Motion Synthesis By Example

A Tutorial in 3 and 3/2 parts

Michael Gleicher

Dept of Computer Sciences
University of Wisconsin - Madison

Motion Synthesis By Example

Lecture 2: Motion Graphs

Michael Gleicher

Dept of Computer Sciences
University of Wisconsin - Madison

Review

- Synthesis by Example
  - Get good examples
  - Put them together in simple ways
- Concatenation in practice
  - Motion graph
  - Well-planned clips
  - Contrived graph structures

Example-Based Synthesis

Capture the detail, subtlety and complexity

Good News:

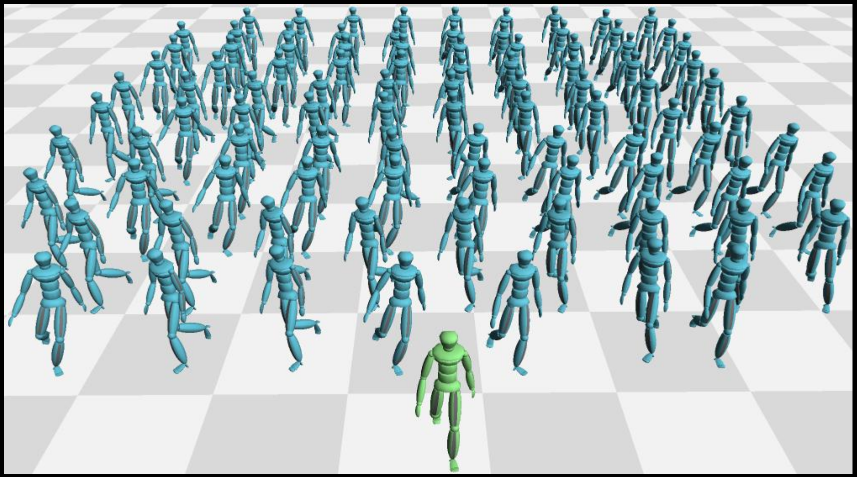
We don’t need to model all the complex things!

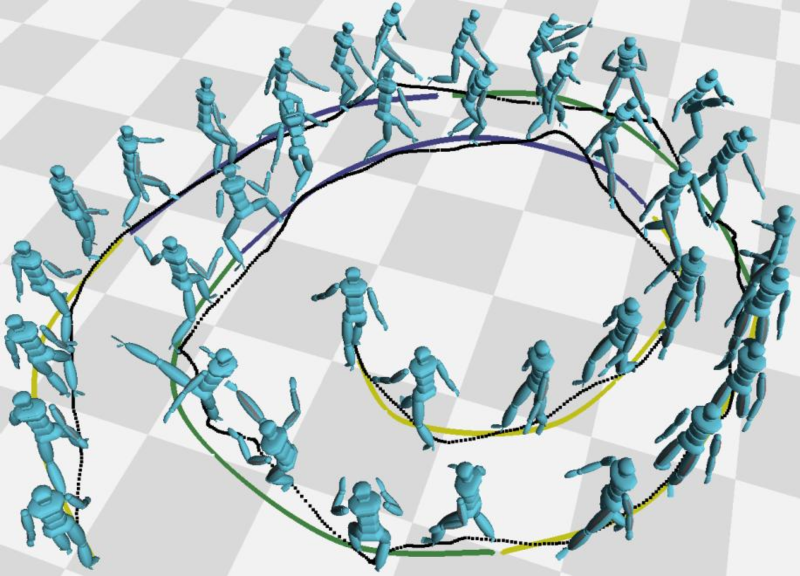
Bad News:

We don’t have a model to generate what we didn’t capture!

Synthesis By Example

Create what you need from what you have





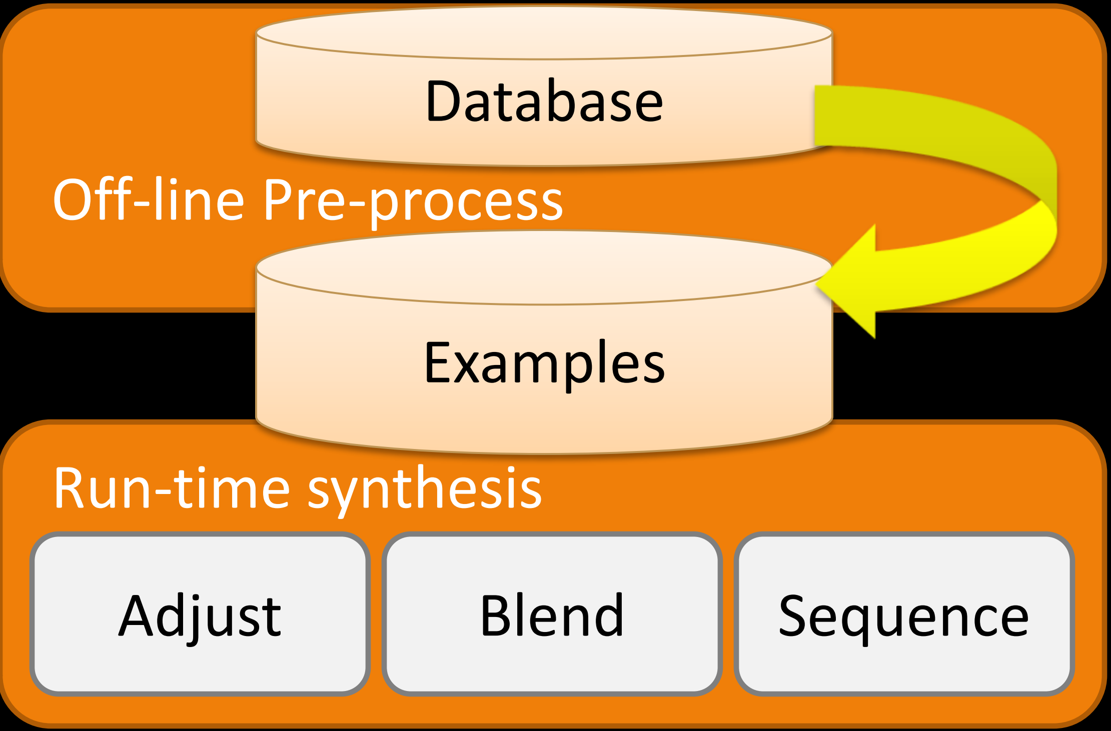
Have: Lots of Clips

Want: Long Streams

Want: Controllable

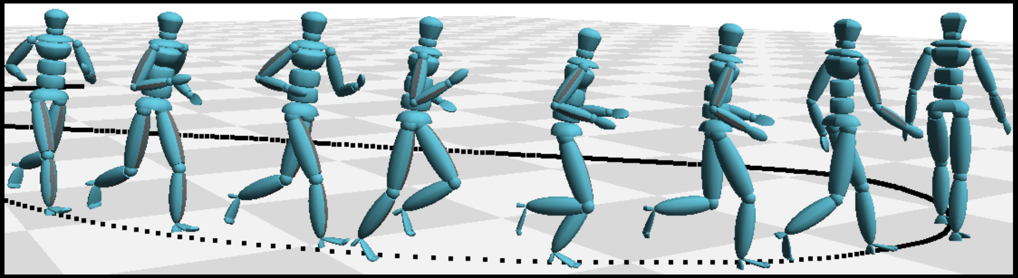
Want: Precise/Continuous

Basic Ideas of Synthesis-By-Example



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)

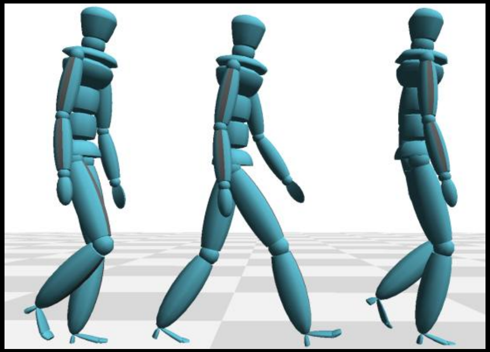
	Practice (real games)	Research
Preparation:	Planning Careful preparation Manual adjustment	Automation Automation Automation
Assembly:	Basic methods Tweaks thrown in	Basic methods Tweaks thrown in
Control:	Carefully crafted&tuned Planning simplifies	Search Pre-Processing

Lecture 2

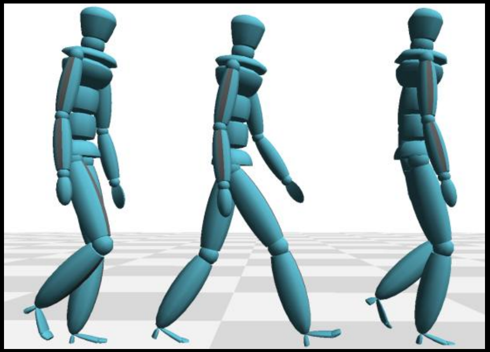
- Automatic graph construction
  - Reduce planning! Use found motions!
- Using a motion graph
  - Search to create motions
  - Interactive control
- Other approaches to interactive control

