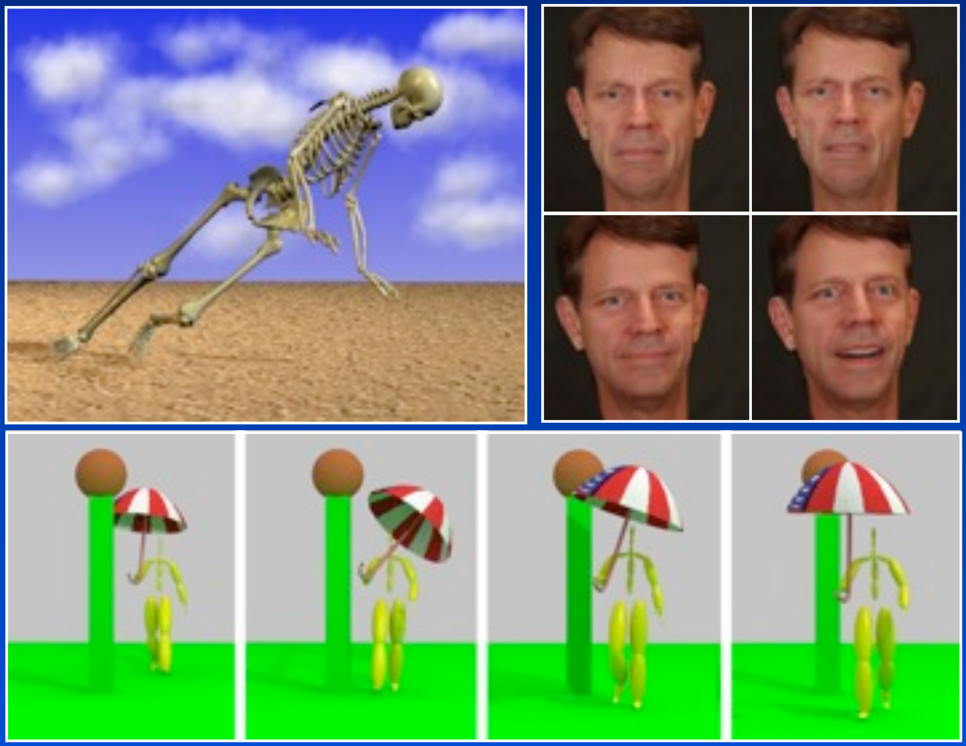
Virtual Characters for Computer Games



Petros Faloutsos
www.cse.yorku.ca/~pfal

Virtual Characters for Computer Games

...and why I should learn Java, inheritance and all those things

Who am I?

New faculty member in digital media

pfal@cse.yorku.ca
www.cse.yorku.ca/~pfal

Main Research Directions

Human Animation

- Motion control
- Facial animation
- Autonomous virtual humans
- Sensing, Interaction with the Environment

Micro-architectures for Interactive Applications

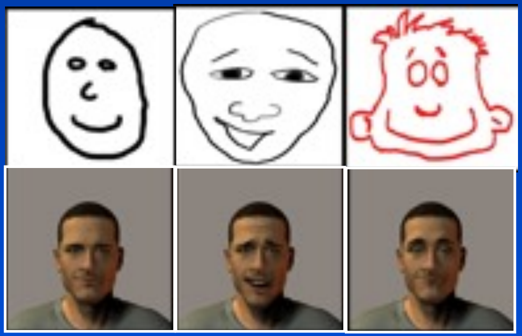
- Real-time rendering
- Real-time physics
- Real-time artificial intelligence

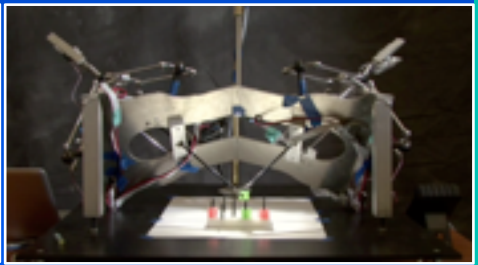
Human Computer Interaction

- Assistive Devices
- Novel interfaces

Digital Media In Medicine

- Automatic Assessment of Surgical Skills
- Telementoring - UCLA Laparobot
- Surgical Assistants





Take home message

Everything you learn can help with/lead to exciting careers

- R&D for special effects studio
- Technical Director for a special effects studio
- Computer Games Programmer
- Software Engineer

I will show you what past u/graduate students have done that lead to great positions

But first some history and basics

Computer Graphics

Pictures!

Motion!

Images copyrighted by PIXAR and Disney

Art

Mathematics

Vision

Optics

Biomechanics

Physics

Engineering

Artificial Intelligence

Programming!

Movies

To reality and beyond !

Movies

Special Effects

Movies

Compositing

Cartoons

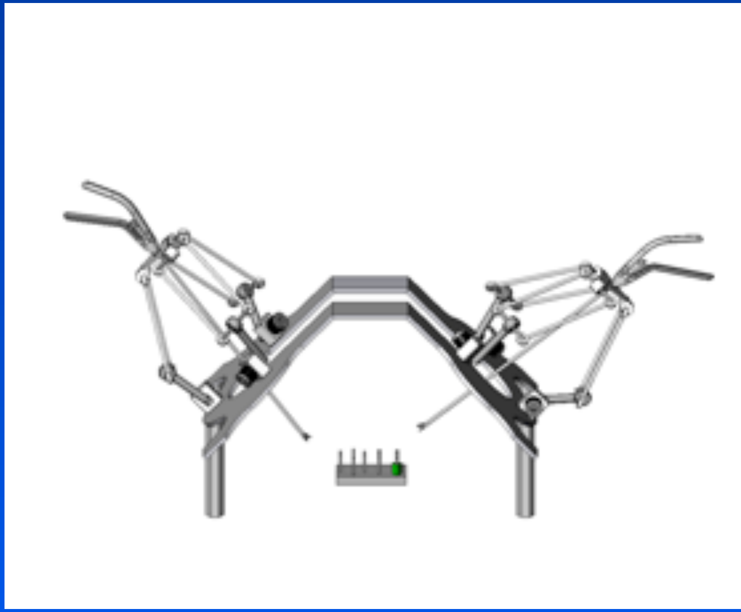
Games

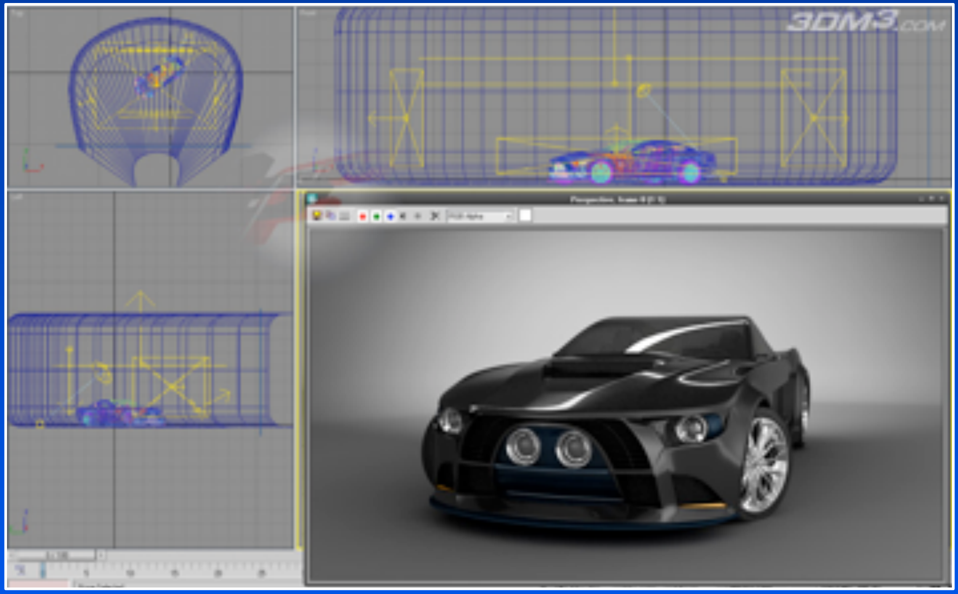
Focus on interactivity

Computer-Aided Design

Precision modeling

Engineering visualization

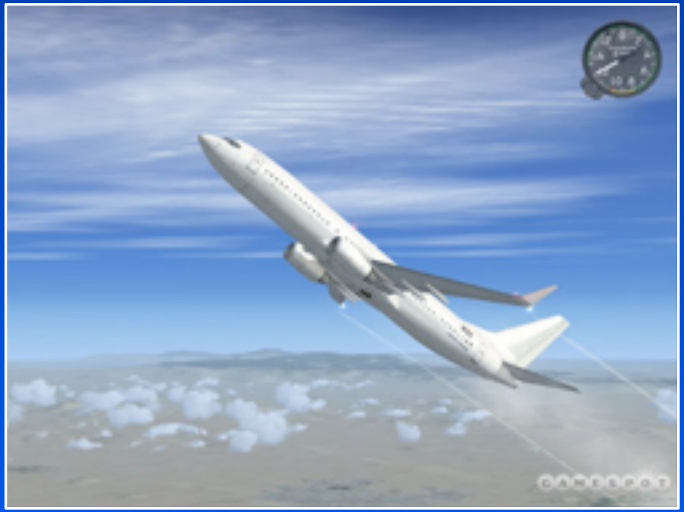


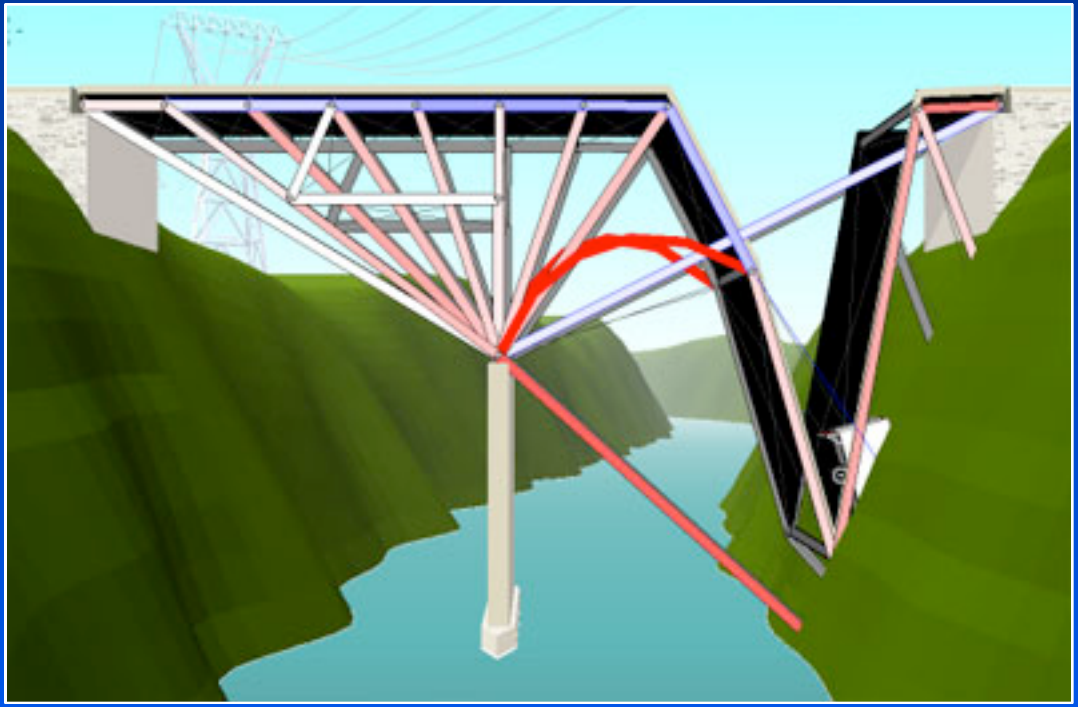


Computer Aided Design

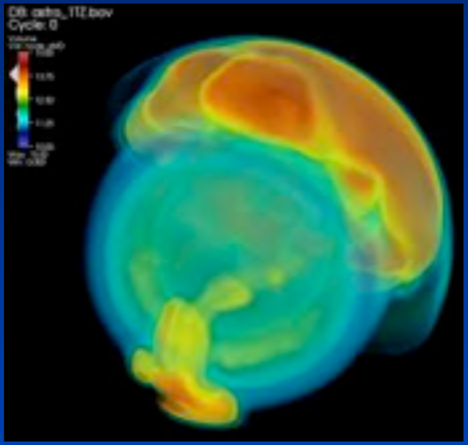
It is not just about visualization

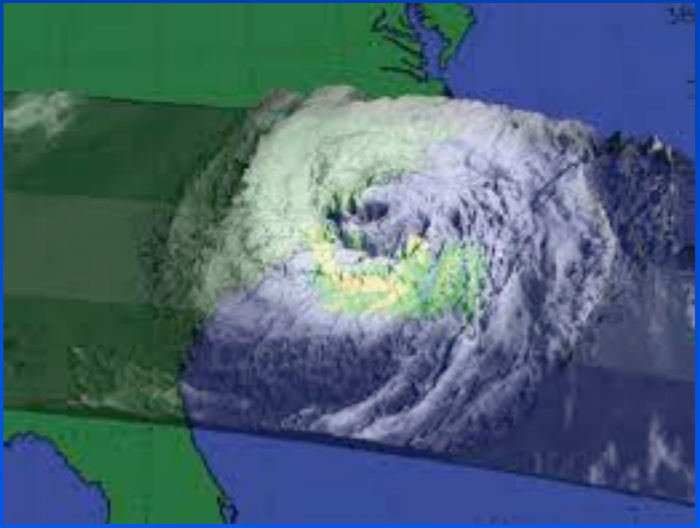
- Simulation is useful





Visualization: Scientific



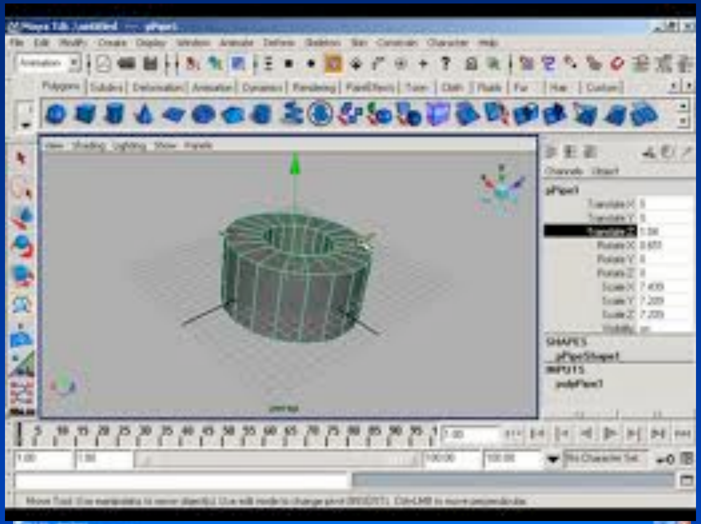


Visualization: Architectural

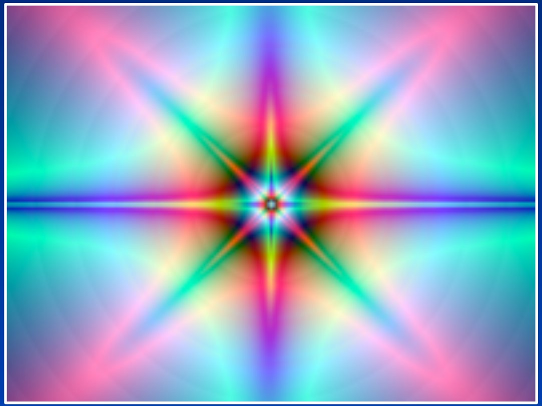
Visualization: info

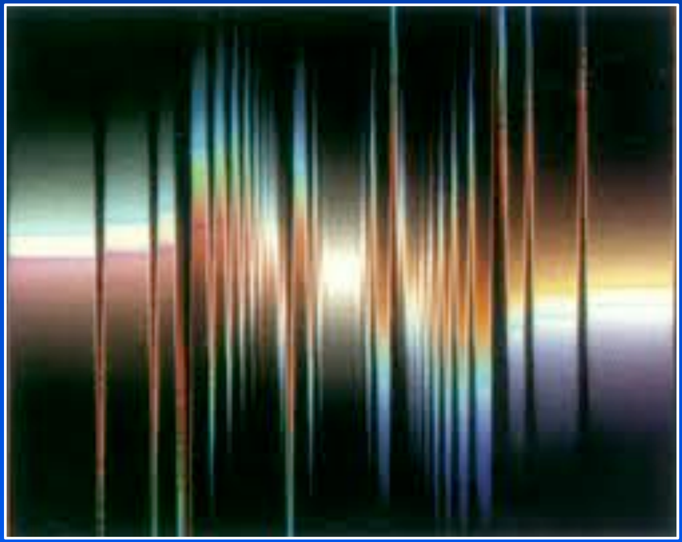
- Geographical Information systems
  - *Maps*
- Personal Information
- Massive dataset visualization

Graphical User Interfaces



Digital Art





Why do we need to program?

