



Real-time animation of human characters' anatomy[☆]

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

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

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



ARTICLE INFO

Article history:

Received 16 March 2018

Revised 23 May 2018

Accepted 28 May 2018

Available online 6 June 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 Ltd. All rights reserved.

1. Introduction

Nowadays, trainers based on Virtual Reality (VR) are becoming more and more popular [1–3]. 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 [4–6].

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

In order to fulfil these requirements, we decided to follow a geometric-based method. Our approach computes a displacement field from the bones' movement. This field is used to transform the virtual tissues. In the following sections, we describe the techniques that have been developed or adapted to use the classical skeletal animation workflow to adjust a given pose to the internal patient tissues. The proposed technique computes most of the tasks during a pre-process stage, allowing to run the pose selection phase at interactive rates.

* This article was recommended for publication by Dr M Teschner.

* Corresponding author.

E-mail address: aaron.sujar@urjc.es (A. Sujar).

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

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

2. Previous work

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

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

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

Baran and Popović [20] propose an algorithm to automate both rigging and weighting stages. In their work, they adapt a previous generated virtual skeleton to a 3D model. Then, the influences of the virtual bones over the vertex of the surface mesh are computed using Laplace diffusion equation. This technique is constrained to B-Rep models and the surface mesh must completely enclose the virtual skeleton. The weighting phase of our animation pipeline extends this technique to deal with the internal anatomy of the model and to take into account the model's bone tissue for building the virtual skeleton. Baran and Popović [20] propose an algorithm to automate both rigging and weighting stages. In their work, they adapt a previous generated virtual skeleton to a 3D model. Then, the influences of the virtual bones over the vertex of the surface mesh are computed using Laplace diffusion equation. This technique is constrained to B-Rep models and the surface mesh must completely enclose the virtual skeleton. The weighting phase of our animation pipeline extends this technique to deal

with the internal anatomy of the model and to take into account the model's bone tissue for building the virtual skeleton. Jacobson et al. [21] improve the weighting phase to deal with point handles, virtual bones and cages. They minimize the Laplacian energy subject to the handles' constraints (among other constraints) to achieve high-order smoothness (see Eq. (1)):

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

where W_j are the weights of the handle j and m is the number of handles. Our approach is equivalent since we resolve Eq. (4) and our solutions make the Laplacian energy functional stationary. Therefore, our technique ensures high-order smoothness. Additionally, our formulation drives to a linear equation system, while this technique requires solving a sparse quadratic programming problem in order to impose the constraint set.

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

Geometrically-based algorithms do not only suffer from volume conservation problems. In [26], the authors propose a real-time geometrically based technique to deal with skin contacts and sliding. Furthermore, their solution does not require a weighting stage. They build a scalar field for each bone. The mesh vertices are characterized by a value on this scalar field. However, this method cannot be directly applied to our problem, since it does not take into account the bones' shape. Therefore, the osseous tissues will be affected by the smooth field function, leading to unrealistic deformations. Rumman and Fratarcangeli [27] follow a physically based approach to animate any articulated character. Since they discretize the character inside, their technique can be applied to animate the character internal anatomy. However, the algorithm computational cost requires a low-resolution volumetric mesh to run in real-time. In contrast, the model proposed in this paper can be applied to more complex meshes.

Leaving aside the classical skeletal animation, biomechanical and musculoskeletal models and physically based techniques have emerged in the last years. Most of the biomechanical models are manually created for a specific part of the human anatomy [28]. On one hand, Patterson et al. [29] efficiently simulate muscle behaviour. On the other hand, their technique requires a long and tedious manual stage that must be performed by an expert. Similarly, Fan et al. [30] describe a physically-based technique to simulate the muscle preserving their volume. The anatomical model can be generated automatically from medical images, but it needs the muscle shapes in its relaxed and final pose. Although, there are techniques that can obtain muscle-skeletal models from medical images [31–33], the cited techniques do not run at interactive rates and require detailed information of the patient. This kind of information is not always available. Furthermore, techniques usually focus on simulating the muscles and the skeleton, without

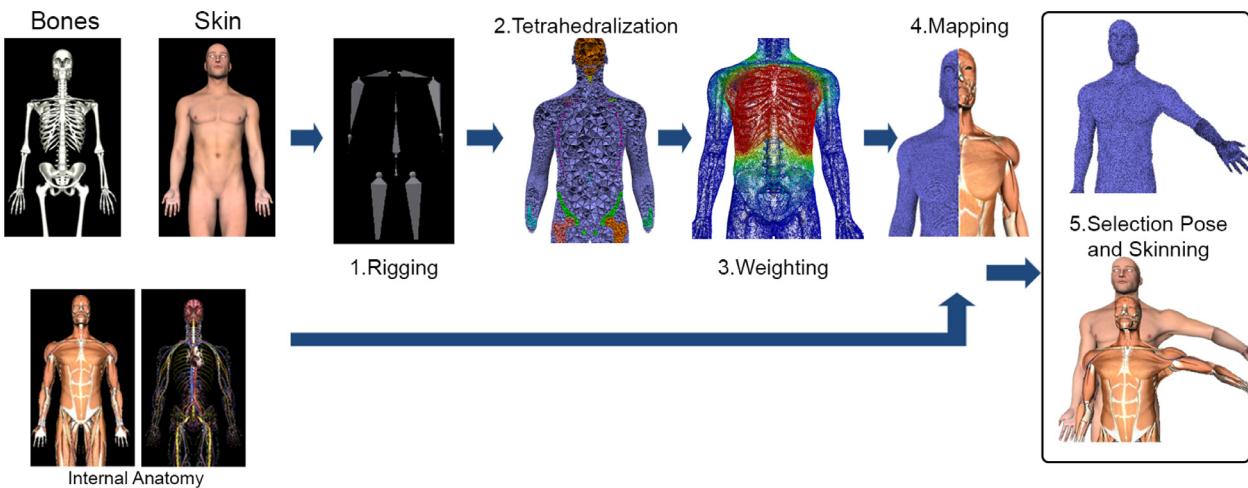


Fig. 1. Algorithm overview.

considering other anatomical structures. Ichim et al. [34] combine kinematic and dynamic methods to achieve real-time in the context of facial animation. Blendshapes are efficient and directly controllable, while physically-based models provide dynamic effects, secondary motions, collision response and incompressibility. However, blendshapes are not adequate to animate articulated characters and skeletal animation techniques are usually preferred.

