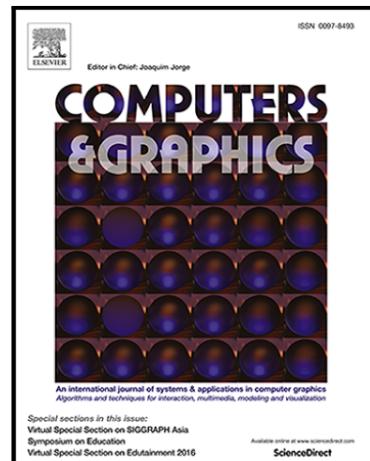


Accepted Manuscript

Real-time animation of human characters' anatomy

Aaron Sujar, Juan Jose Casafranca, Antoine Serrurier,
Marcos Garcia

PII: S0097-8493(18)30090-6
DOI: [10.1016/j.cag.2018.05.025](https://doi.org/10.1016/j.cag.2018.05.025)
Reference: CAG 2955



To appear in: *Computers & Graphics*

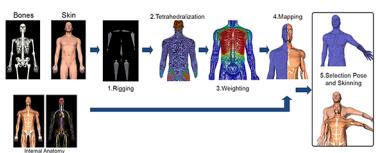
Received date: 16 March 2018
Revised date: 23 May 2018
Accepted date: 28 May 2018

Please cite this article as: Aaron Sujar, Juan Jose Casafranca, Antoine Serrurier, Marcos Garcia, Real-time animation of human characters' anatomy, *Computers & Graphics* (2018), doi: [10.1016/j.cag.2018.05.025](https://doi.org/10.1016/j.cag.2018.05.025)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- Real-time animation of virtual characters internal and external anatomy.
- Design of an automatic skeletal animation pipeline for internal tissues.
- Adapting the patient anatomy to any pose required in a given medical procedure.
- The technique is flexible enough to deal with B-Reps and volumetric models.

Graphical Abstract



Real-time animation of human characters' anatomy

Aaron Sujar^{a,*}, Juan Jose Casafranca^a, Antoine Serrurier^b, Marcos Garcia^a

^aGMRV Grupo de Modelado y Realidad Virtual, Universidad Rey Juan Carlos, Spain

^bRWTH Institute of Medical Informatics, Uniklinik RWTH University Aachen, Germany

ARTICLE INFO

Article history:

Received June 6, 2018

Keywords: linear blend skinning, dual quaternion skinning, real-time animation

ABSTRACT

The animation of articulated characters is a central problem in the computer graphics field. Skeletal animation techniques define a workflow which has proven to be effective for boundary representations (B-Reps). This paper extends the classical skeletal animation pipeline to deal with characters internal tissues. In contrast to most common approaches, the proposed technique automates all the stages of this workflow. Well known skinning algorithms, such as *Linear Blending Skinning*, *Dual Quaternion Skinning* or *Optimized Centers of Rotation* were adapted to allow the use of our technique in applications where interactivity is required. The pipeline proposed in this paper can be used in many computer graphics systems such as games or educational applications to visualize and animate the internal anatomy of a virtual character at interactive rates.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays, trainers based on Virtual Reality (VR) are becoming more and more popular [1, 2, 3]. The usefulness of these simulators is particularly clear for training the required non-cognitive skills needed in many medical procedures. Simulators provide a safe environment where trainees can practice. Improvements in computer performance, emergence of new peripheral devices and development of new and more efficient physically-based simulation techniques allow an effective transfer of skills from the virtual to the real world [4, 5, 6].

The new generation of medical simulators will not only improve the simulation quality, but they will also allow training on real patient-specific data [7, 8]. This approach is not trouble-free: (i) in many cases patient-specific data are not available, (ii) current medical imaging techniques cannot capture all patient tissues adequately, (iii) it is not easy to extract the tissue's mechanical properties from the images. Alternatively, VR medical trainers do not need patient-specific data. Their goal is to transfer the required skills from the simulator to the real world.

These applications just need to create a database of virtual patients which includes the typical anatomical variations that are relevant for the simulated medical intervention. The models in the database do not have to be created from specific real patients. Nevertheless, medical imaging can be used to create several types of anatomical variations from a set of real patients.

The different medical imaging techniques capture the patient data in specific subject positions. Usually, these positions are different from the subject positions required in the simulated medical procedure. This paper describes a novel technique that allows adapting a virtual patient anatomy to any desired pose. Our algorithm was designed following the requirements listed below:

- The algorithm will transform any anatomical structures of the virtual patient into the new pose. Furthermore, the algorithm will work with incomplete anatomical data. This requirement is especially relevant since no imaging technique can capture all the patient tissues. The only two required tissues are the patients' bones and their skins.
- The algorithm will work without a mechanical characterization of the patient tissues.

*Corresponding author: +34 91488-4580
e-mail: aaron.sujar@urjc.es (Aaron Sujar)

- 1 • The algorithm will run at interactive rates in order to allow
2 the user supervision during the pose selection process.

3 In order to fulfil these requirements, we decided to follow a
4 geometric-based method. Our approach computes a displace-
5 ment field from the bones' movement. This field is used to
6 transform the virtual tissues. In the following sections, we de-
7 scribe the techniques that have been developed or adapted to use
8 the classical skeletal animation workflow to adjust a given pose
9 to the internal patient tissues. The proposed technique com-
10 putes most of the tasks during a pre-process stage, allowing to
11 run the pose selection phase at interactive rates.

12 It should be kept in mind that, although our objective is to
13 apply this technique to adapt virtual patient models to a specific
14 pose, this technique can be also used in other scenarios which
15 require visualizing the characters' inner anatomy, such as edu-
16 cation, film industry or video games. Our method is especially
17 useful when real-time rendering is required.

18 This work is an extension of previous paper [9]. This new
19 version of the algorithm has been integrated in a virtual pa-
20 tient generation pipeline. The system proposed in [10] has been
21 modified to provide additional information to automatize the
22 rigging phase (see section 3.1). Furthermore, Le and Hodgins
23 proposed in [11] a new skinning technique that solves most of
24 the limitations of traditional techniques. The interactive phase
25 of their method can be directly applied to ours, but the pre-
26 process stage was adapted to deal with volumetric meshes (see
27 section 3.5).

28 2. Previous Work

29 Nowadays, skeletal animation is the most used 3D computer
30 graphics technique to animate articulated characters. This tech-
31 nique allows transferring the pose of a virtual skeleton to the
32 models surface mesh. Since virtual skeleton DoFs (Degrees of
33 Freedom) are far less than the mesh DoFs, this technique sim-
34 plifies the pose selection. The animation process is divided into
35 four stages: (i) *rigging*, (ii) *weighting*, (iii) *pose selection* and
36 (iv) *skinning*. Firstly, in rigging stage, a specific virtual skeleton
37 is created out of the virtual model. Then, during the weighting
38 stage, each vertex of the surface mesh is linked to one or more
39 bones of the virtual skeleton. Next, the skeleton key poses are
40 defined; and finally, these poses are transferred to the surface
41 mesh in the skinning phase. Generally, an artist is in charge of
42 the first three stages and performs them manually or using user-
43 supervised tools. Therefore, the final result depends to a large
44 extent on the artist's skills. In our system, the user is only in
45 charge of the pose selection. Nevertheless, this final step can be
46 substituted for any animation technique, such as Mo-cap sys-
47 tems.

48 Several techniques for automating the rigging phase can be
49 found in the literature [12, 13, 14]. Some algorithms, such as
50 the one proposed in [15], try to compute a specific virtual skele-
51 ton from the mesh. Alternatively, other techniques adjust a pre-
52 defined skeleton to the current surface mesh [16, 17]. In our
53 system, this approach suits better since it is designed to deal
54 with virtual human models with similar bone structure.

55 Although there are a few works regarding the *rigging* phase,
56 it is hard to find non-supervised *weighting* techniques. Some
57 authors [18, 19] compute the surface mesh weights while build-
58 ing the virtual skeleton. Since we adapt a pre-defined virtual
59 skeleton to a specific human character, this approach does not
60 fit into our method.