Things constantly change

- Evolving hardware (e.g. multi-core)
- Evolving software (e.g. new languages, new structures)
- New techniques

User requirements

- Adjustment specific to applicati
- Adjustment to director's demands

Complexity

- Massive pieces of software
- Maintenance

Basic Technical Elements

- Modeling
  - *How do we model (mathematically represent) objects?*
  - *How do we construct models of specific objects?*
- Animation
  - *How do we represent the motion of objects?*
  - *How do give animators control of the motion?*
- Rendering
  - *How do we simulate the real-world behaviors of light?*
- Interaction
  - *How do we enable humans and computers to interact?*
  - *How do we design human-computer interfaces?*

Modeling

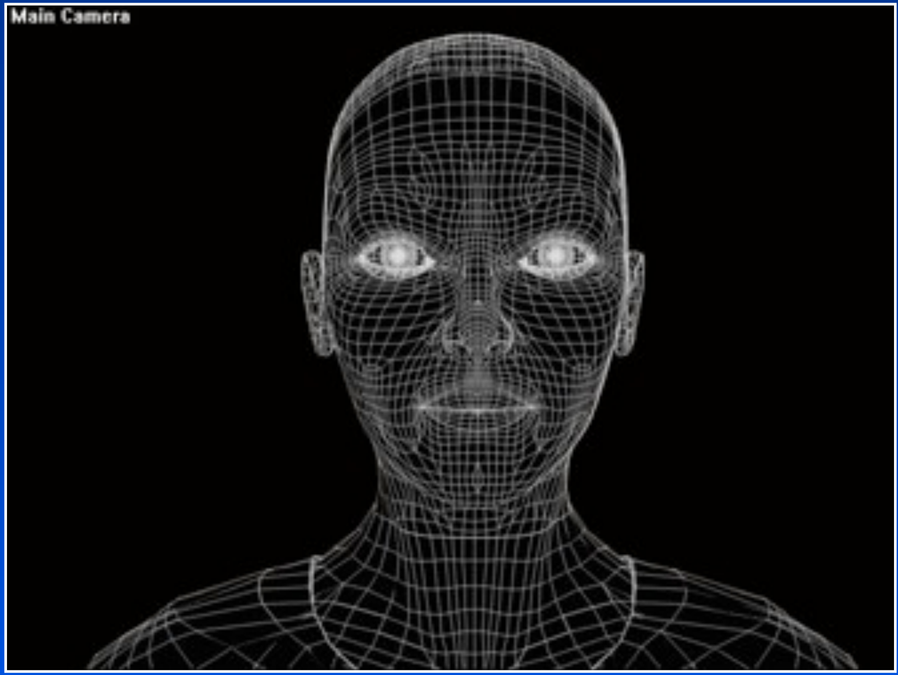
Primitives

- 3D points
- 3D lines and curves
- surfaces (BREPs): polygons, patches
- volumetric representations
- image-based representations

Attributes

- Color, texture maps
- Lighting properties

Geometric transformations



Rendering

Visibility

Simulating light propagation

- Reflection
- Absorption
- Scattering
- Emission
- Interference

Animation

- *Keyframe, motion capture*
- *Physics-based animation*
- *Autonomous motion planning*

Animation

Keyframe, motion capture

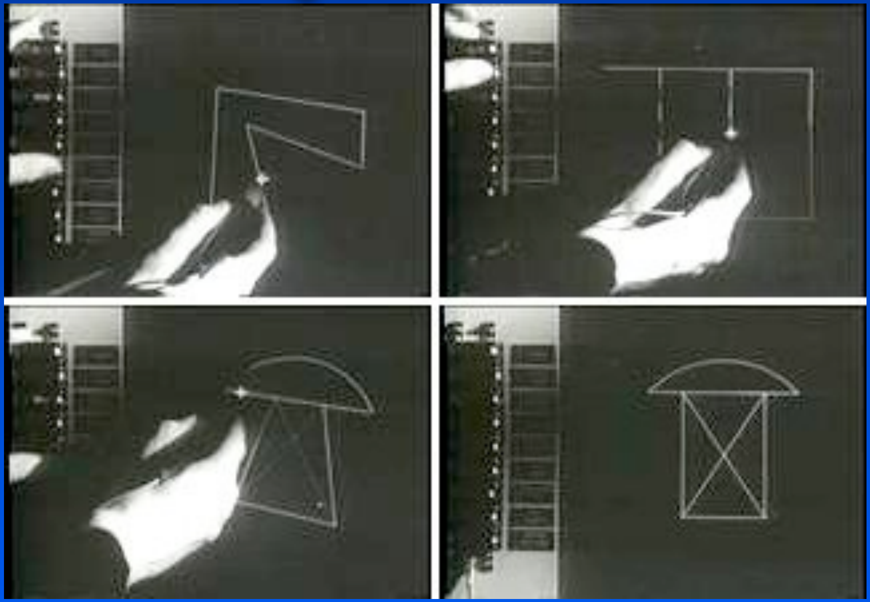
Physics-based animation

Autonomous motion planning

Genesis of Computer Graphics and Interactive Techniques

A PhD project at MIT in the early 1960s

- Ivan E. Sutherland, 1963
  - *"Sketchpad, a man-machine graphical communication system"*



Quiz

Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

- When was the term Computer Graphics first stated?

Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

- When was the term Computer Graphics first stated?
  - *William Fetter of Boeing coins the term "computer graphics" for his human factors cockpit drawings 1960.*

Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

- When was the term Computer Graphics first stated?
  - *William Fetter of Boeing coins the term "computer graphics" for his human factors cockpit drawings 1960.*
- When was the GUI developed?

Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

- When was the term Computer Graphics first stated?
  - *William Fetter of Boeing coins the term "computer graphics" for his human factors cockpit drawings 1960.*
- When was the GUI developed?
  - *GUI developed by Xerox (Alan Kay) 1969*

Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

- When was the term Computer Graphics first stated?
  - *William Fetter of Boeing coins the term "computer graphics" for his human factors cockpit drawings 1960.*
- When was the GUI developed?
  - *GUI developed by Xerox (Alan Kay) 1969*
- When was Tron released?

Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

- When was the term Computer Graphics first stated?
  - *William Fetter of Boeing coins the term "computer graphics" for his human factors cockpit drawings 1960.*
- When was the GUI developed?
  - *GUI developed by Xerox (Alan Kay) 1969*
- When was Tron released?
  - *Disney contracts Abel, III, MAGI and DE for computer graphics for the movie Tron released in 1981.*

Quiz (contd)

Quiz (contd)

- Which is the first animated movie to employ CG?

Quiz (contd)

- Which is the first animated movie to employ CG?
  - *The Great Mouse Detective was the first animated film to be aided by CG.*

Quiz (contd)

- Which is the first animated movie to employ CG?
  - *The Great Mouse Detective was the first animated film to be aided by CG.*
- When was DOOM released ?

Quiz (contd)

- Which is the first animated movie to employ CG?
  - *The Great Mouse Detective was the first animated film to be aided by CG.*
- When was DOOM released ?
  - *1993.*

Quiz (contd)

- Which is the first animated movie to employ CG?
  - *The Great Mouse Detective was the first animated film to be aided by CG.*
- When was DOOM released ?
  - *1993.*
- Which was the first totally computer generated movie?

Quiz (contd)

- Which is the first animated movie to employ CG?
  - *The Great Mouse Detective was the first animated film to be aided by CG.*
- When was DOOM released ?
  - *1993.*
- Which was the first totally computer generated movie?
  - *Toy Story 1995*

Quiz (contd)

Quiz (contd)

- Which is bigger in gross revenue, the Gaming Industry or Hollywood?

Quiz (contd)

- Which is bigger in gross revenue, the Gaming Industry or Hollywood?
  - *The Gaming Industry.*

Quiz (contd)

- Which is bigger in gross revenue, the Gaming Industry or Hollywood?
  - *The Gaming Industry.*
- Which is the best selling game of all time?

Quiz (contd)

- Which is bigger in gross revenue, the Gaming Industry or Hollywood?
  - *The Gaming Industry.*
- Which is the best selling game of all time?
  - *Mario ($193M), Pokemon($155), Final Fantasy ($68M), The Sims ($100)*

First computer game?

First computer game?

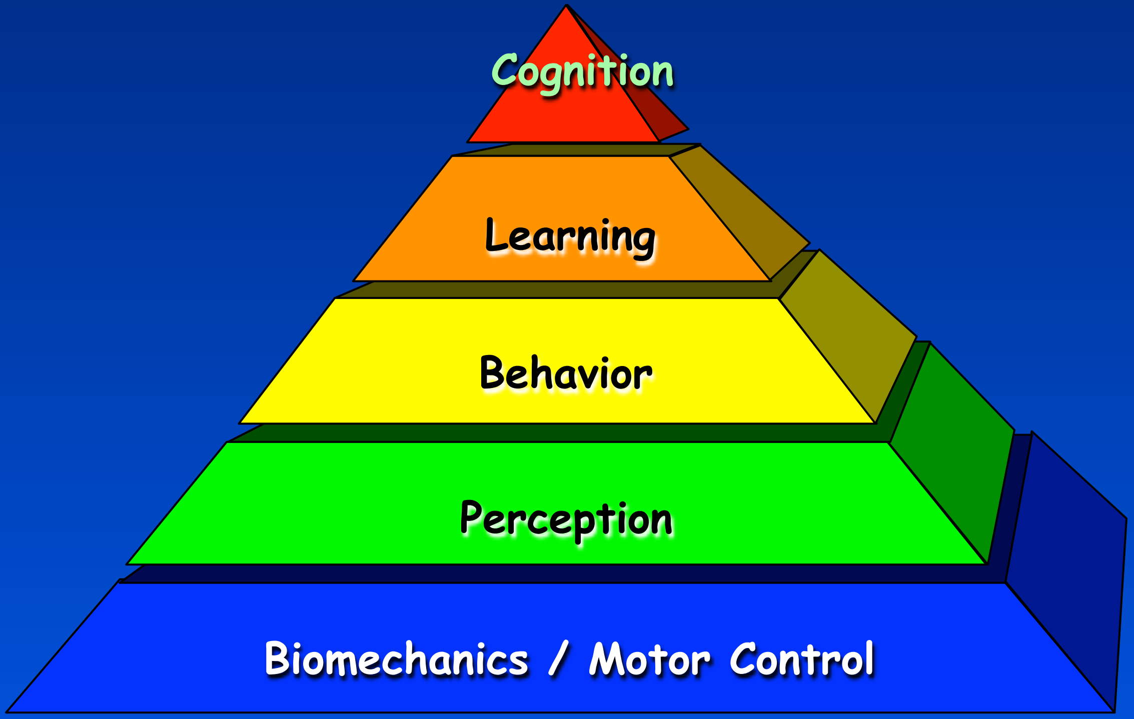
Pong, Atari, 1972

First computer game?

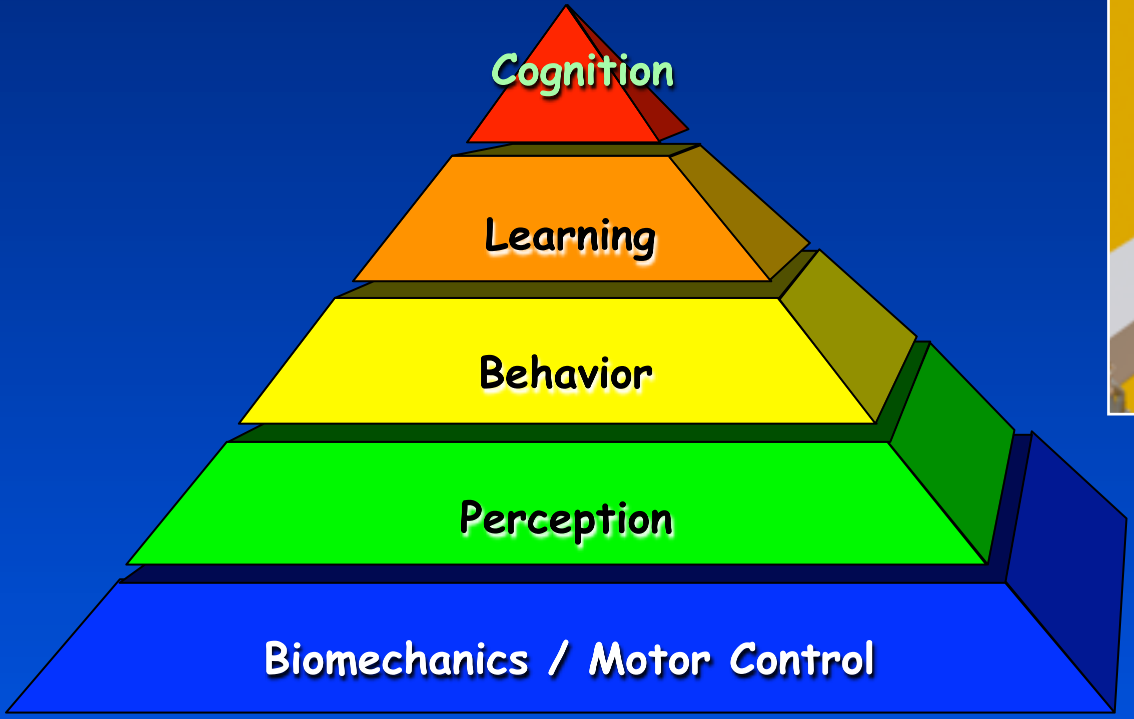
First computer game?

Spacewars PDP –1 MIT, 1961

Intelligent Virtual Characters



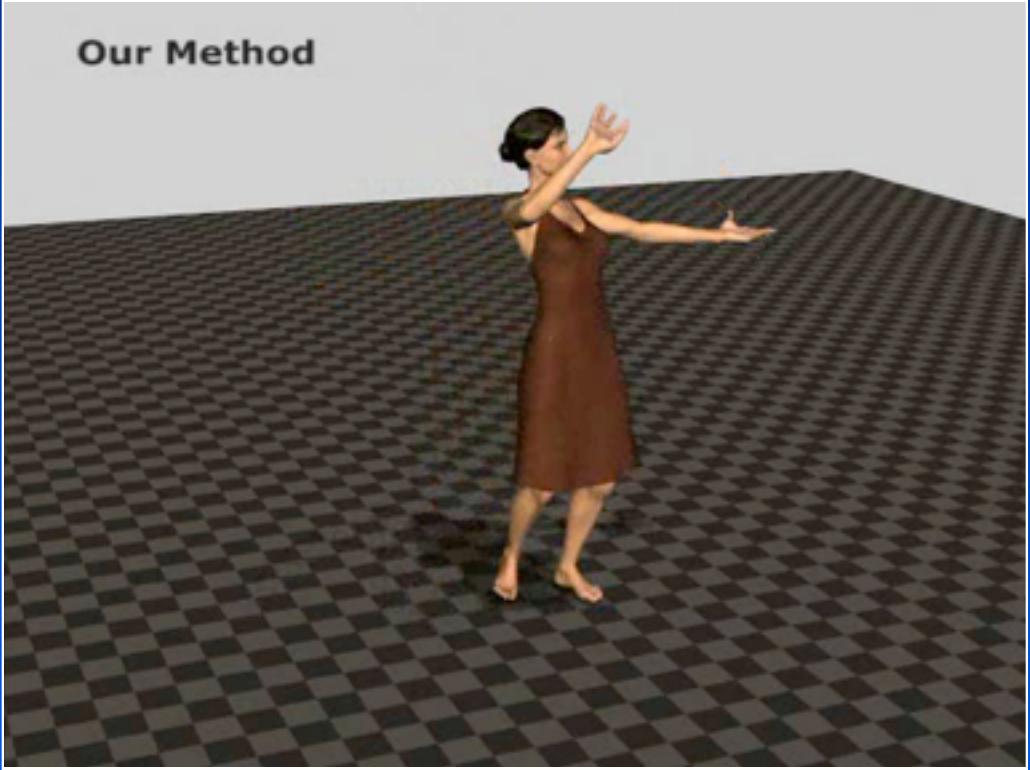
Intelligent Virtual Characters



Secondary issues

Skinning

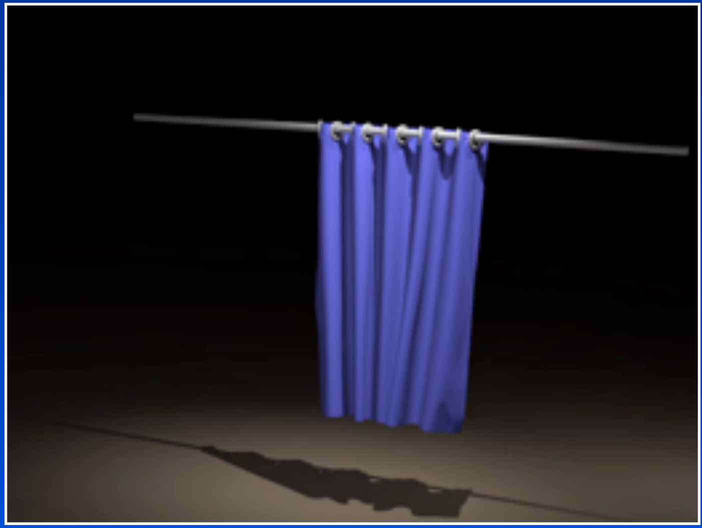
- From skeletons, to fully fleshed models



