Motion Synthesis By Example

Lecture 1b: Synthesis by Example

Michael Gleicher

Dept of Computer Sciences

University of Wisconsin - Madison

Lecture 1

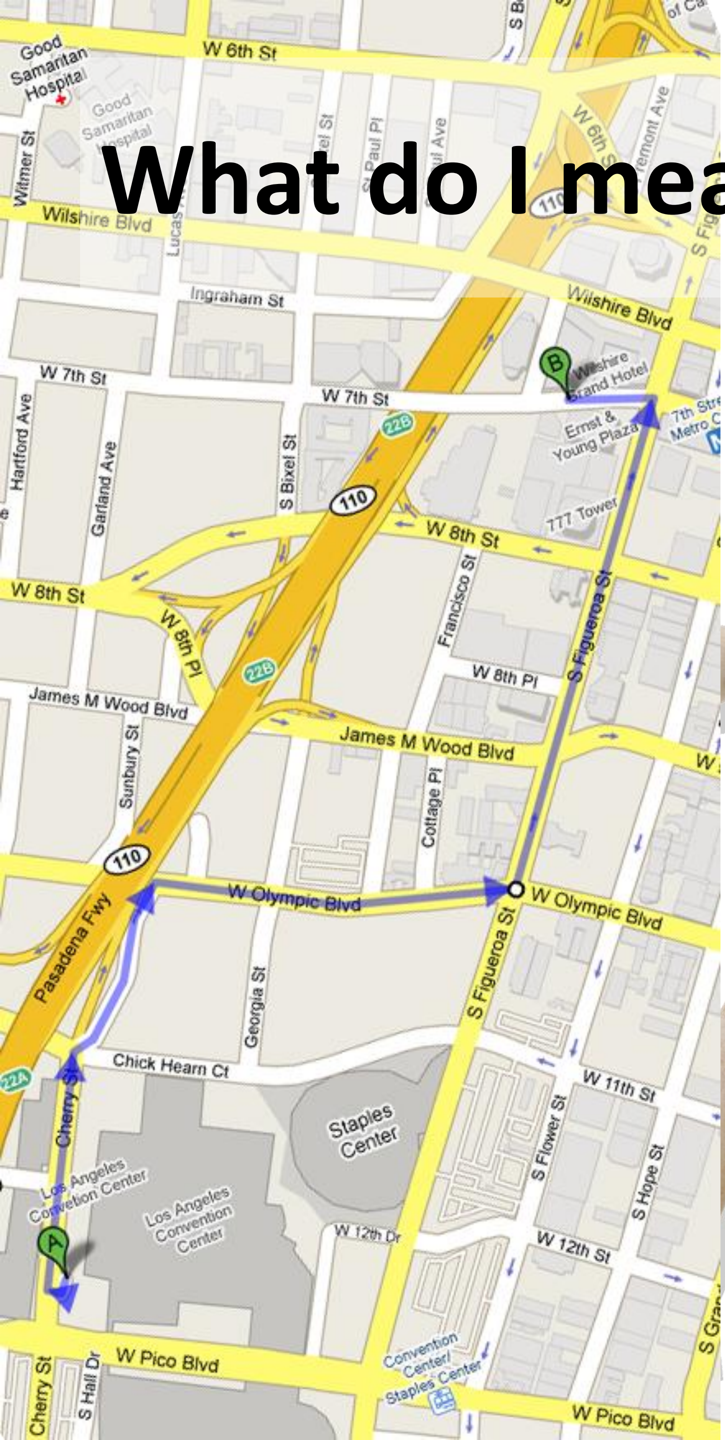
- Some Preliminaries
  - Defining the problem
- Basics of Synthesis-By-Example
  - Defining the building blocks
- Research vs. Practice

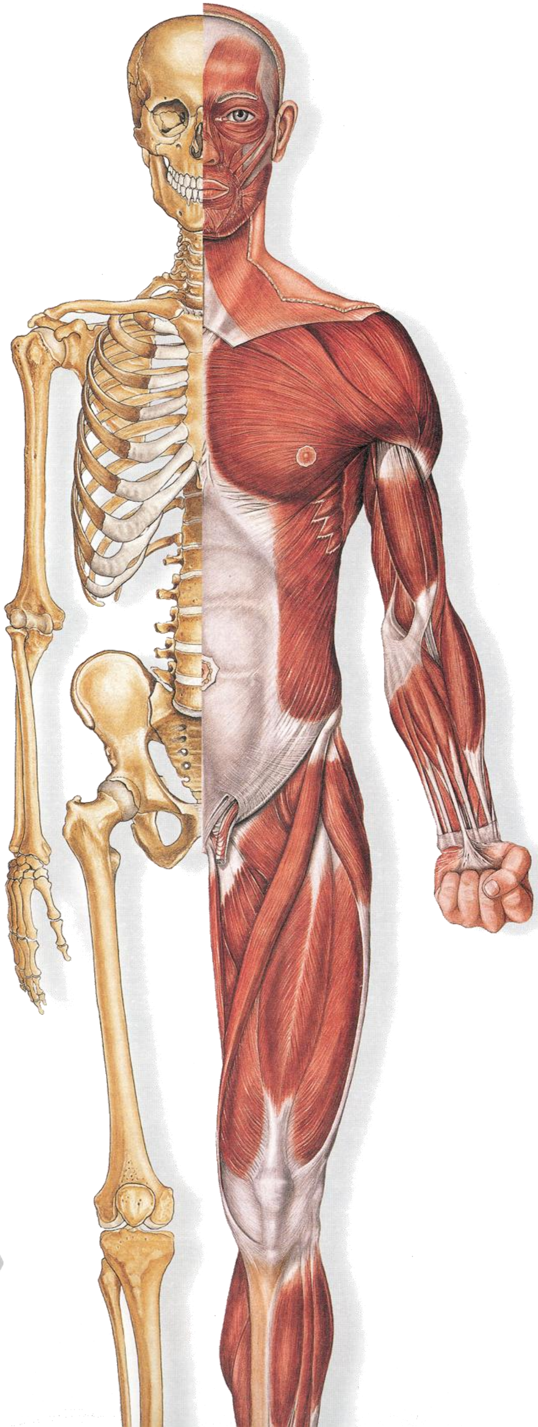
Why?

