

# Chapter 6

## Modeling and Simulating Virtual Anatomical Humans

Forough MadehKhaksar, Zhiping Luo, Nicolas Pronost and Arjan Egges

### 6.1 Introduction

This chapter presents human musculoskeletal modeling and simulation as a challenging field that lies between biomechanics and computer animation. One of the main goals of computer animation research is to develop algorithms and systems that produce plausible motion. On the other hand, the main challenge of biomechanics is investigating the influence of internal and external forces and stimulators on the biological behavior of different tissues. By combining the two approaches, it is possible to produce real-time animation of a user's avatar under different activities and to simulate the related biological effects of that activity. In this chapter we review the challenges and issues of modeling, simulating, and animating virtual anatomical humans, as well as an overview of the benefits and limitations of such systems.

The main advantage of using virtual anatomical models is the capability to study the biomechanical effects of a variety of different activities that real humans perform. The virtual world allows us to visualize these activities. Because of the use of virtual models and simulations, these studies can be done without spending a lot of money on measuring equipment, and without presenting any hazard to human subjects. The biggest challenge in achieving this lies in the conflict between the real-time nature of computer animation and the time-consuming computations required in biomechanics. This chapter serves as a starting point for research into addressing this conflict, by providing an overview of the differences and similarities between the two fields.

Modeling and simulating the human body has been extensively researched in the last decades. For example, recently, large consortia investigated new technologies around virtual representations of the human body [1] and the multi-scale biological data visualization of physiological human articulation [2]. Individual labs also take

---

F. MadehKhaksar · Z. Luo · N. Pronost (✉) · A. Egges  
Virtual Human Technology Lab, Utrecht University, Princetonplein 5,  
3584 CC Utrecht, The Netherlands  
e-mail: nicolas.pronost@uu.nl; url:<http://www.vhtlab.nl>

part in that research such as the Riverside Graphics Lab of University of California developing an anatomically inspired torso simulator for breathing and laughing [3, 4]. The Center of Computer Graphics and Visualization of the University of West Bohemia developed a musculoskeletal model, where bones and muscles are represented by their triangulated surfaces obtained from MRI data, and adopted the action-lines muscle description and FEM musculoskeletal models for the simulation to provide clinicians with a tool fast enough to be suitable for the clinical practice but with enhanced accuracy. The Computer Graphics and Vision Laboratory at UCLA also participated by developing a comprehensive biomechanical model of the human upper body with modeling and controlling nearly all relevant articular bones and muscles, as well as simulating the physics-based deformations of the soft tissues [5].

This chapter is structured as follows. First, we will discuss how to create an accurate virtual anatomical *model* of a human, based on real-world information from animation and biomechanics. Then, we discuss the various techniques that allow us to *simulate* the various aspects of this model such as muscular actuation and deformable objects. Finally, we will discuss a number of practical issues that are of use to anyone who wants to *implement* such a simulation.

## 6.2 Modeling Virtual Anatomical Humans

The ultimate goal of modeling virtual humans is to develop a generation of digital humans that comprise of realistic human models including anatomy, biomechanics, physiology, and intelligence. In this section we present the state-of-the-art approaches regarding the modeling of physical humans in the fields of animation and biomechanics. First the human biomechanics is introduced, then we present how the different anatomical entities are usually modeled, and finally we indicate how acquisition and processing techniques are used to generate such models.

### 6.2.1 *The Human Biomechanics*

Human motion is a complex process driven by various biological triggers. These triggers lead to the exertion of mechanical forces that in the end are visualized as movements by the human body. The role of human biomechanics is to study the structure as well as function of humans through the lens of mechanics. In his classic book, David Winter states that human biomechanics is an interdisciplinary field that describes, analyses, and assesses human movement [6]. Human biomechanics is also intertwined with other fields of movement science such as neurophysiology, exercise physiology, and anatomy.

### 6.2.1.1 Statics, Dynamics and Deformation

The Newton's first law of motion is used in a branch of mechanics known as *statics*. In mechanics, statics refers to the analysis of rigid bodies (solids) that are in a state of equilibrium, i.e. in a state where a rigid body's acceleration is zero which also means that both net forces (for translational movement) and net moments (for rotational movement) are also zero. Studying systems in states of equilibrium are useful to understand what forces are in play or should be taken into consideration. According to Nordin and Frankel [7], statics is generally used in biomechanics to investigate the unknowns in problems that involve the magnitudes of joint reaction forces and muscle tensions.

*Dynamics*, also commonly referred to as *physics*, takes the simulation one step further in that it studies how forces and torques causes the state of motion of an object to change, i.e. how a physical system changes over time with respect to applied loads. When forces are exerted on an object, three things can happen, either the object exhibits a change in linear and/or rotational motion, or in the case of equilibrium, the object experiences a localized shape change over time, a *deformation*. Forces that can be ascribed when an object gets deformed are:

1. *Normal or axial:*

- *Tensile* forces (when an object elongates)
- *Compressive* forces (when an object shrinks)

2. *Tangential:*

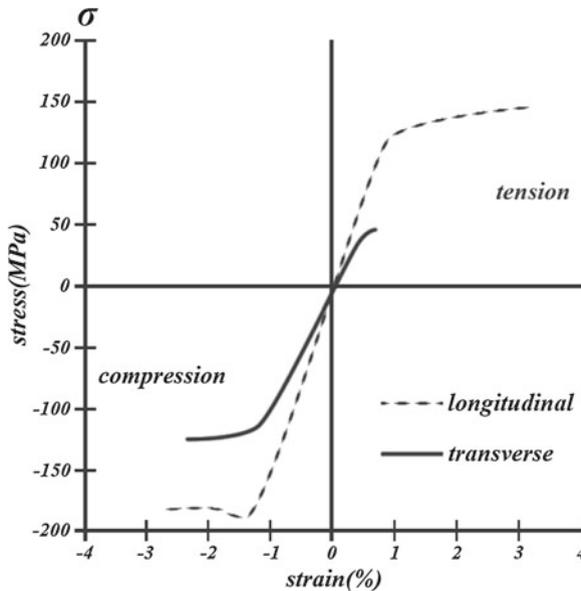
- *Shear* forces (which in some cases result in bending or twisting of an object)

Because deformation of an object depends also on its material properties, having a quantity that defines the average force per unit area is helpful in approximating solutions for the analysis of intrinsic properties of the object under load. This concept known as *stress* is the amount of applied force divided by the area it is applied on. Similarly, two forms of stress are used, namely the *normal* stress  $\sigma$  and *shear* stress  $\tau$ . Normal or axial stress can be calculated when the exerted force lies orthogonal to the cross-affected area under consideration. In cases where the exerted force lies tangent or parallel to the affected area, shear stress can be calculated. Under the assumption that the forces are uniformly distributed along the area and therefore consist of a simple stress pattern, the stress equation can be formally written as:

$$\tau, \sigma = \frac{\mathbf{F}}{A}$$

where  $\mathbf{F}$  is the force and  $A$  the cross-sectional area.

Besides stress, there is also the concept of *strain*, denoted  $\epsilon$ , that quantifies the normalized amount of deformation after an initial configuration, i.e. the amount of displacement of the intrinsic properties of an object from its original length to its current length. Similarly to stress it has also two basic forms, which are *normal* and



**Fig. 6.1** Stress–strain diagram of a typical human cortical bone with strain percentage along the horizontal axis and stress in Mega Pascal along the vertical axis. When forces are exerted in the longitudinal direction instead of the transverse, the material can withstand more stress before it reaches the point of failure, characterized by the longer dotted graph. Also apparent is the toughness of the cortical bone material during compression when compared to tension. Inspired by Ref. [8]

*shear* strain. Normal strain occurs when the displacement happens along the material fibers and is formally written as:

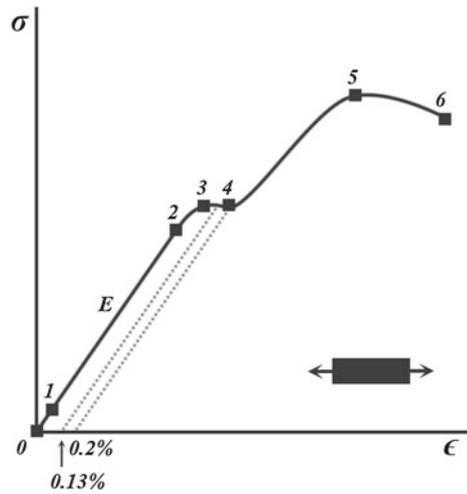
$$\epsilon = \frac{l_1 - l_0}{l_0} = \frac{\Delta l}{l_0}$$

where  $\Delta l$  is the displacement in length and  $l_0$  is the initial length before any deformation occurred. A negative outcome represents a compressive strain, while a positive outcome represents a tensile strain. For shear strains, denoted  $\gamma$ , deformations made by shear forces are measured.  $\gamma$  represents the tangent between relative displacements of shear forces.

