also, the method should be tested for different types of hip movements, and the results should be evaluated separately. In addition to cylindrical and radial penetration depths, new algorithms can be proposed to measure penetration depths in other directions too, in order to find their correlation with hip pathologies. Based on the results, a protocol is needed to be defined for measuring and medically interpreting the hip penetration depths during hip movements. Finally, by improving the methods and defining a final protocol for medical interpretation of the results, an end-user validated real time method can be provided for surgeons and physicians to have a more accurate diagnosis and pre-surgical plan.

The collision detection methods proposed in this thesis can be tested for the other human joints too, to evaluate their potential usability for diagnosing other kinds of joint pathologies. By applying and testing the methods on a wide range of human joints, a standard protocol for evaluating joint pathologies based on radial and curvilinear penetration depths can be set up.

Finally, in order to reduce the sensitivity of hip simulations to hip center of rotation, a standard method for estimating HJC should be found and used in all the future research. For this reason the tissue contact penetration depths during the real hip movement (non-idealized) can be calculated. The results should be then compared with the contact penetration depths calculated during idealized hip movement about different estimated HJC. A HJC estimation method, which returns the smallest difference in penetration depths, can be recommended as a standard method.

Appendices

Appendix A: finding the cylindrical bounding volume of a triangle

The aim is to find the minimum and maximum amount of r and z of the volume bounding a given triangle. One straightforward way is to find the minimum and maximum of r and z component of all the triangle's vertices and consider them as the bounding limits of the triangle. This simple method usually gives a correct answer. But, in some special cases, it can also give a larger $r_{minimum}$ than the real one. This misestimating can show itself clearer, when the distance between the triangle and the axis of rotation is not much larger than the length of the triangle's edge (see Figure 75).

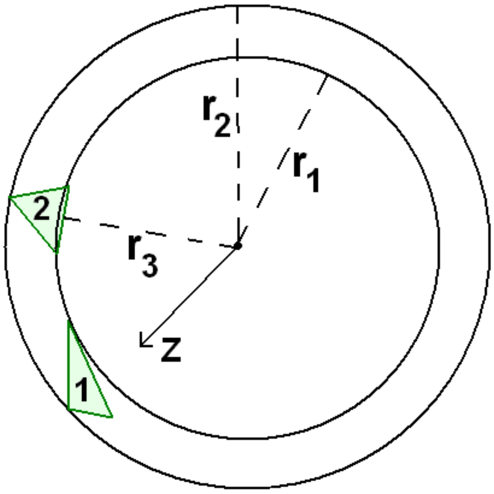

Figure 75: Finding cylindrical bounding volume of triangles. By using the approximated method, triangles 1 and 2 are bounded between r_1 and r_2. By using the exact method, triangle 2 is bounded between r_3 and r_2, where $r_3<r_1$.

If we need the exact bounding volume, instead of using the approximated method based on vertices, the shortest distance from axis of rotation to the triangle should be found and considered as $r_{minimum}$. The other values ($r_{maximum}$, $z_{minimum}$ and $z_{maximum}$) can be exactly found same as before, based on minimum and maximum of the vertices' components. For finding the shortest distance between the axis of rotation and the triangle, we only need to consider the two dimensional representations of the triangle and the axis in the x-y plane. Therefore, the distance between (0, 0) and the projection of the triangle on xy plan should be found. There are different methods to compute distance between a point and a triangle. One can either use the algorithms based on direct computation of the distance between lines and points, the interior-to-edge search method or edge-to-interior search method (Schneider and Eberly 2003).

Appendix B: checking a vertex inside 2D representation of a triangle

We want to know whether a vertex can collide with a triangle during any rotation about a known axis or not. Therefore, we should check if the circular ring passing via the vertex, having the center on the rotational axis, and perpendicular to the rotational axis, is intersecting the triangle or not (see Figure 76). For this reason one can solve the geometrical equations to see whether the circle can intersect a triangle or not.

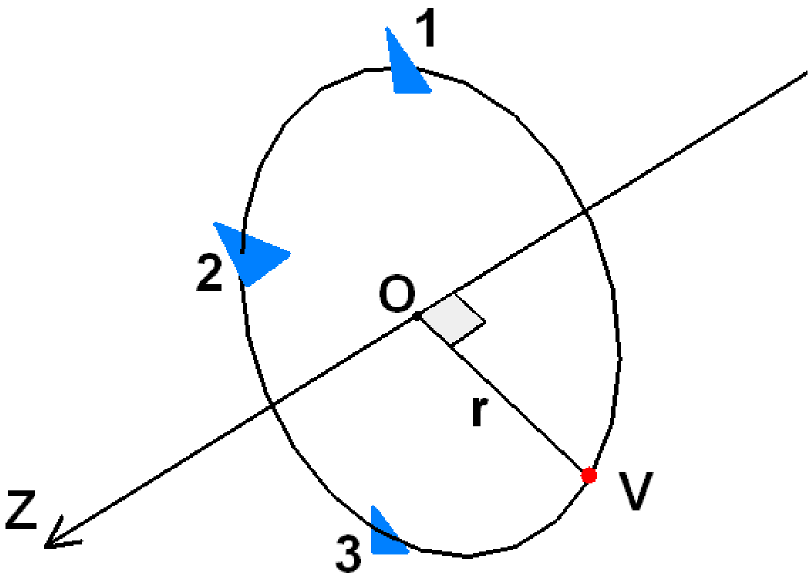

Figure 76: Checking whether a vertex can collide with a triangle. Vector V is rotating about the Z-axis. It can collide with triangle number 2, but not 1 or 3.