Concatenation

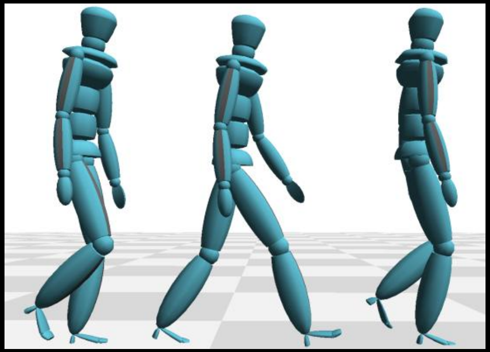
Put clip after clip after clip …



+

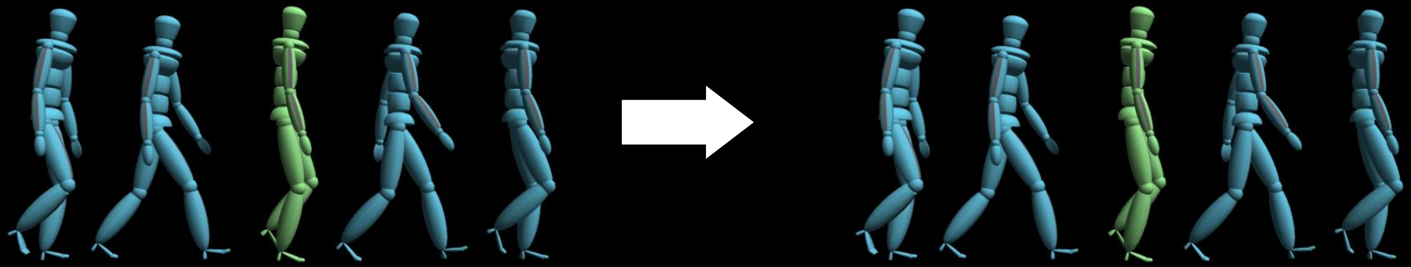


+

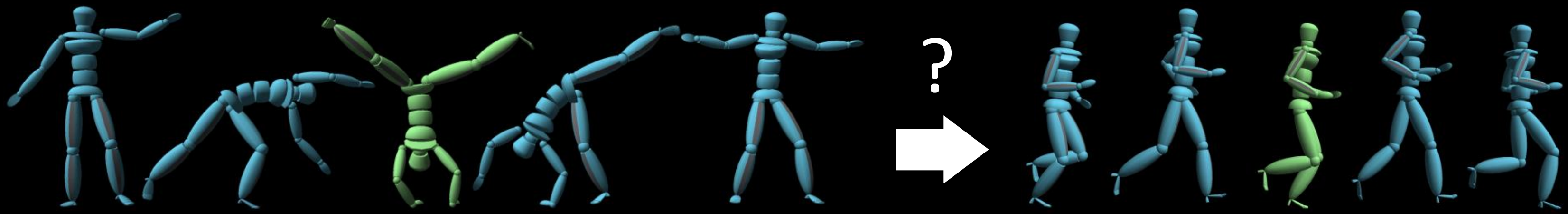


Transitions

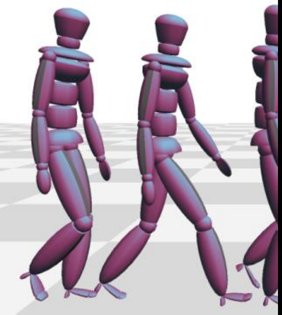
Some transitions are easy



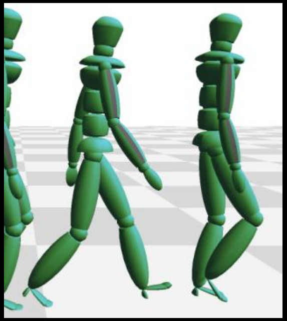
Some transitions are hard

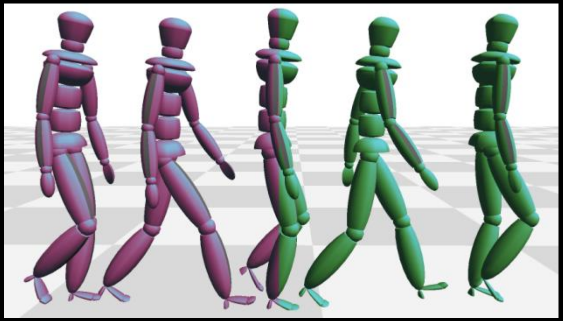


Simple Transition Methods

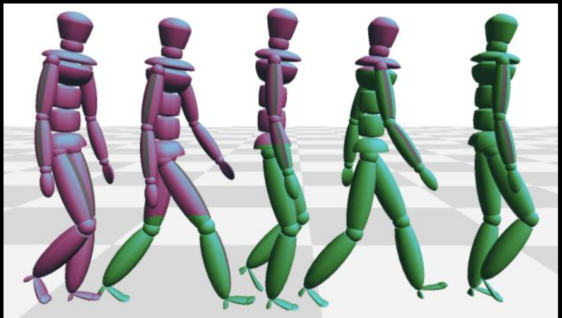


+





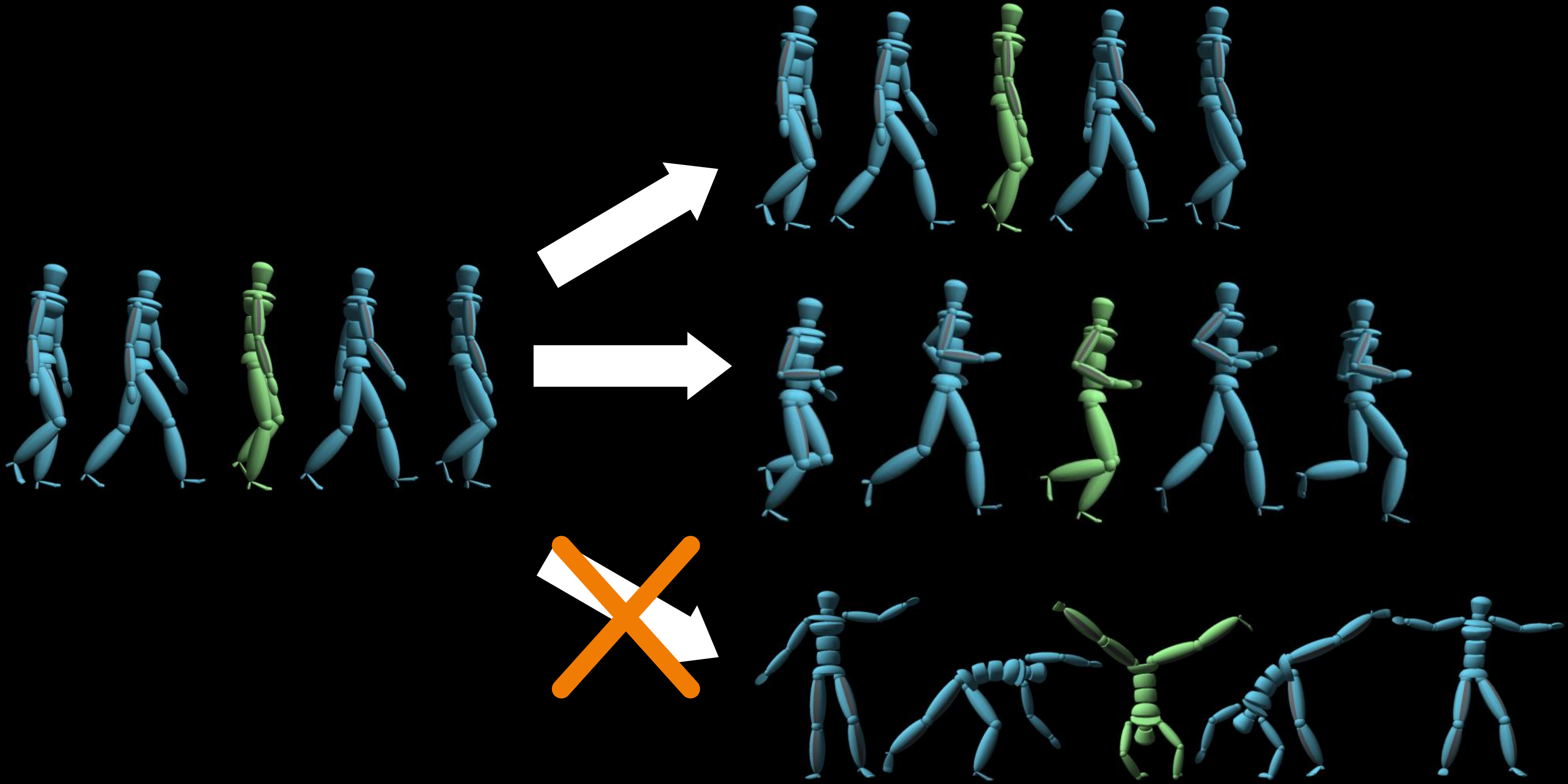
Cut transition



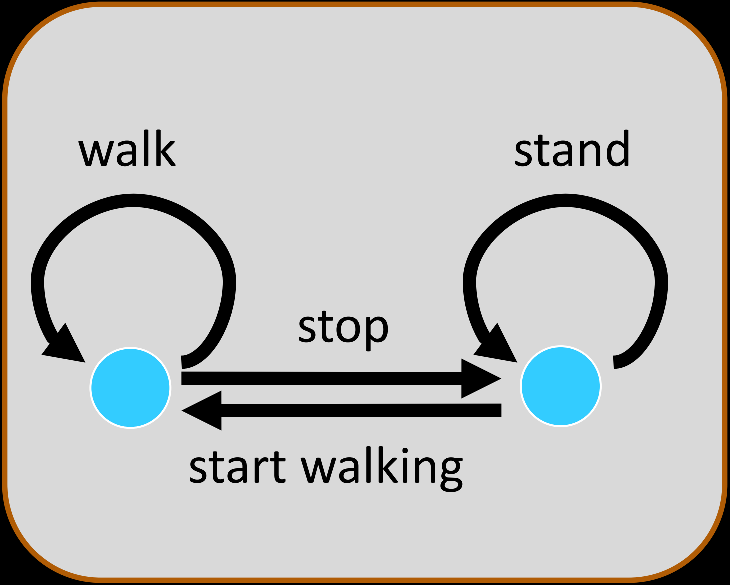
Blend Transition

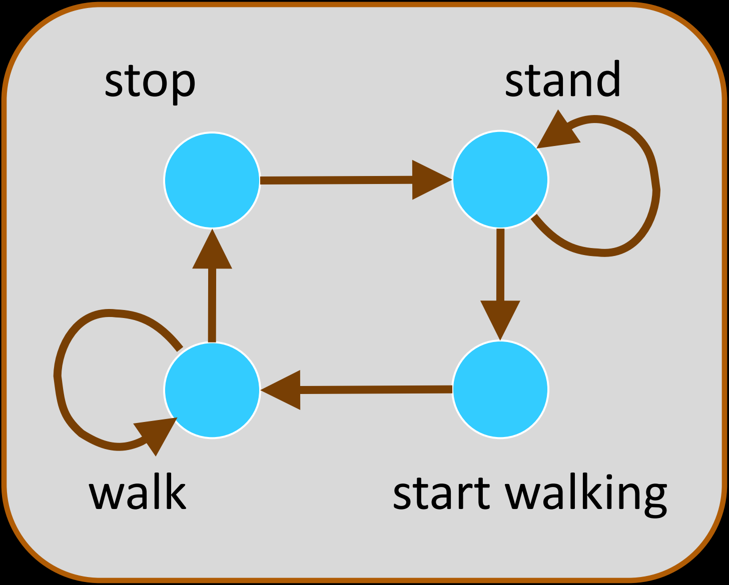
Motion Graphs (aka Move Trees)

Some transitions are easy – remember which



Graph Notation





Edge = clip

Node = choice point

Graph walk = motion

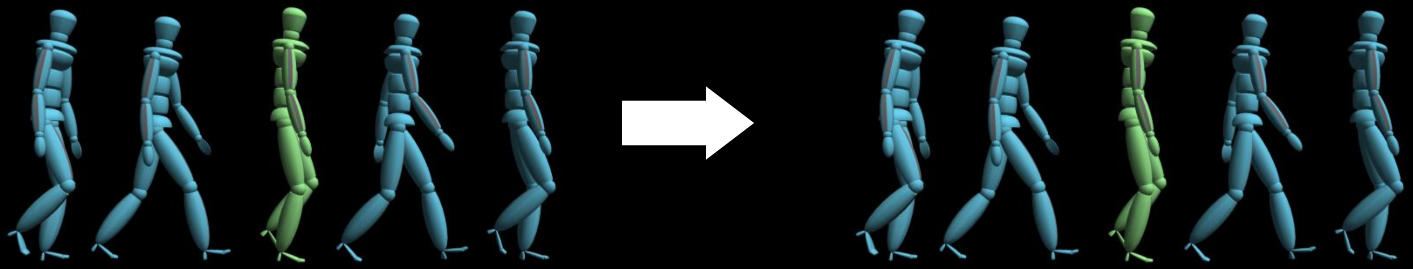
Edge = valid transition

Node = clip

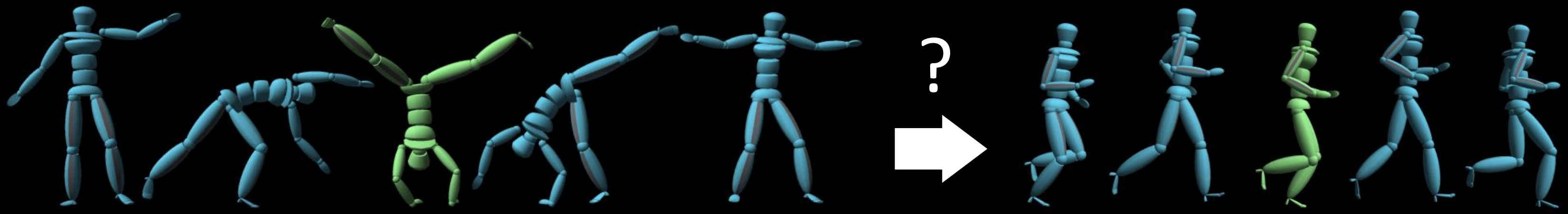
Graph walk = motion

Transitions

Some transitions are easy

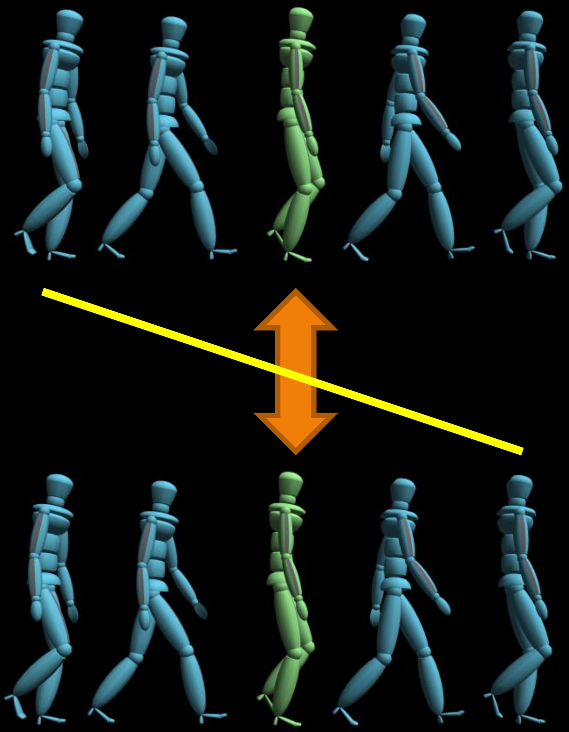


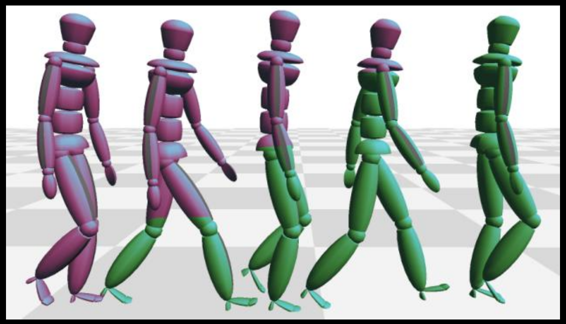
Some transitions are hard



When do transitions work?

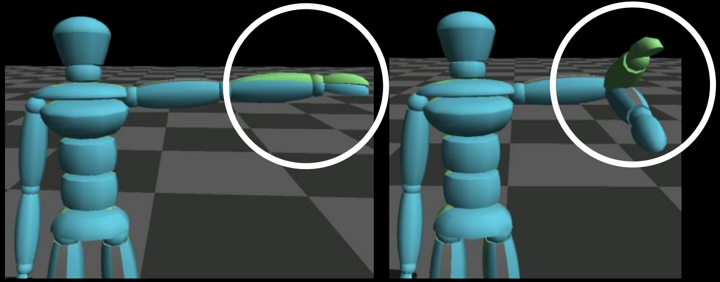
- Blend to avoid bad artifacts



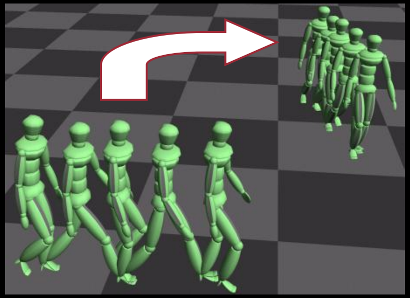


Determining potential transitions

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



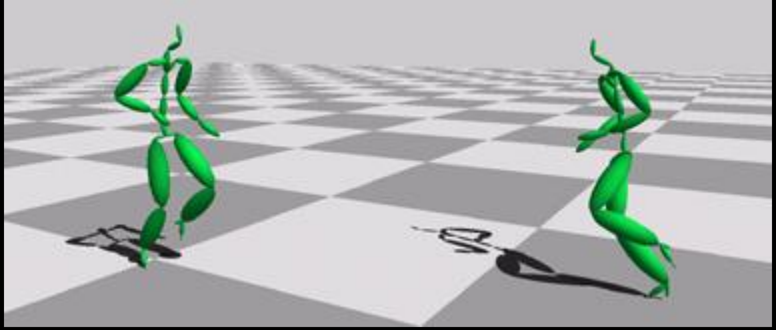
- Need coordinate invariance
  - Different camera ≠ different motion!



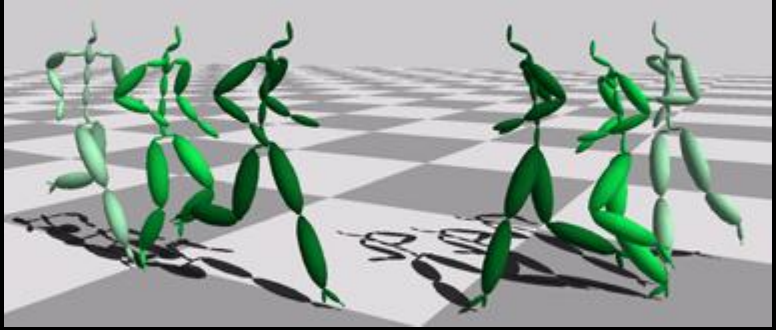
What is Similar?

Factor out invariances and measure

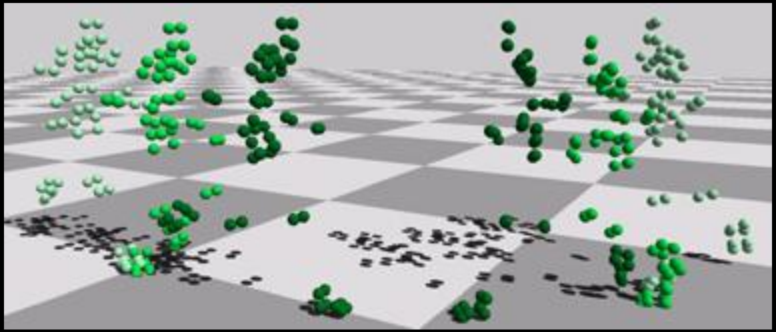
1) Initial frames



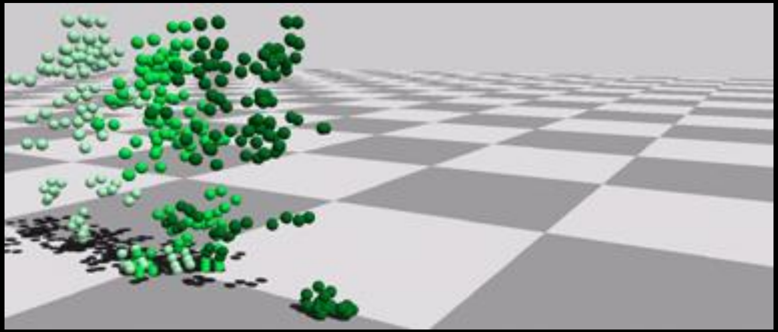
2) Extract windows



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

Do you need this?

