SGI-MEI
Sistemas Gráficos Interactivos
Animación de Personajes (2)

Motion synthesis, Crowds and Perception

Alejandro Beacco Porres

alejandro.beacco@upc.edu

Outline

- 1-Introduction
- 2-Motion synthesis
- 3-Crowds and Navigation
- 4-Integrating Animation and Simulation
- 5-Perception

1.INTRODUCTION

1.INTRODUCTION

1.INTRODUCTION

Uncharted 3 – Naughty Dog

Grand Theft Auto V – Rockstar Ga

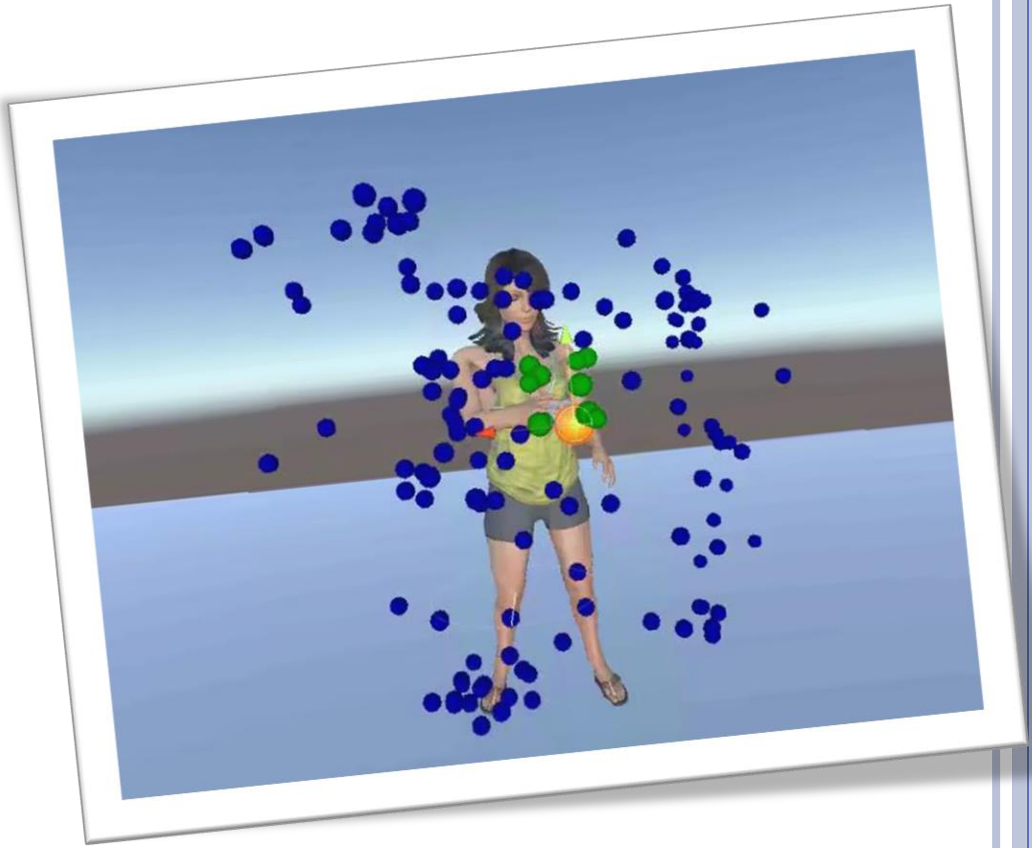
Pathfinder - Thu

1. Introduction

- We need to populate our virtual worlds with virtual characters
  - Characters need to live in this virtual world
  - Show behaviors
  - Do things
  - Move from one point to another
  - Navigation is necessary
- 1 character → 1 Agent
- Multiple Agents → Crowd
  - Efficient algorithms

2-MOTION SYNTHESIS

- 2.1-Techniques
- 2.2-Motion Graphs
- 2.3-Parametric Motion
- 2.4-Following Footsteps
- 2.5-Motion Matching