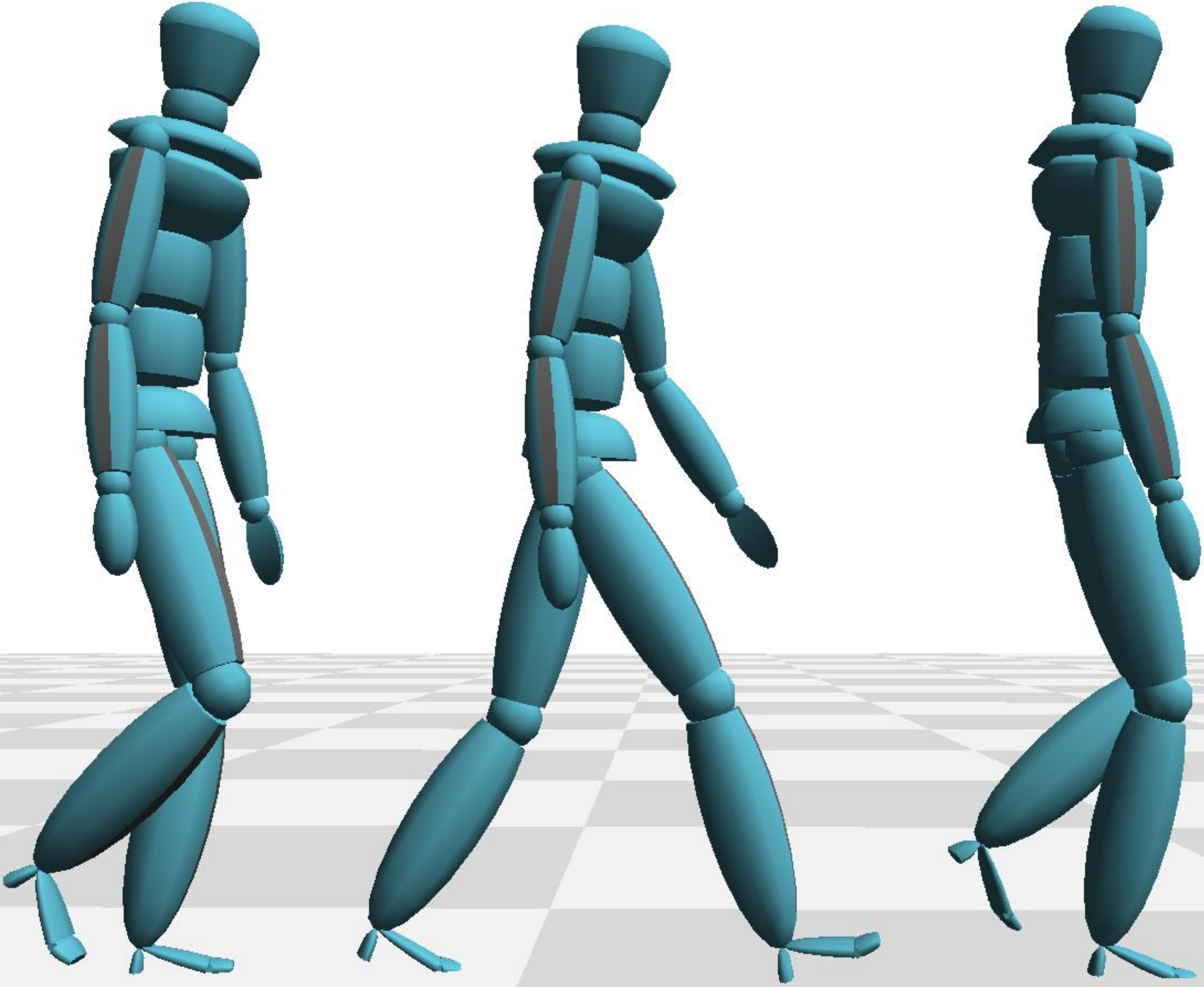
Lecture 1

- Some Preliminaries
  - – Defining the problem
- Basics of Synthesis-By-Example
  - – Defining the building blocks
- Research vs. Practice

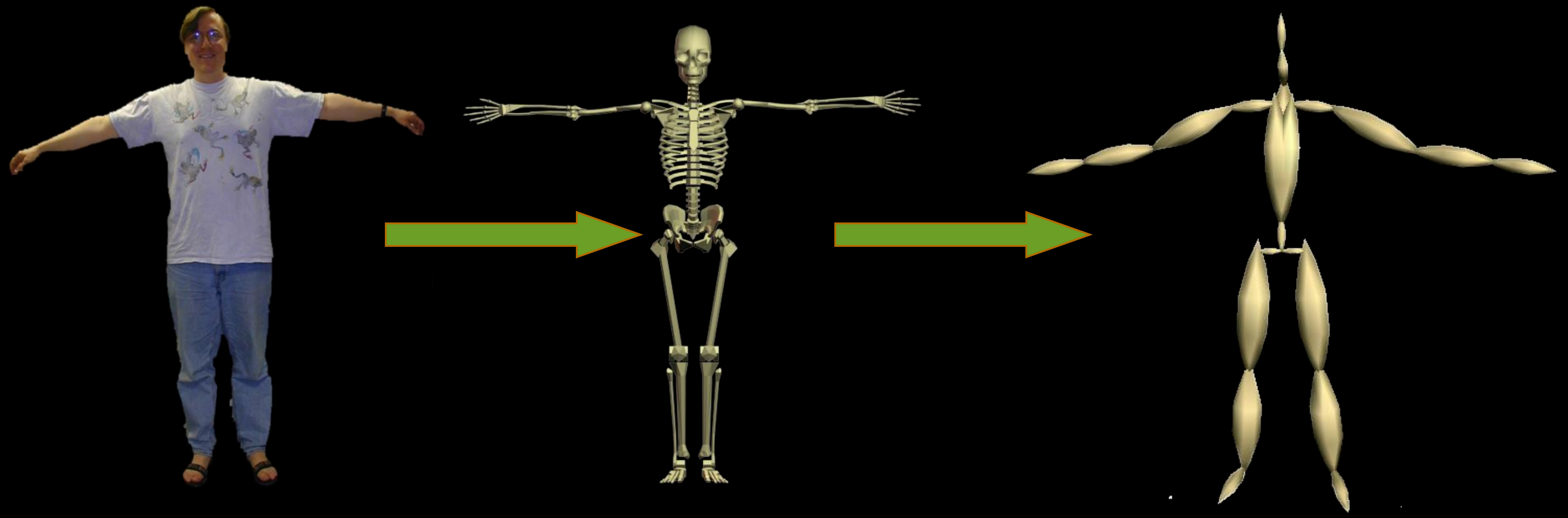
What do I mean by motion?







Skeletal Animation

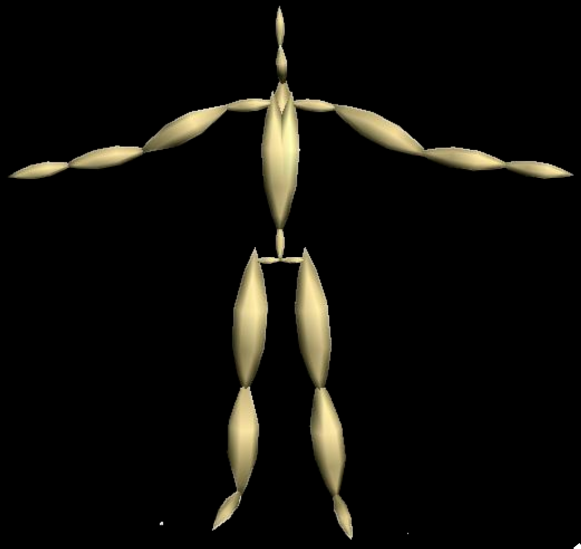


206 bones,
muscles, fat,
organs, clothing,
…

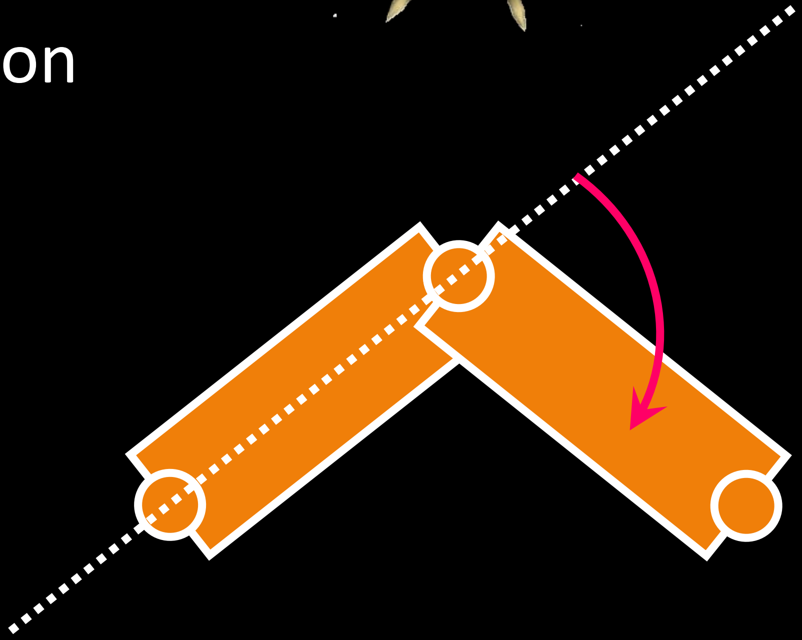
206 bones,
complex joints

53 bones
Kinematic joints

Skeletal Animation



- Translation + Rotations
  - Hard enough
  - Hierarchical Representation
  - Emerging Alternatives
  - Foundations “Exercise”