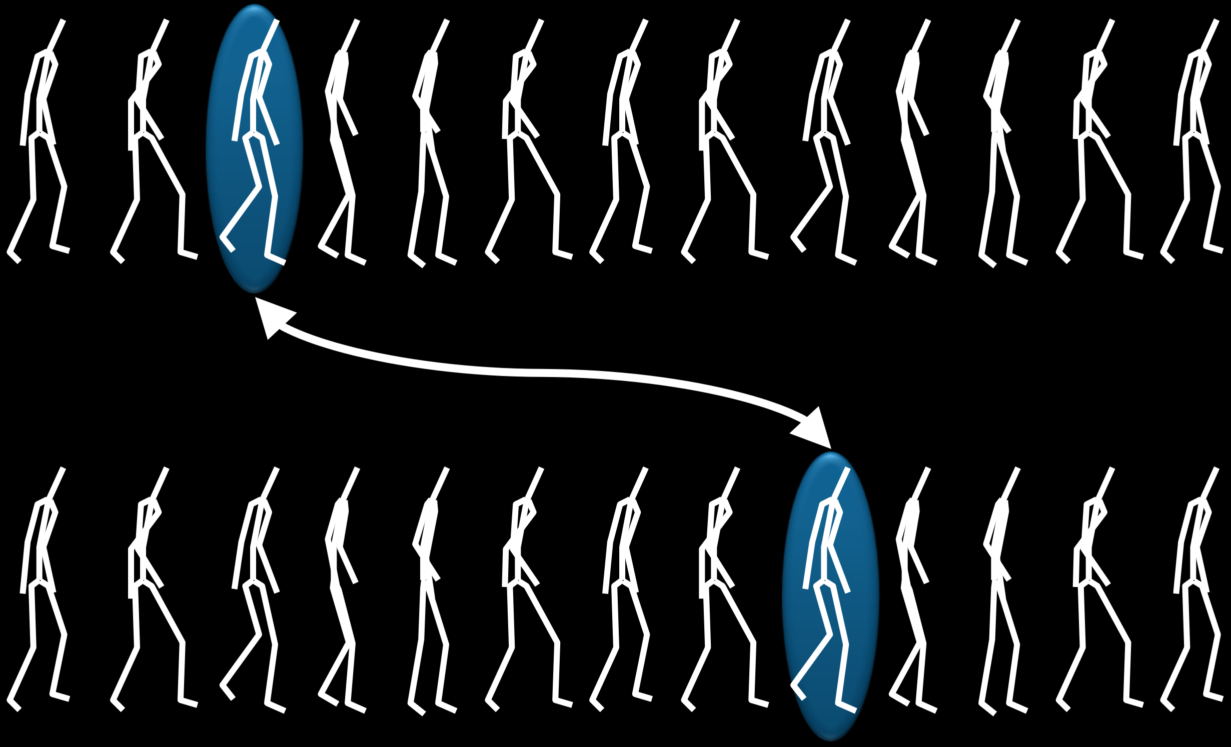
- Many similarity metrics

- Everyone has an opinion
- Some methodical comparison

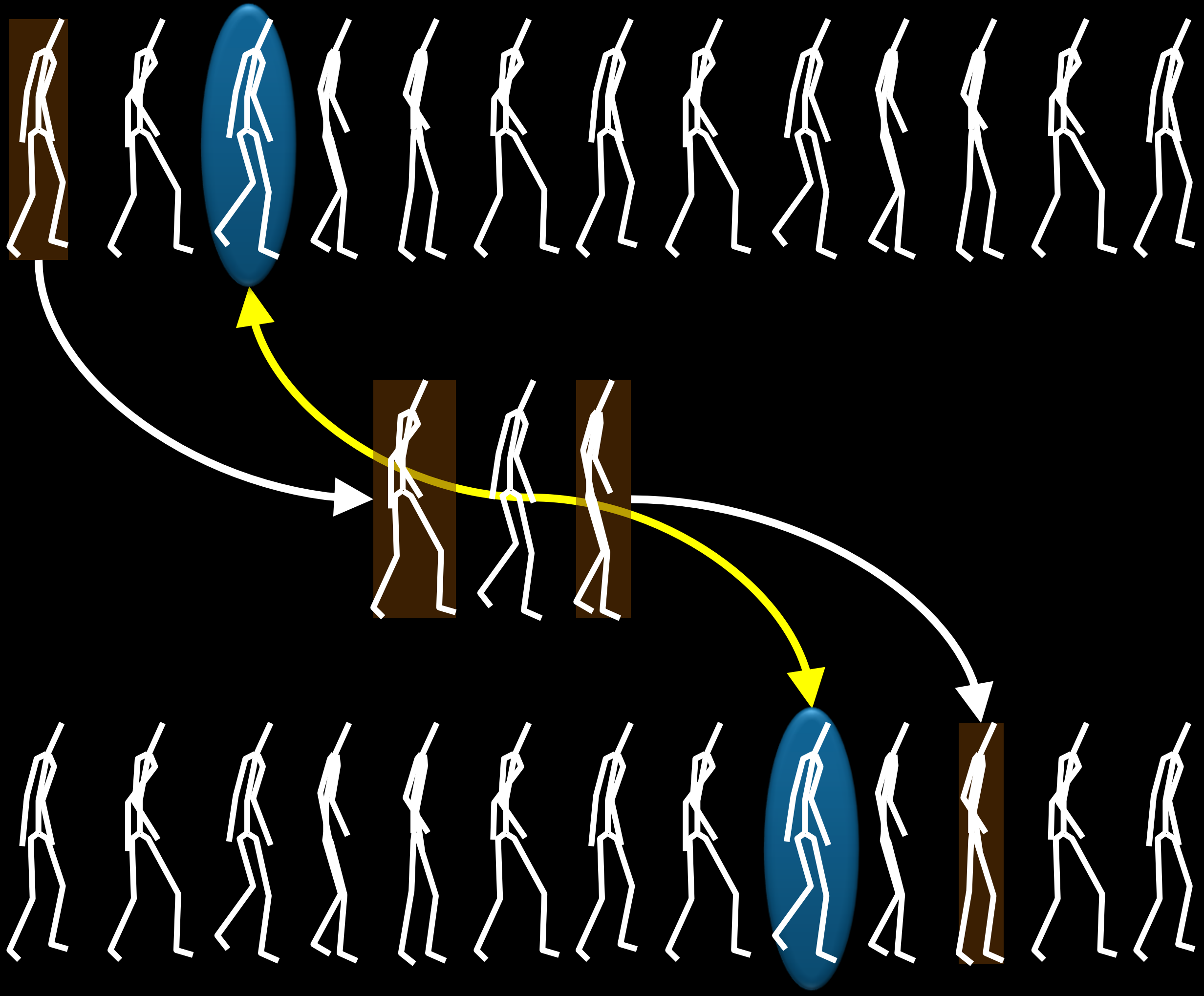
- Complex methods might not be worth it

Building a Motion Graph

- Find Matching States in Motions

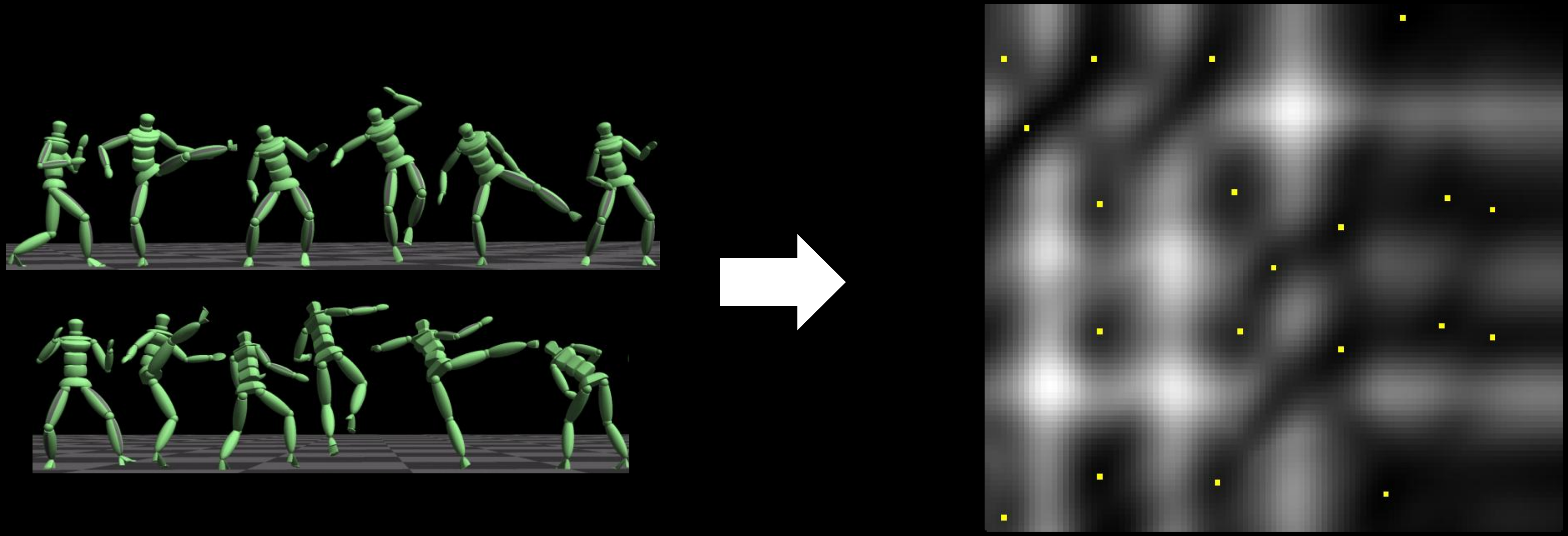


Building a Motion Graph



Finding Transition Points

Every pair of frames now has a distance.



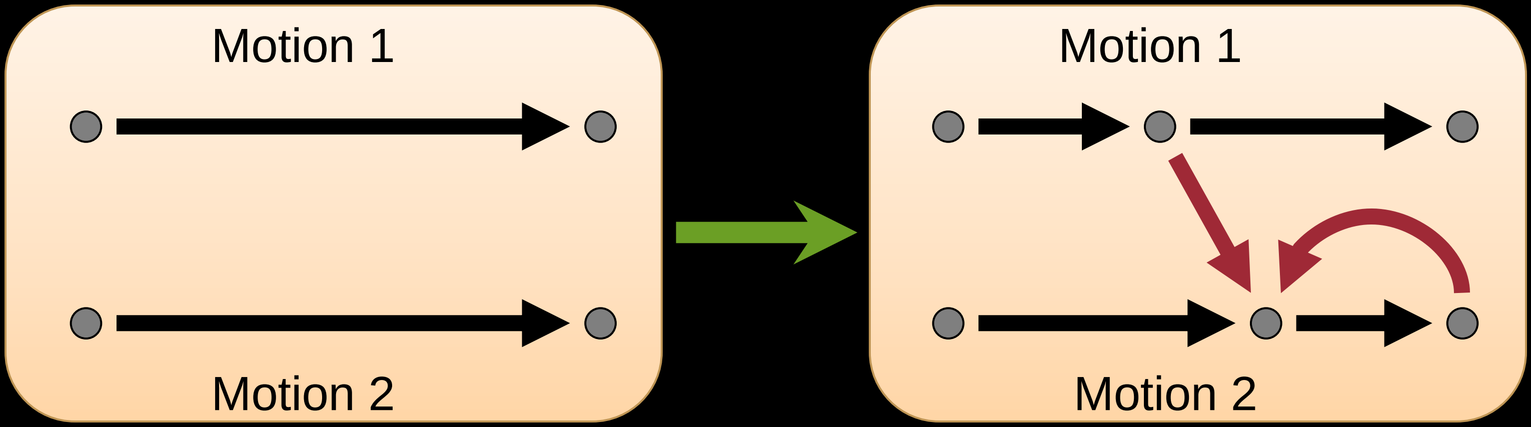
Transitions are local minima below a threshold.

Motion Graphs

Kovar et al, Arikan&Forsyth, Lee et al. – All SIGGRAPH 02
and many other variants since

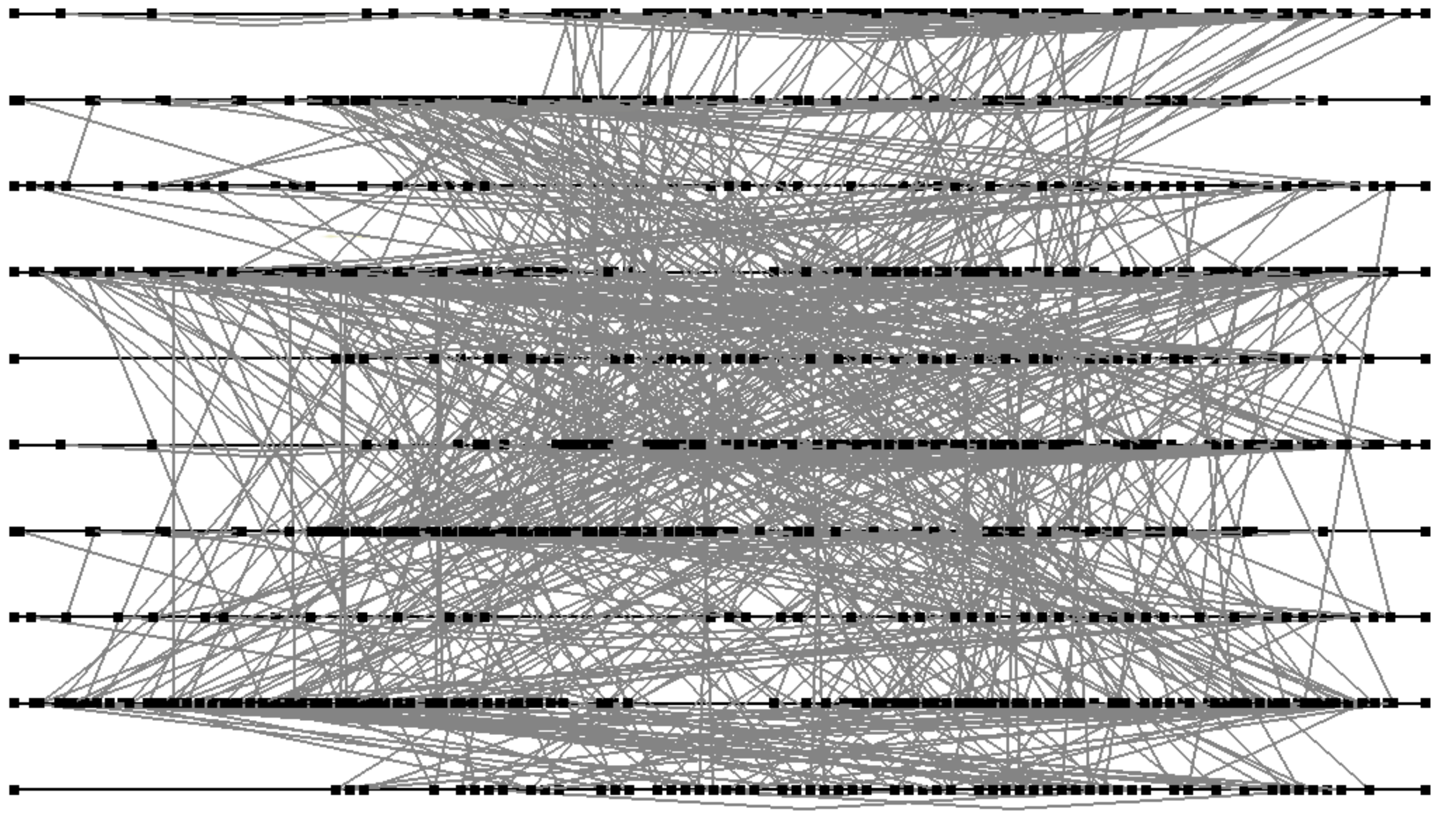
Start with a database of motions

Goal: add transitions at opportune points.

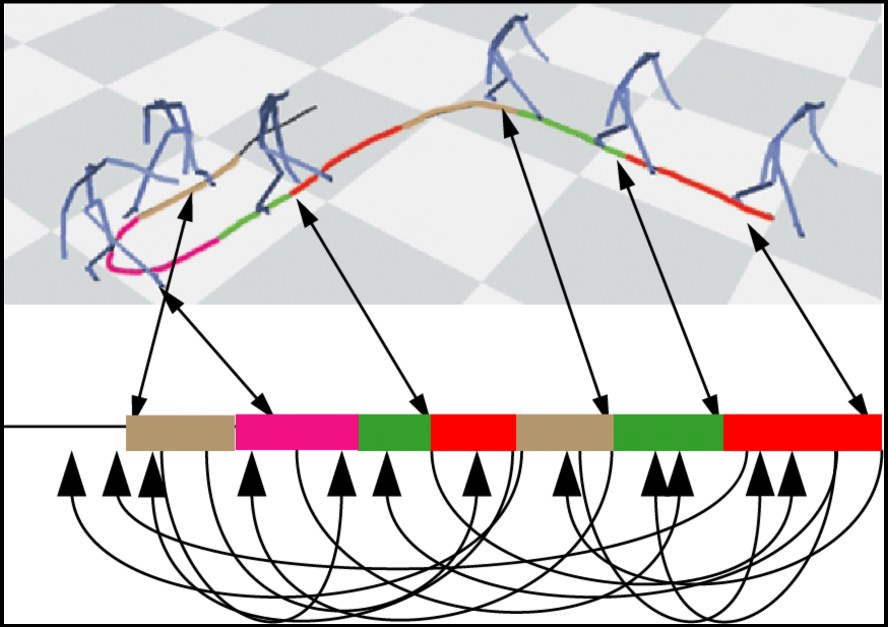


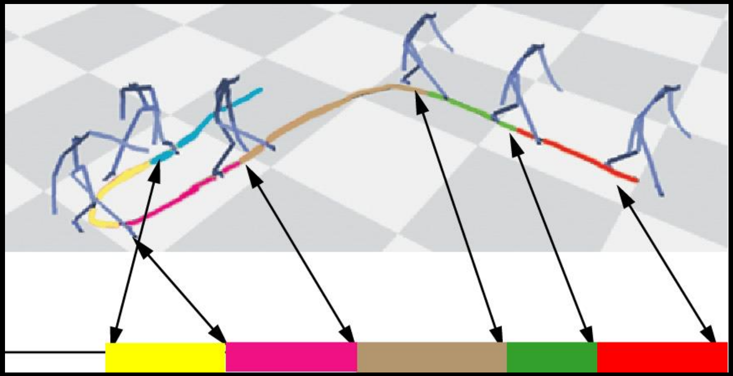
Structure of Motion Graphs

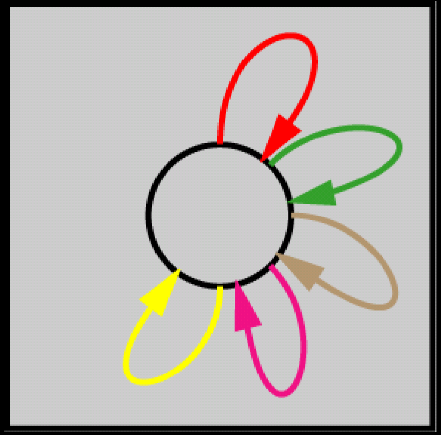
Opportunistically built graphs can be hard to search – especially for quick control



Structured vs. Unstructured Graphs







Gleicher et al. I3D 2003

What can you do with a graph?