2.1-Techniques

- Motion Synthesis → Generation of new motions

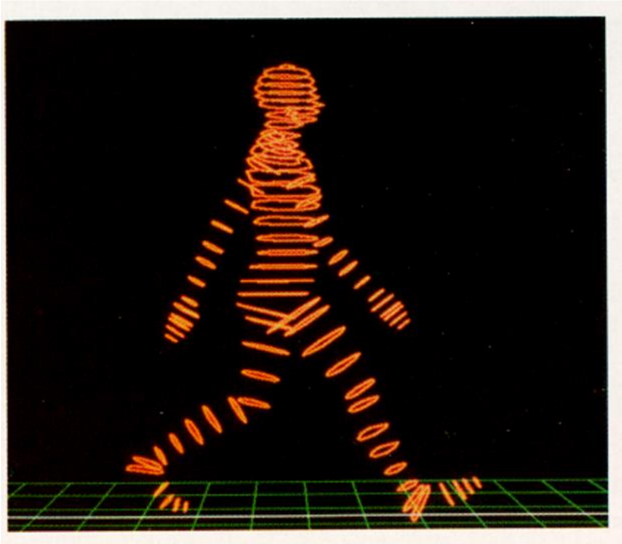
- Procedural techniques
- Physically-based techniques
- **Example-based tecnhiques**

2.1-Techniques

- Procedural Techniques
  - Create motions from scratch
  - Use empirical and biomechanical concepts
  - High level of control
  - Not realistic

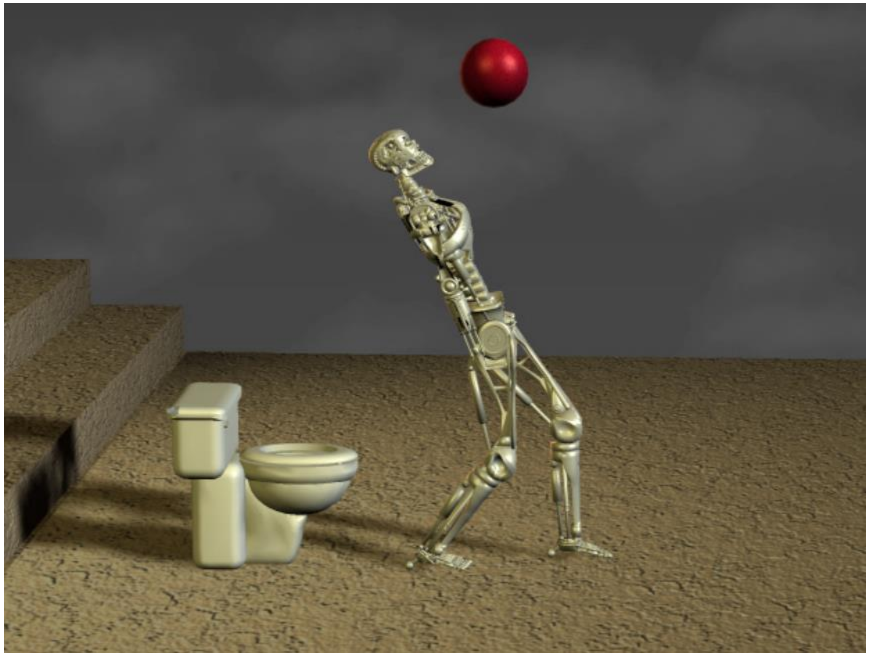
A. Bruderlin and T. W. Calvert. 1989. *Goal-directed, dynamic animation of human walking*. SIGGRAPH Comput. Graph. 23, 3 (July 1989), 233–242.

Boulic, R., Thalmann, N.M. & Thalmann, D. *A global human walking model with real-time kinematic personification*. *The Visual Computer* **6,** 344–358 (1990).