- Applying to other things is hard and open

- Pose space

  (poses are vectors)

- Motions

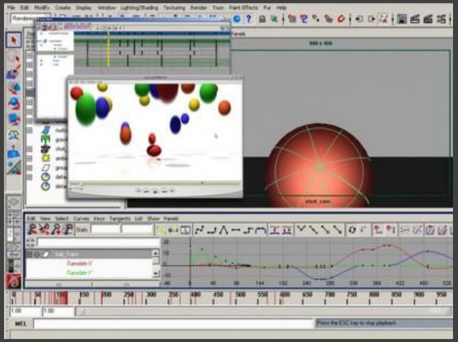
  (motions are functions of time to poses)

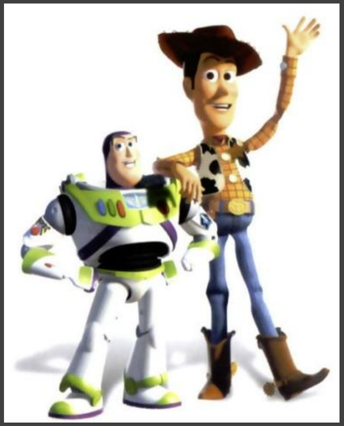
  (usually sampled regularly)

Computer Animation 101

Main Approaches to Motion

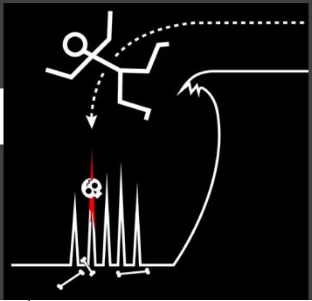
Manual Creation (Keyframe)



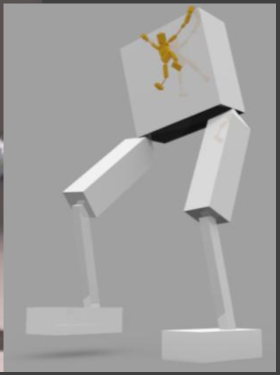


Algorithmic (Simulation)

f=ma



$$\rho\left(\frac{\partial \mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right)=-\nabla p+\mu\nabla^2\mathbf{v}+\mathbf{f}$$



Observation (Motion Capture)

Algorithmic and Simulation?

- Computed motion from a **model**

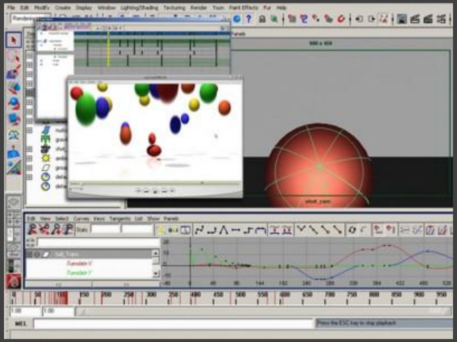
- Sources of models?
  - Physical principles
  - Carefully crafted routines (hacks)
  - …

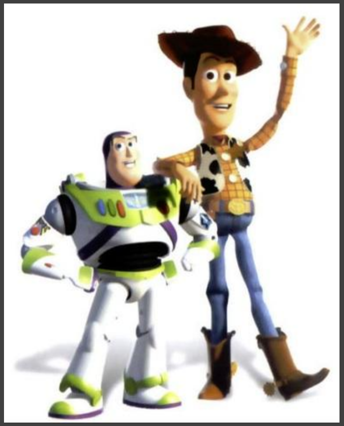
- Issues in quality and control

Computer Animation 101

Main Approaches to Motion

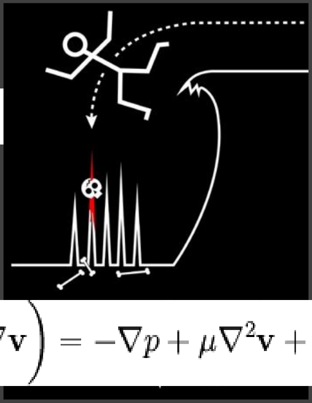
Manual Creation (Keyframe)



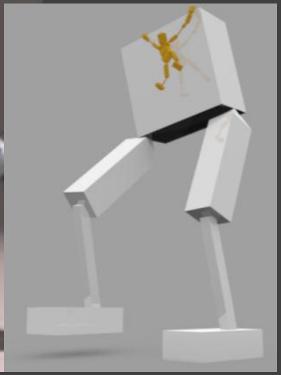


Algorithmic (Simulation)

f=ma



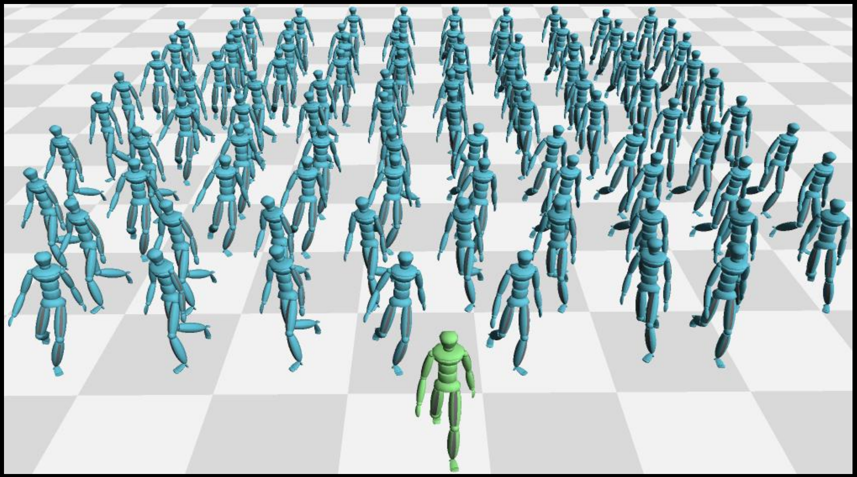
$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{f}$$

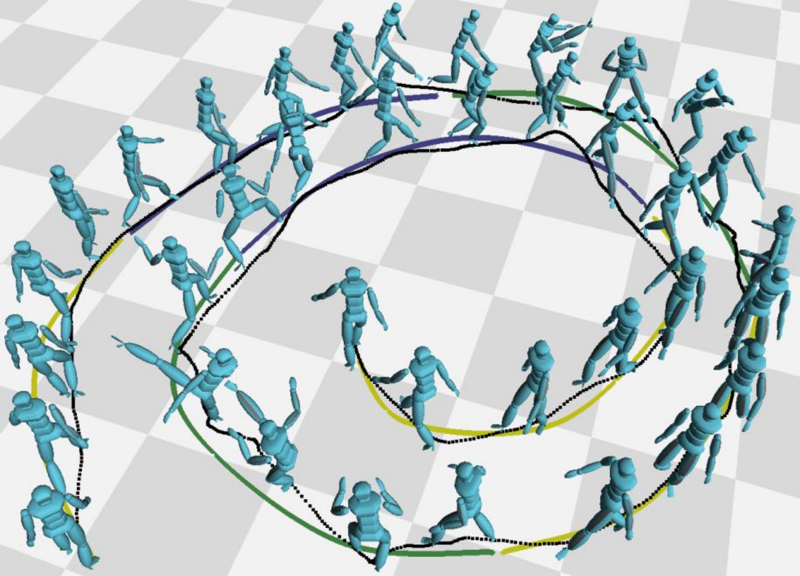


Observation (Motion Capture)

