A muscular rig for smooth skinning in Autodesk Maya

Pontus Björkman
June 2007

Thesis, 10 points, C level
Computer Science

Creative Programming
Supervisor/Examiner: Sharon A Lazenby
Co-examiner: Torsten Jonsson
A Muscular Rig for Smooth Skinning in Autodesk Maya

by

Pontus Björkman

Department of Mathematics, Natural and Computer Science
University of Gävle
S-801 76 Gävle, Sweden

Email:
pontusbjorkman@hotmail.com

Abstract
The limitations of the default skinning methods in Autodesk Maya can be compensated for when seeking realistic skin deformations of a human being. The main focus is put on the construction of an example muscular rig in Autodesk Maya, and an explanation of what criteria the rig is supposed to fulfill. From a merged three dimensional and artistic view, the components of the rig is discussed and evaluated to establish an understanding of how the system needs to work in order to produce realistic skin deformations. The research is performed via design and creation experiments on the various parts of the system, and even though the initial theory has met its demands, some of the practical performance yet abandons a lot to wish.

Keywords: muscles, rigging, deformers, skinning, anatomy
# Table of Contents

1 Introduction ...................................................................................................................................................... 4  
1.1 Problem formulation ................................................................................................................................. 4  
1.2 Purpose .................................................................................................................................................. 4  
1.3 Limitations ............................................................................................................................................. 4  
1.4 Questions ............................................................................................................................................. 5  
1.5 Expected results ................................................................................................................................... 5  

2 Theoretical background .................................................................................................................................. 5  
2.1 Previous research ..................................................................................................................................... 5  
2.2 Kinematics ............................................................................................................................................. 6  
  2.2.1 Joint skeletons ................................................................................................................................... 6  
  2.2.2 Forward kinematics (FK) ................................................................................................................... 6  
  2.2.3 Inverse kinematics (IK) .................................................................................................................... 7  
2.3 Deformers ............................................................................................................................................ 7  
  2.3.1 Lattices ............................................................................................................................................. 7  
  2.3.2 Clusters ........................................................................................................................................... 8  
  2.3.3 Jiggle deformer ................................................................................................................................. 8  
  2.3.4 Wrap deformer .................................................................................................................................. 8  
2.4 Skinning ................................................................................................................................................ 9  
  2.4.1 Rigid binding ................................................................................................................................... 9  
  2.4.2 Smooth binding ............................................................................................................................... 9  
2.5 Influence objects ................................................................................................................................... 9  
  2.5.1 Object binding ............................................................................................................................... 10  
2.6 Constraints ......................................................................................................................................... 10  
2.7 Driven keys ........................................................................................................................................... 11  

3 Method ........................................................................................................................................................ 11  
3.1 Literature study ...................................................................................................................................... 11  
3.2 Description of method ............................................................................................................................ 12  

4 Realization ................................................................................................................................................... 12  
4.1 Kinematic skeleton ................................................................................................................................. 13  
  4.1.1 Problems and solutions ................................................................................................................... 14  
  4.1.2 IK ................................................................................................................................................... 16  
4.2 Polygonal skeleton ................................................................................................................................. 16  
  4.2.1 Problems and solutions ................................................................................................................... 17  
4.3 Modeled muscles ................................................................................................................................. 20  
  4.3.1 Choice of geometry type ............................................................................................................... 20  
  4.3.2 Choice of muscles to use .............................................................................................................. 20  
  4.3.3 The modeled tissue ....................................................................................................................... 20  
  4.3.4 Constraining the geometry ........................................................................................................... 21  
  4.3.5 Mirroring the skeleton .................................................................................................................... 22  
  4.3.6 Mirroring the muscles ................................................................................................................... 23  
  4.3.7 Driven Keys .................................................................................................................................. 23  
4.4 Skinning ................................................................................................................................................ 24  
  4.4.1 Wrap skinning ............................................................................................................................... 24  
  4.4.2 Influence objects ........................................................................................................................... 26  
4.5 Problem areas ...................................................................................................................................... 26  
  4.5.1 Weight painting ............................................................................................................................ 26  
  4.5.2 Jiggle deformer .............................................................................................................................. 26  
  4.5.3 Kinematic instability ..................................................................................................................... 26  
  4.5.4 Packed muscles ............................................................................................................................ 27  

5 Results .......................................................................................................................................................... 29  
5.1 Result analysis ...................................................................................................................................... 29  
  5.1.1 Usability ......................................................................................................................................... 29  
  5.1.2 Expected result ............................................................................................................................... 30  
5.2 Evaluation of method ............................................................................................................................ 30  
5.3 Reflections ............................................................................................................................................ 30
5.3.1 Future research .............................................................................................................. 30

6 Conclusions .................................................................................................................... 31

7 References ...................................................................................................................... 32
1 Introduction

In most cases when we are in a three-dimensional (3D) graphics application and wish to animate a character, it will be attached to a skeleton where we control or manipulate the modeled geometry in question. In Autodesk Maya, there are two separate solutions for making this connection between the geometry and the skeleton, rigid binding and smooth binding. Both of these techniques exhibit individual benefits and disadvantages.

1.1 Problem formulation

The two methods that are available in Autodesk Maya for connecting a modeled geometry to a skeleton are by themselves insufficient, when the need for more advanced animation, level of detail, and realism is essential.

1.2 Purpose

The goal of this research project is to construct and explain a realistic rig of muscles, sub dermal fat and tendons as they are working concurrently with both a modeled skeleton mesh and the joint hierarchy skeleton of a human character. A combination of these representations should be able to simulate realistic deformations of an outer skin layer. The research will also cover animation aspects of the created rig; what methods causes various muscle groups to work in motion, what tweaking configurations that should be utilized, and how the rig concretely should be applied to an object that we wish to deform or animate.

