Motion Transformations with Spacetime Constraints

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

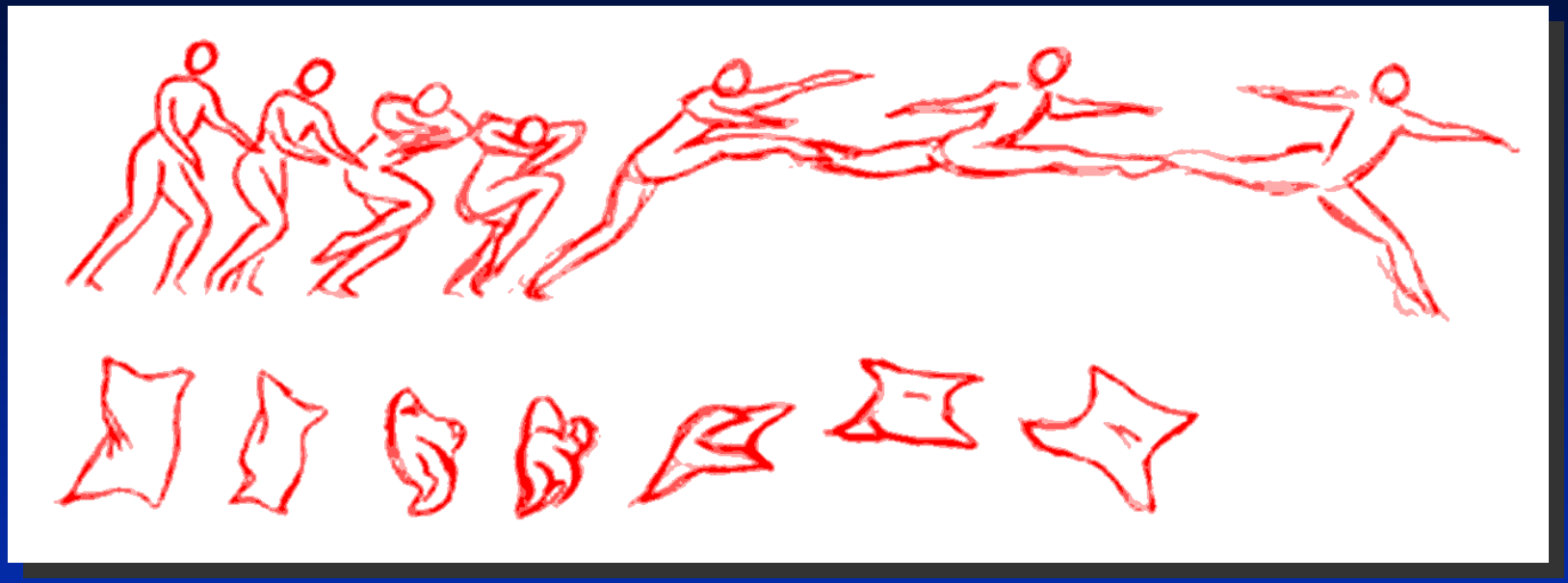
The Dream:
Animation for the rest of us!

- Animation and Special Effects are talent and labor intensive
- Can we make it easier?
  - available to casual users?
  - less labor for pros?
- Projects towards this goal:
  - Motion Adaptation
  - Motion from Performance
  - Video Tracking

Motion Adaptation

- Motion is hard to create
- Easier to borrow, steal, buy, ...
  - Goal: libraries of clip motion
- Most motion is not reusable
  - particular character, action, context...
- Adapt/edit/adjust to be something else?
  - Transform motions
  - wide range of possible transformations

Motion Transformation



Given: Good motion, new needs

Find: New motion

meets new needs

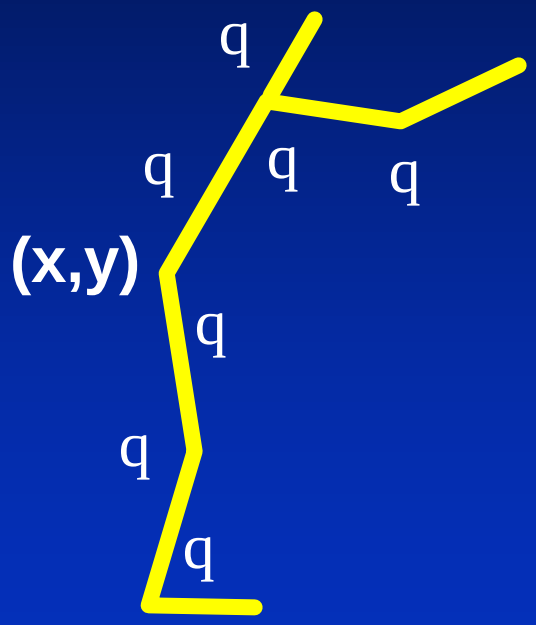
preserves original quality

What do we mean by motion?

- Animated Character
- Pose or Configuration
  - parameters in a vector

$$\mathbf{p} \in \Re^n$$

- Examples are articulated figures (humans)
  - trees of rigid links
  - center + joint angles
  - nothing specific about methods



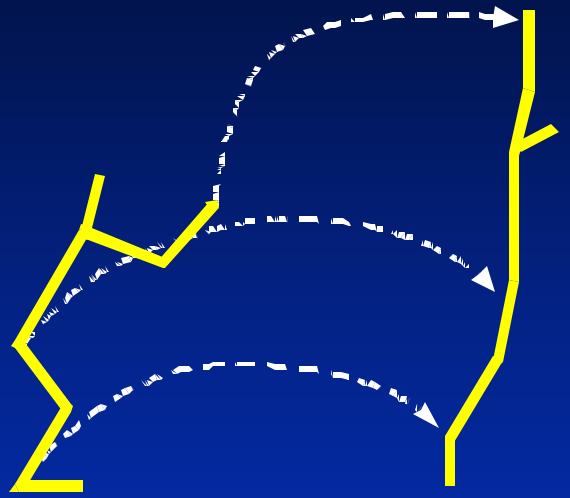
$p = \{x, y, q_1, q_2, q_3, \ldots\}$

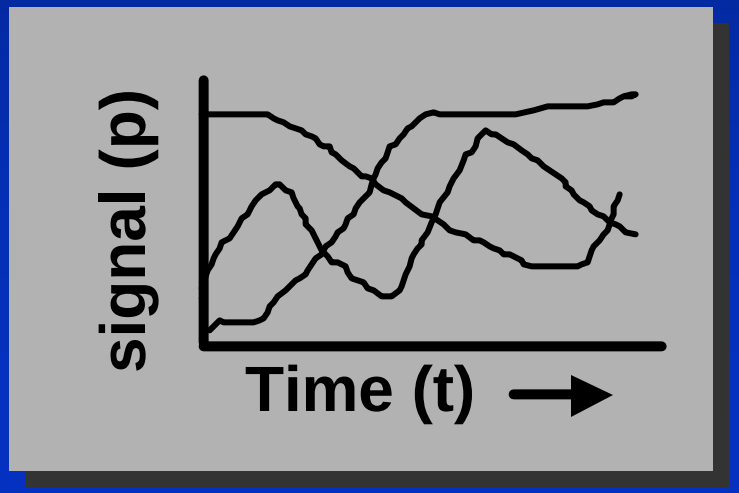
What do we mean by motion? (2)