Synthesis By Example

Create what you need from what you have





Have: Lots of Clips

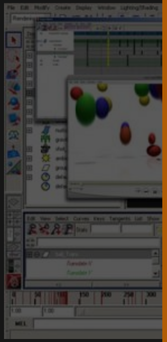
Want: Long Streams
Want: Controllable
Want: Precise/Continuous

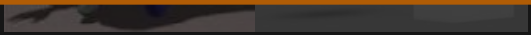
Computer Animation 101

Main Approaches to Motion

Manual Cr

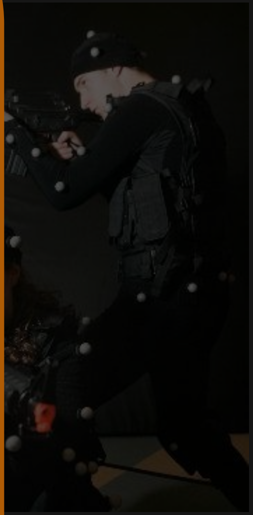
(Key





Observation

Capture)



Is **Synthesis-By-Example**

The Fourth Approach?

Mix of all three?

Algorithmic? (examples=model)

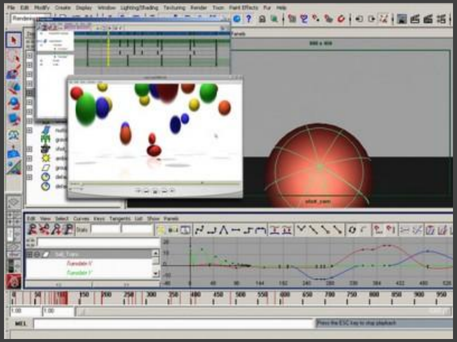
Most like Observation?

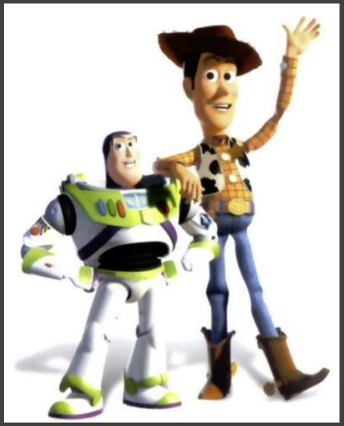
Distinct methods, pros and cons?

Computer Animation 101

Main Approaches to Motion

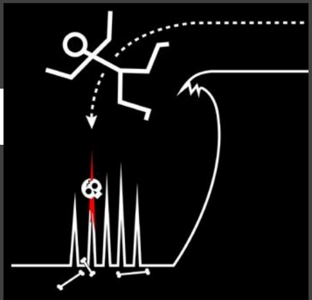
Manual Creation (Keyframe)



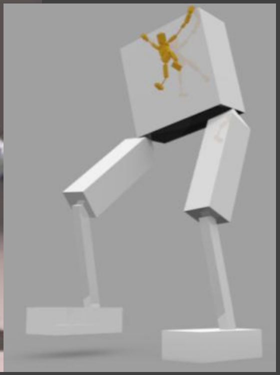


Algorithmic (Simulation)

f=ma



$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{f}$$

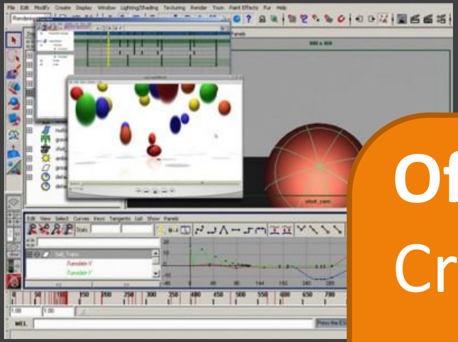


Observation (Motion Capture)

Computer Animation 101

Main Approaches to Motion

Manual Creation (Keyframe)



Offline Generation:

Creates pre-recorded motions

High-Quality, Controllable, …

But not interactive

Observation (Motion Capture)

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

Lecture 1

- Some Preliminaries
  - Defining the problem
- Basics of Synthesis-By-Example
  - Defining the building blocks
- Research vs. Practice

What do we want?

- “Good” Motion

- Interactive Characters
  1. Authoring of whole scenes
  2. Generation of longer movements
  3. Interactive control

Interactive characters are getting pretty good

(and this was 2008!)

Games characters are getting pretty good

Games characters are getting pretty good

But Wait!
That’s a Cut
Scene!

Cut scenes show we’re ready for better motion in actual game play

(interactive characters are more interesting)

- Cut scenes could be better…
- Display / presentation is there
- Need characters that can “act” in story

- Interactive character should be as good
- But they’re much harder

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

What do we want?

- “Good” Motion

- Interactive Characters
  1. Authoring of whole scenes
  2. Generation of longer movements
  3. Interactive control

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 (in the examples)

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!

What do we want

More motion!

- Do more things
- In more different ways
- Consistency

Expressive
(express what the director wants)

More controllable:

- Work within the system
- More responsive to player
- Work with “AI”
- Fit the situation

Interactive
(work in the game to provide mechanics)

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

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

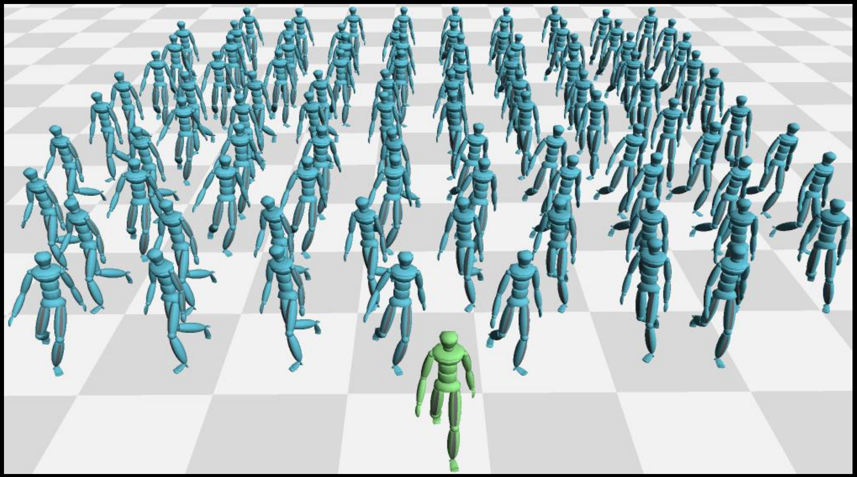
  Without having to model each one
- Easier to scale to diverse repertoire, with acting subtleties, get the directors intent, …

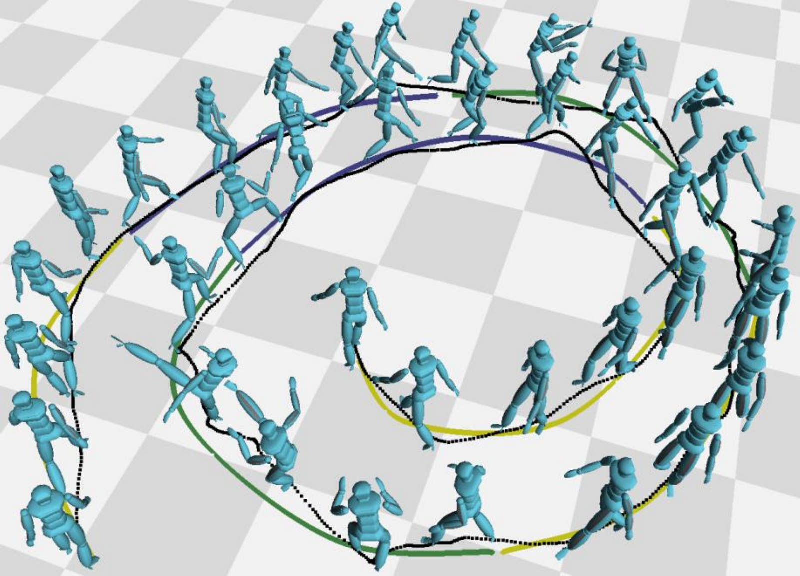
Disadvantage of SBE: No Model!