The reason for choosing this topic of research originates in a genuine fascination for artistic anatomy and musculature, an interest which hopefully can be passed on to future students in various 3D contexts.

1.3 Limitations

Despite the fact that I am aware of the magnitude of work that will be required to bring all of the human muscle system to life, the goal will be to cover a majority of the human body, which are the legs, torso and arms. I have chosen not to include the head, hands and feet as these parts offer an unrealistic amount of work considering the main goal of a realistic representation of human muscles in the given time frame. While this research aims at a visually acceptable result in animation, I will not cover area of topics in detail that strictly belongs to biology and mechanics, such as muscles' accurate, volumetric attributes or muscle fibers' individual characteristics, and distribution inside a muscle. Such data is required only if a correct simulation of physical motion and the forces, which drives the muscles and thereby the body, are to be made. Considering the artistic and visual aspects, the muscle does not provide any type of purpose in the motion itself. Instead they contribute with deformation abilities of the skin in an automated and dynamic manner. Nor will the aspects of texturing the automated deformations of the skin be accounted for, since this is a chapter by itself.
1.4 Questions

- What type of geometry is the most suitable for representing muscles respective deformations in motion?
- What techniques will be used to bind the muscle geometry to the modeled skeleton?
- What methods should be utilized to attach the modeled skin layer to the final muscle rig?
- In what manner do the overall dynamics of real muscle work, and how will this be simulated in the deformations of the muscle tissue in the rig?
- In what way is the three types of tissue (muscles, fat and tendons) most effectively modeled and simulated without using too many types of geometry or unique objects?
- Will the rig be usable from a calculation perspective, or will it be to advanced and heavy for proper and comfortable usage? In which areas would it be possible to optimize the rig?

1.5 Expected results

I expect the muscular rig to be practically usable from a computational aspect as well as to serve as an example of how muscles can be arranged and administrated for a rig, with future students and other animators in mind.

2 Theoretical background

2.1 Previous research

A great deal of the earlier research that has been conducted on areas which for example considers muscles, skin deformation, and animation within the 3D sphere generally tends to direct focus on ways to bypass precisely what I plan to do in this project [2][4]. Previous methods aim at ways to simulate qualified approximations of skin deformations purely with mathematical algorithms as their tools. Anatomically correct muscle simulations often tend to be quite computationally heavy, which is a genuine problem at occasions when real time interaction with the characters is essential. Applications similar to this is for example ultra realistic, extremely high resolution characters in feature film making where the animators have to view the animation in virtually real time to be able to make realistic and artistic decisions. Another example is the gaming industry where hardware rendering always has to be optimized even for limited hardware situations like desktop computers or video game consoles [2]. Wang and Phillips [2] utilizes in their model so-called multi weight enveloping combined with a “training program” to replace the computationally expensive muscular system. The deformation of the animated character is then helped by numerous pre modeled extreme poses, which assists the mesh to preserve an artistically correct look. However, what is important to stress is that the rig that will be
presented in this project is built with realistic skin deformations in mind, and not to produce as optimal real time interaction as possible.

Another well documented area within the 3D visualization realm of muscles and neighboring topics is the medically oriented research. Parts of the research are dedicated to the generating of models of muscles, tendons, and fatty tissue in an automated manner from CTR-scans or conserved, physical cross-sections of real corpses or cadavers. When it comes down to medical or ergonomic research, it is often of great importance that the muscles and surrounding tissue is correctly represented with volumetric properties and fiber distributions [1] [5].

Should we want to access research reports that is similar to what will be presented here, we have to look a bit further back in time, for example similar to Willhelms and Gelder [3], who discussed the use of ellipsoids placed at fix positions across joints on a skeleton to represent dynamic musculature. Primly, it will be this technique that builds the foundation for the work presented here.

2.2 Kinematics

In physics, the term “kinematic” deals with the description of an objects motion without taking any possible influences into consideration.

2.2.1 Joint skeletons

The basis of the kinematic system in Autodesk Maya is the joint hierarchy structure. The on-screen skeleton is only a visualization of the parent - child relationship between separate joints’ transformation nodes [7]. A joint in itself is simply a transformation node with a possible relationship [15].

The visual, arrow-formed bone which automatically extends between every two joints with a direct parent – child relationship has no other specific use than showing in what direction the relationship is set [7].

2.2.2 Forward kinematics (FK)

The most basic way to operate a created skeleton chain is by forward kinematics. The posing consists simply of the rotation of individual joints, and it is important to note that the posing very seldom includes stretching of the distance between two joints. Because of the parent – child relationship in the joint chain, all child joints connected to a rotated joint will follow in the parent joints rotation axis.

The benefit of using FK is total control of all individual joint rotations. The drawback is the difficulty to make goal oriented motions, which is more time consuming [14].
2.2.3 Inverse kinematics (IK)

When inverse kinematics is used on a skeleton an IK-handle is added over the part of the joint chain that will be affected. The IK-handle spans over all the chosen joints and rotates them equally as the handle is moved, creating the goal oriented motion which is hard to achieve by FK.

By default, the IK-handle comes with two separate solvers which individually affect the goal result slightly different. When the IK-handle is moved, the IK-solver calculates how the affected joints should be rotated to reach the new position of the handle. The ikSCsolver (Single Chain) calculates both the individual end rotations and the overall joint chain end rotation towards the IK-handle. Whereas the ikRPsolver (Rotation Plane) extracts the overall rotation from a separate control plane, which determines how the joints relate to each other. If correctly set up, the ikRPsolver generally tends to produce a more predictable line of motion [7].