- A motion maps times to configurations

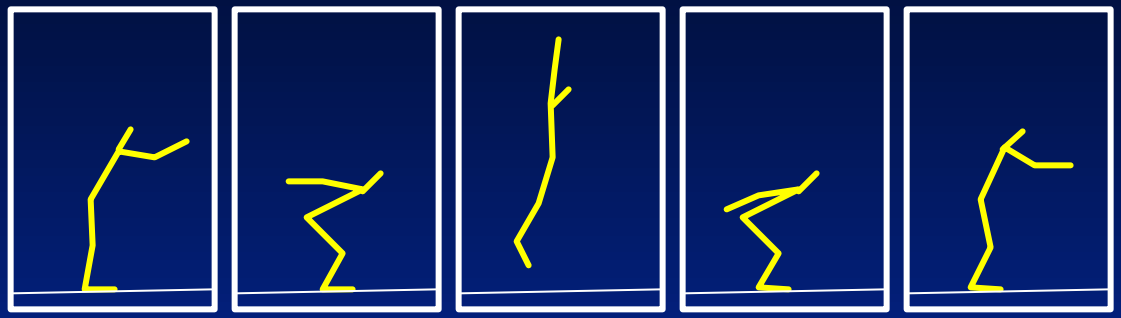
$$\mathbf{m}(t) \in \Re \Rightarrow \Re^n$$

- Vector-valued, time-varying signal
- Representation comes from creation
  - typically interpolation
  - may not be convenient for editing

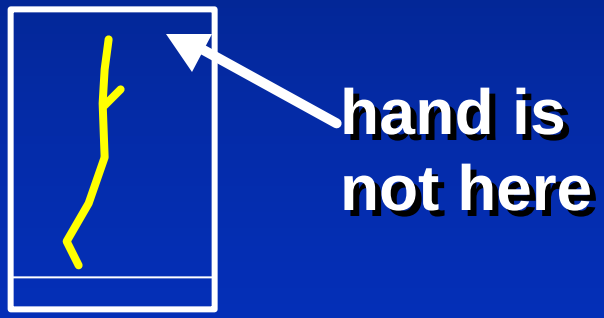




Problem: Motion is Specific

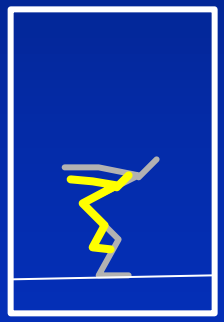


Specific Action



Edit **motion to meet new needs**

Specific Character



different sized character doesn't fit

Retarget **motion to new character**

Previous Techniques

- Generate new motions
  - may lose what we had
- Manually tweak each frame
  - lots of work
  - may not preserve original
- Signal processing
  - works for certain types of alterations
  - may not preserve constraints

Transformation Basics

Change what isn't important to retain what is

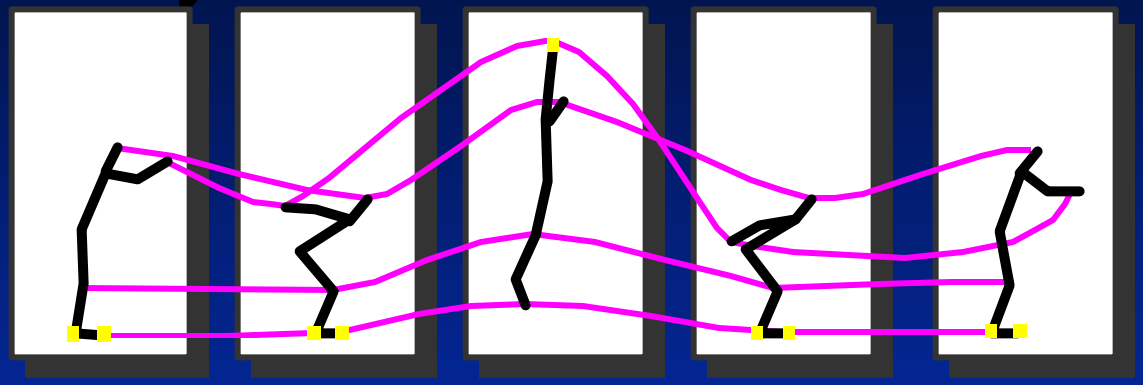
- Hard to define what is important
  - high level properties
  - motion specific
- Stick to what's easy to define
  - geometric constraints
  - basic signal characteristics
  - framework for more metrics later

Transformation as Constrained Optimization

- Find a motion that…
  1. Meets any specific requests
  2. Keeps any specific characteristics of the original
  3. Is as similar as possible to the original
- Quite naturally posed as constrained optimization
  - subject to meeting the constraints (1 and 2) minimize some objective (3)

Spacetime Constraints

Previously: a method for synthesis of physically correct motions



- Consider all constraints simultaneously
  - NOT frame at a time
- Solve for motions
  - “best” motion that meets constraints
- Physics is just a constraint

Why Spacetime?

