Project 3: Optimization-Based Animation

Project Due: Fri., Nov 11

Overview. The most important modern trend in animation is the rise of optimization-based techniques. These approaches try to control some aspect of the animation based on controlling the animation to optimize a "loss function" or "energy function". These functions are typically based on the underlying physics (optimization-based physical simulation), based on matching some recorded data of what the system should look like (data-driven animation), or a desired goal the animator wants the agent to reach (reinforcement learning). In this assignment you will explore both some basic concepts in optimization and use an optimization-based approach to implement inverse kinematic.

You may work with a partner for this – but the two partners must be different than the person you work on with either HW3 or Project 2, one turn-in per pair.

Part 1: Inverse Kinematics Scenario [90 points]
You will need to create a simulation where an animated character uses inverse kinematics to move, grab or reach in an environment. The basic requirements involve using a single IK solver to animate a simple 2D skeleton. For additional credit, you should implement a 3D scenario, plan for more complex skeletons, add obstacle avoidance, user interactions, and compare various IK techniques.
Required components are indicated with a star (*).

Single-arm IK (at least 2D)* (up to 20 points).
Produce an animation where a skeleton of at least 4 joints and at least one end effector reaches to touch an object. The skeleton must have a root that does not move. You must show at least one successful animation, and one animation of what happens when the object is out of reach.

Multi-arm IK (at least 2D) (up to 20 points).
Produce an animation where a skeleton, of at least 8 joints and at least two end effectors, reaches to touch an object. The skeleton must have a central root between the two arms that does not move. Both end effectors must be able to move and touch an object. You must show at least one successful animation for each arm, and one animation of what happens when the object is out of reach both arms. *[This will also count as single-arm IK meaning multi-arm IK totals up to 40 points.]*

Joint limits* (up to 20 points).
Add angle and/or rotational speed limits to some or all of the skeleton joints. Include in your report a video showing the difference in motion with and without these limits. For full point, be sure to choose limits that increase how natural the resulting animations look.

Obstacles (up to 20 points).
Add multiple obstacles to your environment. Detect and prevent any collisions between the skeleton and the obstacles.

User Interaction (up to 10 points).
These points come from allowing the user to interact directly with the simulation itself (not from controlling the camera). To get full points, the user should have a clear, smooth, and natural way to interact with the animation. Discrete interactions such as toggling some behavior on/off will only receive a couple of points. For full credit, support continuous interaction such as allowing the user to use their mouse move an obstacle or to control where the IK system is trying to reach.

Moving IK (at least 2D) (up to 10 points).
Allow the skeleton to move its root in a natural looking fashion that interacts smoothly with the IK-based animation.

Re-rooting IK (at least 2D) (up to 20 points).
Allow the skeleton to walk, climb, or inch its way forward by changing what node is rooted to the ground. For example, walk could be done by rooting the left root, swinging the right foot via IK to reach a point about a meter forward, then treating the right foot as the root of the skeleton and using IK to plan the left foot. Note, do not violate any of the length constraints of the skeleton. *[This will also count as moving IK, meaning re-rooting IK totals up to 30 points.]*

Motion Planning (up to 20 points).
Use a PRM, RRT, or other motion planning technique to allow your skeleton to plan a path through the environment. The path planning should be used either for the root to ensure the skeleton avoid obstacles or it should be used plan an optimal series of target points for feet/hands (or plan both the root and feet/hands). For these points, the motion planner should plan only the root and grip positions, and the IK solver must control all other joints.

3D Simulation & Rendering (up to 20 points).
Simulate and render your IK system in a 3D environment. For full credit, the 3D nature of the motion needs to be clear in the resulting animation though the use of dynamic cameras, 3D lighting, etc.

Skinned Models (up to 10 points).
Integrate your IK system with an existing animation system or game engine (such as Unity3D) in a way that allows the IK to fully control a skinned character. Note, you must use your own IK solver and in no way use solver, planning, or geometry code that is built-in to the system.

IK + Character Animation (up to 10 points).
Integrate your IK system with an existing animation system or game engine (such as Unity3D) in a way which shares control with the built-in animation system. For example, the game engine may control the characters walking cycle, but your IK

solver controls the arms to reach out and grab certain objects. Document carefully what part you animate, and what part is built-in.

Part 2 - Challenge: Comparative Analysis
An additional challenge simulation is required if you are a graduate student, and optional for undergraduates. If you complete the challenge, you have a 72-hour extension on the project.

Alternative IK Solver (up to 20 points (grad), 20 points (undergrad)).
Implement a second IK solving technique beyond the one used in your above simulations. Include in your webpage some analysis comparing the two methods. Which was easier to implement? Which performed better? Which is more flexible? Do both methods support constraints equally well? Etc.

Some reasonable solvers to consider are: CCD, FABRIK, Gradient decent, Basic random search, or RRT (the configuration space is the joint angles). Feel free to research and implement other methods.

Art Contest
If you generate a pretty image (even by accident), save it to submit to the class art contest. A pool of honorable mentions will be given 2 points, and the grand winner gets 5 points. All winners will be chosen *completely subjectively*.

Project Report & Video* (10 points).
Your submission must be in the form of webpage with:

- Images of your physical simulations
- A brief description of the features of your implementation and timestamp of where they occur in your video(s).
- Code you wrote
- List of the tools/library you used
- Brief write-up explaining difficulties you encountered
- One or more videos showcasing features of your simulation
- Submission for the art contest (optional)

These 10 points for the submission itself will be based on the clarity of expression of the report, and to the degree which it quickly communicates what you tried, what worked well, and what didn't.

Additionally, each feature you expect to get credit for must be documented in your submission videos in a way which clearly shows the resulting behavior. If you do not show a feature in your submission video(s) you will not receive credit for it.

Note on Game engines: You may use Unity, Unreal, or another game engine so long as you only use it for rendering. You may not use it for any aspects relating to planning or IK! You can use the engine to skin your model (for points!). You can also use the engine's character animation system to control some of the body, so long as your IK system is fully in control of some aspect of the motion (this is also worth points).

Grading Criteria
Simulations must animate well and look convincing to get full credit. Partially implemented features will receive partial credit. Points past those needed for full credit will count as extra credit, though at a discounted rate (see Scoring below). If you do other things that you think are cool and worth credit let me know beforehand and be sure to document them in the report.

Use of other code and tools
Anything you are getting credit for must be code you wrote for this course. You must write the code for the simulation yourself! External libraries may be used for aspects that are not related to simulation (e.g., rendering, camera motion, video capture) just be sure to document that you used these.

Partners & Groups
You are strongly encouraged to work in pairs for the project. Each pair should turn in only one assignment. Both people will be given the same grade. <u>You cannot repeat the same partner from a Project 2, nor can you use the same partner in HW3.</u>

Project Scoring
Undergrads need 100 points for full credit, though you may choose to submit up to 120 points of work, subject to the following limits:
110 for part 1 (rope/cloth)
20 for the challenge
10 for the report
… if you submit more than the limit, we will grade a random subset.

Graduate students need 120 points for full credit, though you may choose to submit up to 140 points of work, subject to the following limits:
110 for part 1 (rope/cloth)
20 for the challenge
10 for the report
… if you submit more than the limit, we will grade a random subset.

Partial credit will be given. Scores computed as follows (points above 100 possible):
-*Undergraduate*: Grade is $\sqrt{}$(totalPoints * 100) [e.g., 100 points will be full credit]
-*Grad students*: Grade is $\sqrt{}$(totalPoints * 84) [e.g., 120 points will be full credit]

*Extra credit will only be given to assignments with at least an A- on the required features.