- Rely on examples
  - Which may not apply to other situations
- Limited adaptability
  - Simple methods work when “close” to examples
- Larger repertoire (usually) means larger library

Synthesis By Example

Create what you need from what you have

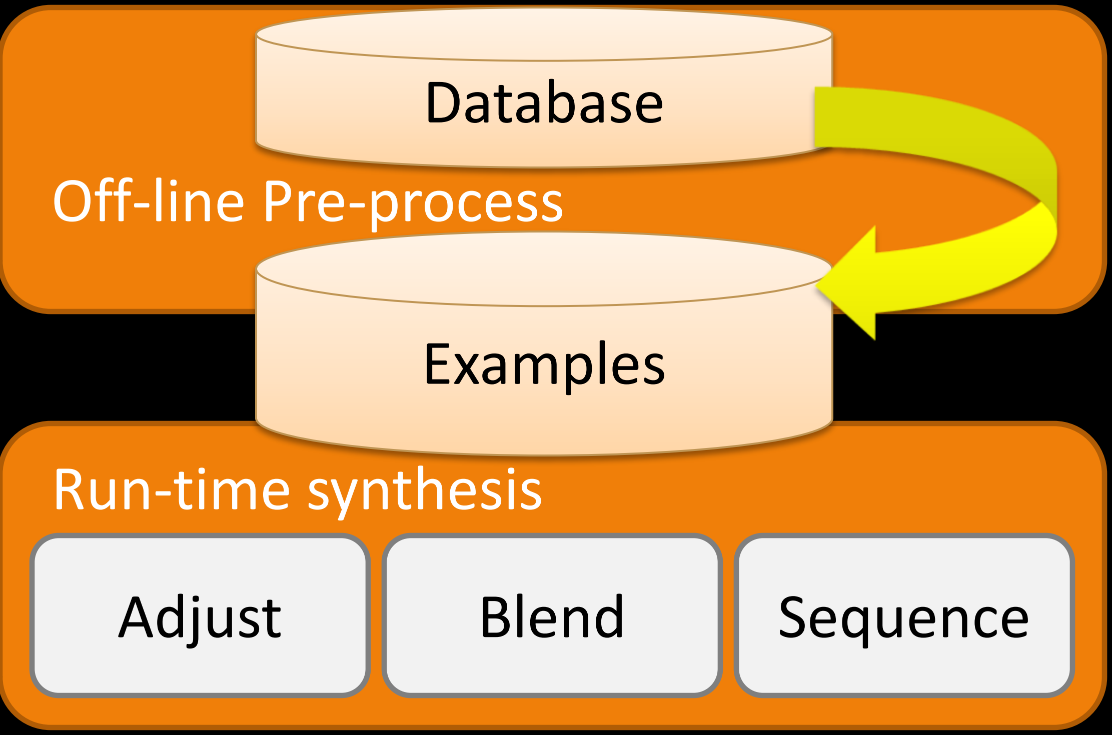




Have: Lots of Clips

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

Basic Ideas of Synthesis-By-Example



Database

Off-line Pre-process

Examples

Run-time synthesis

Adjust

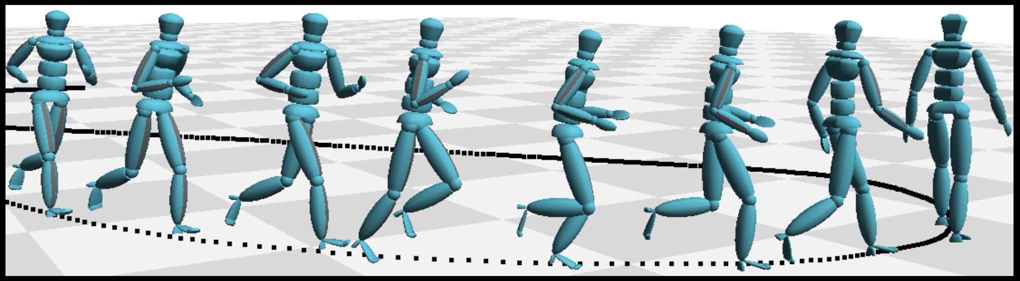
Blend

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)

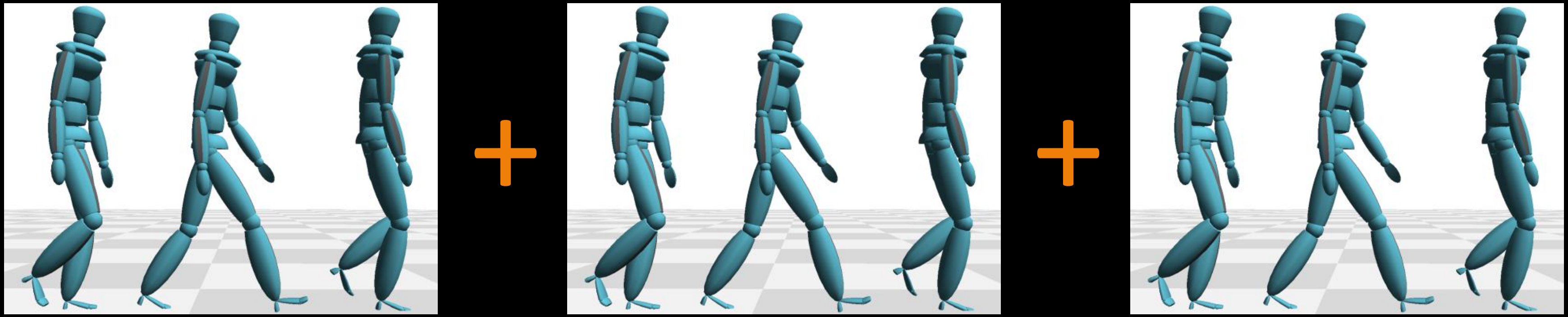
	Practice (real games)	Research
Preparation:	Planning	Automation
	Careful preparation	Automation
	Manual adjustment	Automation
Assembly:	Basic methods	Basic methods
	Tweaks thrown in	Tweaks thrown in
Control:	Carefully crafted&tuned	Search
	Planning simplifies	Pre-Processing

An Example:

Combining motions can be hard
(or easy – in the right cases)