A simpler way is to use 2D representation of the triangle and the vertex. We know that during the rotation, only θ is changing, while z and r are remaining unchanged. Thus, we just need to check whether the 2D representation of the vertex (based on its z and r) is inside the 2D representation of the triangle (based on its z's and r's). The 2D representation of a triangle is not necessarily a normal triangle; that means the edge of the triangle, in such 2D representation, may not be a line but a curve. For simplicity, one can approximate this 2D representation of the triangle as a normal triangle. This approximation does not induce much error, especially when the sizes of triangles are smaller compared to the distance between the triangles and the Z-axis. The error may happen when the vertex can collide with a triangle in an area very close to the edge of the triangle. In this case instead of the correct triangle, the adjacent triangle may be detected as a colliding candidate. This kind of error may not affect the collision results significantly, in normal striking cases. But it can be problematic in case of sliding (which is not the objective case of this method). By such approximation, we can simply find out whether the vertex can collide with a triangle or not. For testing if a 2D point is inside a triangle (in 2D), we can use different methods based on sum of angles, cross product between the edges, and Barycentric technique (Point in triangle test Online), (Schneider and Eberly 2003). The latest one is faster in term of computation.

If we do not want to use such approximation and would like to have an exact computation, we can first find the angular projection of the vertex on the plane including the triangle (see Appendix C). The result can be either zero, one or two points. Then, it is checked whether the projected point(s) is inside the triangle or not. If the answer is positive then the 2D representation of the vertex is inside the 2D representation of the triangle, without any approximation.

Appendix C: calculating the angular distance between a vertex and a polygon

Angular distance between a vertex and polygon is the amount of angle needed to rotate the vertex until crossing the polygon’s plane. The point on the polygon’s plane, which results from crossing the polygon’s plane by the vertex during the rotation, is the rotational projection of the vertex on the polygon’s plane. The coordinates of the vertex in the cylindrical coordinate system are based on θ, r and z, and the z is previously aligned with the rotational axis; thus the rotational projection of the vertex has the same r and z of the vertex. The difference between θ of the vertex and θ of the vertex’s rotational projection is the angular distance between the vertex and the polygon. By having three points of the polygon we can have the equation for its plane:

$$x+Ay+Bz+C = 0 \, . \qquad \text{Eq. 24}$$

If we represent the Cartesian coordinates of the vertex by **V**: $[V_x, V_y, V_z]$, and the Cartesian coordinates of the vertex’s rotational projection on the polygon’s plane by **V'**: $[V'_x, V'_y, V'_z]$ then:

$$V_z=V'_z \, , \qquad \text{Eq. 25}$$

$$r = r' \text{ (in cylindrical coordinate system)}; \; r^2 = V_x^2+V_y^2 = V'^2_x+V'^2_y = r'^2 \, . \qquad \text{Eq. 26}$$

We want to find V' by knowing V and the plane of the polygon. Since V' is laying on the polygon’s plane we have:

$$V'_x + A\, V'_y + B\, V'_z + C = 0\ . \qquad \text{Eq. 27}$$

By applying Eq. 25 and Eq. 27 in Eq. 26:

$$(-A\, V'_y - B\, V_z - C)^2 + {V'_y}^2 = r^2\ . \qquad \text{Eq. 28}$$

Eq. 28 has two answers:

$$V'_y = \frac{-A(BV_z+C) \mp \sqrt{r^2\left(1+A^2\right) - (BV_z+C)^2}}{1+A^2} \qquad \text{Eq. 29}$$

$$V'_x = (-A\, V'_y - B\, V_z - C)\ .$$

If V'_y is not a real value, it means the vertex cannot intersect the polygon's plane during any rotation about the Z-axis. Having two real answers for V'_y means the vertex can cross the polygon's plane in two points. Between these two answers we only consider the answer which is inside the polygon. In a rare case, both of the answer can be inside the polygon, where both of them are correct and should be considered independently. This case can happen when the vertex is almost sliding on the polygon.

Now by having V'_y and V'_x, and using Eq. 1 we can calculate θ of the vertex's rotational projection. If we call the θ of the vertex θ_v, and θ of the vertex's rotational projection $\theta_{v'}$, then the minimum between the absolute values of [$(\theta_{v'} - \theta_v) \bmod 2\pi$] and [$(\theta_{v'} - \theta_v) \bmod 2\pi$] is the angular distance between the vertex and polygon.

Appendix D: checking whether a vertex is penetrating an object in angular direction

By using the polygon's normal vector (pointing out of the mesh), it is checked whether the mobile vertex is penetrating the fixed object or not. We first calculate the component of the polygon's normal vector which is tangential to the rotational trajectory. In Figure 77 the vertex and the vertex's rotational projection on the gray triangle are shown by **V** and **V'** respectively. The normal vector of the polygon (gray triangle) and the vector connecting the vertex's rotational projection on the polygon to the vertex (**V'** to **V**) are shown by the blue vector of **N** and the green vector of **A**, respectively. The goal is to calculate the vector which is tangential to the rotational trajectory (shown by the red vector of **T** in Figure 77). We can find the vector tangential to the rotational trajectory at the vertex's rotational projection (**V'**) by calculating the cross product between axis of rotation (Z-axis here), and any arbitrary vector connecting the axis of rotation to **V'** (shown by black vector of **B** in Figure 77):