61 Baran and Popović [20] propose an algorithm to automate
62 both rigging and weighting stages. In their work, they adapt
63 a previous generated virtual skeleton to a 3D model. Then,
64 the influences of the virtual bones over the vertex of the sur-
65 face mesh are computed using Laplace diffusion equation. This
66 technique is constrained to B-Rep models and the surface mesh
67 must completely enclose the virtual skeleton. The weighting
68 phase of our animation pipeline extends this technique to deal
69 with the internal anatomy of the model and to take into account
70 the models bone tissue for building the virtual skeleton. Baran
71 and Popović [20] propose an algorithm to automate both rig-
72 ging and weighting stages. In their work, they adapt a previous
73 generated virtual skeleton to a 3D model. Then, the influences
74 of the virtual bones over the vertex of the surface mesh are com-
75 puted using Laplace diffusion equation. This technique is con-
76 strained to B-Rep models and the surface mesh must completely
77 enclose the virtual skeleton. The weighting phase of our ani-
78 mation pipeline extends this technique to deal with the internal
79 anatomy of the model and to take into account the models bone
80 tissue for building the virtual skeleton. Jacobson *et al.* [21] im-
81 prove the weighting phase to deal with point handles, virtual
82 bones and cages. They minimize the Laplacian energy subject
83 to the handles' constraints (among other constraints) to achieve
84 high-order smoothness (see equation 1):

$$\arg \min_{W_j, j=1..m} \sum_j^m \frac{1}{2} \int_{\Omega} \|\nabla W_j\|^2 dV, \quad (1)$$

85 where W_j are the weights of the handle j and m is the number of
86 handles. Our approach is equivalent since we resolve equation
87 4 and our solutions make the Laplacian energy functional sta-
88 tionary. Therefore, our technique ensures high-order smooth-
89 ness. Additionally, our formulation drives to a linear equation
90 system, while this technique requires solving a sparse quadratic
91 programming problem in order to impose the constraint set.

92 Since Magnenat-Thalmann *et al.* [22] seminal work, a vast
93 number of skinning techniques have been proposed. Currently,
94 the *Linear Blending Skinning* (LBS) is the most used in real-
95 time applications due to its efficient implementation on modern
96 GPUs. This technique has been described in several papers, e.g.
97 in [23]. The problems associated with this technique (*candy-
98 wrapper effect*, volume loss) are well-known in the field. Kavan
99 *et al.* describe in [24] a technique called *Dual Quaternion*
100 *Skinning* (DQS) in order to solve the LBS limitations without
101 a substantial performance decrease. Although DQS works well
102 in most cases, in some scenarios, it produces a significant vol-
103 ume gain. Numerous solutions to LBS and DQS issues can
104 be found in the bibliography [25] at the cost of decreasing the
105 computational performance or leading to other artefacts. In a
106 more recent work, Le and Hodgins [11] present a skinning al-
107 gorithm (CoR) that computes a specific rotation centre for each
108 mesh vertex. This approach improves the volume conservation

1 without increasing the computational cost.

2 Geometrically-based algorithms do not only suffer from vol-
 3 ume conservation problems. In [26], the authors propose a
 4 real-time geometrically based technique to deal with skin con-
 5 tacts and sliding. Furthermore, their solution does not require a
 6 weighting stage. They build a scalar field for each bone. The
 7 mesh vertices are characterized by a value on this scalar field.
 8 However, this method cannot be directly applied to our prob-
 9 lem, since it does not take into account the bones shape. There-
 10 fore, the osseous tissues will be affected by the smooth field
 11 function, leading to unrealistic deformations. *Rumman and*
12 Fratarcangeli [27] follow a physically based approach to ani-
 13 mate any articulated character. Since they discretise the char-
 14 acter inside, their technique can be applied to animate the char-
 15 acter internal anatomy. However, the algorithm computational
 16 cost requires a low-resolution volumetric mesh to run in real-
 17 time. In contrast, the model proposed in this paper can be ap-
 18 plied to more complex meshes.

19 Leaving aside the classical skeletal animation, biomechanical
 20 and musculoskeletal models and physically based techniques
 21 have emerged in the last years. Most of the biomechanical
 22 models are manually created for a specific part of the human
 23 anatomy [28]. On one hand, *Patterson et al.* [29] efficiently
 24 simulate muscle behaviour. On the other hand, their technique
 25 requires a long and tedious manual stage that must be per-
 26 formed by an expert. Similarly, *Fan et al.* [30] describe a
 27 physically-based technique to simulate the muscle preserving
 28 their volume. The anatomical model can be generated auto-
 29 matically from medical images, but it needs the muscle shapes
 30 in its relaxed and final pose. Although, there are techniques
 31 that can obtain muscle-skeletal models from medical images
 32 [31, 32, 33], the cited techniques do not run at interactive rates
 33 and require detailed information of the patient. This kind of
 34 information is not always available. Furthermore, techniques
 35 usually focus on simulating the muscles and the skeleton, with-
 36 out considering other anatomical structures. *Ichim et al.* [34]
 37 combine kinematic and dynamic methods to achieve real-time
 38 in the context of facial animation. Blendshapes are efficient
 39 and directly controllable, while physically-based models pro-
 40 vide dynamic effects, secondary motions, collision response
 41 and incompressibility. However, blendshapes are not adequate
 42 to animate articulated characters and skeletal animation tech-
 43 niques are usually preferred.

44 It is worth to mention the work of *Dicko et al.* [35]. In this
 45 research, the authors detail a method for modelling the inter-
 46 nal anatomy of a virtual character, transferring the inner tissues
 47 from a reference model to an arbitrary surface mesh. Their tech-
 48 nique can infer the internal structures form MRI data or using a
 49 semi-automatic algorithm. The manuscript reports some issues,
 50 which will be solved in future work, when using MRI data. This
 51 work focuses on the modelling process and very few details are
 52 given on how to animate the resulting meshes. Despite this, they
 53 use their technique to transfer articulated systems but they nei-
 54 ther explain how to compute the rig nor the mesh weights. To
 55 do so, their technique can be easily incorporated into the work-
 56 flow described in this paper. *Dicko et al.* work can be also used
 57 to transfer the reference model muscle lines of action (described

58) to the final character for its biomechanical simulation.
 59 A full volumetric transfer will be required to obtain accurate
 60 muscles movements. Furthermore, physically based simulation
 61 methods are outside of this paper scope due to our real-time in-
 62 teraction requirements. *Kadlecik et al.* [37] present a method to
 63 create anatomical models for its physically-based deformation.
 64 They improve *Dicko et al.* work proposing a fully automatic
 65 reconstruction while preserving realistic bone shapes and us-
 66 ing multiple scans. In certain situations, the authors observe
 67 that the bones protrude through the muscles. Among other rea-
 68 sons, this artefact is caused by the use of multimaterial property
 69 of each body tetrahedron. Our technique does not suffer from
 70 this problem since we compute a smooth deformation field and
 71 since the bones tetrahedrons cannot embed any other structure.
 72 Additionally, this method is not meant to be used in real-time
 73 systems. However, it simulates inertial effects and secondary
 74 motions.

3. Animation Pipeline

75 As it has been already mentioned, in this paper we propose
 76 a technique to adapt the internal and external anatomical infor-
 77 mation of a virtual character to any desired pose. With this pur-
 78 pose, we adapt and design a set of algorithms that establishes
 79 a skeletal animation pipeline to animate the internal tissues of
 80 a virtual human character. In contrast to classical skeletal ani-
 81 mation, the transformations of the virtual bones are not applied
 82 directly to the character's 3D model. Instead, the bones are used
 83 to compute a displacement field which is employed to transform
 84 the character anatomy. This approach can be applied not only to
 85 B-Rep models of the internal tissue but also to volumetric rep-
 86 resentations (Fig.11). It is worth mentioning that most of the
 87 stages of our pipeline are fully automatic.

88 The system requirements are listed in the section 1. Our tech-
 89 nique was designed to deal with incomplete anatomical descrip-
 90 tions, only the skin and bones must be identified. The animation
 91 pipeline is divided into the following stages (Fig. 1):

- 92 • *Rigging:* A predefined virtual skeleton is adjusted to the
 93 character anatomy. The algorithm uses the bones' models
 94 to compute the rotation centre of each joint of the virtual
 95 skeleton.
- 96 • *Tetrahedralization:* In this step, we generate an internal
 97 tetrahedral mesh from the skin and bone models. This vol-
 98 umetric mesh will be used to define a continuous displace-
 99 ment field associated with the movement of the bones.
- 100 • *Weighting:* This stage calculates automatically the influ-
 101 ence of each bone on the tetrahedral mesh vertices.
- 102 • *Mapping:* The character's virtual tissues are assigned to
 103 the tetrahedrons of the volumetric mesh.
- 104 • *Pose selection and skinning:* In this step, the movements of
 105 the virtual skeleton are transferred to the tetrahedral mesh
 106 using a standard skinning algorithm such as LBS, DQS or
 107 CoR. Then, the tetrahedral mesh movements are applied
 108 to the tissues models. In our system, the user interactively

1 selects the virtual character pose. Nevertheless, other techniques
2 such as MoCap systems could be used instead.

3 3.1. Rigging

4 In a similar manner to real bones, the virtual skeleton allows
5 the body movements. The virtual skeleton is represented by
6 a hierarchical set of connected virtual bones. The movement
7 of each virtual bone is defined by a rotation in the bone's local
8 coordinate system. The centre of this coordinate system is
9 called joint. In this step, our technique adjusts a predefined
10 virtual skeleton to the character bone tissue redefining the virtual
11 skeleton joints. Currently, there exist several techniques to
12 adapt a generic virtual skeleton to a 3D character [19, 38, 12].

13 We plan to integrate our algorithm into a system to build virtual
14 patient models for virtual reality medical simulators. Each
15 virtual patient is built registering a generic virtual model into
16 either real patient data (Fig. 12) or average data from multiple
17 real patients. For further details on this process, see [39].

