# Cartoon Motion and Deformation of Skeletons

**Ting Ting Wu**
**Advisor - Nancy Pollard**

## Introduction

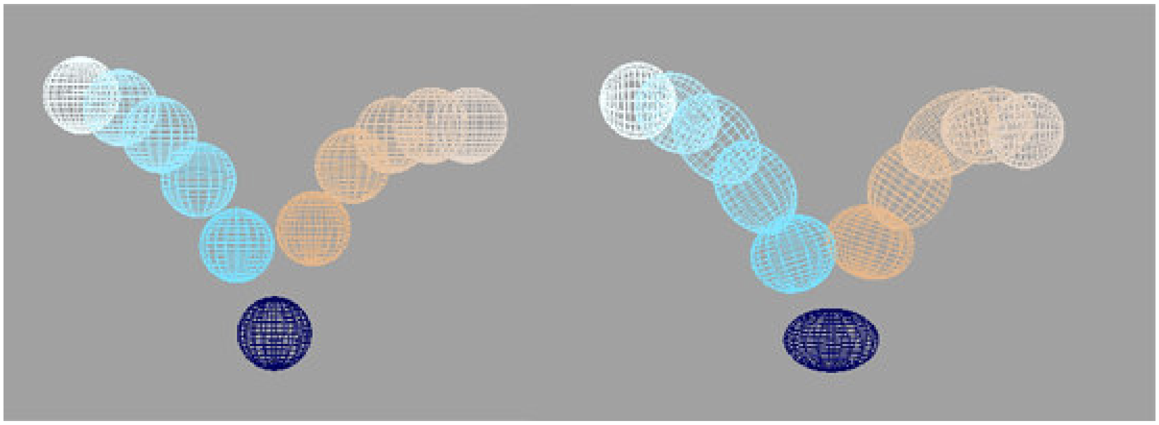


**Figure 1.**

Stretching and squashing in an important technique in conveying expressiveness in 2D animation. The bouncing ball is the quintessential example to demonstrate its effectiveness (**Figure 1**). And while this technique can be used in 3D animation, dealing with these deformations in 3D is very cumbersome. Animators usually work with a fixed skeleton for animating a character, no longer deforming the underlying skeletons. My research aimed to find an algorithm that can take an existing animated skeleton and apply the appropriate deformations over time. The goal was to find a way to add "cartoonyness" by using the stretch and squash method.

The matter of what the "appropriate" deformation is for a skeleton at any point is an ambiguous question. Seasoned animators consider many things like the mood and intention of a character in order to plan how to stretch and squash their characters. They also use stretch and squash to give their characters a sense of weight, and that is something that can be quantified. I started here with my research, by trying to find a physically based system to drive skeleton deformations. Given an input motion, an animator can create different deformation effects by entering a different set of weights.

## Related Work

There has been limited work in the area of imitating 2D animation styles. Among them is Christoph Bregler's 2002 paper "Turning to the Masters: Motion Capturing Cartoons" which looked at cartoon motion and retargeting it to a 3D character, but did not discuss actually generating "cartoony" motions.

Another source was Stephen Chenney, et al's 2002 paper titled "Simulating Cartoon Style Animation", which outlined a method to perform automatic deformations on a simple object with volume-preserving squash and stretch upon collisions. We are interested in deformations for a full skeleton, but it can be broken up into individual bones and treating them like separate deformations, where this paper's insights were relevant.
One last source was "Stylizing Motion with Drawings" by Michael Gleicher, et al in 2003, which viewed deformations in cartoons as a combination of skeletal changes and mesh changes. However, the skeletal deformation of their method was user driven.

## **Data**

In order to obtain a sample "cartoony" walk. I rotoscoped a clip of a Mickey Mouse walk cycle in Maya. The process of extracting 3D information from a 2D animation was not entirely smooth, and gave a satisfactory motion to work with.

We also motion captured an actor mimicking Mickey's walk. In addition, we had a number of "normal" human walks for comparison purposes.

We decided on a set of weights for Mickey based on an approximation of the volume of each of his body parts, human flesh density, and the assumption that if he were real, he would be three feet tall. The human weights, I pulled from approximations given in the 1997 paper by Hodgins and Pollard titled "Adapting Simulated Behaviors For New Characters". **Figure 2** shows the skeletons and locations of masses. **Table 1** shows the weight values.