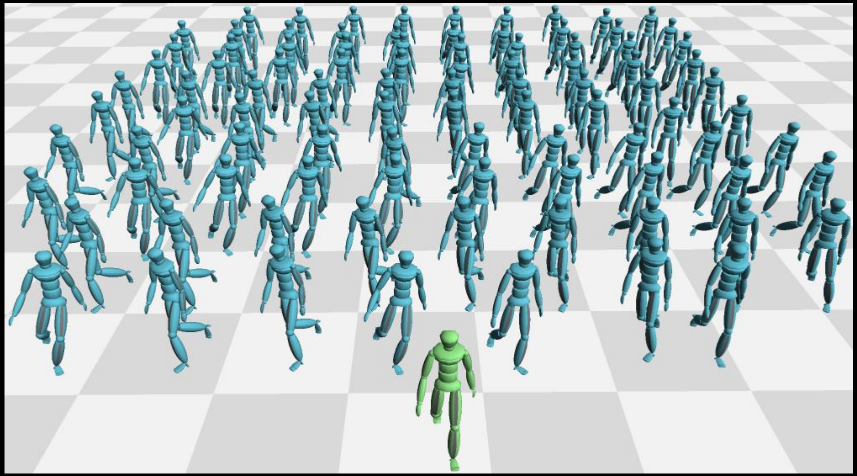
- Any walk on the graph is a valid motion

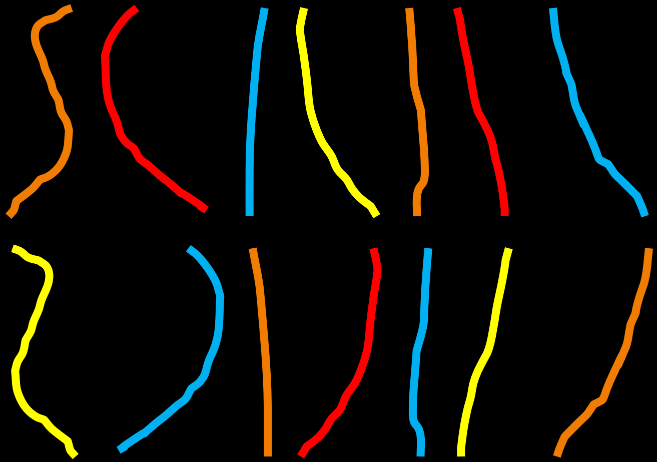
- Random Walks
- Search for Walks that meet constraints
- Make decisions in response to controls

Create Graph
Read Graph
Write Graph
Show Structure
Center Skeleton
Track Skeleton
Random Walk
30
Walk Speed
Start Play

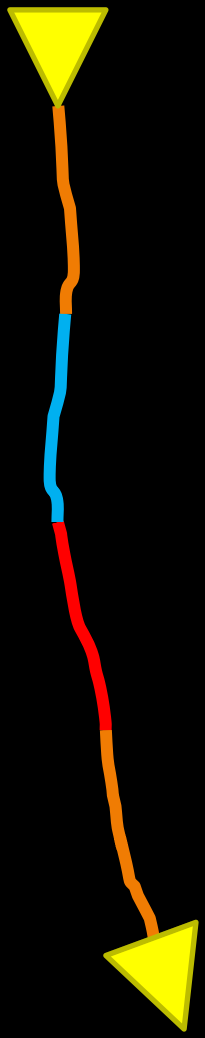
Search to goal

- Search for a walk on the graph (sequence of clips) that meets the goals



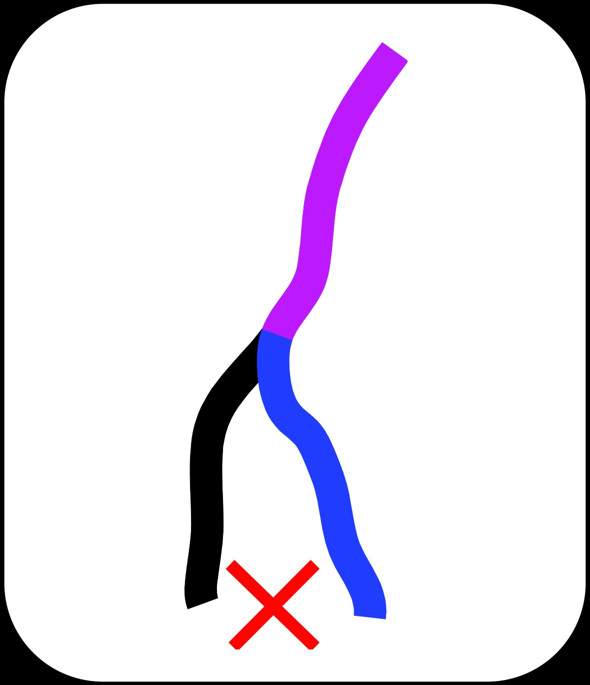


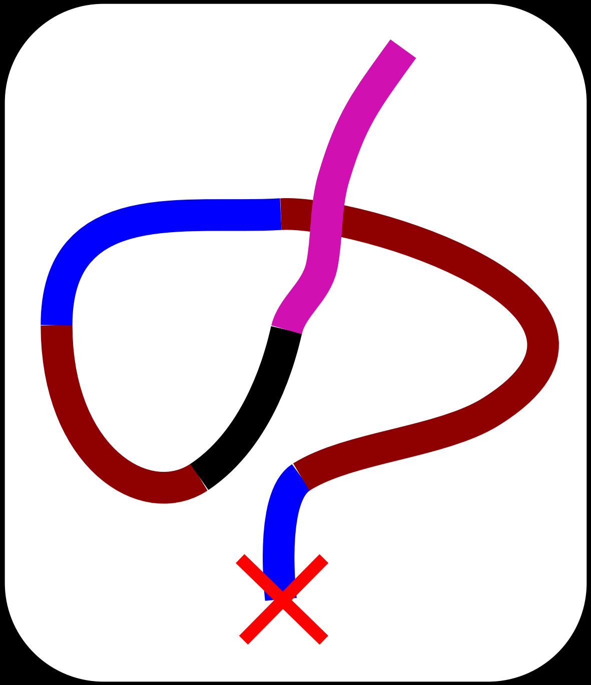
Search to a Goal



- Use your favorite discrete search
- Planning-like problem

Path Quality Tradeoffs





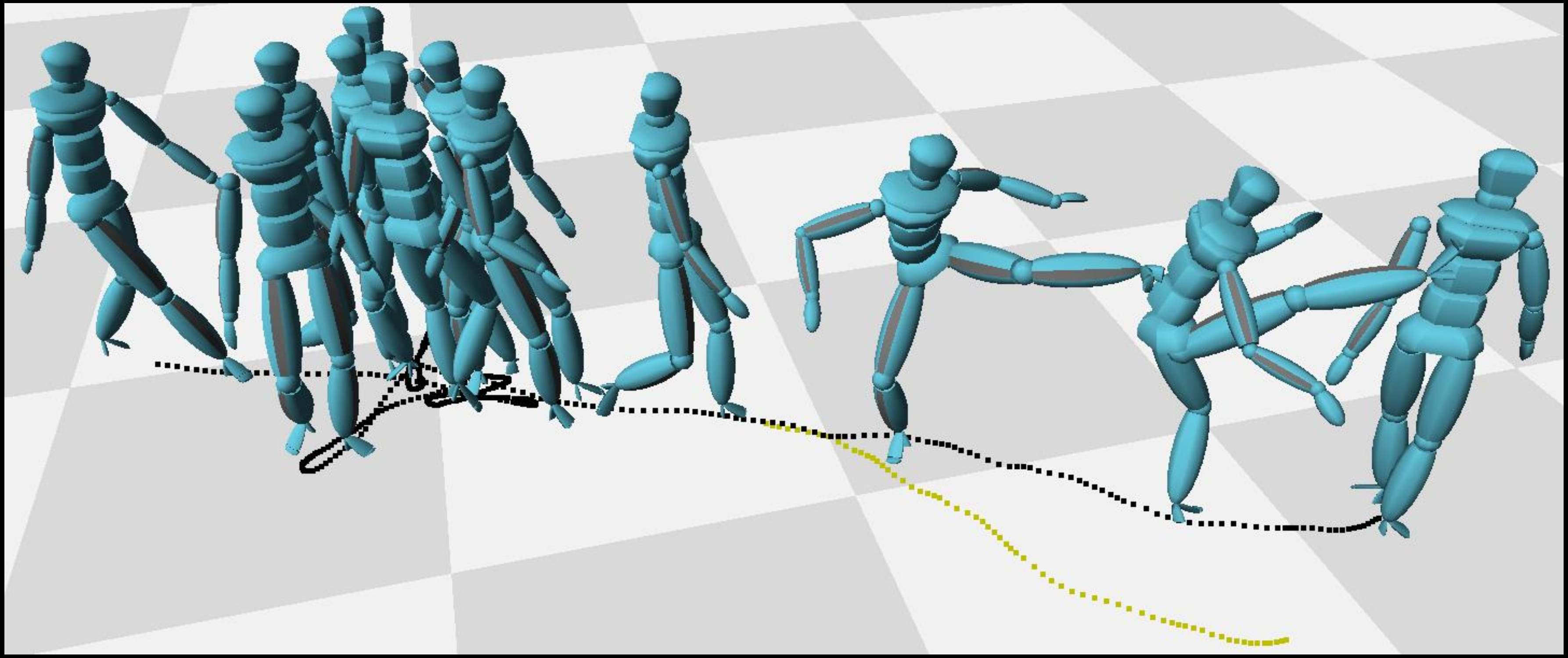
Discrete choices:

Can't get exactly to goals

Discrete choices:

