Motion Editing and Signal Processing

Michael Gleicher
Department of Computer Sciences
University of Wisconsin-Madison
http://www.cs.wisc.edu/~gleicher
Motion Editing and Signal Processing

- Motion
  - What is motion really?
  - How do we represent it?
- Editing
  - How do we change it?
- Signal Processing
  - A science that provides a useful analytical framework and tools for motion
Outline

Goal: Basic Intuitions for Motion Editing

- “The” motion problem
- Properties in motions
- Signal processing basics
- Applications to motion editing
  - sampling and temporal control
  - filtering and other tools
  - adding, blending and warping
What do we mean by motion?

- **Animated Character**
- **Pose or Configuration**
  - parameters in a vector
    \[ p \in \mathbb{R}^n \]
- **Examples are articulated figures (humans)**
  - trees of rigid links
  - center + joint angles
  - nothing specific about methods

\[ p = \{x, y, \theta_1, \theta_2, \theta_3, \ldots \} \]
What do we mean by motion? (2)

- A motion maps times to configurations
  \[ m(t) \in \mathbb{R} \Rightarrow \mathbb{R}^n \]
- Vector-valued, time-varying signal
- Representation comes from creation
  - typically interpolation
  - may not be convenient for editing
• Many ways to represent a motion
• Different creation methods yield different representations
• Equivalent (in terms of output)

\( h-(t-x)^2 \)
Representation Matters

- Different representations respond differently to change
- Different changes are convenient with different representations

\[ h-(t-x)^2 \]
From poses to motions...

• We have good tools for controlling what happens at a given instant (pose)
• Motions add another level of difficulty
• Lots of poses to change (tedious)
• Changes must be coordinated (hard)
• How to leverage our skill with poses?
Properties of Motion

What do we change? What do we preserve?

Describe motion in abstract terms
- Make it angrier!
- But keep the graceful dignity.

The art of motion editing is crossing this chasm.

High Level Properties

Low Level Properties

Have to control small details
- Poses, joint angles, timing, ...
- Not how we talk about motions.
What makes *this* walk *this* walk?
Quiz: Name That Motion!
A Motion Curve

- Where’s the “Mike”? 
- Where’s the sad? 
- Where’s the walk? 
- Where’s the no skate?

How can we preserve (or alter) these properties if we can’t see them?
Easy to destroy motions

- Hard to identify properties
- Easy to destroy them
- How can we know what kinds of changes to make?

Motion Curve:
Left Knee from Walk

<table>
<thead>
<tr>
<th>Angle (radians)</th>
<th>time (frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.5</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-0.5</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>
Maybe we’re looking the wrong way...

- **Spatial Analysis**
  - Need to consider several curves together

- **Temporal Analysis**
  - How things change
  - Frequency Domain Analysis

- **Signal Processing!**
  - A set of tools for looking at data

Motion Curve:
Left Knee from Walk

-1.5  -1  -0.5  0

Time (frames)

Angle (radians)
Signal Processing Review

• A signal is a value that changes
  – Voltage, current, sound pressure, angles, ...
  – Something that carries information

• Signal theory doesn’t care what the value is
Time Domain Analysis

• What is the value at a particular instant?

\[ v = f(t) \]

• Signal source is a black box

• This “view” lets us answer some questions easily
Frequency Domain Analysis

• A different way to look at signals
• New representation, same information
• Different set of questions easy
• Decompose signals into simpler “building blocks”
Fourier Analysis

- Simple Form: Fourier Series
- Analyze periodic signals with sine waves
- More complex extensions for non-periodic signals
Fourier Analysis

- Decompose ANY periodic signal into sine signals (with some caveats)

\[ \frac{1}{1} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7}, \frac{1}{9}, \frac{1}{11}, \frac{1}{13}, \ldots \]

- Possibly an infinite number of sine signals
Frequency Domain Representation

- Represent a signal by the amount of each “basis signal”
- A signal has high frequencies if its decomposition has these terms
- Allows us to see some general properties of the signal easily
- Fourier Transform converts between time and frequency representations
  - Inverse Fourier Transform converts back
Band-Limited Approximations

- Limited frequencies approximate signals
- More frequencies = better approximations
- Gives intuition for what high frequencies “mean”
Intuitions of Frequency Analysis

- Sharp edges require high frequencies to get good approximations
- “Smooth” signals = no high frequencies

Frequency Analysis says what kinds of things happen in a signal

Existence of properties, without specific details
**Sampling**

- Sample the value of a signal at certain instants
- Can’t know what happened in between without additional information
- Frequency limit on signal insures we don’t miss anything
What’s does this have to do with Motion Editing?

