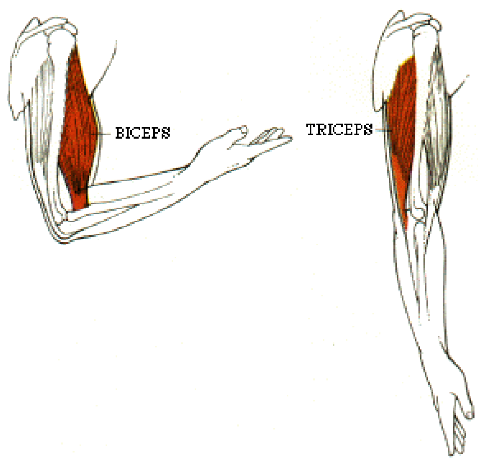
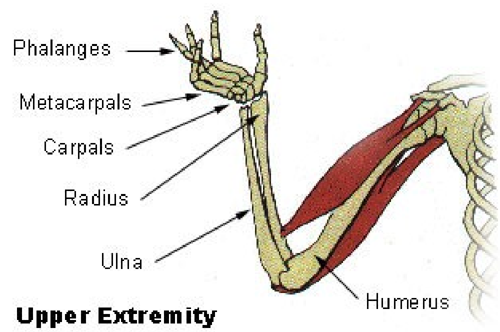
Muscle Deformation Techniques

Jakub Krompolc, creature TD

Muscle Dynamics
(Flexing and Jiggle)

One of the elements that makes creature skin more realistic is the dynamic flexing and spring movement of the muscles.





Main muscle actions related to limbs are <u>flexing</u> and <u>extending</u> (there are other functions but for rigging these two are the most important). From this perspective these muscles are sorted as **Flexor** or **Extensor**. For example biceps is flexor because it folds the arm. Triceps muscle is extensor because it expands the arm to its full length. Imagine the bird wing, as it also folds in or expands to its full potential reach.
This comes from the muscle nature to shorten and as the effect of it to **pull**, as this is the only thing the muscle can do. **It cannot push** or expand itself back to original pose. For this reason creatures have always muscles on both sides to be able to rotate their limbs in both directions.

Machines on other hand, powered by hydraulics, only need a piston with its compressed liquid on one side as it can both pull and push. Still, the articulation and powerfull pistons of hydraulic arm are very similar to bones and muscles of humans, animals and insect. Engineers didn`t have to go far for ideas...

Flexing Setup

Flexing setup can be divided into two parts. **Driver** and **shape**. Driver is a system that could measure speed or length of certain body parts as they change through animation and output values that can drive our character`s geometry. These values are usually between 0-1, which is easy to use in blendShapes or to multiply other effects.
Shape part of the setup produces actual muscle shape animation and is connected to the flexing driver output.
For easy to tweak system it`s important to keep both parts separated. You can in some cases decide to change some heavy shape system for lighter one and this is easier if driver setup is independent.