$\mathbf{T'} = \mathbf{B} \times \mathbf{Z}$ Eq. 30

$\mathbf{T} = \alpha \mathbf{T'}$

$\alpha \in \{-1, 1\}$

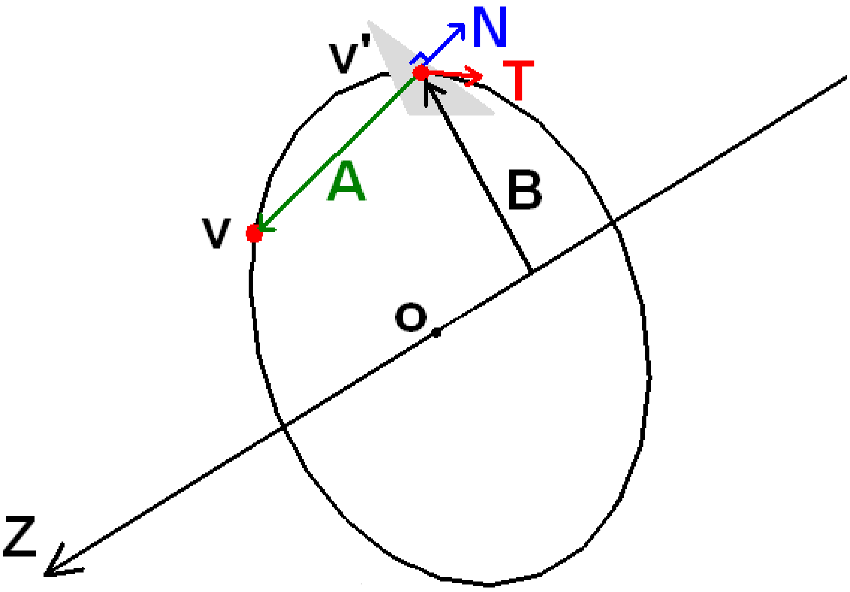

Figure 77: Different vectors used for checking the penetration in angular direction

To choose the correct value for α, we need to know whether **T'** is pointing out of the object or not. For this reason we use the polygon's normal. If the scalar product between **T'** and **N** is negative α is -1, otherwise it is +1:

if (**T'** . **N** < 0) (i.e. angle between **T** and **N** > π/2) *then* **T** = -**T'** Eq. 31

else **T** = **T'**

Now by having **T** we can find out if the vertex is penetrating the object or not. Since the polygon is chosen in the way to have the smallest angular distance to the vertex compared to the other polygons, the vertex is inside the object if the angle between **T** and **A** is larger than π/2 (see Figure 16). Therefore, their scalar product must be smaller than zero when the mobile vertex is penetrating the fixed object:

if (**T** . **A** < 0) (i.e. angle between **T** and **A** > π/2) *then* Eq. 32

V is penetrating the fixed object.

Two extreme cases (practically rare) may happen during determining the penetration (i.e. **T**.**N**=0 & **T**.**A**=0). The first extreme case happens when the fixed object surface polygon is tangent to the circular trajectory of the mobile vertex (i.e. **T**.**N**=0). In this case, since the polygon is chosen as the closest polygon of the fixed object surface to the mobile vertex, the mobile vertex is outside the fixed object if **N.A**>0 (i.e. **N** is pointing to the rotational axis), inside the fixed object if **N.A**<0, and on the fixed polygon if **N**.**A**=0. The second extreme case happens when the tangent vector **T** of the fixed object surface polygon is perpendicular to **A** (i.e. **T**.**A**=0)**.** This very rare situation means that the angular arc between the mobile vertex and its closest fixed polygon is either 0 or π. If |**A**|=0, then the arc sector angle is 0, which means the vertex is exactly on the fixed polygon. If |**A**|>0, the arc sector angle is π. Since the polygon is the closest polygon to the mobile vertex and the maximum arc between two points on a circle cannot exceed π, the mobile vertex is either always inside the fixed object or always out of it, for any amount of rotation. Therefore, the mobile vertex keeps the same default state it had before being rotated (i.e. it is considered non-penetrated if it is not also penetrated before applying the rotation and vice versa). Naturally the objects are not colliding with each other in their default state; thus we can simply consider this rare case as a non penetrating case.

Appendix E: mapping function 3 - quasi uniform radial segmentation

Based on what was explained in section 3.3.3.1.3 (see also Figure 23), we have:

(a_1, a_2, a_3, R) =FunctionType3(x, y, z) Eq. 33

$R = \sqrt{x^2 + y^2 + z^2},$

if $(y\ \ 0)$ *then* $\theta_0 = \text{ArcTan}(x/y)$;

if (if $y<0$ & $x\ \ 0$) *then* $\theta_0 = \text{ArcTan}(x/y)+\pi$;

if $(y<0$ & $x<0)$ *then* $\theta_0 = \text{ArcTan}(x/y)-\pi$;

if $(z\ \ 0)$ *then* $\theta_1 = \text{ArcTan}(y/z)$;

if $(z<0$ & $y\ \ 0)$ *then* $\theta_1 = \text{ArcTan}(y/z)+\pi$;

if $(z<0$ & $y<0)$ *then* $\theta_1 = \text{ArcTan}(y/z)-\pi$;

if $(x\ \ 0)$ *then* $\theta_2 = \text{ArcTan}(z/x)$;

if $(x<0$ & $z\ \ 0)$ *then* $\theta_2 = \text{ArcTan}(z/x)+\pi$;

if $(x<0$ & $z<0)$ *then* $\theta_2 = \text{ArcTan}(z/x)-\pi$;

if $((\pi/4\ \ \theta_0<3\pi/4)$ & $(-\pi/4\ \ \theta_2<\pi/4))$ *then* $a_1 = \theta_0 - \pi/4$, $a_2 = \theta_2 + \pi/4$, $a_3 = 0$;

if $((-\pi/4\ \ \theta_0<\pi/4)$ & $(\pi/4\ \ \theta_1<3\pi/4))$ *then* $a_1 = \theta_0 + \pi/4$, $a_2 = \theta_1 - \pi/4$, $a_3 = 1$;

if $((-3\pi/4\ \ \theta_0<-\pi/4)$ & $(3\pi/4\ \ \theta_2<\pi))$ *then* $a_1 = \theta_0 + 3\pi/4$, $a_2 = \theta_2 - 3\pi/4$, $a_3 = 2$;

if $((-3\pi/4\ \ \theta_0<-\pi/4)$ & $(-\pi\ \ \theta_2<-3\pi/4))$ *then* $a_1 = \theta_0 + 3\pi/4$, $a_2 = \theta_2 + 5\pi/4$, $a_3 = 2$;