Secondary issues

Secondary Motions

- Cloth, hair, soft tissue motion, breathing



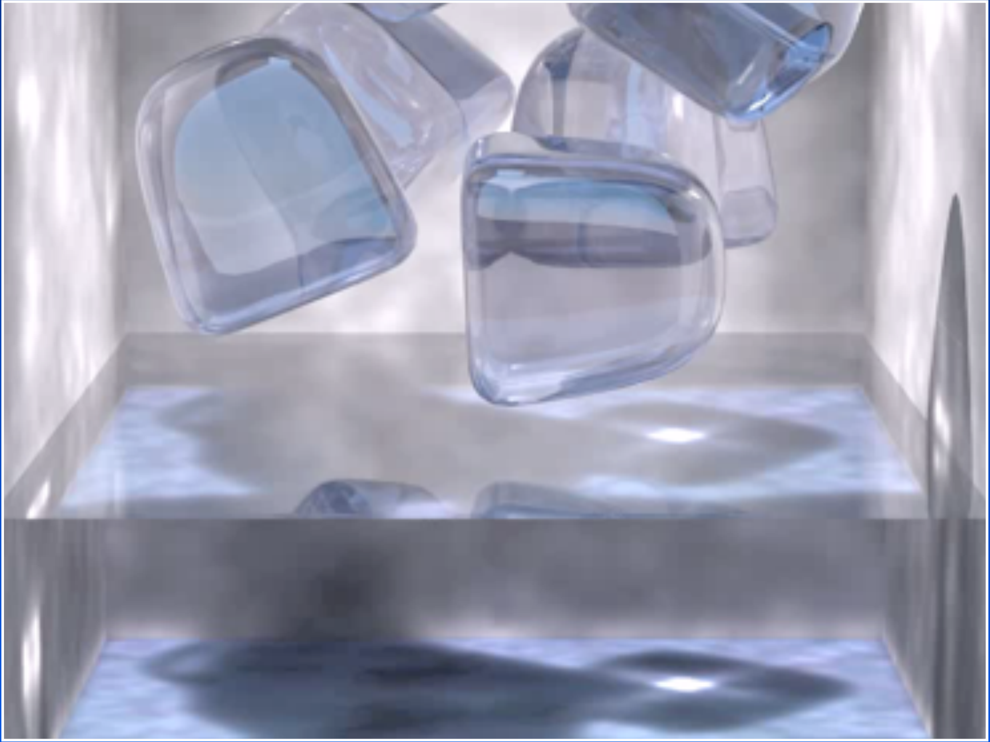
Courtesy of Eftychios Sifakis

Secondary issues

Rendering

- Skin, wrinkles, sweat, blushing etc

Environment

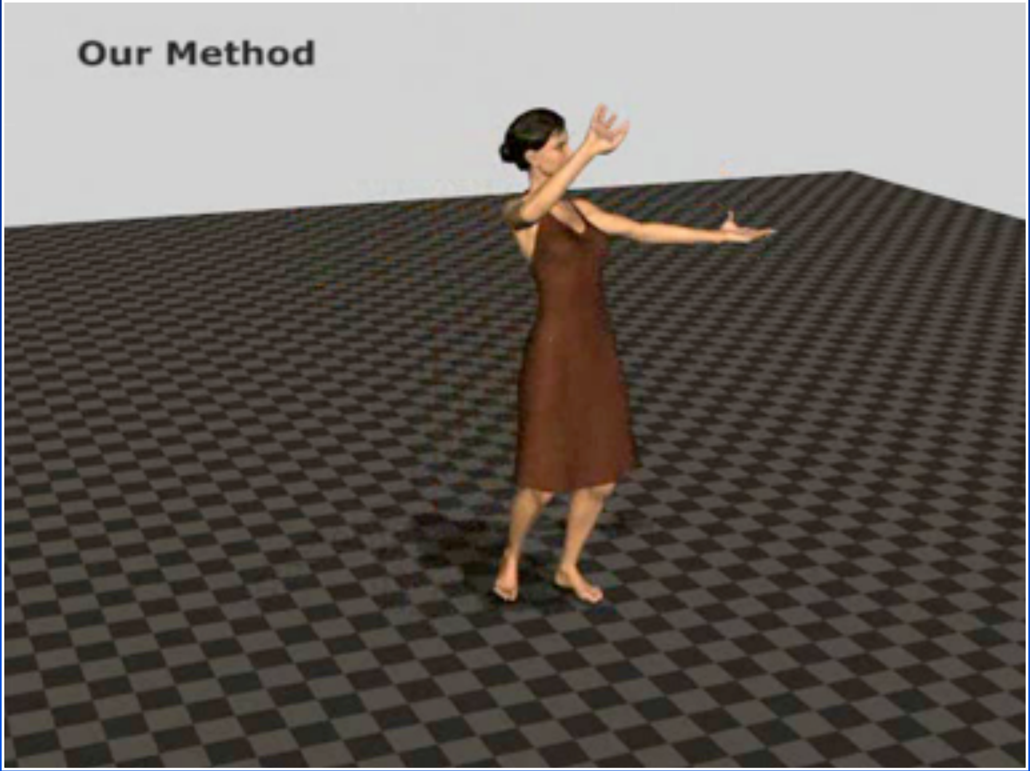


Courtesy of Fedkiew's group

Sound rendering

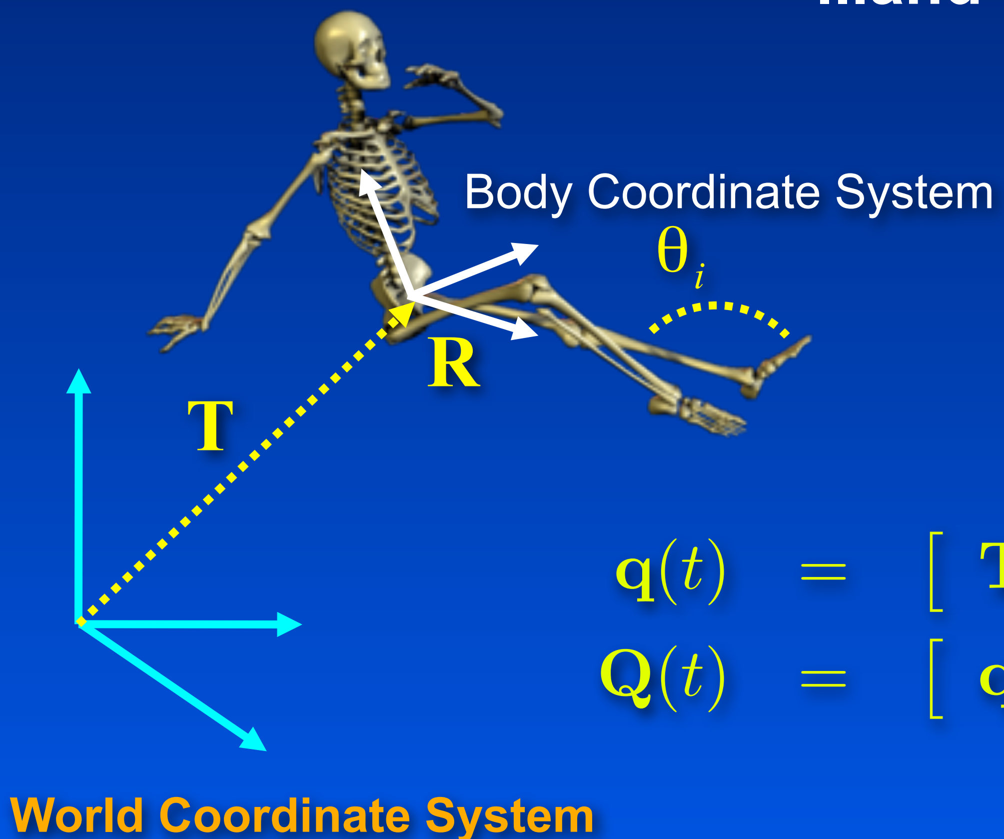
…

Motion Synthesis for Virtual Characters



Modeling: Virtual Character

...and why we love math!



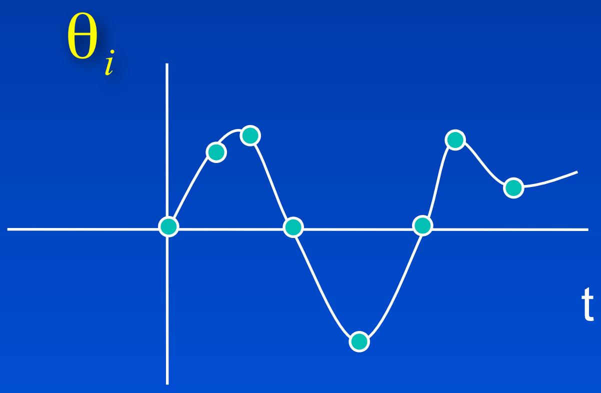
$$\mathbf{q}(t) = \begin{bmatrix} \mathbf{T}(t) & \mathbf{R}(t) & \theta_1 & \cdots & \theta_n \end{bmatrix}$$

$$\mathbf{Q}(t) = \begin{bmatrix} \mathbf{q}(t) & \dot{\mathbf{q}}(t) \end{bmatrix}$$

Motion Synthesis: By hand?

Classic Kinematic Approach

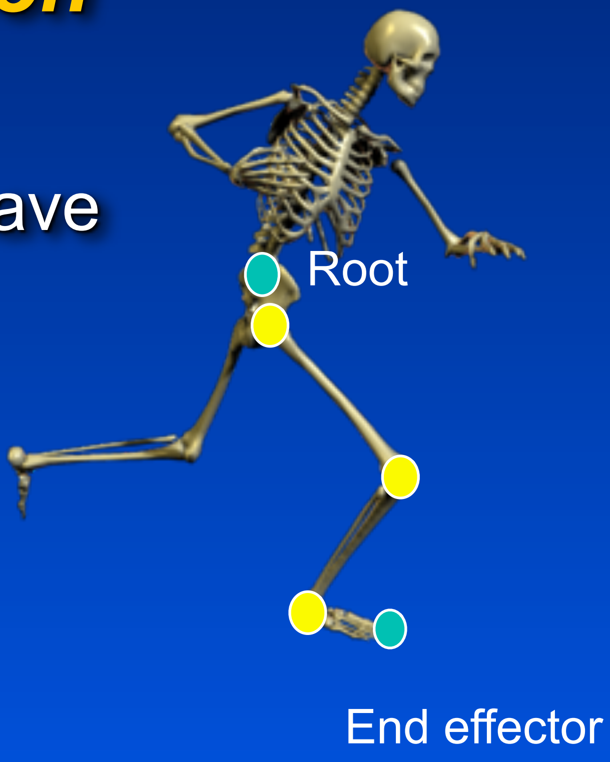
- Keyframes interpolation



More automation: Inverse kinematics

Constrained optimization

- Least squares
- $X = f(q)$ we would like to have $q = f^{-1}(X)$

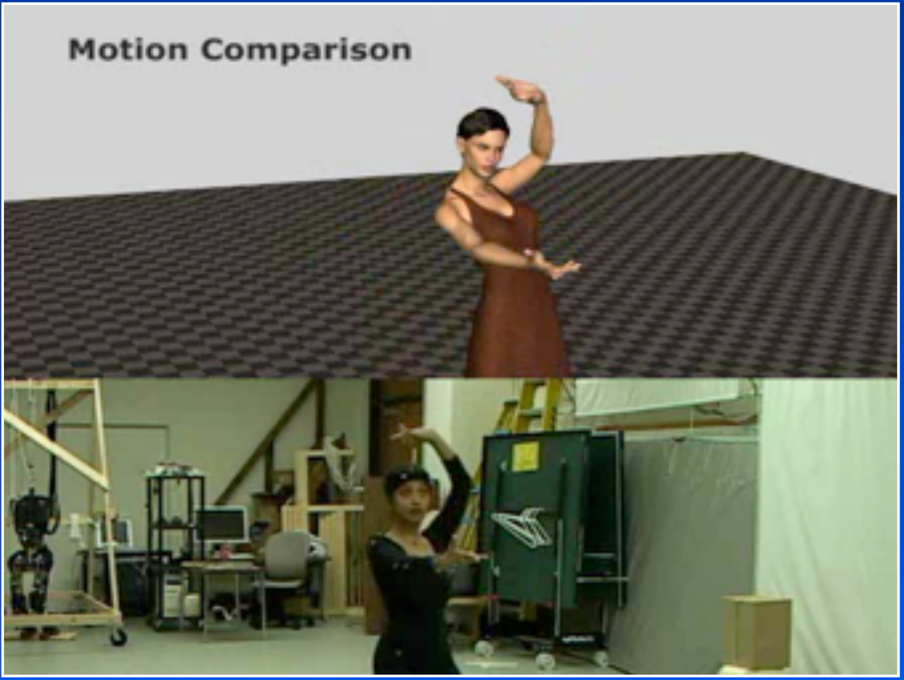


By example: Motion Capture

Record live action → Apply on virtual character

- DOFs q(t) = f(markers(t))

Why not motion capture everything?



By example: Motion Capture

Record live action → Apply on virtual character

- DOFs q(t) = f(markers(t))

Why not motion capture everything?

Problems

- Large databases required
- Applying motions to different characters and environments is tricky
- Interaction

Editing Mocap: Motion Planning

[with Ari Shapiro and Marcelo Kallmann, UC Merced][I3D07]

Given a moving character, compute collision-free motions for the limbs

- Moving obstacles
- Moving target

Key ideas

- Randomized planner (RRT)
- Include time in the search space
- Efficient configuration sampling

Applications

- Motion Correction
- Grasping motions

Original

Automatically Corrected

Editing Mocap: Motion Planning

[with Ari Shapiro and Marcelo Kallmann, UC Merced][I3D07]

Given a moving character, compute collision-free motions for the limbs

- Moving obstacles
- Moving target

Key ideas

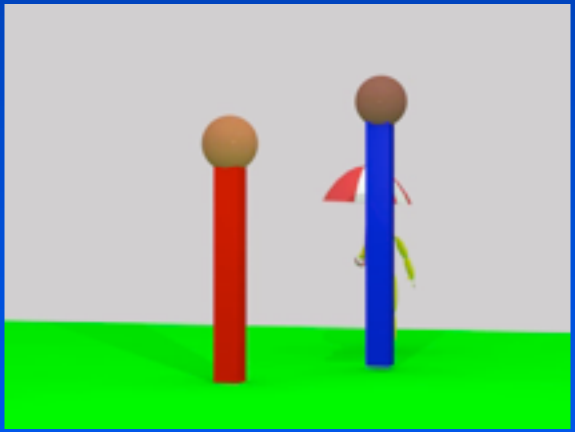
- Randomized planner (RRT)
- Include time in the search space
- Efficient configuration sampling

Applications

- Motion Correction
- Grasping motions

Original

Automatically Corrected



Editing Mocap: Motion Planning

[with Ari Shapiro and Marcelo Kallmann, UC Merced][I3D07]

Given a moving character, compute collision-free motions for the limbs

- Moving obstacles
- Moving target

Key ideas

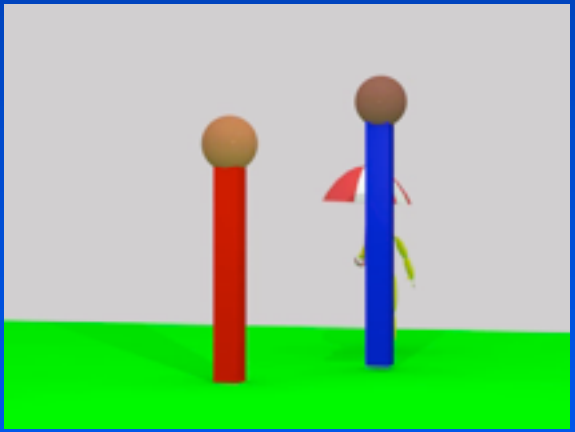
- Randomized planner (RRT)
- Include time in the search space
- Efficient configuration sampling

Applications

- Motion Correction
- Grasping motions

Original

Automatically Corrected



Stealing a hat

Original motions

A) Hat set as a "pick up" target

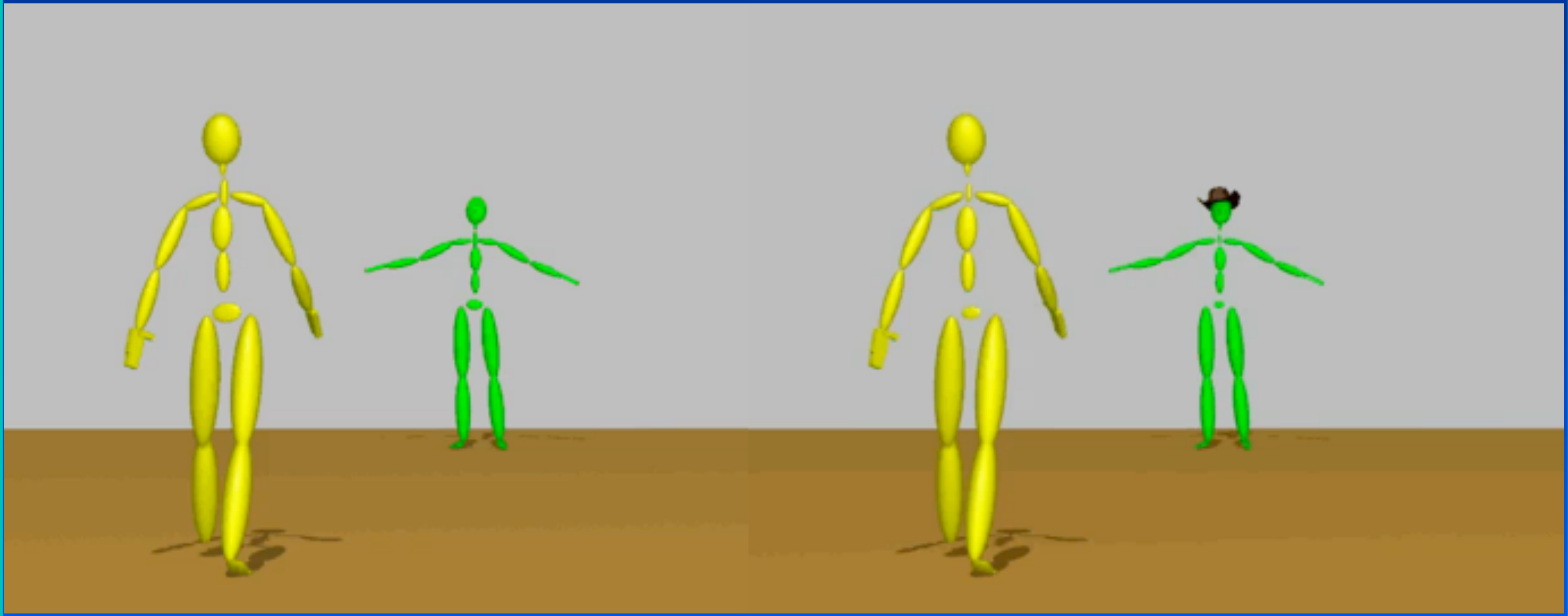
B) Hat is released on "head" target

Stealing a hat

Original motions

A) Hat set as a “pick up” target

B) Hat is released on “head” target

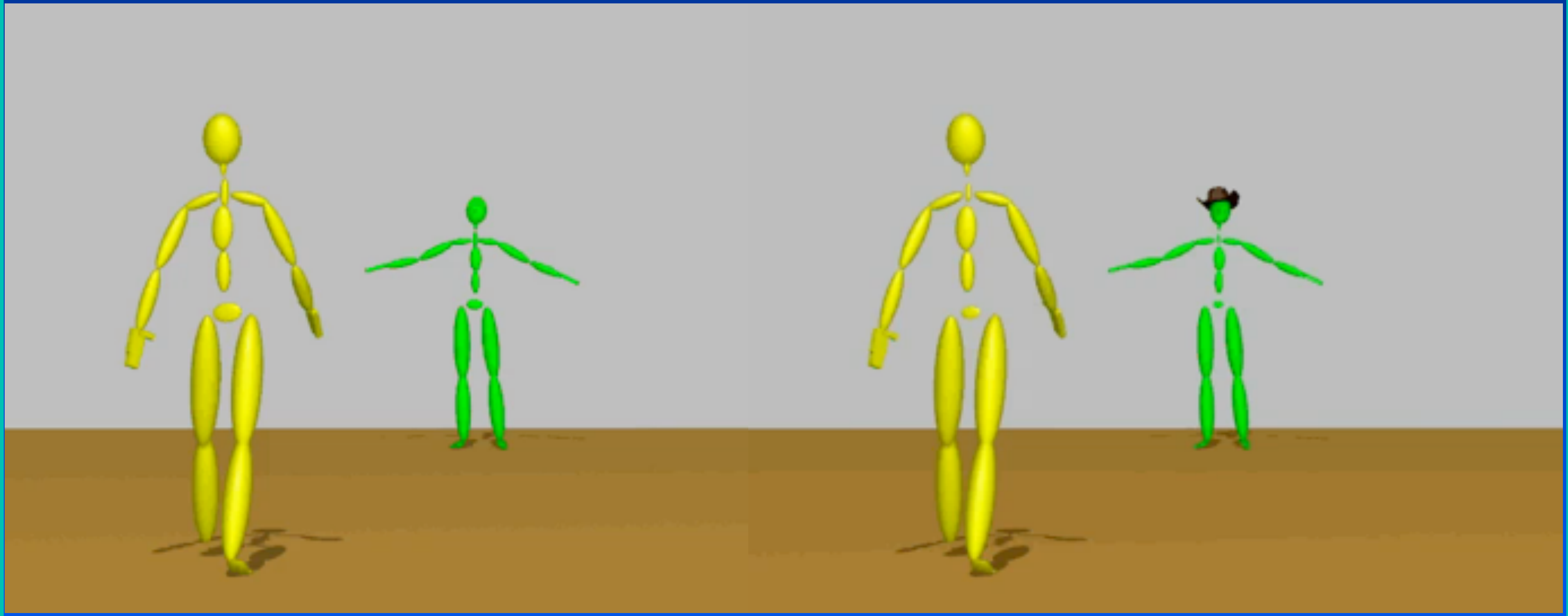


Stealing a hat

Original motions

A) Hat set as a "pick up" target

B) Hat is released on "head" target



Flipping with Physics!

