Motion Synthesis By Example

Animating characters in games from data

Michael Gleicher
Dept of Computer Sciences
University of Wisconsin - Madison
Synopsis

Game characters are getting pretty good
  – How are they doing it?

Synthesis-By-Example is the prevailing approach
  – What is it? Why does everyone like it?

SBE Methods are build from some basic pieces
  – How do they work? What are the building blocks?

Research uses these same building blocks
  – Can research help real games?
Where is this going?

Example video from Heck&Gleicher ‘07
What did you just see?

• Interactively controllable character

• Stream of high-quality motion

• Continuous control (not discrete choices)

• How did it do it?
  – Dynamically assemble motions by putting together pre-recorded clips (lots of them)
Roadmap

• Basics of Synthesis-By-Example (SBE)
  – Concatenation as an example

• SBE in Practice vs. Research
  – Blending-based parameterization as an example

• Creating controllable characters
  – Parametric graphs as an example
What do I mean by motion?
Game characters are getting pretty good
Why are interactive characters hard?

Human Motion is:
• Complex
• Diverse
• Subtle

Expressive
(but express the right things)

Game Characters must be:
• Efficient
• Dynamically controlled
• Responsive
• Situated (-> precise)

Interactive
Games characters are getting pretty good
Games characters are getting pretty good

But Wait!
That’s a Cut Scene!
Motion Capture
(and other recorded motion)

Motion Capture has Matured
• High-End systems evolved
• Low-End systems emerging

Partnership:
Actor, Director, Technologist

Keyframed motion is similar
Pre-Recorded Motions are Easy (ok, easier)

• Motion is a set of geometric measurements
  – Positions, angles over time

• Easy to use – just play it back

• Motion is just data
  – Artist / Performer gave us what we want
  – We don’t know what or why (or need to)

• Individual examples of one movement

• Doesn’t provide interactivity / controllability
How to make *interactive* characters?

Two Approaches

**Model-Based / Algorithmic**
- Generate motions algorithmically
- Craft methods for motions
- Motion complexity handled by clever algorithms
- Develop models per motion
- Have a motion model
  - Generate more motion

**Synthesis-By-Example**
- Assemble new motions from example data
- Simple, generic algorithms
- Motion complexity comes from example data
- No per-motion models
- No motion model
  - Limited adaptability
Example-Based Synthesis

*Capture* the detail, subtlety and complexity

Good News:
We don’t need to model all the complex things!

Bad News:
We don’t have a model to generate what we didn’t capture!
Roadmap

• Basics of Synthesis-By-Example (SBE)
  – Concatenation as an example

• SBE in Practice vs. Research
  – Blending-based parameterization as an example

• Creating controllable characters
  – Parametric graphs as an example
Synthesis By Example

Create what you need from what you have

Have: Lots of Clips

Want: Long Streams
Want: Controllable
Want: Precise/Continuous
Concatenation

Put clip after clip after clip ...
Transitions

Some transitions are easy

Some transitions are hard
Simple Transition Methods

Cut transition + Blend Transition
Motion Graphs
(aka Move Trees)

Some transitions are easy – remember which
Graph Notation

Edge = clip
Node = choice point

**Graph walk = motion**

Edge = valid transition
Node = clip

**Graph walk = motion**
Concatenation-Based Synthesis

Key Idea:
• Only create transitions where simple transitions are likely to work

Historically (in practice, particularly games)
• Craft motions to have easy transitions

In Research (starting around 2002)
• Find metric to automatically determine what motions are “close” enough for transitions to apply

Kovar et al, Arikan&Forsyth, Lee et al. – All SIGGRAPH 02
Basic Ideas of Synthesis-By-Example

