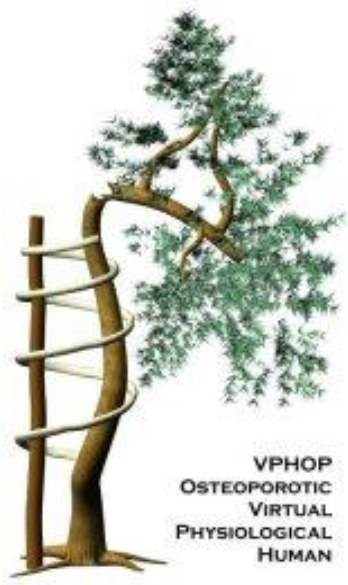

VPHOP

Technology to fight Osteoporosis

Integrated Project # FP7-223865

Deliverable

Final Morphable Musculoskeletal Model

Deliverable D10.2

Work package WP10: Morphable Musculoskeletal Model

Josef Kohout – (UWB)

Gordon J. Clapworthy – (BED)

et al.

31.7.2012

SEVENTH FRAMEWORK PROGRAMME

eHealth

DOCUMENT INFORMATION

IST Project Number	FP7-223865	**Acronym**	VPHOP
Full title	The Osteoporotic Virtual Physiological Human		
Project URL	http://www.vphop.eu		
Document URL	http://www.vphop.eu/associates/reviewers/documents/Deliverables/		
EU Project officer	Joël BACQUET		

Deliverable Number	D10.2	**Title**	Final Morphable Musculoskeletal Model
Work package Number	WP10	**Title**	Morphable Musculoskeletal Model

Date of delivery	**Contractual**	31.7.2012	**Actual**	31.7.2012
Status	Version v1e		final☒	
Nature	Prototype ☐ Report☐ Demonstrator ☐ Other ☒			
Dissemination Level	Public ☐ Consortium☒			

Authors (Partner)	J Kohout (UWB), P Kellnhofer (UWB), J Hájková (UWB), J Rus (UWB), T Janák (UWB), D Cholt (UWB), I Zelený (UWB), E Kohoutová (UWB), P Volkovinský (UWB), G Shanahan (UWB), G J Clapworthy (BED), Y Zhao (BED), Y Tao (BED), G. Gonzalez-Garcia (BED), F Dong (BED)			
Responsible Author			**Email**	besoft@kiv.zcu.cz
	Partner	UWB	**Phone**	+420 377 632 495

Abstract (for dissemination)	The final morphable musculoskeletal model for wrapping a set of muscles under leg movement was developed. It starts from an atlas and deforms this generic model according to data captured from the individual patient to form a personalised model. After this, the motion is fused with the anatomical data and this provides the environment for the muscle deformation calculations to take place. In this, the positions and shapes of the muscles during motion are calculated, with interpenetration being avoided so that muscles wrap properly around the bones and other muscles. A muscle fibre model is then accommodated within the deformed muscle.	
Keywords	Virtual Physiological Human	Osteoporosis

Version Log			
Issue Date	**Rev No.**	**Author**	**Change**
4.7.2012	v1	Josef Kohout	1st draft
4.7.2012	v1b	Matteo Balestri	First quality check
5.7.2012	v1c	Josef Kohout	Numbering of figures changed
24.7.2012	v1d	Gordon Clapworthy	Full editing
24.7.2012	v1e	Josef Kohout	Final editing. Version circulated to the Board

Project Consortium Information

LIST OF DEFINITIONS

Modelling technology: The complex of methods, protocols and ICT technology that makes it possible to convert a set of patients' information into a prediction.

Hypermodel: A "**model of models", an integrative model generated as a composition** of multiple reductionist models.

Biomed Town: A free community of professionals with interests in biomedical research, based on the town metaphor.

Clinical data baseline: A part of the clinical data collection protocol. It consists of the data pool common to all four clinical partners, providing the core database and a minimum set of clinical data for the Technology Assessment.

Special data collection: Additional data collection protocols that involve at least one technology providing partner and at least one clinical partner. They are aimed to produce data to identify, assess, or explore specific aspects of the various modelling technologies being developed by the VPHOP consortium.

Retrospective data: Clinical end point data used to assess the clinical accuracy of the modelling technology and for which the clinical outcome is known at the time of the model-based prediction. In order to reduce the bias, the modeller can be kept blind of such outcomes until the prediction is made.

Prospective data: Clinical end point data used to assess the clinical accuracy of the modelling technology and for which the clinical outcome is unknown when the model-based prediction is made.

Physiome Space: A digital library service designed to help researchers to share their biomedical data and models (www.physiomespace.com)

Mechanical Turk: A term coined by Amazon to describe a crowd-sourcing Internet marketplace that enables computer programmers (known as Requesters) to co-ordinate the use of human intelligence to perform tasks which computers are unable to do. In our context, the term indicates a station on the hypermodel workflow at which human intervention is required. This can be the simple introduction of some limited additional data, or a complete processing application that provides to each service provider a work list, access to the input data, the operations required to perform the processing, and the services by which to upload the result and restart the workflow

OpenClinica: Software installed and configured on a SCS server to upload and share data anonymously from the clinical trial.

LIST OF ABBREVIATIONS

BH	Body height
BV/TV	Bone Volume / Trabecular Volume; index of trabecular bone volume
BVF	Bone Volume Fraction
BW	Body weight
CF	Cadaver Fibres method
Conn.D	Connectivity Density
CPM	Classic Project Management
CT	Computerized Tomography
Ct.Th	Cortical Thickness
CVF	Cement Volume Fraction
DA	Degree of Anisotropy
DoW	Description of Work
DRR	Digitally reconstructed radiographs
DXA	Dual X-ray Absorptiometry
FE	Finite elements
GUI	Graphical User Interface
HTA	Health Technology Assessment
ICT	Information and Communication Technology
MAF	Multimodal Application Framework
MIL	Mean Intercept Length
PK	Petr Kellnhofer
PLX	High Performance Computing machine
ROI	Region of Interest
RF	Risk of Fracture
SMI	Structure Model Index
SoC	Standard of Care
TA	Trabecular arch
Tb.Th	Trabecular Thickness
Tb.N	Trabecular Number
Tb.Sp	Trabecular Spacing
XPM	eXtreme Project Management

IOR	Istituto Ortopedico Rizzoli
TUE	Technische Universiteit Eindhoven
EMP	Empirica Gesellschaftfür Kommunikations- und TechnologieforschungmbH
SCS	SCSsrl
CHA-JWI	Julius Wolff Institut, Charité - Universitätsmedizin Berlin
CHA-CBF	Campus Benjamin Franklin, Charité - Universitätsmedizin Berlin
ETH	Eidgenössische Technische Hochschule
UGE	University of Geneva
UBE	University of Bern
SLC	Sylvia Lawry Centre for Multiple Sclerosis Research e.V.
ANS	ANSYS France SAS
KUL	Katholieke Universiteit Leuven
BED	University of Bedfordshire
UOI	University of Iceland
UOU	University of Uppsala
INS	Institut National de la Santé et de la Recherche Médicale
BSP	Biospace Med
LBM	Laboratoire de BioMécanique, ARTS
BRL	BrainLAB AG
PMS	Philips Medical Systems
UWB	University of West Bohemia
USFD	University of Sheffield

TABLE OF CONTENTS

1. Introduction

Musculoskeletal diseases (such as osteoporosis and neuromuscular disorders) affect hundreds of millions of people around the world. In most cases, these disorders are incurable and their treatment involves changes in lifestyle supported by a rehabilitation program to improve the quality of patient life. To maximise the efficiency of the treatment, a full understanding of the physiology of muscles is required – for example, one needs to understand the role played by different motion activities (e.g., walking, climbing stairs or falling) in the overall risk of bone fracture in order to be able to propose suitable changes in lifestyle.

An estimation of skeletal loading is feasible only through a modelling technique [1]. Musculoskeletal models [2]-[7] in common clinical use assume that the mechanical action of the muscle occurs along a poly-line, namely the action line, joining the origin and insertion points of the muscle, i.e., the sites at which the muscle is attached to the bone by a tendon. In essence, an action line is a representation of muscle fibres. An advantage of action-line models, besides their rapid processing speed, is that the model created for one particular patient can be easily adjusted (usually by uniform scaling) to another patient, though, of course, if the anatomies of the patients differ significantly, this may no longer be true. On the other hand, these models tend to overestimate the predicted joint loads because their assumption of muscle fibre length being uniform over the entire muscle bundle is often not fulfilled in practice [1]. Another drawback is that representing a muscle by a set of *ad hoc* action lines provides very limited insight since this has nothing common with human anatomy.