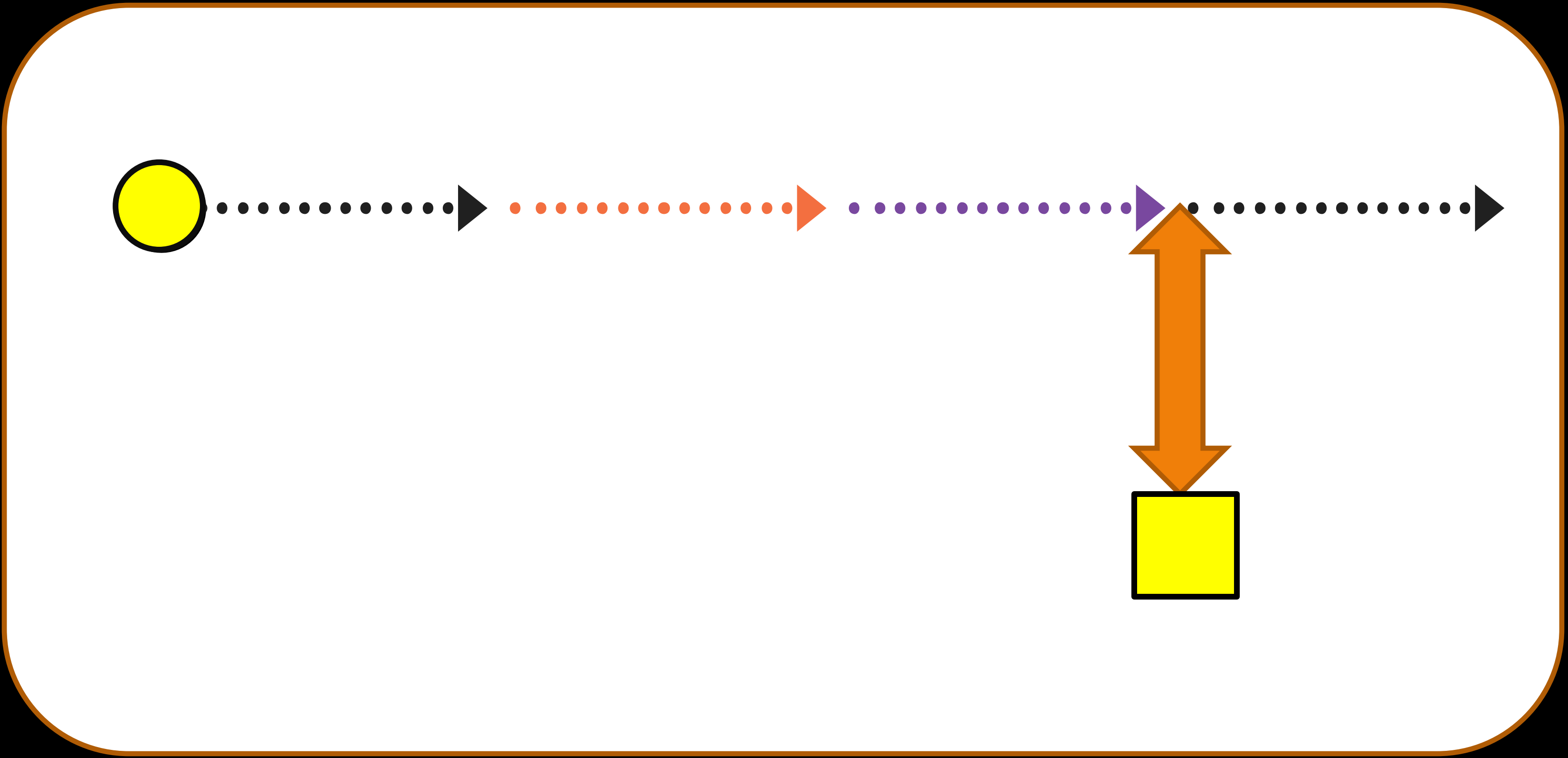
Closest fit might not be a good path

Bad paths happen

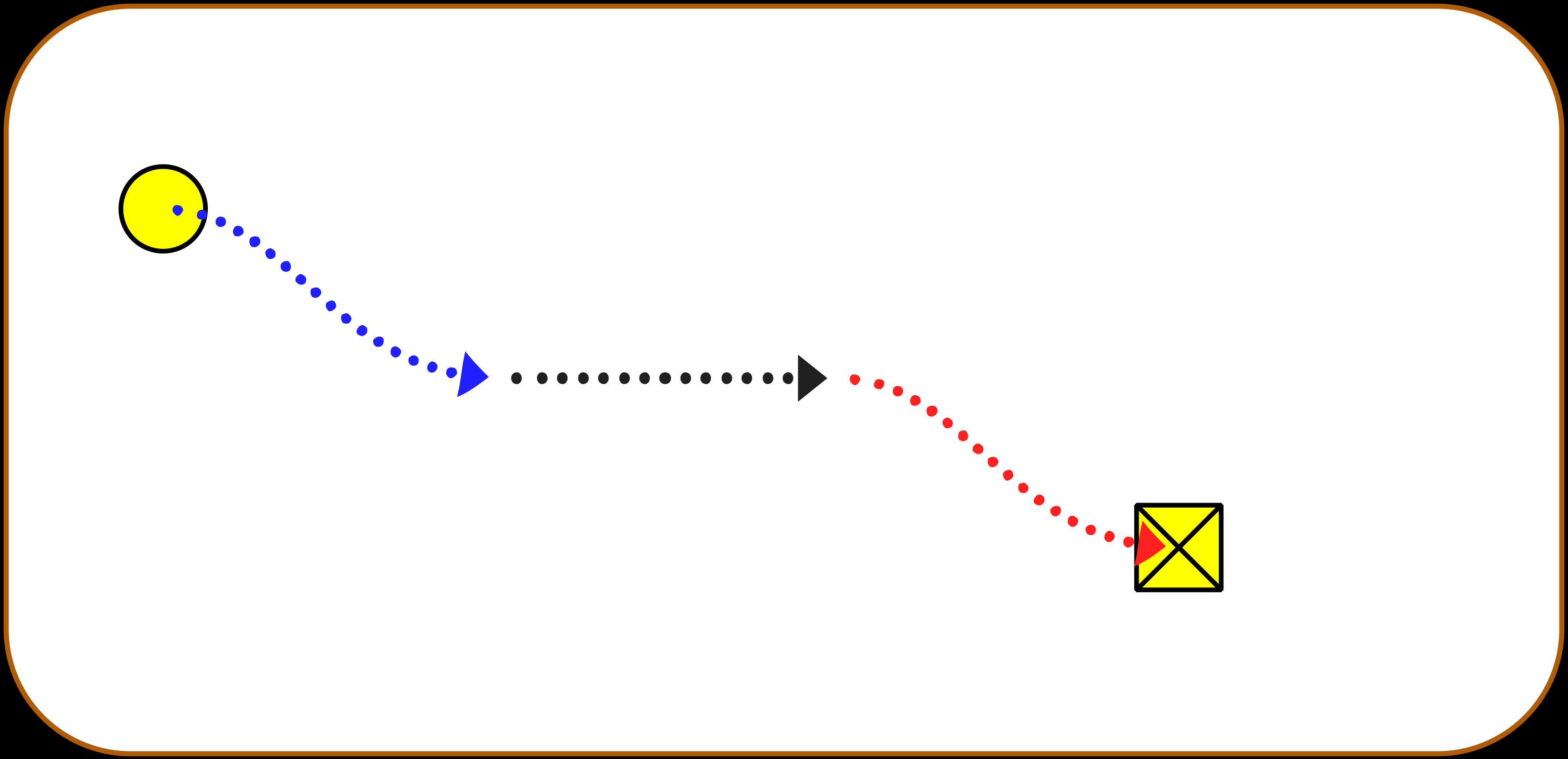


From here to there…

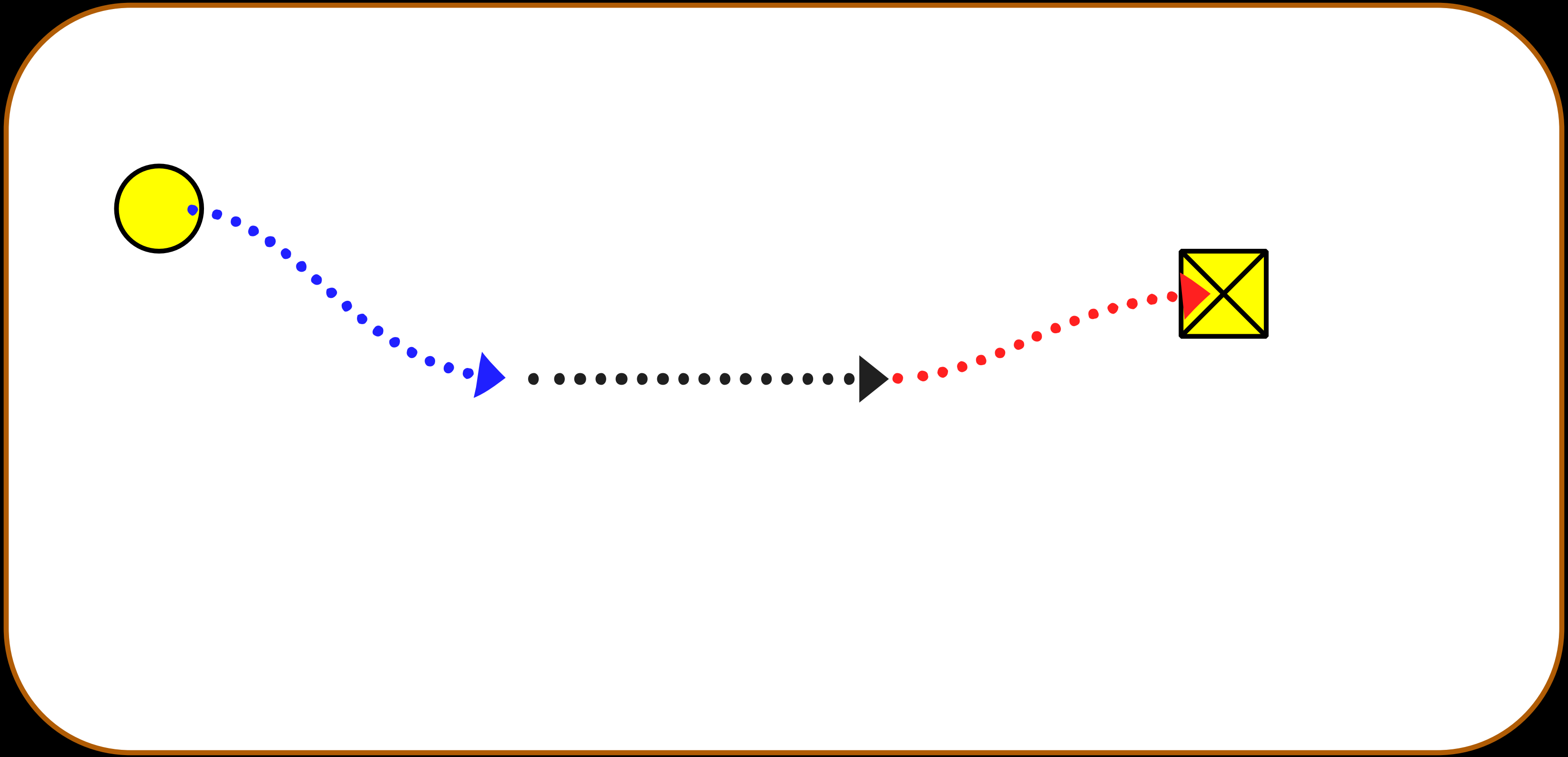
Objective:
Path close to goal



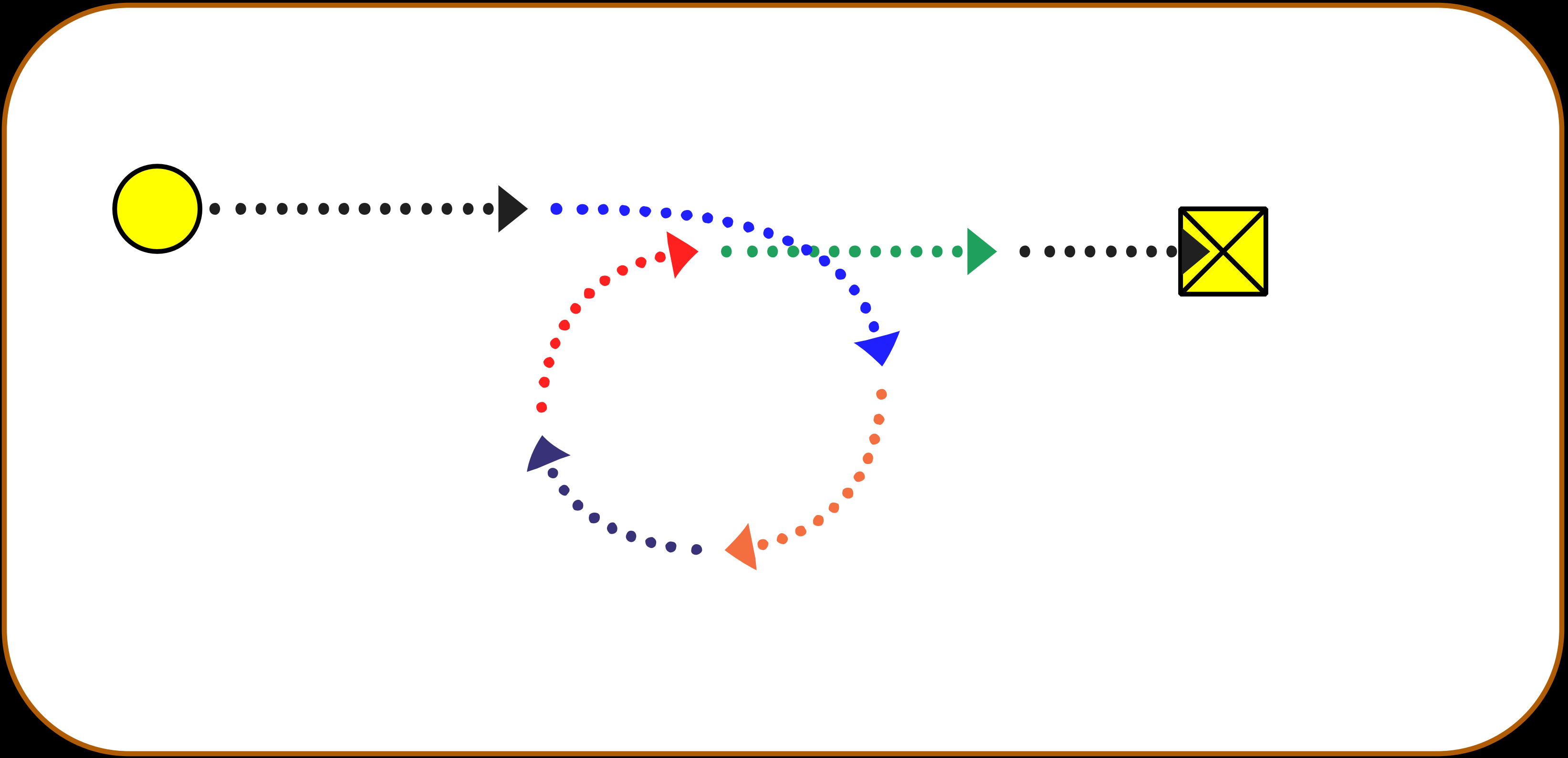
Search for walks on the graph that minimize the goal