2.1-Techniques

- Physically-based Techniques
  - Dynamics and physical properties
  - Realistic animations
  - Realistic torques on joints
  - Less control over animation
  - Computationally expensive
  - Unsuitable for real-time purposes



Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. 2001
Composable controllers for physics-based character animation. SIGGRAPH '01

2.1-Techniques

- Example-based Techniques
  - Reuse of existing motions
  - Motion capture clips → natural results
  - Concatenation of clips → new sequences
  - Blended by parametrization → new clips
  - Offline Analysis → Anotation → Extraction
  - Machine learning approaches

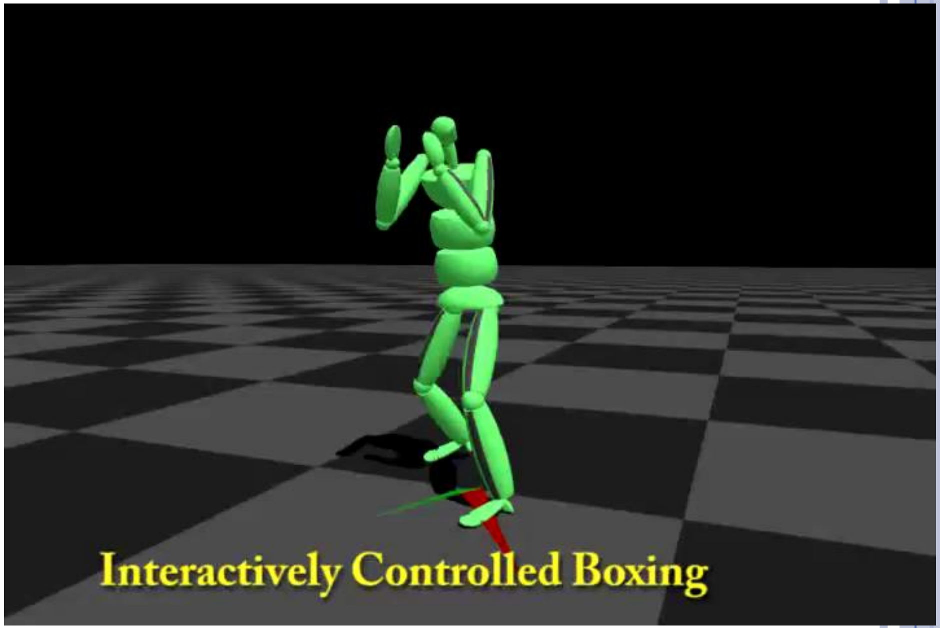
2.1-Techniques

- Motion Concatenation
  - Clips of motion stitched together
  - Transition between them
- Motion patches
  - Spatial and temporal tiles of motion patches
  - Each patch is a collection of motion fragments encapsulating interactions among characters
  - Similar to crowd patches.

Manmyung Kim, Youngseok Hwang, Kyunglyul Hyun, Jehee Lee,
Tiling Motion Patches *SCA 2012*

2.1-Techniques

- Motion Parametrization
  - Interpolate between different existing motions
  - Generate new motions corresponding to a specific parameter