More visually realistic approaches represent a muscle by a B-spline solid whose iso-lines correspond to muscle fibres [7] or by a 3D finite-element mesh whose cells contain information about the direction of the muscle fibres present in its volume [8]. Although good agreement was found when comparing the results with static MRI images taken in different postures, use of these models in the clinical context is highly impractical because generating the meshes is a complex process easily requiring several days for a highly skilled operator and the model cannot be easily adjusted to another patient. Further, once the model has been generated, computing the solution requires several hours on a supercomputer [8].

In our previous work, which was performed within another EC-funded project, LHDL (IST-2004-026932), we developed an alternative approach in which a muscle is represented not only in the traditional way by one or more lines of action describing its attachment to the bones, but also by a triangulated surface mesh (obtained, for example, from MRI data). This mesh is automatically wrapped around bones as they move, and the volume within it is decomposed into an arbitrary number of muscle fibres that can be used instead of action lines to produce biomechanical predictions. These have an expected accuracy somewhere between those provided by action-line methods and the more accurate, but computationally impractical finite-element methods, and they are produced sufficiently rapidly that it is feasible to consider their use in a clinical environment.

There were several major drawbacks in this previous work [9]. Experiments revealed that changes in the volume sometimes exceeded 12%. Also, each muscle was processed independently, with no account being taken of interaction with bones or other muscles, which regularly led to inter-penetration during movement. Furthermore, the mapping technique produced acceptable fibres only for muscles with small attachment areas and broader application was inhibited by the fact that it was not possible to adjust the data model of a walking human to a specific subject nor to fuse it automatically with other motion data.

The objectives of WP10 are to overcome these drawbacks and deliver a full morphable model of the lower half of the musculoskeletal system.

2. Overview of D10.2

We created a new generic musculoskeletal data model of the lower limbs in which models of the bones and muscles are properly tagged (using the ontology developed in the LHDL project) and hierarchically grouped to provide the clinicians with information of various kinds (e.g., muscle attachment areas, corresponding action lines, some muscle fibre paths, joint definitions). This generic atlas is deformed according to data captured from the individual patient to form a fully personalised dynamic model. This is performed in three stages (see Figure 1):

- *atlas scaling*: the generic atlas model is semi-automatically scaled non-uniformly (morphed) to fit the anatomy of a particular patient as defined in EOS dual images,
- *motion data fusion*: the morphed atlas model is fused with motion data defining the kinematics of the skeleton during various physical activities,
- *muscle wrapping*: for each time frame (current-pose position), the positions and shapes of the muscles are calculated; interpenetration is avoided so that muscles wrap properly around the bones and other muscles, and a muscle fibre model is accommodated within the deformed muscle.

Figure 1.Overview of our Morphable Musculoskeletal Model.

Our approach was implemented in C++ (MS Visual Studio 2010) using the Multimod Application Framework (MAF) [10] version 2.2, which is a visualisation system based mainly on VTK [11], and integrated into LHPBuilder, an application first released in LHDL and now further developed within VPHOP, which is available to the consortium via the Biomed Town portal[1], together with the WP10 Morphable Musculoskeletal Model user manual, the generic atlas created and the test data.

[1] https://www.biomedtown.org/biomed_town/vphop/consortium/wp10/repository/Tools
https://www.biomedtown.org/biomed_town/vphop/consortium/wp10/repository/Data

3. Detailed Description

3.1. Generic Musculoskeletal Atlas Model

Our generic musculoskeletal data model of the lower limbs was created from one of the LHDL cadaver data stored in several available .MSF files, some downloaded from PhysiomeSpace[2], others provided by IOR. This required

- organisation of data into a hierarchy that is consistent and can be easily processed, definition of joints and muscle wrapper VMEs (MAF data entities)[3];
- construction of regions (e.g., right thigh) and definition of joints between these regions;
- assigning tags to VMEs so that particular bones, muscles, regions, landmarks for motion fusion and joints could be identified;
- addition of other semantic information, e.g., relationships between regions and bones;
- improving the quality of the surface models of the muscles of the right leg using various operations, such as Poisson surface reconstruction, removal of non-manifold edges, hole filling, smoothing, etc., that are available in the LHPBuilder, MeshLab[4] and Blender[5] software;
- construction of lower resolution models of the muscles and bones;
- correction of the definitions of attachment areas.

Details of the above can be found in deliverables D10.1, D1.1c, D1.2c, and D1.1d.

As the LHDL cadaver data are incomplete, the atlas created is also incomplete (e.g., no muscles for shanks, missing information about attachment points).

3.2. Atlas Scaling

The generic model described above can be scaled to fit the anatomy of a particular patient as defined in a pair of orthogonal DXA or X-ray images as depicted in Figure 2; in our case, these are produced by the EOS device, which uses a very low radiation dosage. After loading the images, patient landmarks must be manually positioned on the EOS images – see Figure 3. As bones are more distinguishable than muscles in EOS images, landmarks from bones are easier to select and more plausible as inputs. Our method allows the user to position a set of landmarks which are easily located - the centroid of the femur ball, the centroid of the patella and, optionally, the centroid of the ankle.

[2] https://www.physiomespace.com/

[3] VME is a programmable data entity in MAF composed by a time-varying dataset, a time-varying matrix that defines the pose of the VME with respect to its parent in the VME Tree hierarchy, and a set of metadata that provides all the textual attributes of the VME.

[4] http://meshlab.sourceforge.net/

[5] http:// www.blender.org/

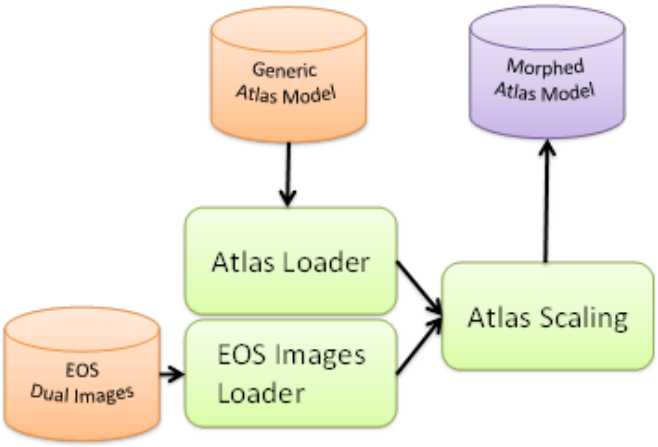

Figure 2. Workflow of Atlas Scaling.

These landmarks are used in our operation to scale the input model; this is performed automatically and non-uniformly. The operation exploits total energy minimization, preserving local surface details by the differential Laplacian shape descriptor [12] while moving the landmark points close to their target locations. Constraints of the tendon attachments and contiguity of bone or muscle are included to retain musculoskeletal structures and avoid inter-penetration. By performing the above actions, the generic musculoskeletal atlas is scaled to a patient-specific model.

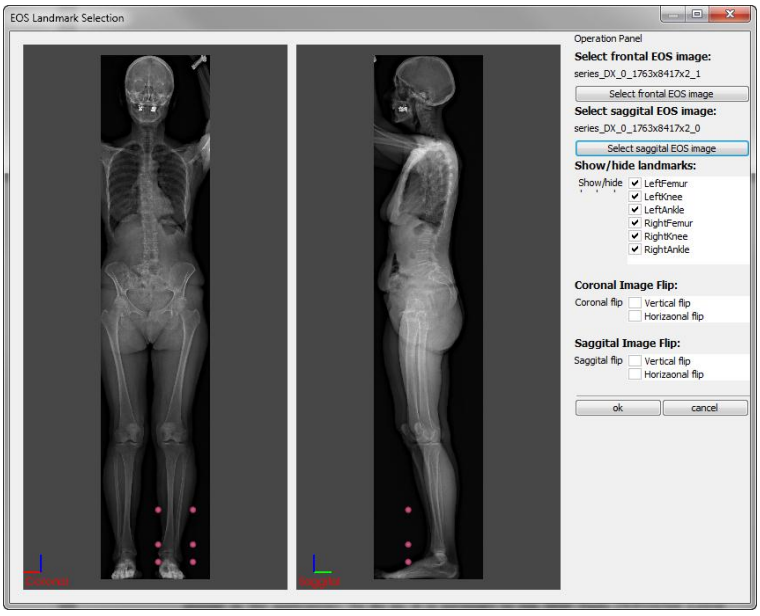

Figure 3. Landmark positioning in EOS images.

Currently, only bones, muscles and muscle attachment areas at their highest resolution are scaled, as can be seen in Figure 4, however, extending the scaling operation to other VMEs, e.g., bone groups, region surfaces, or items at lower resolutions, is straightforward. All that is required is to express the relationship between the geometry of the other VMEs and the bones (or muscles) to be scaled and then, after the scaling is completed, to exploit this relationship to reconstruct the geometry of the other VMEs. The technique for this has been already developed and it is extensively used in various parts of morphable musculoskeletal model approach.

