

CS 283 HW4 Write-up

Inverse Kinematics

I implemented an inverse kinematics solver for a multi-joint arm using a Cyclic Coordinate Descent algorithm. Given an arbitrary number of bones > 1 , it solves IK for a target point at the mouse cursor. It gracefully handles out of reach cases by attempting to place the end effector as close as possible to the target, and then ending the recursion of the algorithm.

Cyclic Coordinate Descent

For each joint, starting at the furthest from the root, vectors are drawn from the current joint to the end effector, and from the current joint to the target location. The segment (bone) of the current joint is rotated by the angle that will place the end effector along the vector pointing from the current joint to the target location.

After each iteration of this algorithm, it checks whether the end effector is within acceptable distance from the target. If it is, then calculations end until a new target is supplied. Otherwise, it calls the CCD algorithm again.

If the amount the end effector has rotated is below a given threshold, to the point that the amount it has moved is trivial, the calculations also end, as the end effector has gotten as close as possible to the given target.

Videos

Here are some video examples of the program run with different numbers of bone segments. Apologies for some of the stuttering, I believe it's the fault of my video capture software, as it only occurred while recording these videos.

14 Bones:

<http://www.youtube.com/watch?v=7tfWBkTyCrc>

7 Bones:

<http://www.youtube.com/watch?v=9bM1ysYheaA>

3 Bones:

<http://www.youtube.com/watch?v=wQ36mCWjU0c>

Misc.

The IK solver runs along the x-y plane, as I had difficulty working the calculations for targets with a z component, and interfacing with the program using a mouse becomes much more complex with the z axis involved.

I used the framework from CS184 HW2, so some additional files are necessary to run the program (FreeImage, glew, glut, etc.). It is run with hw4.txt as a command argument. The meat of the code is in main.cpp and display.cpp, with some structs added in variables.h.