It is worth to mention the work of Dicko et al. [35]. In this research, the authors detail a method for modelling the internal anatomy of a virtual character, transferring the inner tissues from a reference model to an arbitrary surface mesh. Their technique can infer the internal structures form MRI data or using a semi-automatic algorithm. The manuscript reports some issues, which will be solved in future work, when using MRI data. This work focuses on the modelling process and very few details are given on how to animate the resulting meshes. Despite this, they use their technique to transfer articulated systems but they neither explain how to compute the rig nor the mesh weights. To do so, their technique can be easily incorporated into the workflow described in this paper. Dicko et al. work can be also used to transfer the reference model muscle lines of action (described in [36]) to the final character for its biomechanical simulation. A full volumetric transfer will be required to obtain accurate muscles movements. Furthermore, physically based simulation methods are outside of this paper scope due to our real-time interaction requirements. Kadlec et al. [37] present a method to create anatomical models for its physically-based deformation. They improve Dicko et al. work proposing a fully automatic reconstruction while preserving realistic bone shapes and using multiple scans. In certain situations, the authors observe that the bones protrude through the muscles. Among other reasons, this artefact is caused by the use of multimaterial property of each body tetrahedron. Our technique does not suffer from this problem since we compute a smooth deformation field and since the bones tetrahedrons cannot embed any other structure. Additionally, this method is not meant to be used in real-time systems. However, it simulates inertial effects and secondary motions.

3. Animation pipeline

As it has been already mentioned, in this paper we propose a technique to adapt the internal and external anatomical information of a virtual character to any desired pose. With this purpose, we adapt and design a set of algorithms that establishes a skeletal animation pipeline to animate the internal tissues of a

virtual human character. In contrast to classical skeletal animation, the transformations of the virtual bones are not applied directly to the character's 3D model. Instead, the bones are used to compute a displacement field which is employed to transform the character anatomy. This approach can be applied not only to B-Rep models of the internal tissue but also to volumetric representations (Fig. 11). It is worth mentioning that most of the stages of our pipeline are fully automatic.

The system requirements are listed in Section 1. Our technique was designed to deal with incomplete anatomical descriptions, only the skin and bones must be identified. The animation pipeline is divided into the following stages (Fig. 1):

- **Rigging:** A predefined virtual skeleton is adjusted to the character anatomy. The algorithm uses the bones' models to compute the rotation centre of each joint of the virtual skeleton.
- **Tetrahedralization:** In this step, we generate an internal tetrahedral mesh from the skin and bone models. This volumetric mesh will be used to define a continuous displacement field associated with the movement of the bones.
- **Weighting:** This stage calculates automatically the influence of each bone on the tetrahedral mesh vertices.
- **Mapping:** The character's virtual tissues are assigned to the tetrahedrons of the volumetric mesh.
- **Pose selection and skinning:** In this step, the movements of the virtual skeleton are transferred to the tetrahedral mesh using a standard skinning algorithm such as LBS, DQS or CoR. Then, the tetrahedral mesh movements are applied to the tissues models. In our system, the user interactively selects the virtual character pose. Nevertheless, other techniques such as MoCap systems could be used instead.

3.1. Rigging

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

We plan to integrate our algorithm into a system to build virtual patient models for virtual reality medical simulators. Each virtual patient is built registering a generic virtual model into either

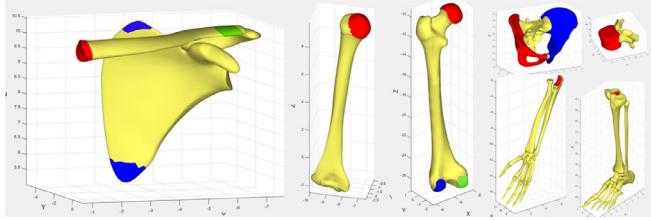


Fig. 2. 3D representations of a few bones. The areas coloured are used to compute the joint positions and orientations.

real patient data (Fig. 12) or average data from multiple real patients. For further details on this process, see [39].

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

3.2. Tetrahedralization

At this point, the algorithm discretizes the inside of the virtual model into a tetrahedral mesh. In the next step (Section 3.3), we compute each bone's influence over this mesh. Then, tetrahedral mesh is used to estimate a displacement field (Section 3.5). This field is transferred to the virtual model using the mapping calculated in Section 3.4. In order to control the discretisation process, the tetrahedral mesh is not directly generated from the virtual mesh. Instead, we generate a volumetric image from the skin and bone tissue models. In this 3D image, bones' voxels and characters' inside are labelled.

The size of the 3D image depends on the voxel and model's bounding box size. To keep the memory and time requirements low, we have selected a voxel size value in such a way that the 3D image size is always smaller than $250 \times 700 \times 120$.

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

Once the 3D image has been built, we use it to create a tetrahedral mesh [41]. In order to set up this algorithm, we must achieve a trade-off between accuracy and efficiency. We tuned the algorithm to increase the mesh resolution around the skin and bone boundaries. In our experiments, we kept the number of tetrahedrons below 3.5×10^6 and the number of vertices below 8×10^5 . Finally, the tetrahedral mesh is tagged using the osseous model (Fig. 4).

3.3. Weighting

The displacement field inside of the tetrahedral mesh is calculated by interpolating the displacement of each vertex. The movements of the tetrahedral model's vertices are associated with the

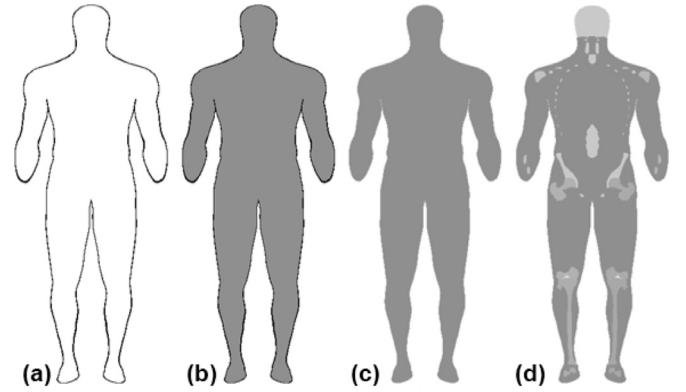


Fig. 3. Coronal slices of the volumetric image at different stages of the voxelization procedure.