A third type of IK-solvers is the spline IK. It differs from the previously mentioned IK solvers because, the IK-handle is not movable. When the IK is set up, a NURBS curve is created along the joint directions. The calculated motion goal for the joint rotation will now instead be to follow the curve as it is manipulated [7].

2.3 Deformers

Deformers are a category of tools in Autodesk Maya, which allows the user to manipulate large quantities of geometry control points on an intuitive basis with a smooth deformation result. The alternative to deformers would be to manually move every vertex by hand, a truly tedious task.

2.3.1 Lattices

A lattice is created as two separate main nodes, one base node and one free form deformation node (FFD) [7]. The visual “lattice” in itself consists of any chosen number of cross sections in all three axes. Inside the FFD-node, a number of cubical divisions are created out of the axial cross sections. Each of these quadrants influences all vertices or other types of geometry control points (on the object to which the lattice is assigned) within themselves. This behavior enables any number of geometry control points to be manipulated by only the eight points, which creates the local lattice quadrant. On the object in question, the FFD-lattice’s influence is derived from its relation to the lattice base node. The relative distance between the base node and a FFD point determines the deformation created on the object inside the lattice. By default, any part of the object geometry that is outside of the base node quadrant is unaffected by the lattice [13].

The lattice control points individually or collectively can be configured to act beneath clusters, jiggles or even other lattices. They can also be soft body modified or be skinned to a skeleton to indirectly skin the geometry which it controls [13].
2.3.2 Clusters

A cluster is the most basic deformer. All vertices, which are affected by a deformer, is said to be a member of that node. This membership is editable and vertices can be added or removed from the affected list. Like some other deformers the cluster affects each vertex by a weight, which determines how much influence the cluster has over a specific vertex. Because a vertex only has a position and no rotation or scale attributes, the cluster provides the membership vertices with a relative pivot to translate, rotate and scale around. This is important as it not only works at volumetric, such as polygons or NURBS, geometry control points but also for spline vertices or lattice control points. At a higher level of component editing, this enables vertices to be parented to other objects via the cluster, which is not possible by default [12].

2.3.3 Jiggle deformers

A jiggle deformer works somewhat different than other deformers as it does not have a transform or shape node, therefore there is no “physical” object to move around in the 3D-scene. The effect the deformer has on an object is also different since it is a secondary animation, which means an animation that is not directly key-able (with key frames) but is produced automatically by Autodesk Maya from pre made settings. To manipulate this type of animation is to manually and numerically alter its control attributes [7][14].

The jiggle deformer is in a non-physically accurate way to be able to automatically simulate jiggle effects in animated objects with a static relationship to dynamic forces such as gravity [14]. The jiggle node filters the object transformation and/or component transformation thru its attributes and produces an extension of the key-able animation.

2.3.4 Wrap deformers

A wrap deformer is in many ways similar to a lattice in its usage. It is built out of two main nodes, a deformer node and a base node. The deformers nodes’ transform, the component attributes are compared to the base nodes’ and the deformation effect is derived from the possible difference between the two. Any deformable object (NURBS, polygons or splines) can be used as a wrap deformer, which means that a wrap deformer is an already existing object which affects another object with its attributes. Since objects generally tend to have a cage-like structure, the result of using wrap deformers is comparable to lattices. However, by default the affected object does not have to be inside of the base nodes shape for the effects to take place, which is the case of lattices. This is due to that wrap deformers work with drop of distances from its components instead of volumetric quadrants [7][11].
2.4 Skinning

Skinning is the process of binding geometry to a skeleton, which enables kinematic control of the object. In Autodesk Maya there are mainly two techniques for skinning: rigid binding and smooth binding.

2.4.1 Rigid binding

With rigid binding a vertex can be influenced by only one joint in the skeleton hierarchy, which leads to stiff and straight deformations of the skinned geometry. The stiffness is due to an attribute called partitioning, which forces a skinned vertex to be a member of only one joint. This can also be true for all deformers. Since a vertex can only be influenced by a single joint it inherits all of the joints’ transformation attributes straight off, which yields coarse edges as joints are transformed. In cases when the stiffness is not wanted there is the option to use flexors to control the deformed geometry to a desired shape and softness [7]. Although, this system permits little room to automate the deformation process where a great deal of it would have to be manually hand animated.

2.4.2 Smooth binding

With smooth binding, a vertex to the contrary can be influenced by more than one joint at the same time since it is not part of any partition. Each vertex is assigned an influence weight, which is distributed across the affecting joints. The sum of all joint weights, which influences the vertex always adds up to 1. Each and every joint can individually affect the vertex with a weight value between 0 and 1, resulting in an interpolation of the respective joint’s transformations over the vertex. The weights are post-editable and it is therefore possible to rearrange the weight values to better produce the desired deformation result [7]. However, the disadvantage of this technique is that it has problems with the preservation of the skinned geometry volume. The default deformation procedure result of the animated object tends to appear like soft rubber. This is, though, compensated with influence objects.

2.5 Influence objects

The individual problems of the two skinning methods quickly become evident when it comes to animating more advanced or photorealistic characters. This is because none of the techniques by default can carry out the tasks of simulating the desired deformations in automated fashion. However, the technique that is the most dynamic and flexible is the smooth binding, because it has the ability to have an influence object added to it. These objects can replace the weighting function of the joint skeleton structure either at object or at component level. An influence object can be any type of object in Autodesk Maya, such as NURBS, polygons, splines or locators. They can be placed at points in or on the geometry where a higher level of control and interactivity of the deformations is desired than what is provided by the initial joint setup. The greatest benefit of the influence objects is their ability
to preserve shape and volume as the skin weight system takes the whole object into consideration as it recalculates the weights for the vertices in the affected area [7].