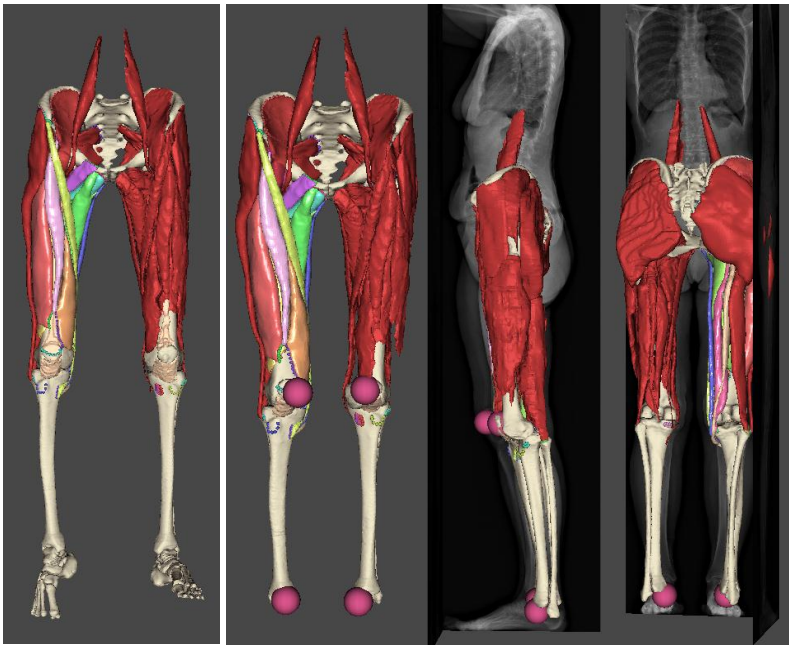

Figure 4. Bones and muscles of generic atlas model: (left) before and (right) after scaling.

3.3. Motion Fusion

Once the model is adapted, it can be fused with motion data defining the kinematics of the skeleton during various physical activities, e.g., walking, stair climbing or falling to one side. Our motion data contains time-variant landmarks acquired from tracking markers placed on the subject. The iterative closest point (ICP) algorithm is used for the registration of these time-variant landmarks and corresponding landmarks that are specified in our model. For each time frame, the ICP algorithm translates and rotates the bones according to the input motion data, which results in a change of the coordinates of landmarks for attachment areas and of the end points of the action lines, since the position of these is relative to the position of the corresponding bone.

If the anatomy of the moving subject is not similar to that represented in the static model, motion retargeting must be applied to obtain correct results. A Kálmán-like filter [13] is used to convert the input motion data to a physically plausible motion that preserves the desirable properties of the motion data in the moving object – see Figure 5. Kinematic constraints can be specified interactively, and the filter can perform motion retargeting based on these constraints, with interactive performance.

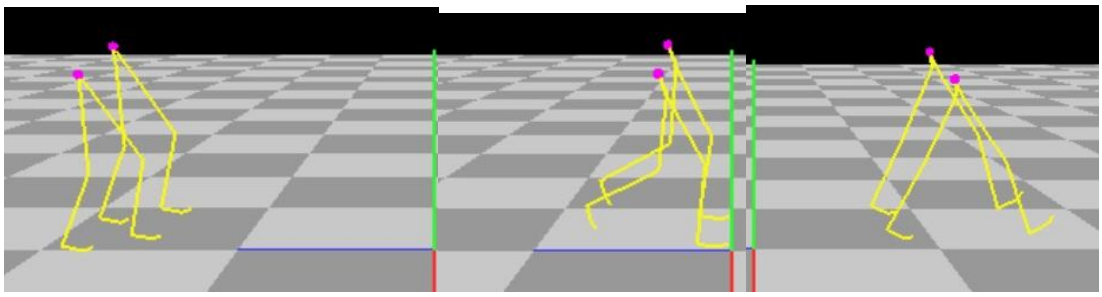

Figure 5. Example of the result of motion retargeting.

3.4. Muscle Wrapping

When the physical model is fused with the motion, it must be ensured that each muscle surface, represented by a triangulated mesh, and its associated muscle fibres, represented by a set of poly-lines, are wrapped around the bones and the other muscles during the motion. Two completely different pathways to fulfil this objective have been developed, as illustrated in Figure 6.

In the *Energy Minimization* pathway, as the skeleton moves, the shape of the muscle surface mesh changes in such a way that the muscle volume is conserved and inter-penetration with bones and other muscles is avoided so that the muscles wrap properly. The deformed muscle is then decomposed into muscle fibres as specified by the template of the fibre geometry – details are given below.

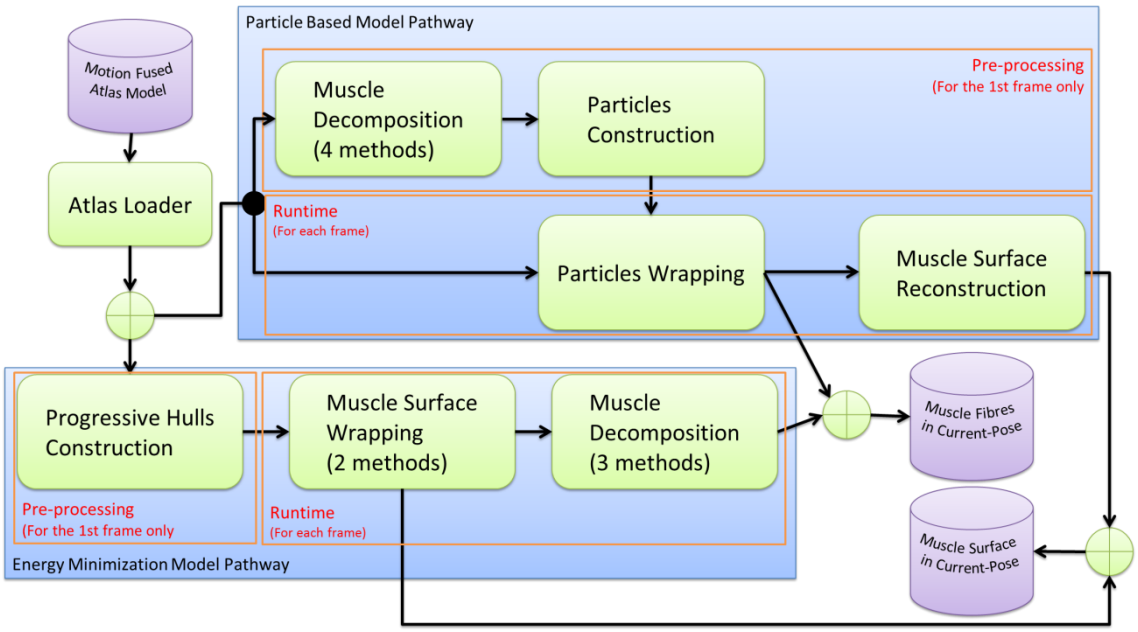

Figure 6. Muscle wrapping workflow; green spheres are used at places where more options exist (only one path is to be chosen).

The *Particle Based* pathway starts with the fibre decomposition and then transforms the set of fibres produced into a mass-spring system whose end-nodes are fixed to the bones so that when the bones move, the equilibrium of the system is violated. This triggers a recalculation of the positions of the inner nodes; again, penetration between a node and a bone surface is avoided.

3.4.1. Construction of Progressive Hulls

In order to make the processing faster and more robust, it was found that, for each muscle to be processed, a coarse outer hull should be specified containing the input mesh within its interior and preserving its overall shape. Our method [14] for constructing such a hull is based on edge-collapse decimation in which the new position of the collapsed end-points of the edge is computed to lie above planes defined by one-ring triangles of these end-points, whilst minimising the volume of the decimated mesh. The method may take a couple of minutes to process a medium-sized mesh (100K triangles), such as in the pelvis. The CPU implementation is approximately 5 times slower than the GPU implementation, but is more robust (which is caused by the natural sequential character of the method). An example of the hull can be seen in Figure 7. We note that the constructed hull is automatically stored to be reused whenever possible, which increases the overall performance of our approach.