Best answers depend on your repetiore

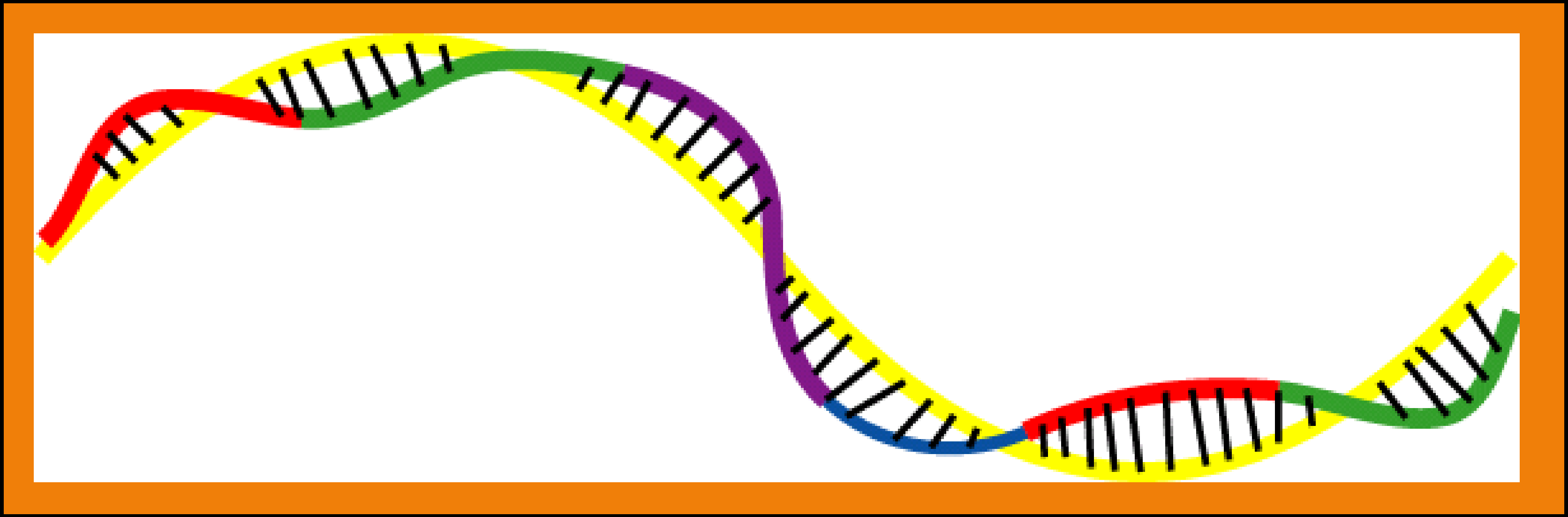


Getting exactly there, might have other issues

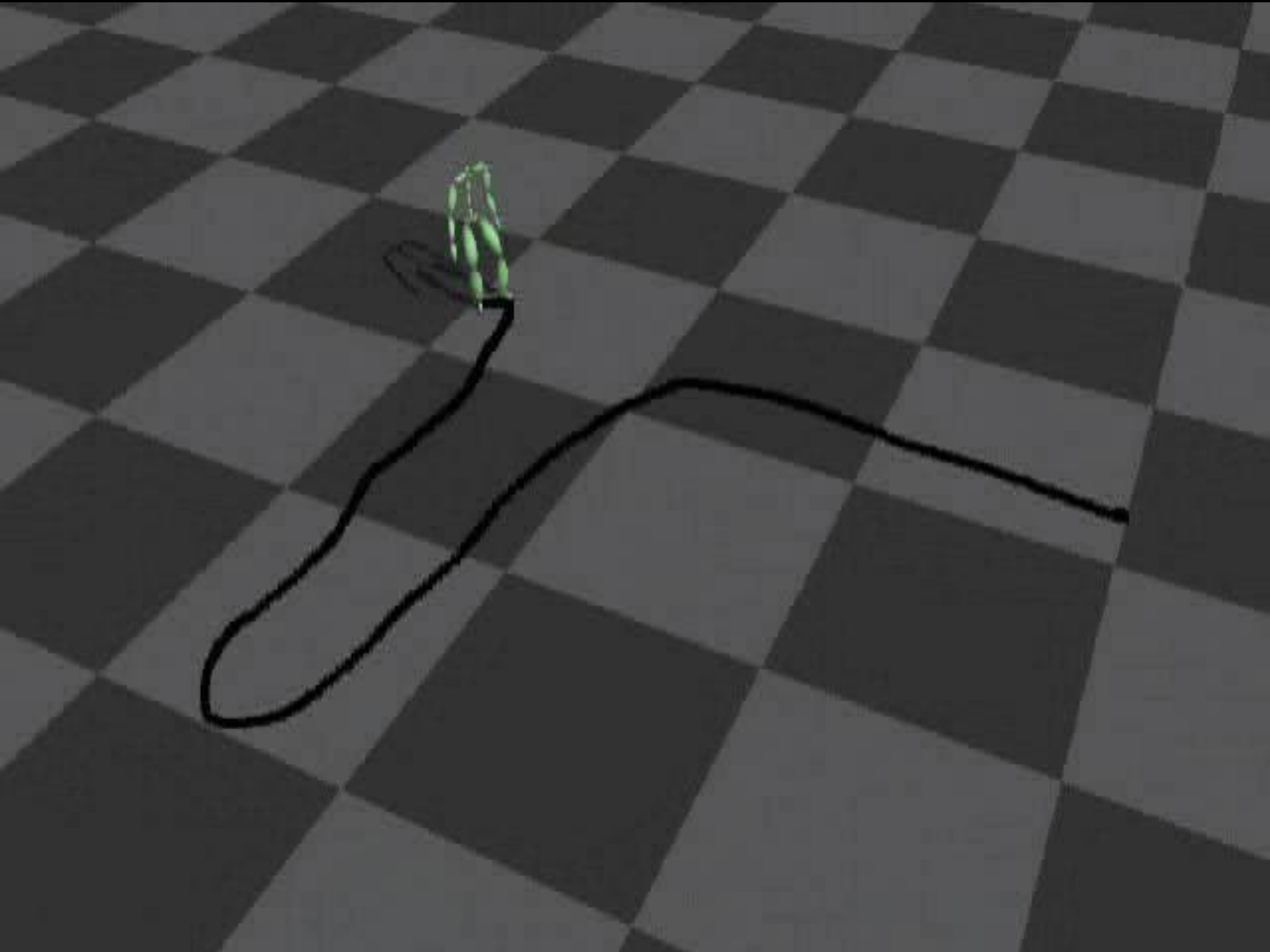


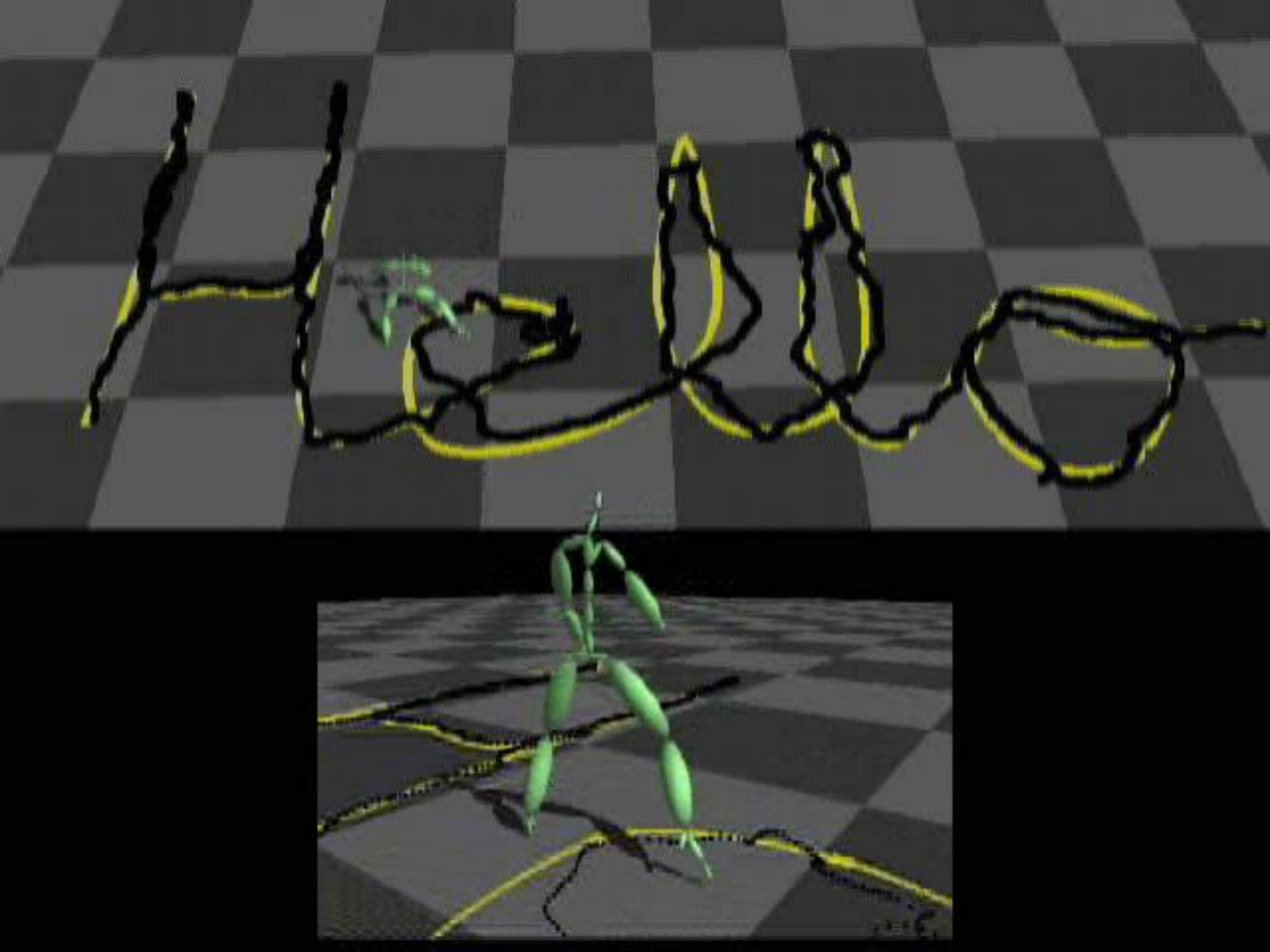
Path Following

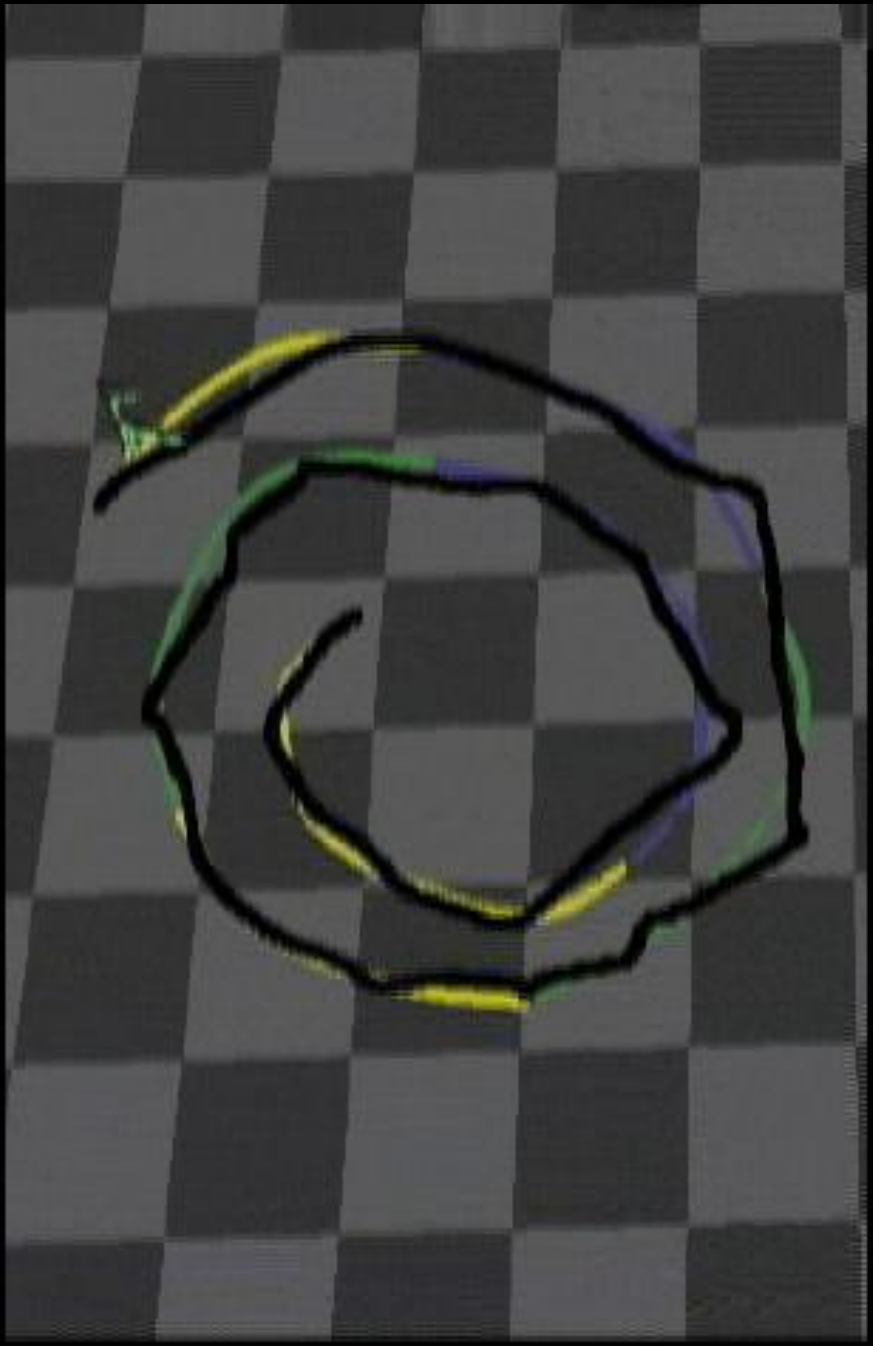
Kovar 2002

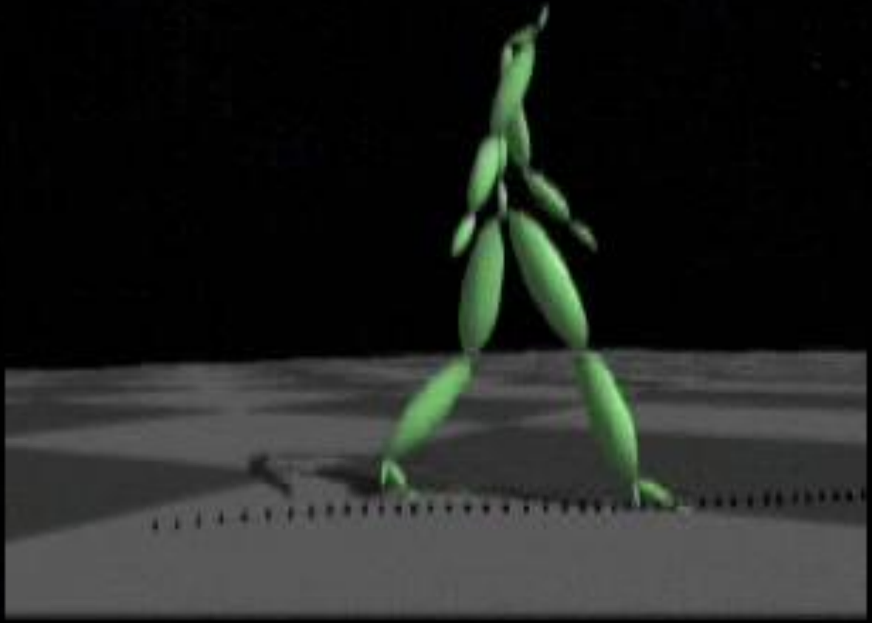


- Minimize distance to path (over whole path)
- We used branch-and-bound




Hello



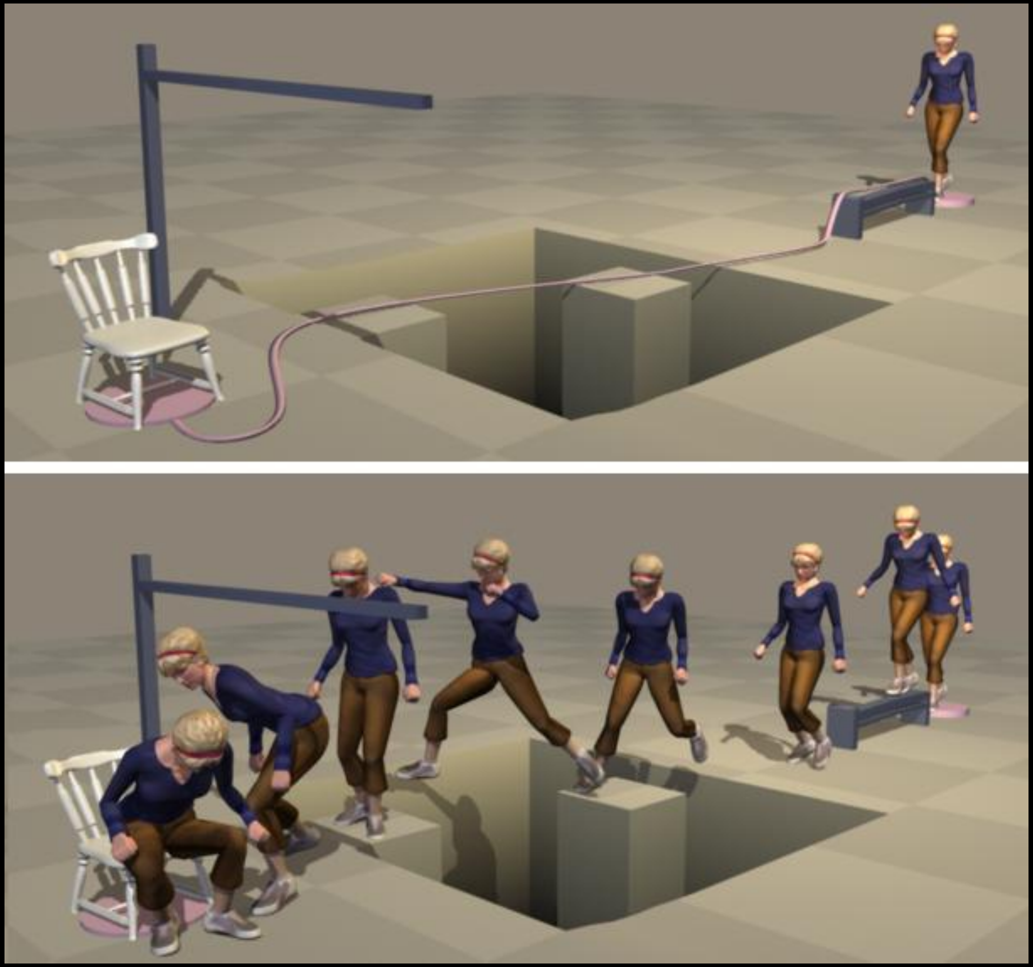


More Complex

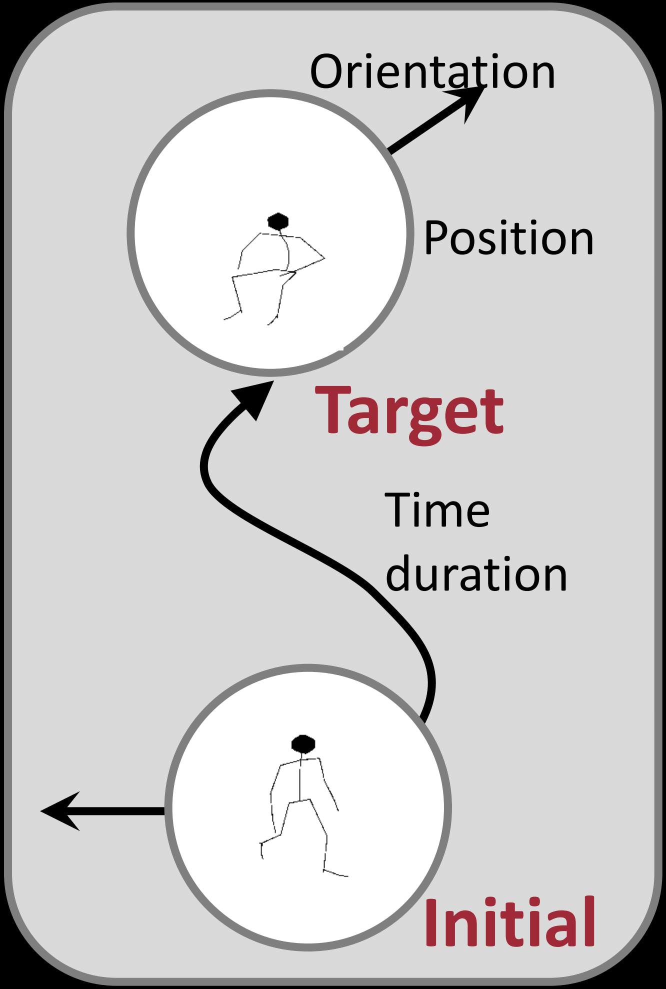
Safanova and Hodgins 2007

- Better search

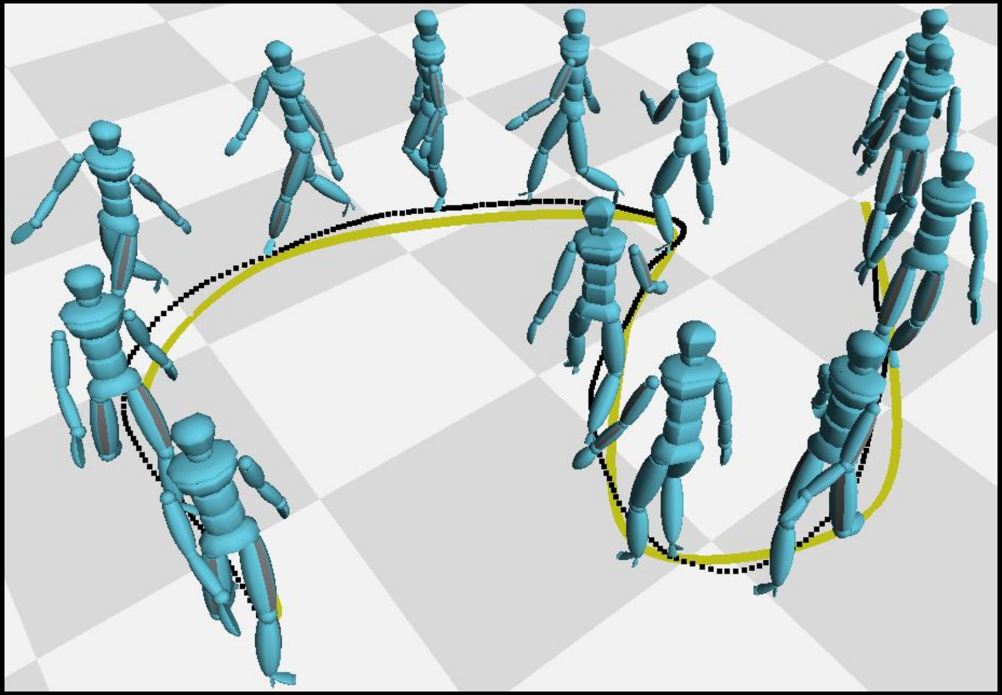
- Better graphs



Planning vs. Synthesis



Planning



Synthesis

An Aside

How do you get from here to there in practice?

- Separate path planning from movement

- Character follows correct path
- Animation “in-place” to make it look better

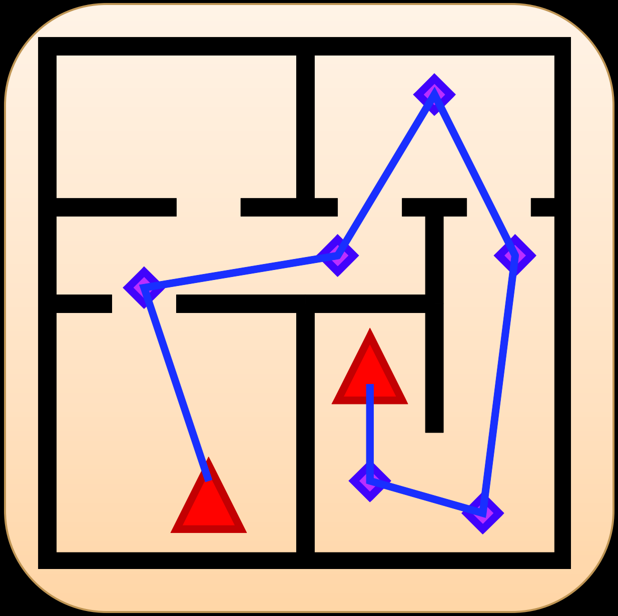
Multi-Level Solutions