Studying the relationship between stress and strain provides insight into the intrinsic behavior of an object under different load conditions. Figure 6.1 shows a typical stress–strain diagram of the human cortical bone.

When an object experiences tensile or compressive forces in one direction it can contract or expand in other directions that lie orthogonal to the direction of the applied forces. This characteristic is known as the *Poisson effect*. This effect can be measured by taking the ratio between the change in normal strain with the change in shear strain and is referred to as *Poisson's ratio*, denoted  $\nu$ . Formally it is written as:

**Fig. 6.2** Engineering stress–strain diagram for illustration purposes, showing key characteristics of a hypothetical object that experienced different tensile forces



$$\nu = - \frac{d\epsilon_n}{d\epsilon_s}$$

where  $\epsilon_n$  and  $\epsilon_s$  denote respectively the normal and shear strains.

### 6.2.1.2 Elasticity and Plasticity

A deformed object is considered *elastic* when it deforms back to its original configuration after the applied load is removed and *plastic* when at least parts of the deformation are permanent. The graph in Fig. 6.2 illustrates a stress–strain curve of a hypothetical object that experienced different tensile forces. The numbers along the curve represent several known characteristics that play a key role in the analysis of the material.

Point 0 represents the initial configuration, when no forces are exerted on the object. Point 1 represents the *true elastic limit*. Between point 0 and 1 no dislocations are visible on the atomic level or molecular scale. Point 2 represents the *proportionality limit*, after this point the elasticity of the material does not behave linearly. The segment of the curve between point 0 and 2 is said to be linear-elastic or Hookean implying a conformance to *Hooke’s law*. The principle of Hooke’s law, which in mechanics is used for models such as springs, can be described in terms of material functions of stress and strain for both normal and tangential forces:

$$\sigma = E\epsilon$$

$$\tau = G\gamma$$

where  $E$  denotes the elastic modulus or *Young’s modulus* for normal forces and  $G$  denotes the shear modulus or *modulus of rigidity* for tangential forces.

**Table 6.1** Mechanical properties of three biological materials: compact bone, trabecular bone, and tendon-ligament. Adapted from Ref. [9]

Material	Ultimate strength (MPa)	Modulus (GPa)	Elongation (%)
Compact bone	100–150	10–15	1–3
Trabecular bone	8–50	–	2–4
Tendon, ligament	20–35	2–4	10–25

Young's modulus is representative for the stiffness of a material, the higher the elastic slope, the stiffer the material. Point 3 represents the *elastic limit*, also known as the yield strength or yield point. The segment between point 2 and 3 defines the complete elastic behavior of the material, removing any load that falls within this segment would allow the material to deform back to its original configuration. After this point, the material behaves plastic and any further deformation is considered permanent. In this example, the load that is applied on the object is removed at the point on the curve between 3 and 4. The dotted line represents the linear regression of the elastic deformation with the same Young's or rigidity modulus that eventually intersects the strain axis. The intersection point on this axis is called the *plastic strain* that quantifies the amount of permanent strain, in this example 0.13%. Point 4 represents the *offset yield point* that can be derived using the offset method. The offset method is used for cases where it is difficult to measure the exact point when the material yields. Similarly to the previous derivation of plastic strain, the method works by drawing a line, which starts generally with a strain offset of 0.1–0.2%, and progresses in parallel with the linear-elastic segment of the curve. The intersecting point 4 between this line and the original curve is considered the offset yield point. Point 5 on the curve represents the ultimate tensile strength of the material followed by point 6 that represents the rupture point of the material; when the object finally fractures. An overview of mechanical properties for three biological materials is shown in Table 6.1.

## 6.2.2 The Musculoskeletal System

### 6.2.2.1 The Bones and Joints

The human body is a complex structure composed of a variety of interacting anatomical entities. Among them, the skeletal system includes the bones, cartilages, ligaments, and tendons. The average adult skeleton has 206 bones although actual number of bones slightly varies from person to person. The skeleton is usually divided into the axial and appendicular skeletons. The axial skeleton forms the upright axis of the body. It is divided into the skull, auditory ossicles, hyoid bone, vertebral column, and thoracic cage, or rib cage. The axial skeleton protects the brain, the spinal cord, and the vital organs housed within the thorax. The appendicular skeleton consists of the bones of the upper and lower limbs and the girdles by which they are attached to

the body. The term girdle means a belt or a zone and refers to the two zones, pectoral and pelvic, where the limbs are attached to the body.

Bones are connected to each other by *articulations*, also referred to as *joint*. A joint is then the union between two or more bones. The joints are located at the bones extremities, where the participant bones are in contact with each other and relative motion may occur. Many joints allow only limited relative movement, and some even allow no apparent movement. The structure of a given joint is directly correlated with its *degrees of movement*, or *degrees of freedom* (DOF). To model a human joint, a perfect model should consider the mechanics of the joint structure and its biomechanical interactions. It should not only visually react as the real joint but also include the real biomechanical behavior and biological properties. Biomechanical systems share many properties with mechanically engineered systems, and it is possible to employ mechanical engineering simulation software to investigate the mechanical behavior of diverse biological mechanisms. But as a challenge in biomechanical modeling of human articulation, unlike their man-made counterparts, biomechanisms rarely exhibit the simple, uncoupled, pure-axial motion that is engineered into mechanical joints such as sliders, pins, and ball-and-socket joints [10]. To ensure valid biological simulation and modeling, models must be accurate and subjected to sensitivity and design optimization analyses, demanding vast amounts of computation [10]. Most of existing models in computer graphics are far from being perfect in that sense. In the other hand, they do not need to be as little focus is given to biomedical application. Moreover, they should be simple enough to work in real-time. Most of the existing models in biomechanics focus on biomechanical accuracy and computationally expensive calculations. Accuracy, efficiency and visualization are generally competing and compromising tasks in biomechanics and computer graphics.

In addition, joint modeling in computer graphics is accompanied with a trade-off between generic but simplified models and specific but accurate models. It is possible to find works that consider a large range of fundamental types of the human articulation system [11], but they still behave differently from real joints. Joints are typically modeled as rotational constraints around one or more fixed axes and/or translational constraints along one or more of the three Cartesian directions. While in real joints the motion axes can change position by rotation [12]. Wilhelms [13] introduced a general method that simplifies the skeleton and where the geometry of bones is composed of three ellipses. A tree hierarchy is then utilized in which every group of bones is considered a segment of the body. The root segment is connected to the world reference frame by a free (six DOF) joint and the other segments connected to each other by constrained joints. Each segment has a parent segment and zero or more child segments. In addition, each segment is described in its own local reference frame, and the geometric relations between all the segments are known and used to calculate the global position of each body part. This structure is still nowadays used in almost every modeling and simulation frameworks.

In cases where more complexity is necessary, specific models, e.g. of individual joints and specific regions, have been proposed. As an example, Maurel and Thalmann [14] presented a model based on restrictions for the displacement of the scapula over the thorax. In this work a shoulder model is developed and each of the

clavicular, scapular and arm joints has three DOFs for rotation, and the scapula is bound to the thorax by means of a five DOFs joint (three rotations and two translations constrained to the surface of the thorax).

### 6.2.2.2 The Muscles, Tendons and Ligaments

Muscles are the active tissues that generate forces and cause motion. There are three different types of muscle in the body: cardiac, smooth and skeletal muscles. *Skeletal muscles* play a major role in motion. Their contraction is controlled through the somatic nervous system and this contraction produces active muscle forces that lead to postural stability and human body movement. Skeletal muscles are also major body components in size and mass, and partly characterize the body shape. Hence they constitute a critical part of biomechanics analysis and animation modeling. Skeletal muscles are attached to the bones by *tendons* while *ligaments* connect bones to bones. Tendons transmit the contractile force produced by muscle contraction to the bones and improve stability.

Skeletal muscles are composed of *muscle fiber bundles* called fascicles which themselves consist of internal fibers. Different internal arrangements of the muscle fascicles make different *muscle architectures*. In the simplest architecture, fascicles are all parallel to the length of the muscle. But in most of human muscles there is an angle between the fascicle's tendinous attachment direction and the longitudinal axis of the muscle. This angle is called *pennation angle*.

The earliest musculoskeletal mathematical model of skeletal muscle was suggested by Gasser and Hill [15]. That model was a one-dimensional representation of the muscle, called *action line*, and captured the global muscle mechanical properties. This model allowed later to describe the macroscopic relationships between muscle actuation, force-length and velocity along a muscle path, known as the *Hill's type model* [16]. In this model, the muscle is modeled by three components including the series element (SE), the parallel element (PE), and the contractile element (CE). A skeletal muscle is considered as a large sarcomere that is the contractile element and accounts for producing active muscle force which is dependent on the muscle length and time-varying neural signal. The series element represents the additional passive viscoelastic properties contributing for the tendon and aponeuroses. And the parallel element represents the behavior of the connective tissues epimysium, perimysium and endomysium. This model describes the contraction force of a muscle as the sum of the three elements.

