Center-Of-Mass

Believable Physically-Accurate Character Movement

Ben Kenwright[+]
www.xbdev.net

Abstract

In this paper, we present and demonstrate the practical importance of a character's center-of-mass both in balancing and in producing believable character animations. A character's overall center-of-mass is a low-dimensional approximation that is computationally fast and straightforward for identifying un-natural looking and physically implausible character motions. We show that in the majority of cases that exhibit un-natural looking puppet-like motion that appear odd and implausible can primarily be put down to incorrect placement and movement of the character's overall center-of-mass. Furthermore, while the center-of-mass can be responsible for making a character's movement look plausible it is also important for balancing and is central for intelligent human-like foot placement information in biped type models. We show how to calculate mass and inertia information dynamically at run-time for individual limbs and the overall character model.

Keywords: Center-of-Mass, Character Animation, Balancing, Inverted Pendulum, Responsive, Physics-Based, Low-Dimensional Model

---

1. Introduction

A character's center-of-mass (CoM) (also known as his center-of-gravity (CoG)) is crucial for understanding how to create realistic life-like character movements. Physically plausible and natural character movement is interesting and important for both graphics and robotics. The crucial factors are the position of the center-of-mass, timing (i.e., where and when it should be at a specific time), and speed (i.e., how its velocity and direction change with time and posture).

While an experienced and skilled animator can identify and rectify a bad motion and provide a good approximation of where and why character's center-of-mass location should be, the majority of us cannot.

The center-of-mass of any character is a single point that can be used to support the character and it will remain balanced and stationary. We can approximate a human's center-of-gravity around his pelvis area. However, depending upon a character's pose (e.g., rolling, jumping, and running) it can shift and move. For example, when you lean forwards to pick up a heavy object you will re-configure your body posture so that the center-of-mass is moved outwards and behind you so that you can compensate for the heavy object you are trying to pick up. You accomplish this by pushing out your pelvis behind you and crouching down so that your center-of-mass is closer to the ground.

[Model - number of muscles, image]

1.1. Motivation

The motivation of this paper is aimed at moving away from static, impassive, life-less key-framed animation libraries towards more dynamics and interactive methods that use physics-based techniques to produce more natural, non-repetitive and human-like character movements.

1.2. Contribution

The contribution of this paper is to demonstrate and explain the fundamental principles of a character's mass properties and how it can be used to produce and correct movements so they are more life-like and natural. Analyzing and understanding a simplified low-dimensional character model can aid in producing a more life-like and physically accurate character pose that possesses balancing characteristics and dynamics properties.

2. Related Work

There has been a tremendous amount of research into the subject of physics-based character movement. The mass properties are central for a physics-based model and are responsible for how it should move and respond to disturbances.

Oba [**OBA10**] presents an interesting introduction to the workings and misconceptions to the workings of a biped character's center-of-mass and how it is crucial for producing natural looking movements that mimic real-world humans.

The recent work by Kenwright [**KDM11**][**KEN12a**][**KEN12b**] has explored and demonstrated how a simplified low dimensional character model can be combined with an inverted pendulum technique to produce responsive, dynamic, life-like character animations without key-frame data.

3. Overview

The equations for calculating an articulated character's overall center-of-mass is straightforward. We use a physics-based model to represent an articulated character as an interconnected set of rigid body limbs. For example, we can calculate and solve the articulated constrain problems using rudimentary equations based around Newton's laws [**KM12**].

4. The Laws of Motion

The laws of motion are unbreakable and well defined. For game character motion, we use classical mechanics, which builds upon the laws put forward by Newton in 1687. The three fundamental laws by Newton are relatively straightforward and easy to remember and must be obeyed for an object to be move realistically and physically-correctly. For example, an object cannot arbitrarily modify its mass or change direction arbitrarily in mid-air.

5. Gravity ($9.8ms^{-2}$)

On earth, the gravitational constant of acceleration g, for an arbitrary object, is 9.8 metres per second. Hence, if you drop an object from a distance of 9.8 metres it will take exactly one second to hit the ground (neglecting air friction). Furthermore, while the object

---

+e-mail: bkenwright@xbdev.net

is in free fall – it CANNOT speed up and slow down randomly. This is also applied to angular rotation. Since an object (that is a character), rotating in space must continue to rotate and CANNOT arbitrarily change direction without touching or having an external force applied (i.e., law of conservation of momentum).

6. Falling Objects

Shocking an object in your simulation software does not fall by default due to gravity. To emulate an object falling due to gravity we have to apply a downwards force to the objects center-of-mass to emulate the gravitational acceleration. This falling effect due to gravity is known as 'free falling'. Crucially, the gravitational acceleration constant g is independent of the objects weight since it is an 'acceleration' not a force. We can use Newton's law to calculate the necessary force to mimic an object free falling due to gravity by multiplying the gravitational acceleration constant by its mass as shown in Equation **6-1**.

$$\begin{aligned} F &= mg \\ v_{n+1} &= v_n + \frac{F}{m} dt \\ x_{n+1} &= x_n + v_{n+1} dt \end{aligned} \qquad \textbf{6-1}$$

where v_{n+1} and v_n is the velocity of the current and next frame, x_{n+1} and x_n is the position of the current and next frame, *m* is the objects mass, *dt* is the time step, and *F* is the Force.

We can unify the simulation integration steps into a single equation to show how a free falling object's position changes with time. Since a free falling object (starting from rest) move in the vertical direction only (i.e., the y-axis) we can calculate its position using Equation **6-2**.

$$y = -\frac{1}{2} g t^2 \qquad \textbf{6-2}$$

where y is the position in metres along the y-axis, g is the gravitational acceleration constant (i.e., 9.8), and t is the time in seconds (note the negative sign due to the object moving downwards).