18 In the generic model bone tissue (before the registering process), some anatomically meaningful regions of the bone tissue
19 are manually identified, labelling some of their vertices (Fig.
20 2). We are assuming that when the generic model is registered
21 to patient-specific data, these identified regions will also represent
22 the same anatomical region of the patient. In order to make
23 the virtual skeleton calculations more robust, the algorithm considers,
24 when it is possible, large regions for which most of the
25 vertices will be registered properly, outnumbering the failure
26 cases. After the registering process, these labelled regions are
27 used to compute the local reference system of each bone. We
28 estimate two orthogonal vectors and the bone rotation centre
29 from a set of annotated vertices. The third vector of the basis
30 is calculated as the cross product of the first two vectors. This
31 process was specifically designed for each bone of our virtual
32 skeleton and it is implemented and described in the Matlab code
33 and the document provided as supplementary material.

35 3.2. Tetrahedralization

36 At this point, the algorithm discretises the inside of the virtual
37 model into a tetrahedral mesh. In the next step (Sec. 3.3),
38 we compute each bone's influence over this mesh. Then, tetra-
39 hedral mesh is used to estimate a displacement field (Sec. 3.5).
40 This field is transferred to the virtual model using the mapping
41 calculated in Sec. 3.4. In order to control the discretisation
42 process, the tetrahedral mesh is not directly generated from the
43 virtual mesh. Instead, we generate a volumetric image from the
44 skin and bone tissue models. In this 3D image, bones' voxels
45 and characters' inside are labelled.

46 The size of the 3D image depends on the voxel and models
47 bounding box size. To keep the memory and time requirements
48 low, we have selected a voxel size value in such a way that the
49 3D image size is always smaller than 250x700x120.

50 The voxelization process starts labelling the voxel colliding
51 with the skin (Fig. 3.a). Then, the inside voxels are labelled using
52 the technique described in [40] (Fig. 3.b). Finally, following
53 the same procedure, the bone tissue is labelled (Fig. 3.d).

54 Once the 3D image has been built, we use it to create a tetra-
55 hedral mesh [41]. In order to set up this algorithm, we must

56 achieve a trade-off between accuracy and efficiency. We tuned
57 the algorithm to increase the mesh resolution around the skin
58 and bone boundaries. In our experiments, we kept the number
59 of tetrahedrons below 3.5×10^6 and the number of vertices below
60 8×10^5 . Finally, the tetrahedral mesh is tagged using the
61 osseous model (Fig. 4).

62 3.3. Weighting

63 The displacement field inside of the tetrahedral mesh is cal-
64 culated by interpolating the displacement of each vertex. The
65 movements of the tetrahedral model's vertices are associated
66 with the movement of one or more bones. This stage calculates
67 how the movements of the bones influence the tetrahedral mesh
68 vertices.

69 $w_{i,j}$ weights how the bone j influences the vertex i and must
70 fulfil the following conditions:

$$w_{i,j} \geq 0 \quad \forall i \in V \wedge \forall j \in B \quad (2)$$

$$\sum_{j \in B} w_{i,j} = 1 \quad \forall i \in V \quad (3)$$

71 where V is the set of tetrahedral mesh vertices and B is the set
72 of the virtual skeleton bones.

73 Baran and Popović in [20] explain the properties the weights
74 $w_{i,j}$ must meet: (i) the vertex weights must be independent of
75 the mesh topology or resolution; (ii) the weights must vary
76 smoothly along the volume. In order to ensure these two prop-
77 erties, they proposed to use Laplace diffusion equation. The
78 influence of the bones is propagated across the volume in the
79 same way as the temperature does. The idea behind our work is
80 similar but we have adapted the formulation because it cannot
81 be directly applied to our problem: (i) their algorithm can only
82 be applied to B-Rep models and (ii) those representations must
83 enclose the virtual skeleton completely.

84 With the aim of computing the weights W_j of a bone j , we
85 solve the steady case posed by Eqn. 4. In order to resolve Enq.
86 4 for a bone j , we impose the following boundary conditions:
87 we consider that the value W_j is 1 inside the tetrahedrons la-
88 belled as j ; and the value W_j is 0 inside the tetrahedrons labelled
89 as k , where k is any bone not equal to j .

$$\nabla^2 W_j = 0. \quad (4)$$

90 The previous equation is discretised using the *Finite Element
91 Method* (FEM) and the barycentric coordinates of the tetrahe-
92 dron as shape functions (further information in [42]). Eqn. 5
93 shows the final discretised diffusion equation:

$$\mathbf{A}\mathbf{W}_j = \mathbf{b}_j, \quad (5)$$

94 where \mathbf{A} is the system coefficient matrix, the vector \mathbf{b}_j depends
95 on the boundary conditions of the bone j and \mathbf{W}_j is a vector
96 that contains $w_{i,j}$ for every non-labelled vertex i . \mathbf{A} is the same
97 matrix for all the bones and it is symmetric and positive definite.
98 This allows us to calculate the Cholesky decomposition of this
99 matrix only once, and use it to solve the linear system for each
100 bone.

101 It is worth mentioning that this formulation fulfils the con-
102 straints shown in Eq. 2 and 3. First, the maximum and mini-
103 mum values will only be reached on the bones and those val-
104 ues are 0 and 1. Second, if we consider $\mathbf{b} = \sum_{j=0}^n \mathbf{b}_j$ and

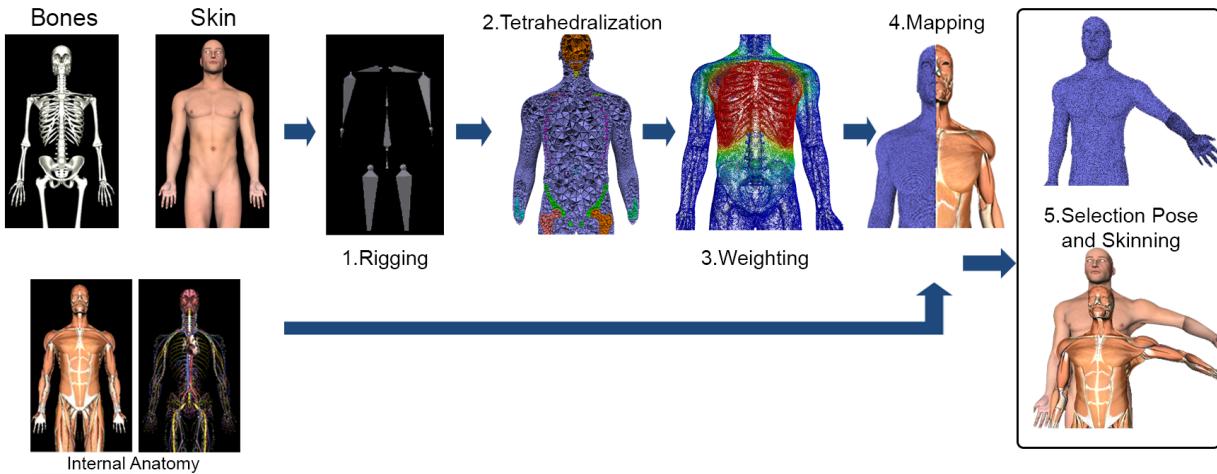


Fig. 1. Algorithm overview.

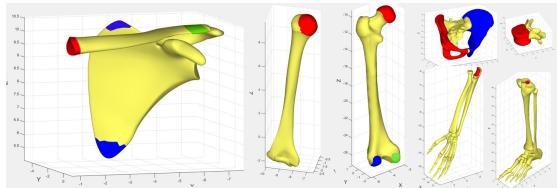


Fig. 2. 3D representations of a few bones. The areas coloured are used to compute the joint positions and orientations.

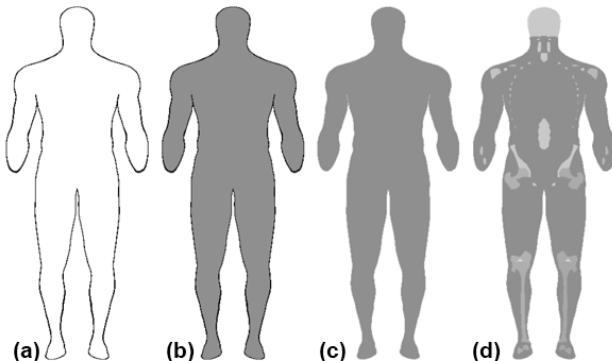


Fig. 3. Coronal slices of the volumetric image at different stages of the voxelization procedure.

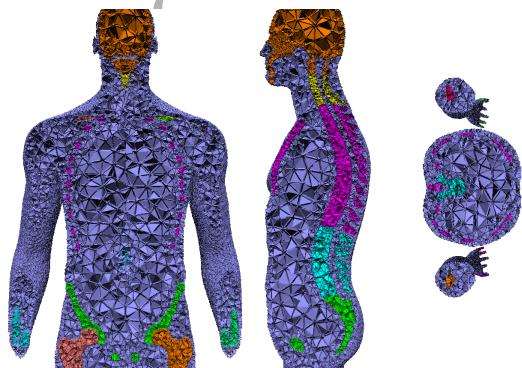


Fig. 4. Coronal, sagittal and axial slices showing the tetrahedralization phase results. The tetrahedrons labelled as bones are highlighted in different colours.