[with Majkowska, Zordan] [SCA 07]

Physics-based replication of ballistic motions

- Single back flip to double back flip
- Do not try this at home!
- ….seriously.

Flipping with Physics!

[with Majkowska, Zordan] [SCA 07]

Physics-based replication of ballistic motions

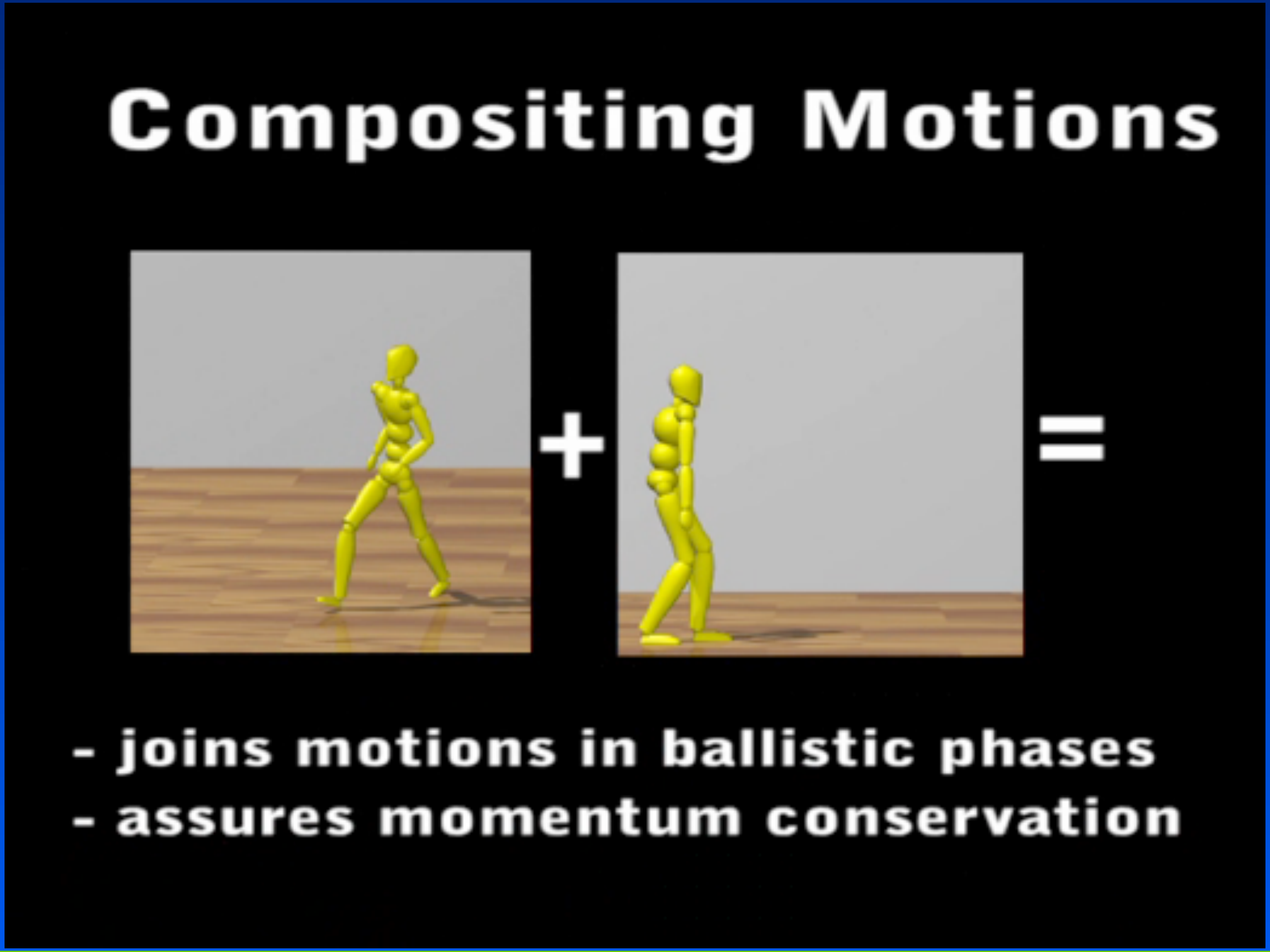
- Single back flip to double back flip
- Do not try this at home!
- ….seriously.

Flipping with Physics!

[with Majkowska, Zordan] [SCA 07]

Flipping with Physics!

[with Majkowska, Zordan] [SCA 07]



Flipping with Physics!

[with Majkowska, Zordan] [SCA 07]

Summary of the method

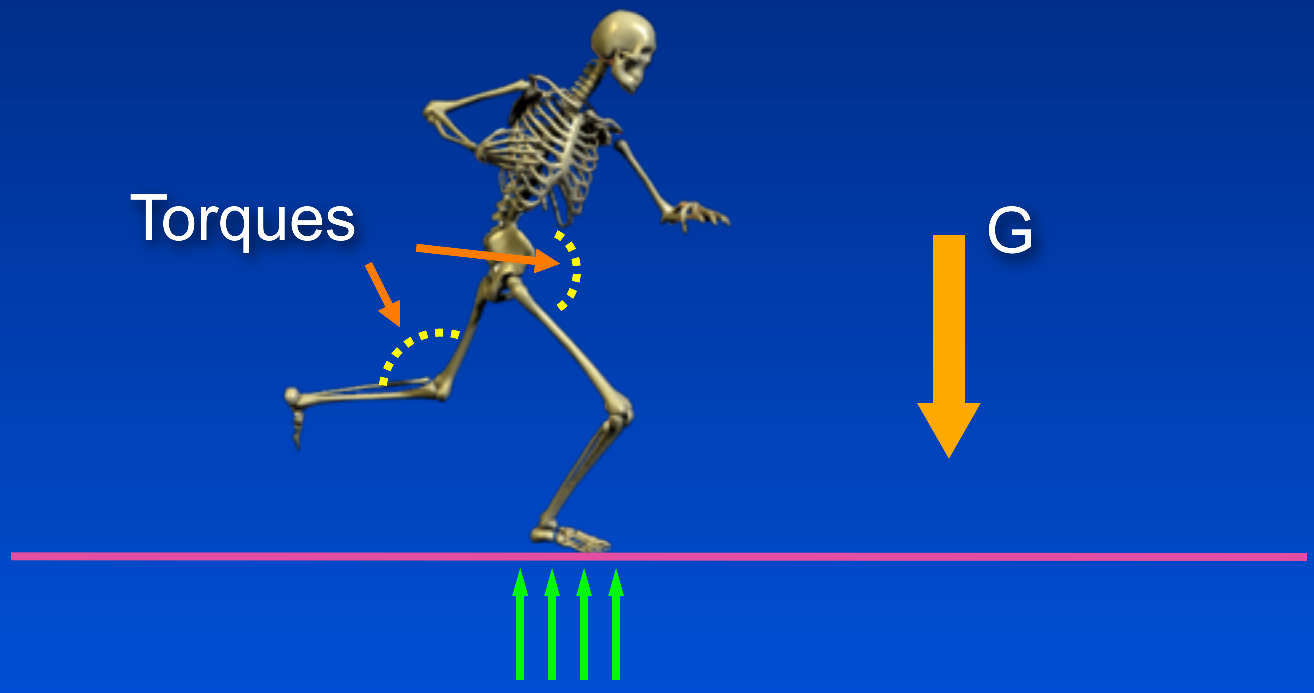
- Estimate mass and momenta from the motion.
- Search the data to find a looping sequence.
- Rotate the looping sequence around the axis of angular momentum as needed.
- Adjust the trajectory of the center of mass to preserve linear momentum.
- Retime the take-off phase.

Motion Synthesis: Physics-based Simulation

Reality:

physics + control

Physics



Applying Newtonian Mechanics

Equations of motion

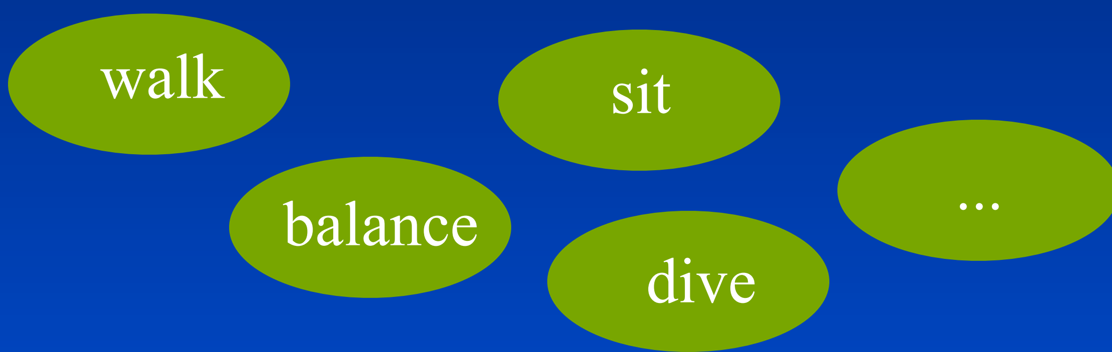
$$\mathrm{M}(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = \sum_i \mathbf{J}_T^T \mathbf{f}_i + \sum_j \mathbf{J}_R^T \tau_{ext,j} + \sum_k \mathbf{J}_R^T \tau_{int,k}$$

Inertial Forces | Gyroscopic Forces | External Forces | External Torques | Joint Torques

$[\mathbf{f}_i \ \tau_{ext,j} \ \tau_{int,k}]$ →  → State $[\mathbf{q}\ \dot{\mathbf{q}}]$

Problem: Control

Divide and conquer



What is a “Controller”?

An algorithm that computes torques at the joints of a character to produce a desired motion

Input: jump 0.5m → controller →

→ New state: jump!

Torques: $\mathbf{F}_t$ → $\mathbf{F} = \mathbf{M}\ddot{\mathbf{q}}$

Questions

How do we design controllers?

How do we switch between controllers?

Designing Controllers

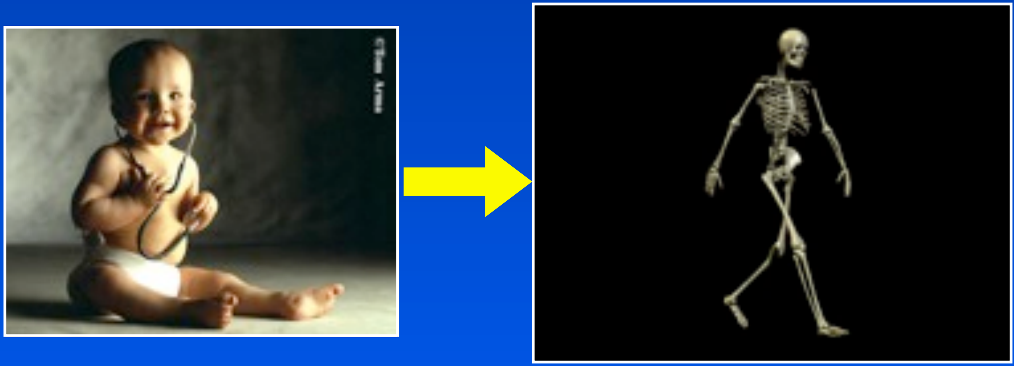
Challenges

- The human body has many degrees of freedom
- Our motions are dynamic (unstable) and/or highly optimized
- Thus, the control space is large with many local optima
- Natural look is difficult to describe mathematically

Learning?

- How do babies learn to move?

Optimal Control

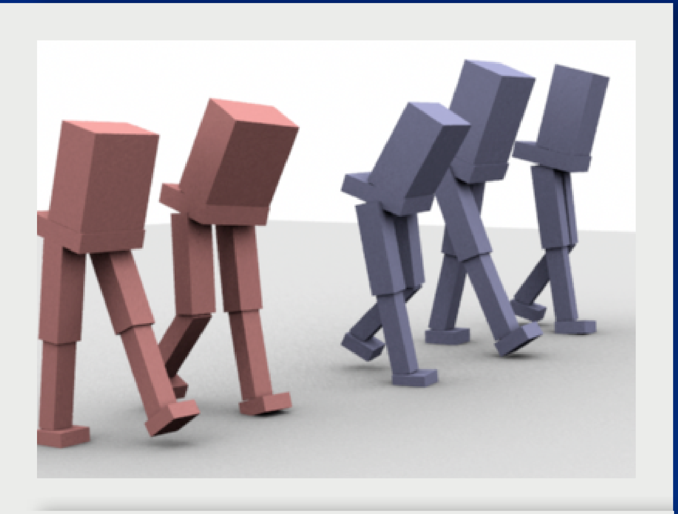


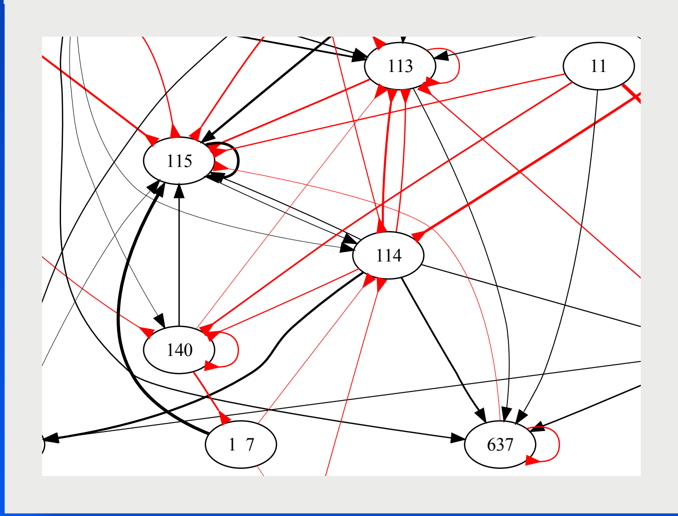
Evolving Physics-based Controllers for bipeds [with Brian Allen]

Goal: ***Blackbox controller generator***

Control structure: ***Neural Network***

Evolution through a genetic approach





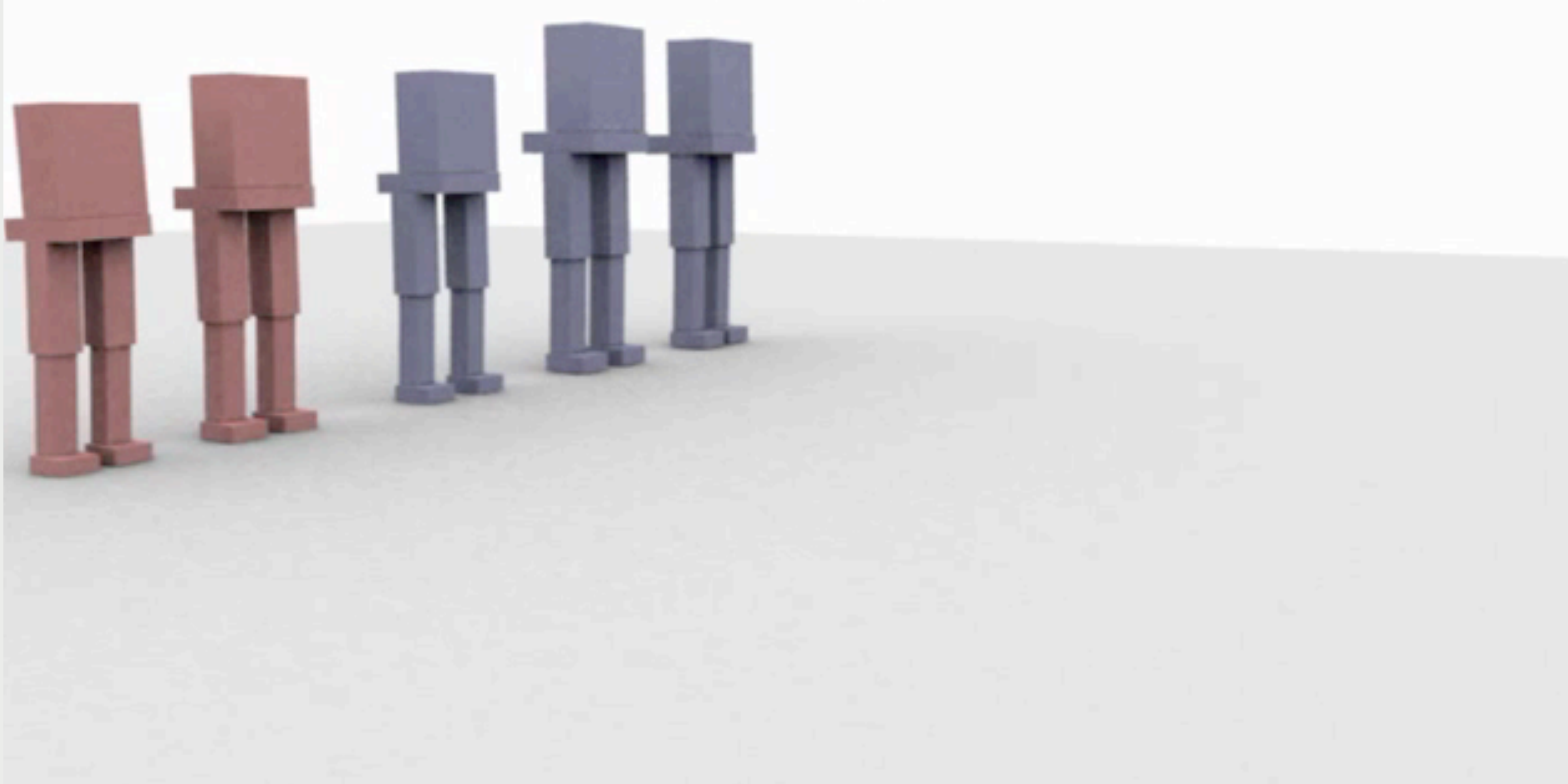
Key innovations

- Evolve the network topology and the muscle parameters
- Use no prior knowledge (patterns etc)
- Use simple fitness function
- Introduce NEAT to graphics
  - *Historical markers*
  - *Speciation*

Humanoids of average male (blue) and female (red) height, weight and hip width.