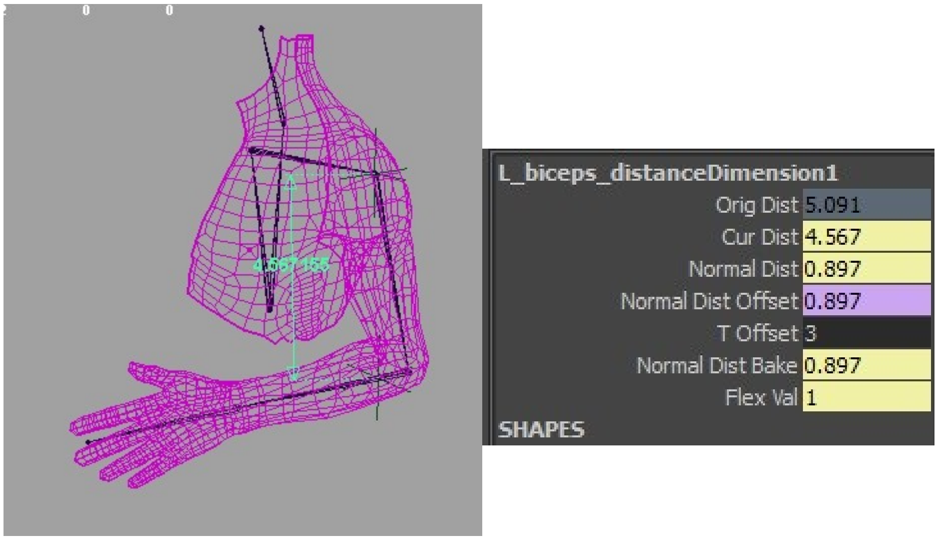
Flexing driver systems

Before describing an automated setup, it should be noted that all flexing of muscles can be also easily **keyframed**. Simply create one slider control per muscle (constrain it to some joint), add a flex attribute of range 0-1 and connect it to your flexing shape or other muscle flex system (sculpt, profile change, etc.).

As **automated** setup, this can be driven usually based on **distance** nodes attached to your skeleton or even rivetted to your mesh (using shape before the change you applied, to prevent cycle) or by measuring **angle** of limb rotation. For angle I would recommend to use angleBetween node or other vector based solution and not to rely on euler values of the joints. The setup of angleBetween pose driver is more described in first chapter about blendShape correctives.
More advanced systems can combine different measuring techniques together.

Important feature of muscle flexing is the **time offset** before the rotation of the leg or arm. A in living organism the muscle flexes first, and then the arm rotates as a result of this. So the driving system also needs to behave as a simulation, evaluating data from the future frames.

This is shown on the example scene below using the same bicep flexing.



Maya 2011 file: 04_muscleDynamics/muscleflex_biceps_timeExp_01.ma

In this case it gets triggered 3 frames in advance (see T Offset attribute). It`s important to bake the value we read in advance into keyframes if we use

getAttr -t time object.attribute

Otherwise the return value would be current value. This is why in this example I have created an attribute on “L_biceps_distanceDimension1” object for baking “normalDistBake”, which is connected to “normalDist” showing normalized distance. In second scene, this is baked and the expression getAttr command reads the time correctly. If animation changes, attribute needs to be reconnected and rebaked again.

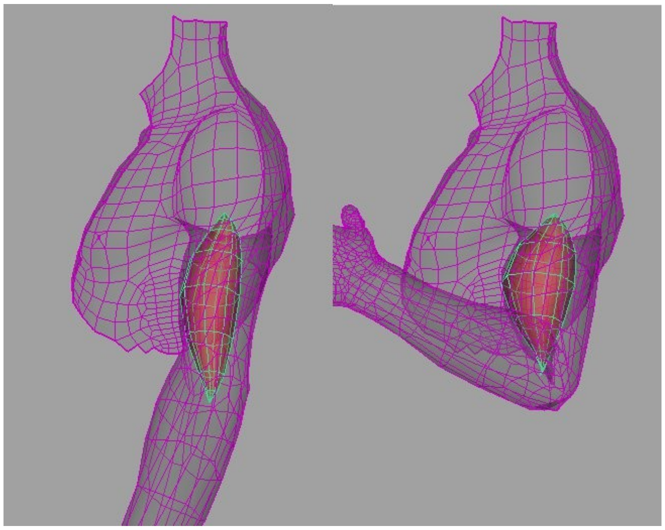
Maya 2011 file: 04_muscleDynamics/muscleflex_biceps_timeExp_02.ma

We could also using expression set currentFrame to future frame time, read the value and go back to current time. But this could have undesired effects on other simulation systems in the scene, for example muscle or skin jiggle.

Other uses of expressions could be measuring SPEED of shortening the muscle and getting the ACCELERATION. Then based on either speed or acceleration value we can again drive the amount of flexing.
For more complex or realistic setups, it`s common to combine results of more measuring systems on connected parts of the body and use the combined result. This is good for secondary effects or for smaller muscles which are effected by main muscles.