- Can’t just look at individual frames
  - temporal qualities (smoothness, grace, ...)
  - need to look ahead and behind
- Considering a duration of motion allows for a richer vocabulary of motion qualities

Transformation by Spacetime Constraints

- Define geometric constraints on frames of the motion
  - properties to preserve
  - new goals to establish
- Find new motions that:
  - satisfy constraints
  - match original motion
- Spacetime constraints consider entire motion simultaneously

Ideally:

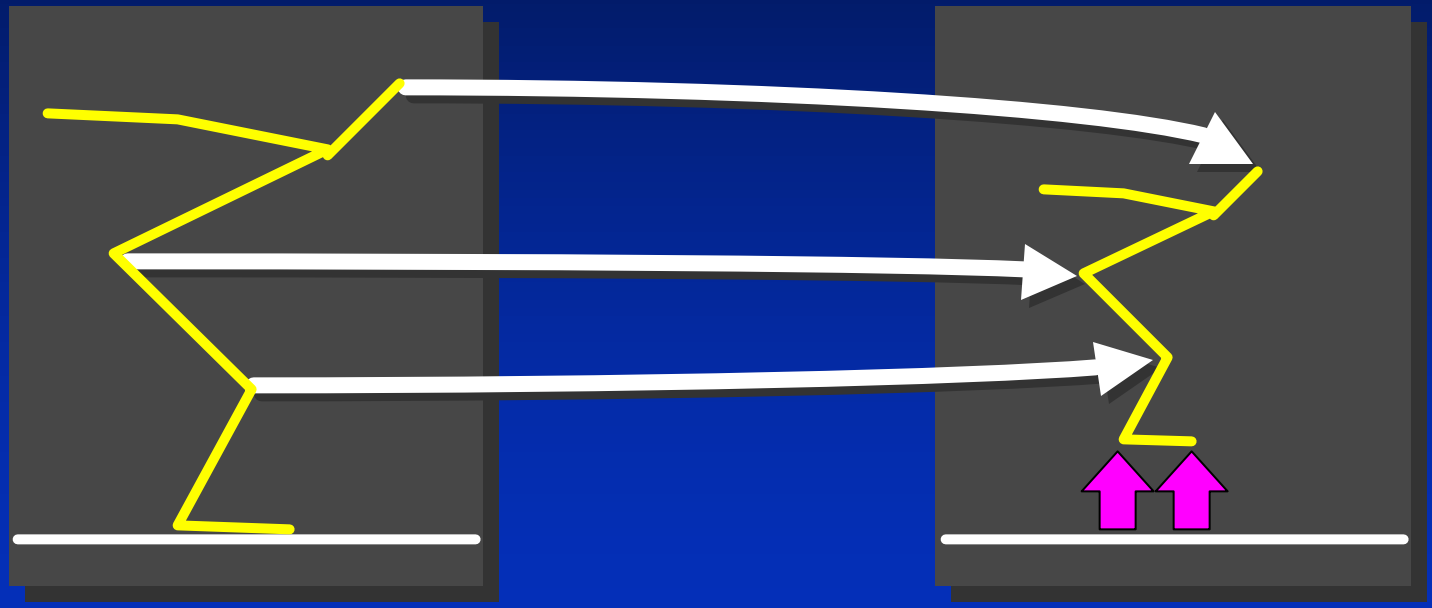
- Define motion concepts mathematically
  - smooth, physical, realistic, walking
  - depressed, energetic, drunken
  - with proper ballet form
  - like Gene Kelley in *Singing in the Rain*
  - *...*
- Determine which are appropriate
  - and which the user cares about
- And employ these complex criteria in the spacetime formulation

In Practice...

- Take a more pragmatic approach
  - choose what's "easy"
  - practical solutions now, better motion metrics later
- Avoid "objectionable" adaptations
  - violate key constraints
  - large magnitudes of change from original
  - frequency characteristics unlike original

Basic Idea 1: Constraints

Exact parameter values may not be important



Geometric constraints often are important

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 motions) should be frequency bounded

Spacetime Constraints for Motion Adaptation

- **Motion Editing with Spacetime Constraints**
  - provide direct manipulation editing
  - adjust constraints over motion
  - emphasize solution speed over quality
- **Retargetting Motion to New Characters**
  - apply motion to differently sized character
  - find adaptation to re-establish constraints
  - avoid uncharacteristic adaptations

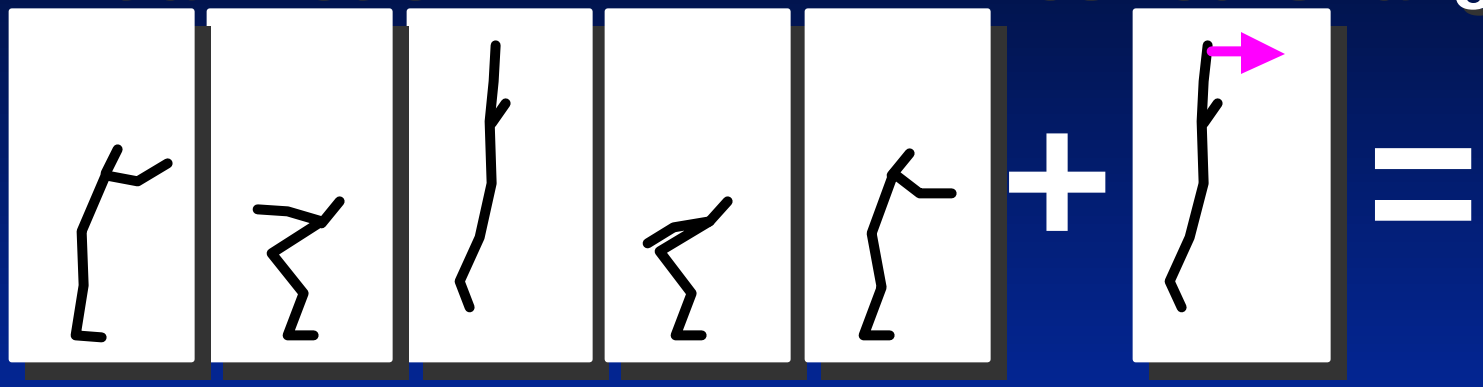
Motion Editing with Spacetime Constraints

- A model for motion editing
  - Constraints over whole motion
  - User adjusts and adds constraints
- Solve for new motions
  - satisfy constraints, preserve motion
  - emphasize fast solutions for interactivity
  - bad solution? user adds more constraints!
- User interface issues
  - Must visualize motion and changes
  - Must specify and edit constraints