The Hill's type model was improved by Zajac [17] to a dimensionless aggregate model which can be scaled to represent subject specific musculo-tendon units (MTUs). The force components are modeled from the measurement of isolated muscle fibers, which directly reflect the non-linear properties due to the sliding filaments. The series elastic elements can then be grouped with the tendon and removed from the model. Pennation effects are directly included into the model. For an extended view on skeletal muscle modeling we refer the reader to Lee et al.'s survey [18].

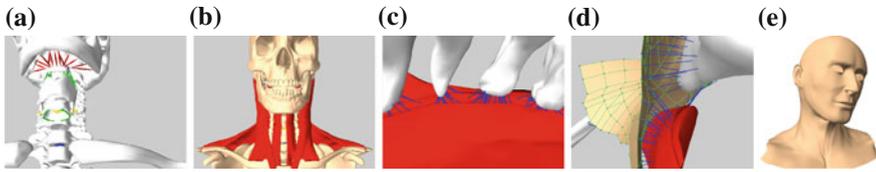
One-dimensional representation of muscles is sufficient in many applications, but many cases also require three-dimensional modeling. Three-dimensional modeling of muscle not only allows studying more complex structures but also leads to more realistic simulations. Three-dimensional muscle simulations can be obtained by using *finite element methods* (FEM) by subdividing muscles (and other anatomical entities) into small elements and applying continuum mechanics (see Sect. 6.2.1). As one of the earliest work, Chen and Zeltzer [19] modeled individual muscles as coarse linear elastic finite elements and used a Hill's type model to approximate the constitutive behavior. Active muscle forces were approximated as parametric functions and embedded into selected edges between vertices of a FEM-based solid.

Blemker and Delp [20] developed and evaluated a new formulation for creating three-dimensional finite element models that represent complex muscle geometry and the variation in moment arms of fibers within a muscle. This 3D muscle model has the advantage to represent complex muscle path motion but it is computationally expensive and impractical to use in real-time applications. At the same time, Teran et al. [21] used a finite volume method (FVM) with quasi-incompressible, transversely isotropic, hyper-elastic constitutive model to simulate soft tissue contraction and deformation. B-spline solids were used to model fiber directions, and the muscle activation levels were derived from key-frame animations. They claimed that FVM inherently requires less computation and memory usage in comparison with FEM. Later, Lee et al. [22] introduced one of the most detailed biomechanical model of the human upper body composed of a dynamic articulated skeleton, Hill's type muscle actuators including the force–velocity relation, and realistic finite element simulation of soft tissues and skin deformation. They used inverse dynamic with target poses and co-activation as input to compute muscle activation. The activation is then used to simulate skeleton dynamics and soft tissue deformation. The skin and underlying soft tissues were also simulated using FEM.

By using these detailed three-dimensional representations, accurate musculoskeletal simulation is achievable. In addition to be more accurate in comparison to one-dimensional representations, three-dimensional representations lead to more realistic visualization which is usually one goal in computer animation. But the computation cost and time consuming procedures of these methods make them impractical in many real world applications. Moreover, making three-dimensional representations of anatomical entities require generic or subject-specific data, as discussed in Sect. 6.2.3.

### 6.2.2.3 The Connective Tissue and Skin

The human skin has experimentally been approximated as a layered, nonlinear, thin, elastic and incompressible material. In computer animation and graphics, many models of physical skin have been proposed, notably for anatomy-based character rigging, and especially for body parts where skin deformation is clearly visible. It is sometimes important that a model can exhibit dynamics effects (e.g. jiggling and bulging) and not only pure kinematics effects from geometric rigging techniques, as it thus



**Fig. 6.3** Example of physical simulation of layered anatomical human neck. **a** Skeleton, **b** Underlying musculature, **c** Muscle to bone attachments, **d** Elastic skin and connective tissue **e** Result of a dynamic skinning

produces much more convincing character animation. In general, such models are broken down into several well-defined layers that contribute to the overall visual appearance and behavior. The fat and connective tissue layers both separate and attach the skin to the underlying muscle and bone layers. For instance, the fat layer can be specified simply as a thickness between the skin and muscle layers, and the connective tissue modeled as rubber bands strung between points on the skin surface and on the muscle layer surface [23].

The discretization of the numerical modeling of the physical skin is widely employed in deformable character with a geometric- or physically-based rigger. When a character skeleton is rigged, the influence of a joint can be calculated. That influence gives the transformations of the skin surface vertices to apply according to the character pose. These transformations can either be propagated by inverse and forward kinematic forces to the skin directly or through attached soft tissues (see Fig. 6.3). Two popular surface modeling approaches exist. The first one accounts for the elasticity only from the consideration of the thin feature. In the second, volumetric models with volume conservation can take the incompressible factor into account.

Focusing on the elastic surface is one possible approach to a fast solution for physically-based deformation of anatomical humans. Turner and Thalmann [23] discretized the skin surface using the finite difference method (FDM), i.e. used a rectangular mesh of three-dimensional points as representation together with the physical characteristics (e.g. mass, elasticity) and current state information (e.g. position, velocity). Spring is also a popular approximation to the elasticity of a material and the mass-spring system is widely used for skin modeling, especially facial deformation, which can date back to the layered elastic model used by Terzopoulos and Waters for facial animation [24]. In this model, a 3D mass-spring lattice is attached to a human skull model and deformed by muscles which take the form of force constraints between the skin surface and the underlying bone. The springs are biphasic to emulate the non-linear behavior of real skin. Galoppo et al. [25] presented a fast method to capture dynamic deformations on the surface of a soft body including a rigid core, and they extended their method later to apply to soft body characters with multiple rigid bones [26]. Shi et al. [27] developed a surface-based deformable model to enrich the skeleton-driven character skinning in real-time, with physically-based secondary deformation. They learned the material parameters, mainly the stiffness, of the surface model from a surface mesh and a few sample sequences of its physical behavior. However, these simulations only consider the elastic energy from skin-layer

deformation and do not include the deformation inside the volume, so it does not capture pose-dependent deformations correctly.

The incompressibility of the human skin is realized in mathematical models by simulating the volume conservation of the deformable bodies. Terzopoulos and Waters [24] used a multi-layered mass-spring structure to yield the volume preserving constraints to simulate the effects of incompressible fatty tissue. However, the difficulty in handling the volume preservation in a mass-spring system makes the finite element methods widely used as an alternative approach. The nonlinear Green-Lagrangian strain tensor can express large deformations correctly whereas expensive computation is required. However, using a quasi-static approximation, it is still possible to have a practical character skinning. Teran et al. [28] presented a quasi-static solution for flesh deformation driven by a skeleton. Lee et al. [5] used a similar method to compute the deformation of the soft tissue in their biomechanical model of the human upper body. Though such quasi-static solutions are much faster than a fully dynamics simulation with the same model size, they do not capture the dynamic behaviors of the soft body such as jiggling. Fast simulation also can be pursued by simplifying the computation using a linearized strain, or infinitesimal strain under the assumption of small deformation [29]. Nonetheless, it causes serious problems such as inflation of the body especially when the deformation contains rotational part. To alleviate this shortcoming, many corotational methods (e.g. [30, 31]) are proposed to remove as much of the rigid rotation as possible by using local coordinate frames following the global motion of the body. Though corotated linearized strain has been widely used in interactive graphics applications, it is still valid only when the deformation is very small.

In deformable character animation, the size of the skin model can be large, and often requires a lifelike skin deformation, especially in animation film and interactive graphics application. A high resolution finite element model has to be built with a large amount of elements, which can eventually be million-scale [5]. This makes the simulation totally impractical by current common computing power.

Methods using modal reduction to reduce the complexity of a finite element system have been investigated, whereas they are not sufficient to capture the nonlinear deformations. Alternatively, many current techniques called *mesh embedding* use the concept of spatial embedding, where a relatively low-dimensional coarse volumetric mesh enclosing the entire deformable body is generated, and it expresses the behavior of the body and embeds a fine geometry which is also for the visualization of the skin deformation. One of these techniques relies on a free-form deformation lattice attached to the skeleton [32].

Mesh embedding has been widely used to simulate soft bodies as it reduces the DOF of the deformable bodies without losing the fine geometry of the characters and the internal skeleton can be handled more easily in the embedding mesh system compared to the modal reduction [33]. For that reason, Lee et al. [5] embed a high-resolution skin surface as the visualization geometry into the simulation mesh in their comprehensive upper human body model.