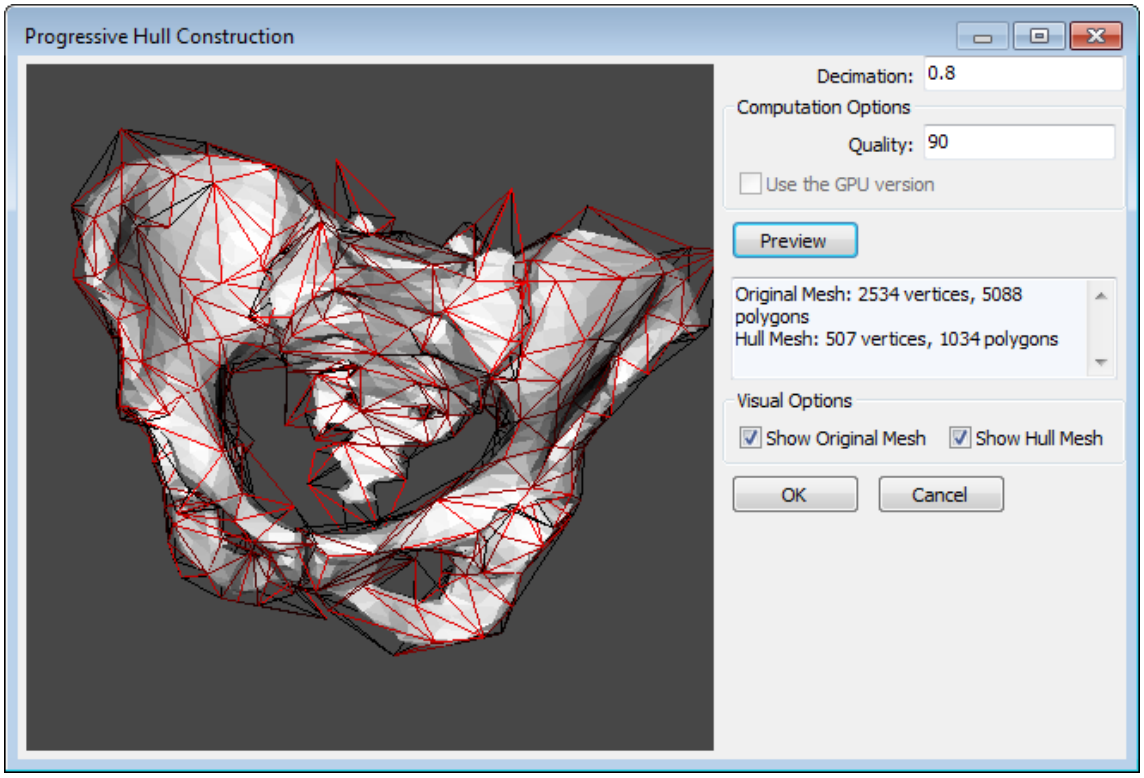

Figure 7. Pelvis and its outer hull.

3.4.2. Muscle Surface Wrapping

As noted in Section 1, each muscle is associated with one or more action lines which describe its general path between the attachment areas on the bones. It is important to point out that these are defined with great care as an inaccurate specification of action lines can have a significant impact on the accuracy of the muscle surface wrapping since, in our model, an action line serves as a muscle "skeleton" – a reference line around which the muscle is built.

To calculate the change that takes place during the motion, we consider the muscle shapes at successive time frames. At the current time step, the muscle is in the so-called current-pose position. The aim is to find its position in the next time frame subject to the various constraints discussed.

Starting from the current-pose position, the path of the action line associated with the muscle is recalculated using the obstacle set method described by Garner & Pandy [2] or the global method by Audenaert & Audenaert [6]. Once the new path is found, the difference between it and the previous path determines how the muscle surface should change during its transition to the subsequent time frame.

We have developed two methods to deform the muscle while preserving its volume and, optionally, inducing impenetrability of muscles and bones. The first, called the PK method [9], exploits total energy minimisation. In our case, the energy of each vertex of the input muscle surface mesh is derived from its position relative to vertices in its local neighbourhood and from various "soft constraints" (e.g., the position of the vertex relative to action lines) or "hard constraints" (e.g., the participation of the vertex in relation to the total volume). Any change in a constraint induces a change in the energy, and the technique tries to reposition the vertices of the original mesh so as to minimise the total deformation energy. We note that for non-manifold

meshes or meshes with boundaries or self-intersections, the computation of energy is unstable and typically produces unpredictable results during the deformation, as can be seen in Figure 8.

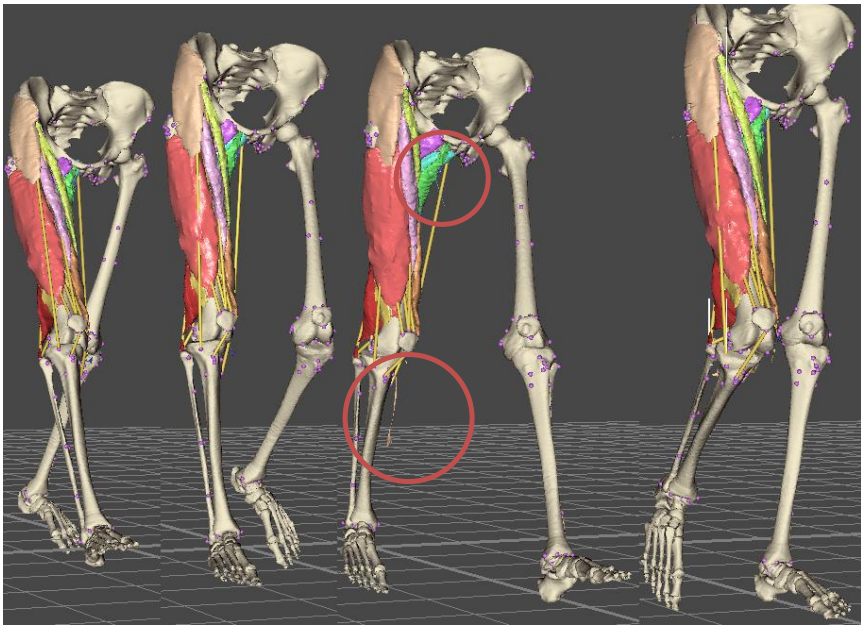

Figure 8. Non-manifold muscles wrapped using the PK method, with apparent artefacts.

The method creates a system of linear equations with a non-linear constraint that is solved by an iterative Gauss-Newton method with Lagrange multipliers. The process is not only time (N^3) and memory (N^2) consuming but also numerically unstable, so an initial approximate solution is found using the coarse outer hull mesh (see the previous section); this is then refined by a final few iterations for which the original mesh is used.

When the method runs without inducing impenetrability of muscles and bones, each muscle is processed independently of the others, leaving the responsibility of having an acceptable co-penetration between muscles to a proper definition of their action lines. Processing a muscle of medium size (10K triangles) in this mode requires, on average, about 500 ms on commodity hardware, whilst keeping the volume error below 0.04% (provided that the quality of the muscle models is good). All muscles in our data can be processed individually in under 2 seconds.

When the impenetrability of muscles and bones is required, the method starts with a rigid transformation of the whole rest-pose musculoskeletal model so that the pelvis in this new rest-pose position matches the pelvis in the current-pose. After that, it processes all muscles, simultaneously checking at each major iteration step if the meshes (of muscles and bones) intersect; at present, this is done by ray-casting, exploiting bounding volumes to speed up the tracing. During this process, a bone is represented by its surface mesh at a relatively low resolution, i.e., an outer hull is not exploited since the bone is rigid and undergoes no deformation.

If an intersection is detected, the vertices at the intersection are shifted backwards; details can be found in [15] but also ***in D10.1 and D1.1d.*** This violates the other constraints, e.g., shape or volume of the muscle, and these are generally restored only partially during the following iterations. As a result, the wrapped muscle surface is typically no longer smooth and the volume removed to prevent penetration is readily apparent, as can be seen in Figure 9.

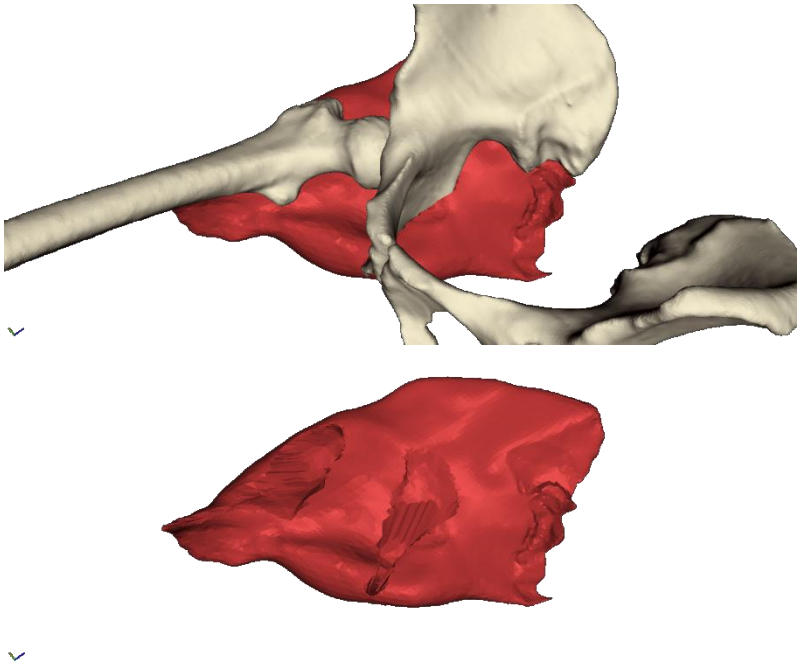

Figure 9. Gluteus Maximus muscle wrapped using PK method with impenetrability induced between it and bones. The PK method required 20.1 seconds, and a volume error of 7.45% was achieved.