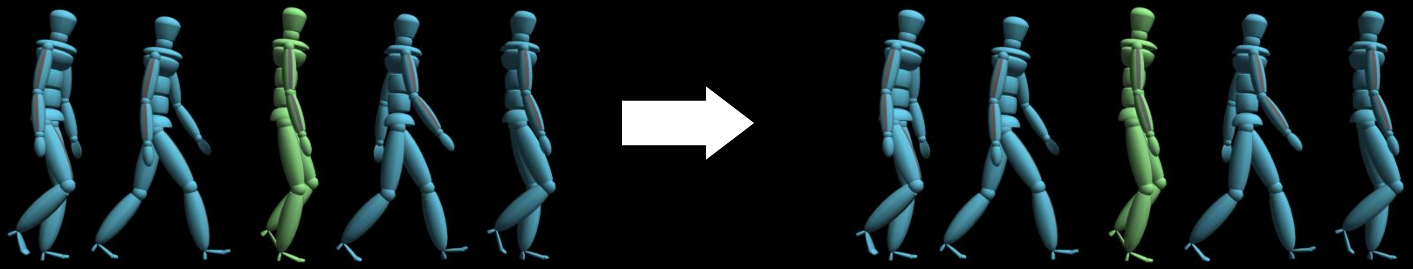
Concatenation

Put clip after clip after clip …

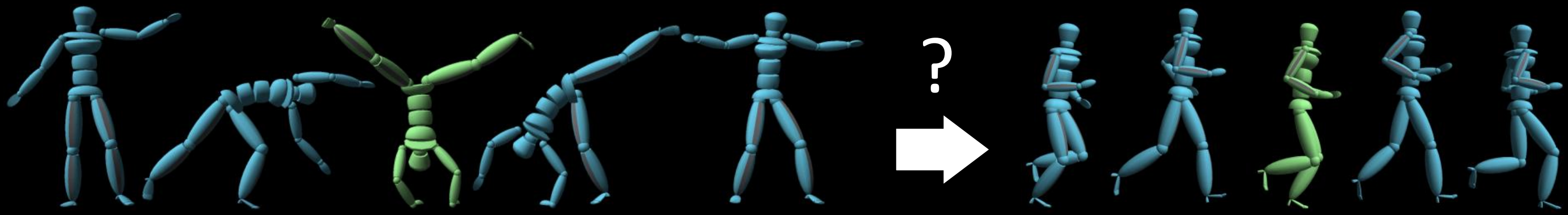


Transitions

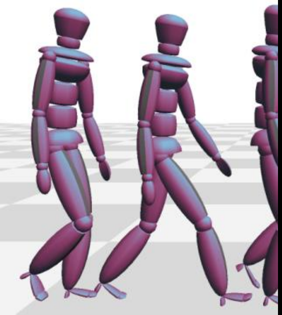
Some transitions are easy



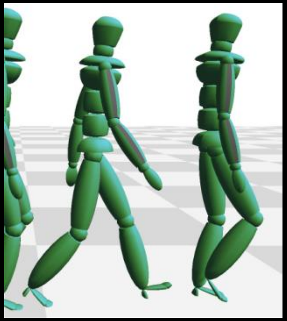
Some transitions are hard

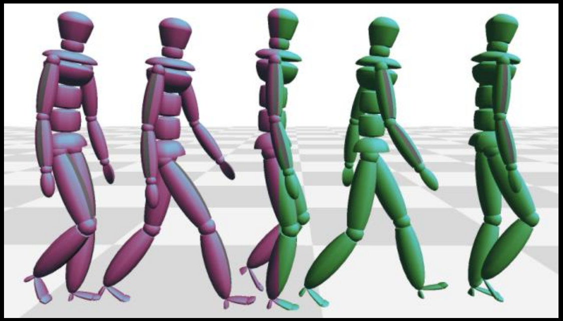


Simple Transition Methods

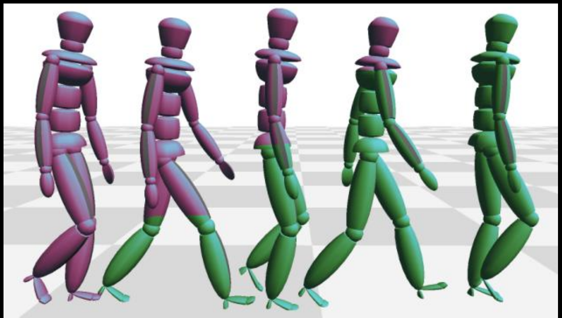


+





Cut transition



Blend Transition

Better methods?

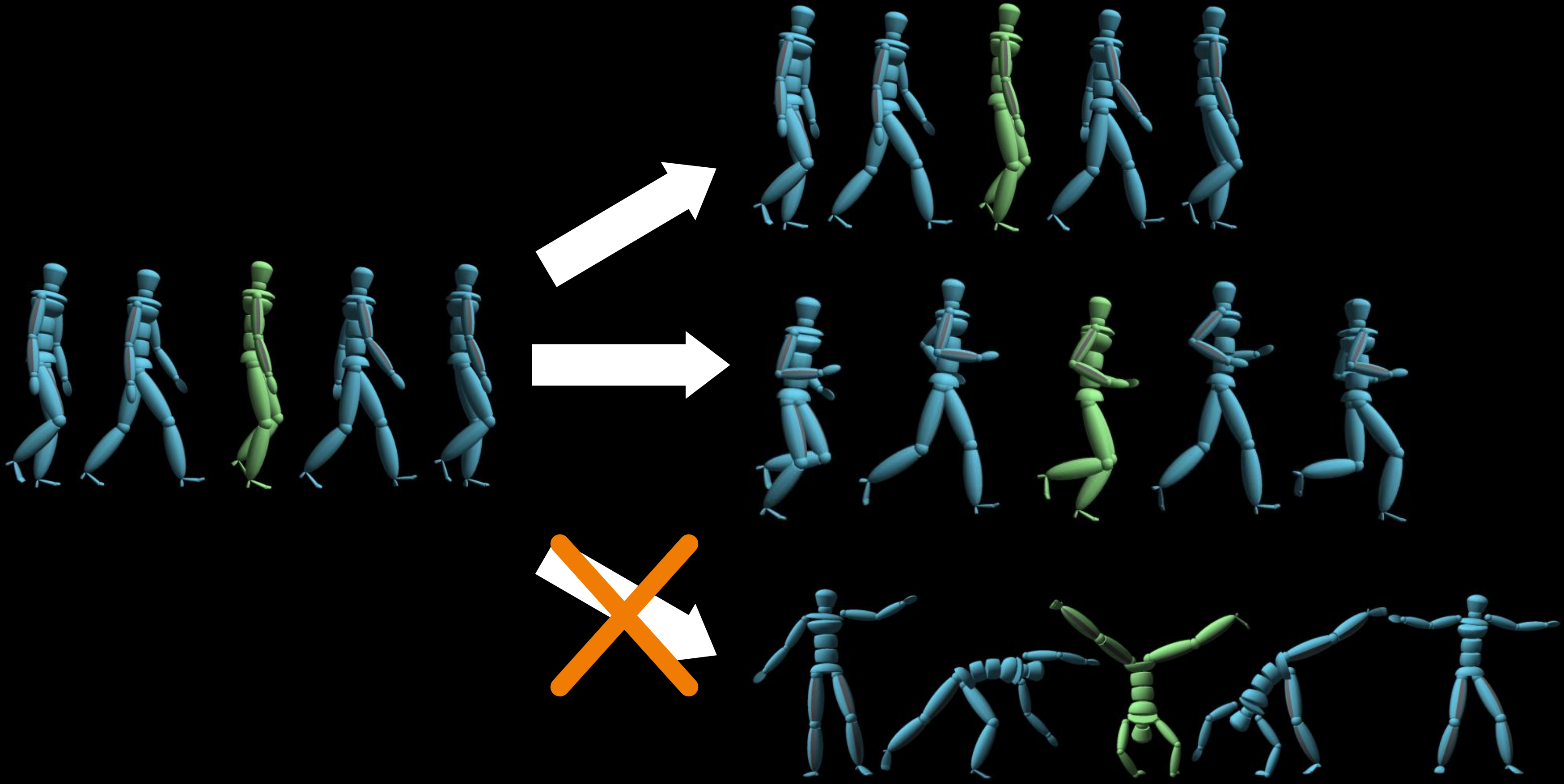
- A different approach:
  - Seek to use simple components