An Editing Example

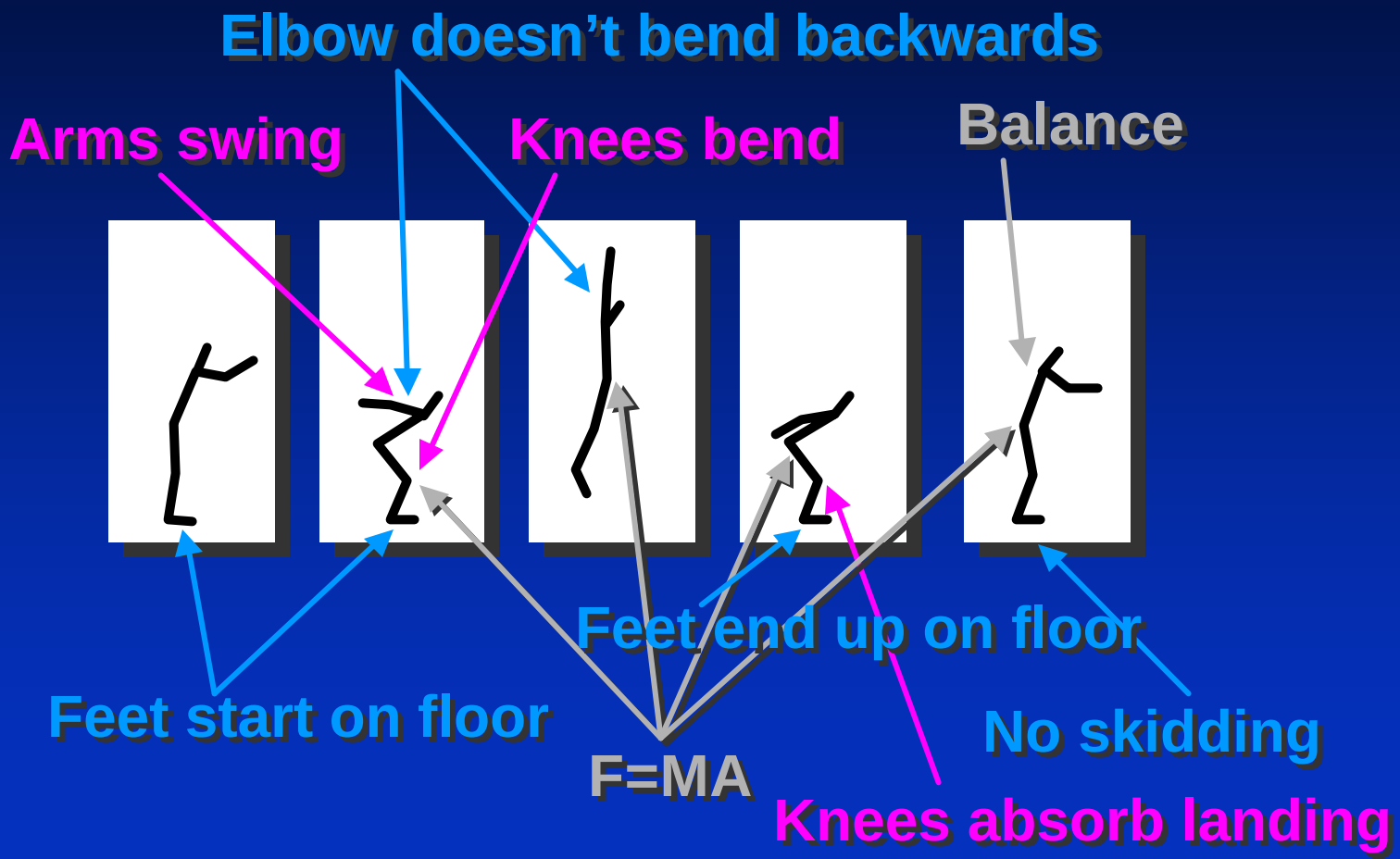
Initial Motion

Desired Change

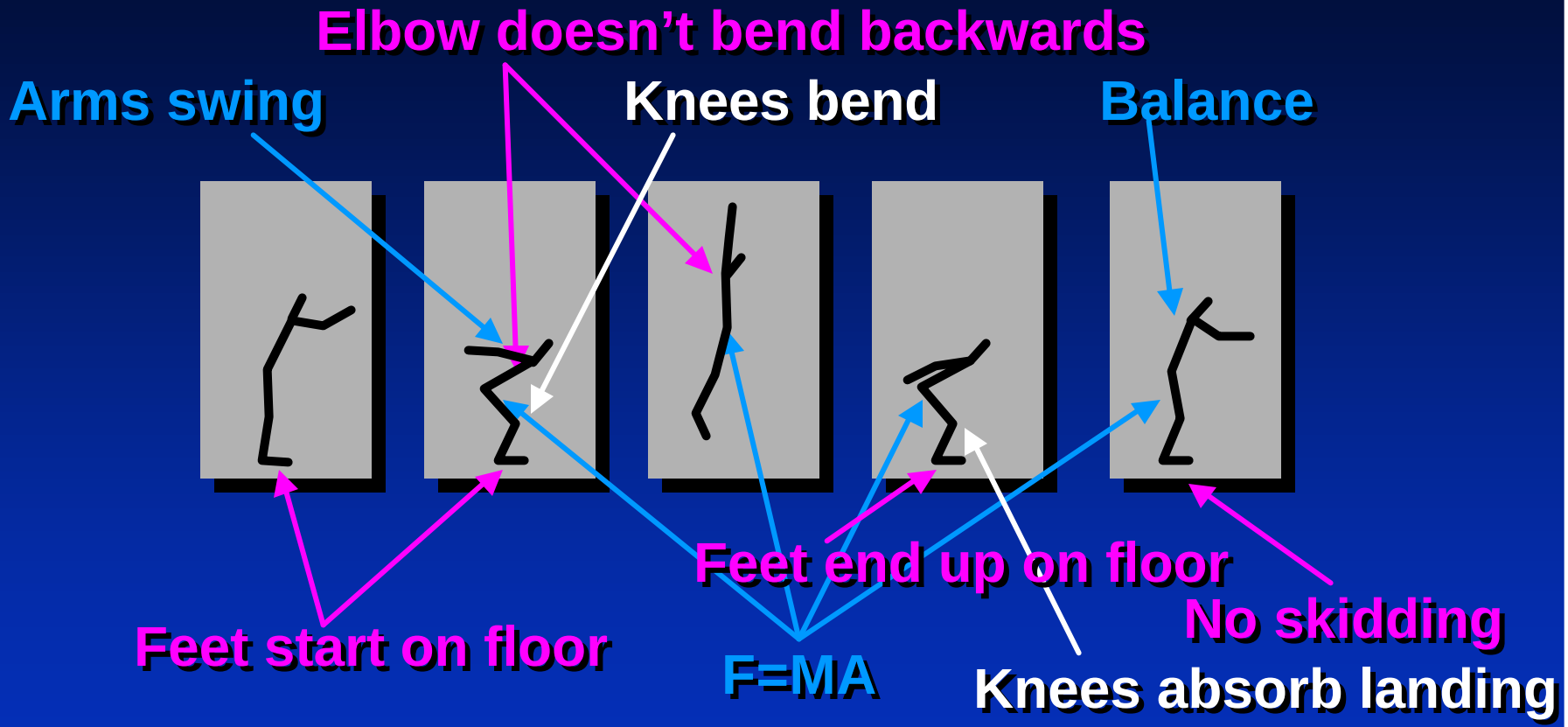


- What is a good answer?
  - Character meets new goal
  - Preserves original
  - Resulting motion is a jump
- Hard to define in general

What makes a jump a jump?