if $((-3\pi/4\ \ \theta_1<-\pi/4)$ & $(3\pi/4\ \ \theta_0<\pi))$ *then* $a_1 = \theta_0 - 3\pi/4$, $a_2 = \theta_1 + 3\pi/4$, $a_3 = 3$;

if $((-3\pi/4 \;\; \theta_1<-\pi/4)$ & $(-\pi \;\; \theta_0<-3\pi/4))$ *then* $a_1 = \theta_0 + 5\pi/4$, $a_2 = \theta_1 + 3\pi/4$, $a_3 = 3$;

if $((-\pi/4 \;\; \theta_1<\pi/4)$ & $(\pi/4 \;\; \theta_2<3\pi/4))$ *then* $a_1 = \theta_1 + \pi/4$, $a_2 = \theta_2 - \pi/4$, $a_3 = 4$;

if $((-3\pi/4 \;\; \theta_2<-\pi/4)$ & $(3\pi/4 \;\; \theta_1<\pi))$ *then* $a_1 = \theta_1 - 3\pi/4$, $a_2 = \theta_2 + 3\pi/4$, $a_3 = 5$;

if $((-3\pi/4 \;\; \theta_2<-\pi/4)$ & $(-\pi \;\; \theta_1<-3\pi/4))$ *then* $a_1 = \theta_1 + 5\pi/4$, $a_2 = \theta_2 + 3\pi/4$, $a_3 = 5$;

$a_1 \in [0, \pi/2)$; $a_2 \in [0, \pi/2)$; $a_3 \in \{0, 1, 2, 3, 4, 5\}$.

Appendix F: checking whether a vertex is penetrating an object in radial direction

We used a method based on Barycentric technique (Schneider and Eberly 2003), for the case the polygons are triangles. The method can be extended to any kind of polygons too, by considering each polygon as a combination of two or more triangles. First of all we find out whether a polygon's plane is intersected (if the fixed object is inside the mobile object) or not intersected (if the mobile object is sliding inside the fixed object) by the line connecting the origin to the mobile vertex. If the answer is negative, the vertex cannot be penetrating the polygon, and therefore there is no need to do additional tests.

If the vertex, the origin (center of rotation/sliding), a triangle's vertex and the triangle's normal are represented by $\mathbf{V}$:$[V_x, V_y, V_z]$, $\mathbf{O}$:$[O_x, O_y, O_z]$, $\mathbf{A}$:$[A_x, A_y, A_z]$ and $\mathbf{N}$:$[N_x, N_y, N_z]$, respectively, then the distance between the origin and the triangle's plane in the direction of the vector connecting the origin to the vertex, can be found by (see Figure 78):

$$\mathbf{V}' = (\mathbf{V} - \mathbf{O}) / | \mathbf{V} - \mathbf{O} | \qquad \text{Eq. 34}$$

$$\text{Dist} = (\; (\mathbf{A} - \mathbf{O}) \,.\, \mathbf{N} \;) / (\mathbf{V}' \,.\, \mathbf{N})$$

If "Dist" is not larger than distance between the vertex and the center (in case the fixed object is inside the mobile object) or smaller than distance between the vertex and the center (in case the mobile object is sliding inside the fixed object), then the vertex may be penetrated and we should do one more test for confirmation. In fact, we need to know whether the vertex is in the radial direction of the polygon or not. For this reason, we should check if the ray connecting the center to the vertex can intersect the polygon or not. If $\mathbf{A}$:$[A_x, A_y, A_z]$, $\mathbf{B}$:$[B_x, B_y, B_z]$ and $\mathbf{C}$:$[C_x, C_y, C_z]$ are three vertices of the triangle, by using Barycentric technique (Schneider and Eberly 2003) we have:

$$b_0 = B_x - A_x \qquad \text{Eq. 35}$$

$$b_1 = B_y - A_y$$

$$c_0 = C_x - A_x$$

$$c_1 = C_y - A_y$$

$$p_0 = O_x + \text{Dist} \times V'_x - A_x$$

$p_1 = O_y + Dist \times V'_y - A_y$

$L = c_1 \times b_0 - c_0 \times b_1$

$u = (p_1 \times c_0 - p_0 \times c_1) / (-L)$

$v = (p_1 \times b_0 - p_0 \times b_1) / L$.

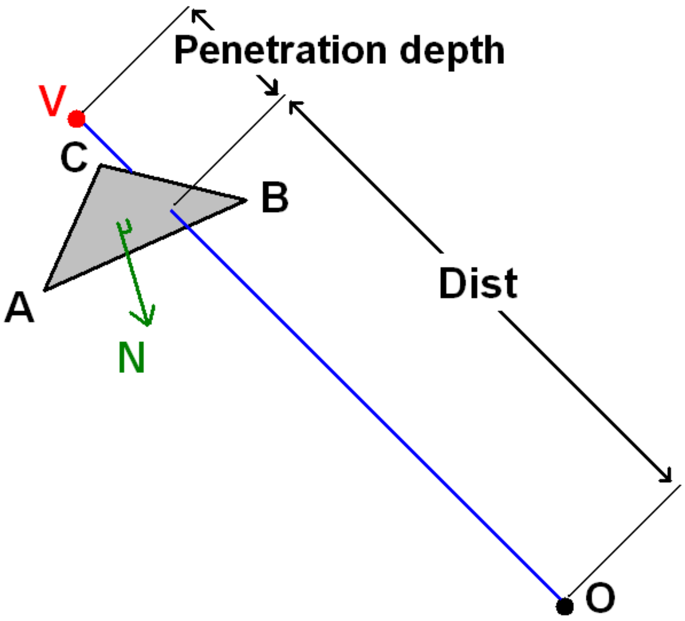

Figure 78: Checking whether a polygon's plane (gray triangle) is intersected by the line connecting the origin (**O**) to the mobile vertex (**V**). **A**, **B** and **C** are the triangle's vertices and **N** is the triangle's normal. "Dist" is the distance between the origin and the triangle's plane in the direction of the vector connecting the origin (**O**) to the vertex (**V**).