2.5.1 Object types

Based on the level of influence, an object is affecting the skinned geometry, where various things can be achieved. If the object is affecting on the component level, it is possible to add deformers to the influence object and thereby rig automated animations. Also, at the component level the object is in-depth editable with driven keys [7].

The various types of objects, which can be used as influence objects that individually possesses abilities that makes them more or less appropriate for, in this case, muscular representation.

- Polygons - Polygons are simple to administrate, uncomplicated to model with and have volume preserving abilities. Due to polygons’ inherit edge features their main disadvantage is their lack of detail at low resolutions. To be able to fine tune close-up mesh deformations, there has to be a sufficient number of tweak-able polygons, which soon tends to be computationally expensive if the influence objects is set to affect at component level.
- NURBS - Objects created out of NURBS surfaces are by nature soft edged even at a low number of isoparms, and therefore produces a smooth result within the internal deformation of the influence object. Similar to polygons, the NURBS objects preserve the volume of the deformed skin surface in desired fashion. Within research areas that pertain to various forms of anatomical simulations for medical purposes, NURBS or NURBS-like surfaces are commonly utilized when muscles are to automatically generate.
- Splines and locators - The central use of these in skinning is to control single details. As neither of them has any particular volume, where they are not well adapted to preserve it.

2.6 Constraints

The most basic way to constrain one object to another object is by parenting it, where the first object is beneath the second object in the hierarchy, thus forcing the child to inherit the parent’s transformation attributes. In Autodesk Maya, there is however a number of other functions for building similar object relationships without forcing one object to be child to the other. This is achieved by adding a constraint node to the object, which is supposed to be affected by the other objects’ transformation attributes.

The constraints are divided into a number of option categories depending on what type of constrain they apply to the object. Each category comes with numerous precision options and/or attributes on exactly how to set up the relationship, thus making the tools accurate and useful if understood correctly.
Examples of constrain categories are Point, Aim and Parent.

- The point constrain has the ability to lock the affected objects translation values to the target object, thus moving the affected object together with the target without rotating it. Like with all constraints the object axes which should be constrained can be freely chosen [10].
- The parent constrain allows the affected object to be locked in all axes to both the translation and rotation of the target [10].
- The aim constrain works slightly different by comparing the affected objects local rotation axes with the global coordinates and thereby having the ability to derive an aim against the target object, therefore wherever the target is located, the affected object always points with the selected axis towards it [10].

2.7 Driven keys

Most of the animations made in Autodesk Maya are time oriented, therefore the objects move, rotate, spin, or deform according to key frames set on a time line. However, on some occasions an object’s attributes preferably should be transformed due to changes in another object’s attribute values rather than keys in the time line. Technically, a driven animation works in the same way as time animation. Both are built on a two-dimensional axes chart with key frames to mark specific value changes in both axes and various types of curves to derive the value change in-between set key frames. The difference between the two are that all time animation has the time line as base axis while a driven animation can have any key-able attribute from any type of node [7].

3 Method

3.1 Literature study

The literature that has been read and assimilated for this research project mainly regards three aspects of information which needed to be addressed in order to complete the realization of the research and this report. The information areas are:

- Previous research
- Autodesk Maya tool specifications
- Anatomical theory

For the Previous research section of the research report mainly scientific articles from the ACM database are used. The Autodesk Maya tool specifications information are gathered from the Autodesk Maya 8.5 reference manual and various video tutorial DVDs. Information on muscular and skeletal appearance and motion are collected essentially via literature regarding artistic anatomy.
3.2 Description of method

Since the result of this research project is an “artifact,” the methodology falls under the design and creation area. The product is supposed to solve a particular problem, and thus requires construction planning, a construction phase, testing, and evaluation.

The planning or the idea of the product is purely theoretical, while the construction phase is built on tool experimentation and logged trial and error methods. The process of judging whether the research project manages to solve what is stated in the problem formulation is quite subjective. Since the muscular rig might not perform as expected, the minimum criteria should produce a better deformation result on an automated basis than the default settings. If the rig produces natural looking, muscular deformations, it can be called a success. Thus the testing of the rig only includes bending, twisting, and moving the body and observing the deformations that occur during the process.

4 Realization

In the real human body (and for most animals), physical motion is achieved by skeletal muscles which contracts and relaxes in complex patterns in order to draw, lift and release itself in the desired direction. When walking, the muscles practically allow the center of gravity of the body to fall forward to gain space and then catching itself with the legs before falling over. All of these muscular activities are basically controlled at a voluntary state of mind, but can also be maintained at a subconscious level [16]. Thus, walking and keeping the body balanced does not have to be consciously processed. However, in the world of artistic 3D animation, muscles do not serve the same purpose as in the real world. In 3D animation, the muscles have absolutely nothing to do with the motion of the joint hierarchy structure, generally called the “skeleton.” Because this joint skeleton drives itself and fully is its own master, it is, thus, called a kinematic skeleton. Instead, the reason to have properly functioning muscles in combination with the kinematic skeleton is due to the limitations of the default skinning techniques in Autodesk Maya. A dynamic setup of modeled geometry, posing as muscles, automatically should be able to compensate for limitations of the normal skinning process, thus providing acceptable skin deformation results on an automated basis. In order to make this process to work properly, a lengthy line of options, tools and stages in Autodesk Maya has to be able to work alongside each-other in a smooth manner.