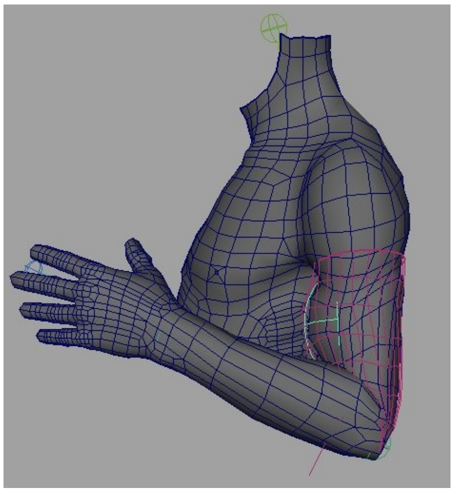
Flexing shape setups

The effect of flexing a muscle can be achieved by different techniques. The most simple is using **helper joints** usually by applying translation and scale. Using these was basically covered in Helper Joints chapter, so we will focus on shape-based setups here.



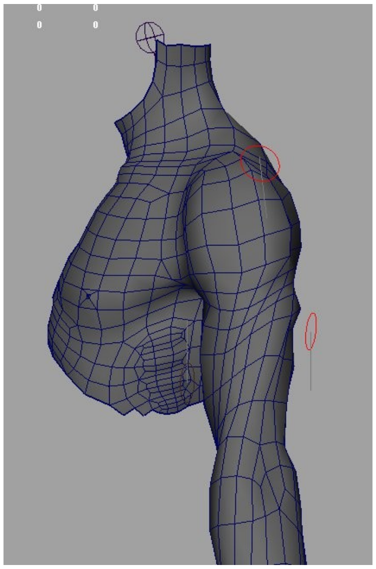
Maya 2011 file: 04_muscleDynamics/muscleflex_muscleShape_01.ma

As on the above example, you could be modeling your **muscle object profiles** (above) into specific shapes and then driving them based on the pose. This is ok for general shaping, but doesn`t offer much detail. You can also apply dynamics to these profile transforms.
Important to notice, **Maya Muscles** toolset offers similar profile flexing setup as above and can be used as well.



Maya 2011 file: 04_muscleDynamics/muscleflex_muscleSculpt_02.ma

Other could be using **sculpt objects** and model the mesh using them, pushing out the mesh (or pulling in). The advantage of using sculpt objects is adding sliding skin effect.
This could be quite expensive for performance when applied on the hires mesh. One way to speed up this setup is to apply sculpt on influence mesh representing only the area of arm and adding this as component influence to the hires mesh. Shown on the example above.
You can apply dynamics on your sculpt object in the same way as with the **profile** technique.



Maya 2011 file: 04_muscleDynamics/muscleflex_biceps_blendShape_02.ma

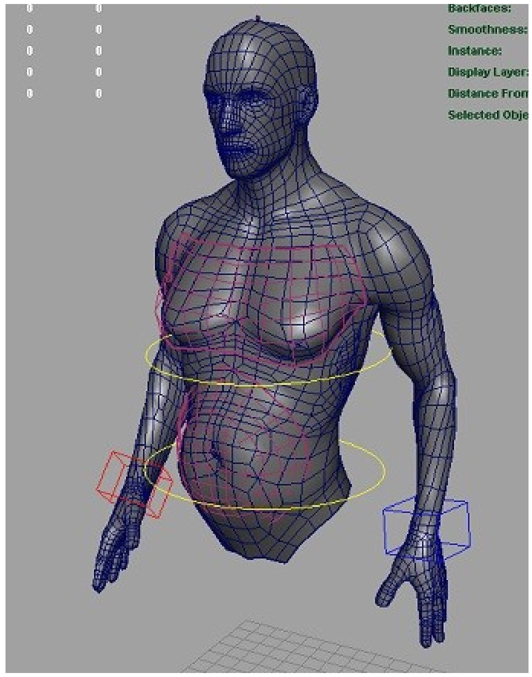
One that is very simple and fast but most importantly offers detailed control over every point is using **blendShapes**. Some fake sliding skin effect can be done as part of the blendShape with sliding baked into the shape. This can work with less extreme changes, if sliding moves in linear way.
With blendShape flexing, you will basically model your flexed muscles in bind pose as a number of shapes for each muscle area and then drive each shape depending on the movement.