Our experiments have shown that the volume error may exceed 1%; however, this is still well within the typical physiologically acceptable limit of 6% [16]. To process a single muscle of medium size, with impenetrability being induced only with bones, i.e., muscles are allowed to intersect each other, the method takes, on average, 23 seconds on commodity hardware. Increasing the number of meshes involved, increases the overall time substantially; the method may take easily 20 minutes even for a couple of meshes. This is caused by the inefficiency of the algorithm used for penetration detection. A new algorithm, which is based on the collision detection of sphere hierarchies, has been developed and is currently in its test phase.

It is possible that the PK method will fail to provide the user with realistic results, if a muscle, after the initial rigid transformation, fully penetrates the volume of a bone, though we did not experience this problem in our experiments. Even the Rectus Femoris muscle that penetrated the femur bone significantly was wrapped correctly – see Figure 10. Another method, called the interpolated PK method has now been developed to lower the risk of this problem.

This new method extends the PK method by introducing several new positions between the rigidly transformed rest-pose and the current-pose in order to diminish the difference between the two positions being processed by the PK method. For example, if we assume that we have three positions: the rest-pose = P_0, P_1, and P_2 = the current-pose, the method starts with the calculation of paths of action lines and positions of vertices of bones at P_1 using linear interpolation of those at P_0 and P_2. Afterwards, the original PK method is run first to wrap muscles from P_0 into P_1, and then, in a successive run, these muscles are wrapped into P_2.

An advantage of the interpolated PK method is that places at which some penetration was detected during the wrapping are less apparent – compare Figure 11 with Figure 9. On the other hand, the overall time required increases significantly, mainly because coarse outer hulls must be constructed before each run of the PK method.

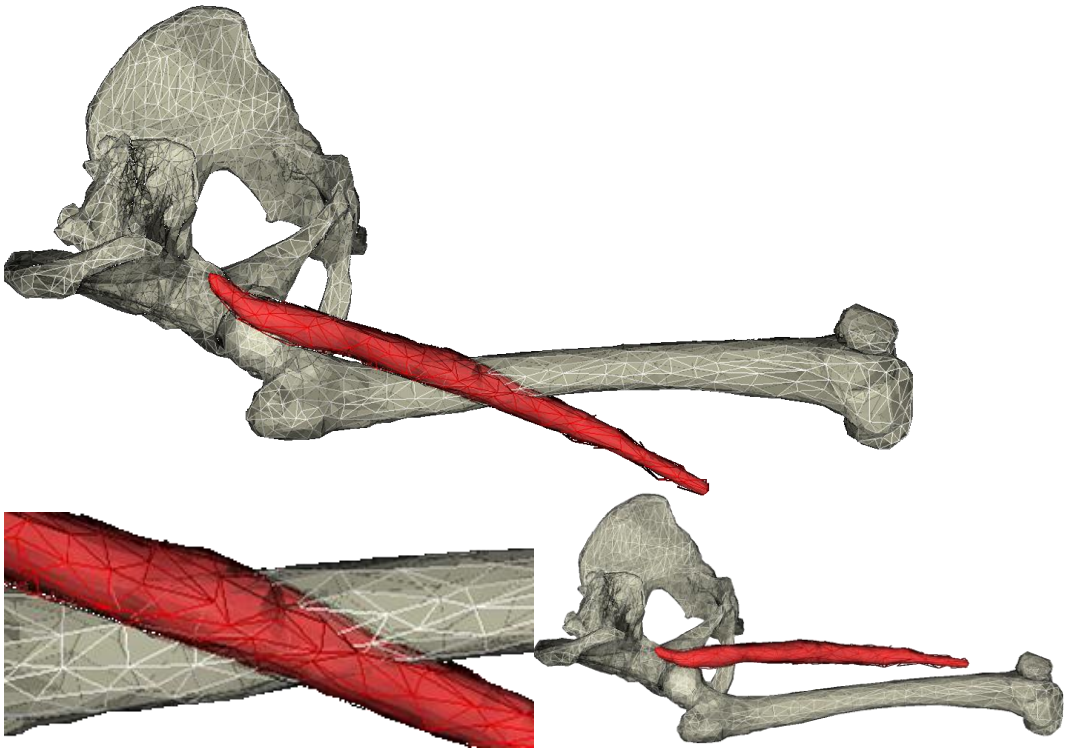

Figure 10. Rectus Femoris after the initial rigid transformation (top and bottom left) and after being wrapped (bottom right) using the PK method with impenetrability induced between it and bones. The PK method required 47 seconds and a volume error of 0.04% was achieved.

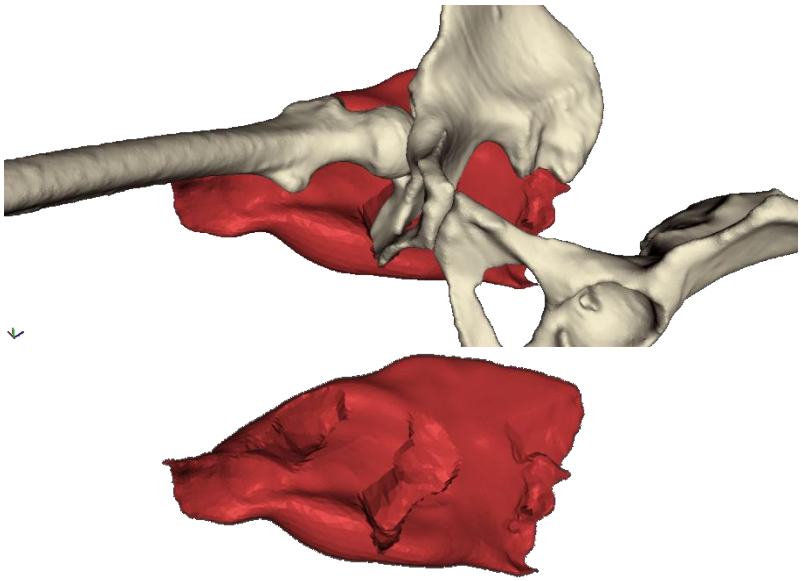

Figure 11. Gluteus Maximus muscle wrapped using the interpolated PK method with impenetrability induced between it and bones. The interpolated PK method required 114.5 seconds and a volume error of 8.41% was achieved.

3.4.3. Muscle Decomposition

The volume of a muscle can be decomposed into an arbitrarily large number of muscle fibres; with each muscle fibre being represented by a polyline of user-specified resolution, i.e., the number of linear segments from which the polyline is formed. Several decomposition methods have been developed to cope with the variety of muscle shapes.

The Advanced Slicing Method (ASM) [17],[18] improves the original method developed in the LHDL project. It starts with the production of polyline muscle fibres of the requested number and resolution in the interior of a unit cube, according to the template of fibres represented by Bézier curves as defined by Blemker & Delp [8]. Next, this unit cube and its associated polylines are subjected to an affine transformation such that the transformed cube is an oriented bounding box (OBB) of the muscle to be decomposed, and the attachment areas of the fibres in the cube are best aligned with those specified in the atlas for the muscle.

The transformed cube is sliced by planes perpendicular to the principal axis of the muscle and the contours that arise from the slicing are morphed on to the contours of the muscle obtained by the same slicing, employing the technique described by Ju et al. [19] to ensure consistency between the slices – see Figure 12. This maps the fibre vertices into the interior of the muscle.

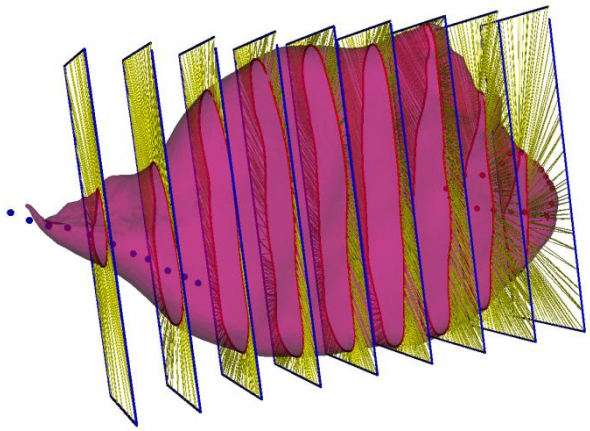

Figure 12. Slicing of Gluteus Medius and the transformed cube (blue polygons).

It was found that, for muscles with large attachment areas, the fibres produced have unrealistic paths in the proximity of their attachment areas. Hence, two cutting planes perpendicular to the principal axis and passing through the extreme landmarks of attachment areas (in the direction of the principal axis) are constructed and used to cut out the unwanted parts of fibres. In the next step, the cut parts of fibres are reconstructed by our muscle fibre extrapolation technique [18] and smoothed to reduce the noise present in the result.

