Inverse Kinematics

November 24, 2017

Abstract

Your abstract.

1 Introduction

Skeleton based animation is a frequently used technique for both 2d and 3d animation. An important mathematical principle behind this technique is kinematics which allows the specification of the movement of objects given forces on them. For animation, two different forms of kinematics are used. Forward kinematics, which take a given rotation or motion of a joint and compute its impact on any dependent joints, and inverse kinematics which takes a desired motion of an end effector and computes the necessary rotation and motion required to generation that motion.

2 Inverse Kinematics

Inverse Kinematics for two joints can be trivially solved with trigonometry, however it becomes more complicated for any number of joints greater than two. The method used to compute inverse kinematic transformations for more than two joints is known as the Jacobian Transpose. The problem addressed by this algorithm is the necessity of converting a linear transformation to a rotational transformation. This algorithm operates on each joint independently in turn, in essence it takes each joint and computes the change in the end effector produced by a change in that joint. From these changes it is possible to compute the change in a joint required to effect a given change in the end effector.

To enable this computation, a matrix called the Jacobian is computed. The Jacobian is a matrix of partial derivatives of a non-linear transformation function. In the case of inverse kinematics the non-linear function is the transformation of the end effector by the angles of the preceding joints. This function can be understood as a series of translations by the limb lengths followed by rotations by the joint angles. Each row of the Jacobian is a partial derivative representing the consequences of a change in one joint on the position of the end effector.

3 Examples

To demonstrate inverse kinematics I have created a program that implements it in two dimensions for a simple three joint limb. A desired end point is able to be specified with the mouse. Each joint on the limb has a full range of rotation and the limbs are rigid fixed-length arcs.

Extending the inverse kinematics algorithm to work in greater dimensionality or with a greater number of joints simply requires the Jacobian to be extended, this means that the Jacobian transpose method of computing inverse kinematics is highly extensible and flexible.

4 Conclusions

Inverse kinematics is a powerful and flexible tool for enabling skeleton based animation. Its low computational overhead and high level of extensibility allows it to be used in rendering of complex geometry for animated movies and for real time animations for games. Extensions of Inverse Kinematics that I did not demonstrate in my program are accounting for flexibility of the limbs and limiting the range of motion on a limb. Both of these can also be accomplished using a

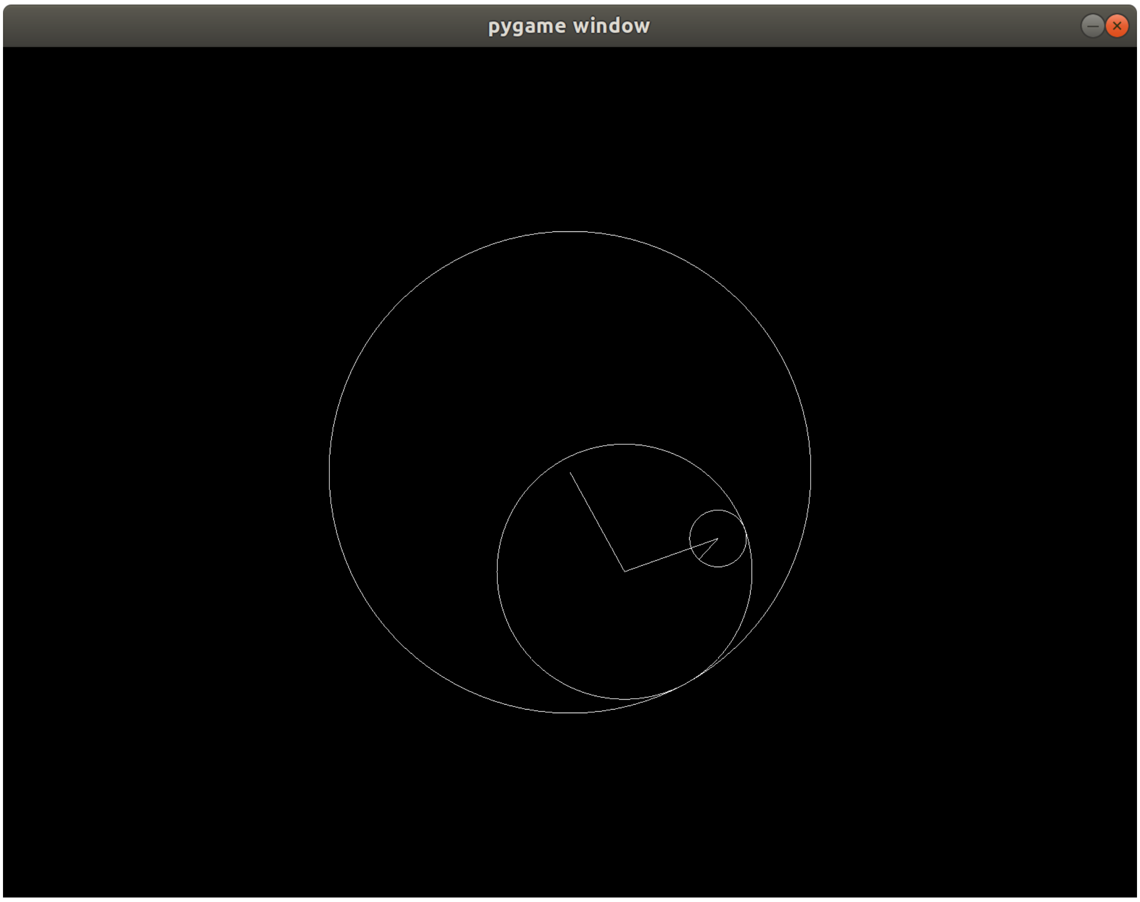

Figure 1: The program running

Jacobian matrix however they require changes to the specification of the equation of the position of the end effector.

References