This example scene above shows setup with blendShape flexing of deltoid fibres, biceps and triceps muscles. Simple one-dimension slide controls offer keyframe animation access to flexing. This can be of course driven automaticly as well (by angle, distance etc.).

Muscle and skin jiggle

Another important part of dynamics is the jiggle movement of skin and muscles. Again, there are several ways to do this. You can modify your flexing driver to provide values of higher frequency based on acceleration changes. Let`s say when the arm or leg suddenly stops. This can drive the same flexing blendShapes and provide a simple jiggle effect.

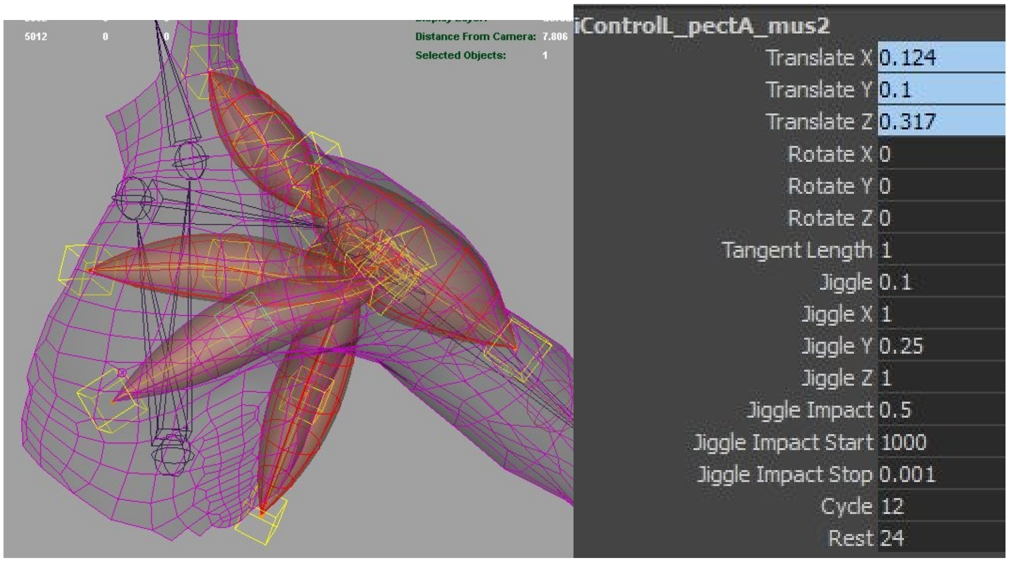
You can also setup a jiggle directly on a geometry as a jiggle deformer. For more advanced setup you can apply the jiggle on a separate lower resolution mesh and drive your hires mesh by this. In the example below Maya jiggle was used (red mesh driving body mesh).



04_muscleDynamics/muscleflex_belly_jiggle_03.ma

Very effective way to setup jiggle is to enable it on your Maya Muscles or setup your own rigid body dynamics. The dynamics should be applied on the middle control of the muscle or on any other middle muscle null if you don`t have controls.

Below is an example scene reused from previous Muscle Objects tutorial with some low dynamics on pectoral muscles. It is good to be rather subtle about these things and test a lot on final animation to get the best settings. Notice the main jiggle value is 0.1. Quite low.
Very useful is breaking the jiggle values to 3 local translate axis, as often we only want the muscle to jiggle in limited space.