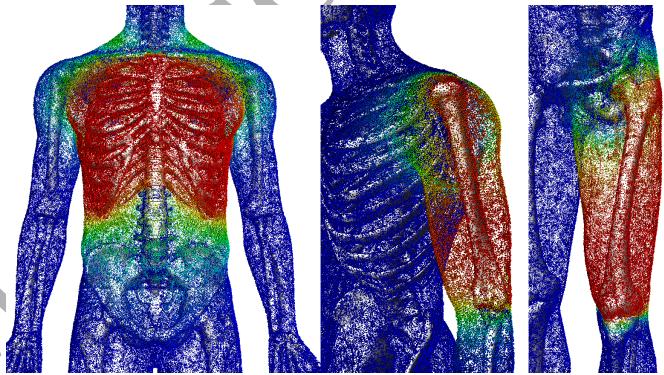


Fig. 5. Tetrahedral mesh with colour coding representing the weights of the chest, the humerus and the femur.

$\mathbf{W} = \sum_{j=0}^n \mathbf{W}_j$, where n is the number of bones, $\mathbf{AW} = \mathbf{b}$ describes a system where all the boundary points will take the value 1. Therefore, since the maximum and minimum values can only be reached on the boundary, all elements of \mathbf{W} take the value 1, proving Eq. 3.

Figure 5 shows the influence of the chest, humerus and femur bones on the tetrahedral mesh vertex. The red areas show values close to 1 and the blue ones close to 0.

3.4. Mapping

Section 3.5 explains how the displacement field (defined in the tetrahedral mesh) is transferred to the virtual tissues. A preliminary step, presented in this section, is to link the anatomical model vertices to the tetrahedrons of the volumetric mesh. The simpler approach to map a set of n to a tetrahedral mesh of m tetrahedrons has a cost of $O(n m)$. We have accelerated the process storing the tetrahedral mesh in a *Spatial Hash Table* [43]. Every tissue model vertex is mapped using a hash table by searching iteratively the closest tetrahedron. The naïve approach takes 4 hours 55 minutes in mapping 1277325 vertices to 2584115 tetrahedrons, where 4.6% of the vertices are outside the tetrahedral mesh. This time is reduced to 36.35 seconds when using Spatial Hashing.

1 3.5. Pose Selection and Skinning

2 In this stage, the user selects interactively a skeleton pose
 3 which is transferred to the virtual character model. Currently,
 4 our implementation allows animating the virtual model using
 5 direct kinematic, loading pre-recorded pose or animation. Tech-
 6 niques such as retargeting [44] or inverse kinematic can be di-
 7 rectly applied to our system.

8 The skeleton animations are transferred to the volumetric
 9 mesh using a skinning technique and then, the transformed
 10 tetrahedrons are used to define a deformation field which is
 11 used to animate the character's tissue models. To perform the
 12 first step we implemented the skinning technique described in
 13 [11]. We decided to use CoR because it solves most of the
 14 DQS and LBS issues and it is fully compatible with DQS and
 15 LBS (it also needs the vertex weights). CoR computes specific
 16 centres of rotation for all the tetrahedral mesh vertices. Since
 17 this information is precomputed, this algorithm has a negligi-
 18 ble impact on the system performance in comparison with DQS
 19 and LBS. In fact, CoR interactive step can be efficiently im-
 20 plemented on modern graphics architectures. This algorithm is
 21 based on the idea that vertices with similar skinning weights
 22 must follow similar transformation. In their work, *Le and Hod-*
 23 *gins* [11] proposed the similarity function shown below:

$$24 \quad s(\mathbf{w}_p, \mathbf{w}_s) = \sum_{Vi \neq j} w_{p,i} w_{p,j} w_{s,i} w_{s,j} \exp -\frac{(w_{p,i} w_{s,j} - w_{s,i} w_{p,j})^2}{\sigma^2} \quad (6)$$

25 where \mathbf{w}_p and \mathbf{w}_s are the weight vector of vertices p and s , and
 26 σ is configuration parameter. In our system, σ takes a non-
 27 zero positive value, lower than 0.1. This similarity function
 28 is used to compute the new rotation centres. We adapted the
 29 equation proposed in [11] to deal with tetrahedral meshes in the
 30 following manner:

$$31 \quad \text{cor}_p = \frac{\sum_{t \in T} s(\mathbf{w}_p, \frac{\mathbf{w}_{t1} + \mathbf{w}_{t2} + \mathbf{w}_{t3} + \mathbf{w}_{t4}}{4}) V_t \mathbf{c}_t}{\sum_{t \in T} s(\mathbf{w}_p, \frac{\mathbf{w}_{t1} + \mathbf{w}_{t2} + \mathbf{w}_{t3} + \mathbf{w}_{t4}}{4}) V_t} \quad (7)$$

32 where cor_p is the new rotation centre of vertex p , t is a tetrahe-
 33 dron that belongs to the tetrahedral mesh T , V_t is the volume of
 34 the tetrahedron t , \mathbf{c}_t is the centroid of the tetrahedron t and \mathbf{w}_{t1} ,
 35 \mathbf{w}_{t2} , \mathbf{w}_{t3} and \mathbf{w}_{t4} are the weights of the vertices of the tetrahe-
 36 dron t . Once the rotation centres are computed they can be used
 37 in the interactive step as it is shown in [11].

38 The displacement field of a point inside of a tetrahedron can
 39 be calculated by interpolating the displacement values of its
 40 vertices. We interpolate this field using the barycentric coordi-
 41 nates of each tetrahedron. Therefore, the computed displace-
 42 ment field is continuous but not differentiable inside the volu-
 43 metric mesh and a constant transformation matrix can be cal-
 44 culated for each tetrahedron (see [45] for further information).
 45 The transformation matrix of a given tetrahedron is applied to
 46 the tissue vertices associated to that tetrahedron. Both tasks,
 47 the computation of the transformation matrices and how they
 48 are applied to the tissue vertices, are performed on the system
 49 graphics card.

50 4. Implementation details

51 In our system, the pose selection process is supervised by a
 52 user. Therefore, the animation and skinning stages must run
 53 at interactive rates. These two stages can be implemented to
 54 meet this requirement. On the other hand, the tetrahedraliza-
 55 tion, the weighting, the mapping and the computation of the rota-
 56 tion centres of the tetrahedral mesh vertex are expensive from
 57 a computational point of view. Fortunately, these steps can be
 58 performed only once in a pre-process module.

59 5. Results

60 This section shows some results and explains the tests per-
 61 formed to evaluate our technique. First, we evaluated the
 62 pipeline computational performance. In these tests, we used
 63 four different anatomical models: the *ZygoteBodyTM* male
 64 (ZM) and female (ZF) models [46], the *AnatomiumTM* male
 65 model [47], as well as a bar model linking four bones to il-
 66 lustrate the various skinning techniques explored in the study.
 67 Figure 6 summarizes the size of these models and their associ-
 68 ated volumetric meshes. The animations used in the evaluation
 69 were obtained from *Carnegie Mellon University Motion Cap-*
 70 *ture Database*. [48]. All tests were run on an Intel®i7-4820K
 71 @ 3.7GHz PC with a NVIDIA®GeForce GTX 770 graphics
 72 card and 16GB of RAM. Table 1 shows the time spent in pre-
 73 processing the input data. The total time never exceeds seven
 74 minutes. This step is done only once for each model. The per-
 75 formance of the skinning step is shown in table 2. The anima-
 76 tion step includes the volumetric mesh skinning and transfer-
 77 ring the tetrahedron deformations to the tissue models. Clearly,
 78 the skinning phase runs at interactive rates, even with our high
 79 detailed meshes. In addition, table 2 compares the performance
 80 between three different skinning methods. We would like to
 81 stress that the three skinning techniques have almost the same
 82 performance.

83 In figures 7 and 8, we compare the results obtained with
 84 several skinning techniques. Figure 7 depicts how the bones'
 85 movements deform the tetrahedral mesh. Figure 8 illustrates
 86 how the displacement field of the tetrahedral mesh is transferred
 87 to the anatomical structures. The LBS volume loss and the DQS
 88 volume gain are appreciable in both images. In these examples,
 89 CoR solves most of the artefacts introduced by LBS and DQS.