What makes a jump a jump?



Geometric Constraints - implement as constraints

Signal Characteristics - get from signal matching

Other Constraints - could be added later

Mathematically...

- **Configuration:** $\mathbf{p} \in \Re^n$
- **Motion:** $\mathbf{m}(t) \in \Re \Rightarrow \Re^n$
- **Initial Motion:** $\mathbf{m}_0(t)$
- **Constraint:** $f(\mathbf{m}(t)) = c$

  $f(\mathbf{m}(t)) \geq c$
- **Variational Constraint:** $\underset{t \in t_1 \ldots t_2}{\forall} \; f(\mathbf{m}(t)) = c$

  $\underset{t \in t_1 \ldots t_2}{\forall} \; f(\mathbf{m}(t)) \geq c$

The Problem

- Find $\mathbf{m}(t)$ such that
  - the constraints are satisfied $\mathbf{f}(\mathbf{m}(t))=\mathbf{c}$
  - an objective function $g(\mathbf{m}(t))$ is minimized
- A variational, constrained, optimization

The Questions

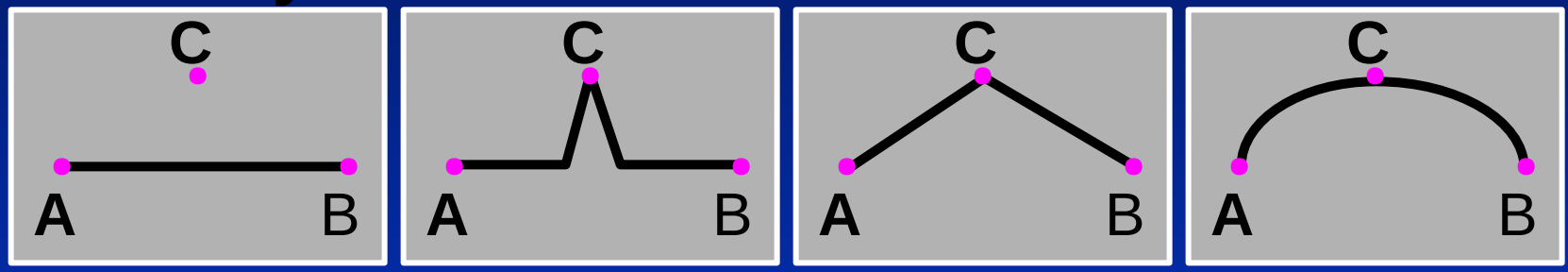
- What $\mathbf{f}$ (constraints)?
- What g (objective)?
- What representation for $\mathbf{m}(t)$?
- How to solve it?
- How to present it to the user?

Constraints

- Describe features of motion
  - limitations on character
  - essential constraints on motion
  - future: physics, form, ...
- Palette of controls for user
- Nonlinear functions, inequalities
- Implement variational by sampling

The Objective

- Measure difference between m(t) and $m_0(t)$
- Many choices



Match Value Velocity Acceleration

- No obvious, general right answer
  - –results are non-intuitive
  - –choice effects solution difficulty
  - –off-load importance with constraints

Strategies (project phases)

1. **Do something simple (show it works)**
2. ~~**Find useful objectives, present to user**~~
3. **Make it go fast**

Why speed over correctness?

- What is “correct” anyway?
- Good objectives too hard to solve
- Constraints preserve important features
- **Interactivity!**

Solving Non-Linear, Constrained Optimization Problems (a very brief primer)

- There is no guaranteed method
- Established, centuries old heuristics...
- Iteratively refine a solution
  - each step moves closer (hopefully)
- Solve series of approximate problems
  - choose solvable sub-problems
  - linear systems, quadratic optimization
- Sequential Quadratic Programming

How to make it fast

- Get a fast computer
- Do good computer science (algorithms/caching/...)
- Exploit sparsity
- Precision isn’t important
  (trade everything for speed)
- Differentialness
- Constrain the search space

Motion Displacement Mapping

- Define $\mathbf{m}(t)=\mathbf{m}_0(t)+\mathbf{d}(t)$
- Search for $\mathbf{d}(t)$

- Advantage: representation independence
  - pick representation for displacement based on desired changes, ease of solution
  - and implicit constraints!
- Use B-Spline displacement curves
  - band limited by definition!

