Left

right

Retargetting Motion to New Characters

Michael Gleicher
Vision Technology Center
Autodesk
Mountain View, CA

Retargetting Motion to New Characters

Michael Gleicher
Department of Computer Science
University of Wisconsin
Madison, WI

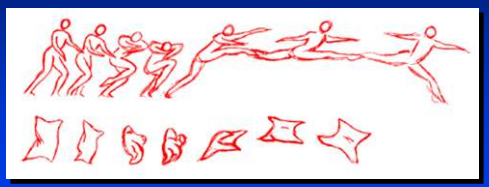
Motion Retargetting

Apply the motion of one character to another

- **Re-use motions created for another character**
- **Target character mimics motions**
  - retain the quality of the original motion
- **Retain, not necessarily improve, the motion**
  - garbage in, garbage out
  - aesthetic appropriateness not considered

Motion Retargetting

Apply the motion of one character to another



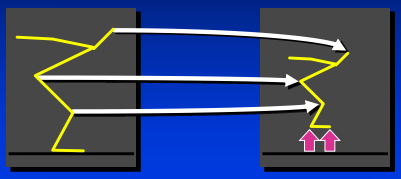
More Specifically

Focus on characters with same structure

- **Corresponding degrees of freedom**
  - example: articulated figures with same joints
- **Use as first step in more general problem**
- **Parameter values can be transferred**
  - results are may not be the same
  - must adapt motion values (curves)

More Specifically

Parameters can be reapplied



But the result may be different

Previous Work

- **Compute a new motion** (parameterized controllers, resynthesis, ...)
- **Adjust each frame or key** (using inverse kinematics or manual adjustment)
- **Apply signal processing to motion**
- **Use Spacetime Constraints**
  - they're not just for physical synthesis

Why didn't you?

- **Synthesize a new motion?**
  We like our original motion!
- **Solve the constraints on each frame using IK?**
  Can't preserve temporal characteristics of motion
- **Use Signal Processing?**
  Doesn't allow for re-satisfying constraints
- **Use global optimization (spacetime constraints)?**

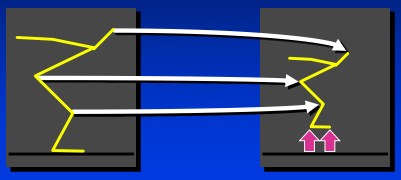
What is Important?

Change what isn't important to retain what is

- **Hard to define what is important**
  - motion specific
  - high-level qualities
- **Stick to what's easy to define**
  - geometric constraints
  - basic signal characteristics

What is Important?

Exact parameter values may not be important



Geometric constraints often are important

Basic Idea 1: Constraints

Basic geometric constraints are the most critical characteristics of a motion

- These constraints must be maintained when applying the motion to a different character
- Retargetting must adapt a motion to re-establish any violated constraints

Basic Idea 2: Frequency content

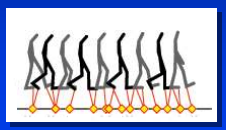
High frequencies are significant characteristics of motions

- Altering high frequencies changes motions
- Adaptations should avoid disturbing high frequencies
- Adaptations (not the underlying motion) should be frequency bounded

Retargetting Recipe

1. Define Constraints

2. Apply to new character



Retargetting Recipe

3. Add Translational Offset
(approximate answer)

4. Solve for adaptation
(solve constraints)

The Constraint Problem

- **Constraints for specific, geometric attributes**
  - specify position of hand, foot on floor
  - joint limits, feet above floor
- **Constraints placed at specific times**
  - create durations as a series of individual times
- **Functions of character's parameters**
  - include character's kinematics

The Constraints (notation)

Motion $\mathbf{m}(t) \in \mathfrak{R} \Rightarrow \mathfrak{R}^n$

Constraint Function $\mathbf{f}(\mathbf{q}) \in \mathfrak{R}^n \Rightarrow \mathfrak{R}$

Constraint $\mathbf{f}(\mathbf{q}) = c$

$\mathbf{f}(\mathbf{q}) \geq c$

Initial Motion $\mathbf{m}_0(t)$

Each constraint has the form:

$\mathbf{f}(\mathbf{m}(t_0)) = c$ **or** $\mathbf{f}(\mathbf{m}(t_0)) \geq c$

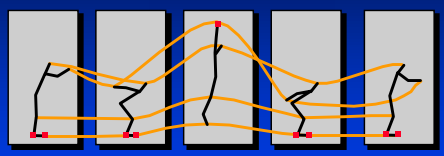
Finding Adaptations

Frequency bounded constraint solving

- **Can't consider frames independently**
  - Individual changes may add high frequencies
  - Need to look ahead and behind
- **Must consider entire motion simultaneously**

Why Spacetime?

solve for* motions*, not frames



Find the best motion that satisfies the constraints

Implementing the Frequency Limit

- **Find a motion that:**
  - satisfies the constraints
  - avoids adding high frequencies
  - minimizes the difference to original
- **Impose frequency limits by selecting the form of the answer**
- **Solve a single, large, numerical problem to compute the adapted motion**

Computing the Adaptation

original motion

$$\mathbf{m}(t) = \mathbf{m}_0(t) + \mathbf{d}(t)$$

motion displacement map

- **solve for d(t)**
- **pick form of d(t) to restrict frequencies**
  - B-Splines, knot spacing sets frequency bound
  - solver computes values for the knots

What Frequency Limit?