Different methods for different aspects

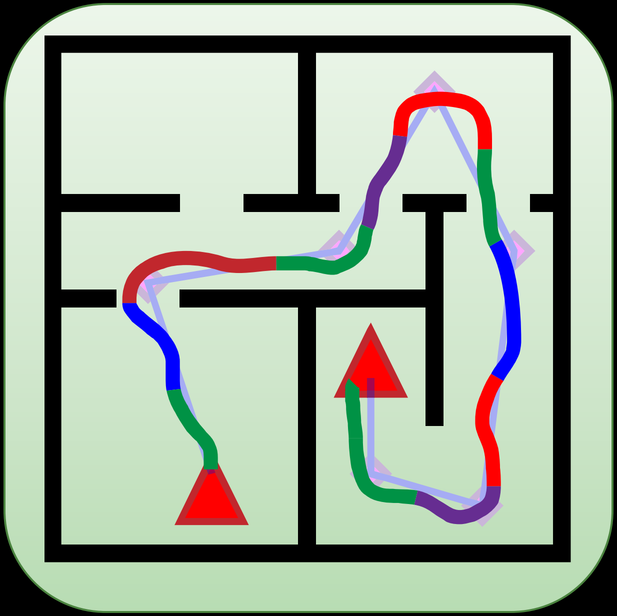
- Motion planning to get rough path
- Motion synthesis to follow path
  - Possibly only gets close
- Motion Adjustment to exactly meet goals

Example Multi-Level Solution

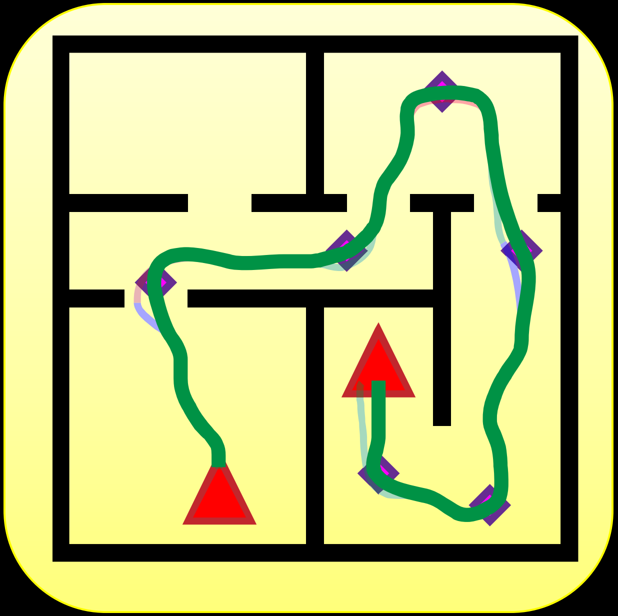
Sung, Kovar, Gleicher SCA 05



Motion Planning:
PRM-based



Motion Synthesis:
Greedy search of structured graph

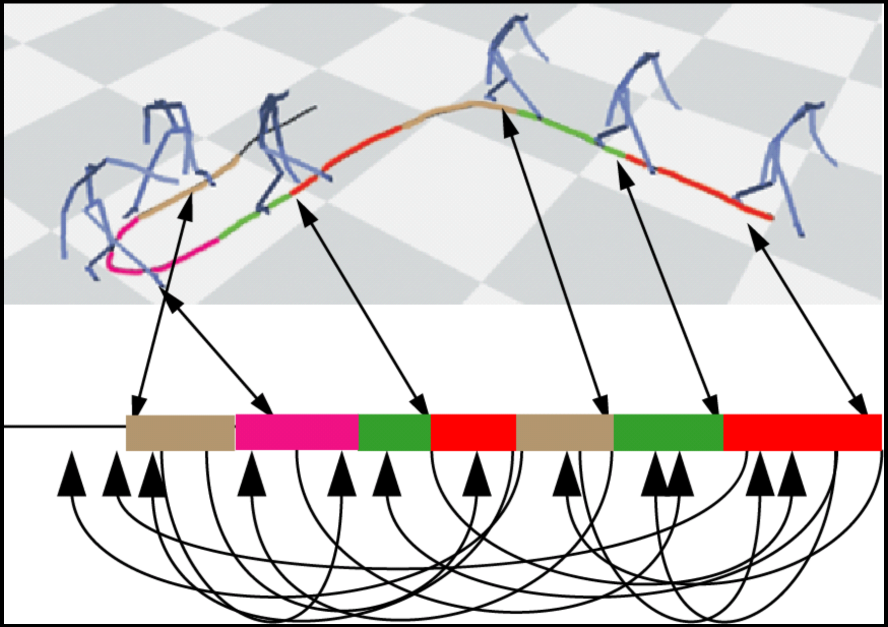


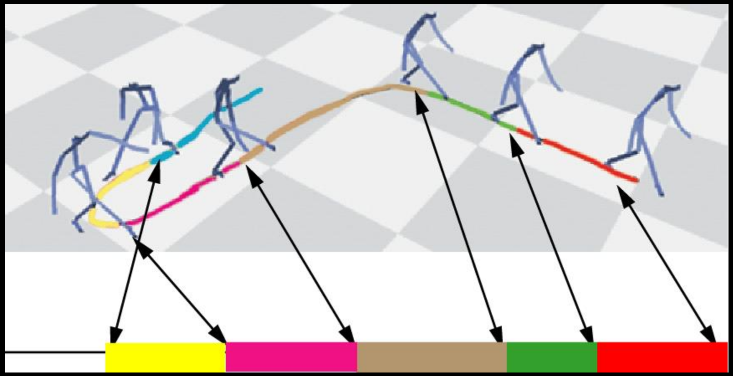
Fine Adjustment:
Distribute error

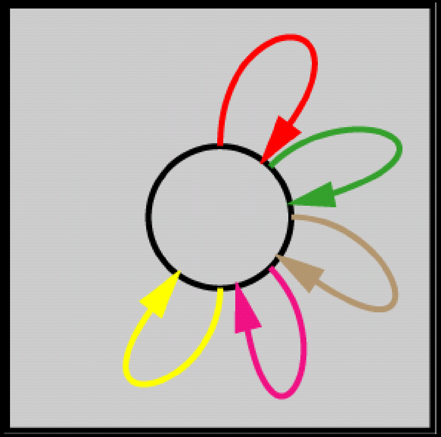
Did we solve the right problem?

Interactive Control

How do you do interactive control?







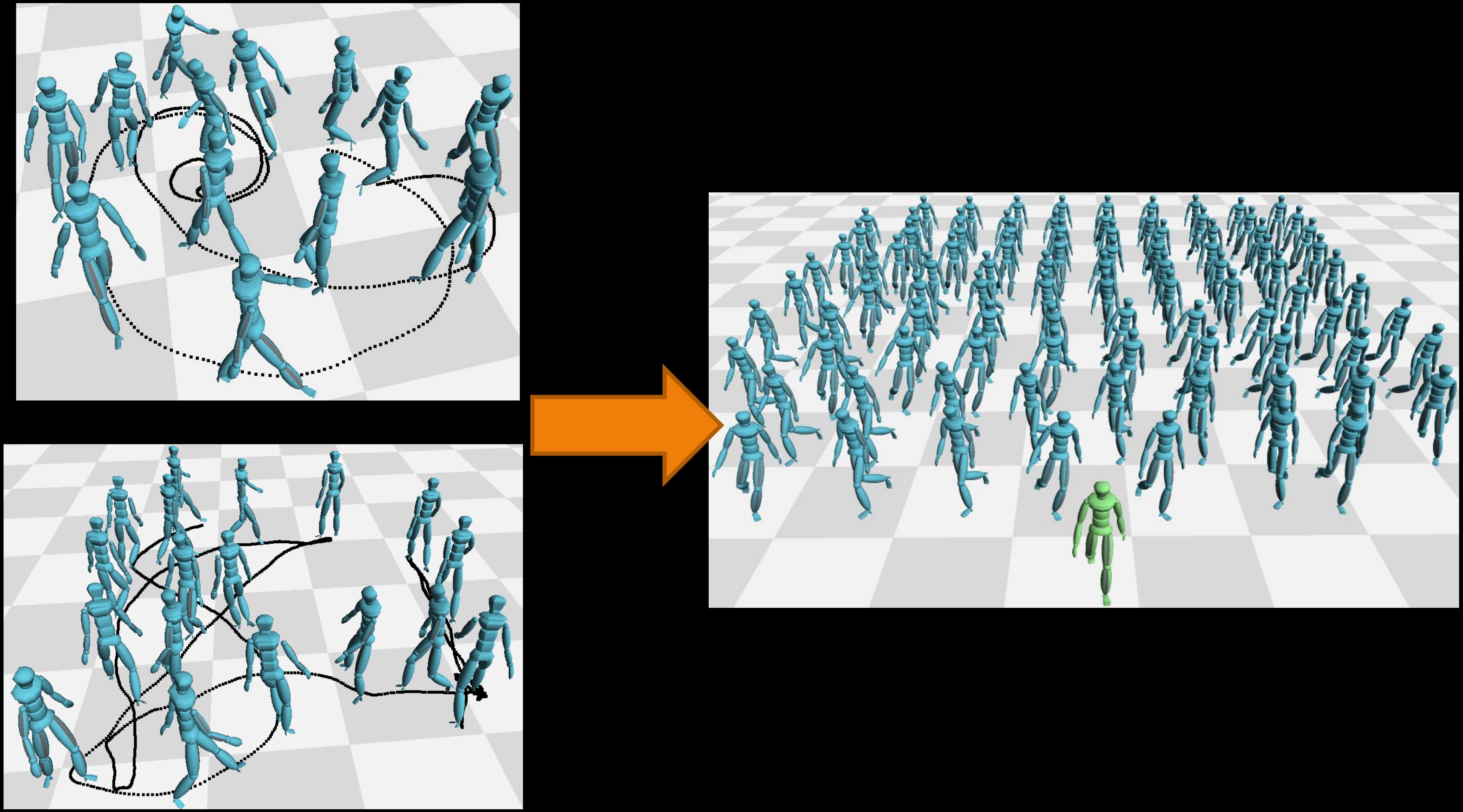
Gleicher et al. I3D 2003

Automating Interactive Graphs

- Automate construction of contrived graphs

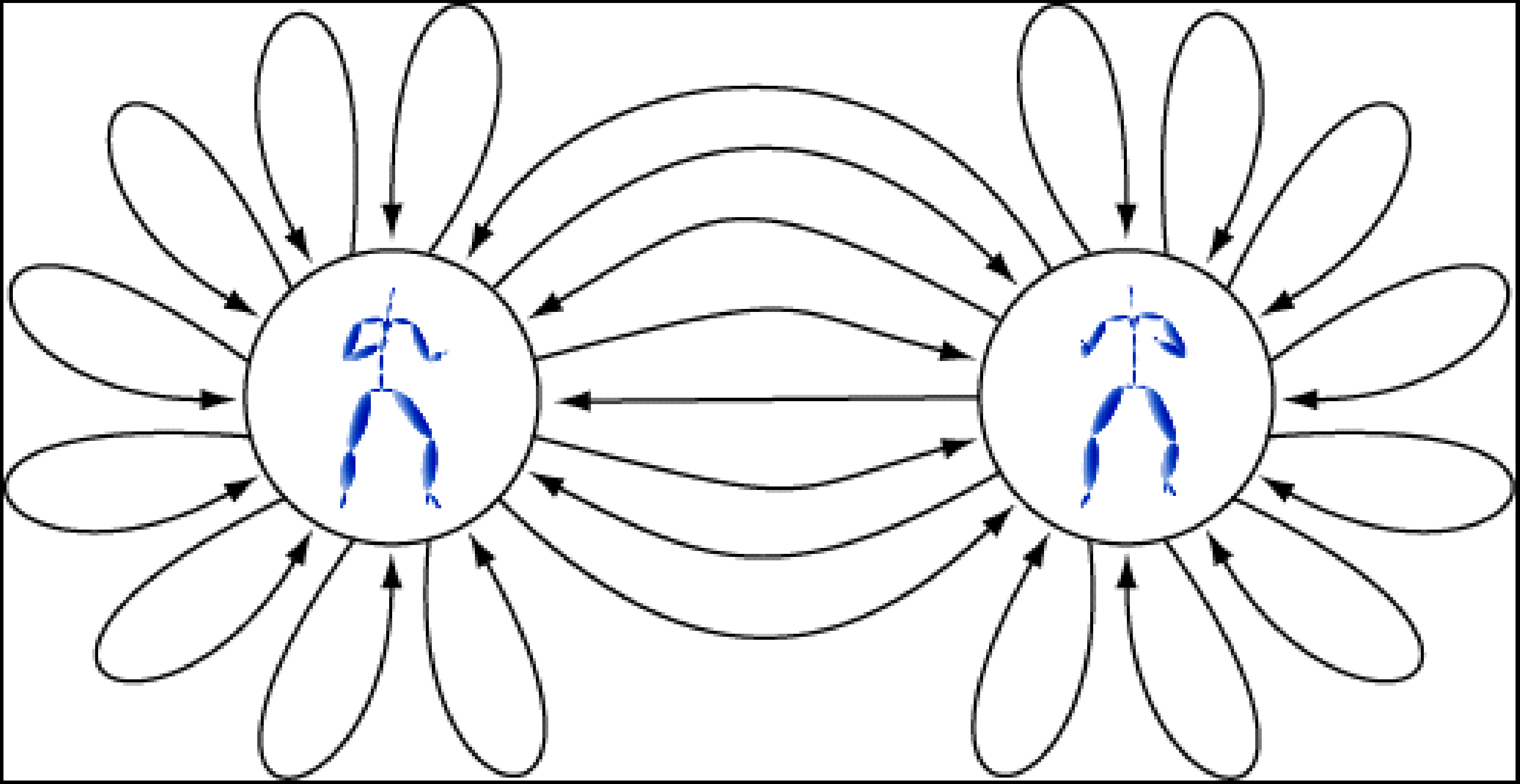
- Do a little searching for what works
  - Precompute searches / Reinforcement Learning