The Numerical Problem

$\mathbf{x}$ = concatenation of B-Spline controls

$\mathbf{d(t)} = B(\mathbf{t},\mathbf{x})$

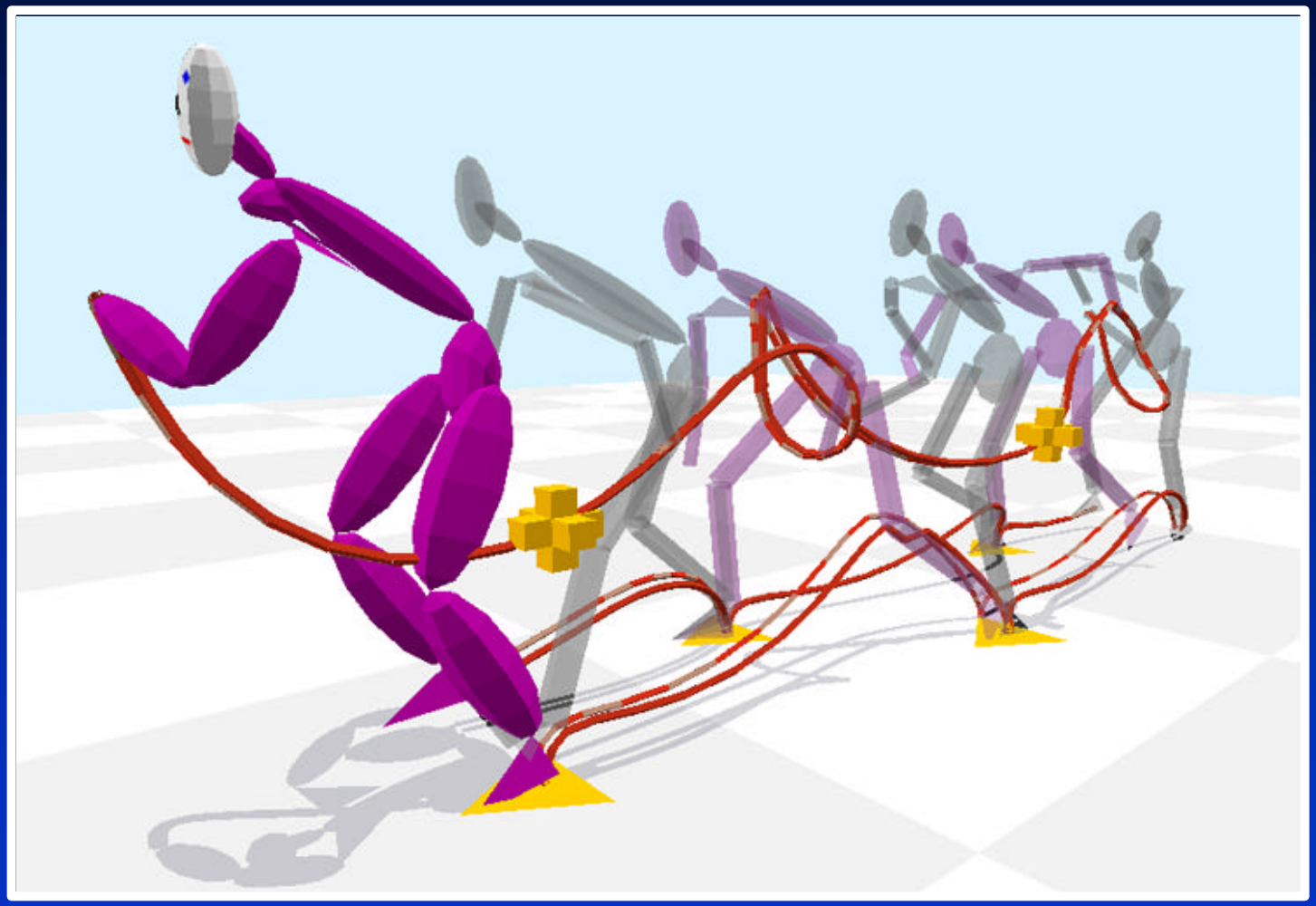
$\mathbf{g(x)} = 1/2\ \mathbf{xMx}$ (**M** is a diagonal matrix)

$\mathbf{f(x)} = \mathbf{c(k(m_0(t)}+B(\mathbf{t},\mathbf{x})))$

k = character's kinematics

c = constraint function

Feedback

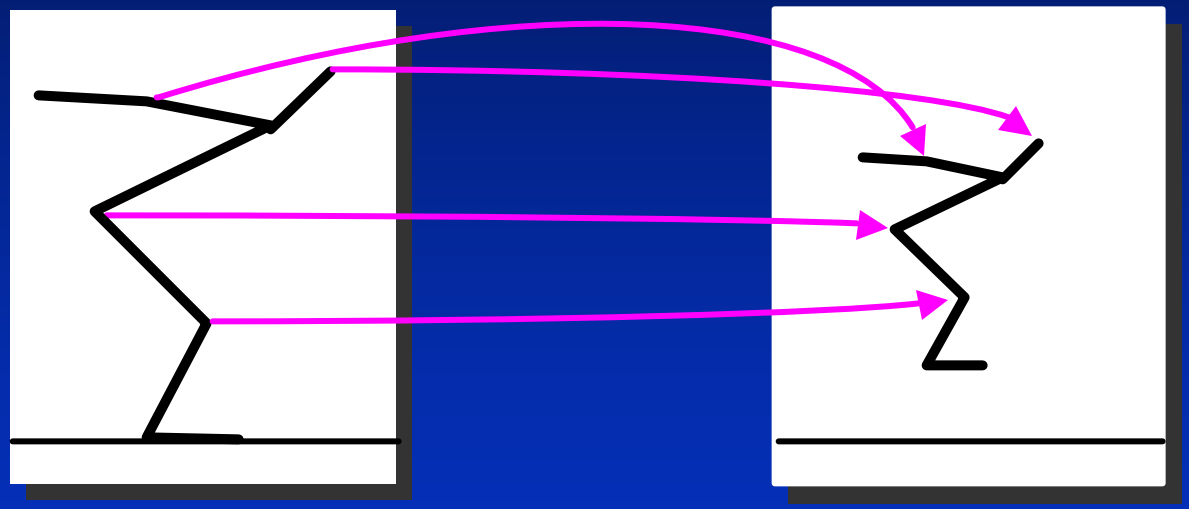


What to look for in the video

- All in real time on a Macintosh
- All interaction is direct manipulation
- Up to 5400 constraints (final example)
  - at most a handful are specified by user
- Various display mechanisms
  - cycling, strobing, stream lines, ...
- Initial solutions OK, but usually adjusted