If (u, v 0) and (u+v 1) then the vertex is in the radial direction of the triangle; therefore it is penetrating the triangle. The penetration depth is the unsigned difference between "Dist" and "the distance between the vertex and the center":

$$\text{Penetration depth} = | \text{Dist} - |\mathbf{V} - \mathbf{O}| | . \qquad \text{Eq. 36}$$

Bibliography

Akalan, N E, M Ozkan, and Y Temelli. "Three-dimensional knee model: Constrained by isometric ligament bundles and experimentally obtained tibio-femoral contacts." *Journal of Biomechanics* 41 (2008): 890-896.

Arbabi, E, J Schmid, R Boulic, D Thalmann, and N Magnenat-Thalmann. "Assessing the Sensitivity of Hip Tissues Contact Evaluation to the Methods Used for Estimating the Hip Joint Center of Rotation." *Technical Report,* 2009.

Arbabi, E, R Boulic, and D Thalmann. "A Fast Method for Finding Range of Motion in the Human Joints." *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* Lyon, France, 2007. 5079-5082.

Arbabi, E, R Boulic, and D Thalmann. "Fast Collision Detection Methods for Joint Surfaces." *Journal of Biomechanics* 42, no. 2 (January 2009): 91-99.

Arbabi, E, S Chegini, R Boulic, M Tannast, S J Ferguson, and D Thalmann. "The Penetration Depth Method – A Novel Real Time Strategy for Evaluating Femoro-acetabular Impingement." *Technical Report*, 2009.

Arbabi, E, S Chegini, R Boulic, S J Ferguson, and D Thalmann. "Estimating Hip Joint Contact Pressure from Geometric Features." *Workshop on 3D Physiological Human.* Zermatt, Switzerland, 2008.

Armand, M, et al. "Computer-Aided Orthopedic Surgery with Near-Real-Time Biomechanical Feedback." *Johns Hopkins Apl Technical Digest* 25 (2004): 242-252.

Baciu, G, W S-K Wong, and H Sun. "Recode: An Image–Based Collision Detection Algorithm." *Journal of Visualization and Computer Animation* 10 (1999): 181-192.

Bandi, S., and D. Thalmann. "An adaptive spatial subdivision of the object space for fast collision detection of animating rigid bodies." *Eurographics.* 1995. 259–270.

Baraff, D, and D Witkin. "Large steps in cloth simulation." *SIGGRAPH, ACM.* 1998. 43-54.

Bell, A, D Petersen, and R Brand. "A comparison of the accuracy of several hip center location prediction methods." *Journal of Biomechanics* 23 (1990): 617-621.

Bergmann, G, et al. "Hip contact forces and gait patterns from routine activities." *Journal of Biomechanics* 34 (2001): 859-871.

Bizzini, M, H P Notzli, and N A Maffiuletti. "Femoroacetabular Impingement in Professional Ice Hockey Players: A Case Series of 5 Athletes After Open Surgical Decompression of the Hip." *American Journal of Sports Medicine* 35, no. 11 (2007): 1955-1959.

Boudriot, U, J Hilgert, and F Hinrichs. "Determination of the rotational center of the hip." *Archives of Orthopaedic and Trauma Surgery* 126 (2006): 417-420.

Bush, T R, and P E Gutowski. "An approach for hip joint center calculation for use in seated postures." *Journal of Biomechanics* 36, no. 11 (2003): 1739-1743.

Camomilla, V, A Cereatti, G Vannozzi, and A Cappozzo. "An optimized protocol for hip joint centre determination using the functional method." *Journal of Biomechanics* 39 (2006): 1096-1106.

Cappozzo, A. "Gait Analysis methodology." *Human Movemenent Science* 3 (1984): 27-54.

Chan, Y H. "Correlational analysis." *Biostatistics 104, Singapore Med* 44 (2003): 614–619.

Chang, L, and N Pollard. "Constrained least-squares optimization for robust estimation of center of rotation." *Journal of Biomechanics* 40, no. 6 (2007): 1392-1400.

Chegini, S, M Beck, and S J Ferguson. "Femoro Acetabular Impingement as a Possible Initiator of Cartilage Degeneration." *7th International Symposium On Computer Methods In Biomechanics And Biomedical Engineering.* Antibes, France, 2006. 705-710.

Chegini, S, M Beck, and S J Ferguson. "The Effects of Impingement and Dysplasia on Stress Distributions in the Hip Joint during Sitting and Walking: A Finite Element Analysis." *Journal of Orthopaedic Research* 27, no. 2 (2008): 195-201.

Eckstein, F, et al. "Non-invasive determination of cartilage thickness throughout joint surfaces using magnetic resonance imaging." *Journal of Biomechanics* 30, no. 3 (1997): 285-289.

Ferguson, S J, J T Bryant, and K Ito. "The material properties of the bovine acetabular labrum." *Journal of Orthopaedic Research* 19 (2001): 887-896.

France, L, et al. "A Layered Model of a Virtual Human Intestine for Surgery Simulation." *Medical Image Analysis* 9, no. 2 (2005): 123-132.

Funfzig, C, T Ullrich, and D W Fellner. "Hierarchical Spherical Distance Fields for Collision Detection." *IEEE Computer Graphics and Applications* 26, no. 1 (2006): 64-74.

Ganz, R, J Parvizi, M Beck, M Leunig, H Nötzli, and K A Siebenrock. "Femoroacetabular impingement: a cause for osteoarthritis of the hip." *Clinical Orthopaedics and Related Research* 407 (2003): 112-120.

