Animation from Example?

- Animation is useful!
  - Expressive, powerful, communicative, ...
  - Video, film, virtual environments, games, ...

- Motion is hard!
  - Subtle, difficult to characterize, ...
  - Lots of data, ...

- Good motion is precious!

If we can’t get new motions, we need to make use of existing ones!
Talk Roadmap

Animation from Observation:
How do we get good motions?

Motion Retargetting:
How do we use (and re-use) motions?

Animation by Adaptation:
Can we be adept enough at changing motion that re-use really is an option?

Animation by Example:
Can we synthesize new motions based on what we’ve seen already?
Research Roadmap

Animation from Observation:
Motion Capture 1998-2000

Motion Retargetting:
Motion Retargetting 1998
Constraint Editing 1997-2000

Animation by Adaptation:
Path Editing 2001, 2003
Footskate Cleanup 2002

Animation by Example:
Motion Graphs, 2002, 2003, …
Cut to the chase...

An Example

- How do you make a character sneak around?
- Start with some captured motion of a person sneaking around
- Synthesize a new motion of a character “sneaking” somewhere else
Show sneak video

I have no idea why video doesn’t work well in Powerpoint
Play SNEAK.AVI now
What did you just see?

- Small amount of example motion
- Examples of what I want
  - Actions
  - Quality
- Character did something different
  - New path
- Character did it the same way
  - Preserves “style” and “quality”
How to make a Character “Sneak”? 

- What is sneaking?
  - Hard to define mathematically
  - Abstract qualities matter
    - Style, mood, realism, ...
  - Details matter
    - Feet not sliding on the floor
    - Subtle gestures
Computer Animation 101: How do we get motion?

- Create it manually (keyframing)
  - Common method used for film
  - VERY talent and labor intensive
- Synthesize it by procedural methods
  - Physical simulation, or ad-hoc methods
  - Can’t get exactly what you want
- Capture it from a performer
  - Motion Capture
  - Animation from Observation
Motion Capture

Observations

Computationally Tractable Representations

Note:
Motion capture means capturing the motion, not the process of Animation by Observation!
Mocap Pipeline
(Animation from Observation)

- It’s more than just pointing cameras at somebody!
- Getting the observations is just one part of the process
- All examples I will show are from optical motion capture
Motion Capture Technology: Optical Tracking

- Use markers and special cameras
- Tracking + Math
Processing Pipeline

Raw Data

- Identify
- Clean
- Cleanup

Clean Marker Data

Solve to Skeleton

Clean Skeletal Data

Define Character

- Define Character Controls
- Define Skeleton

Define Animation Data

Retargeting & Control Mapping

Clean Skeletal Data

Application

OBSERVE

RETARGET

ADAPT

SYNTHESIZE
Why Edit Motion?

- What you get is not what you want!

- You get observations of the performance
  - A specific performer
  - A real human
  - Doing whatever they did
  - With the noise and “realism” of real sensors

- You want animation
  - A character
  - Doing something
  - And maybe doing something else...
The General Challenge

- Get a specific motion
  - From capture, keyframe, ...
  - Specific character, action, mood, ...

- Want something else

- But need to preserve original
  - But we don’t know what to preserve
  - Can’t characterize motion well enough*

*This is a working assumption of my research. I’d love to be proven wrong.
Three Problems

- Where does X live in the data?
  - Where $X \in \{\text{style, personality, emotion, ...}\}$
  - The things to keep or add
- Small artifacts can destroy realism
  - Eye is sensitive to certain details
  - Amazing what you can’t get away with
    - See Kovar, Schreiner and Gleicher, SCA ’02
    - Discussed later in this talk
- How to *specify* what you want
How do we handle these problems?

- Don’t know which details are important!
- Must preserve ALL details
  - Since you don’t know what is important
- Need to understand artifacts better

Motion Capture Animators are conservative:
Want excess precision just in case!
Another Approach
Constraint-Based Motion Editing

- Identify specific details in motions that must be preserved
  - *Constraints* such as footplants
- Make conservative changes to motions
  - Things that generally don’t cause problems
  - Add low frequencies
  - Blends with similar motions
- Re-establish constraints (solve)
  - Avoid creating new artifacts
A concrete example...
Retargetting Motion to New Characters

- Goal: one motion, a cast of characters
- Focus on similar structure
Retargetting Recipe

1. Define Constraints

2. Apply to new character
Retargetting Recipe

3. Approximate Answer

4. Solve constraints (band-limited adaptation)
The trick...

Temporal artifacts are important!

- Can’t add snaps
- Can’t add pops or wobbles or ...

- Need to look at durations of motion
Constraint Solutions for Editing

- Spacetime (single, large, non-linear optimization)
  - Gleicher ’97, Gleicher ’98, Popovic and Witkin ’99
- Hierarchical Splines
  - Lee and Shin ’99
- IK + Filter
  - Gleicher ’00
- Importance-Based
  - Shin Lee Gleicher Shin ’01
- IK + Blending
  - Kovar Schreiner Gleicher ’02
Retargetting Results

- Play retarget.avi
Why is this good?

- Found motion + found character
- Overall quality of the motion preserved
- Makes interesting animation
  - New characters move like performer
- Complete pipeline
  - Character mapping
What’s wrong?

- Character not considered
  - How would the character do this?
  - Simple mapping methods

- Non-linear optimization is a pain
- Results could be better
  - Small details not right
  - Some “big” details ignored
- Controls are too low level
- Clip in, Clip out
Towards a general paradigm...