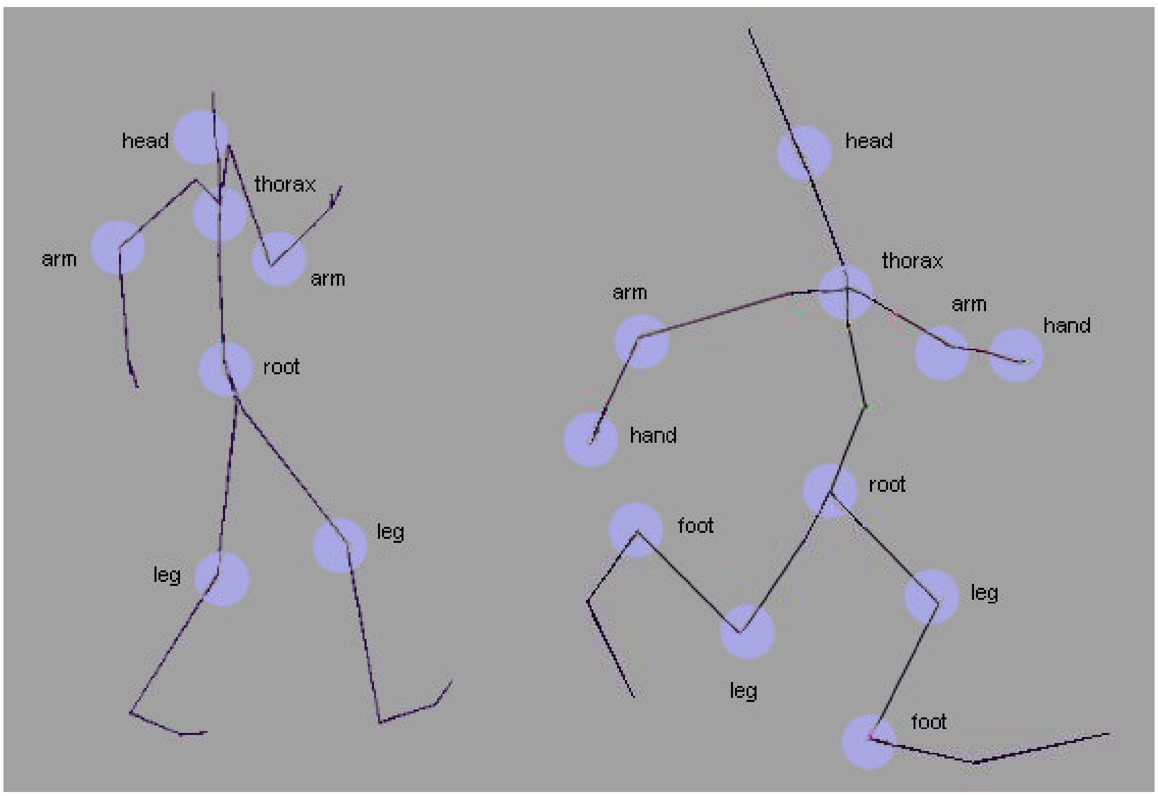



**Figure 2.** The location of body masses in the human skeleton (left), and Mickey's skeleton (right). (Note: Figures not to scale.)

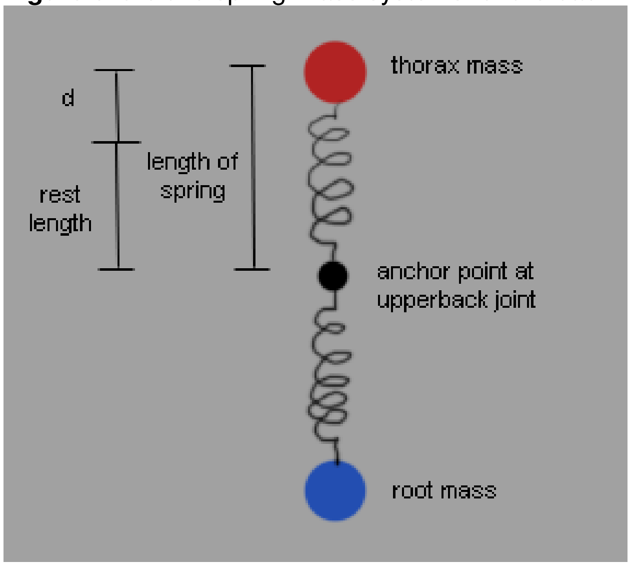
**Table 1.** mass values in kg

| Body Parts | Human Character | Mickey |
|---|---|---|
| Head | 21.304 | 4.22 |
| Thorax | 4.091 | 17.62 |
| Root | 7.069 | 11.08 |
| Arm | 4.022 | 2.51 |
| Leg | 4.241 | 8.32 |
| Hand | 1.403 | |
| Foot | 5.292 | |

## Spring-Mass System

A sensible model for the deformation of bones driven by weight would be a spring-mass model. In particular, we decided to look at the changes in the upperback and lowerback lengths and tried to fit a spring-mass system to the data. **Figure 3**. We treated the thorax as a mass on a spring anchored at the upperback, and the root as a mass on a spring anchored at the upperback.

**Figure 3.** the two spring-mass systems for the back.




Both of the spring-mass systems abide by this equation:

$$F = m*a = -k*d - b*v$$

where F is the force, m is the mass acting on the spring, a is the acceleration, d is the displacement from the resting spring location, v is the velocity, k is the spring stiffness coefficient, and b is the damping coefficient. We already have the mass and displacement information at each time interval for Mickey's rotoscoped walk and from there we can derive the velocity and acceleration, then we would like to use those values to approximate the k and b coefficients, which should encode how Mickey's bones deforms as a response to motion. To approximate for k and b, we used bivariate linear regression.