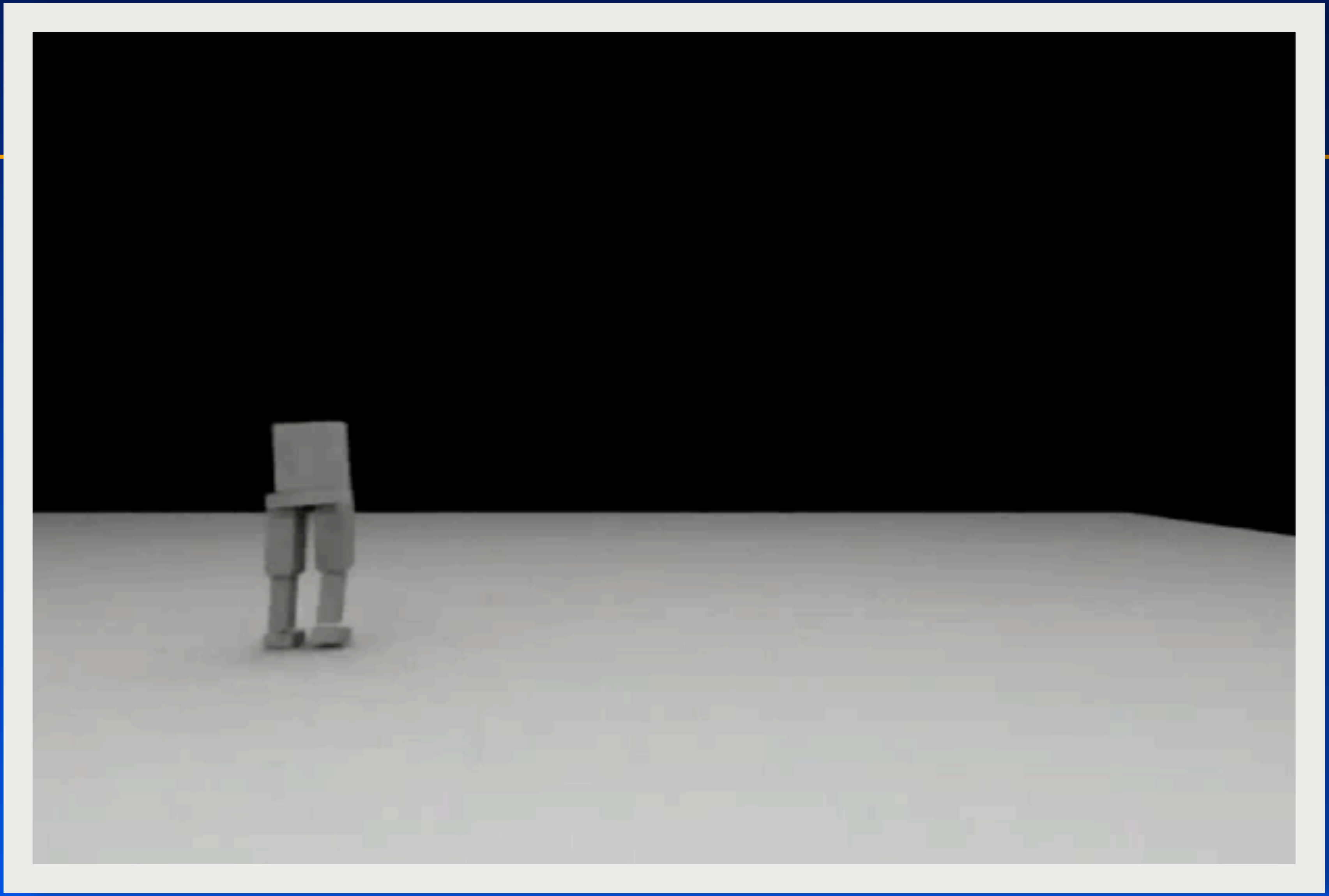
Humanoids spanning one standard-deviation of male (blue) and female (red) height and weight.

Humanoids spanning one standard-deviation of male (blue) and female (red) height and weight.



"The Ministry of Silly Walks"

(with apologies to Monty Python)



"The Ministry of Silly Walks"

(with apologies to Monty Python)

Biomechanical modeling of the head and neck

Sung Hee Lee and Demetri Terzopoulos

Muscle modeling

Control

Tension control

Heads Up!
Biomechanical Modeling and Neuromuscular Control of the Neck

Sung-Hee Lee   Demetri Terzopoulos
University of California, Los Angeles

SIGGRAPH2006

Composing Controllers

[SIGGRAPH 2001] [C&G 2001 (Best Paper Award)]

Goal:

Create physics-based characters that can react to their environment and to user interaction

Results

Falling and getting up

- 5 Controllers:
  - *Default*
  - *Fall*
  - *Roll over*
  - *Get up*
  - *Balance*

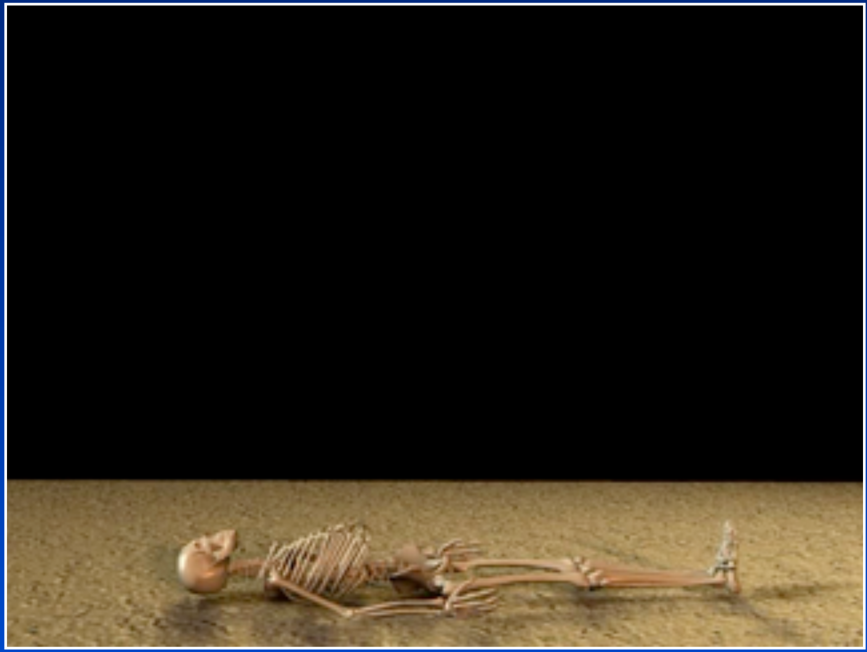
Multiple solutions - Planning

Controllers
- Default
- SitUpGetUp
- Balance

Controllers
- Default
- Kip
- Balance

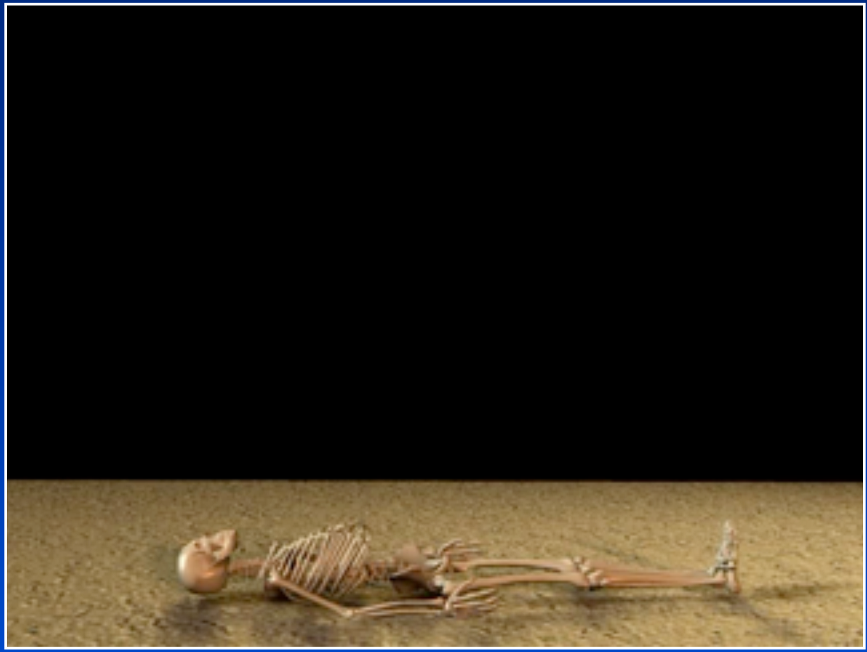
Multiple solutions - Planning

Controllers
- Default
- SitUpGetUp
- Balance



Controllers
- Default
- Kip
- Balance

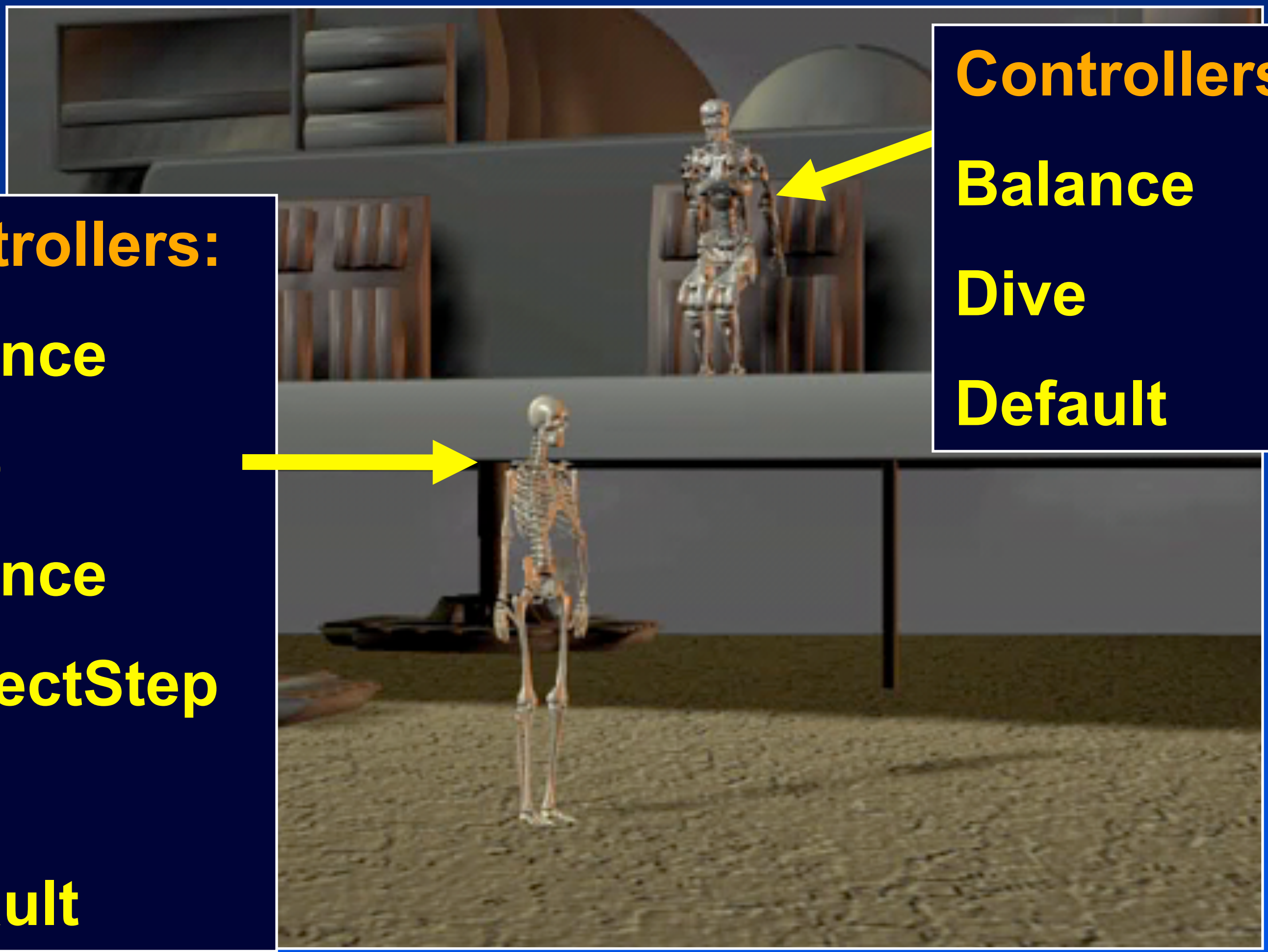
Multiple solutions - Planning



Controllers
- Default
- SitUpGetUp
- Balance

Controllers
- Default
- Kip
- Balance

Multiple Characters



Controllers:

- Balance
- Step
- Balance
- ProtectStep
- Fall
- Default

Controllers:

- Balance
- Dive
- Default

Multiple Characters

Hybrid Control with Ari Shapiro and Fred Pighin

[SIGGRAPH Tech Sketch 03, Pacific graphics 03]