Genda, E., N. Konishi, Y. Hasegawa, and T. Miura. "A computer simulation study of normal and abnormal hip joint contact pressure." *Arch. Orthopaedic and Trauma Surgery* 114 (1995): 202-206.

Genoud, P, H Sadri, C Dora, L Bidaut, R Ganz, and P Hoffmeyer. "The hip joint range of motion: a cadaveric study." *Conference of the European Society of Biomechanics.* 2000.

Gilles, B. *Anatomical and kinematical modelling of the musculoskeletal system from MRI.* PhD Thesis, Geneva, Switzerland: University of Geneva, 2007.

Gilles, B, L Moccozet, and N Magnenat-Thalmann. "Anatomical Modelling of the Musculoskeletal System from MRI." *International conference on medical image computing and computer assisted intervention (MICCAI).* Springer, 2006. 289-296.

Gottschalk, S, M Lin, and D Manocha. "OBB-Tree: A hierarchical structure for rapid interference detection." *SIGGRAPH 96 Conference Proceedings.* 1996. 171–180.

Günther, K P, F Thielemann, A Hartmann, and P Bernstein. "Combined hip-dysplasia and femoroacetabular impingement: Diagnosis and simultaneous surgical treatment." *Orthopäde* 37 (2008): 577-586.

Heidelberger, B, M Teschner, and M Gross. "Real-Time Volumetric Intersections of Deforming Objects." *Vision, Modeling, Visualization.* Munich, Germany, 2003. 461–468.

Hipp, J A, N Sugano, M B Millis, and S B Murphy. "Planning aetabular redirection osteotomies based on joint contact pressures." *Clinical Orthopaedics and Related Research* 364 (1999): 134-143.

Hodge, W A, R S Fijan, K L Carlson, R G Burgess, W H Harris, and R W Mann. "Contact pressures in the human hip joint measured in vivo." *Proceedings of the National Academy of Sciences USA.* 1986. 2879-2883.

Hudson, T, M Lin, J Cohen, S Gottschalk, and D Manocha. "V-Collide: Accelerated Collision Detection for VRML." *2 nd Annual Symposium on the Virtual Reality Modeling Language.* 1997. 119-125.

Ito, K, M Leunig, and R Ganz. "Histopathologic features of the acetabular labrum in femoroacetabular impingement." *Clinical Orthopaedics and Related Research* , 2004: 262-271.

Kang, M. "Hip joint center location by fitting conchoid shape to the acetabular rim region of MR images." *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* San Francisco, USA, 2004. 4477-4480.

Kang, M, H Sadri, L Moccozet, and N Magnenat-Thalmann. "Hip Joint Modeling for the Control of the Joint Center and the Range of Motions." *5th IFAC symposium on modeling and control in biomedical systems.* Elsevier Science, 2003. 20-24.

Kirkwood, R, E Culham, and P Costigan. "Radiographic and non-invasive determination of the hip joint center location: effect on hip joint moments." *Clinical Biomechanics* 14 (1999): 227-235.

Klein, j, and G Zachmann. "Adb-Trees: Controlling the Error of Time-Critical Collision Detection." *8th International Fall Workshop Vision, Modeling, and Visualization.* 2003. 19-21.

Krishnan, S, A Pattekar, M C Lin, and D Manocha. "Spherical Shell: A Higher Order Bounding Volume for Fast Proximity Queries." *3rd workshop on the algorithmic foundations of robotics on Robotics: the algorithmic perspective.* Houston, Texas, United States, 1998. 177-190.

Kubiak-Langer, M, M Tannast, S B Murphy, K A Siebenrock, and F Langlotz. "Range of motion in anterior femoroacetabular impingement." *Clinical Orthopaedics and Related Research* 458 (2007): 117-124.

Langlotz, F, R Bachler, U Berlemann, L P Nolte, and R Ganz. "Computer assistance for pelvic osteotomies." *Clinical Orthopaedics and Related Research* 354 (1998): 92-102.

Larsson, T, and T Akenine-Möller. "Collision Detection for Continuously Deforming Bodies." *Eurographics.* 2001. 325–333.

Lavigne, M, J Parvizi, M Beck, K A Siebenrock, R Ganz, and M Leunig. "Anterior femoroacetabular impingement: part I. Techniques of joint preserving surgery." *Clinical Orthopaedics and Related Research* 418 (2004): 61-66.

Lenaerts, G, et al. "Subject-specific hip geometry affects predicted hip joint contact forces during gait." *Journal of Biomechanics* 41 (2008): 1243-1252.

Lin, M C, and J F Canny. "Efficient Collision Detection for Animation." *3rd Eurographics Workshop on Animation and Simulation.* 1992.

Maciel, A, R Boulic, and D Thalmann. "Efficient Collision Detection within Deforming Spherical Sliding Contact." *IEEE Transaction on Visualization and Computer Graphics* 13, no. 3 (2007): 518-529.

—. "Towards a parameterization method for virtual soft tissues based on properties of biological tissue." *5th IFAC Symp. on modeling and control in biomedical systems.* Elsevier Science, 2003. 235-240.

Macirowski, T, S Tepic, and R W Mann. "Cartilage stresses in the human hip joint." *J Biomech Eng* 116 (1994): 10–18.

Malhi, A, and R X Gao. "PCA-Based Feature Selection Scheme for Machine Defect Classification." *IEEE Transaction on Instrumentation and Measurment* 53, no. 6 (2004): 1517-1525.

Mardones, R M, C Gonzalez, Q Chen, M Zobitz, K R Kaufman, and R T Trousdale. "Surgical treatment of femoroacetabular impingement: evaluation of the effect of the size of the resection." *Journal of Bone & Joint Surgery* 87-A (2005): 273-279.

Martin, H D. "Clinical Examination of the Hip." *Operative Techniques in Orthopaedics* 15 (2005): 177-181.