90 However, CoR cannot ensure volume conservation as
 91 physically-based techniques do. We compared CoR with a
 92 physically-based model. Since we do not have a proper char-
 93 acterization of the tissues mechanical properties, the goal of
 94 the implementation is to guarantee volume conservation. We
 95 use a co-rotational FEM formulation to solve the steady-state
 96 problem for a linear, isotropic and homogeneous material. Be-
 97 sides, the deformations are measured using the *Cauchy* strain
 98 tensor. The boundary conditions, needed to solve the steady-
 99 state problem, are given by the positions of the vertices labelled
 100 as bones. The co-rotational formulation calculates the internal
 101 forces caused by the deformations in a non-rotated configura-
 102 tion. Then, the internal forces are rotated again into the final
 103 configuration [45]. The algorithm needs to compute the ele-
 104 ment rotations in the final configuration. For this purpose, the

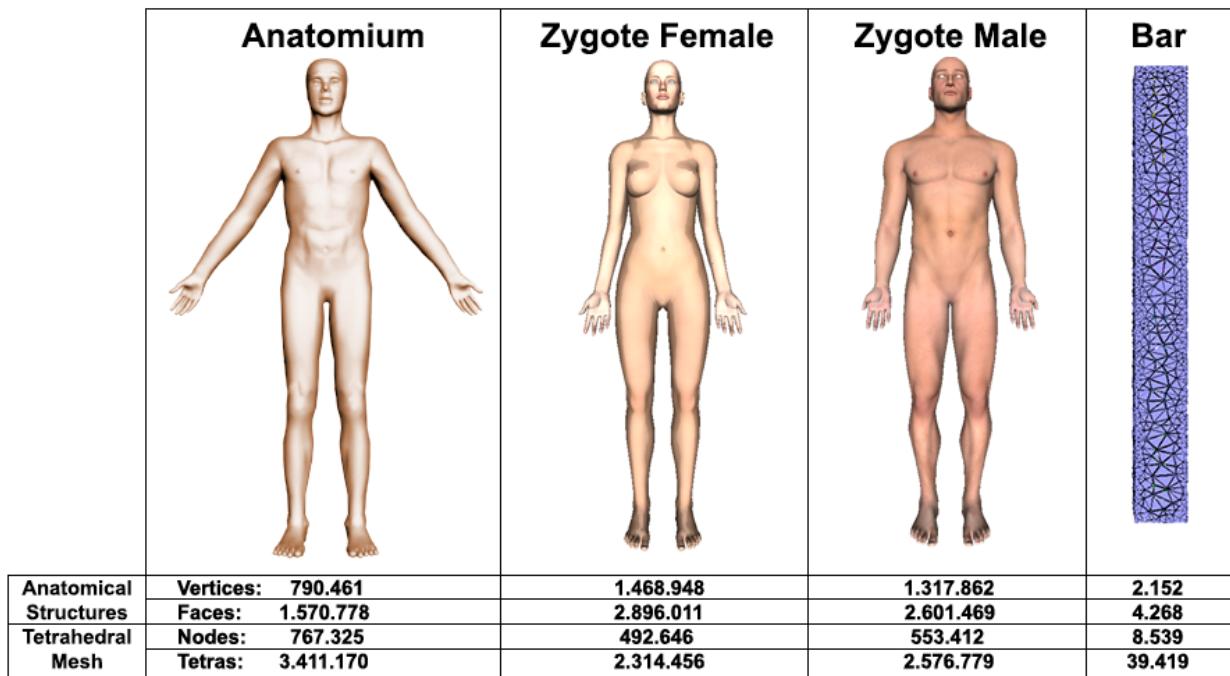


Fig. 6. Size of the models used in the tests.

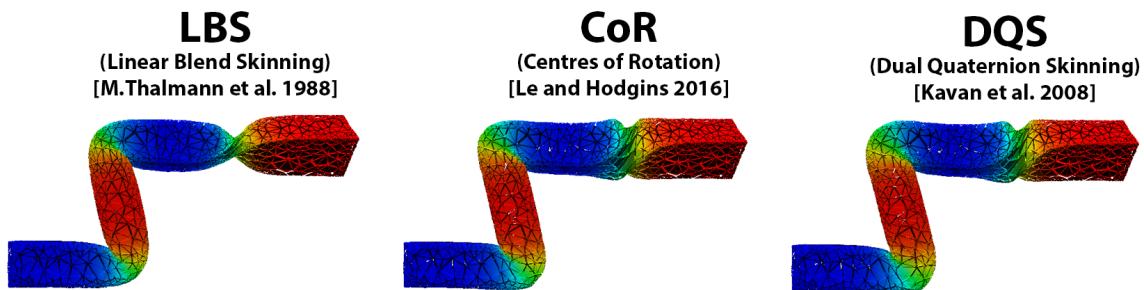


Fig. 7. Bar deformation for 3 skinning techniques. Joint rotations: 100 bending, -100 bending, 135 twisting.

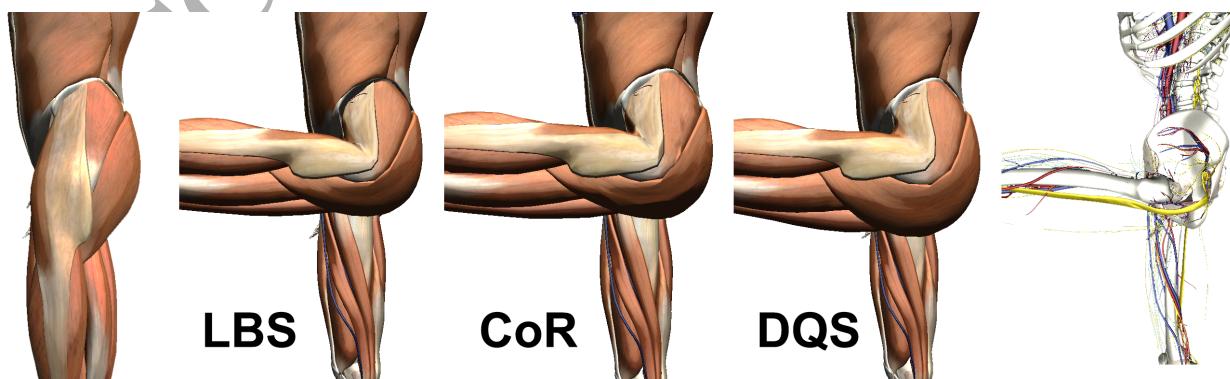


Fig. 8. Left: muscles of the model in a rest position; three middle images: muscles of the model deformed by leg bending for 3 skinning technique (LBS lost volume at the top of the thigh, DQS increases the volume at gluteus, CoR prevents these artefacts); right: other tissues of the model with the same deformation.

Table 1. Performance of the preprocessed stages. Time spent in each stage in milliseconds.

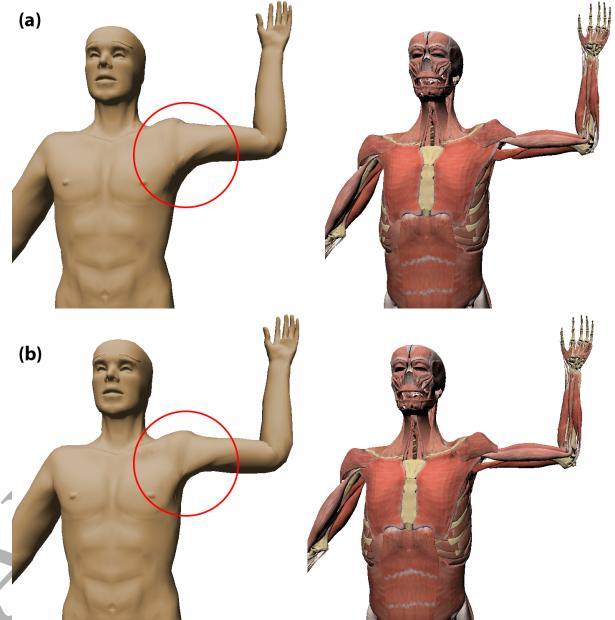
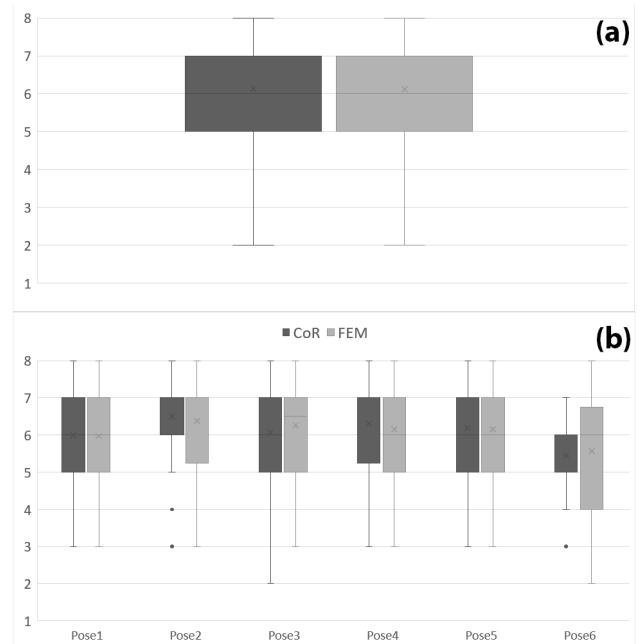
Model	Rigging	Tetrahedralization	Weighting	Mapping	CoR Preprocess
ZM	32698	69044	11762	51160	138005
ZF	33251	63401	9171	71635	208886
A	31891	120465	23318	44521	115214

Table 2. Mean and minimum frame rate during a walk cycle for the skinning stage.