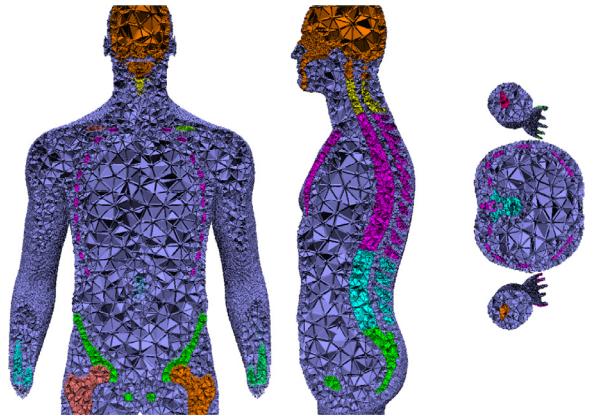


Fig. 4. Coronal, sagittal and axial slices showing the tetrahedralization phase results. The tetrahedrons labelled as bones are highlighted in different colours.

movement of one or more bones. This stage calculates how the movements of the bones influence the tetrahedral mesh vertices.

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

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

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

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

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

With the aim of computing the weights W_j of a bone j , we solve the steady case posed by Eq. (4). In order to resolve Eq. (4) for a bone j , we impose the following boundary conditions: we consider that the value W_j is 1 inside the tetrahedrons labelled as j ; and the value W_j is 0 inside the tetrahedrons labelled as k , where k is any bone not equal to j .

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

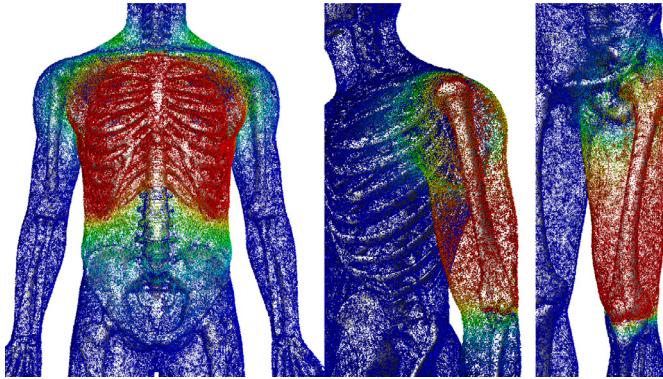


Fig. 5. Tetrahedral mesh with colour coding representing the weights of the chest, the humerus and the femur (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.).

The previous equation is discretized using the *Finite Element Method* (FEM) and the barycentric coordinates of the tetrahedron as shape functions (further information in [42]). Eq. (5) shows the final discretized diffusion equation:

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

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

It is worth mentioning that this formulation fulfils the constraints shown in Eqs. (2) and (3). First, the maximum and minimum values will only be reached on the bones and those values are 0 and 1. Second, if we consider $\mathbf{b} = \sum_{j=0}^n \mathbf{b}_j$ and $\mathbf{W} = \sum_{j=0}^n \mathbf{W}_j$, 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).

Fig. 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 $O(nm)$. 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 h 55 min in mapping 1277,325 vertices to 2584,115 tetrahedrons, where 4.6% of the vertices are outside the tetrahedral mesh. This time is reduced to 36.35 s when using Spatial Hashing.

3.5. Pose selection and skinning

In this stage, the user selects interactively a skeleton pose which is transferred to the virtual character model. Currently, our implementation allows animating the virtual model using direct kinematic, loading pre-recorded pose or animation. Techniques such as retargeting [44] or inverse kinematic can be directly applied to our system.

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

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

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

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

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

The displacement field of a point inside of a tetrahedron can be calculated by interpolating the displacement values of its vertices. We interpolate this field using the barycentric coordinates of each tetrahedron. Therefore, the computed displacement field is continuous but not differentiable inside the volumetric mesh and a constant transformation matrix can be calculated for each tetrahedron (see [45] for further information). The transformation matrix of a given tetrahedron is applied to the tissue vertices associated to that tetrahedron. Both tasks, the computation of the transformation matrices and how they are applied to the tissue vertices, are performed on the system graphics card.

4. Implementation details

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

5. Results

This section shows some results and explains the tests performed to evaluate our technique. First, we evaluated the pipeline computational performance. In these tests, we used four different anatomical models: the *ZygoteBody™* male (ZM) and female (ZF) models [46], the *Anatomium™* male model [47], as well as a bar model linking four bones to illustrate the various skinning techniques explored in the study. Fig. 6 summarizes the size of these models and their associated volumetric meshes. The