Melax, S. "Dynamic plane shifting bsp traversal." *Proceedings of Graphics Interface.* 2000. 213-220.

Michaeli, D A, S B Murphy, and J A Hipp. "Comparison of predicted and measured contact pressures in normal and dysplastic hips." *Medical Engineering & Physics* 19, no. 2 (1997): 180-186.

Miller, E H, H J Schneider, J L Bronson, and D McLain. "A new consideration in athletic injuries. The classical ballet dancer." *Clinical Orthopaedics and Related Research*, 1975: 181-191.

Moglo, K E, and A Shirazi-Adl. "On the coupling between anterior and posterior cruciate ligaments, and knee joint response under anterior femoral drawer in flexion: a finite element study." *Clinical Biomechanics* 18 (2003): 751-759.

Moore, M, and J Wilhelms. "Collision Detection and Response for Computer Animation." *Computer Graphics* 22, no. 4 (1998): 289 - 298.

Murphy, S, M Tannast, Y J Kim, R Buly, and M B Millis. "Debridement of the Adult Hip for Femoroacetabular Impingement Indications and Preliminary Clinical Results." *Clinical Orthopaedics and Related Research,* no. 429 (2004): 178–181.

Nishio, H, M Altef-Ul-Amin, K Kurokawa, and S Kanaya. "Spherical SOM and Arrangement of Neurons Using Helix on Sphere." *IPSJ Digital Courier* 2 (2006): 133-137.

Nötzli, H P, T F Wyss, C H Stoecklin, M R Schmid, K Treiber, and J Hodler. "The contour of the femoral head-neck junction as a predictor for the risk of anterior impingement." *The Journal of Bone and Joint Surgery* 84 (2002): 556-560.

Palmer, I. J., and R. L. Grimsdale. "Collision detection for animation using spheretrees." *Computer Graphics Forum* 14, no. 2 (June 1995): 105-116.

Piazza, S, N Okita, and P Cavanagh. "Accuracy of the functional method of hip joint center location: effects of limited motion and varied implementation." *Journal of Biomechanics* 34, no. 7 (2001): 967–973 .

"Point in triangle test." *Blackpawn Tex.* Online. http://www.blackpawn.com/ (accessed 2007).

Rapperport, D. J., D. R. Carter, and D. J. Schurman. "Contact finite element stress analysis of the hip joint." *Journal of Orthopaedic Research* 3 (1985): 435-446.

"Self-organizing maps on non-euclidean spaces." In *Kohonen Maps*, by H Ritter, 97-108. 1999.

Russell, M E, K H Shivanna, N M Grosland, and D R Pedersen. "Cartilage contact pressure elevations in dysplastic hips: a chronic overload model." *Journal of Orthopaedic Surgery* 3 (2006): 1-6.

Schmid, J, and N Magnenat-Thalmann. "MRI Bone Segmentation using Deformable Models and Shape Priors." *International conference on medical image computing and computer assisted intervention (MICCAI).* Springer, 2008. 119-126.

Schneider, P J, and D H Eberly. *Geometric tools for computer graphics.* San Francisco, USA: Morgan Kaufmann, 2003.

Scifert, C. F., T. D. Brown, D. R. Pedersen, and J. J. Callaghan. "A finite element analysis of factors influencing total hip dislocation." *Clinical orthopaedics and related research* 355 (1998): 152-162.

Seidel, G K, D M Marchinda, M Dijkers, and R W Soutas-Little. "Hip joint center location from palpable bony landmarks-a cadaver study." *Journal of Biomechanics* 28, no. 8 (1995): 995-998.

Si, H. "TetGen." *A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator.* Version 1.3. 2005.

Si, H, and K Gärtner. "An Algorithm for Three-Dimensional Constrained Delaunay Tetrahedralizations." *4th International Conference on Engineering Computational Technology.* Lisbon, Portugal, 2004.

Siston, R, and S Delp. "Evaluation of a new algorithm to determine the hip joint center." *Journal of Biomechanics* 39 (2006): 125-130.

Spillmann, J, and M Teschner. "Contact Surface Computation for Coarsely Sampled Deformable Objects." *Proceedings of Vision, Modeling, and Visualization.* 2005. 289-296.

Tannast, M, D Goricki, M Beck, S B Murphy, and K A Siebenrock. "Hip damage occurs at the zone of femoroacetabular impingement." *Clinical Orthopaedics and Related Research* 466, no. 2 (2008): 273-280.

Tannast, M, K A Siebenrock, and S E Anderson. "Femoroacetabular impingement: radiographic diagnosis--what the radiologist should know." *American Journal of Roentgenology* 188, no. 6 (2007): 1540-1552.

Tannast, M, M Kubiak-Langer, F Langlotz, M Puls, S B Murphy, and K A Siebenrock. "Noninvasive three-dimensional assessment of femoroacetabular impingement." *Journal of Orthopaedic Research* 25, no. 1 (2007): 122-131.

Teran, J, E Sifakis, S S Blemker, V Ng-Thow-Hing, C Lau, and R Fedkiw. "Creating and simulating skeletal muscle from the visible human data set." *IEEE Transaction on Visualization and Computer Graphics* 11, no. 3 (2005): 317-328.

Teschner, M, B Heidelberger, M Mueller, D Pomeranets, and M Gross. "Optimized Spatial Hashing for Collision Detection of Deformable Objects." *Vision, Modeling, Visualization.* 2003. 47-54.

Teschner, M, et al. "Collision Detection for Deformable Objects." *Computer Graphics Forum* 24, no. 1 (2005): 61-81.

Theodoridis, S, and K Koutroumbas. *Pattern Recognition.* CA, USA: Academic Press, 2003.

Trenkel, S, R Weller, and G Zachmann. "A Benchmarking Suite for Static Collision Detection Algorithms." *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision.* Plzen - Bory, Czech Republic: Union Agency, 2007.