**Off-line Pre-process**

- Database
- Examples

**Run-time synthesis**

- Adjust
- Blend
- Sequence

**Preparation:**
Extract / process example from source data such that assembly methods work

**Assembly:**
At run time assemble examples using a few generic (simple) methods

**Control:**
Choose what is assembled to meet needs (e.g. driven by user, meet goals, …)
## SBE in Practice vs. Research

*(practice has been doing it longer)*

<table>
<thead>
<tr>
<th></th>
<th>Practice (real games)</th>
<th>Research</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preparation:</strong></td>
<td>Planning</td>
<td>Automation</td>
</tr>
<tr>
<td></td>
<td>Careful preparation</td>
<td>Automation</td>
</tr>
<tr>
<td></td>
<td>Manual adjustment</td>
<td>Automation</td>
</tr>
<tr>
<td><strong>Assembly:</strong></td>
<td>Basic methods</td>
<td>Basic methods</td>
</tr>
<tr>
<td></td>
<td>Tweaks thrown in</td>
<td>Tweaks thrown in</td>
</tr>
<tr>
<td><strong>Control:</strong></td>
<td>Carefully crafted&amp;tuned</td>
<td>Search</td>
</tr>
<tr>
<td></td>
<td>Structured data makes problem easier</td>
<td>Pre-Processing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control theory</td>
</tr>
</tbody>
</table>
Determining potential transitions

- Need to account for derivative continuity
- Joint angles are difficult to compare directly
  - Effect of perturbation (e.g., rotate shoulder) depends on pose

- Need coordinate invariance
  - Different camera ≠ different motion!
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

Kovar et al 2002, and others – see Kovar’s thesis for discussion
Building a Motion Graph

- Find Matching States in Motions
Finding Transition Points

Every pair of frames now has a distance.

Transitions are local minima below a threshold.
Motion Graphs
Kovar et al, Arikan&Forsyth, Lee et al. – All SIGGRAPH 02 and many other variants since
Start with a database of motions
Goal: add transitions at opportune points.
Structure of Motion Graphs

Opportunistically built graphs can be hard to search – especially for quick control
Structured vs. Unstructured Graphs

Gleicher et al. I3D 2003
Roadmap

• Basics of Synthesis-By-Example (SBE)
  – Concatenation as an example

• SBE in Practice vs. Research
  – Blending-based parameterization as an example

• Creating controllable characters
  – Parametric graphs as an example
Beyond discrete choices: Parameterization

- How to get a range of movements
- Given only a discrete set of examples
- Result is a motion – this is not just IK
Parameterization by Blending

- Use many examples
- Blend examples to get in-between motions

- General
- Gets specific effects
- Gets subtle coordinated effects
Parameterization by blending

• In research since (at least) Rose et al ’98
• In games for a longer time

• Manual Process
  – Carefully aligned motions
  – Custom crafted parameterizations
**Basic Ideas of Synthesis-By-Example**

**Off-line Pre-process**
- Database
- Examples

**Run-time synthesis**
- Adjust
- Blend
- Sequence

**Preparation:**
Extract / process example from source data such that assembly methods work

**Assembly:**
At run time assemble examples using a few generic (simple) methods

**Control:**
Choose what is assembled to meet needs (e.g. driven by user, meet goals, ...)

---

**Examples:**
- Adjust
- Blend
- Sequence
SBE in Practice vs. Research
(practice has been doing it longer)

<table>
<thead>
<tr>
<th>Practice (real games)</th>
<th>Research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation:</td>
<td></td>
</tr>
<tr>
<td>Planning</td>
<td>Automation</td>
</tr>
<tr>
<td>Careful preparation</td>
<td>Automation</td>
</tr>
<tr>
<td>Manual adjustment</td>
<td>Automation</td>
</tr>
<tr>
<td>Assembly:</td>
<td></td>
</tr>
<tr>
<td>Basic methods</td>
<td>Basic methods</td>
</tr>
<tr>
<td>Tweaks thrown in</td>
<td>Tweaks thrown in</td>
</tr>
<tr>
<td>Control:</td>
<td></td>
</tr>
<tr>
<td>Carefully crafted&amp;tuned</td>
<td>Parameterization</td>
</tr>
<tr>
<td>Structured data makes problem easier</td>
<td></td>
</tr>
</tbody>
</table>
Automating Registration

- Apex of kick corresponds
- Even though motions of different lengths
Automating Blending

Registration Curves

• Automatic Alignment
  – More detailed time warps
  – Other alignments (curvature, constraints)

• Allowed for more things to be blended
  (not just less effort)

Kovar&Gleicher SCA ‘03
Automatically Finding Examples

Match Webs – Query by Example

- Search database for all motions that can be blended with an example (relatively efficiently)
- Finds the registration for each match
- Automatically construct parameterization

Kovar&Gleicher SIGGRAPH ‘04
Automatically find examples in data
What amounts to blend?