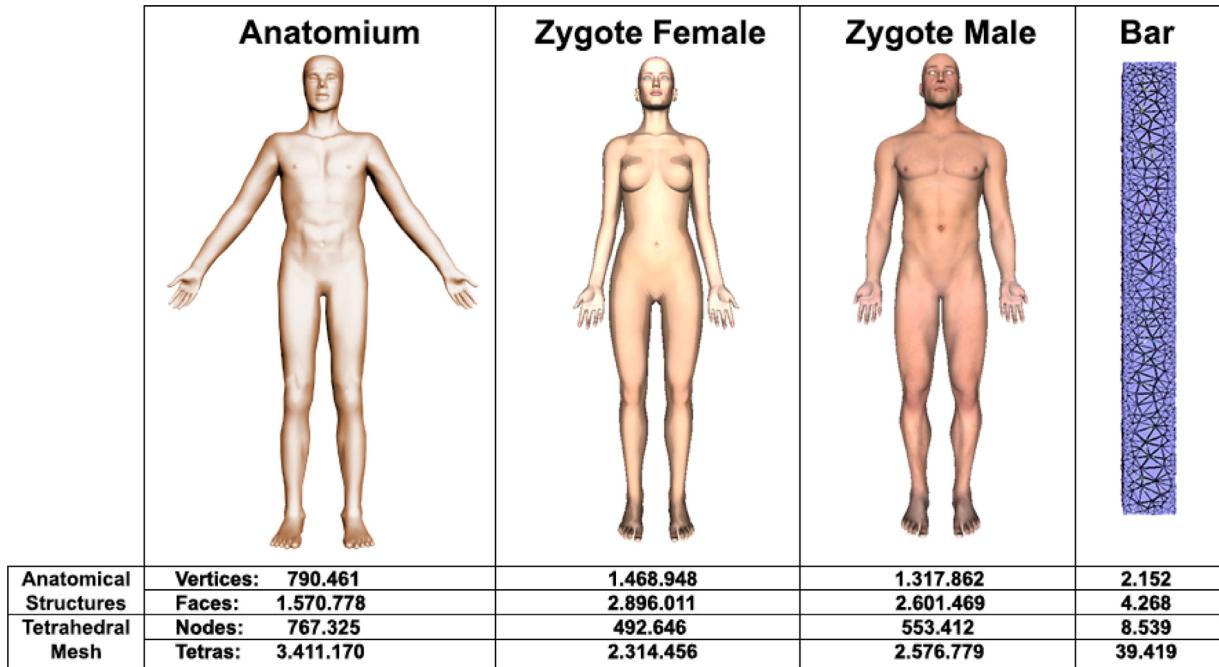


Fig. 6. Size of the models used in the tests.

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

Model	Rigging	Tetrahedralization	Weighting	Mapping	CoR preprocess
ZM	32,698	69,044	11,762	51,160	138,005
ZF	33,251	63,401	9171	71,635	208,886
A	31,891	120,465	23,318	44,521	115,214

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

animations used in the evaluation were obtained from Carnegie Mellon University Motion Capture Database [48]. All tests were run on an Intel®i7-4820K @ 3.7GHz PC with a NVIDIA®GeForce GTX 770 graphics card and 16GB of RAM. Table 1 shows the time spent in preprocessing the input data. The total time never exceeds seven minutes. This step is done only once for each model. The performance of the skinning step is shown in Table 2. The animation step includes the volumetric mesh skinning and transferring the tetrahedron deformations to the tissue models. Clearly, the skinning phase runs at interactive rates, even with our high detailed meshes. In addition, Table 2 compares the performance between three different skinning methods. We would like to stress that the three skinning techniques have almost the same performance. In Figs. 7 and 8, we compare the results obtained with several skinning techniques. Fig. 7 depicts how the bones' movements deform the tetrahedral mesh. Fig. 8 illustrates how the displacement field of the tetrahedral mesh is transferred to the anatomical structures. The LBS volume loss and the DQS volume gain are appreciable in both images. In these examples, CoR solves most of the artefacts introduced by LBS and DQS.

However, CoR cannot ensure volume conservation as physically based techniques do. We compared CoR with a physically-based

model. Since we do not have a proper characterization of the tissues' mechanical properties, the goal of the implementation is to guarantee volume conservation. We use a co-rotational FEM formulation to solve the steady-state problem for a linear, isotropic and homogeneous material. Besides, the deformations are measured using the Cauchy strain tensor. The boundary conditions, needed to solve the steady-state problem, are given by the positions of the vertices labelled as bones. The co-rotational formulation calculates the internal forces caused by the deformations in a non-rotated configuration. Then, the internal forces are rotated again into the final configuration [45]. The algorithm needs to compute the element rotations in the final configuration. For this purpose, the solution is refined iteratively. The elastic used model can be tuned with two parameters: the Poisson ratio and the Young module. The Poisson ratio controls the volume conservation and it should take a value close to 0.5 (the real value has to be lower to ensure numeric stability). Since the material is homogenous, this value has no impact on the outcome. We chose the Young module to improve the stability of the system. We tested different values and we selected the one that conditioned the system matrix better for inversion (checking the matrix condition number). Fig. 9 illustrates how the implemented FEM-based model solves some volume issues. However, CoR works well in most scenarios and we believe that users would not prefer one over the other. To test this hypothesis, we have run a user study. A total of 16 subjects participate in our study (2 females, 14 males; between the ages of 20–52; 13 of them are computer graphics professionals). In our experiments, participants were asked to rate the realism of the deformation presented in 22 static images, using a Likert from 1 to 8. The images displayed 6 poses and several models and internal tissues. Half of them were created with FEM and the others with

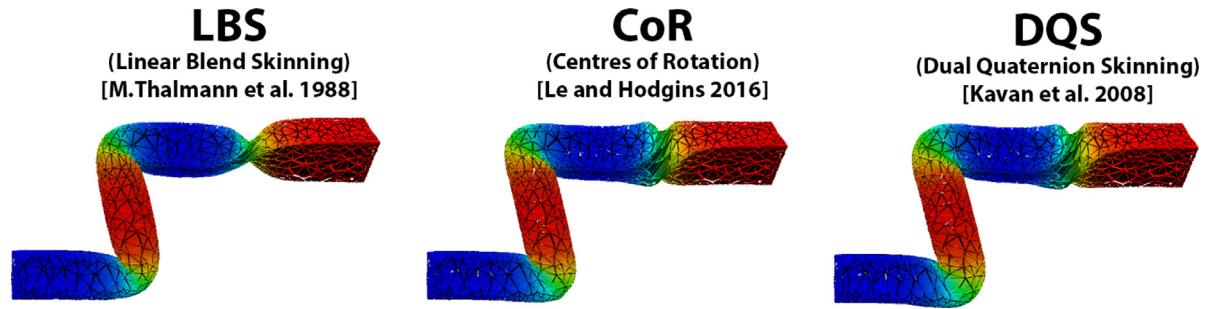


Fig. 7. Bar deformation for 3 skinning techniques. Joint rotations: 100 bending, -100 bending, 135 twisting.