Model	LBS	DQS	CoR
ZM	111-90	100-90	100-76
ZF	76-60	66-52	60-50
A	166-142	166-125	142-100

1 solution is refined iteratively. The elastic used model can be
2 tuned with two parameters: the *Poisson ratio* and the *Young module*. The *Poisson ratio* controls the volume conservation
3 and it should take a value close to 0.5 (the real value has to
4 be lower to ensure numeric stability). Since the material is ho-
5 mogenous, this value has no impact on the outcome. We chose
6 the *Young module* to improve the stability of the system. We
7 tested different values and we selected the one that conditioned
8 the system matrix better for inversion (checking the matrix con-
9 dition number). Figure 9 illustrates how the implemented FEM-
10 based model solves some volume issues. However, CoR works
11 well in most scenarios and we believe that users would not pre-
12 fer one over the other. To test this hypothesis, we have run a user
13 study. A total of 16 subjects participate in our study (2 females,
14 14 males; between the ages of 20 to 52; 13 of them are com-
15 puter graphics professionals). In our experiments, participants
16 were asked to rate the realism of the deformation presented in
17 22 static images, using a Likert from 1 to 8. The images dis-
18 played 6 poses and several models and internal tissues. Half
19 of them were created with FEM and the others with CoR. We
20 showed a reference image of the tissues in its rest pose together
21 with the deformations generated by FEM and CoR. The ref-
22 erence images were always displayed first and then the order
23 of the FEM and CoR deformations were randomized to avoid
24 bias. The online questionnaire designed for this test can be
25 found at the URL: <https://goo.gl/WprGP6>. The assump-
26 tion of homoscedasticity holds but normality does not. There-
27 fore, we compared the results obtained for FEM and CoR using
28 the non-parametric Wilcoxon signed rank for matched samples.
29 The result of this test confirms that the differences between both
30 techniques are not significant ($p\text{-value} > 0.9$). Figure 10 sum-
31 marizes the obtained results.
32

33 Figure 11 shows how the proposed pipeline is applied to a
34 voxel-based representation of a virtual character. Instead of
35 transferring the weights of the tetrahedral mesh nodes to the
36 embedded tissue vertices, our technique computes a displace-
37 ment field inside the virtual character and, consequently, our
38 pipeline can be applied to both B-Rep and volumetric mod-
39 els. In the volumetric case, we determine the pixels inside each
40 tetrahedron in the deformed configuration and then, they are
41 mapped back to the non-deformed configuration using the in-
42 verse of the displacement field. For the first step, we iterate

**Fig. 9. Comparison between CoR (a) and a physically-based model (b) de-
signed to preserve volume. CoR slightly increases the volume in the armpit
region.****Fig. 10. Boxplot comparing the results obtained in the user study. Image
(a) summarizes the global results, while the image (b) compares the results
obtained for the different poses used in our experiments.**

1 over each tetrahedrons bounding box and we use the barycentric
 2 coordinates to identify the pixels inside the tetrahedron. This
 3 process can be easily parallelized on modern GPUs, allowing
 4 real-time rendering.

5 Our algorithm can be used to transform models generated
 6 from real patient data from the rest position to any desired pos-
 7 ition. Figure 12.a displays: the skin (left), the pre-processed
 8 tetrahedral mesh (centre) and the bones (right) of a virtual char-
 9 acter in rest position; while figure 12.b shows the same three
 10 structures in its deformed configuration. This example has been
 11 built from real medical images. The registration algorithm [39]
 12 could not provide a full anatomical description of the patient
 13 since only local images were available. Many medical proce-
 14 dures are performed in a local area. Therefore, not having a
 15 full anatomical description of the virtual patient should not be
 16 a major issue. In brief, figure 12 illustrates how our technique
 17 works with incomplete information.

18 Figure 13 exemplifies how our technique can be applied to
 19 the volumetric and surface-based representations of the same
 20 character. For this example, we used the Voxel-Man’s Seg-
 21 mented Inner Organ model ([49], [50]), which is a segmentation
 22 of the Visible Human dataset. Finally, additional results for the
 23 ZM and ZF models are shown in Figure 14.

24 6. Conclusions and Future Work

25 In recent years, virtual reality is proving its potential in the
 26 medical training field. One of the main advantages of those
 27 tools is to allow the trainees to face a large variety of scenar-
 28 os. Therefore, the virtual patient database is a key component
 29 of these systems. In this context, our system adapts a virtual
 30 patient to the pose required in a given medical procedure. Re-
 31 cently, musculoskeletal simulation has significantly advanced
 32 in the computer graphics field. These techniques usually re-
 33 quire an accurate description of the patient tissues, which is not
 34 always available. This paper presents a pipeline to transform
 35 any available patient anatomy to a desired pose. This technique
 36 automates the skeletal animation paradigm to deal with inter-
 37 nal tissues of the model. Our algorithm works with incomplete
 38 anatomical models and does not need a mechanical description
 39 of the tissue behaviours. The technique is flexible enough to
 40 deal with B-Reps and volumetric models as long as the osseous
 41 tissue is properly labelled.

42 Our technique follows a geometrically-based approach.
 43 Therefore, it provides a heuristic solution. For this reason, this
 44 technique is not suitable for surgical planning. Alternatively,
 45 medical trainers do not need a specific real patient model but
 46 a set of anatomically different patients. Our system provides
 47 plausible poses for training and educational purposes. As fea-
 48 ture work, we plan to implement this solution in a virtual reality
 49 system to assess its functionality in a practical environment. We
 50 would like to underline that our system can be used to animate
 51 the internal anatomy of virtual characters in other fields such as
 52 video games, film industry, etc.

53 Additionally, the skinning phase of our pipeline is very fast,
 54 particularly in comparison with physically-based models. On
 55 the one hand, accurate models rarely achieve interactive frame

56 rates. On the other hand, less accurate models such as point
 57 base dynamics must work with less complex 3D models to en-
 58 sure real-time animations [27].

59 The rigging phase is integrated with our virtual patient gener-
 60 ator system. Other automatic [20] or manual rigging techniques
 61 can be used instead. The only requirement is that the virtual
 62 skeleton should match the bone tissue of the virtual character.
 63 We plan to adapt new rigging algorithms in the near future.

64 References

- [1] Ullrich, S, Grottko, O, Fried, E, Frommen, T, Liao, W, Rossaint, R, et al. An intersubject variable regional anesthesia simulator with a virtual patient architecture. International journal of computer assisted radiology and surgery 2009;4(6):561–570.
- [2] Bayona, S, Akhtar, K, Gupte, C, Emery, RJ, Dodds, AL, Bello, F. Assessing performance in shoulder arthroscopy: The imperial global arthroscopy rating scale (igars). The Journal of Bone & Joint Surgery 2014;96(13):e112.
- [3] Deserno, TM, da Oliveira, JE, Grottko, O. Regional anaesthesia simulator and assistant (rasimas): medical image processing supporting anaesthesiologists in training and performance of local blocks. In: Computer-Based Medical Systems (CBMS), 2015 IEEE 28th International Symposium on. IEEE; 2015, p. 348–351.
- [4] Seymour, NE. Vr to or: a review of the evidence that virtual reality simulation improves operating room performance. World journal of surgery 2008;32(2):182–188.
- [5] Dawe, SR, Windsor, JA, Broeders, JA, Cregan, PC, Hewett, PJ, Mader, GJ. A systematic review of surgical skills transfer after simulation-based training: laparoscopic cholecystectomy and endoscopy. Annals of surgery 2014;259(2):236–248.
- [6] Buckley, CE, Kavanagh, DO, Traynor, O, Neary, PC. Is the skillset obtained in surgical simulation transferable to the operating theatre? The American Journal of Surgery 2014;207(1):146–157.
- [7] Willaert, W, Aggarwal, R, Van Herzele, I, Cheshire, N, Vermassen, F. Recent advancements in medical simulation: Patient-specific virtual reality simulation. World Journal of Surgery 2012;36(7):1703–1712. URL: <http://dx.doi.org/10.1007/s00268-012-1489-0>. doi:10.1007/s00268-012-1489-0.
- [8] Votta, E, Le, TB, Stevanella, M, Fusini, L, Caiani, EG, Redaelli, A, et al. Toward patient-specific simulations of cardiac valves: state-of-the-art and future directions. Journal of biomechanics 2013;46(2):217–228.
- [9] Casafranca, JJ, Sjar, A, Garca, M. An Interactive Algorithm for Virtual Patient Positioning. In: Sbert, M, Lopez-Moreno, J, editors. Spanish Computer Graphics Conference (CEIG). The Eurographics Association; 2015,doi:10.2312/ceig.20151197.
- [10] Quijano, S, Serrurier, A, Aubert, B, Laporte, S, Thoreux, P, Skalli, W. Three-dimensional reconstruction of the lower limb from biplanar calibrated radiographs. Medical Engineering and Physics 2013;35(12):1703–1712.
- [11] Le, BH, Hodgins, JK. Real-time skeletal skinning with optimized centers of rotation. ACM Transactions on Graphics (TOG) 2016;35(4):37.
- [12] Pan, J, Chen, L, Yang, Y, Qin, H. Automatic skinning and weight retar-
geting of articulated characters using extended position-based dynamics. The Visual Computer 2017;:1–13.
- [13] Jacobson, A. Part ii: Automatic skinning via constrained energy optimi-
zation. SIGGRAPH Course 2014;:1–28.
- [14] Pantuwong, N, Sugimoto, M. A novel template-based automatic rigging
algorithm for articulated-character animation. Computer Animation and
Virtual Worlds 2012;23(2):125–141. URL: <http://dx.doi.org/10.1002/cav.1429>. doi:10.1002/cav.1429.
- [15] Bharaj, G, Thormählen, T, Seidel, HP, Theobalt, C. Automatically rig-
ging multi-component characters. In: Computer Graphics Forum; vol. 31.
Wiley Online Library; 2012, p. 755–764.
- [16] Feng, A, Casas, D, Shapiro, A. Avatar reshaping and automatic rigging
using a deformable model. In: Proceedings of the 8th ACM SIGGRAPH
Conference on Motion in Games. ACM; 2015, p. 57–64.
- [17] Avril, Q, Ribet, S, Ghafourzadeh, D, Dionne, O, Ramachandran, S,
de Lasa, M, et al. Animation setup transfer for 3d characters. In: Com-
puter Graphics Forum; vol. 35. Wiley Online Library; 2016, p. 115–126.