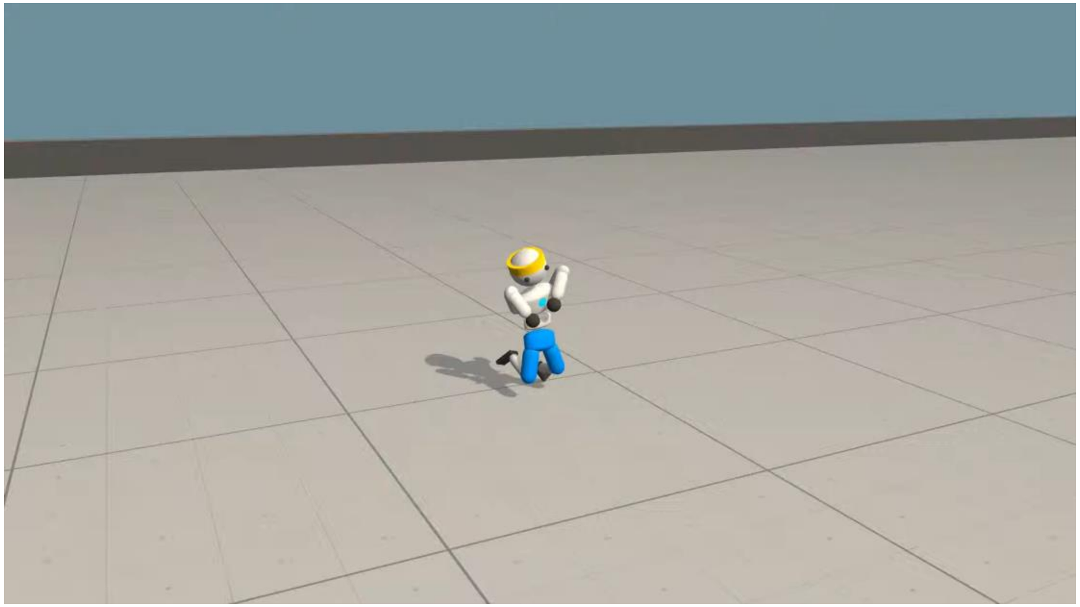
Rachel Heck and Michael Gleicher. 2007. *Parametric motion graphs*. I3D '07

2.1-Techniques

- Machine learning approaches
  - Reinforcement learning
  - Deep learning
  - Motion Matching
  - …

2.1-TECHNIQUES

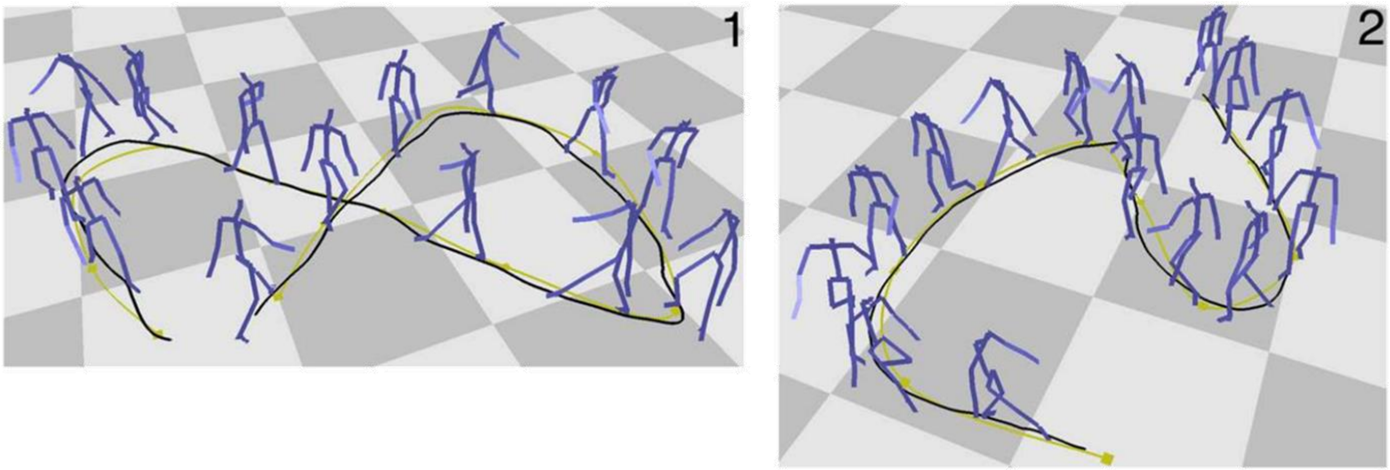
- Machine learning Unity ML-Agents
- A place for you to start learning and playing
- Make a more engaging game w/ ML-Agents | Machine learning bots for game development | Reinforcement learning | Unity
- GitHub - Unity-Technologies/ml-agents: Unity Machine Learning Agents Toolkit



2.2-Motion Graphs

- Given a corpus of motion capture data → Automatic creation of a directed graph called *Motion Graph*
  - Encapsulates connections among the database