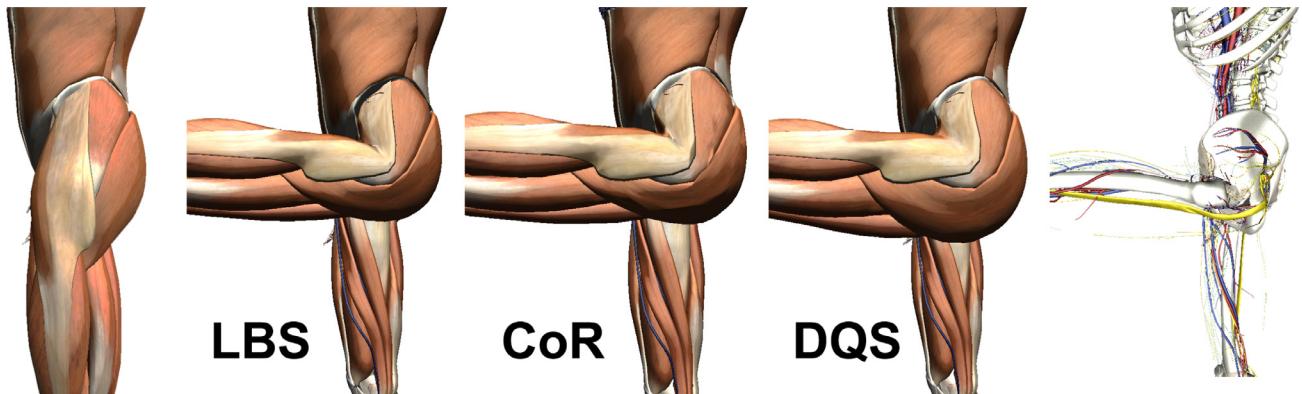


Fig. 8. Left: muscles of the model in a rest position; three middles images: muscles of the model deformed by leg bending for 3 skinning technique (LBS lost volume at the top of the thigh, DQS increases the volume at gluteus, CoR prevents these artefacts); right: other tissues of the model with the same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

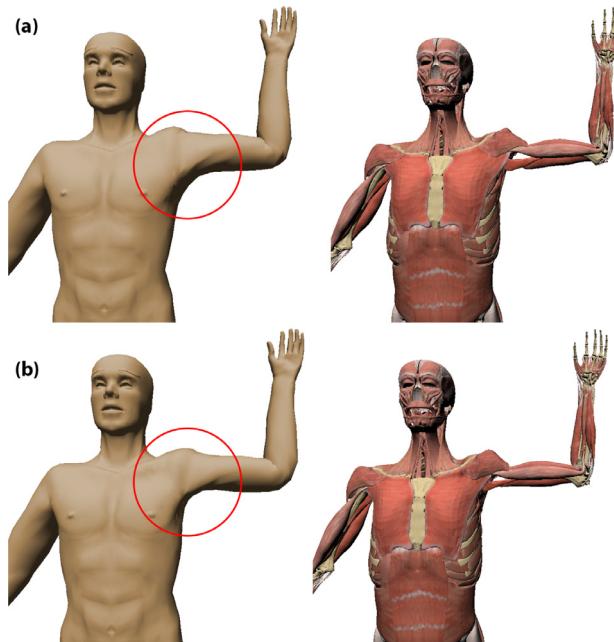


Fig. 9. Comparison between CoR (a) and a physically-based model (b) designed to preserve volume. CoR slightly increases the volume in the armpit region.

CoR. We showed a reference image of the tissues in its rest pose together with the deformations generated by FEM and CoR. The reference images were always displayed first and then the order of the FEM and CoR deformations were randomized to avoid bias. The online questionnaire designed for this test can be found at the URL: <https://goo.gl/WprGP6>. The assumption of homoscedasticity holds but normality does not. Therefore, we compared the

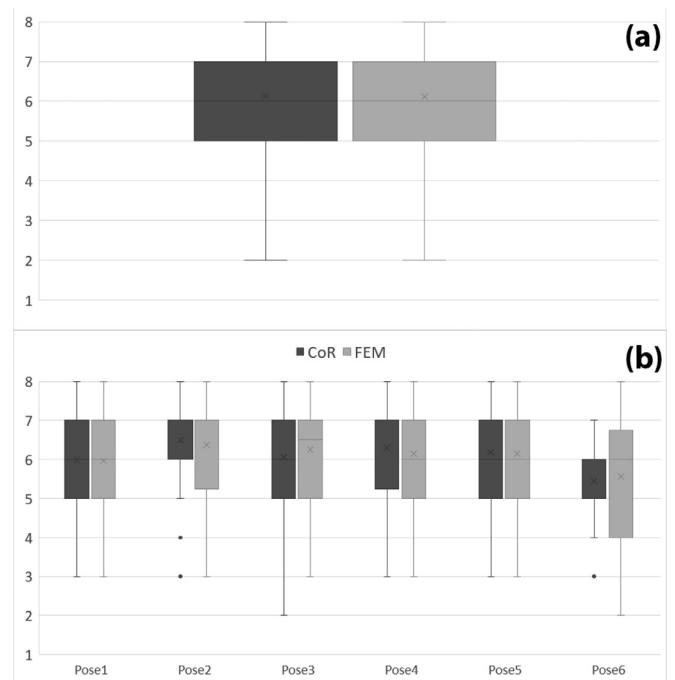


Fig. 10. Boxplot comparing the results obtained in the user study. Image (a) summarizes the global results, while the image (b) compares the results obtained for the different poses used in our experiments.

results obtained for FEM and CoR using the non-parametric Wilcoxon signed rank for matched samples. The result of this test confirms that the differences between both techniques are not significant (p -value > 0.9). Fig. 10 summarizes the obtained results.