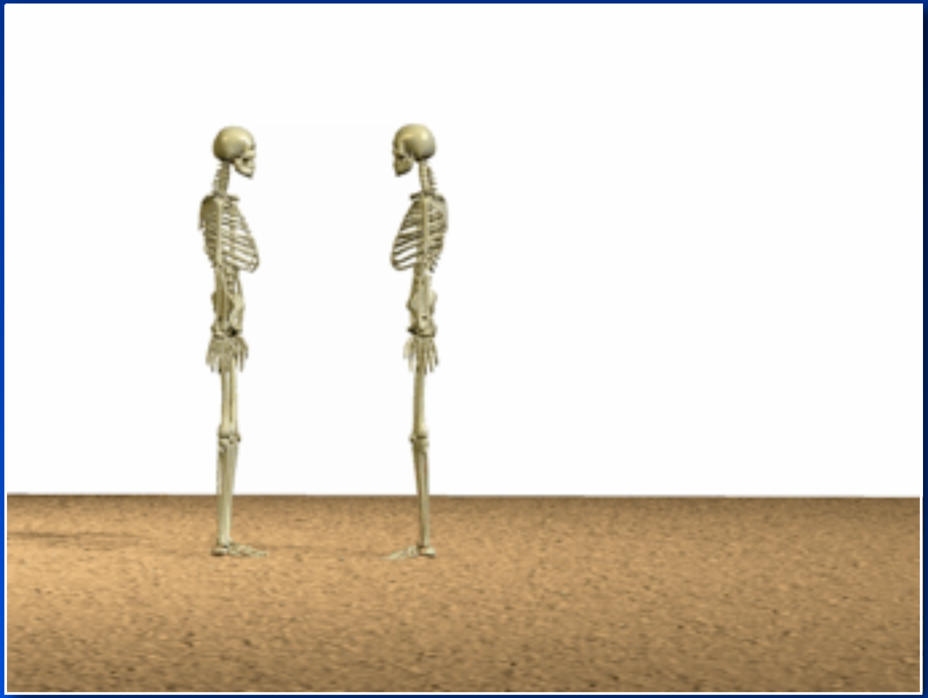
Combine kinematic and dynamic control

- Leverage advantages of both

Key idea

- Use motion data until there is interaction, then use physics

First demonstration of hybrid control



Kinematic Control

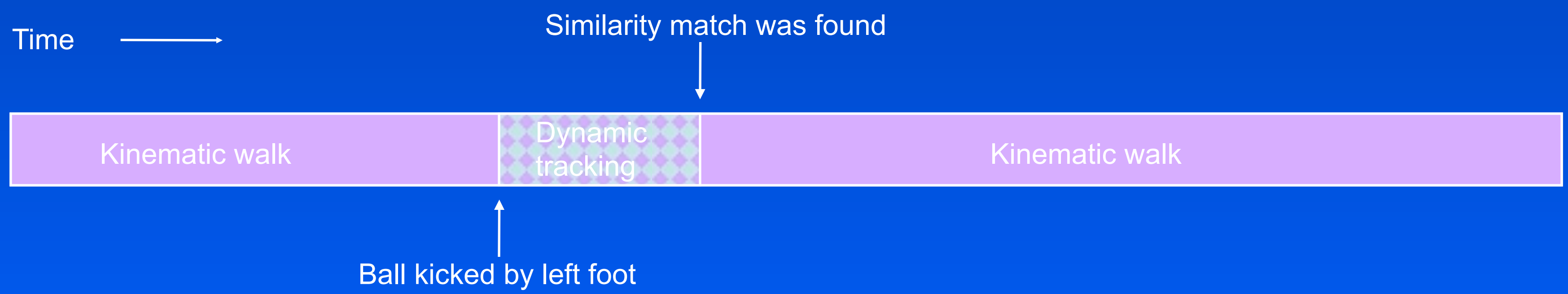
Motion captured walk

Time ⟶

Kinematic walk

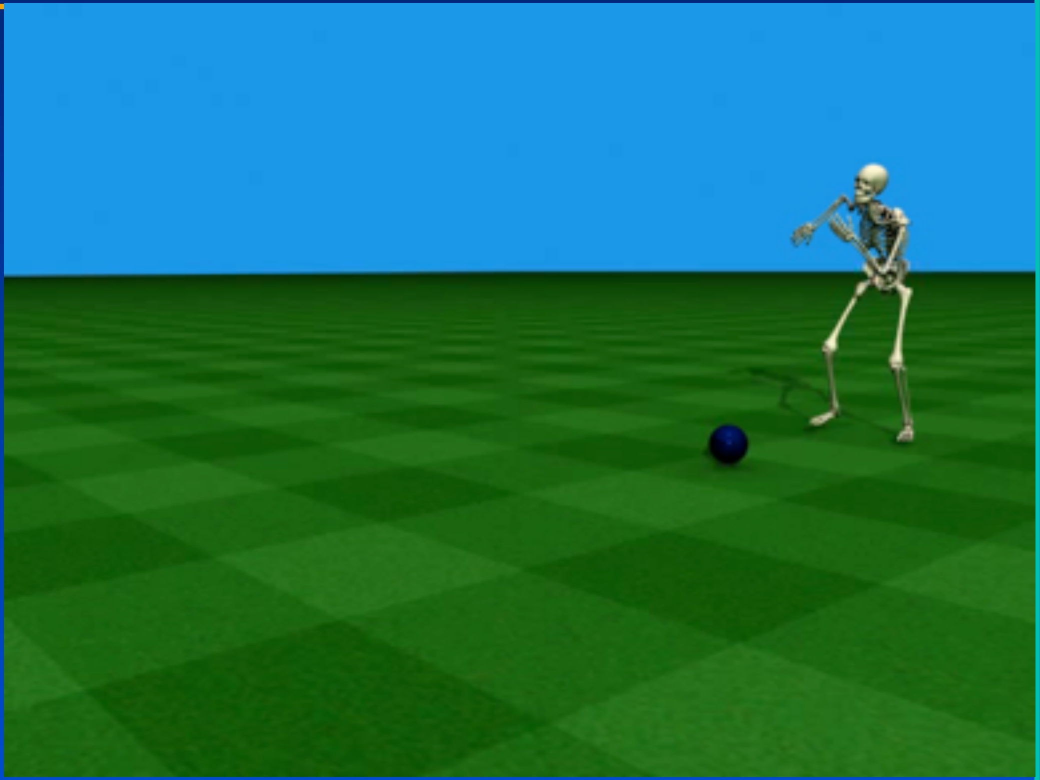
Dealing with a minor Disturbance

Dynamic ball placed in walk path



Major Disturbance: Switch To Dynamic Control

Heavy dynamic ball obstructing walk path



Time ⟶

Kinematic walk	Dynamic tracking	Flip and roll	Get Up

↑

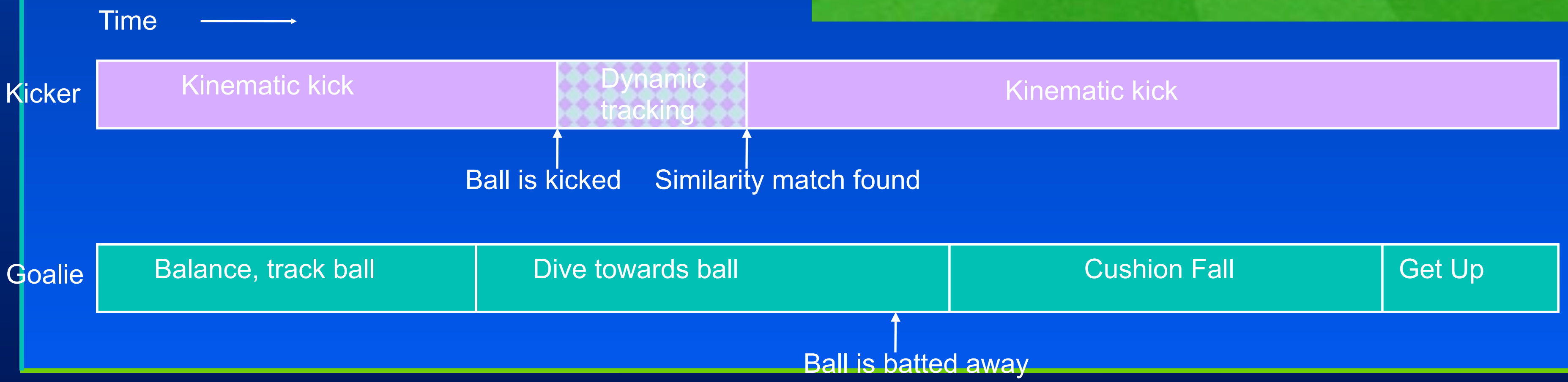
Ball caught on right foot

Multiple Characters and Objects

Kinematic kicker

Dynamic ball

Dynamic goalie



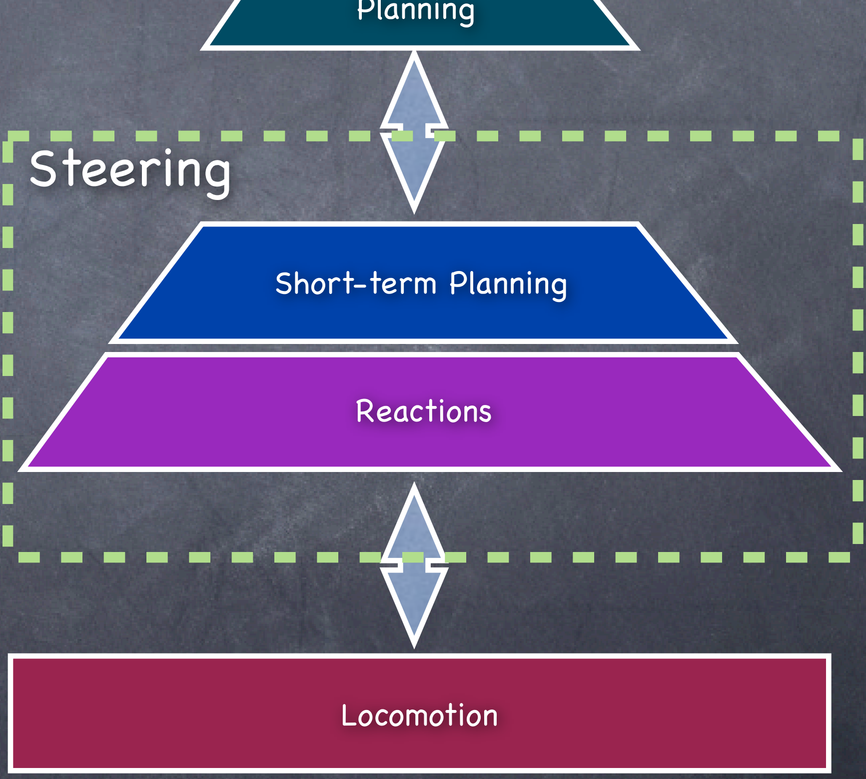
Crowd Simulation

(...and why we love AI and Algorithms!)

Steering:

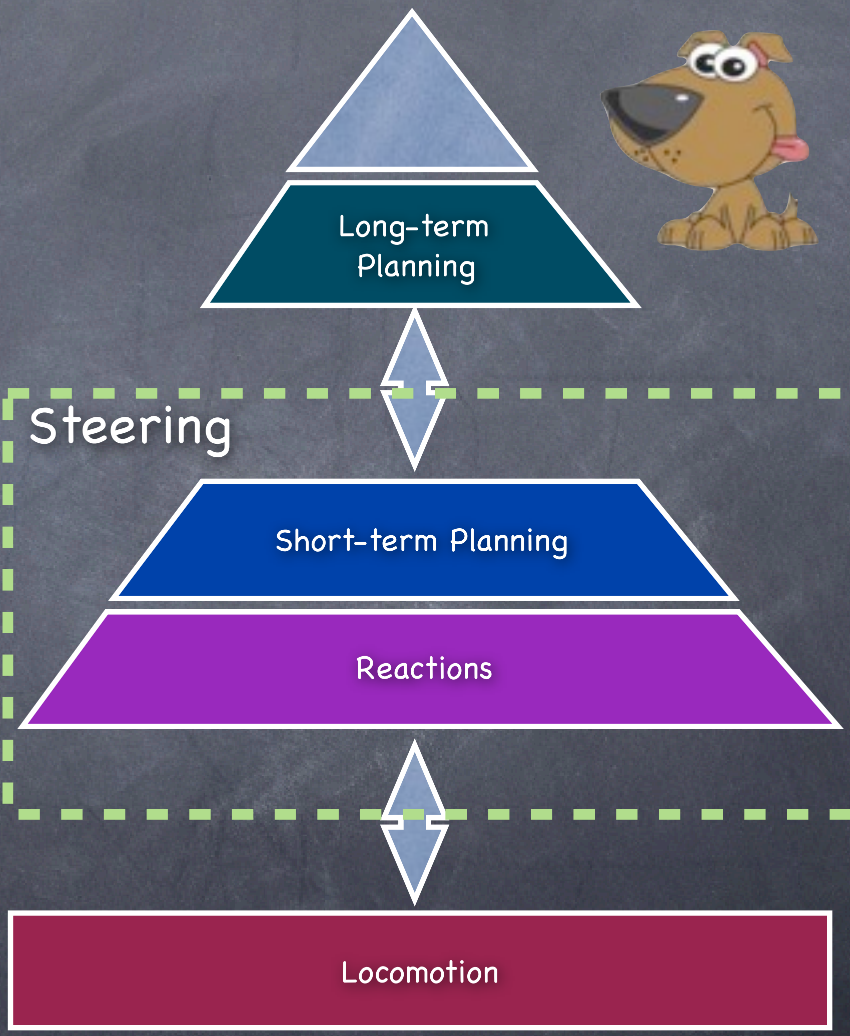
Agent moves from A to B at the presence of dynamic and static obstacles

Modeling the agent's steering decision process



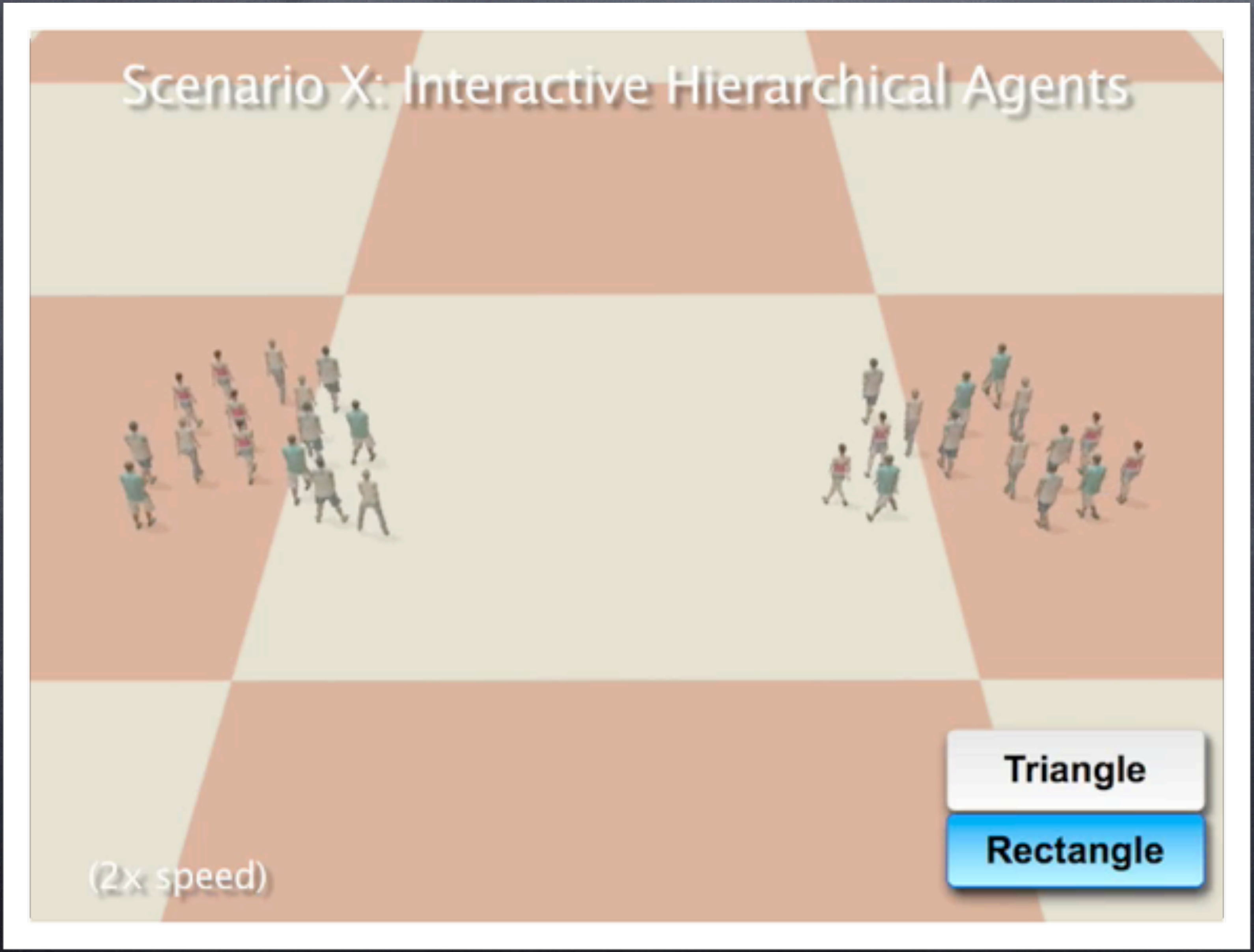
Why is it complex?

- Agent Individuality
- Agent coordination
    - Verbal and Non-verbal communication
- Social Etiquettes
- State and Context specific behaviors
- Limited sensory information
- Prediction based decisions
- Deadlock resolution
- Memory
- Locomotion constraints



Authoring Behaviors [Schuerman et al. CASA 2010]

Authoring Behaviors [Schuerman et al. CASA 2010]



Scaling Problem: e.g. 100K Agents

- Maintaining the world database
- Persistence of agents (levels of detail?)
- Querying the database
  - Nearest neighbors
  - Visible objects
- Animating the agents
  - Attention control
  - Locomotion
- Parallelization?