- Synthesize motion by following paths on the graph



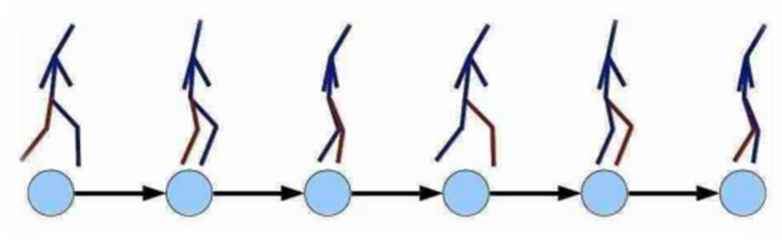
2.2-Motion Graphs

- Motion graph
  - Directed graph
  - Edges
    - Pieces of original motion
    - Automatically generated transitions
  - Nodes
    - Choice points → seamless join of motions
      - Poses with minimal distances

- No need to capture specific transition motions

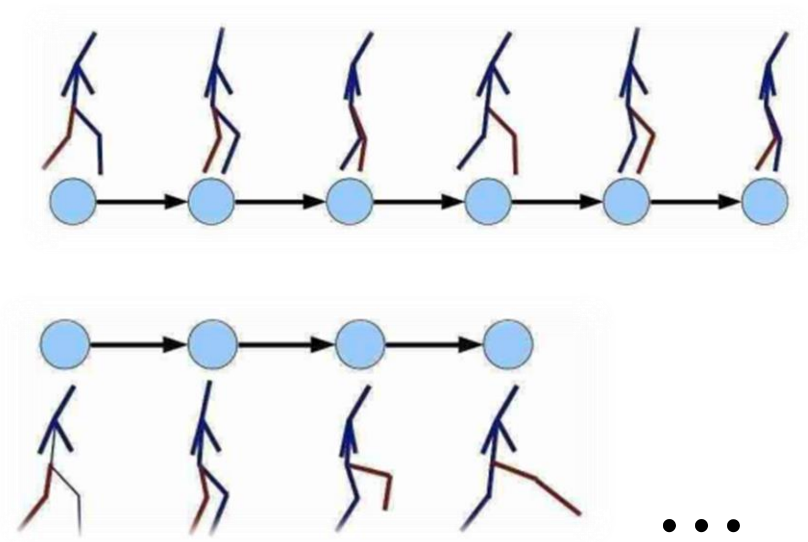
2.2-Motion Graphs

- Every motion clip is a graph
- Node = pose frame
- Edge = transition between frames