Automatically find examples in data



Snap-Together Motion

Gleicher, Kovar, Shin, Jepsen 2003

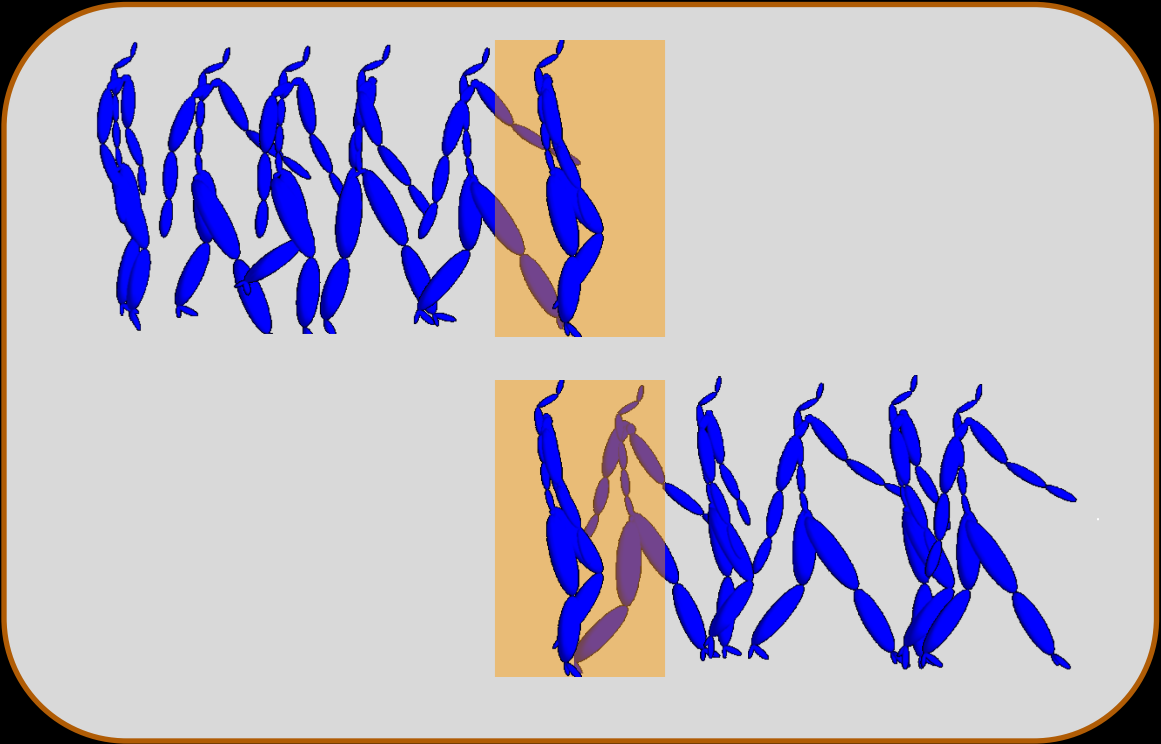


- Find those key nodes (poses)

Snappable Motions

A different way to think about transitions

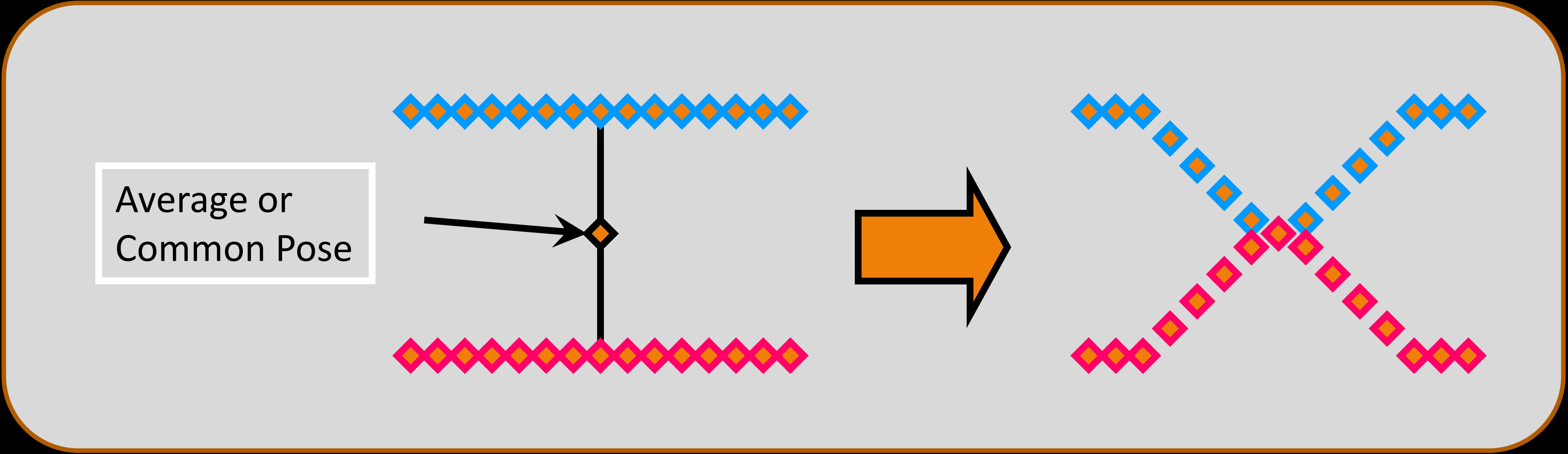
- Want motions that match exactly
  - – Match pose and derivatives (at multiple scales)



Make them match

Transition to common pose

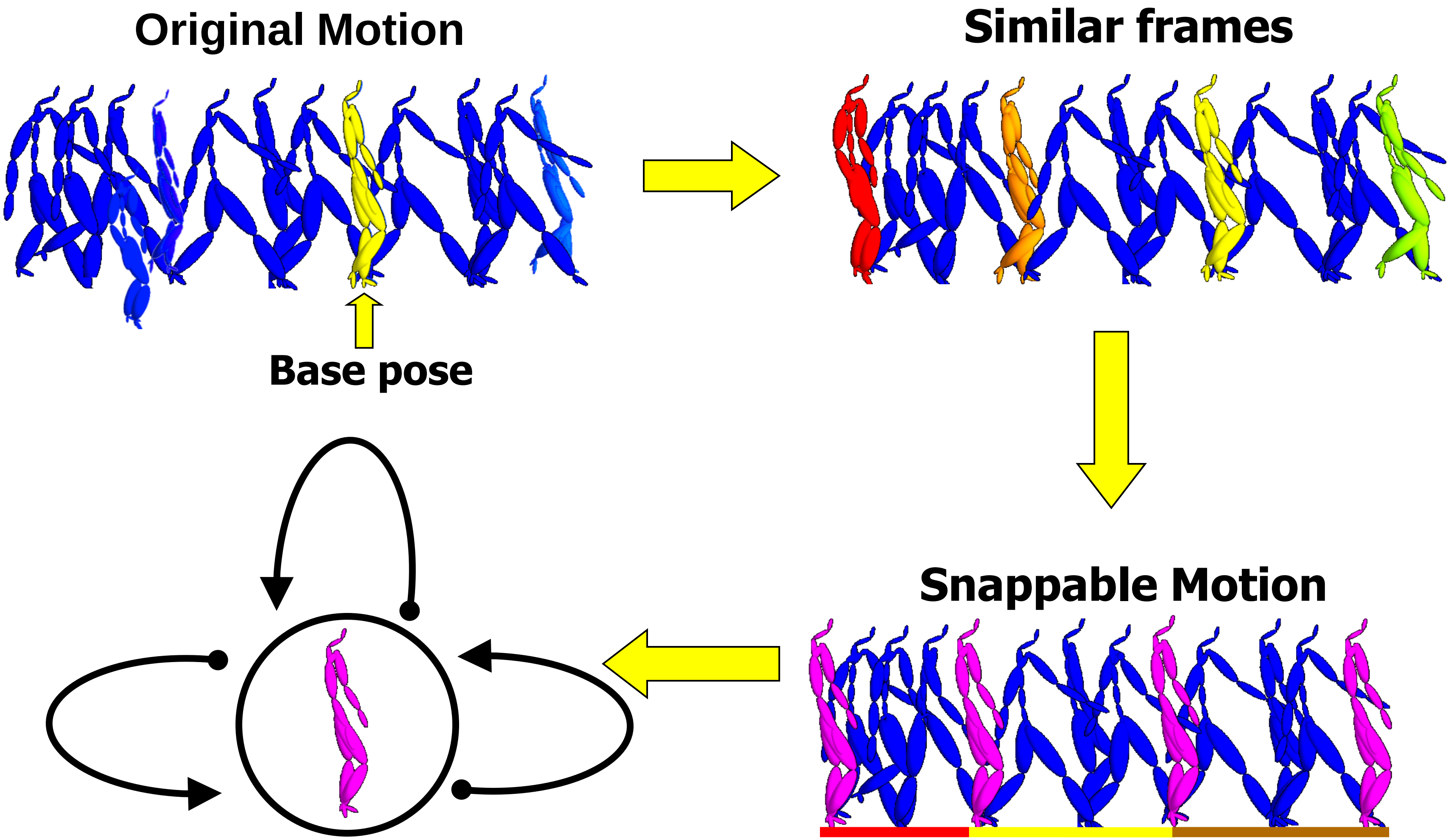
- Make common poses in motions
  - For things that start out close enough



Semi-Automatic Graph Construction

- Pick set of *match frames*
  - User selects
  - System picks “best” one
- Modify motions to build hub node
- Check graph and transitions

Snap-Together Motion!

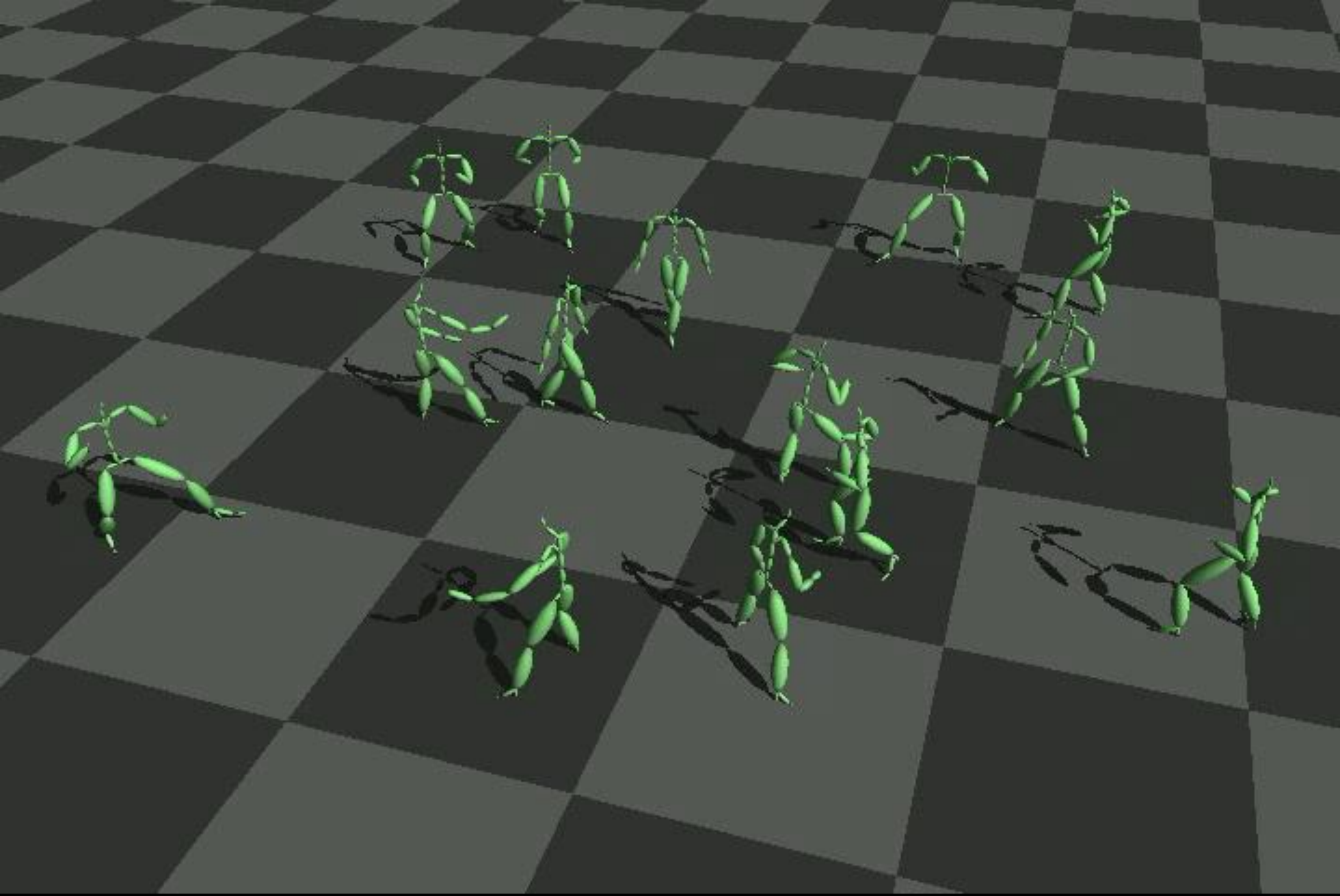


Original Motion

Base pose

Similar frames

Snappable Motion



SBE in Practice vs. Research
(practice has been doing it longer)

Practice (real games)

- Careful Preparation
- Well planned data sets
- Carefully chosen examples
- Manual data preparation
- Careful design simplifies control

Research

- Automation
- Less control over data
- Automate search for examples
- Automatic data preparation
- Clever search for control with arbitrary data

Does automation help with scalability?

- Yes – it allows working with more data

- No – lack of control creates new problems
  more data means more problems

- Potentially – we need to do more research