Once we obtained approximates for k and b, all we need is the mass, velocity and acceleration from the human motion data to figure out the displacement value of how the bone has deformed.

The one issue with regard to this last part is that the velocity and acceleration values are that of the moving mass on the spring. In the case of the human motion-capture data, the skeleton does not change, therefore velocity and acceleration is always zero. However, there is a way to generate some reasonable values for a and v. **Figure 4.** During the walking motion, the joints are all moving in the world coordinates, so we can simply take its global velocity and global acceleration and project it onto the direction of the spring. **Figure 5** shows a plot of some one of the set of results. Note that the acceleration data from the rotoscoped motion was very jagged even after much smoothing, as a result, the force approximation based on k and b does not fit too well. However, the animation of the altered walk is quite interesting and does enhance the original walk with more personality. The resulting animations are included.

**Figure 4.** Generating a velocity based on the velocity in world coordinates.

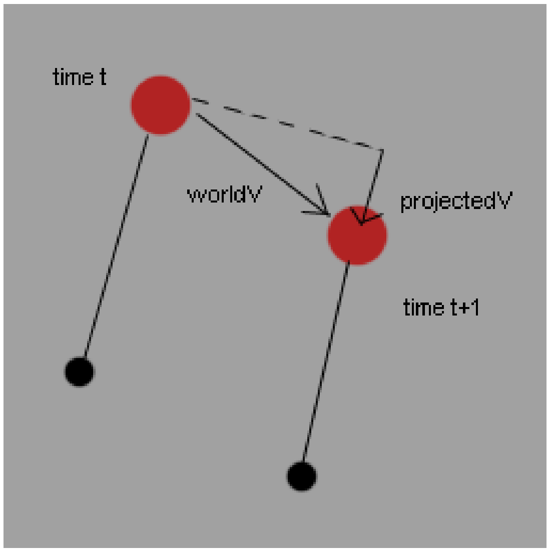
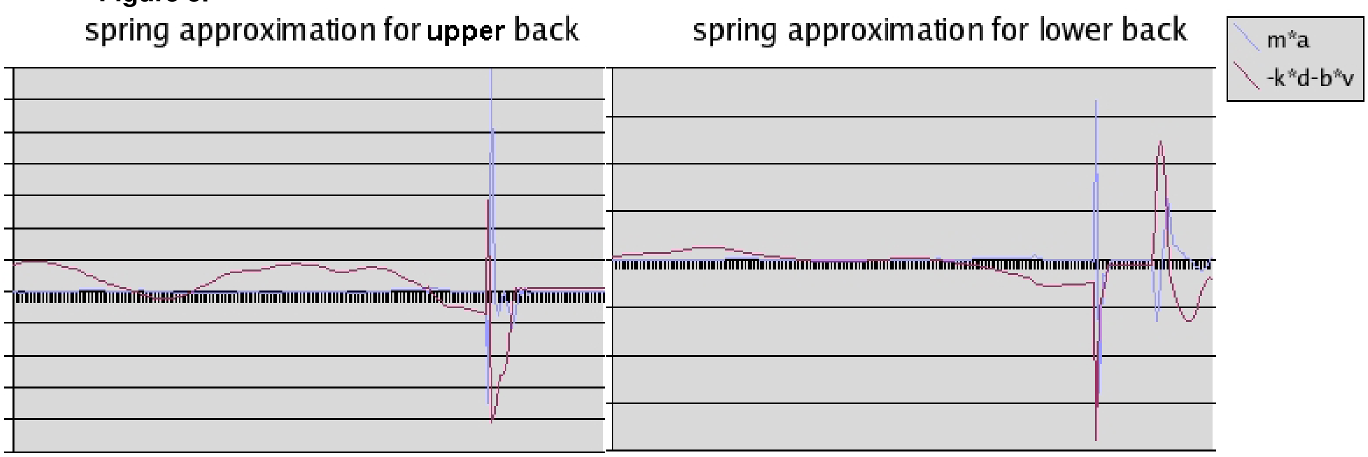



**Figure 5.**




## Conclusion and Future Work

Critique of the resulting animation included that it appeared too bouncy, even for Mickey's signature double-bounce walk. Another option would be to replace the spring-mass model with Chenney's stretch and squash model, in which one inputs parameters like the maximum stretch, the minimum squash, the deforming rates, and restitution, then it is a matter of interpolating the deformation in between.

Something that would definitely help is obtaining more professional hand animated motion that has stretch and squash. From there we can obtain cleaner acceleration data. Another direction to take this would be, instead of looking at a physical model, looking at motion curves of the joints in world coordinate frame. Traditional 2D animators do not bind themselves to physical equations, but rely heavily on using smooth curves to create interesting motion. (Williams, Richard. The Animator's Survival Kit**)** Hence, that might be a better way to tackle this problem.