Since in skinning, the realism delivered to observers lies in the visualization of skin deformation, some frameworks prefer to replace the skin simulation by

a post-processing shape deformation component. Zordan et al. [3], in their breathing simulation model, recorded the trajectories of pre-selected points attached to the model used as control vertices of a NURBS surface. The surface shape is then updated to show skin deformation. However, the surface shape is implicitly defined by how the control vertices are selected and limited to the captured data. High-quality rendering of the skin can be considered to be done offline using advanced skin rendering engines like Pixar's RenderMan [34] or NVIDIA's Mental Ray [35], to enrich the visualization referenced to real human skin.

### ***6.2.3 Acquisition and Processing of the Human Anatomy***

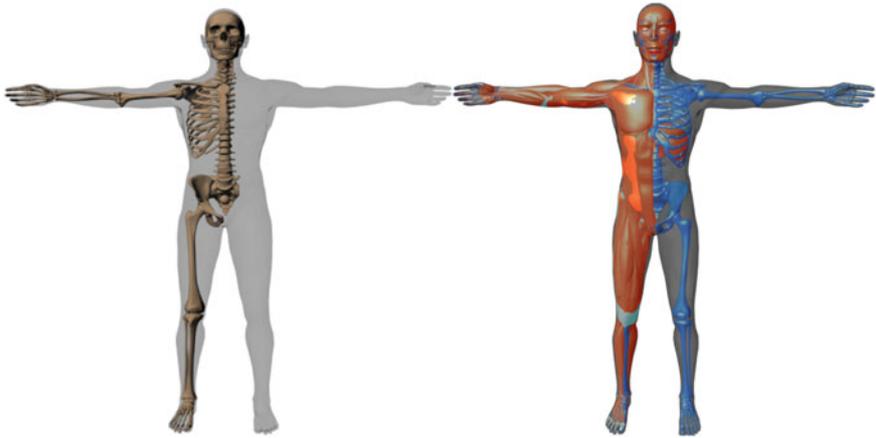
In this section we present the typical pipelines for automatic generation of volumetric meshes of anatomical structures. We introduce techniques to acquire anatomical structures from the real-world and to process these structures into a musculoskeletal model that can be simulated.

#### **6.2.3.1 Artistic Anatomy**

In computer animation and graphics applications, most human figure models use a simplified articulated skeleton consisting of relatively few connected segments. The bones are constrained by joints which allow them to move relative to one another within limits. Since the focus is on the capability of articulation, it is usually not necessary for the skeleton geometry to conform to the real world, even when movable joints are modeled with sets of curves [36, 37]. Nonetheless, subject-specific and accurate geometries are generally required in biomedical research like surgery simulation.

In an anatomical human (see Fig. 6.4), the musculature system is typically more complex than the skeletal system. Indeed in the human anatomy, muscles are arranged side-by-side and in layers on top of bones and other muscles. They often span multiple joints and typically consist of different types of tissue, allowing some portions to be contractile and others not. Depending on their state of contraction, muscles have different shapes and influence their surface form in different ways.

In an effort to achieve real-time performances, muscles are usually constructed from NURBS or spline patch (see Fig. 6.5). Scheepers et al. [11] developed anatomy-based models of skeletal muscles used to flesh-out a skeleton. They use ellipsoids to represent muscle bellies and deformation is achieved by updating the volume when the lengths of the principal axes are adjusted. As a general muscle model, they construct a bicubic patch mesh by sweeping a varying ellipse along a cubic Bezier curve, and reach the deformation by manipulating the control points sampled along the curves. Pratscher et al. [39] use elliptical muscle models and a procedural deformation for the muscle simulation in their anatomy-based character rigging system.



**Fig. 6.4** The Ultimate Human Model data set by cg Character, an artistic and anatomically accurate human model [38]



**Fig. 6.5** Muscle deformation by manipulating the control points of a NURBS surface

These geometrical skinning techniques consider the anatomy such that they can reproduce muscle bulging, yet the deformation cannot be considered as realistic.

### 6.2.3.2 Segmented MRI and CT scans

3D representation of musculoskeletal models have been used extensively in animation to make better visualization and in biomechanics to study musculoskeletal function in different clinical cases [40]. These representations can be generic or subject-specific. Generic musculoskeletal models have some limitations. Most of the software packages for biomechanical analysis of muscle function are based on biomechanical studies of cadaveric specimens and use the musculoskeletal geometry of a healthy, average-sized adult male with normal musculoskeletal geometry [41–43]. These models are useful for studying general trends in large populations. However, if modeling results are to be used to plan surgical procedures, guide treatment decisions and trial implant designs, they must take into account a subject’s individual musculoskeletal anatomy. The musculoskeletal system is very intricate and large anatomical variations exist among individuals. The different musculoskeletal geometry due to size or pathology can also affect the accuracy of results derived from generic models. Neuromuscular diseases, such as cerebral palsy, are often responsible for producing

differences in bony geometry and muscle attachment sites. To understand the effects of joint disease, a subject's unique articulation shape and cartilage thickness must be known, both of which contribute to the contact pressure distribution. The need to customize musculoskeletal anatomy is an essential step in the modeling process, if the predictions of computer models are to be useful to clinicians [44].

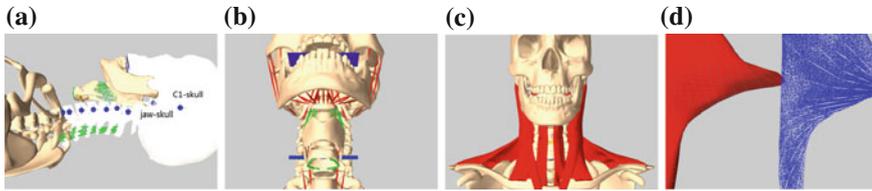
Scheys et al. [45] have demonstrated the inaccuracy of gait kinematics calculated from scaled generic models in subjects with increased femoral anteversion. Since the results of simulations are often sensitive to the accuracy of the functional musculoskeletal model, individualized musculoskeletal models may be a better alternative. Hence, for an accurate representation of subject-specific anatomical structures, segmentation of medical images are extensively used. Medical imaging techniques such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT) and also Positron Emission Tomography (PET) are used in order to generate 3D volumetric data sets of the human body and to study *in vivo* the complex geometric relationships among the muscles, bones, and other structures [46, 47]. However, it is time consuming and requires extensive imaging protocols to capture the muscle and joint geometries at different limb positions. Subject-specific musculoskeletal modeling also addresses the problem of image segmentation, which consists of extracting anatomical structures from medical image data such as MRI. Semi-automatic or fully automatic segmentation methods are fast but inaccurate since muscle distinction is often difficult or impossible to assess with currently used methods. Thus, soft tissue volumetric representations are most often and most accurately acquired by defining contours manually. Blemker et al. [20] built for example volumetric finite element representations of muscles from manually segmented MRIs.

### 6.2.3.3 Preparing Anatomical Data for Simulation

The anatomical data resulting from artist models or segmented scans usually need to be processed to produce a model that can be simulated. In this section we consider two case studies.

In our first case study, we want to focus on the production of FEM-ready volumetric meshes from raw segmented surface models of medical images [48]. At first, it comes handy to store areas such as attachment sites and tendon vertices in an index-invariant structure by defining them by geometry as some tasks may change the amount and ordering of the vertices. Then smoothing out the surface geometries is usually performed by a three-parameter low-pass filter [49] to remove acquisition artifacts. Missing entities can be generated semi-automatically such as tendon extremities but the most important step is the resolution of self-intersections and overlaps. While using Boolean operators for solving overlaps, Peeters and Pronost [48] proposed an algorithm to remove self-intersections in anatomical entities. The final steps consist in generating the volume meshes with the relevant materials and designing the FEM constraints from the index-invariant structure of the first step.

In our second case study, we want to create a realistic and comprehensive model for the human neck, starting from the surface models of an anatomical artistic human data



**Fig. 6.6** **a** Skeleton (*green lines* represent damped springs), **b** Deep muscles (*red lines*), **c** Superficial muscles and **d** Muscle fibers

set (see Fig. 6.6). Our skeletal structure contains the skull, jaw, C1–C7 cervical bones. First, one to three DOF rotational joints are inserted between adjacent vertebrae, and the jaw–skull system by carefully locating the pivot points (blue dots in Fig. 6.6a). Joints are also limited to a certain range from a study on the human neck [50]. Bones that are involved in the neck mechanics are modeled as fixed rigid bodies. Notice that two extra unilateral planar constraints can be added between the upper and lower teeth (blue planes in Fig. 6.6b), hence to resist penetration between jaw and skull. For stabilization, damped springs can be attached between certain adjacent vertebrae (green lines in Fig. 6.6a). At that point, the stiffness of the ligaments can be included [51]. In order to simulate the aspects of the throat, the hyoid, thyroid, and cricoid bones can be incorporated. A revolute joint (blue cylinder in Fig. 6.6b) is created between thyroid and cricoid, and damped springs (green lines in Fig. 6.6b) are attached between hyoid–sternum, thyroid–cricoid and cricoid–sternum. The skeleton structure is then simulated using a multi-body approach.