• A Motion is a signal
  – use the same tools as for everything else
• Vocabulary for discussing motions
  – frequency analysis
• Tools for manipulating signals
  – can be applied to our problems
An Important Observation

- Frequency content is an important concept for motions
- High frequencies in a motion are a noticeable thing
  - Our perceptual system is tuned to them
  - Almost always have “meaning” impact, sudden moves, ...
- Care must be taken with high frequencies
Motion is tough

And it’s not just motion capture’s problem

**Motion Capture**
- Sample every frame
  - No structure.
  - No intent.
  - Lots of data!

**Hand Animation**
- Data at Keyframes
  - Structured?
  - Remember why?
  - Maybe lots of data.

Dense samples can represent rich frequency content

Good animators create complex frequency content

Noise, sensor errors

Quality takes work
Approach 1: Key Reduction

• Create “Keyframe” data from samples
• Find splines that fit through data
• Pro:
  – less data to deal with
  – use standard keyframe editing tools
• Con:
  – difficult to get exact fit, accuracy tradeoffs
  – doesn’t recreate structure
Approach 2: Motion Signal Processing

- Apply signal processing techniques to motion
- The motion is just a signal after all
  - signal processing can be independent of representation
- Look to other fields for ideas of things to do to motions
Example 1: Time Manipulations

- Change time, not values
  \[ v = m( w(t) ) \] where \( w \) maps time->time
  - Notice - we don’t say much about \( m \)

- Simple, obvious things to do
  - shift time
  - scale time
  - stretch time
  - warp time
Speeding Up a Motion

- Easy - just scale time: \( m(t) = m(2 \times t) \)

Start with a signal

Take every 2nd sample

Now twice as fast!
Faster!

• That was easy! Let’s try 3 times speed!

Start with a signal

Take every 3rd sample

Now thrice as fast?
Even Faster?

- Hmmmm. How about 3 1/2 times speed?

Start with a signal

Take every 3.5 samples

This look slower!
Aliasing!

- Frequencies that are too high appear as lower frequencies
- Solution: pre-filter
  - make sure signal has no frequencies that are too high before sampling
- Signal processing tells us how to do resampling correctly
Time Warping

- Interpolate corresponding time values
- Different regions scaled differently
Example 2: Frequency Filtering

- Attenuate different frequencies
  - Low Pass: cut out high frequencies
  - Band Pass: allow a certain range through
  - High Pass: cut out low frequencies
- Ideal filters hard to implement
  - Real filters are far from ideal!
- Easy: linear filters
  - FIR: finite impulse response
FIR Filters

- Take a weighted average of nearby samples
- Discrete Convolution
- The weights are called the "kernel"
- Choice of kernel determines function of filter

\[ o(t) = k_0 * i(t-1) + k_1 * i(t) + k_2 * i(t+1) \]

<table>
<thead>
<tr>
<th>in</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

\[ o(2) = k_0 * i(1) + k_1 * i(2) + k_2 * i(3) \]
FIR Filters

- Take a weighted average of nearby samples
- Discrete Convolution
- The weights are called the “kernel”
- Choice of kernel determines function of filter

\[ o(t) = k_0 \ast i(t-1) + k_1 \ast i(t) + k_2 \ast i(t+1) \]

<table>
<thead>
<tr>
<th>in</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

\[ o(3) = k_0 \ast i(2) + k_1 \ast i(3) + k_2 \ast i(4) \]
**FIR Filters**

- Take a weighted average of nearby samples
- Discrete Convolution
- The weights are called the “kernel”
- Choice of kernel determines function of filter

\[
o(t) = k_0 * i(t-1) + k_1 * i(t) + k_2 * i(t+1)
\]

<table>
<thead>
<tr>
<th>in</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

\[
o(4) = k_0 * i(3) + k_1 * i(4) + k_2 * i(5)
\]
FIR Filters

- Take a weighted average of nearby samples
- Discrete Convolution
- The weights are called the “kernel”
- Choice of kernel determines function of filter

\[ o(t) = k_0 * i(t-1) + k_1 * i(t) + k_2 * i(t+1) \]