The planned workflow of how the rig will be built can be summarized to:

- Creating a polygonal skeleton
- Creating a kinematic skeleton
- Creating polygonal muscles
- Constraining the polygonal skeleton to the kinematic skeleton
- Constraining the polygonal muscles to the skeletons
- Implementing driven keys where needed
- Adding a skinned, wrap base geometry to the kinematic skeleton
- Adding the muscles as influence objects and painting the weights
- Adding the jiggle deformers to the largest muscles
- Adding a high-resolution polygonal skin by wrap skinning

4.1 Kinematic skeleton

The foundation of the whole rig is the joint hierarchy structure, which in large is the only mechanism that will control the entire body. The body will follow the joint chain precisely in its movements, as everything in various levels is locked to it.

The joint skeleton is set up more or less according to standard procedures and could in fact be much more advanced. However, for the purpose of this research project the level of control is enough, but could for example also work with a Full Body IK (FBIK) without any problems. There are, though, some minor conditions in this rig that is special in order to work properly with a realistic bone and muscle framework.

Figure 1: The kinematic skeleton controls the entire rig.
4.1.1 Problems and solutions

One issue that is extremely delicate is the positions of the joints in the kinematic skeleton compared to the ones of the modeled skeleton (see below). If the kinematic joints are deficiently placed the bending of the modeled skeleton’s limbs will be equally faulty, thus impairing the overall end deformation by displacing the muscles.

Another matter is the spine, which in reality bends a small amount. When the body is rotated the main rotation point is the pelvis, not the spine. This is true also for when the body is bending heavily forwards or backwards. The small bend that occurs in the actual spine area is located in the lower parts beneath the ribs, comprising the first six or seven vertebrae. The spine is more or less stiff in the rib area, allowing very little rotation before the neck begins. Since a spline IK with a simplified control-curve is used to manipulate the spine shape, it posed a problem as it deformed the whole vertebrae length and not just the base. The solution was to remove all kinematic joints in the section of the spine that is supposed to be rigid, replacing them with a single joint stretching the entire length, thus rendering the rib area stiff.

Figure 2: Correct position of the hip joint.
Figure 3: Correct position of the shoulder joint.

Figure 4: Incorrect position of the knee joint.
4.1.2 IK

The choice of IK solvers for the kinematic skeleton was made by specific pre testing in the relevant areas. The SC solver was favored over the RP solver for the arms and legs. Although the RP solver is generally more stable and predictable, it has difficulties dealing with extreme poses due to the devious position of the pole vector, which often results in flipping of the respective ball-joint. Both the spine and the neck are controlled by one spline IK each, which meets more or less all requirements for that particular branch of the system.

4.2 Polygonal skeleton

In the beginning of the research and construction of the model, the importance of the modeled skeleton was somewhat overlooked, thought of only as a guide for where to put the muscles on the kinematic skeleton. However, the modeled skeleton has become, in itself, just as important as the muscles or the kinematics. It is a great deal more intuitive and natural procedure to actually bind the muscles to the modeled skeleton then to the kinematics itself. The volumetric feel and visual feedback of the modeled skeleton as it moves along the kinematic skeleton is simply superior. Therefore, it is of utmost importance that the polygonal skeleton is realistically modeled, where it strictly correlates with artistic anatomical studies.
Most of the bones in the polygonal skeleton are parent-constrained directly to the corresponding bone in the kinematic version. Although, there are a few exceptions where individual bones have problematic movement as the skeleton is animated and thus requires special solutions to how they are attached to the kinematic joints. These bones are the shoulder blades (scapula), forearm bones (radius and ulna) and the knee caps (patella).

Figure 6: The pieces of the polygonal skeleton are a template for muscular positioning as well as an influence object in later stages of the construction.

4.2.1 Problems and solutions

The scapula is loosely attached to the backside of the torso with large, flat muscles [9]. As the arm rotates the scapula moves with it in patterns that empirically has been difficult to mimic with regular constraints, because of the nature of the constraints themselves. The scapula needs to be able to inherit two separate rotations to move correctly, one from the chest and one from the rising of the clavicle bone, which does not seem to be possible by default. The solution was to parent the scapula to the base clavicle joint, letting it inherit all transformations. This, however, renders a faulty rotation of the scapula as the arm is lifted, something that is compensated for by driven keys. The rotation of the shoulder joint drives the rotation of the scapula into a correct position.

When the forearm is rotated the lateral radius bone crosses the medial ulna. The radius rotates to some extent along its length axis while the ulna’s
rotation is locked to the elbow. To achieve the crossing behavior, a type of dual pivot system is required. It is built by placing locators in each end of each bone and then parent-constraining the radius and the ulna to the locators positioned in the elbow. By aim-constraining the elbow locators towards the locators in the wrist, it is guaranteed that the bones always will point towards them, which is important as the wrist joint is rotated. To make the bones rotate, the wrist locators is parent-constrained with an offset to the wrist joint, forcing the locators to revolve the joint at a fixed distance.

The knee cap is loosely connected to the leg via the frontal quadriceps at the top and by the knee ligament at the bottom [9]. As the knee is bent the cap is pulled slightly downwards by the knee ligament without rotating. The easiest way to attain an acceptable result is to solve it by first parenting the cap to the knee joint and then implementing driven keys, so that the rotation of the knee joint drives the position and rotation of the knee cap.