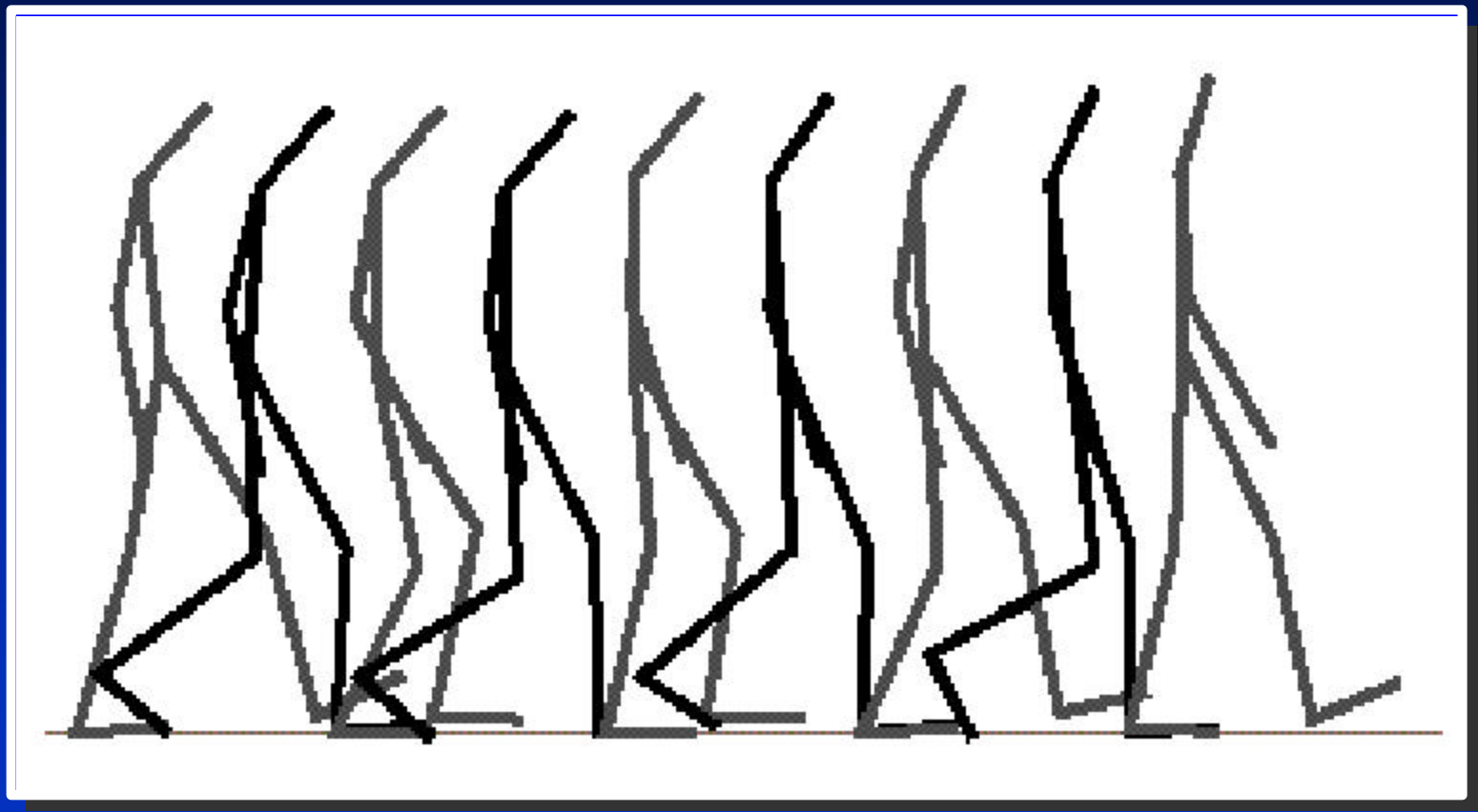
Retargetting Motion to New Characters

- Consider characters with identical structure, but different limb lengths

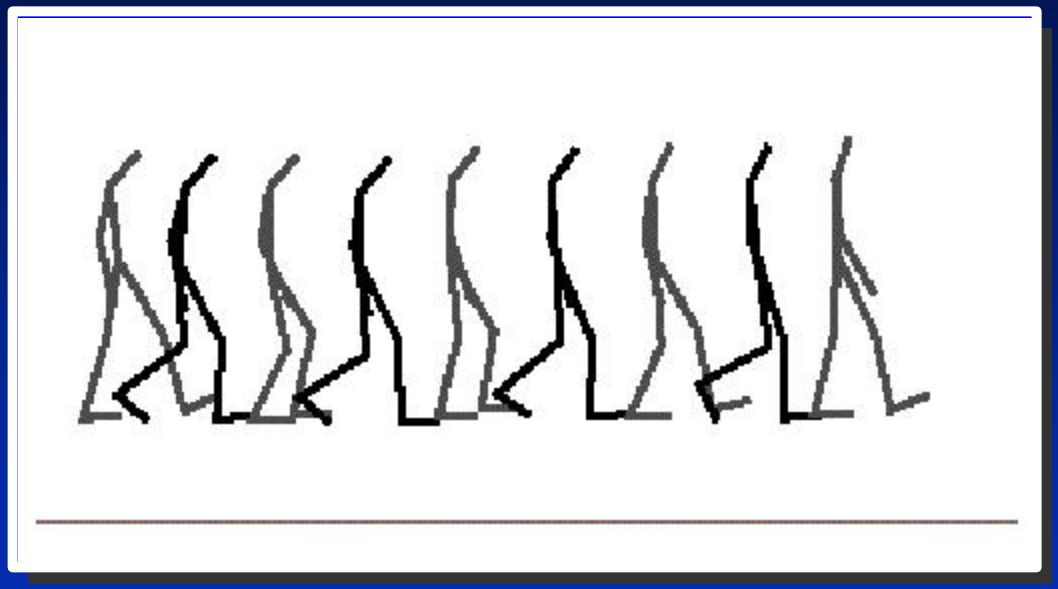


- Parameters can be re-applied
- But some things are different

What makes *this* walk *this* walk?



Hint:
It may not be invariant to the size of the character...

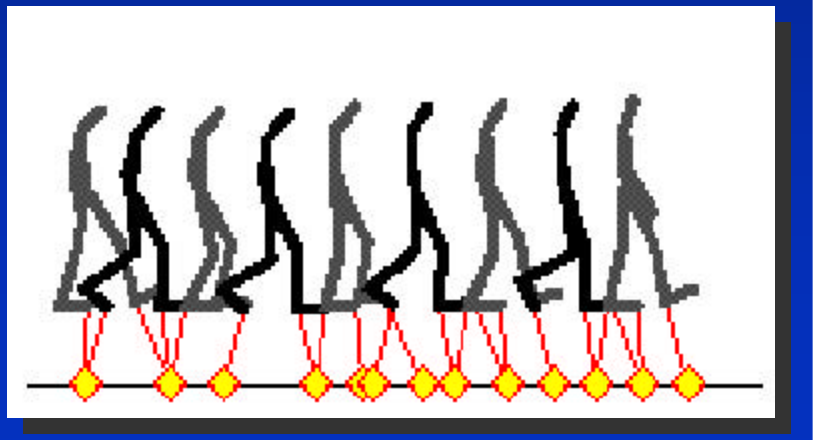


Adaptation of the motion makes it more like original (by the important metrics)