Then we have to generate the volumetric (e.g. tetrahedral) meshes of our deformable bodies using a mesher such as TETGEN [52] or CGAL [53]. Applying some standard geometrical operations beforehand may be useful, such as *edge bridge* followed by *fill hole* to fill the holes, and/or employ self-intersection removal algorithms [48, 54]. In order to obtain more accurate simulations, a maximum tetrahedron volume constraint may be needed to ensure that the size of the finite elements is reasonably small.

Next, muscle fibers can be modeled as polygon curves with a small number of segments along the direction of muscle contraction. To achieve it, we can select points at each attachment area to the bone and additional points on the surface of the polygon mesh. According to the principle that each pair of points should lie on a line in the direction of muscle contraction, we can pair the points up. Linear interpolation between two points can then be performed to generate the segments. Next, Hill’s type muscle models (see Sect. 6.2.2.2) are defined at the segmental scale. The method proposed by Tan et al. [55] can be used to calculate the force on nearby elements induced by muscle contraction. Figure 6.6c–d shows an example of muscles and muscle fibers generated for the trapezius. Finally, the FE models are coupled to the skeletal system by attaching nodes to the bones.

## 6.2.4 Acquisition of the Function of Human Anatomy

Once a 3D model of the human body is built, motion and force can be captured and used to derive simulations in order to add realism to computer animation and allow for accurate results in biomechanical simulation. Capturing data related to kinematics variables that describe movement can be two-dimensional or three-dimensional. Kinematics data describe the motion of a system without consideration to its mass or the acting forces and usually consist of coordinates of tracked markers or joint articular trajectories. This data can be applied to a biomechanical model in order to analyze the movement of a subject, but also to an animation to drive the motion of a user's avatar. Kinetic data describes the forces that produce or result from movement. The most commonly used equipment to collect motion and force data are video cameras, optoelectronic systems, electromyography, force plates, and inertial systems.

### 6.2.4.1 Video Cameras and Optoelectronic Systems

Video recording and optoelectronic systems are used in motion capturing to track poses over time of an actor or a patient being tested. While video capture can be solely based on the subject's shape, in practice often markers are placed on the body as well. Markers are either solid shapes covered with retro-reflective tape in *passive* marker systems, usually sensitive to infrared light (IR), or they are emitting diodes (LEDs) in *active* marker systems.

These latter LED markers are pulsed sequentially, so the system automatically knows the identification of each marker. The system can then maintain the identification of each marker, if it is occluded by other body parts, or if it temporarily moves outside of the viewport. Mixing up markers cannot occur in such a system and therefore markers can be placed very close together. However, these systems have the disadvantage that it requires much more equipment to be placed and held by the user. Also for long duration trials, the heat generated by the LEDs might be a problem.

On the other hand, passive markers reflect light which is emitted by LEDs (usually IR and near IR) placed around cameras. IR filters are used on the camera lenses and thresholds are set to distinguish between bright light from markers and dim light considered as background noise. Each marker trajectory must be identified with a label and tracked throughout the trial. When markers are lost from view or their trajectories cross, this may cause loss of their proper identification. There is also a limitation on how close markers can be placed together. These systems have the advantage of using lightweight reflective markers without the need for electrical cables or batteries on the user [56].

Two-dimensional analysis requires only one camera positioned perpendicular to the plane of movement. Any marker movement outside this plane will be distorted. Three-dimensional analysis looks at movement in all directions and requires more than one camera. To achieve 3D analysis, a computer software calculates the 3D

coordinates for each marker based on the 2D data from two or more cameras and the known location and orientation of the cameras. In practice, the points of interest on the subject must be visible by at least three cameras at all times in order to reconstruct their positions [56, 57]. If markers are used, skin movement artifacts caused by the skin moving over the musculoskeletal structure must be removed from the data before reconstruction of movement can be performed.

#### 6.2.4.2 Electromyography

Electromyography (EMG) is used to detect and measure the small electric signal produced by muscles during contraction. Electrodes (sensors) are used to detect electric signals and there are two types: *surface* electrodes and *wire* electrodes. Surface electrodes are placed on the skin and wire electrodes inserted into the muscle. Surface electrodes have gained widespread use due to their ease of application and because skin penetration is not required. However, placing the sensors on the skin can give erroneous readings for a specific muscle as other muscles lying around or on top can cause cross-talk in the signal. Deep muscle signals can be reliably obtained only with intramuscular wire electrodes. Inserting electrodes into the muscle itself gives more accurate readings for the muscle activation but is invasive. Surface electrodes come in two basic types: *passive* and *active*. Active electrodes have become quite popular, as they provide signal amplification at the electrode site [56, 58].

#### 6.2.4.3 Force Plates

A force plate is used to measure ground reaction forces exerted by a person and consists of a steel plate and transducers at each corner. When a load is applied to the plate, transducers detect it and the load is converted into an electrical signal. The magnitude and direction of the force (vertical and shear forces) are measured and the instantaneous center of pressure can be calculated. Two types of force plates are available: *piezoelectric* and *strain gauge*. Piezoelectric force plates use quartz transducers which generate an electric charge when stressed. They do not require a power supply to excite the transducers. However, special charge amplifiers and low noise coaxial cables are required to convert the charge to a voltage proportional to the applied load. In general, piezoelectric force plates are more sensitive and have a greater force range than strain gauge types. Strain gauge force plates use strain gauges to measure the stress in specially constructed load cells when a load is applied. They do not require the special cabling and charge amplifiers of the piezoelectric type. However, they require excitation of the strain gauge bridge circuit. The size of force plate and the range of readings depend on the application. Force plates can be combined with kinematics data and inverse dynamic systems to work out parameters such as the net moment about each joint, muscle forces and joint contact forces [59].

#### 6.2.4.4 Inertial Systems

Inertial systems include *accelerometers* and *gyroscopes* and are based on the principle of measurement of inertia. Accelerometers operate on a mass-spring principle. Two charged plates are separated and a capacitance or resistance between them is given as a function of their separation. One plate is suspended over the other on flexible mounting and acceleration causes this mounting to flex giving a change in plate separation. The change in capacitance or resistance is measured and the change in separation calculated. The second derivative of the change in separation with respect to time gives the acceleration at the attachment point [57].

Gyroscopes are devices used for measuring orientation and can be used in gait analysis to give segment orientation. In order to obtain the limb orientation the angular acceleration must be integrated twice with respect to time and this will amplify any initial errors. The sensors themselves are small and lightweight, and can detect a large range of angular velocities. If gyroscopes are combined with accelerometers then the data can be used to easily obtain the kinematics of the subject's movement [57, 60].

### 6.3 Simulating Virtual Anatomical Humans

#### 6.3.1 Rigid Body Physics and Muscular Actuation

Physical simulation offers the possibility of truly responsive and realistic animation. We have observed in the last decade a renewed interest in the use of physical simulation for interactive character animation and simulation, and many recent publications demonstrate tremendous improvements in robustness, visual quality and usability. For a detailed review about physics-based character animation, we refer the reader to the survey paper [61]. In a physics-based system, virtual humans need forces and torques to actively move around. In order for such forces or torques to be realistic, they must originate from within the character. We use the term *actuators* to describe the mechanisms that generate the forces and torques that make a character move. Common frameworks make use of a combination of joint torques (straightforward DOF actuation model), external forces (e.g. to control the global translation and rotation) and virtual forces (joint torque emulation of external force effect). In addition to these actuation schemes, a fair amount of recent works are dedicated to actively actuate virtual anatomical humans through simple muscle actuators.

Biological systems are actuated through contraction of muscles, which are attached to bones through tendons. When muscles contract, they generate torques at the joint(s) over which they operate, causing the attached limbs to rotate in opposite directions. In addition to contracting when activated, muscles (and tendons, in a lesser degree) have the ability to stretch. This makes them behave like unilateral springs and allows for an efficient mechanism for energy storage and release. Since muscles can only pull, at least two muscles are required to control a single DOF

(so-called antagonistic muscles). In physics-based character animation, use of muscle-based actuation models is uncommon, because of the increased number of DOFs that require control and decreased simulation performance [62]. However, examples of muscle based actuation do exist, and we witness an increased interest in using more advanced muscle-based actuation models for controlling physics-based characters [63–65]. This reflects the need for more accurate anatomical virtual humans.