![Figure 7: The dual pivot system for the radius and ulna.](image-url)
Figure 8: As the shoulder is lifted, the scapula is rotated around its pivot, which is positioned in the upper corner near the spine.
4.3 Modeled muscles

4.3.1 Choice of geometry type

Since the end result of using either polygons or NURBS for muscle representation does not diverge notably, the choice of which one to employ for this matter comes down to personal opinions. Due to personal experience polygons was favored over NURBS.

4.3.2 Choice of muscles to use

The muscular thickness in the human body is in many places compositied out of multiple layers of thinner muscle tissue. Since this research project aims at a visually acceptable but not a physiologically correct result, it is only the surface shape of the layered muscle patches that is of any interest. As long as the muscle representation has a combined, correct volume, many muscles can be merged to act as single units. The same applies for strands of thinner muscles that run tightly alongside each other without separating as their respective joints move. Thus, this makes it possible to extensively rationalize the number of objects that has to be calculated in, foremost, the torso but also to some degree in the arms and legs.

4.3.3 The modeled tissue

The process of creating the polygonal muscles is a straightforward procedure as the representations is supposed to accurately mimic the real muscular shapes. Attributes that has to be taken into consideration when modeling are polygonal simplicity, volumetric generalization, and muscular position and behavior.

For the rig to work effectively when skinned, it is important to have the polygon count in mind as each vertex individually will affect the skinned mesh.

As stated above, a lot of the muscles in the body are positioned beneath other muscular tissue and are therefore redundant when it comes to directly influencing the surface skin. When building the surface muscles the underlying volume needs to be considered so that the overall thickness of tissue can be correctly represented.

As for the modeled bones, it is equally important for the muscles to be positioned correctly. If they are not, the deformation of the muscles will be impaired when rotating the joints.
4.3.4 Constraining the geometry

A key factor in how the rig is performing pertains to the method of how the modeled muscles are connected to the skeletons. The setup has to be able to produce an automatic response in muscular length to the rotation of the muscle’s respective joints and at the same time maintain general, internal muscle direction. The meaning of this is that the muscle has to be able to extend in length, as well as contract in comparison to the original length. To preserve internal direction is to force the ends of the muscle to always be parallel to each other. This is important for all muscles that directly stretch across a joint, which almost all muscles in the arms and the legs do.

To achieve such a result (for the average muscle in either the arms or the legs) by using a lattice, two cluster and four constraints, the lattice controls the deformation of the geometry via its default usage. By trial and error, the setup of the lattice should be set to 4*4*4 local divisions, which render a softer deformation, and have at least 4 divisions on the global Y-axis. The reason for 4 divisions on the Y is due to the adding of clusters contra the local divisions. Since a lattice control point (the same as a vertex) only has position attributes and no scale or rotate options, there need to be a pivot added where these points are connected. If not, automatic control over the points will not be possible. The cluster provides its transformation node to the control points, which makes it possible to constrain the points in the lattice in any desired manner. The two lower rows of control points are added to one cluster and the top two rows to another lattice. As this process is conducted,
it is important that the muscle is rotated where the upcoming deformation direction is aligned with the global Y-axis, allowing the lattice to produce a correct geometry deformation by only altering shape in its Y-axis. The clusters are then aim-oriented towards each other, thus forcing the ends of the lattice to be parallel. By using the clusters, the geometry is oriented back to its original model position on the skeleton. Finally, the clusters are parent-constrained to the each attachment point for the muscle on the skeleton, although only for the translation attributes as the rotation attributes already has an input connection from the aim constraint.

When it comes to the muscle in the torso the workflow is less direct, and every muscle unit needs its own configuration of lattices, clusters and constraints. However, what generally can be mentioned about the setup is that most of the constraints will be of the parent type. There is little use for the aim constraints in this matter as the large, flat muscles do not work in the same way as the ones in the arms and legs. Therefore, it is important to parent constrain all clusters to the skeleton and not use the point constraint, because it will render faulty rotations on the muscle if the body is moved.

![Figure 10: The rigging system contains approximately 100 lattices and 250 clusters.](image)

### 4.3.5 Mirroring the skeleton

The reason to mirror sections of the skeleton instead of just creating the other side is to save time. By mirroring particular sections of the skeleton it is possible to preserve and transfer driven keys to the other side of the body. The method to accomplish this by grouping all of the separate hierarchies
beneath individual group nodes, which permits the use of the negative x mirror operation. The groups are then imported and the hierarchies reattached to the original skeleton.

4.3.6 Mirroring the muscles

To save a great deal of time when setting up the musculature on the skeleton, the wise choice to make is to first create one side of the body completely and then mirror-copying it over to the other side. This, though, is not an intuitive process. The problem area is mainly the constraints and the cluster. When adding the constraints to the clusters, Autodesk Maya locks the clusters’ transformation node to the constraint, thus preventing any type of copying or mirroring to work properly. For the mirroring to work, all of the constraints locking the translation attributes of the clusters have to be removed. As the clusters of a single muscle are controlling all transformation aspects of that geometry, it is the clusters that should be mirrored. However, the clusters drag the lattice with them, which in its turn pulls the muscle geometry. Practically, the clusters needs to be parented into a separate group than the lattice and muscle, otherwise all components will be mirrored, which produces an undesired result. The cluster group is then mirrored by using the negative x operation.

This method, however, produces a transform issue. When the clusters are to be constrained back on the skeleton, a flipping problem will arise. The clusters are under a negative x transform, and when they are parent-constrained to the positive x skeleton, they will flip as the clusters picks up the positive translation attributes. To work around this problem, the clusters can be un-parented from their respective group. This works because Autodesk Maya preserves the negative x-flip outside the group by compensating in the clusters’ translate and rotation attributes. The un-parenting operation thus provides the cluster with only positive attributes while still maintaining position and rotation. This enables normal parent-constraining with the positive x, skeleton geometries. However, bones belonging to the mirrored sections of the skeleton are also under the negative x transform, and thus will work without modifications with the negative x clusters.