A muscle of medium size (10K triangles) can be decomposed in about 1 second, even for a large number of fibres (256) and a high resolution (50 segments). As a resolution of 20-30 segments has been found to be sufficient for any muscle tested, and 50-100 fibres usually bring visually plausible results (see Figure 13), the method is suitable for interactive visualisation.

The method suffers from two drawbacks: if the attachment area is very large, fibres do not spread over the whole area, and if an attachment area is close to being perpendicular to the cutting plane, muscle fibres may end on one side of this attachment area. A proper validation of this method by an application expert is required to decide if these two drawbacks are significant or if the output is acceptable.

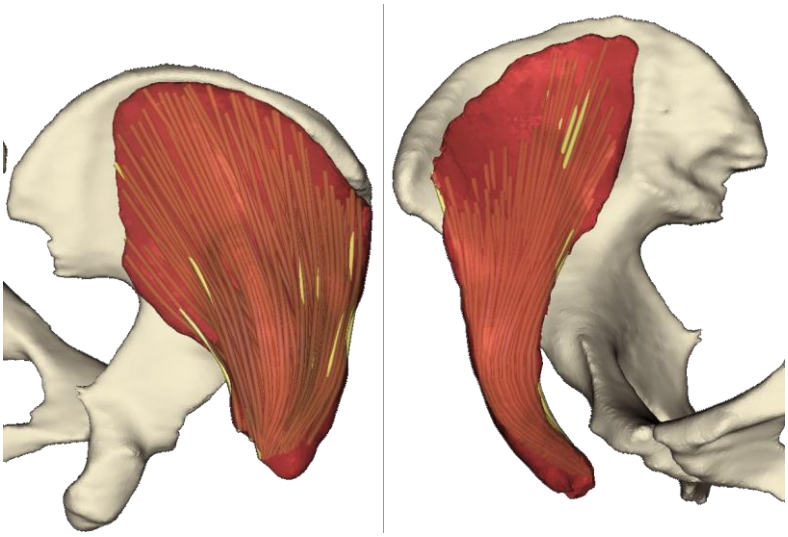

Figure 13. Muscle fibres of the Gluteus Medius and the Iliacus produced by the ASM decomposition method. 81 fibres of 20 segments were produced. The method required 470 ms for the Gluteus Medius and 582ms for the Iliacus.

The alternative approach is based on Kukacka decomposition method and differs from the previous method in two aspects. First, it slices the muscle according to some isovalue of the harmonic scalar field [20] that interpolates scalar values assigned to the contours of the origin and the insertion attachment areas (see Figure 14), which means that the muscle contour is not planar but may be twisted. Next, no correction of the paths of muscle fibres in the proximity of attachment areas is needed as they correspond to what is expected.

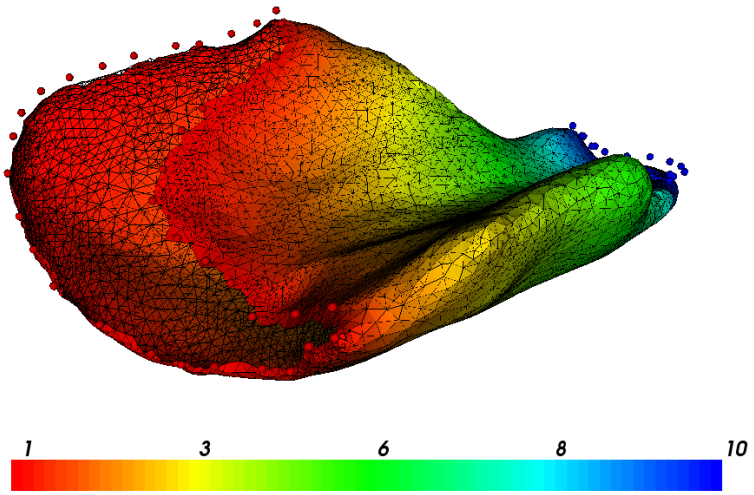

Figure 14. Harmonic scalar field computed on the surface of Gluteus Medius.

The method produces fibres that are distributed over the whole attachment area, as can be seen in Figure 15. On the other hand, for a complex muscle, the fibres produced might be twisted due to the inconsistency of the contours being morphed (planar rectangle vs. 3D twisted curve). In the current implementation, the harmonic scalar function is calculated at each time, which slows down the method by a couple of seconds. We note, however, that the function could be calculated just once and stored as a scalar field with the muscle geometry, in which case the performance of this method should be comparable with the ASM method.

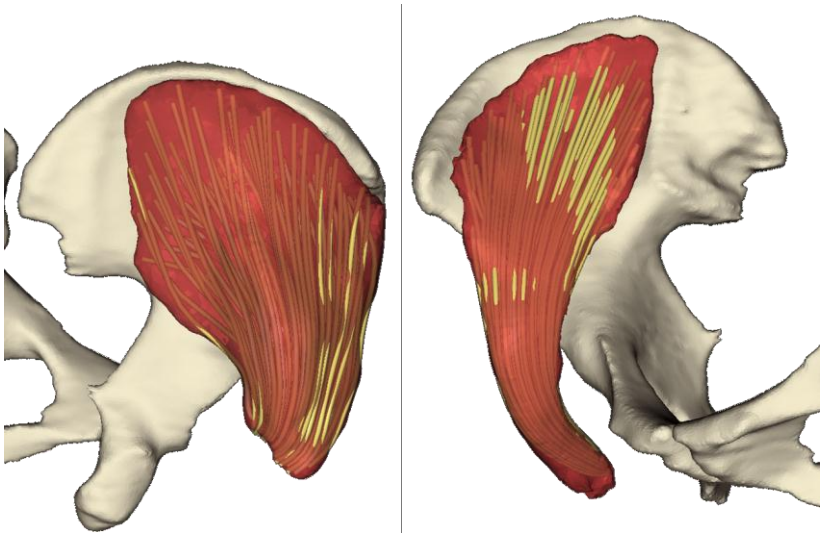

Figure 15. Muscle fibres of the Gluteus Medius and the Iliacus produced by the Kukacka decomposition method. 81 fibres of 20 segments were produced. The method required 3,230ms for the Gluteus Medius and 3,064ms for the Iliacus.

The third method, called MMSS, starts with the construction of a mass-spring system in which a muscle fibre is represented by mass particles interconnected by springs, and adjacent particles from different fibres are also interconnected by springs. Particles lie on a regular 3D grid, i.e., a parallel fibres architecture is assumed. The medial axis of the muscle is computed, and the muscle is then sliced by planes perpendicular to this axis. Muscle contours are uniformly sampled to find points on the surface on to which the particles on the boundary are moved and fixed, which induces the movement of internal particles in order to keep the mass-spring system in balance. The process is iterative and stops when the mass-spring system restores balance.

Figure 16. Muscle fibres of the Gluteus Medius and the Iliacus produced by the MMSS decomposition method. 81 fibres of 20 segments were produced. The method required 10,238 ms for the Gluteus Medius and 13,188 ms for the Iliacus.

As can be seen in Figure 16 and Figure 17, this method produces fibres that are uniformly distributed within the volume and do not twist unnaturally, which might be observable in the previous methods. On the other hand, if the attachment area is nearly parallel with the medial

axis, as in the case for the Iliacus muscle in Figure 17, the method is unable to spread fibres over the whole attachment area. Currently, the CPU implementation of the method requires tens of seconds to produce the fibres.

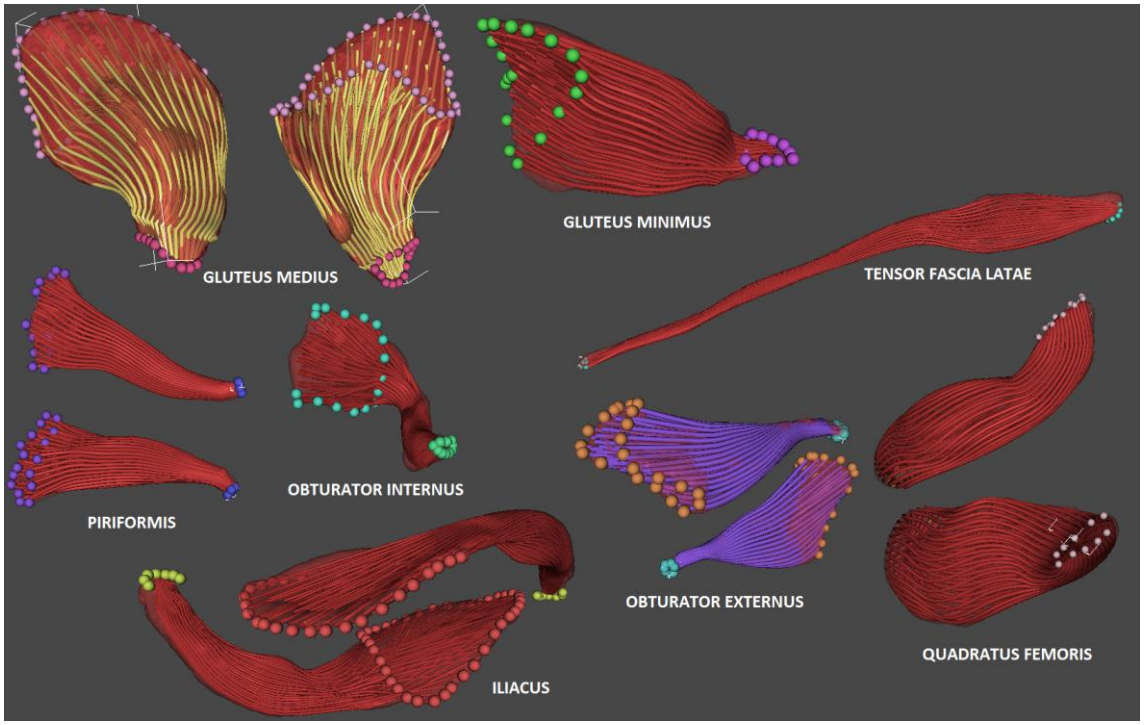

Figure 17. Muscle fibres of various muscles produced by the MMSS decomposition method.