In one of the latest work to date, Wang et al. [66] propose a biologically-motivated locomotion controller. Their lower-body model is actuated by sixteen Hill's type MTUs (see Sect. 6.2.2.2). To determine muscle excitation patterns, biologically-motivated laws are used for muscle control, and stance and swing phases. The parameters of these control laws are set by an optimization procedure that satisfies a number of locomotion task terms while minimizing a biological model of metabolic energy expenditure. This work demonstrates the importance of modeling constraints on torque generation due to muscle physiology, both in restricting the space of possible torque trajectories and in providing a realistic model of effort.

### 6.3.2 Deformable Body Simulation

The human body consists of intricate deformable tissues. To achieve realistic animation and to study biomechanics of deformation in medical applications, the realistic deformation of the human body system is required. Several approaches have been proposed to model human body deformations. As the emphasis of this chapter is on challenges in biomechanics and animation, we classified modeling approaches into non-physically based methods and physically based methods.

#### 6.3.2.1 Non-Physically Based Methods

Non-physically based methods are useful methods in many applications especially when a high level of geometric control is needed. They usually use simplified physical principles to achieve visually appealing results. The most important non-physically based methods described here are based on surface data (parametric and polygonal surface and implicit surface) and free-form deformation.

##### 3D Surface

Parametric and polygonal surfaces can be used to model deformable bodies. One method to model deformation is using *splines*. Splines are used as a tool to create and interpolate curves and surfaces mainly in the field of computer aided geometric design. This technique is based on the representation of both planar and 3D curves and surfaces by a set of control points or landmarks. Bezier curves are widely used to model smooth curves. The curve is completely contained in the convex hull of its control points which can be graphically displayed and used to manipulate the curve

intuitively. Transformations such as translation, scaling and rotation can be applied on the curve by applying the respective transform on the control points. Bezier splines are sets of low-order Bezier curves patched together to represent more complex shapes. Another type of spline is Basis spline (B-spline) which is a generalization of Bezier curves. They depend on the  $k$ -nearest control points at any point on the curve. Combining B-splines allows creating B-spline surfaces. Another generalization of Bezier splines is non-uniform rational basis spline (NURBS). The primary difference of this type is the weighting of the control points which makes them rational.

Another method to model deformation of bodies is using *implicit surfaces*. Implicit surfaces are introduced as an extra layer coating any kind of structure that moves and deforms over time. They can provide an efficient collision detection mechanism by offering a compact definition of a smooth surface around an object [67]. An implicit surface [18] generated by a set of skeletons  $s_i$  ( $i = 1, 2, \dots, n$ ), with associated field functions  $f_i$ , is defined at the isovalue  $c$  by:

$$\left\{ P \in \mathbb{R}^3 \mid f(P) = c \right\}$$

where

$$f(P) = \sum_{i=1}^n f_i(P)$$

The field function is generally a decreasing function of the distance from a given point  $P$  to the associated skeleton. Based on the type of field function, various implicit surfaces have been developed such as blobs, metaballs, soft objects, and convolution surfaces [18].

### Free Form Deformation

Free Form Deformation (FFD) consists in deforming the space embedding objects. FFDs provide simple and fast control, but they do not permit direct manipulation. Also, the regular lattice spacing used by FFDs prevents the detailed control needed to produce more complex shapes [18]. As one of the early works, Chadwick et al. [68] employed FFDs to represent muscle deformation. Articulated skeletons, located inside muscles, transform a surrounding lattice and cause changes in the shape of the muscles. As another example, Moccozet and Thalmann [69] presented a generalized method for FFDs that combines the traditional model with techniques of scattered data interpolation based on Delaunay and Driehlet–Voronoi diagrams. They applied the method to model deformations around a human hand model.

#### 6.3.2.2 Physically Based Methods

While non-physical based methods are useful methods in many applications, physically based methods are a better choice for biomechanical human modeling with

medical application. Physically based methods lead to more realistic simulation and also provide the capability to study a case as a mechanical simulation and to find the real behavior of tissues during deformation. In this method, *partial differential equations* (PDEs) which govern the evolving shape of the deformable objects and their motion through space should be solved. The major difficulty in these methods lies in the complexity of the physical phenomena that should be simulated and computationally solved from the PDEs. To overcome this difficulty, one should simplify the model and apply numerical techniques to solve the PDEs. We describe hereafter two relevant methods based on mass-spring systems and finite elements.

### Mass-spring systems

A mass-spring system is a physically based technique that has been widely and effectively used for modeling deformable objects. An object is modeled as a collection of point masses connected by massless springs in a lattice structure. Springs connecting point masses exert forces on neighboring points when a mass is displaced from its rest position. The elastic force acting on mass  $i$  connected by a spring to mass  $j$  is given by:

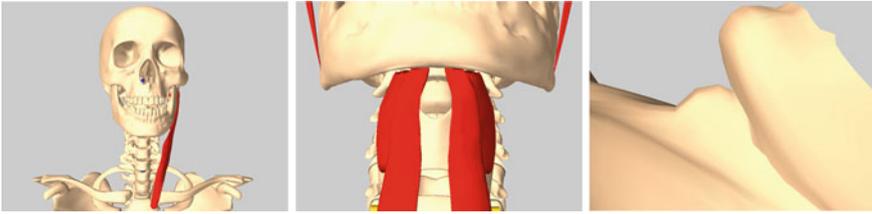
$$f_{ij} = k (|x_{ij}| - l_{ij}) \frac{x_{ij}}{|x_{ij}|}$$

where  $x_{ij} = x_j - x_i$  and  $x_i, x_j$  are the locations of point masses  $i$  and  $j$ , respectively,  $l_{ij}$  is the rest length between them and  $k$  is the spring's stiffness. Applying this equation on all the points leads to a differential system of ordinary equations that can be solved explicitly using various algorithms [18]. Mass-spring systems are easy to construct, and both interactive and real-time simulations of mass-spring systems are possible. Also it has the ability to handle large deformations. As a disadvantage, usually spring constants are approximated from measured material property and allocating suitable constants that express all tissue properties in a natural way is difficult.

Mass-spring systems have been widely used in facial animation. As an example in biomechanical modeling, mass-spring systems were used by Nedel and Thalmann [70] to simulate muscle deformation. Muscles were represented at two levels, action lines and muscle shapes. The muscle shapes were deformed using a mass-spring mesh. They used angular spring to control the volume of muscles during deformation and smooth out mesh discontinuities.

### Finite element method

Finite element method (FEM) is another physically-based technique that has been widely used in soft tissue modeling. Contrary to mass-spring systems that treat the mechanics as a discrete process, FEM views it as a continuum (see sect. 6.2.1.1). For that reason, FEM usually leads to more accurate physical models compared to



**Fig. 6.7** Illustrations of an anatomical simulation of the neck, including skeletal muscles actuation and Adam's apple simulation

mass-spring systems. Each finite element model of a 3D object consists of a solid body with mass and energies throughout. In FEM, an object is divided into small elements (e.g. hexahedra or tetrahedra in 3D, quadrilaterals or triangles in 2D) joined at discrete nodes. Displacements and positions in an element are approximated from discrete nodal values using interpolation functions:

$$\Phi(x) \approx \sum_i h_i(x) \Phi_i$$

where  $h_i$  is the interpolation function for the element containing  $x$  and  $\Phi_i$  is the scalar weight associated with  $h_i$ .

Choosing the appropriate element type and the interpolation functions depends on the object geometry, accuracy requirements, and computational budget. Higher order interpolation functions and more complex elements lead to more accurate solutions however it needs more computation. To sum up, finite element methods provide more realistic simulations than mass-spring methods but they are more computationally expensive. To achieve real-time deformation, reducing the computation time is necessary. Unfortunately, it is not always possible to use large timesteps to achieve this reduction, due to the large deformations that occur with the soft tissues present in the human body, in particular muscles. Some examples of usage of this technique to simulate muscle deformation have been mentioned in Sect. 6.2.2.2.

## 6.4 Implementing Virtual Anatomical Humans

Several open simulation architectures have been presented to the biomedical community in recent years. In this section, we introduce four well-known subsystems and provide this information as a reference to whom would like to quickly start researching and developing in anatomical modeling and simulation of humans. Comparisons are implicitly given by presenting some of our research in the simulation of neck muscles and the Adam's apple.

We researched the simulation of neck muscles and the Adam's apple based on anatomical modeling and simulation (see Fig. 6.7). The muscles are modeled as

deformable bodies and the skeletal mechanism as a dynamic multibody system. Besides the simulation of the underlying structures, we also simulate skin deformation with the ultimate goal of achieving realistic real-time facial animation (see Sects. 6.2.2.3 and 6.2.3.3). To reach the desired simulation, we leverage features from ArtiSynth [71]. ArtiSynth is an open source, Java-based biomechanical simulation environment for modeling complex anatomical systems composed of both rigid and deformable structures. It provides fully coupled FEM/multibody capabilities.