Must pick proper frequency bounds

- **Too high?**
  - adds jerkiness to motion
- **Too low?**
  - overfitting makes "big" changes to motion
- **Just right?**

Choosing the Frequency Limit

A Heuristic Method

- **Decompose original motion into frequency bands**
- **Choose highest frequency band with lots of energy**

Or, use trial and error

Constraint Solving: Method 1

Sequential Quadratic Programming (SQP)

- **Too few constraints? Many possible solutions**
  - define an objective function to pick "best"
  - pick simple objective to make easy to solve
- **Constrained minimization**
- **Solve a sequence of linear sub-problems**
  - linearize non-linear constraints at each step

Constraint Solving: Method 2

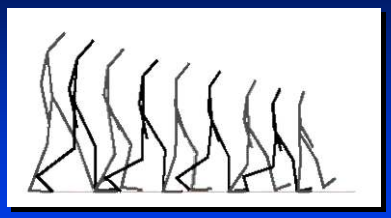
Non-linear least squares

- **Too many constraints? No exact solution**
  - minimize residual error
  - add constraints to make over-determined
- **Unconstrained minimization**
- **Solve a sequence of linear sub-problems**
  - linearize non-linear constraints at each step

Character size not constant

Target size needs to be known in each frame – it doesn't have to be the same

Character size not constant

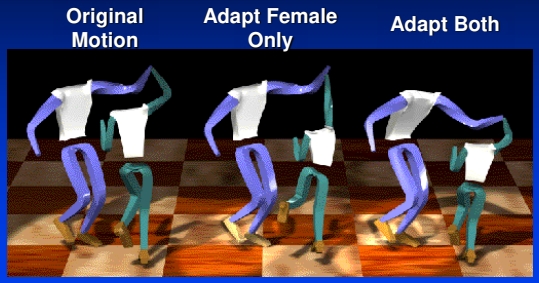


World not constant

Adaptation can change any parameters – not just those for the changed character

- **We can choose which parameters are affected by the adaptation**
- **Solve for everything at once**

World not constant

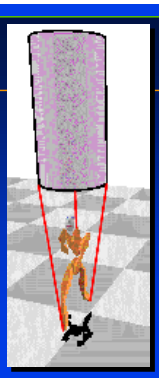


Different Structure

When the parameters aren't the same, the problem is harder

- **Couple corresponding "body" parts**
- **Characters must be similarly sized**
  - Retargetting makes characters the same size
- **Minimize distance between old and new points**
- **Must deal with different numbers of parameters**

Skipping Can Example



1. human motion
2. retarget to can-proportioned human
3. tie corresponding points
4. solve for new motion

Why does it fail?

- **Implementation limitations**
- **Need richer constraints**
  - balance, strength, collision, ballet form, ...
- **Fundamental over-simplifications**
  - similarity computed on poses
  - additive adaptations (no scaling or time-shift)
  - limit of adaptation (sometimes, need new motions)

Some Bloopers

Balance? Posture?

Twitchy hands?

Summary

We can retarget motions created for one character to another

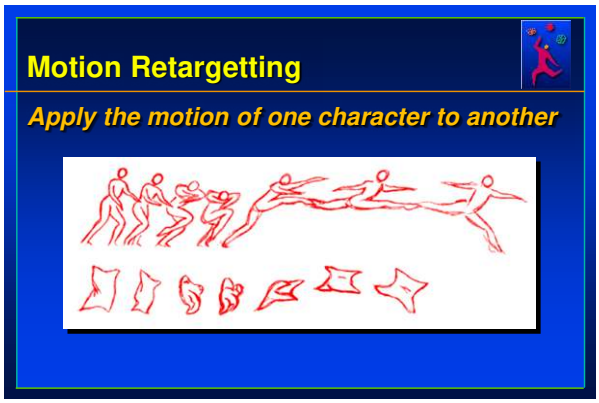
- **Re-establish geometric constraints**
- **Avoid adding high frequencies**
- **Compute adaptation with spacetime constraints**

Because I thought you'd ask....

Answers to frequently asked questions

- **I don't know.**
- **Nothing is specific to mocap. That's just what I had.**
- **Yes, I'd love your examples to try.**
- **The examples take a few seconds on a mac.**
- **The heads were lost in a bad mocap accident.**
- **The method is not specific to articulated figures.**
- **`gleicher@cs.wisc.edu` (after August 24)**

Motion Retargetting



Apply the motion of one character to another