- Automatically map controls to blend weights
- Sampling + Scattered Data Interpolation
Automation helps blending!

- Reduce amount of labor
- Makes larger example sets practical
- Allows for more complex blends
- More precise control
  - Better parameterizations
- Surprising what examples it can find
The advantages of blending

More choices!
(potentially infinite)
Not as many examples

From Kovar & Gleicher
SIGGRAPH ‘04
Automated Parameterization

- Build space from all blendable examples
- Blend many examples
- Hard to QA
- Irregular data access
- Unintended diversity
Unintended diversity

• Real data has variability
• Controlled capture reduces it
• An attraction of keyframing

stumbled
distracted by lights
tired at end
wasn’t sure which way to go
Roadmap

• Basics of Synthesis-By-Example (SBE)
  – Concatenation as an example

• SBE in Practice vs. Research
  – Blending-based parameterization as an example

• Creating controllable characters
  – Parametric graphs as an example
Parametric Motion Graphs

Turning parametric clips into a character

**Nodes:** Represent Parametric Motion Spaces

**Edges:** Describe How to Transition Between These Spaces

Like everything else, these have been used in games. Variants have existing for a long time.
Parametric Graphs

• Each node is a range of motions

• Each node has a range of beginnings and endings
Parametric Transitions
How to make transitions between parametric spaces?

• In practice (?)
  – Build spaces that just work
  – Accept the occasional bad blend

• In research
  – Force common pose (Shin&Oh SCA ’06)
  – Find allowable ranges (Heck&Gleicher I3D ‘07)
Method: Data Extraction

*K-Nearest Neighbor Interpolation*
Interactively Controlled Running
Review: Synthesis-By-Example

- Build motions from examples

Simple building blocks
- Concatenation
- Blending

- Research: Automation and advanced control
Why is Synthesis-By-Example so pervasive in games?

Advantage of Synthesis-by-Example

• Actors* are directable  
  (* or good keyframe animators)  
  – Can do a range of things a range of ways  
  – Consistency in performance  
  – Relatively easy to get desired examples

• Get different motions, styles, subtleties
  Without having to model each one

• Easier to scale to diverse repertoire, with acting subtleties, get the directors intent, ...
Disadvantage of SBE: No Model!

- Rely on examples
  - Which may not apply to other situations
- Limited adaptability
  - Simple methods work when "close" to examples
  - Aren’t responsive to the situation
- Larger repertoire (usually)
- Scalability?

2 ways to make things better:

1. Use more examples
2. Use examples better

#2 is more likely to scale up
Lessons of #1 may tell us something
Where does motion come from?

Two Approaches

Model-Based/Algorithmic

• Generate motions algorithmically
• Craft methods for motions
• Motion complexity handled by clever algorithms
• Develop models per motion
• Have a motion model
  – Generate more motion

Approach 2 ½:
Use examples to derive models

Does this get the best (or worst) of both?

• No per-motion models
• No motion model
  – Limited adaptability
Does this help real games?

• Automation means less work
  – Possible for academic to do it (no artists!)
  – But artists don’t like to give up control / QA

• Automation can do more complex things
  – Much larger example sets
  – Much more complex parameterizations
Thanks

• The students who did the work
  – Lucas Kovar, Rachel Heck, Mankyu Sung, ...
• The rest of our group
• The people who supported us financially
  – Original research supported by NSF, UW Grad School
• The people who gave us data and challenges
  – House of Moves / Vicon (Taylor Wilson), EA, Demian Gordon, Rockstar Vienna, (and a whole lot of others)

• You – for listening