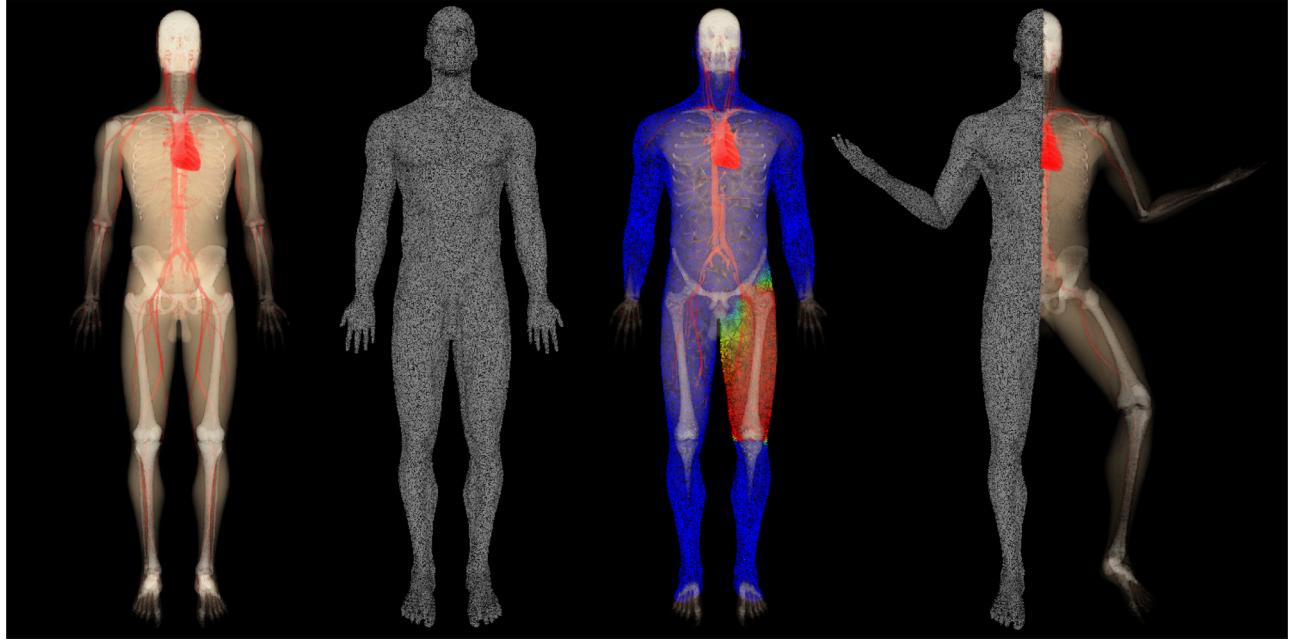


Fig. 11. Overview of the process of animating a volumetric character. From left to right: (1) original volumetric model, (2) tetrahedral mesh, (3) tetrahedral mesh with colour coding representing the weights of the femur as illustration (high weights towards red, low towards blue) and (4) tetrahedral mesh and volumetric data in a user-defined pose (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

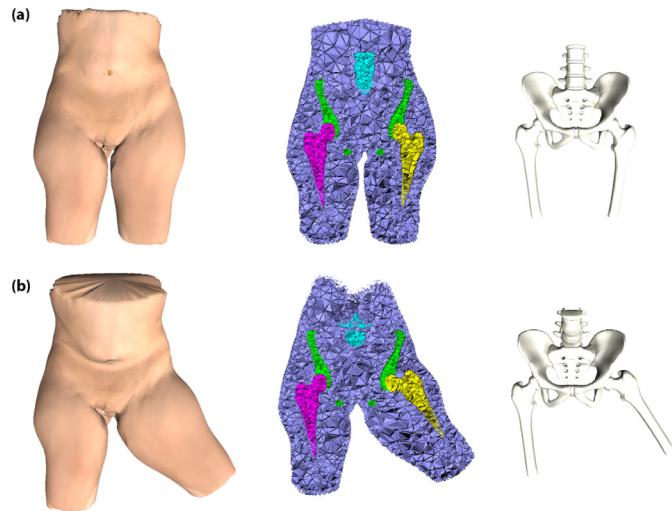


Fig. 12. (a) shows a virtual patient model built from real patient data in its rest position. The positioning of the model can be performed using our algorithm (b).

Fig. 11 shows how the proposed pipeline is applied to a voxel-based representation of a virtual character. Instead of transferring the weights of the tetrahedral mesh nodes to the embedded tissue vertices, our technique computes a displacement field inside the virtual character and, consequently, our pipeline can be applied to both B-Rep and volumetric models. In the volumetric case, we determine the pixels inside each tetrahedron in the deformed configuration and then, they are mapped back to the non-deformed configuration using the inverse of the displacement field. For the first step, we iterate over each tetrahedron's bounding box and we use the barycentric coordinates to identify the pixels inside the tetrahedron. This process can be easily parallelized on modern GPUs, allowing real-time rendering.

Our algorithm can be used to transform models generated from real patient data from the rest position to any desired position.

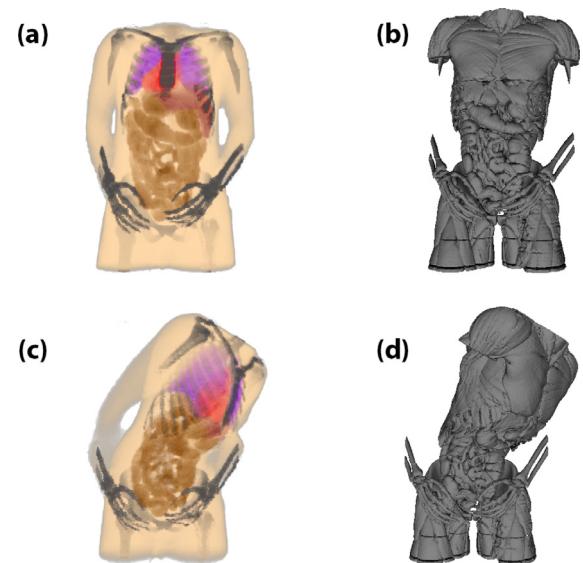


Fig. 13. Our technique can be applied to the volumetric (a) and (c) and surface-based representations (b) and (d). Images (a) and (b) show the model inner organs in its rest position. Images (c) and (d) illustrate the volumetric and the B-rep models transformed into a given pose.

Fig. 12(a) displays: the skin (left), the pre-processed tetrahedral mesh (centre) and the bones (right) of a virtual character in rest position; while **Fig. 12(b)** shows the same three structures in its deformed configuration. This example has been built from real medical images. The registration algorithm [39] could not provide a full anatomical description of the patient since only local images were available. Many medical procedures are performed in a local area. Therefore, not having a full anatomical description of the virtual patient should not be a major issue. In brief, **Fig. 12** illustrates how our technique works with incomplete information.