The final method, called Cadaver Fibres (CF), exploits the fact that for some muscles in our generic musculoskeletal atlas model, a set of poly-lines representing several (usually four) fibres on the muscle surface, and a point cloud representing muscle tendons were manually measured during an autopsy of the same subject from whom the muscle model was created.

This method starts with construction of the muscle fibres on the surface by interpolation of the measured fibres using a constrained cubic spline [21], which was found to suppress undesirable oscillations of the fibre curve. Next, the fibres are connected to their nearest tendons and the method proceeds with the construction of muscle fibres in the interior of the muscle. First, a ray is cast from each vertex of the fibres produced so far in the direction of the normal vector in this vertex and the distance between the intersection points of this ray and the surface of muscle is measured. Vertices of internal fibres are then constructed by uniform sampling of the edge with one end-point in the vertex, having the same direction as the ray and a length equal to the measured distance. An example of the decomposition can be seen in Figure 18.

Although the paths of the muscle fibres in the proximity of the attachment areas are wrong in principle, the fibres produced might still be suitable for the verification of the paths of the fibres produced by other decomposition methods as this method exploits real muscle fibre data, whereas all other methods are purely artificial by their nature. We note that the method is still under development, though its immaturity does not affect this deliverable as whole.

Figure 18. Muscle fibres of Iliacus produced by the CF decomposition method.

3.4.4. Particle Construction

Particles are generated by uniform sampling along the fibres produced one of the muscle decomposition methods described in the previous section. Each particle on the boundary of the fibres is checked to find out if it lies in the proximity of any bone. If so, it is marked to be fixed to its closest bone. Particles are then interconnected by springs using a given spring layout pattern – see Figure 19. From all the patterns tested, the N-nearest neighbours was found to be the most suitable solution as it allows fast convergence of the model without using an excessive number of springs. The spring stiffness constants for the springs, and the mass and radius of a particle were chosen empirically. We note that the stiffness constants for springs along the fibre may differ from those between the fibres to reflect the anisotropy of muscle tissue

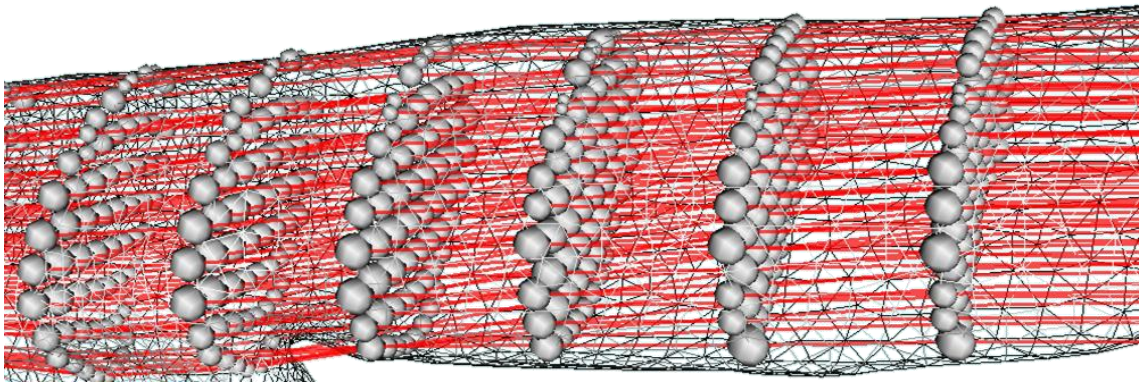

Figure 19. Particles interconnected using a 6-neighbour cubic layout.

3.4.5. Particle Wrapping

To speed up the wrapping, each particle is first transformed rigidly based on the transformation matrix of its nearest bone – see Figure 20. Each unfixed particle on the boundary of the fibres is further checked for penetration with the bones for the current time frame; the positions of any penetrated particles are adjusted to place them on the boundary surface of the bone. This enables muscle-bone inter-penetration to be avoided.

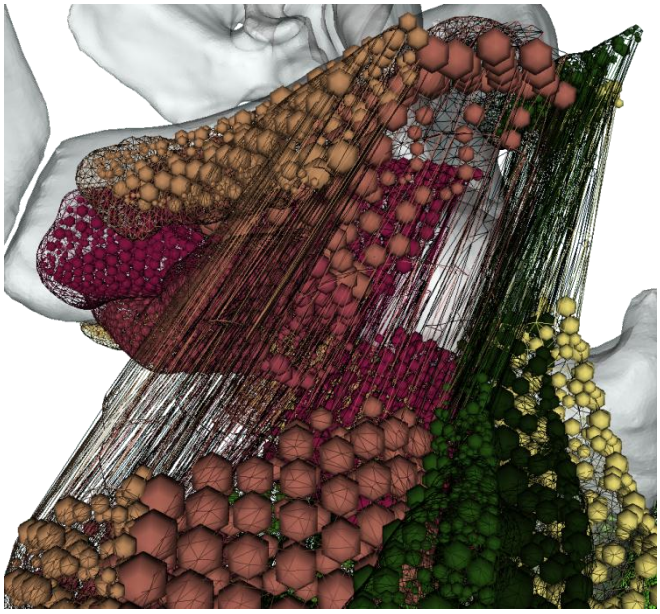

Figure 20. Particles after their initial transformation.

Particle-based simulation is then performed to adjust the positions of other particles automatically, based on the mass-spring system, i.e., unfixed particles are successively translated in order to minimize the energy of the whole mass-spring system and restore its equilibrium. A mechanism for the collision detection of particles from different muscles, and of particles and bones, is run regularly to impose impenetrability between the muscles and bones. Lazily updated bounding volume hierarchies are exploited to speed up the process. The process stops when the equilibrium is restored or when the maximum number of iterations is reached. An example is provided in Figure 21.

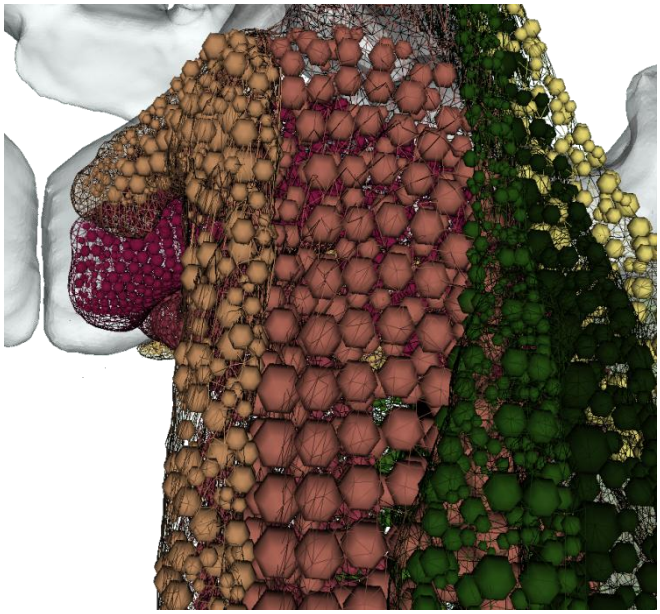

Figure 21. Particles after the mass-spring system simulation.

Although a speed of up to five frames per second can be achieved for a semi-large data set, the fidelity of the results tends to be rather poor, although sufficient for a quick visualization. To

achieve higher quality results, a less interactive speed – several to several tens of seconds – is required. This is, however, still faster than the PK method implemented in LHPBuilder or for any approach based on the Finite Element Method. Figure 22 shows muscle fibres corresponding to the wrapped particles. We note that the type of muscle fibres depend on the decomposition method used, which in this experiment was mostly the fastest (but inaccurate) LHDL method.