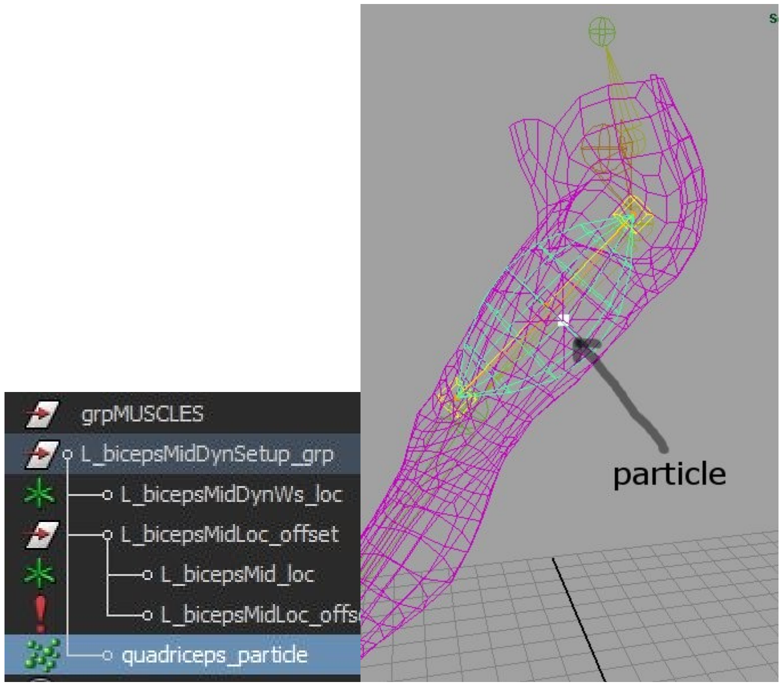
Maya 2011 file: 04_muscleDynamics/muscleObj_shoulder_setup.ma

Important feature of muscle jiggling is translate limit in space. This is what Maya Muscles are missing. The control might go too far sometimes in extreme animations and look unrealistic. So the settings need to be adjusted or animated per shot before simulation. That`s why translate limit is useful. So you know the muscle will not get too far and break your character`s shape. The example of this setup is below applied to a leg quadriceps. It looks fine, but the problem can occur in animation with high speed movement in world space.

Maya 2011 file: 04_muscleDynamics/muscleflex_leg_jiggle_limit_01.ma

The problem with Maya Muscle jiggle is a lack of any transform getting the jiggle values from the muscle spline. It would be easy to adjust those values for example by a *clamp* utility node. There is actually output from muscle spline with a jigglePoint value but the no input. So even if you clamp the values, you cannot change the final effect on the muscle spline, which deforms the muscle.

In the scene below I made a customized muscle setup using Maya Muscle with its jiggle turned off and driving middle muscle control using a simple Maya particle.



Maya 2011 file: 04_muscleDynamics/muscleflex_leg_jiggle_limit_02.ma

The particle was made using a command like on this example

```
particle -n "my_particle" -p 0 20 0;
```

Then I have setup a goal for this particle, which was a locator at the same position where I made the particle. The goal weight is set quite high, so muscle doesn`t stay too behind (goal can be also setup using Particle menu > Goal).

```
goal -g "locator1" -useTransformAsGoal 1 -w 0.8 "my_particle";
```

Then I connected particle shape`s worldCentroid (world center point of particle shape) to my locator in world space. To this I point constrained another locator which I made in the same space as muscle middle control. Then I used translate limits on this local space locator. And the actual muscle control was constrained to this locator.
This was very comfortable way of keeping the dynamic muscle control within Maya unit limits, proportional to the character`s body size.

Using Cloth

I should mention the use of cloth simulation in muscle and skin dynamics. You can use Maya nCloth and experiment with local influence on parts of your skin. Higher resolution cloth usually starts to look saggy and wrinkled and it`s good to use on thick and heavy skin animals like reptiles (dragons), elephants and generally groin and neck areas of quadruped animals, because they are hanging and sliding. For more soft effects lower resolution cloth can be subdivided and drive higher resolution geometry at selected areas.

Conclusion

Muscle dynamics are adding more realism to character deformation. It`s the illusion of weight and moving flesh. You can apply these even at earlier stages of your creature setup and this method is also scalable. Meaning, you can go for complex solutions as well as cheaper and still gain benefits.
Adding flexing and jiggle effects needs proper testing using as final animation as possible, because every animation style needs different settings. It`s also better to be subtle and careful with adding these effects. This is secondary layer and shouldn`t overshadow the main animation (unless this is required). It should stay within the range and proportions of the character and style of its animation, realistic or cartoony.