Fig. 13 exemplifies how our technique can be applied to the volumetric and surface-based representations of the same character.

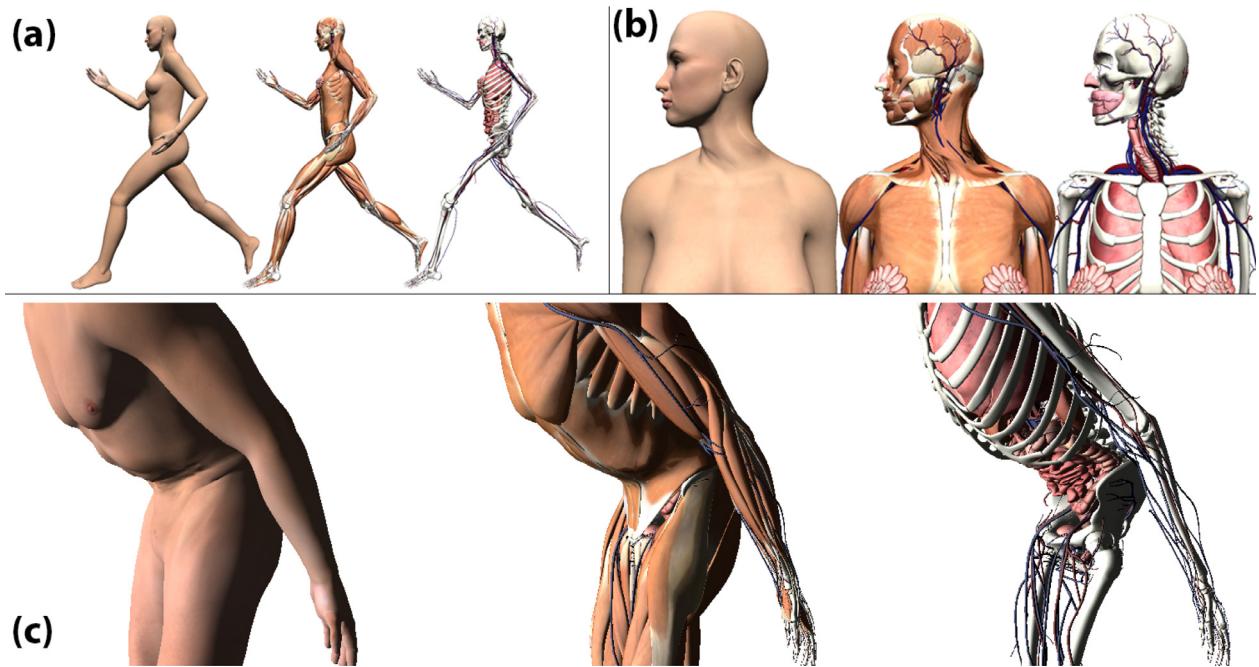


Fig. 14. Several results obtained with CoR. (a) A key frame of a running cycle applied to ZF. (b) A neck bend applied to ZF. (c) A belly bend applied to ZM.

For this example, we used the Voxel-Man's Segmented Inner Organ model [49,50], which is a segmentation of the Visible Human dataset. Finally, additional results for the ZM and ZF models are shown in Fig. 14.

6. Conclusions and future work

In recent years, virtual reality is proving its potential in the medical training field. One of the main advantages of those tools is to allow the trainees to face a large variety of scenarios. Therefore, the virtual patient database is a key component of these systems. In this context, our system adapts a virtual patient to the pose required in a given medical procedure. Recently, musculoskeletal simulation has significantly advanced in the computer graphics field. These techniques usually require an accurate description of the patient tissues, which is not always available. This paper presents a pipeline to transform any available patient anatomy to a desired pose. This technique automates the skeletal animation paradigm to deal with internal tissues of the model. Our algorithm works with incomplete anatomical models and does not need a mechanical description of the tissue behaviours. The technique is flexible enough to deal with B-Reps and volumetric models as long as the osseous tissue is properly labelled.

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

Additionally, the skinning phase of our pipeline is very fast, particularly in comparison with physically-based models. On the one hand, accurate models rarely achieve interactive frame rates. On the other hand, less accurate models such as point base dynamics must work with less complex 3D models to ensure real-time animations [27].

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

Acknowledgments

The research leading to these results has received funding from the following entities: the Spanish Goverment [Cajal Blue Brain Project, Spanish partner of the Blue Brain Project initiative from EPFL; and grants TIN2014-57481-C2-1-R, TIN2017-83132-C2-1-R and FPU15/05747] and European Commission [under grant agreements: HBP - Human Brain Project - FET Flagship HBP604102; and RaSimAS: Regional Anaesthesia Simulator and Assistant - FP7-ICT 610425]. The mocap data used in this work has been downloaded from Carnegie Mellon University. Some examples were based on the Visible Human male data set (National Library of Medicine) and the Segmented Inner Organs (Voxel-Man).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.cag.2018.05.025](https://doi.org/10.1016/j.cag.2018.05.025).

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. *Int J Comput Assist Radiol Surg* 2009;4(6):561–70.
- [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). *J Bone Joint Surg* 2014;96(13):e112.
- [3] Deserno TM, de Oliveira JE, Grottko O. Regional anaesthesia simulator and assistant (RASIMAS): medical image processing supporting anaesthesiologists in training and performance of local blocks. In: Proceedings of the 2015 IEEE twenty-eighth international symposium on computer-based medical systems (CBMS). IEEE; 2015. p. 348–51.
- [4] Seymour NE. Vr to or: a review of the evidence that virtual reality simulation improves operating room performance. *World J Surg* 2008;32(2):182–8.
- [5] Dawe SR, Windsor JA, Broeders JA, Cregan PC, Hewett PJ, Maddern GJ. A systematic review of surgical skills transfer after simulation-based training: laparoscopic cholecystectomy and endoscopy. *Annal Surg* 2014;259(2):236–48.