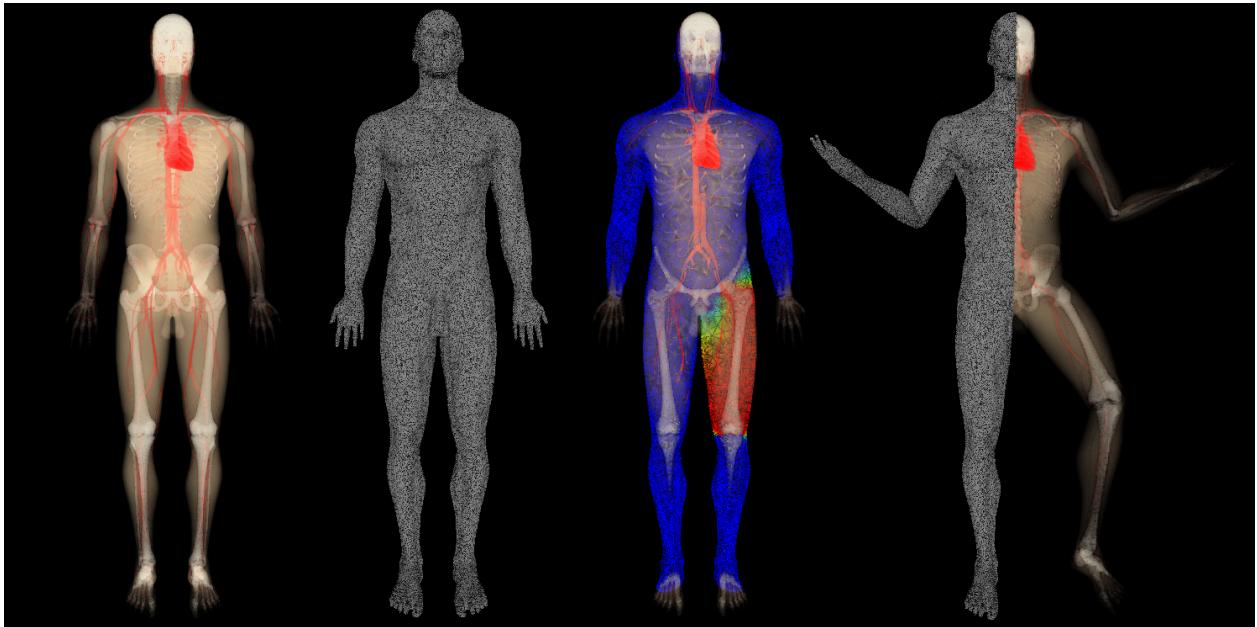


Fig. 11. Overview of the process of animating a volumetric character. From left to right: (1) original volumetric model, (2) tetrahedral mesh, (3) tetrahedral mesh with colour coding representing the weights of the femur as illustration (high weights towards red, low towards blue) and (4) tetrahedral mesh and volumetric data in a user-defined pose.

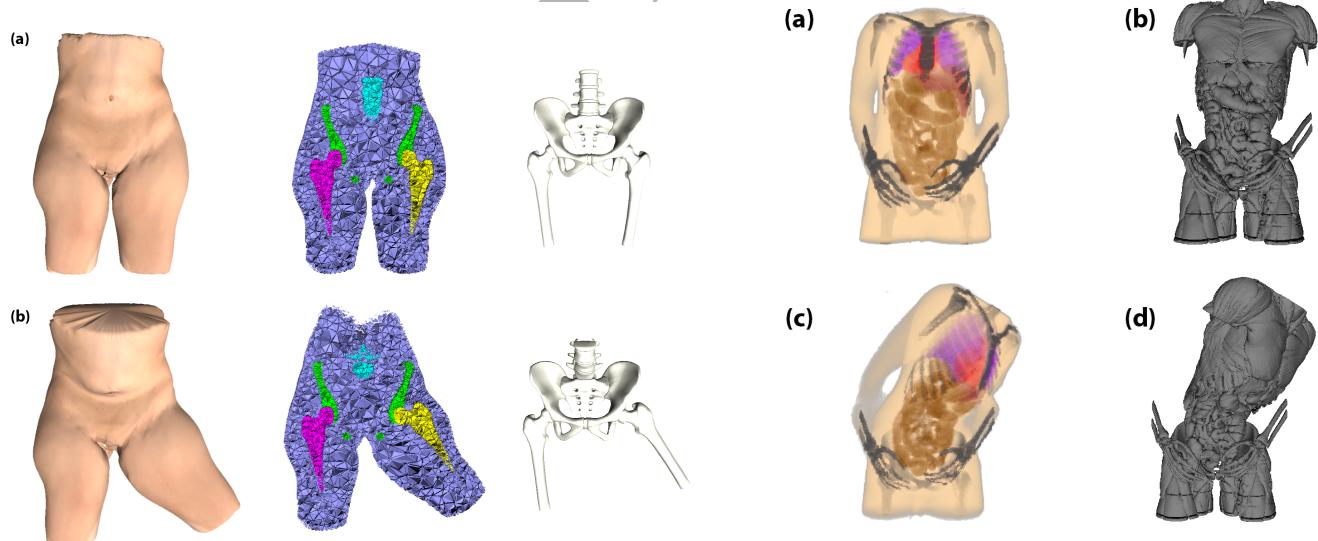


Fig. 12. Figure (a) shows a virtual patient model built from real patient data in its rest position. The positioning of the model can be performed using our algorithm (b).

Fig. 13. Our technique can be applied to the volumetric (a) and (c) and surface-based representations (b) and (d). Images (a) and (b) show the model inner organs in its rest position. Images (c) and (d) illustrate the volumetric and the B-rep models transformed into a given pose.

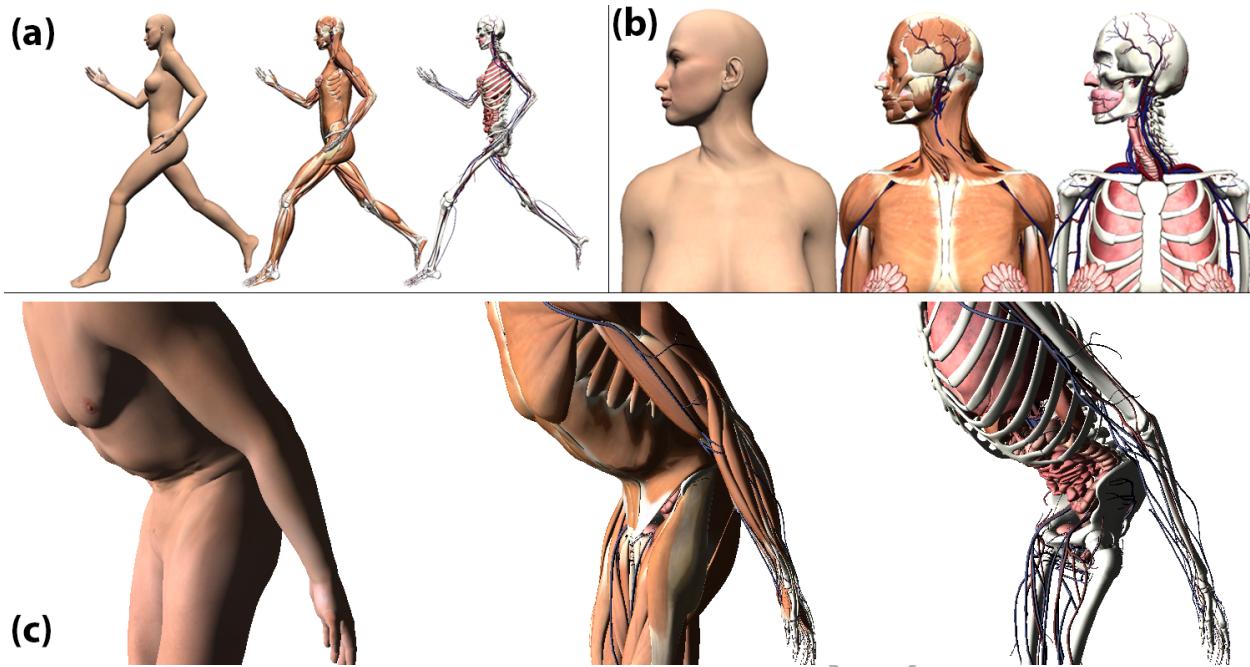


Fig. 14. Several results obtained with CoR. (a): a key frame of a running cycle applied to ZF. (b): a neck bend applied to ZF. (c): A belly bend applied to ZM.