Don't computer games do it?
GTA IV

user controlled

Don't computer games do it?
GTA IV

Computer Games Cut Lots of Corners

- Model mostly homogeneous crowds
- Use levels of details (e.g. primary, secondary characters)
- Simulate only a few agents at a time
- Rely on physics for collision prevention
- Simplify and modularize the problem with as little communication as possible between modules
  - Particle disks representing humans
- Pre-compute as much as possible

Example of problems with simple sliding particles

Example of problems with simple sliding particles

Space-Time Planning of Footsteps

[CASA 10,CASA 11]

Footsteps with precise timing information

- Sufficiently detailed information to motion synthesis
- Dynamic collision bounds
- Efficient space-time planning satisfies user defined constraints and physical costs (effort)
- Heterogeneous agents

Locomotion model in steering

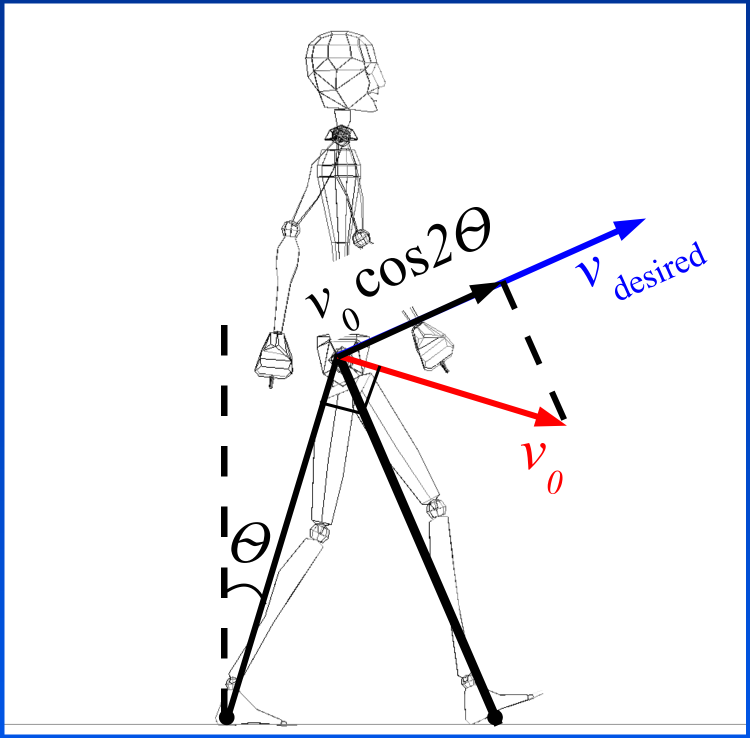
- Inverted pendulum with proper locomotion constraints

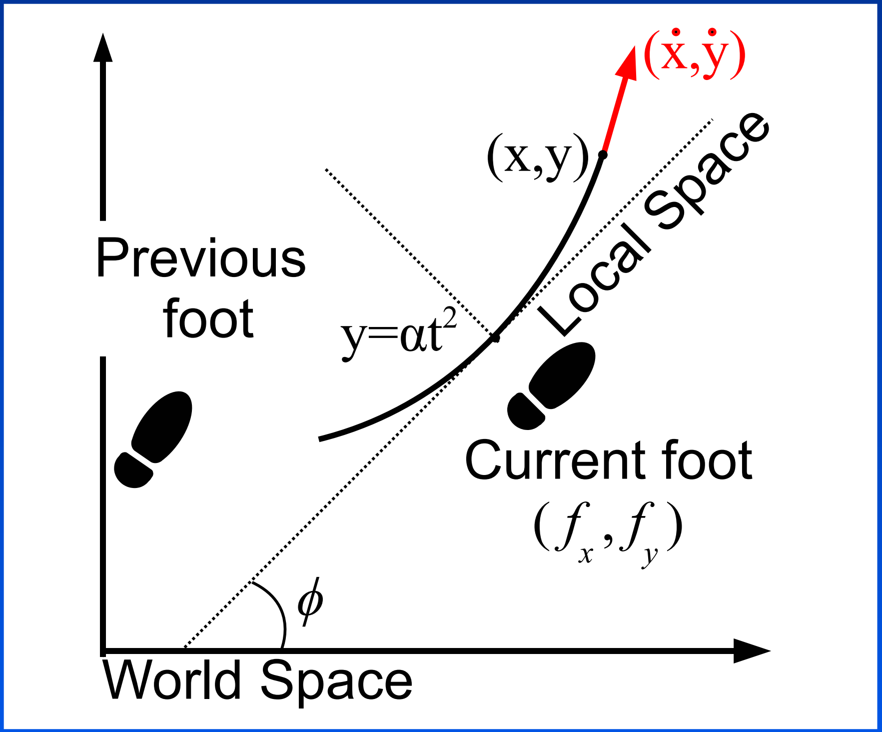
Motion synthesis has to follow footsteps rather than a velocity vector

Space-Time Planning of Footsteps
[CASA 10,CASA 11]

Our locomotion model in steering:

- 3D inverted Pendulum

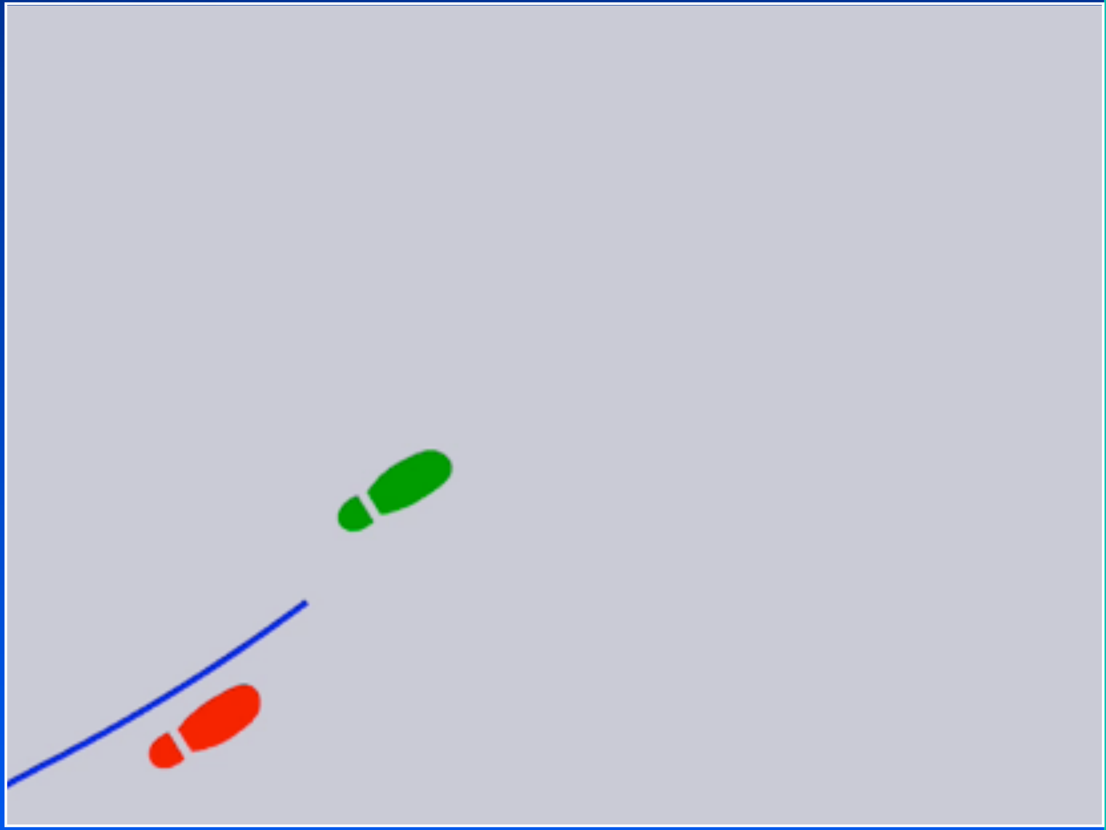




Space-Time Planning of Footsteps

Computing a sequence of steps 10 meters ahead

- The 3D inverted pendulum model defines the action space and the cost of action for a space-time planner



Space-Time Planning of Footsteps

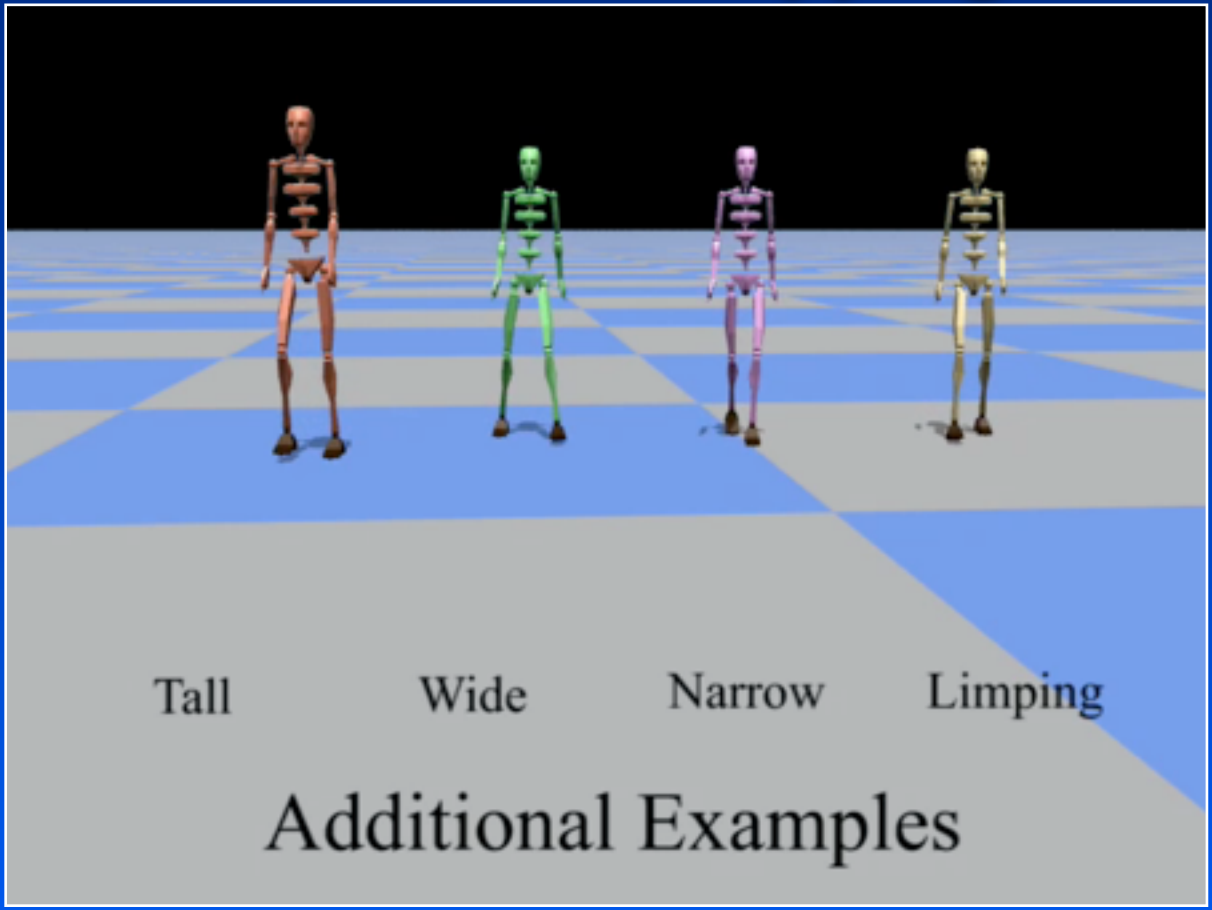
[CASA 10,CASA 11]

User constraints in the stepping model

Space-Time Planning of Footsteps

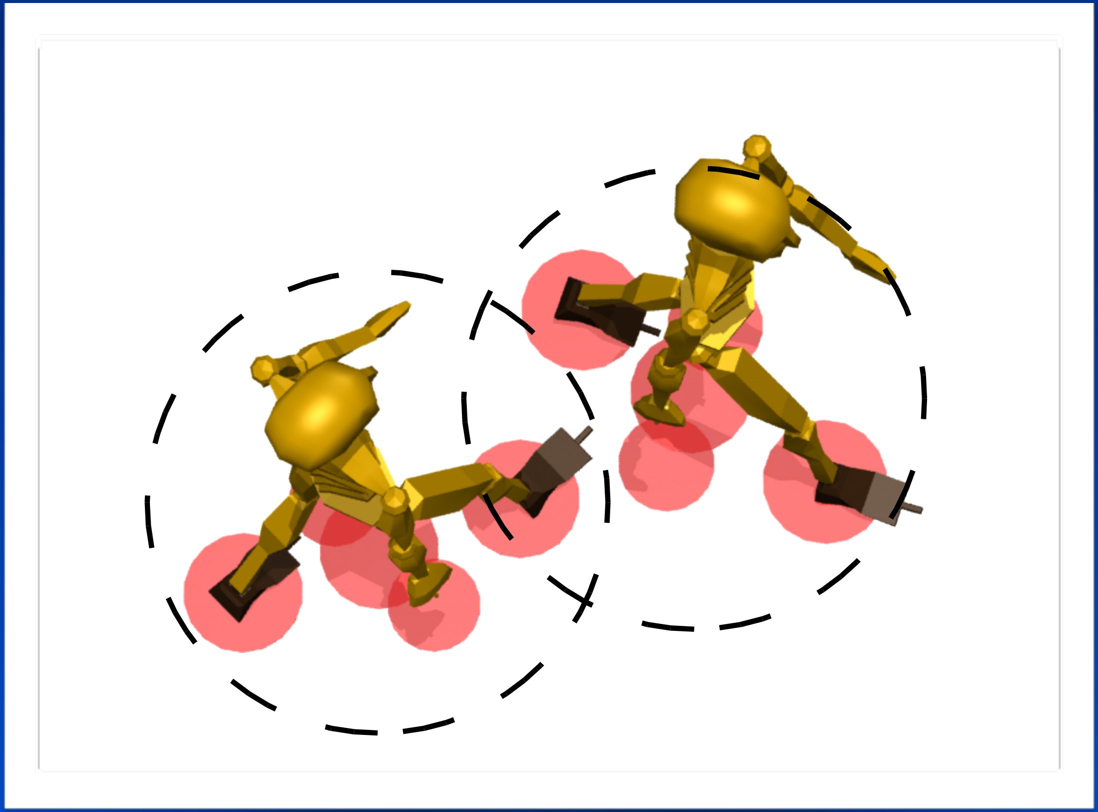
[CASA 10,CASA 11]

User constraints in the stepping model



Space-Time Planning of Footsteps

Time-varying collisions bounds

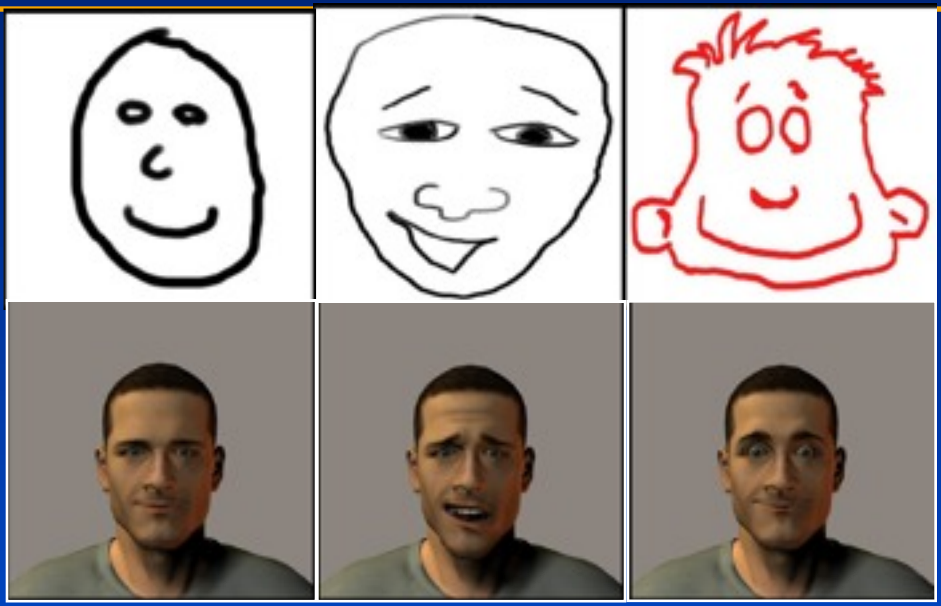


Space-Time Planning of Footsteps

[CASA 10,CASA 11]

Sketch-based Facial Animation

[with Gabriele Nataneli] [ISVC 07, IEEE CG&A 10]



Goal: From sketches to 3D faces

- Enable quick prototyping
- Make life easy for animators
- Support low-power mobile devices (e.g. iPhone)

Challenge: Sketches are arbitrary

Eye brow can be one stroke

... or multiple strokes

Sketch abstraction and representation is the key

- must be able to generalize sketch elements
- must be able to discriminate between elements

Robust sketch recognition

Main idea

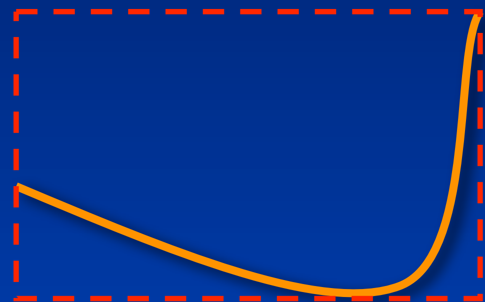
- Find a simpler problem that can be classified well
- Transform more complex problems into the simpler one

Contributions

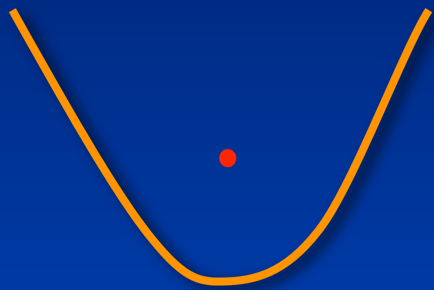
- A machine learning approach for clean sketch recognition
- A set of statistically well-behaved features for stroke abstraction
- A robust approach for grouping of strokes to get clean sketches

Abstraction of sketches through Shape Attributes

Individual



Bounding Box



Centroid

Relational

Horizontal Ordering

Depth

Three-step framework

Step I: Labeling -- Results: e.g. stroke1:left eyebrow, stroke2:mouth

- Segmentation
- Recognition
- Grouping

Step II: Match sketch elements to artist-provided face templates -- Results: e.g. stroke1: angry eyebrow, stroke2: open smile

- Distance based
  - *Average Hausdorff distance*
  - *Frechet distance*
- Attribute based (convexity, topology, bounding box)

Step III: Quantify the intention of each element -- Results: e.g. little smile, medium angry eyebrows