7. Calculating a Shapes Center-of-Mass

We use the 'divide and conquer' approach to calculating the mass properties of a character's rigid bodies. We can calculate the mass properties for convex shapes (i.e., weight and center-of-mass). We can also calculate the mass properties for concave objects by subdividing the shape into simpler convex parts. We make the assumptions that the object has a uniform density.

While there are different approaches to calculating a shape's mass properties, we use the straightforward and uncomplicated method of subdividing the shape into tetrahedrons. We can calculate the mass properties of a simple tetrahedron easily. Then we can combine the mass properties of the individual tetrahedrons to give the total mass properties of the overall shape.

$$M_n = \frac{1}{6}(V_2 - V_4) \times (V_3 - V_4) \bullet (V_1 - V_4) \qquad \textbf{7-1}$$

where M_n is a scalar quantity representing the n'th tetrahedrons mass volume, and $V_1, V_2, V_3 \, and \, V_4$ .are the four vertices of the tetrahedron.

Hence, the total volume of the convex shape is the sum of the tetrahedron volumes as given by Equation **7-2**.

$$M = \sum_{i=1}^{n} M_n \qquad \textbf{7-2}$$

Furthermore, the center-of-mass for each individual tetrahedron tetrahedrons is calculated using Equation **7-3**.

$$COM_n = \frac{(V_1 - V_4) + (V_2 - V_4) + (V_3 - V_4)}{4} \qquad \textbf{7-3}$$

where COM_n is the center-of-mass for the n'th tetrahedron, and $V_1, V_2, V_3 \, and \, V_4$ .are the four vertices of the tetrahedron.

Finally, the mass and volume information for each tetrahedron allows us to go ahead and calculate the overall center-of-mass for the shape using Equation **7-4**.

$$COM_{total} = \frac{1}{M} \sum_{i=1}^{n} COM_n M_n \qquad \textbf{7-4}$$

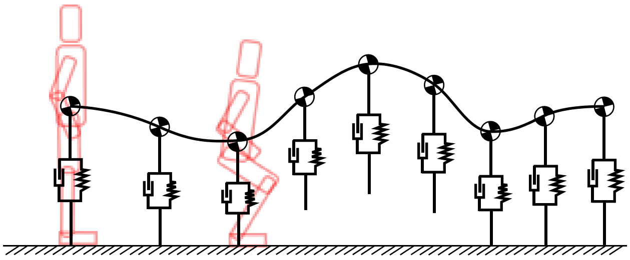

Figure 1. Leg-knee is analogous to a spring-damper mechanism so that we can store and output energy to achieve lift (e.g., in jumping) while damping is necessary to produce a stable controllable solution (i.e., avoid continuously oscillating or bouncing).

8. Conclusion

Analysing how an animated character's mass moves over time can help identify implausible and un-natural looking animations. While accurate mass information may be unavailable for character models, we have shown that a straightforward method of calculate approximate mass values using the character skinning mesh.

9. References

[KDM11] B. Kenwright, R. Davison, G. Morgan, "Dynamic Balancing and Walking for Real-Time 3D Characters," Motion in Games, 2011.

[KEN12a] B. Kenwright, "Generating Responsive Life-Like Biped Characters," The Third Workshop on Procedural Content Generation in Games (PCG 2012), May 29th, p. 1-8, 2012

[KEN12b] B. Kenwright, "Responsive Biped Character Stepping: When Push Comes To Shove," Cyberworlds, September 2012.

[PG96] K. Perlin and A. Goldberg, "Improv: A system for scripting interactive actors in virtual worlds," in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996, pp. 205–216.

[PERL95] K. Perlin, "Real time responsive animation with personality," Visualization and Computer Graphics, IEEE Transactions on, vol. 1, no. 1, pp. 5–15, Mar. 1995.

[KM12] B. Kenwright, G. Morgan (2012), "Practical Introduction to Rigid Body Linear Complementary Problem (LCP) Constraint Solvers", In Algorithmic and Architectural Gaming Design. pp. 159-205, 2012

[ODE12] Open Dynamics Engine, http://www.ode.org/ (accessed September 05, 2012).

[NGD12] Newton Game Dynamics, http://www.newtondynamics.com/ (accessed September 05, 2012).

[HIK12] HumanIK, Autodesk Gameware, http://gameware.autodesk.com/humanIK, 2012

[EUPH12] Euphoria, www.naturalmotion.com, 2012

[HAVO12] Havok, www.havok.com, 2012

[OBA10] E. Oba, "Centre of Mass: Tools and Techniques for Animating Natural Human Movement," 20 May 2012, Online: http://gamasutra.com/view/feature/132982/center_of_mass_tools_and_.php

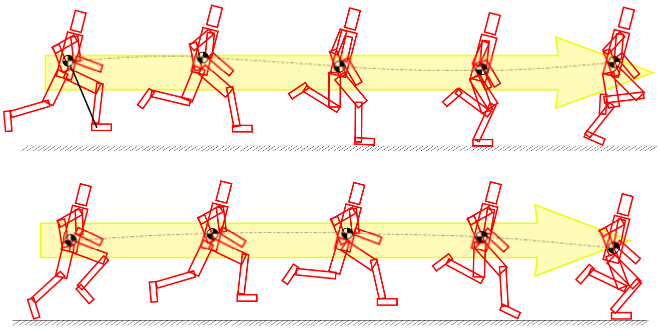

Figure 2. Center-of-Mass trajectory for a simplified biped character model running (i.e., dynamic motion with the feet intermittently leaving the ground between step transitions).