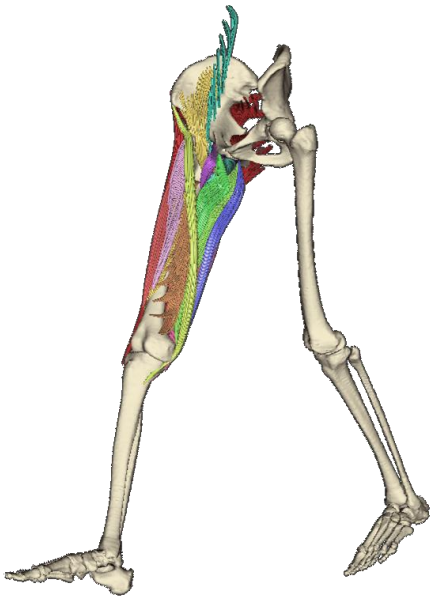

Figure 22. Muscle fibres of a walking human. All muscles were processed in 188.6 seconds.

3.4.6. Muscle Surface Reconstruction

Each vertex of the muscle surface is expressed relative to its N closest particles so that when the particles move, so does the vertex. Although particles are uniformly distributed in the rest-pose muscle, they may no longer be uniformly distributed in the current pose, and, therefore, this method tends to produce visible artefacts at places where the muscle is bent, as can be seen in Figure 23. A solution to this problem is currently being developed. However, we note that biomechanical calculations to estimate skeletal loading work with the fibres of a muscle but not with its surface, so while for cosmetic reasons a pleasing surface model is desirable, an imperfect reconstruction of the surface is not a major factor in relation to the delivery of the final morphable musculoskeletal model.

There are two other things that are worth pointing out. First, it can be seen that the particle-based pathway solves the penetration with the bones smoothly (compare Figure 23 with Figure 9 and Figure 11). Second, the error in the muscle volume preservation depends markedly on the number and distribution of muscle fibres in the volume.

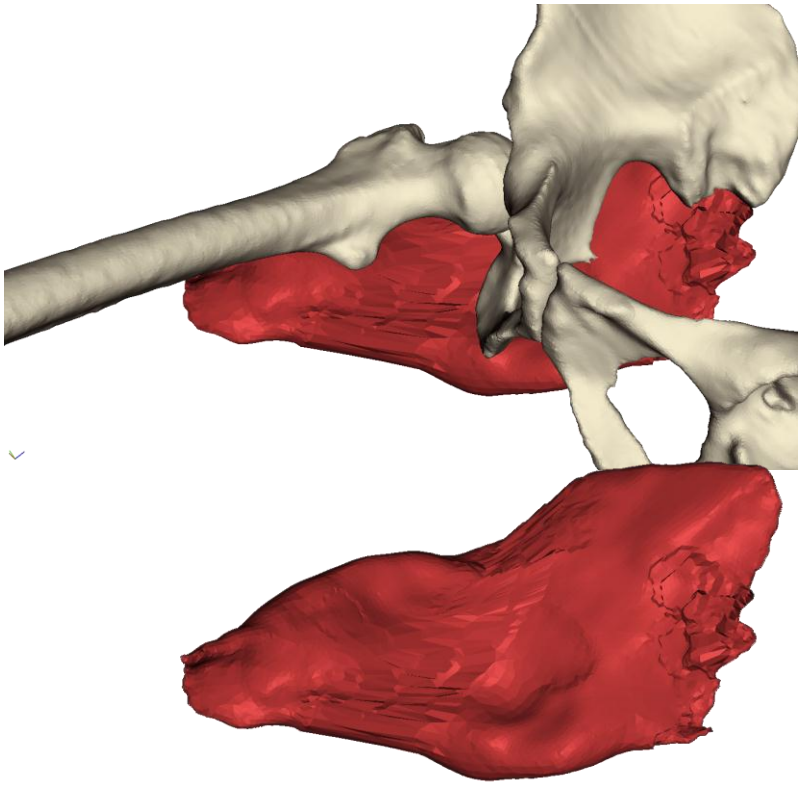

Figure 23. Gluteus Maximus muscle wrapped using a particle-based model pathway. The method required 10.1 seconds and a volume error of 17.9% was achieved.

4. Future Work

The individual components of our morphable musculoskeletal model have been tested and the results published; verification of the model as a whole is currently in progress and should be completed in the next few months. Further work, based on experience gained, continues to optimise various parts of the process.

A new fast algorithm for detecting penetration between meshes in the energy minimization model pathway, which has been recently developed, should replace the current rather slow algorithm. Particle wrapping should be parallelized to improve its performance; a parallel GPU version is now available but it still has to be tested and integrated. A new method for reconstructing the muscle surface from wrapped particles should be developed (it could exploit mean value coordinates [22] that were used in other parts of our solution with success). Work on the CF decomposition method will also proceed. Finally, the current morphable musculoskeletal model is delivered under MAF version 2.2 because MAF version 3.0, which was developed as a part of VPHOP activities (WP2) and is the core of VPHOP hypermodel, was not ready at the time when the development of our model started. Hence, it is planned to migrate our code to MAF3.

5. References

[1] Erdemir A, McLean S, Herzog W, van den Bogert AJ, Model-based estimation of muscle forces exerted during movements, Clinical Biomechanics 2007, 22(2):131-154.

[2] Garner BA, Pandy MG, The obstacle-set method for representing muscle paths in musculoskeletal models, Computer Methods in Biomechanics and Biomedical Engineering 2000, 3(1):1-30.

[3] Jensen RH, Davy DT, An investigation of muscle lines of action about the hip: A centroid line approach vs. the straight line approach, Journal of Biomechanics 1975, 8(2):103-110.

[4] Delp SL, Loan JP. A computational framework for simulation and analysis of human and animal movement. IEEE Computing in Science and Engineering 2000, 2(5):46-55.

[5] Arnold AS, Salinas S, Asakawa DJ, Delp SL, Accuracy of muscle moment arms estimated from MRI-based musculoskeletal models of the lower extremity, Computer Aided Surgery 2000,5:108-119.

[6] Audenaert A, Audenaert E, Global optimization method for combined spherical-cylindrical wrapping in musculoskeletal upper limb modelling, Computer Methods and Programs in Biomedicine 2008, 921:8-19.

[7] Ng-Thow-Hing V, Anatomically based models for physical and geometric reconstruction of animals, PhD Thesis, University of Toronto, Department of Computer Science, 2001.

[8] Blemker SS, Delp SL, Three-dimensional representation of complex muscle architectures and geometries, Annals of Biomedical Engineering 2005, 33(5):661-673.

[9] Kohout J, Kellnhofer P, Martelli S, Fast deformation for modelling for musculoskeletal system, In: Proc. GRAPP 2012, Rome, February 2012, pp. 16-25.

[10] Viceconti M, Zannoni C, Testi D, Petrone M, Perticoni S, Quadrani P, Taddei F, Imboden S, Clapworthy GJ, The Multimod Application Framework: a rapid application development tool for computer aided medicine, Computer Methods and Programs in Biomedicine 2007, 85(2):138-151.

[11] Schroeder W, Martin K, Lorensen B, The Visualization Toolkit, Third Edition, 2004.

[12] Sorkine O, Differential representations for mesh processing, Computer Graphics Forum 2006, 25(4):789-807.

[13] Tak S, Ko H, A physically-based motion retargeting filter. ACM Trans. Graph. 2005, 24(1):98-117.

[14] Cholt D, Progressive hulls: application on biomedical data, In: Proc. CESCG 2012, Smolenice, April 2012, pp. 9-16.

[15] Kellnhofer P, Kohout J, Time-convenient deformation of musculoskeletal system, In: Proc. Algoritmy 2012, Podbanske, September 2012, *accepted for publications*.

[16] Aubel A, Thalmann D, Efficient muscle shape deformation, In: Proc. IFIP 2000, pp. 132-142.

[17] Kohout J, Clapworthy GJ, Martelli S, Viceconti M, Muscle fibres modelling, In: Proc. GRAPP 2012, Rome, February 2012, pp. 58-66.

[18] Kohout J, Clapworthy GJ, Martelli S, Viceconti M, Fast realistic modelling of muscle fibres, 2012, Communications in Computer and Information Science (CCIS), Springer-Verlag (accepted for publication).

[19] Ju T, Schaefer S, Warren J, Mean value coordinates for closed triangular meshes, ACM Transactions on Graphics 2005, 243:561-566.

[20] Dong S, Kircher S, Garland M, Harmonic functions for quadrilateral remeshing of arbitrary manifolds, Comput. Aided Geom. Des. 2005, 22(5): 392-423.

[21] Kruger CJC,Constrained cubic spline interpolation for chemical engineering applications, August 2002, http://www.korf.co.uk/spline.pdf.

[22] Hormann K, Floater MS, Mean value coordinates for arbitrary planar polygons, ACM Transactions on Graphics 2006, 25(4):1424-1441.