Turk, G. *Interactive collision detection for molecular graphics.* Technical Report, TR90-014, University of North Carolina at Chapel Hil, 1990.

Van Den Bergen, G. "Efficient collision detection of complex deformable models using AABB trees." *Journal of Graphics Tools* 2, no. 4 (1997): 1-14.

Volino, P, and N Magnenat-Thalmann. "Efficient Self-Collision Detection on Smoothly Discretized Surface Animations using Geometrical Shape Regularity." *Computer Graphics Forum* 13, no. 3 (1994): 155–166.

—. "Implementing fast cloth simulation with collision response." *International Conference of Computer Graphics (CGI), IEEE Computer Society.* 2000. 257-266.

Wiberg, G. "Studies on dysplastic acetabular and congenital subluxation of the hip joint: with special reference to the complication of osteo-arthritis." *Acta Chirurgica Scandinavica* 58 (1939): 7-38.

Won, Y Y, I H Chung, N S Chung, and K H Song. "orphological study on the acetabular labrum." *Yonsei Med J* 44 (2003): 855-862.

Wu, G, et al. "ISB recommendation on definitions of joint coordiante system of various joints for the reporting of human joint motion-part I: ankle, hip and spine." *Journal of Biomechanics* 35, no. 4 (2002): 543-548.

Yershova, A, and S M LaValle. "Deterministic sampling methods for spheres and SO (3)." *IEEE international conference on Robotics & Automation.* New Orleans, LA, United States, 2004. 3974-3980.

Zachmann, G. "Rapid collision detection by dynamically aligned DOP-trees." *Proceedings of IEEE Virtual Reality Annual International Symposium.* Georgia, Atlanta, USA, 1998. 90-97.

Zachmann, G, and E Langetepe. "Geometric Data Structures for Computer Graphics." *Tutorial at ACM SIGGRAPH.* 2003.

Curriculum Vitae

Name	Ehsan ARBABI
Date and place of birth	17 September 1981, Leeds - United Kingdom
Nationality	Iranian and British (dual)
Languages	Persian (Mother tongue), English (Fluent), French (Intermediate), Arabic (Intermediate)

EDUCATION

Ph.D. in Computer Science 11.2005-06.2009

Virtual Reality Lab, School of Computer & Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Thesis: Contact Modeling and Collision Detection in Human Joints.

M.Sc. in Biomedical Engineering (Bioelectric) 09.2003-10.2005

Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

Thesis: Investigation on Relation between Effective Features Used for Different Classifiers in BCI.

B.Sc. in Electrical Engineering (Electronics) 09.1999-09.2003

Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

Thesis: Investigation on p-n Junctions Breakdown in Semiconductors.

EXPERIENCES

Virtual Reality Lab, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. 11.2005-06.2009

Research and Teaching Assistant

- Collaborating with a team of researchers from different research centers in Switzerland (at The National Centre of Competence in Research), and working on a computer based medical simulator for hip joint.
- Supervising Master students' semester projects.
- Teaching and project evaluating during the practical course of Computer Graphics.

Electrical Engineering Dep., Sharif University of Technology, Tehran, Iran 03.2005-07.2005

Teaching Assistant

- Providing and evaluating student projects for the graduate course of Biosignal Processing.

Biomedical Engineering Lab., Sharif University of Technology, Tehran, Iran 07.2004-09.2004

Research Assistant

- Employing biomedical signals recorder (Electroencephalogram) for collecting brain signals and applying them in signal classification projects.

Micro Technology Lab., Sharif University of Technology, Tehran, Iran 04.2003-09.2003

Research Assistant

- Fabricating and evaluating semiconductor devices, by working on processes such as diffusion, thermal oxidation and metal coating.

PUBLICATIONS

- E. Arbabi, S. Chegini, R. Boulic, M. Tannast, S. J. Ferguson and D. Thalmann, “The Penetration Depth Method – A Novel Real Time Strategy for Evaluating Femoro-acetabular Impingement,” *Technical Report*, 2009.

- E. Arbabi, J. Schmid, R. Boulic, D. Thalmann and N. Magnenat-Thalmann, “Assessing the Sensitivity of Hip Tissues Contact Evaluation to the Methods Used for Estimating the Hip Joint Center of Rotation,” *Technical Report*, 2009.

- E. Arbabi, R. Boulic and D. Thalmann, “Fast Collision Detection Methods for Joint Surfaces,” *Journal of Biomechanics*, vol. 42, no. 2, January 2009, pp. 91-99.

- E. Arbabi, S. Chegini, R. Boulic, S. J. Ferguson and D. Thalmann, “Estimating Hip Joint Contact Pressure from Geometric Features,” *Workshop on 3D Physiological Human*, Zermatt, Switzerland, December 2008.

- E. Arbabi, R. Boulic and D. Thalmann, “A Fast Method for Finding Range of Motion in the Human Joints,” *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'07)*, Lyon, France, August 2007, pp. 5079-5082.

- E. Arbabi and M. B. Shamsollahi, “Discrimination between 'Reading Concentration' and 'Listening Concentration' in Electroencephalogram,” *International Journal of Scientific Research (IJSR)*, vol. 16, 2006.

- E. Arbabi and M. B. Shamsollahi, “Comparison between Different Types of Features Used for Classification in BCI,” *12th International Conference on Biomedical Engineering (ICBME2005)*, Singapore, December 2005.

- E. Arbabi and M. B. Shamsollahi, “SNR in WSS Jittered Sampled Signal,” *13th European Signal Processing Conference (EUSIPCO2005)*, Antalya, Turkey, September 2005.

- E. Arbabi, M. B. Shamsollahi and R. Sameni, “Comparison between Effective Features Used in Bayesian and SVM Classifiers in Brain-Computer Interface,” *27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS’05)*, Shanghai, China, September 2005.