Nonetheless, we still design the musculoskeletal dynamics using OpenSim's neck model [50]. OpenSim [41, 43] is a multibody simulator with inverse modeling capabilities designed for musculoskeletal studies. In such simulator, it is impossible to directly obtain deformation of muscle shape conforming to real world. The muscles in OpenSim are 1D action lines using the Hill's type model, and therefore the system does not support deformation of anatomical structures. It hinders the skinning of the musculoskeletal structure, nonetheless critical for animation purposes.

As nonlinear FEM produces more accurate muscle deformation, we use the FEBio solver [72] to simulate the muscles. FEBio is a non-linear finite element toolkit with special support for tissue modeling and some support for rigid-bodies, contact and constraints. Our FE models are generated by a meshing algorithm on the polygon meshes of an artistic anatomical data set. Because FEBio only support the skeleton modeling weakly, especially the integration with the muscles, we have to give much manual efforts to model the skeleton, and attach the finite nodes to the relevant bones by the selection functions it provides.

Nonlinear FEM simulation is usually too computationally expensive to be used in interactive graphics application. As aforementioned, the mesh embedding is a useful solution to speed up the performance while it still can keep high-fidelity emulation. Sofa [73] is such an environment which implements several schema of mesh embedding such as barycentric interpolation and mesh topology mapping. There is one highlight of its architecture, the modularization, where each simulation component (e.g. mechanical state, FEM simulation, collision, mesh embedding and visualization) is an individual module. Through its GUI, we can drag modules into the scene graph to form a simulator. Moreover, the modules can be customized by modifying a set of parameters. To simulate a deformable body in real-time, Sofa is highly recommended while it is not a comprehensive biomechanical modeling and simulation environment.

In order to study the biomechanics of a system for simulation in character animation, we advise to use ArtiSynth. It provides several material implementations including nonlinear, linear and corotational linear materials, and explicit and implicit time integrators. Automatic coupling of the skeleton-muscle systems can be implemented by spatial search using various bounding volume hierarchies. However, in practice, we have experienced an easier and more accurate manual selection of attachment nodes in the FEBio environment.

To sum up, each open simulation environment has its unique advantages and supports for different tasks. We can make extensive use of one or multiple of such environments to start a research or development project. We believe that it

is a promising solution to adopt results from biomechanical modeling for the production of believable character animations.

## 6.5 Conclusion

In order to realistically animate the human body, it is necessary to take into account the biomechanical aspects of modeling. Although this makes procedures more complicated and thus computationally expensive, it will result in more meaningful medical simulations. Below, we describe possible steps to undertake in order to model and simulate the complex human musculoskeletal structure.

- Defining tissue material properties and mechanical principles that should be applied to simulate the human body.
- Studying the anatomy of the tissues which contribute to the modeling of the bones, muscles, joints and skin.
- Making anatomical geometry of the tissues either generic or subject-specific based on study case. Patient specific models can be made by using medical images for specific users.
- Preparing the anatomical geometries to be used in the modeling procedure.
- Defining a method that models each tissue given a realism criterion and that contains the biological behavior of the tissues. The method may also take into account interactions of different tissues.
- Defining different techniques to control the function of the anatomical structures, producing human movement and simulating clinical cases. These techniques may make use of resulting forces and motion capture data to derive the simulations.
- Defining a suitable method to show deformations of soft tissues making the simulations as close to reality as desired.

Achieving realistic animation that includes plausible biomechanics leads to more visually appealing visualization of virtual humans and may allow medical application as well. However, as mentioned in this chapter, the modeling and simulation procedures are time consuming and typically result in low frame rates. One of the reasons for such performance lies on the nonlinear mechanical properties of anatomical structures. In addition, the detailed geometries that are necessary for the continuum simulation of deformable bodies relate directly to the computation time. In an effort to reduce computation time, we observe a trade-off between simplification of mechanical relations and tissue properties, and accuracy. Considering bones as rigid bodies is one example. Another possibility consists in using methods that combine offline and online computations. And hopefully, the computer graphics community has in the last decade proposed many efficient techniques in that regard.

Moreover, we have to notice that virtual anatomical human modeling and simulation often induce a large number of parameters to be tuned to reach a desired result. In practice this task may reveal itself as difficult and time consuming. A good starting point is to initially adopt parameter values from the experimental works and then

tune them until converging to a satisfactory solution according to measurements. In a virtual scene where the deformation of human body is simulated, graphical visualization can be incorporated into the system. For example, in the simulation of the human skin, high-end rendering could be used to make the visualization more intuitive to observers. The better the visualization, the more realistic the simulation can become.

**Acknowledgments** This work has been supported by the Dutch research project COMMIT—Virtual Worlds for Well-Being [74].

## References

1. 3D anatomical human. EU Research Project. <http://3dah.miralab.ch/>, 2010.
2. Multi scale biological modalities for physiological human articulation. EU Research Project. <http://multiscalehuman.miralab.ch/>, 2013.
3. Zordan, V. B., Celly, B., Chiu, B., & DiLorenzo, P. C. (2006). Breathe easy: Model and control of human respiration for computer animation. *Graphical Models*, 68(2), 113–132.
4. DiLorenzo, P. C., Zordan, V. B., & Sanders, B. L. (2008). Laughing out loud: Control for modeling anatomically inspired laughter using audio. *ACM Transactions on Graphics*, 27(5), 125:1–125:8.
5. Lee, S.-H., Sifakis, E., & Terzopoulos, D. (2009). Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics*, 28(4), 99:1–99:17.
6. Winter, D. A. (2005). *Biomechanics and motor control of human movement* (3rd ed.). Hoboken: Wiley.
7. Nordin, M., & Frankel, V. H. (2012). *Basic biomechanics of the musculoskeletal system* (4th ed.). USA: Wolters Kluwer Health.
8. Gibson, L. J., & Ashby, M. F. (1999). *Cellular solids: Structure and properties*. Cambridge Solid State Science Series. Cambridge: Cambridge University Press.
9. Spivak, J. M., DiCesare, P., Feldman, D., Koval, K., Rokito, A., & Zuckerman, J. D. (1999). *Orthopaedics: A study guide*. New York: McGraw-Hill.
10. Seth, A., Sherman, M., Eastman, P., & Delp, S. (2010). Minimal formulation of joint motion for biomechanisms. *Nonlinear Dynamics*, 62(1–2), 291–303.
11. Scheepers, F., Parent, R. E., Carlson, W. E., & May, S. F. (1997). Anatomy-based modeling of the human musculature. In *Proceedings of the 24th annual conference on Computer Graphics and Interactive Techniques*, 1997, pp. 163–172.
12. Yamaguchi, G. T., & Zajac, F. E. (1989). A planar model of the knee joint to characterize the knee extensor mechanism. *Journal of Biomechanics*, 22(1), 1–10.
13. Wilhelms, J. (1997). Animals with anatomy. *IEEE Computer Graphics and Applications*, 17(3), 22–30.
14. Maurel, W., & Thalmann, D. (2000). Human shoulder modeling including scapulo-thoracic constraint and joint sinus cones. *Computers and Graphics*, 24(2), 203–218.
15. Gasser, H. S., & Hill, A. V. (1924). The dynamics of muscular contraction. *Proceedings of the Royal Society of London. Series B, containing papers of a biological character*, 96(678), 398–437.
16. Hill, A. V. (1938). The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 126(843), 136–195.
17. Zajac, F. E. (1988). Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*, 17(4), 359–411.
18. Lee, D., Glueck, M., Khan, A., Fiume, E., & Jackson, K. (2010). A survey of modeling and simulation of skeletal muscle. *ACM Transactions on Graphics*, 28(4).