4.3.7 Driven Keys

In some muscle areas, the regular lattice deformation will not produce an acceptable result when the joints are rotated. These areas are especially the musculature that stretches across ball-joints, which practically is the shoulder (deltoidus) and the rump (gluteus maximus). Since these muscles must be able to handle extreme positions, special deformation assistance is needed. By implementing driven keys, it is possible to drive the individual positions of the control points in the lattices to form a desired deformation result.
4.4 Skinning

To initiate the skinning process, the cut up, low resolution wrap mesh (see below) is bound to the kinematic skeleton with the default, smooth-skinning tools.

4.4.1 Wrap skinning

When an influence object is added to a smooth bound mesh, it is weighed against all vertices in that skin object. If there is more than one influence object, they will be weighed against each other, including joints, where all possible influence objects have to take every other influence object into consideration to be able to calculate the correct end weight on each skin cluster point. The human body contains approximately 640 skeletal muscles [8][16], and even if, for this purpose, not all of these muscles are accounted for. Therefore, the number of influence items that will be affecting the skin mesh is counted in hundreds. Needless to say, such a number of objects might render a final model unusable due to computational overload.

One technique to ease the burden of calculation is to use wrap skinning, which utilizes wrap deformers to indirectly skin the high resolution mesh to the skeleton and every one of its attached deformation objects. To produce a precise deformation of the high resolution mesh, a low resolution copy of it is made, although high enough in resolution not to lose general detail. This lower resolution copy is then cut into several pieces, and preferably cut at places where as little deformation as possible occurs. Since all of these new pieces can be skinned individually to the skeleton, but with a seamless deformation effect, the workload is a great deal less when adding influence objects. This is because an added influence object only has to be weighed against the polygons of the respective wrap objects pieces, which contains fewer polygons than if the wrap was not cut into pieces. Even more cost reducing is the fact that the influence object is only weighed against the respective other influence objects of that wrap piece, and not the entire set of muscles, which would be the case if wrap deformers were not used.
Figure 11: The wrap base geometry pieces are color coded for easier management when setting up the wrap deformer relationship.

Figure 12: The high resolution mesh that was not implemented due to the failure of the weight painting.
4.4.2 Influence objects

In order to be able to use the modeled geometry as influence objects for the skinned wrap mesh, the level on which the skin cluster influences operate has to be altered. Normally they are affected at object level, and thus respond only to the whole influence objects’ transformation. For the effects to take place when the muscle geometry is deformed via lattices, it is required that the skin clusters obtain the influences also at the component level. This is because the lattices do not alter the muscle object’s transformation nodes, but only the vertex positions.

4.5 Problem areas

Even if there are a lot of features of the muscular rig that do work according to plan, there is however also some unresolved issues. If given more time to continue the research project, all of these problems could be unraveled.

4.5.1 Weight painting

For the entire rig to work adjacent to a high resolution mesh, a proper weighting of the skin cluster influences is critical. If the weights are not properly placed, the deformations of the wrap base geometry will not be smooth, but edgy and loaded with vertex displacement artifacts. Up to date, no proper technique has been found for placing the weights in a correct manner, thus impairing the overall function of the rig extensively.

4.5.2 Jiggle deformers

Depending on how stretched a muscle is by its tendons and how constrained it is by near musculature, the muscle will jiggle in some extent when it is moved. This feature, though, is much more obvious in slow motion movie capture, and there are few muscles that jiggle visibly in normal viewing speed. As time ran short the jiggle system was down prioritized. However, some tests ware run on single muscles, and the system works both when it comes to deforming the muscle and, indirectly, deforming the wrap base mesh. Like the lattices, jiggle deformers act at the component level, and therefore the skin clusters has to be influence-editable at the component level for the jiggles to work.

4.5.3 Kinematic instability

The reason to why the ikSCsolver was favored over the ikRPsolver in the initial tests was that it seemed to be more stabile at extreme joint rotations in the arms. The problem, though, is not the IK solver, but some issue with the setup of the arm joint skeleton itself. When it comes to animation, the ikSCsolver is highly less suitable due to the nature of the solver. Since the ikSCsolver calculates the end transformation of the ikHandle by both translation and rotation, there will be additional animation graphs for the
rotation attributes. For the ikScsolver, this poses a problem when it comes to both stability and the ability to smoothing the animation graphs. Since transformation of the ikRPsolver only produces translation changes, the animation graphs only will contain translation curves, which render a more intuitive result when the curves are to be modified. The matter with the incoherent arm joints still does not have a solution.

4.5.4 Packed muscles

Another cosmetic issue is that when the arms and legs are rotated heavily at the ball joints, the muscles of the upper part of the limbs seems to merge together and become flat in the direction that the limb is rotated. The explanation to this is probably that real muscles respond to internal tissue collision, whilst their 3D modeled counterparts do not. As the limbs are rotated in reality, the muscles are internally pushed along with the rotation. Since the modeled muscles of the rig do not collide with each other, there is no force that commands the geometries to preserve the overall, combined volume of the muscles.

Figure 13: Due to the defective weight painting, when limbs are moved or the body is rotated, a great number of deformation artifacts occur in the skin.
Figure 14: As the arm is positioned towards extreme angles, the elbow joint spins and rotates uncontrollably.

