Animation by Example: Concatenation Approaches

Michael Gleicher
and the UW Graphics Group
University of Wisconsin-Madison
www.cs.wisc.edu/~gleicher
www.cs.wisc.edu/graphics

Review of Lectures 1-3
- Represent Humans by Skeleton
- Create Motions by Motion Capture
  - Record specific movements
  - Motions as vector-values signals
- Blends to combine similar motions
- Transitions by blending
  - Only for similar motions

The Challenge
- High Quality, Expressive Motion
  - Need motion capture (examples)
- Flexible, long-running, controllable
  - Need synthesis
- Synthesis from Examples!

Survey of Techniques
Flexibility:
- Link motions to make sequences (today)
- Blend motions to gain control (next lecture)

Survey of Projects
- Motion Graphs
  - Link motions to make long sequences
- Snap Together Motion
  - Synthesis for interactive systems
- Match Webs
  - Find related motions in a database
- Registration Curves + Parametric Families
  - Combine motions to make spaces
- Plus some others...

Work with Lucas Kovar, Hyun Joon Shin, ...

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
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

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”

When are motions similar?
- What do we mean by similar?
- Believe that blending will “work”
- Heuristic based on geometry
  - Not perfect method
    - Measure and use threshold
    - Apply threshold conservatively
What is Similar?

- Factor out invariances and measure
  1) Initial frames
  2) Extract windows
  3) Convert to point clouds
  4) Align point clouds and sum squared distances

Other methods

- Compare angles
  - But how to weight?
- Use derivatives
  - But what scale? What order?
- Root-orientic (no optimization)
  - But what about curvature?
- Different people have suggested different methods

An easy point to miss:
Motions are Made Similar

- “Undo” the differences from invariances when assembling
- Rigidly transform motions to connect

Building a Motion Graph

- Find Matching States in Motions

Idea: automatically add transitions within a motion database

Quality: restrict transitions
Control: build walks that meet constraints

Motion Graphs
Kovar, Gleicher, Pighin ‘02

Start with a database of motions, each with type and constraint information.
Goal: add transitions at opportune points.

Other Motion Graph-like projects elsewhere
Differ in details, and attention to detail
Automatic Graph Construction

- Find many matches (opportunistic)
- Good: Automatic
- Good: Lots of choices

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

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

An example:
- Building a Motion Graph

An example:
- Using a Motion Graph

- Given a path
- Find a motion that minimizes distance
- Combinatorial optimization
Why is this good?
- Search the graphs for motions
- Look ahead avoids getting stuck
- Cleanup motions as generated
- Plan “around” missing transitions
- Optimization gets close as possible

Not OK for Interactive Apps!
Need different tradeoffs

What about interactive?
- Different set of tradeoffs!
- Runtime must be:
  - Responsive
  - Low overhead
- Willing to sacrifice quality to get

Contrived Graph Structure?
Search: Look ahead to get where you need to go
React: Always lots of choices. Something close to need.

Gamers use these

Snapable Motions
- What if motions matched exactly?
  - Match both state and derivatives
  - Match reasonably at a larger scale
Make motions match exactly
- Add in displacement maps
- Bumps we add to motions
- Modify motions to common pose
- Compute changes at author time

Semi-Automatic Graph Construction
- Pick set of *match frames*
- User selects
- System picks “best” one
- Modify motions to build hub node
- Check graph and transitions

Automatic Authoring
Original Motion
Similar frames
- Base pose
- Snappable Motion

Building the Motion Graphs

Runtime
- Synthesized Motion

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.