- Simple [0,1] value between pre-defined upper and lower bounds

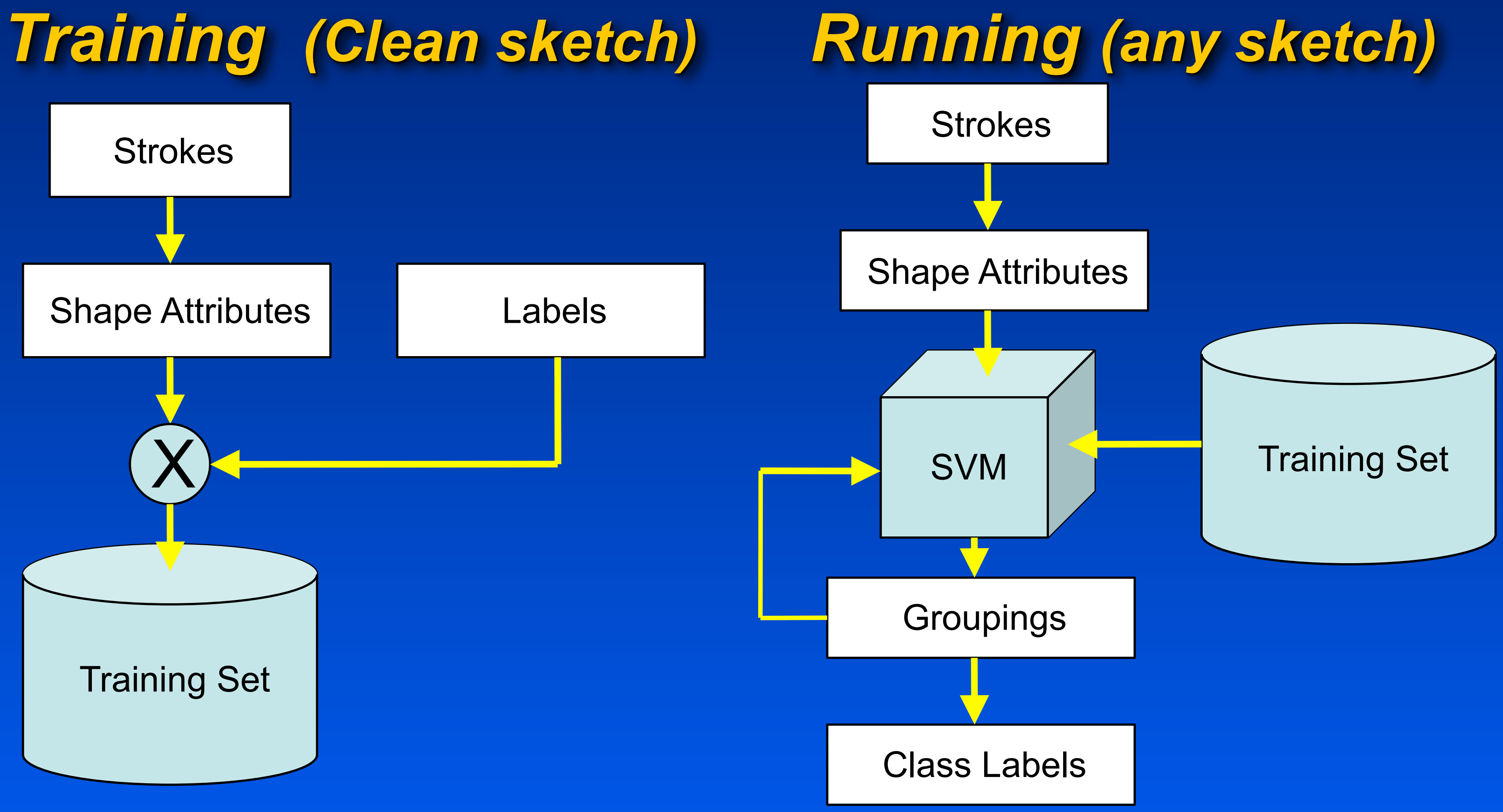
Set of Shape Attributes [Arnheim 74]

Shape attributes are the features for the SVM

Bounding Box Width
Bounding Box Height
Bounding Box Aspect Ratio
Centroid X
Centroid Y
Horizontal Ordering
Vertical Ordering
Overall Stroke Count
Depth

- Cross validation (93 % accuracy)

Step I: Labeling with SVM classifier



Grouping

General problem is NP-hard : Many choices

Formalism

Sketch: Set of strokes

$$S = \{S_1, S_2, \ldots, S_n\}$$

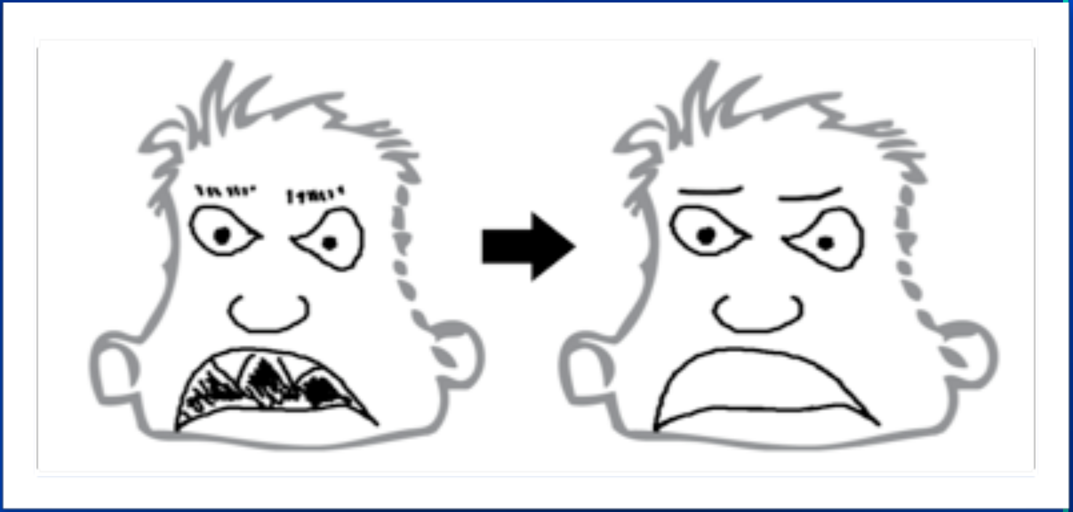
Grouping: set of groups

$$G = \{g_1, \ldots g_m\}$$

Such that

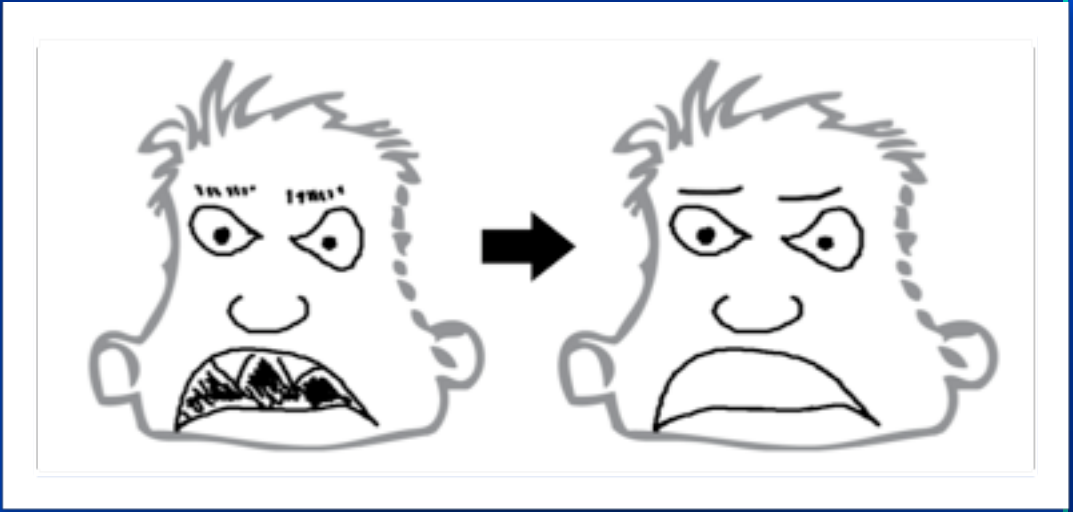
$$g_i \subset S$$

$$g_i \cap g_j = \emptyset \quad \text{for} \quad i \neq j$$



Grouping

General problem is NP-hard



Must prune

- Structural
- Overlap
- Semantic

Heuristic search

- Keep the one that produces the highest number of distinct strokes

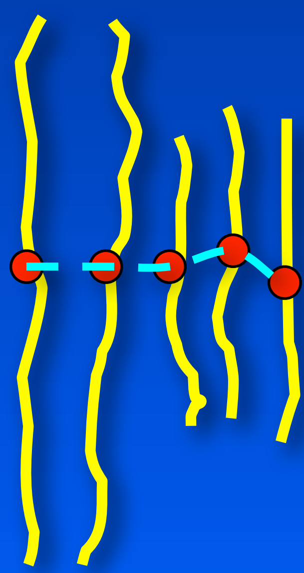
Structural grouping

Independent of training set

Similar to perceptual organizations

Primarily two kinds

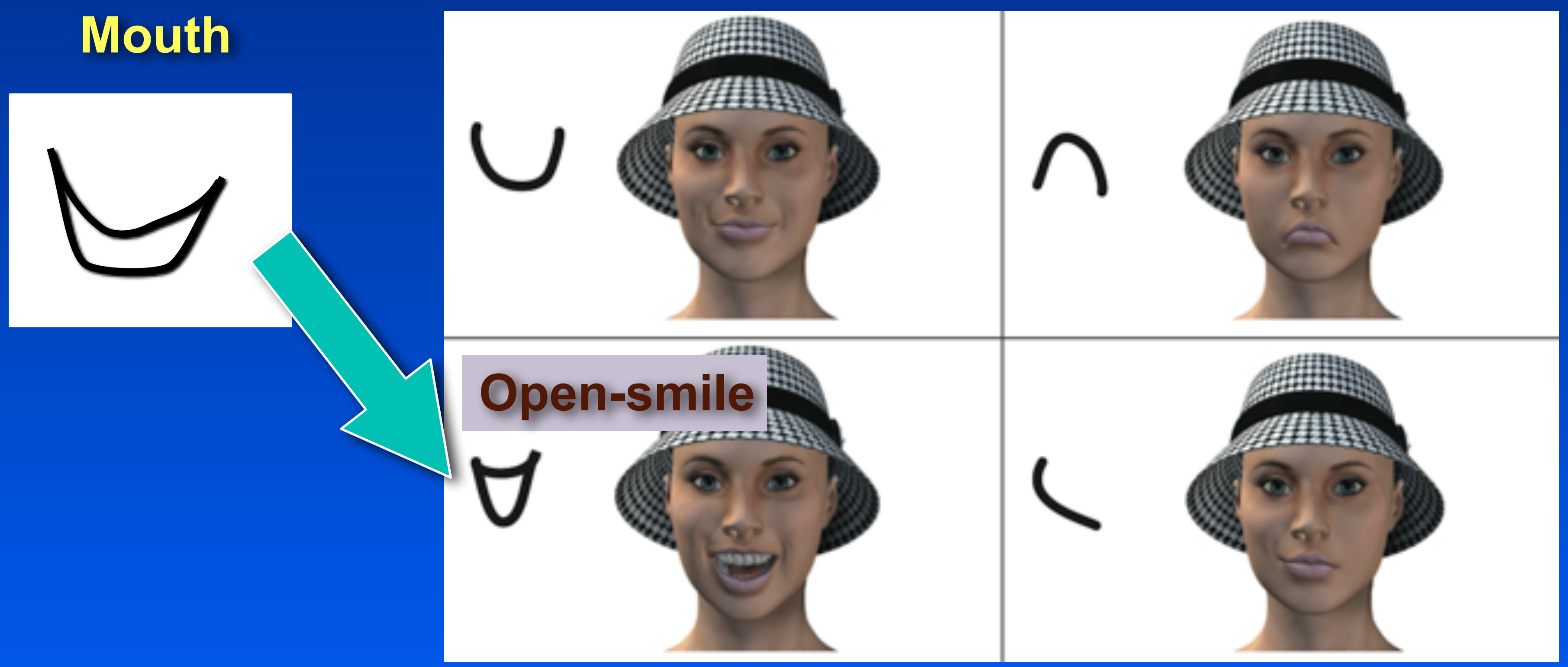
- Proximity



Continuity

Step II: Template matching

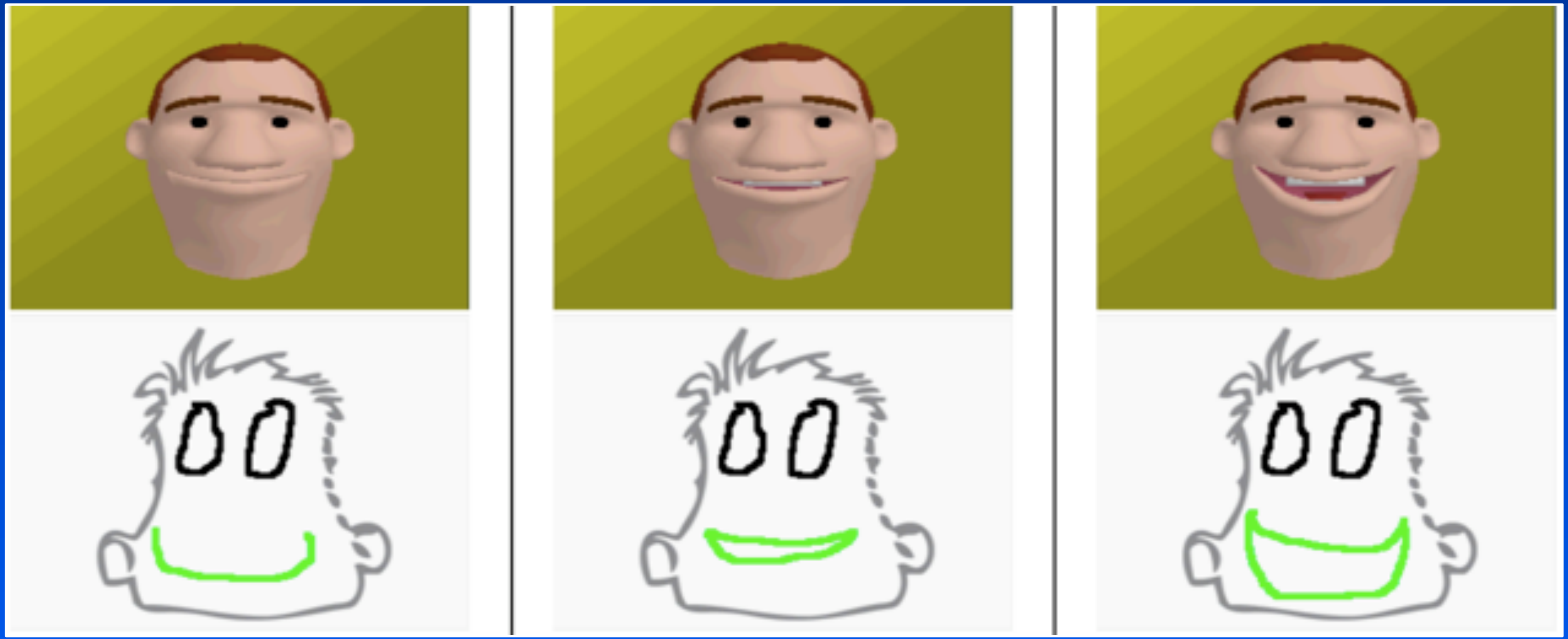
Labelled Stroke ***Template***



Step III: Refinement

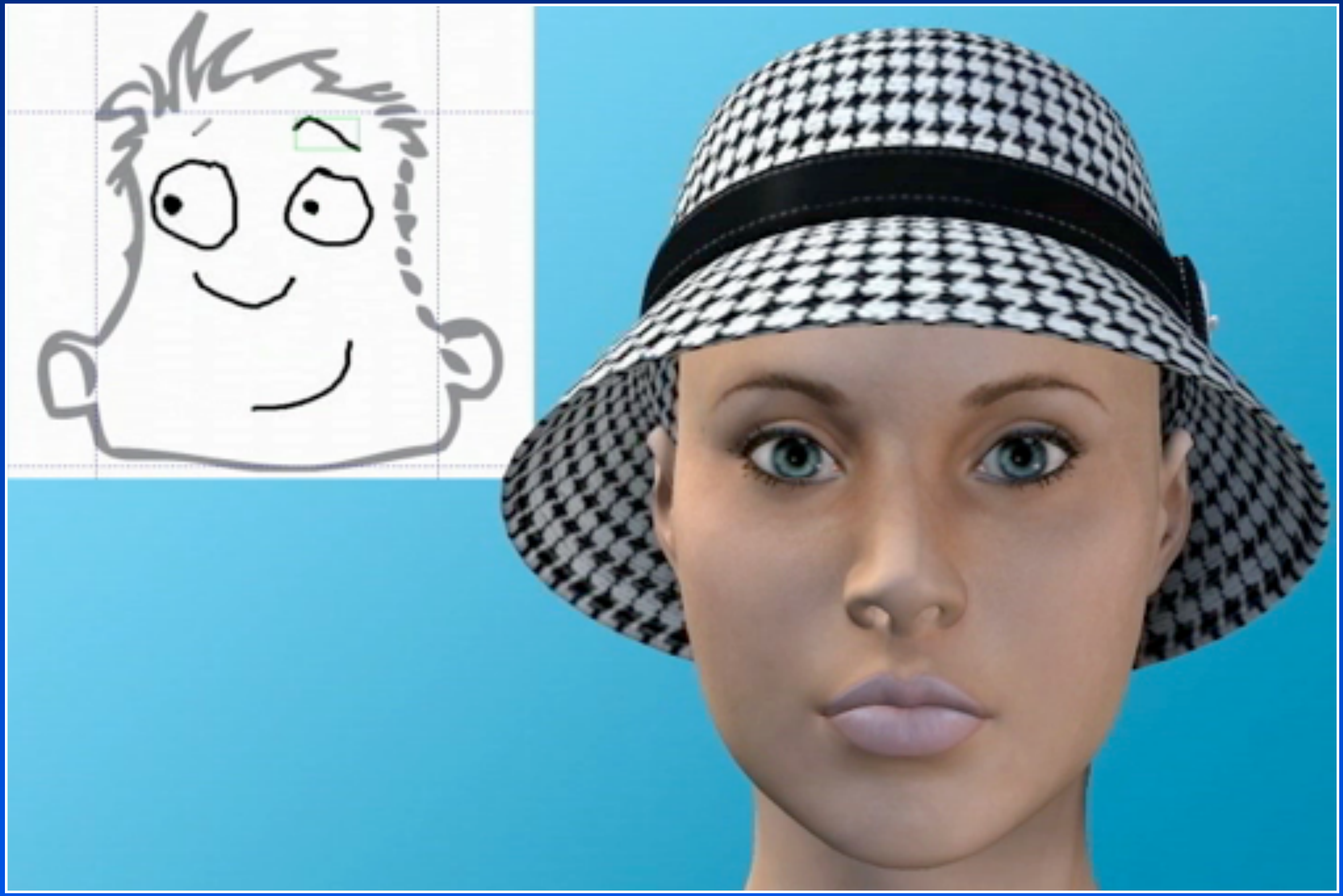
Identify quantitatively the intention of the stroke

- Upper bounds on shape attributes in the templates



Results

Results



Take home message

Everything you learn can lead to exciting careers

- R&D for special effects studio
- Technical Director for a special effects studio
- Computer Games Programmer
- Software Engineer