19. Chen, D. T., & Zeltzer, D. (1992). Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. *SIGGRAPH Computer Graphics*, 26(2), 89–98.
20. Blemker, S. S., & Delp, S. L. (2005). Three-dimensional representation of complex muscle architectures and geometries. *Annals of Biomedical Engineering*, 33(5), 661–673.
21. Teran, J., Sifakis, E., Blemker, S. S., Ng-Thow-Hing, V., Lau, C., & Fedkiw, R. (2005). Creating and simulating skeletal muscle from the visible human data set. *IEEE Transactions on Visualization and Computer Graphics*, 11(3), 317–328.
22. Lee, S.-H., Sifakis, E., & Terzopoulos, D. (2009). Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics (TOG)*, 28(4), 99.
23. Turner, R., & Thalmann, D. (1993). The elastic surface layer model for animated character construction. *Communicating with virtual worlds*. Tokyo: Springer Verlag.
24. Terzopoulos, D., & Waters, K. (1991). Techniques for realistic facial modeling and animation. *Computer animation*. Tokyo: Springer-Verlag.
25. Galoppo, N., Otaduy, M. A., Mecklenburg, P., Gross, M., & Lin, M. C. (2006). Fast simulation of deformable models in contact using dynamic deformation textures. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, pp. 73–82.
26. Galoppo, N., Otaduy, M. A., Tekin, S., Gross, M., & Lin, M. C. (2007). Soft articulated characters with fast contact handling. *Computer Graphics Forum*, 26, 243–253.
27. Shi, X., Zhou, K., Tong, Y., Desbrun, M., Bao, H., & Guo, B. (2008). Example-based dynamic skinning in real time. *ACM Transactions on Graphics*, 27(3), 29:1–29:8.
28. Teran, J., Sifakis, E., Irving, G., & Fedkiw, R. (2005). Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 181–190.
29. Kim, J., & Pollard, N. S. (2011). Fast simulation of skeleton-driven deformable body characters. *ACM Transactions on Graphics*, 30(5):121:1–121:19.
30. Müller, M., Dorsey, J., McMillan, L., Jagnow, R., & Cutler, B. (2002). Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002, pp. 49–54.
31. Müller, M., & Gross, M. (2004). Interactive virtual materials. In *Proceedings of Graphics Interface*, 2004, pp. 239–246.
32. Sederberg, T. W., & Parry, S. W. (1986). Free-form deformation of solid geometric models. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 1986, pp. 151–160.
33. Nesme, M., Kry, P. G., Jeřábková, L., & Faure, F. (2009). Preserving topology and elasticity for embedded deformable models. *ACM Transactions on Graphics*, 28(3).
34. Pixar. RenderMan, <http://renderman.pixar.com/>
35. NVidia. Mental Ray, <http://www.nvidia-arc.com/mentalray.html>
36. Lee, S.-H., & Terzopoulos, D. (2008). Spline joints for multibody dynamics. *ACM Transactions on Graphics*, 27(3):22:1–22:8, 2008.
37. Pronost, N., Sandholm, A., & Thalmann, D. (2010). Correlative joint definition for motion analysis and animation. *Computer Animation and Virtual Worlds*, CASA 2010 Special Issue, 21, 183–192.
38. The ultimate human model data set. cgCharacter. <http://www.cgcharacter.com/ultimatehuman.html>
39. Pratscher, M., Coleman, P., Laszlo, J., & Singh, K. (2005). Outside-in anatomy based character rigging. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 329–338.
40. Pronost, N., Sandholm, A., & Thalmann, D. (2011). A visualization framework for the analysis of neuromuscular simulations. *The Visual Computer*, 27(2), 109–119.
41. Delp, S. L., & Loan, J. P. (1995). A graphics-based software system to develop and analyze models of musculoskeletal structures. *Computers in Biology and Medicine*, 25(1), 21–34.
42. Damsgaard, M., Rasmussen, J., Christensen, S. T., Surma, E., & de Zee, M. (2006). Analysis of musculoskeletal systems in the anybody modeling system. *Simulation Modelling Practice and Theory*, 14(8), 1100–1111.

43. Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., et al. (2007). OpenSim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, *54*(11), 1940–1950.
44. Fernandez, J. W., & Pandy, M. G. (2006). Integrating modelling and experiments to assess dynamic musculoskeletal function in humans. *Experimental Physiology*, *91*(2), 371–382.
45. Scheys, L., Desloovere, K., Spaepen, A., Suetens, P., & Jonkers, I. (2011). Calculating gait kinematics using MR-based kinematic models. *Gait and Posture*, *33*(2), 158–164.
46. Assassi, L., Charbonnier, C., Schmid, J., Volino, P., & Magnenat-Thalmann, N. (2009). From MRI to anatomical simulation of the hip joint. *Computer Animation and Virtual Worlds*, *20*(1), 53–66.
47. Gilles, B., Moccozet, L., & Magnenat-Thalmann, N. (2006). Anatomical modelling of the musculoskeletal system from MRI. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006, pp. 289–296.
48. Peeters, P., & Pronost, N. (2013). A practical framework for generating volumetric meshes of subject-specific soft tissue. *The visual computer*, pp. 1–11.
49. Taubin, G. (1995). Curve and surface smoothing without shrinkage. In *5th International Conference on Computer Vision*, 1995, pp. 852–857.
50. Vasavada, A. N., Li, S., & Delp, S. L. (1998). Influence of muscle morphometry and moment arms on the moment-generating capacity of human neck muscles. *Spine*, *23*(4), 412–422.
51. Lee, S.-H., & Terzopoulos, D. (2006). Heads up!: Biomechanical modeling and neuromuscular control of the neck. *ACM Transactions on Graphics*, *25*(3), 1188–1198.
52. Si, H. (2013). TetGen: A quality tetrahedral mesh generator and a 3D delaunay triangulator. <http://tetgen.berlios.de>
53. Oudot, S., Rineau, L., & Yvinec, M. (2005). Meshing volumes bounded by smooth surfaces. In *Proceedings of 14th International Meshing Roundtable*, 2005, pp. 203–219.
54. Zaharescu, A., Boyer, E., & Horaud, R. P. (2007). Transformesh: A topology-adaptive mesh-based approach to surface evolution, vol. II. In *Proceedings of the 8th Asian Conference on Computer Vision*. LNCS, *4844*, 166–175.
55. Tan, J., Turk, G., & Liu, C. K. (2012). Soft body locomotion. *ACM Transactions on Graphics*, *31*(4), 26:1–26:11.
56. Clayton, H. M., & Schamhardt, H. C. (2001). Measurement techniques for gait analysis. *Equine locomotion*, pp. 55–76. London: W.B. Saunders.
57. Nigg, B. M., Herzog, W., & Wiley, J. (1999). *Biomechanics of the musculo-skeletal system*, vol. 2. New York: Wiley.
58. Stokes, I. A. F., Henry, S. M., & Single, R. M. (2003). Surface EMG electrodes do not accurately record from lumbar multifidus muscles. *Clinical Biomechanics*, *18*(1), 9–13.
59. Allard, P., Stokes, I. A. F., & Blanchi, J. -P. (1995). *Three-dimensional analysis of human movement*. IL: Human Kinetics Champaign.
60. Mayagoitia, R. E., Nene, A. V., & Veltink, P. H. (2002). Accelerometer and rate gyroscope measurement of kinematics: An inexpensive alternative to optical motion analysis systems. *Journal of Biomechanics*, *35*(4), 537–542.
61. Geijtenbeek, T., & Pronost, N. (2012). Interactive character animation using simulated physics: A state-of-the-art review. *Computer Graphics Forum*, *31*(8), 2492–2515.
62. Weinstein, R., Guendelman, E., & Fedkiw, R. (2008). Impulse-based control of joints and muscles. *IEEE Transactions on Visualization and Computer Graphics*, *14*(1), 37–46.
63. Grzeszczuk, R., & Terzopoulos, D. (1995). Automated learning of muscle-actuated locomotion through control abstraction. In *ACM SIGGRAPH Papers Conference Proceedings*, Los Angeles, CA, USA, 1995, pp. 63–70.
64. Hase, K., Miyashita, K., Ok, S., & Arakawa, Y. (2003). Human gait simulation with a neuromusculoskeletal model and evolutionary computation. *The Journal of Visualization and Computer Animation*, *14*(2), 73–92.
65. Murai, A., & Yamane, K. (2011). A neuromuscular locomotion controller that realizes human-like responses to unexpected disturbances. *International Conference on Robotics and Automation (ICRA) 1*, 2–3, 1997–2002.

66. Wang, J. M., Hamner, S. R., Delp, S., & Koltun, V. (2012). Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics*, 31(4), 25:1–25:11.
67. Cani-Gascuel, M.-P., & Desbrun, M. (1997). Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3(1), 39–50.
68. Chadwick, J. E., Haumann, D. R., & Parent, R. E. (1989). Layered construction for deformable animated characters. *SIGGRAPH Computer Graphics*, 23(3), 243–252.
69. Moccozet, L., & Magnenat-Thalmann, N. (1997). Dirichlet free-form deformations and their application to hand simulation. In *Computer Animation '97*, 1997, pp. 93–102.
70. Nedel, L. P., & Thalmann, D. (1998). Modeling and deformation of the human body using an anatomically-based approach. In *Proceedings of Computer Animation '98*, 1998, pp. 34–40.
71. ArtiSynth. A 3D biomechanical modeling toolkit for physical simulation of anatomical structures. [www.artisynth.org](http://www.artisynth.org)
72. Maas, S. A., Ellis, B. J., Ateshian, G. A., & Weiss, J. A. (2012). FEBio: Finite elements for biomechanics. *Journal of Biomechanical Engineering*, 134(1), 011005.
73. Allard, J., Cotin, S., Faure, F., Bensoussan, P. J., Poyer, F., Duriez, C., Delingette, H., & Grisoni, L. (2007). SOFA: An open source framework for medical simulation. In *Medicine Meets Virtual Reality (MMVR) 15*.
74. COMMIT. (2013). Virtual worlds for well-being, Dutch Research Project. <http://www.commit-nl.nl/>