- [6] Buckley CE, Kavanagh DO, Traynor O, Neary PC. Is the skillset obtained in surgical simulation transferable to the operating theatre? *Am J Surg* 2014;207(1):146–57.
- [7] Willaert W, Aggarwal R, Van Herzele I, Cheshire N, Vermassen F. Recent advancements in medical simulation: patient-specific virtual reality simulation. *World J Surg* 2012;36(7):1703–12. 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. *J Biomech* 2013;46(2):217–28.
- [9] Casafranca JJ, Sjar A, Garca M. An interactive algorithm for virtual patient positioning. In: Sbert M, Lopez-Moreno J, editors. Proceedings of the 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. *Med Eng Phys* 2013;35(12):1703–12.
- [11] Le BH, Hodgins JK. Real-time skeletal skinning with optimized centers of rotation. *ACM Trans Graph* 2016;35(4):37.
- [12] Pan J, Chen L, Yang Y, Qin H. Automatic skinning and weight retargeting of articulated characters using extended position-based dynamics. *The Visual Computer*. Springer; 2017. p. 1–13. doi:10.1007/s00371-017-1413-6.
- [13] Jacobson A. Part II: Automatic skinning via constrained energy optimization. *SIGGRAPH Course* 2014:1–28.
- [14] Pantuwong N, Sugimoto M. A novel template-based automatic rigging algorithm for articulated-character animation. *Comput Anim Virt Worlds* 2012;23(2):125–41. doi:10.1002/cav.1429.
- [15] Bharaj G, Thormählen T, Seidel H-P, Theobalt C. Automatically rigging multi-component characters. In: Proceedings of the computer graphics forum, Vol. 31. Wiley Online Library; 2012. p. 755–64.
- [16] Feng A, Casas D, Shapiro A. Avatar reshaping and automatic rigging using a deformable model. In: Proceedings of the eighth ACM SIGGRAPH conference on motion in games. ACM; 2015. p. 57–64.
- [17] Avril Q, Ribet S, Ghaffourzadeh D, Dionne O, Ramachandran S, de Lasa M, et al. Animation setup transfer for 3d characters. In: Proceedings of the computer graphics forum, Vol. 35. Wiley Online Library; 2016. p. 115–26.
- [18] Borosán P, Jin M, DeCarlo D, Gingold Y, Nealen A. RIGMESH: automatic rigging for part-based shape modeling and deformation. *ACM Trans Graph* 2012;31(6):198.
- [19] Huang C-H, Boyer E, Ilic S. Robust human body shape and pose tracking. In: Proceedings of the 2013 international conference on 3DTV-conference. IEEE; 2013. p. 287–94.
- [20] Baran I, Popović J. Automatic rigging and animation of 3d characters. *ACM Trans Graph* 2007;26(3). 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. 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, Canada: Canadian Information Processing Society; 1988. p. 26–33. <http://dl.acm.org/citation.cfm?id=102313.102317>.
- [23] Lewis JP, Cordiner M, Fong N. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: Proceedings of the twenty-seventh annual conference on computer graphics and interactive techniques SIGGRAPH'00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.; 2000. p. 165–72. ISBN 1-58113-208-5. 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. doi:10.1145/1409625.1409627.
- [25] Rumman NA, Fratcangeli M. State of the art in skinning techniques for articulated deformable characters. In: Proceedings of the eleventh international conference on computer graphics theory and application, GRAPP 2016; Part of the eleventh 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 M-P. Robust ISO-surface tracking for interactive character skinning. *ACM Trans Graph* 2014;33(6):189:1–189:11. doi:10.1145/2661229.2661264.
- [27] Abu Rumman N, Fratcangeli M. Position-based skinning for soft articulated characters. In: Proceedings of the computer graphics forum, Vol. 34. Wiley Online Library; 2015. p. 240–50.
- [28] Lee S-H, Sifakis E, Terzopoulos D. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans Graph* 2009;28(4):99:1–99:17. 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. 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. doi:10.1145/2601097.2601215.
- [31] Blehmke SS, Asakawa DS, Gold GE, Delp SL. Image-based musculoskeletal modeling: applications, advances, and future opportunities. *J Magn Reson Imaging* 2007;25(2):441–51.
- [32] Gilles B, Reveret L, Pai D. Creating and animating subject-specific anatomical models. In: Proceedings of the computer graphics forum 2010, Vol. 29(28); 2010. p. 2340–51.
- [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 A-E, 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; 2016. p. 107–17. ISBN 978-3-905674-61-3. <http://dl.acm.org/citation.cfm?id=2982818.2982834>.
- [35] Dicko A-H, Liu T, Gilles B, Kavan L, Faure F, Palombi O, et al. Anatomy transfer. *ACM Trans Graph* 2013;32(6):188:1–188:8. doi:10.1145/2508363.2508415.
- [36] Thelen DG. Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *J Biomed Eng* 2003;125(1).
- [37] Kadlecik P, Ichim A-E, 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. *Comput Anim Virt Worlds* 2014;25(1):3–16.
- [39] de Oliveira JE, Giessler P, Deserno TM. Image registration methods for patient-specific virtual physiological human models. *Proceedings of the visual computing for biology and medicine, 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. doi:10.1016/S1077-3142(02)00030-9.
- [41] Jamin C, Alliez P, Yvinec M, Boissonnat J-D. CGALmesh: a generic framework for delaunay mesh generation. Research Report RR-8256. INRIA; 2014. <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. <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. In: Proceedings of vision, modeling, visualization VMV03; 2003. p. 47–54.
- [44] Choi K-J, Ko H-S. On-line motion retargetting. *J Vis Comput Anim* 1999;11:223–35.
- [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; 2004. p. 239–46. ISBN 1-56881-227-2. <http://dl.acm.org/citation.cfm?id=1006058.1006087>.
- [46] ZygoteBody. Zygote Media Group. 2018. <https://www.zygotebody.com>.
- [47] Anatomium. 21st Century Solutions Ltd. 2018. <http://www.anatomium.com>.
- [48] CMUMCD. Carnegie mellon university motion capture database. 2018. <http://mocap.cs.cmu.edu/>.
- [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. *J Visual* 2002;5(3):212. doi:10.1007/BF03182328.
- [50] Voxel-Man. Segmented inner organs (SIO) group. 2018. <https://www voxel-man.com/segmented-inner-organs-of-the-visible-human>.