Retargetting Recipe

1. Define Constraints

2. Apply to new character

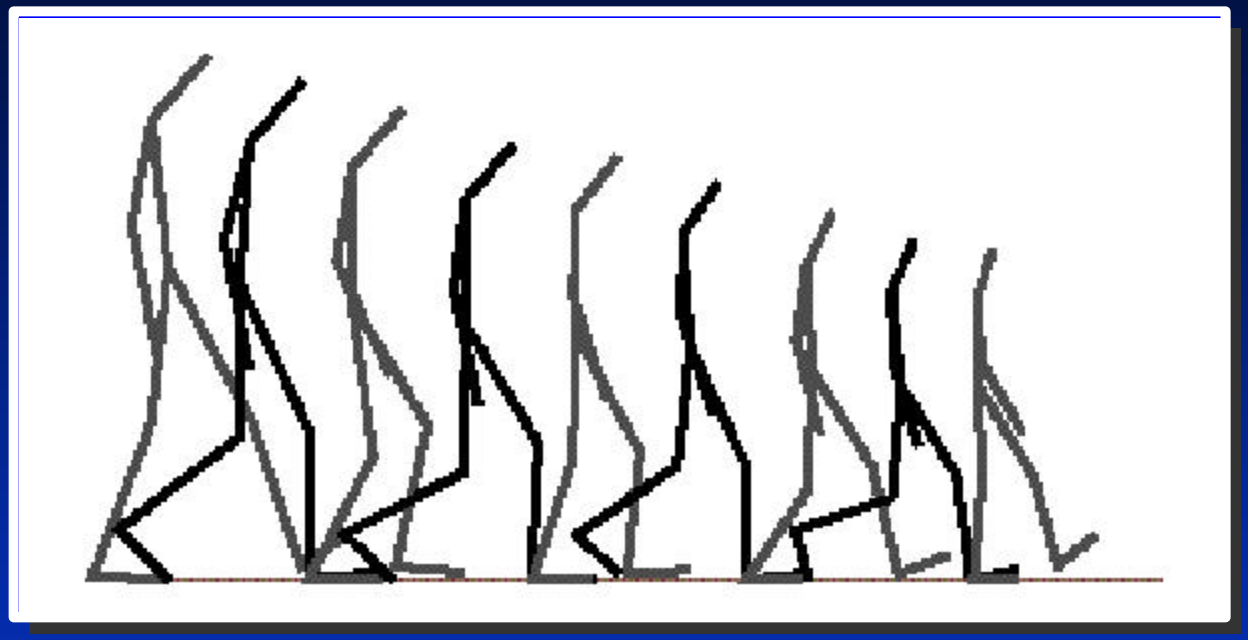


Retargetting Recipe

3. **Add Translational Offset**
   (approximate answer)

4. **Solve for band-limited adaptation**
   (solve constraints)

Motion for Morphing



- Retarget motion to time-varying size

World not constant

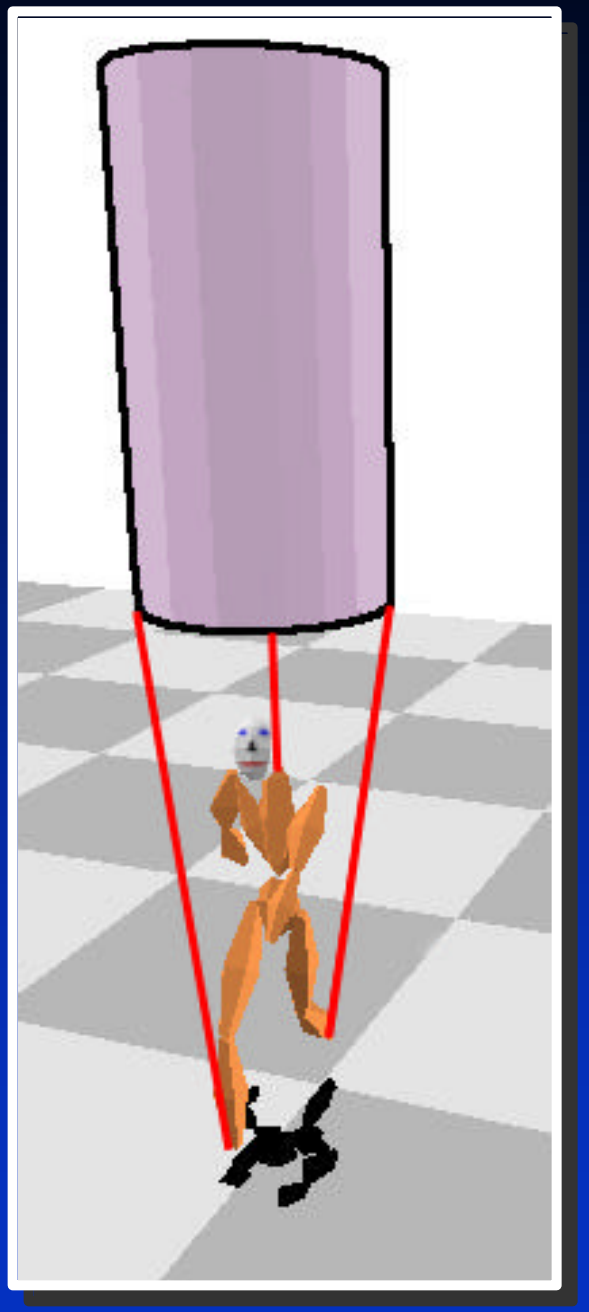
Original Motion

Adapt Female Only

Adapt Both

Characters with Different Structure

- Creativity vs. Automation
- User defines point correspondences
- Adapt similar structure to same size first
- Different numbers of degrees of freedom
  - least squares (too few)
  - objective (too many)



Some Bloopers

Balance?
Posture?

Twitchy
hands?

What next?

Make animation easier to produce!

- **Where do we get motions from?**
  - motion capture, databases, synthesis, ...
- **What high-level properties to find?**
  - how to encode, specify, compute, ...
- **How to automate the process?**
  - constraint detection, property identification, ...
- **How to apply the results?**
  - skelletal motion is only half the battle
  - Automatic Anthropomorphism

Summary

- ***Spacetime Constraints*** **are used to adapt motions.** (not just physical synthesis)
- ***Motion Displacement Maps*** **provide representation independence.**
- **Concessions to pragmatism afford a realistic approach.** (and opportunities for future work)
- **Practical Solutions demonstrated on *real problems.***