Figure 15: Since the muscles do not collide with each other, they get merged as their respective fastening locations start to overlap.
5 Results

5.1 Result analysis

Considering the goal of the research project, the achieved end result was not a success. The goal was not fulfilled. The reason for this is the failure to paint the skin cluster weights. By not succeeding with the weights, the produced deformations of the wrap base geometry do not pass the default smooth skinning in realistic appearance. However, when comparing the deformation results of the muscular-rigged skin contra the joint-rigged skin, there are major differences. The default smooth binding produces soft deformations but without the ability to preserve volume of the geometry, while the muscular version, to the contrary, yields harsh deformations but keep the volume. Thus, the unmodified weight result of the muscular rig is similar to the default effects of a rigid skin bind, although with a greater potential to it.

If the practical bugs and problems are put aside, the rig in itself is theoretically working according to the construction plans. Most of the muscles are able to correctly stretch and contract in response to the activity of the joint skeleton hierarchy and, in the end, have the ability to affect the skin layers with their flexible shapes.

5.1.1 Usability

While the rig works as planned in theory and fails at the end in practice, there is still the question of the actual usability from a calculation aspect. The rig has been tested on a 64 bit system running on a 2.4 gigahertz Intel Core2 Duo processor, 2 gigabytes of RAM and a 640 megabyte nVidia 8800 GTS graphics card. While the rig by itself runs fluently, though, when adding the 15024 polygon wrap base geometry, the complexity slows the performance to a near halt as the limbs are moved. Since no tests have been run on the fully developed end skin, the calculation time for animating the high resolution mesh can only be estimated. As the high resolution skin contains approximately 240,000 polygons, almost 16 times more polygons than the wrap base, it can safely be assumed that animation interactivity would not be more fluent. The massive amount of calculation workload originates in the fact that all of the circa 100 influence objects are affecting the wrap base geometry on the component level.

There are, though, ways to work around the computational shortcomings. By not displaying the full resolution skins, they do not need to be calculated. As long as all the desired poses deform sufficiently well, the wrap base mesh and high resolution skin can be visually replaced by a low resolution dummy-skin. This geometry is not skinned to the skeletons, but parented in pieces, so that a certain degree of visual feedback will be achieved nonetheless, and still maintaining real time interaction speeds. Once the animations are created the high resolution skins is switched back on.
5.3.1 Future research

The expected result for the research project is partially fulfilled. As discussed above, the expectation that the rig should be practically useful from a calculation aspect cannot be said to be successful. Nor are the deformations acceptable. At the other hand, the muscular setup works in both theory and in practice, and can be said to accomplish its goals, as well as to serve as a raw model for how a similar rig can be created and administrated by other practitioners of Autodesk Maya.

5.2 Evaluation of method

The execution of the intended methodology can from some aspects be criticized. During the construction of the rig, a number of experiments have been conducted in order to figure out the various functions of the system. Many of these tests were initially performed not on the actual rig, but on smaller dummy-systems. The problem with this method is that there is a great deal of parameters and objects that is affecting each other. Since the dummy-systems usually were constructed to test only one new function of the rig, they did not take other functions into considerations. Thus, some of the pre-tested parts were not tested whether they worked together or not. If better implementation tests had been conducted, some later problems might have been avoided. Examples of this are the ikSCsolver versus the ikRPsolver, or the deficient arm joints. Also, at the end of the research, only a few tests were able to take place since the rig was not proficiently finish.

5.3 Reflections

The reason to include a similar muscular rig into a production pipeline would naturally be to solve the same problem that this research project is trying to unravel. Since most productions do not have access to state of the art, in-house solutions like the multi weight enveloping [2], the more general purpose muscular rig is a practical alternative when the need for high-end deformation realism is required. Another area of use could be to create a simplified version, to merge with a ragdoll system for the next generation of computer games. Ragdoll systems needs automatic deformations and cannot rely on pre-fabricated animations. As the demands of model versatility escalates alongside the demand for greater visual experience, somewhere along the road the real time interactivity of specific game characters needs to be enhanced.

5.3.1 Future research

If the research project presented here was to be developed further, there is the possibility to add many more features to it. To start with, all of the deficiencies of the current rig could be solved. Further, hands, feet, and a head could be added to complete the rig anatomically, although, all of these add-ons would pose a major amount of extended labor as the musculature and bone structure of these elements are more advanced in their functions. Another feature that could be added is muscle tensing, so that there is an
option for tensing the muscles without considering any particular joint transformation.

6 Conclusions

The purpose of this research project is to construct, explain, and demonstrate what a realistically set up muscular rigging system for a human being might look like in Autodesk Maya. By first building a framework for all of the later components by modeling a realistic polygonal skeleton, the measurements of the body is defined. As the polygonal skeleton is ready, a second skeleton consisting out of a kinematic joint hierarchy is created. The latter structure will control all of the other components in the end. Next, all of the desired muscles of the body are created out of polygonal shapes, which then are constrained to the skeletons via a system of lattices and clusters. After this, an average resolution polygonal body mesh is cut into several pieces that later is going to act as a wrap deformer for the high resolution skin mesh. The wrap mesh is then smooth skinned to the kinematic skeleton, and all of the muscles are added as influence objects. When all of the influences have been calibrated, the high resolution skin mesh is added to the system via wrap skinning.

Since there were a large number of problems during the realization of the research project, the goal of having a skin mesh that produces realistic and smooth deformations on an automatic basis was not achieved. The feature’s of the rig that was not completely solved in the given time was influence weighting of the skin, the jiggle effect of the muscles and the wrap skinning.
7 References