Animation by Adaptation

- Need a fast and easy way to deal with the most important constraints
  - Footskate Cleanup
- Need ways to deal with motion at a high-level
  - Path Editing and Motion Tiles
  - Synthesis-based on description
- Need to get beyond clips
Footskate Cleanup
Kovar, Schreiner, Gleicher ’02

- Address the most common constraint
- Footplants are the primary connection between character and world
- Problems are very noticeable
Goals of our method

- **Precise**: Footplants enforced exactly
- **Unobtrusive**: Avoid adding noticeable new artifacts
- **Simple and efficient**: No nonlinear optimizations
- **Local**: Only a small neighborhood needed to solve each frame
- **Fast and Reliable**: closed form math
Key insights

- Some things are noticeable
  - Small amounts of footskate
  - The addition of discontinuities
  - Unnatural accelerations of limbs

- Some things are not noticeable
  - Low frequencies added to motions
  - Small amount of limb stretch
Basic Idea

- Place foot to meet constraints
  - Adjust leg to meet ankle
- Make sure body is close enough
  - Need to be close enough to both feet
- Adjust leg angles
  - Standard single-limb inverse kinematics
- Blend results onto free frames
  - Avoid discontinuities at constraint switches
The Details Matter...

- Get geometric calculations right
  - Continuity when there aren’t switches
- Careful when constraints switch
  - Blend foot positions and orientations
  - Blend root position, when possible
- Avoid unnatural accelerations
  - Fast speeds look like discontinuity
  - Just as unnatural
How to use this

- Editing methods just need to get close
  - Avoid nasty artifacts (high-frequencies)
- Footskate cleanup fixes many important details

- Makes editing methods easier to devise and implement
A useful example: Path Editing  

- How to change a straight motion into a curved one (or vice versa)

- Abstract control:
  - Path as an abstraction of the motion
  - Path as a control

- Fit low order polynomial curves (splines)
  - Provides abstraction
  - Provides easy control over large scale

- Factor into path and details
Path Editing Details

- Need to consider orientation
  - Oriented coordinate system for factoring

- Need to avoid stretching
  - Arc-length parameterization of path

- Easy to do
  - Programming assignment in class!
Path Editing Video

- Show path.avi
Beyond Paths

- Paths are great for movement
- Fail for location dependent actions
  - Can’t position them (arc-length pulls them)
  - Details of movement connect to path (cusps and sharp turns)

- Solution: add rigid “Tiles”
  - Annotate motions when path editing works
Motion Tiles
Selle and Gleicher ’03

- Rigid tiles for “location specific” actions
  - Position and orient them
- Non-stretchable paths to connect them

- Interactive editing paradigm
  - Requires non-linear constraints to preserve arc-length
- Modular footskate cleanup implementation
  - Allows us to have good motions from all of our systems
Motion Tiles Video
Getting Beyond Clips

- Want to generate a wider range
- Want more control

Applications need streams of motion
  - Dynamically generated

Applications need “long clips”
Idea: Put Clips Together

■ New motions from pieces of old ones!

■ Good news:
  ■ Keeps the qualities of the original (with care)
  ■ Can create long and novel “streams” (keep putting clips together)

■ Challenges:
  ■ How to connect clips?
  ■ How to decide what clips to connect?
Connecting Clips
Transition Generation

- Transitions between motions can be hard

- Simple method work *sometimes*
  - Blends between aligned motions
  - Cleanup footskate artifacts

- Just need to know when is “sometime”
Identifying Transition Points

1) Initial frames
2) Extract windows
3) Convert to point clouds
4) Align point clouds and sum squared distances
Start with a database of motions, each with type and constraint information.

Goal: add transitions at opportune points.
Motion Graphs

Idea: automatically add transitions within a motion database

- Walk straight
- Turn 90 degrees

Edge = clip
Node = choice point
Walk = motion

Quality: restrict transitions
Control: build walks that meet constraints
Using a motion graph

- Any walk on the graph is a valid motion
- Generate walks to meet goals
  - Random walks (screen savers)
  - Search to meet constraints
- Other Motion Graph-like projects elsewhere
  - Differ in details, and attention to detail
Motion Graph Examples

- Play pathFit.avi
- Play pathFit-MultiStyle.avi
What’s next...

- Better synthesis
  - Self-awareness (what can you do?)
  - Parameterized motions
  - Better goal specifications
  - Low-cost runtimes
- Multiple interacting characters
- Crowds
- Characters
Snap-Together Motion
(w/ Hyun-Joon Shin, Andy Jepsen, Lucas Kovar)

- Low-cost/highly controllable runtimes
  - Pre-compute all transitions
  - Contrived graph structures
  - Multi-way transitions are hub nodes
- User guided authoring process
Crowds

- Aggregate behavior is important
- Individuals are not
  - As long as they are plausible

- Real world example:
  - Feeding of the hounds at Cheverny
  - 90 dogs, kennel master knows them
Thanks!

- To the UW graphics gang.
- Animation research at UW is sponsored by the National Science Foundation, Microsoft, and the Wisconsin University and Industrial Relations program.
- House of Moves, IBM, Alias/Wavefront, Discreet, Pixar and Intel have given us stuff.
- House of Moves, Ohio State ACCAD, and Demian Gordon for data.
- And to all our friends in the business who have given us data and inspiration.