<table>
<thead>
<tr>
<th>in</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

\[ o(5) = k_0 * i(4) + k_1 * i(5) + k_2 * i(6) \]
FIR Filters

- This example had a kernel size of 3
- Ends require special care
- Trick is to pick the kernels correctly

\[
o(t) = k_0 \cdot i(t-1) + k_1 \cdot i(t) + k_2 \cdot i(t+1)
\]

<table>
<thead>
<tr>
<th>in</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

\[
o(6) = k_0 \cdot i(5) + k_1 \cdot i(6) + k_2 \cdot i(7)
\]
A Simple “Low Pass” Filter

- Very common kernel is B-Spline (approximation to a Gaussian)
- \( k = [\ 1/4, \ 1/2, \ 1/4 \ ] \)
- Soft low-pass filter
- Affects a range of frequencies
  - apply repeatedly to dull lower frequencies
Cascades of filters

- A filter breaks a signal in two
- Chain them together to break a signal into pieces
- Alter each piece, and add them back together

\[
\begin{align*}
\text{in} & \quad F \quad x & \quad x & \quad + & \quad \text{out} \\
& F \quad x & & & \\
& F \quad x & & & \\
& x & & & \\
& & & + & \\
& & & + & \\
& & & & +
\end{align*}
\]
Motion Misconception 1

- Low pass filters do not necessarily remove noise!
  - They remove high frequencies!
- A heuristic for identifying noise
  - Real motions usually don’t have much high frequencies, so if they’re there...
- High frequencies give motions their “snap”
Example 3: Adding / Blending

- Mix two (or more) signals together
- Interpolate between motions
  - Interpolate the character between its pose in the motions
- Get poses in between two originals
  - may or may not be meaningful
- Must do interpolation between poses
  - some kinds of signals (quaternions) don’t really add
Time Alignment

• Poses must correspond for blends to make any sense
  – Halfway between lying and standing?
  – Halfway between foot up and down in walk?

• Warp time to make things work out
  – Manually specify correspondences
  – Automatic dynamic time warping
Uses of Blending

• Blends work if they are quick
  – What’s halfway between a walk and a run?
  – Who cares if it’s really brief!

• Transitions
  – Make sure the poses meet up

• Looping
  – Transition between the beginning and end
Does this work?

Yes!
• Easy to do!
• Use for short overlaps
• Representation Independent!

No!
• No guarantees
• Unprincipled
• Need Time Alignment
• Need Corresponding Motions
• Must deal with angles
How about adding something else?

- Create a “special motion” just for adding into a motion
- Pick a special motion with desired properties
  - if you want to preserve the pose at certain times, the motion has zero value
  - if you don’t want to add certain frequencies, pick a motion without those frequencies
**Motion Displacement Maps (aka Motion Warps)**

- Keyframe the displacement maps
  - zero where you don’t want to change
  - interpolate the changes to make

- Compute changes at key frames
  - subtract from original

- Interpolate to propagate changes

Motion Displacement Maps (aka Motion Warps)
Motion Warps

- Choose how to interpolate the changes

Interpolating Cubic

linear
Control of Scale

- Key spacing sets the size of changes
- Representation of original motion doesn’t matter
- Can make any size change you want
Control of Scale

• We specify as much (or little) as we want - not how much representation says!
Motion Warping

It Works!
- Easy to do
- Independent of initial motions form
- Use existing tools
- Propagate changes
- Control of range
- Preserve frequency content of original

It has problems
- No control over non-key frames
- Addition may not be meaningful
- Can’t preserve constraints
- Need interpolating displacement curve
Other Motion Signal Processing Tricks

• Apply any signal processing operation to a motion
  – you may (or may not) get a useful result

• Motion concepts and signal concepts may not map

• There are lots of methods for signal processing to try
  – wave shaping, non-linear filtering, noise reduction, ...
Some other resources

• **Course Materials on-line (coming soon!)**
  
  http://www.cs.wisc.edu/graphics/MotionEditing

• **Motion Editing Mailing List**
  – majordomo@cs.wisc.edu
  – subscribe moedit