- [18] Borosán, P, Jin, M, DeCarlo, D, Gingold, Y, Nealen, A. Rigmesh: automatic rigging for part-based shape modeling and deformation. ACM Transactions on Graphics (TOG) 2012;31(6):198.
- [19] Huang, CH, Boyer, E, Ilic, S. Robust human body shape and pose tracking. In: 3DTV-Conference, 2013 International Conference on. IEEE; 2013, p. 287–294.
- [20] Baran, I, Popović, J. Automatic rigging and animation of 3d characters. ACM Trans Graph 2007;26(3). URL: <http://doi.acm.org/10.1145/1276377.1276467>. doi:10.1145/1276377.1276467.
- [21] Jacobson, A, Baran, I, Popović, J, Sorkine, O. Bounded biharmonic weights for real-time deformation. ACM Trans Graph 2011;30(4):78:1–78:8. URL: <http://doi.acm.org/10.1145/2010324.1964973>. doi:10.1145/2010324.1964973.
- [22] Magnenat-Thalmann, N, Laperrière, R, Thalmann, D. Joint-dependent local deformations for hand animation and object grasping. In: Proceedings on Graphics Interface '88. Toronto, Ont., Canada: Canadian Information Processing Society; 1988, p. 26–33. URL: <http://dl.acm.org/citation.cfm?id=102313.102317>.
- [23] Lewis, JP, Cordner, M, Fong, N. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '00; New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5; 2000, p. 165–172. URL: <http://dx.doi.org/10.1145/344779.344862>. doi:10.1145/344779.344862.
- [24] Kavan, L, Collins, S, Žára, J, O'Sullivan, C. Geometric skinning with approximate dual quaternion blending. ACM Trans Graph 2008;27(4):105:1–105:23. URL: <http://doi.acm.org/10.1145/1409625.1409627>. doi:10.1145/1409625.1409627.
- [25] Rumman, NA, Fratarcangeli, M. State of the art in skinning techniques for articulated deformable characters. In: 11th International Conference on Computer Graphics Theory and Application, GRAPP 2016; Part of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2016. SciTePress; 2016..
- [26] Vaillant, R, Guennebaud, G, Barthe, L, Wyvill, B, Cani, MP. Robust iso-surface tracking for interactive character skinning. ACM Trans Graph 2014;33(6):189:1–189:11. URL: <http://doi.acm.org/10.1145/2661229.2661264>. doi:10.1145/2661229.2661264.
- [27] Abu Rumman, N, Fratarcangeli, M. Position-based skinning for soft articulated characters. In: Computer Graphics Forum; vol. 34. Wiley Online Library; 2015, p. 240–250.
- [28] Lee, SH, Sifakis, E, Terzopoulos, D. Comprehensive biomechanical modeling and simulation of the upper body. ACM Trans Graph 2009;28(4):99:1–99:17. URL: <http://doi.acm.org/10.1145/1559755.1559756>. doi:10.1145/1559755.1559756.
- [29] Patterson, T, Mitchell, N, Sifakis, E. Simulation of complex nonlinear elastic bodies using lattice deformers. ACM Trans Graph 2012;31(6):197:1–197:10. URL: <http://doi.acm.org/10.1145/2366145.2366216>. doi:10.1145/2366145.2366216.
- [30] Fan, Y, Litven, J, Pai, DK. Active volumetric musculoskeletal systems. ACM Trans Graph 2014;33(4):152:1–152:9. URL: <http://doi.acm.org/10.1145/2601097.2601215>. doi:10.1145/2601097.2601215.
- [31] Blemker, SS, Asakawa, DS, Gold, GE, Delp, SL. Image-based musculoskeletal modeling: Applications, advances, and future opportunities. Journal of magnetic resonance imaging 2007;25(2):441–451.
- [32] Gilles, B, Reveret, L, Pai, D. Creating and animating subject-specific anatomical models. Computer Graphics Forum 2010;29(8):2340–2351.
- [33] Schmid, J, Sandholm, A, Chung, F, Thalmann, D, Delingette, H, Magnenat-Thalmann, N. Musculoskeletal simulation model generation from mri data sets and motion capture data. In: Recent advances in the 3D Physiological Human. Springer; 2009, p. 3–19.
- [34] Ichim, AE, Kavan, L, Nimier-David, M, Pauly, M. Building and animating user-specific volumetric face rigs. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '16; Aire-la-Ville, Switzerland, Switzerland: Eurographics Association. ISBN 978-3-905674-61-3; 2016, p. 107–117. URL: <http://dl.acm.org/citation.cfm?id=2982818.2982834>.
- [35] Dicko, AH, Liu, T, Gilles, B, Kavan, L, Faure, F, Palombi, O, et al. Anatomy transfer. ACM Trans Graph 2013;32(6):188:1–188:8. URL: <http://doi.acm.org/10.1145/2508363.2508415>. doi:10.1145/2508363.2508415.
- [36] Thelen, DG. Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. J Biomech Eng 2003;125(1).
- [37] Kadlecik, P, Ichim, AE, Liu, T, Krivanek, J, Kavan, L. Reconstructing personalized anatomical models for physics-based body animation. ACM Trans Graph 2016;35(6).
- [38] Feng, A, Huang, Y, Xu, Y, Shapiro, A. Fast, automatic character animation pipelines. Computer Animation and Virtual Worlds 2014;25(1):3–16.
- [39] de Oliveira, JE, Giessler, P, Deserno, TM. Image registration methods for patient-specific virtual physiological human models. VCBM 2015;15:31–40.
- [40] Suzuki, K, Horiba, I, Sugie, N. Linear-time connected-component

- labeling based on sequential local operations. *Comput Vis Image Underst* 2003;89(1):1–23. URL: [http://dx.doi.org/10.1016/S1077-3142\(02\)00030-9](http://dx.doi.org/10.1016/S1077-3142(02)00030-9). doi:10.1016/S1077-3142(02)00030-9.
- [41] Jamin, C, Alliez, P, Yvinec, M, Boissonnat, JD. CGALmesh: a Generic Framework for Delaunay Mesh Generation. Research Report RR-8256; INRIA; 2014. URL: <https://hal.inria.fr/hal-00796052>.
- [42] Lewis, R, Nithiarasu, P, Seetharamu, K. Fundamentals of the Finite Element Method for Heat and Fluid Flow. Wiley; 2004. ISBN 9780470020814. URL: <http://books.google.es/books?id=dzCeawOSjRkC>.
- [43] Teschner, M, Heidelberger, B, Müller, M, Pomeranets, D, Gross, M. Optimized spatial hashing for collision detection of deformable objects. Proceedings of Vision, Modeling, Visualization VMV03 2003;:47–54.
- [44] Choi, KJ, Ko, HS. On-line motion retargetting. *Journal of Visualization and Computer Animation* 1999;11:223–235.
- [45] Müller, M, Gross, M. Interactive virtual materials. In: Proceedings of Graphics Interface 2004. GI '04; School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society. ISBN 1-56881-227-2; 2004, p. 239–246. URL: <http://dl.acm.org/citation.cfm?id=1006058.1006087>.
- [46] ZygoteBody,. Zygote media group. <https://www.zygotebody.com/>. 2018.
- [47] Anatomium, . 21st century solutions ltd. <http://www.anatomium.com/>. 2018.
- [48] CMUMCD, . Carnegie mellon university motion capture database. <http://mocap.cs.cmu.edu/>. 2018.
- [49] Tiede, U, Pommert, A, Pflessner, B, Richter, E, Riemer, M, Schiemann, T, et al. A high-resolution model of the inner organs based on the visible human data set 2002;5:212–212.
- [50] Voxel-Man, . Segmented inner organs (sio) group. <https://www.voxel-man.com/segmented-inner-organs-of-the-visible-human/>. 2018.