Quick!
We Need a Dancing Duck!

Andrew Gardner - University of Wisconsin, Madison

So You Want a Dancing Duck.

• A quick and effective way to animate many characters with a many motions
• An overview of the process includes:
  – Gather model and motion
  – Match model and motion poses
  – Bind the model to the motion
  – Tweak the results using constraint-based motion editing techniques

Start With a Model.

• Off the web
• Created from scratch
• Must be able to bind to a hierarchical motion skeleton

(Art model courtesy Hou Soon Ming)
Grab a Motion.

- Off the web
- Freshly captured
- From company moCap archives
- Must be hierarchical and humanoid in nature

Size up the Situation.

- Our chosen character size and motion size are clearly different

Resize the Mesh?

- Simple to do
- If limb proportions differ, this leads to distortion
Resize the Motion.

- Simple measurements lead to bone-length changes
- Leads to a few problems

Wow! Done Already?

- The resized motion looks good, right?
- But wait, no poses in the motion match the character’s pose…

Assigning Motion to the Model.

- Could take arbitrary motion pose and rotate it into model’s pose
- Could build a skeleton for the model from scratch with matching bone lengths
Rotating Skeleton Into Place?

- Tedious process
- Difficult to get certain rotations correct (wrist, shoulders, etc)

Building Skeleton From Scratch?

- Appears to be very easy
- The joint coordinate systems may not match

Coordinate System Overview.

- Due to capture process and/or file format specifications, motions may have joint coordinate systems oriented in different directions
- This is best shown with examples
Shoulder Example 1.

• A -90 degree rotation around the Z axis in this case yields the following:

Shoulder Example 2.

• In this case, a 90 degree rotation around the X axis yields the same result as before:

So…

• Though the previous skeletons looked identical, their rotation data was very different
• When building a skeleton from scratch, coordinate systems of the new skeleton must match the motion’s
• Very difficult to accomplish
Our Approach.

- Instead of a skeleton, create a point cloud of joints in the mesh
- Trivial for the user
- Works with nearly any 3D modeling package out there

Our Approach - Prepare Motion.

- Need to pose the motion in a way which allows intuitive adjustments to match model's bind pose
- Zombie (or Frankenstein) pose

Rotate the Motion's Skeleton.

- Rotate each bone in the motion to match model point cloud information
- Relies heavily on knowledge of humanoid kinematics
- Maintains coordinate systems
Gotcha’s With Our Approach.

• Heuristic relies on humanoid motion simplification
• Point cloud insufficient to convey all joint orientations
• If motion is not humanoid, humanoid kinematics do not apply

Humanoid Motion Simplification.

• People’s joints move in very complex ways
• Example: shoulder/elbow rotation

Motion Simplification Contd.

• When rotating shoulder into position, it is very difficult to take all rotational degrees of freedom into account
• We only allow for rotation around two axes
Motion Simplification Contd.

• Simplification problems manifest themselves in subtle ways

Limb Forward Direction

• Hands, elbows, feet, and knees especially problematic
• Feet have a free guide – the toes point forward

Limb Forward Direction Contd.

• Hands have no forward vector
• Easy to add forward point in cloud, but somewhat unintuitive
Non-Humanoid Motion.

• Unless you know at least some of the complexities behind the creature’s degrees of freedom, all bets are off

Finally We Can Bind!

• Despite all the gotcha’s, our technique still works quite well
• With our matching skeletons, any bone-based hierarchical skinning technique can be applied

Now Are We Done?

• Most certainly not!
• The model is not interacting with its environment as it should
• Its feet are nowhere near the ground
A Simple Transformation Fix?

- Model’s root is still following old root translations
- A simple translation to the floor will yield sliding feet and too much bounce

How Do We Fix This Mess?

- Constraint-based motion editing to the rescue!

Constraints – What Are They?

- A place in the motion where some relation must be true
- Please Refer to Dr. Michael Gleicher’s SIGGRAPH 1998 entitled Retargeting Motion to New Characters as well as the numerous others located in our bibliography
Example...

- Grabbing an apple on a table
- We place a constraint between the hand of the character and the apple

Example Contd.

- Shrinking the character, reveals that it cannot grab the apple with the same joint rotations

Example Contd.

- With constraints satisfied (solved for), shorter character adaptively leans and stretches to grab apple
Constraints and Our Example.

• In our case, adding constraints where the motion’s feet hit the floor would probably be sufficient
• As seen before, our new mesh is clearly not satisfying its foot-hits-floor constraints

Solving for the Constraints.

• A quick solve for the constraints yields our model walking nicely along the floor

Tweaking Our Solution.

• Constraint-based motion editing is an interactive and iterative process
• Though constraints are met, motion may not be as we would like
• Further tweaking can be done
Moving Constraints.

• Moving/scaling constraints to different positions may be more desirable

Summary.

• Our approach from an artist’s view:
  – Find any model and motion
  – Create a point cloud for the model (adding an extra point for the hands)
  – Run our pose-equating process
  – Create reasonable constraints
  – Run our constraint-solver
  – Tweak constraints until satisfied

Advantages of Our Approach.

• Coding the pose-matching portion is straightforward
• Robustness – it works with nearly every humanoid motion
• Reuse – it can breathe new life into old motions
• Speed – once system is in place, many motions can be retargeted quickly
Disadvantages of Our Approach.

• Coding an effective constraint-solver is very difficult
• Knowledge of motion’s kinematics must be mastered before pose-matching can be universally effective

Special Thanks To…

• Alias|Wavefront
• ViewPoint
• Discreet
• Mainframe Entertainment
• Hou Soon Ming (http://www.its-ming.com)

…and the UW Madison Graphics Group.

Oh, and About That Duck…

Didn’t we want a dancing duck out of the deal?