- Good points of simple
  - Runtime is known, efficient, …
  - Staying close to the motion preserves quality
  - General (few assumptions about motion)

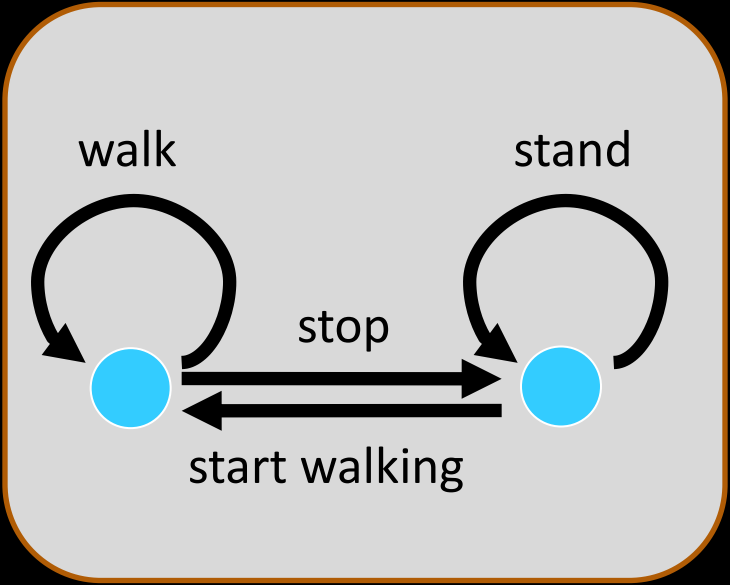
- Better building blocks are less studied

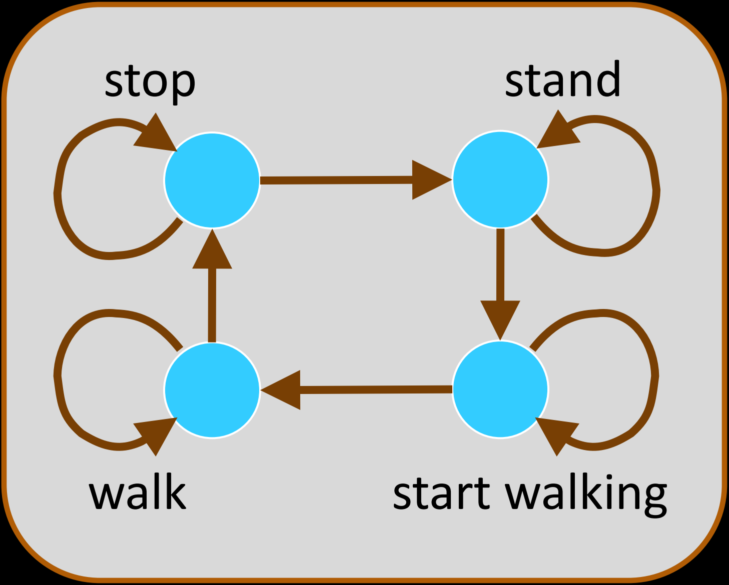
Motion Graphs
(aka Move Trees)

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

SBE in Practice vs. Research

(practice has been doing it longer)

Practice (real games)

Preparation:

Planning
Careful preparation
Manual adjustment

Carefully plan and create examples so they fit together

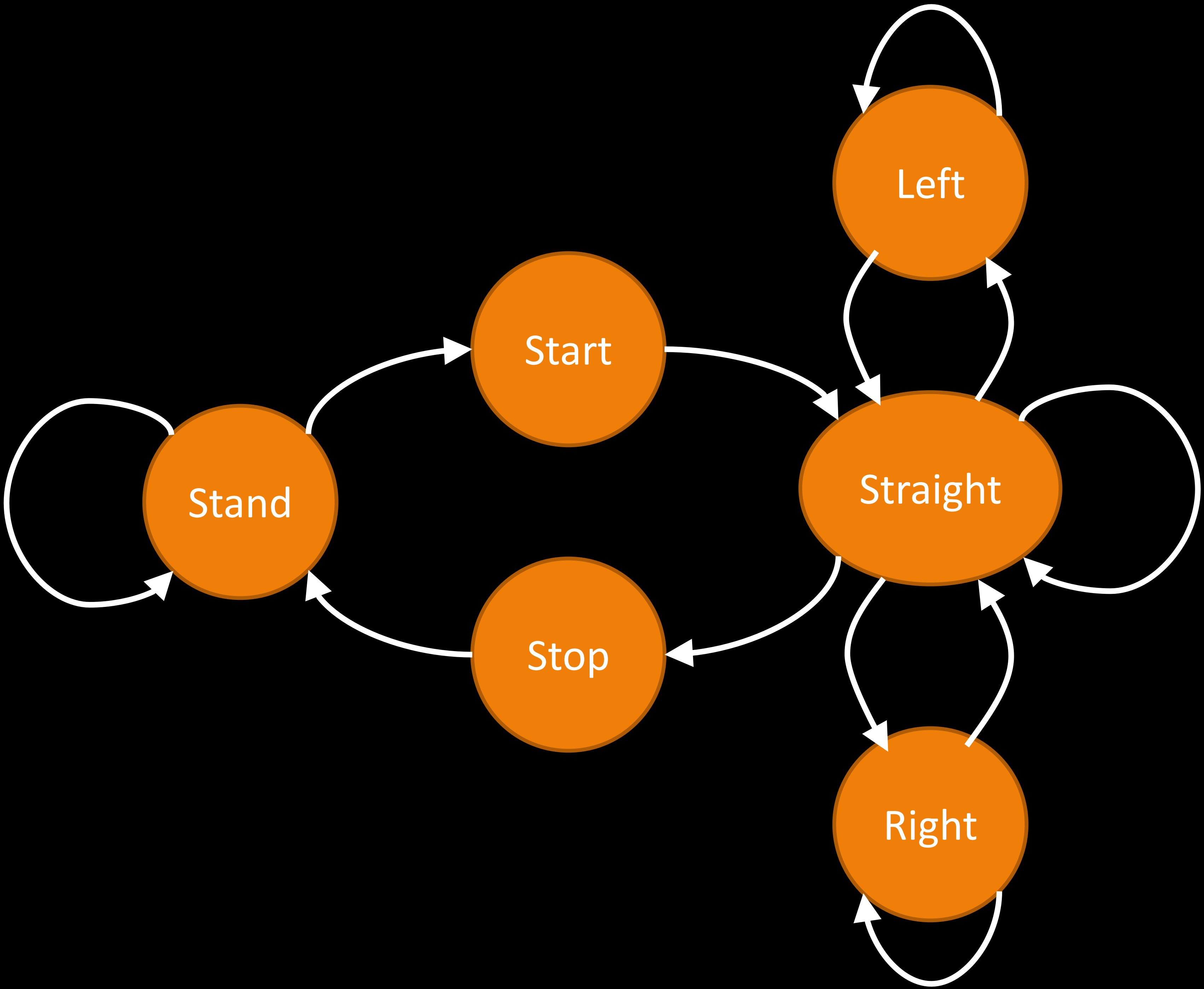
Assembly:

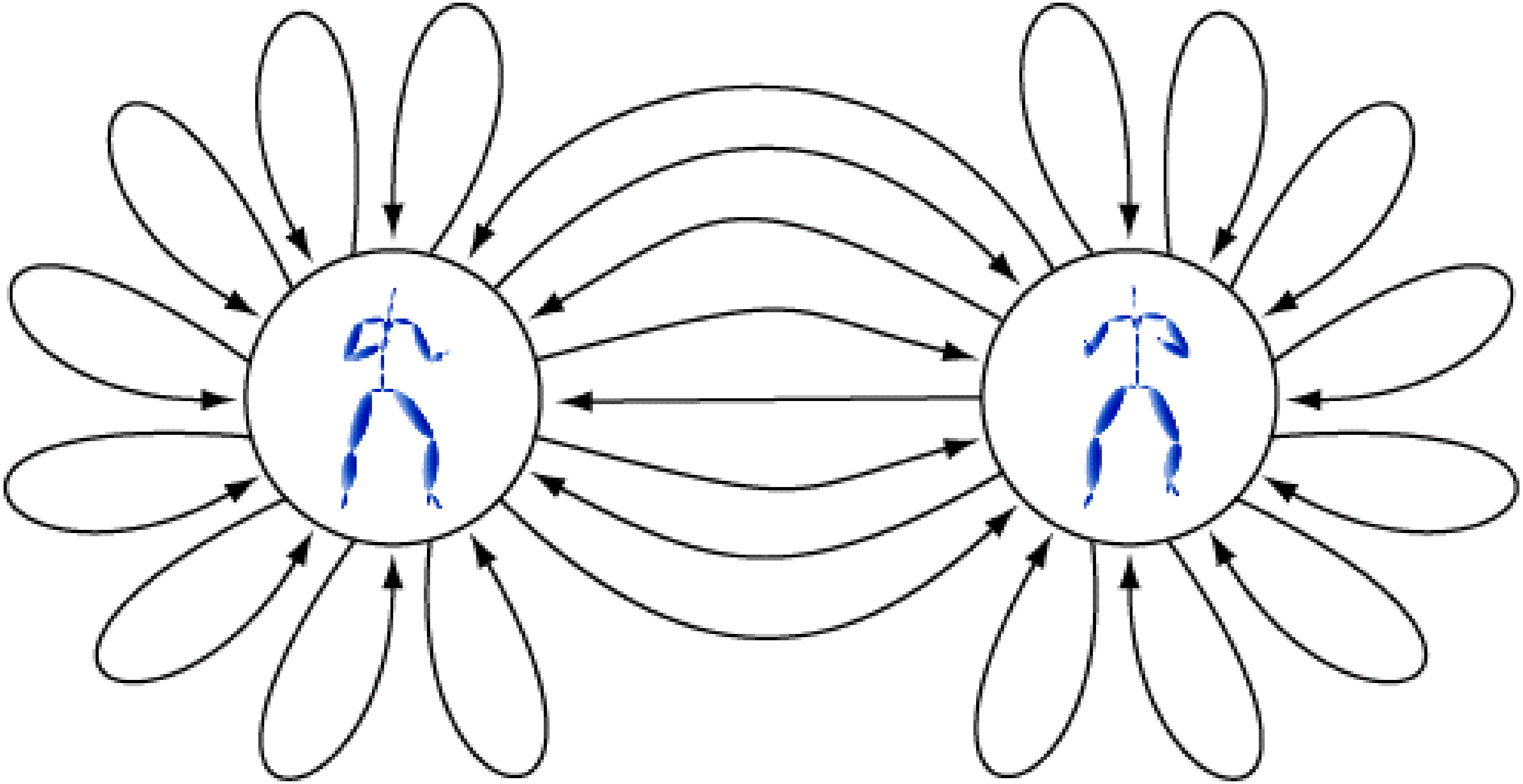
Basic methods
Tweaks thrown in

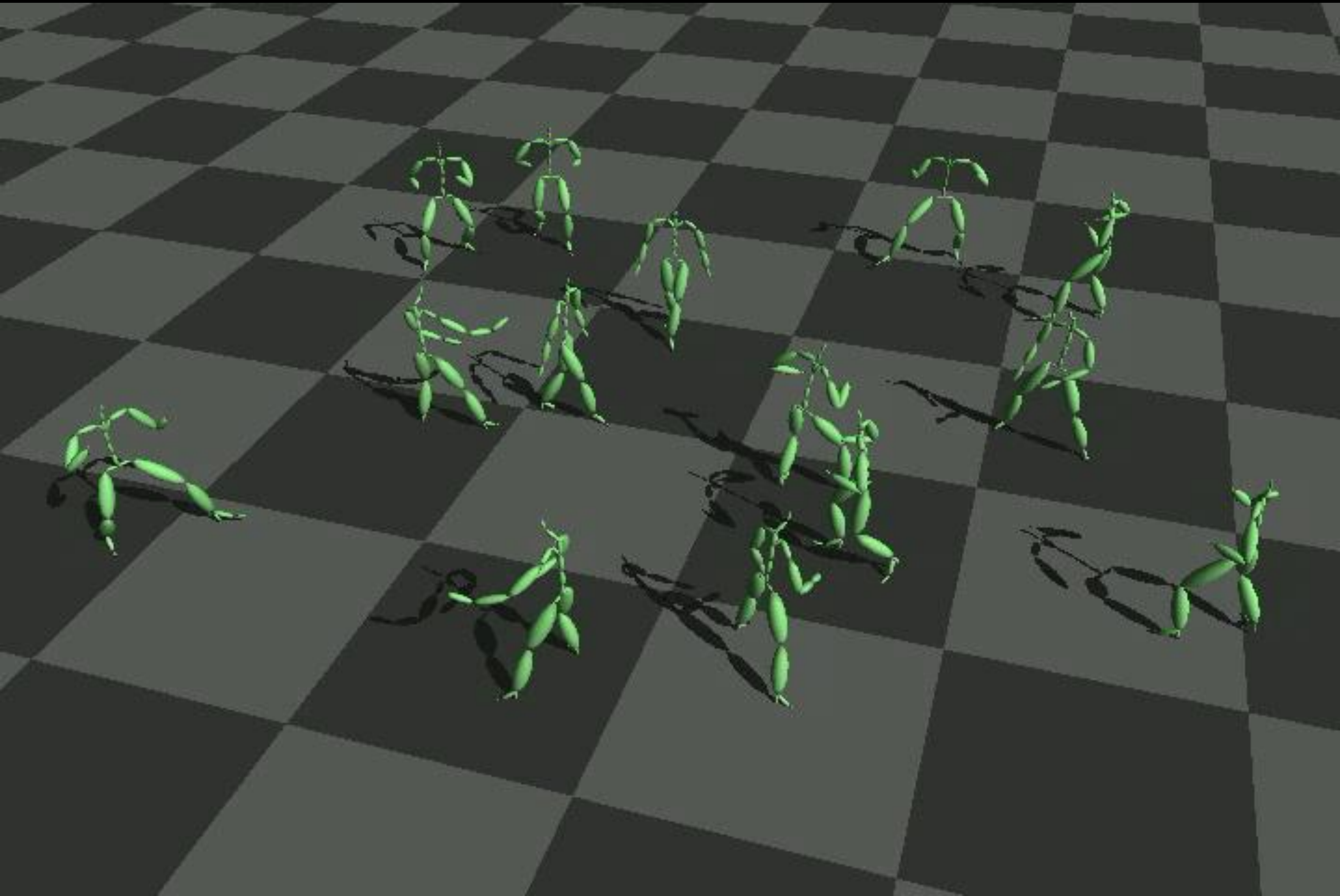
Control:

Carefully crafted&tuned
Planning simplifies

Simple graph
Choices easy


Left
Start
Straight
Stand
Stop
Right





In Practice…

- Carefully create motions so they fit
  - Check each and every motion and transition
- Build simple, planned graph structures
  - Easy choices based on control
- Generally little planning
  - Happens independently of animation

Coming attraction, Lecture 2:

Motion Graphs in Research

- Can we avoid planning? Use “found” motion?
- Can we automate things?
  - Automatic graph construction
  - Automatic graph usage
- Can we solve more complicated problems?