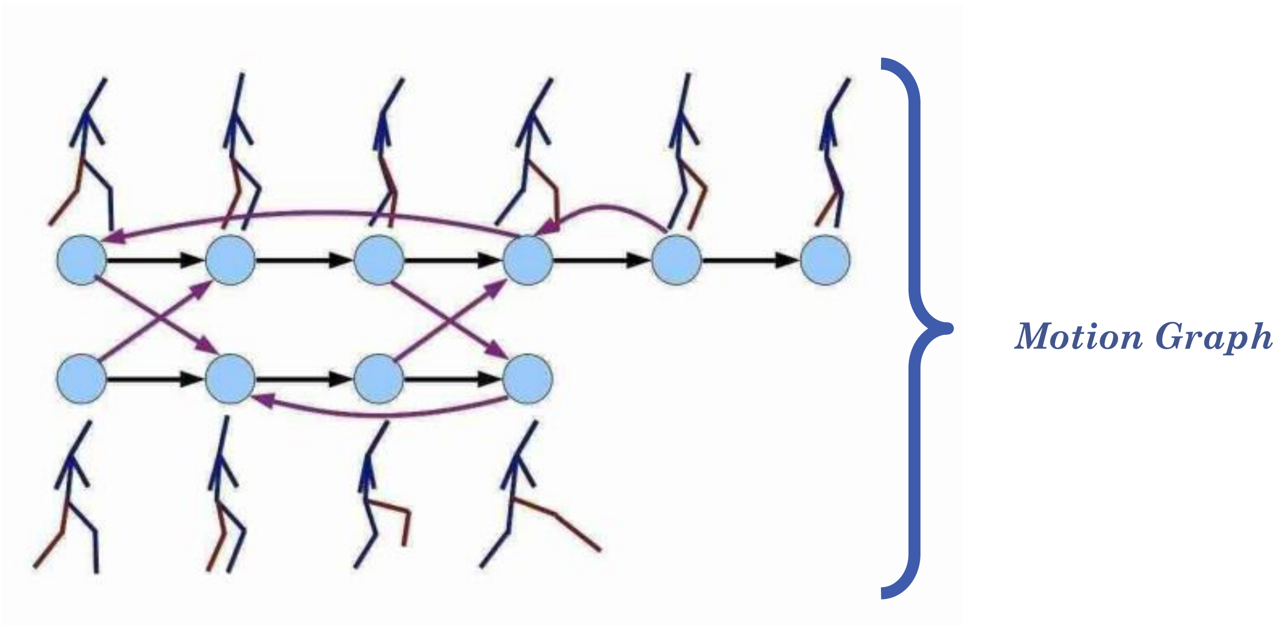
2.2-Motion Graphs

- You start with as many graphs as clips in your motion database



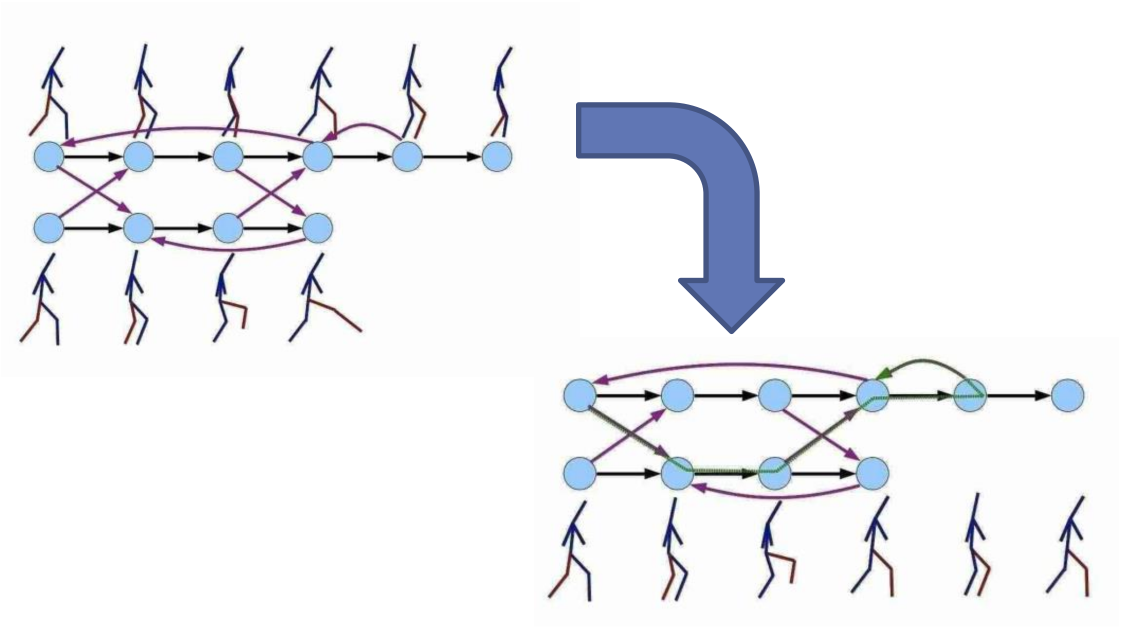
2.2-MOTION GRAPHS

- Find similar poses between clips (within a clip and between clips)
- Add transitions between them



2.2-Motion Graphs

- Any walk on this graph generates a new motion



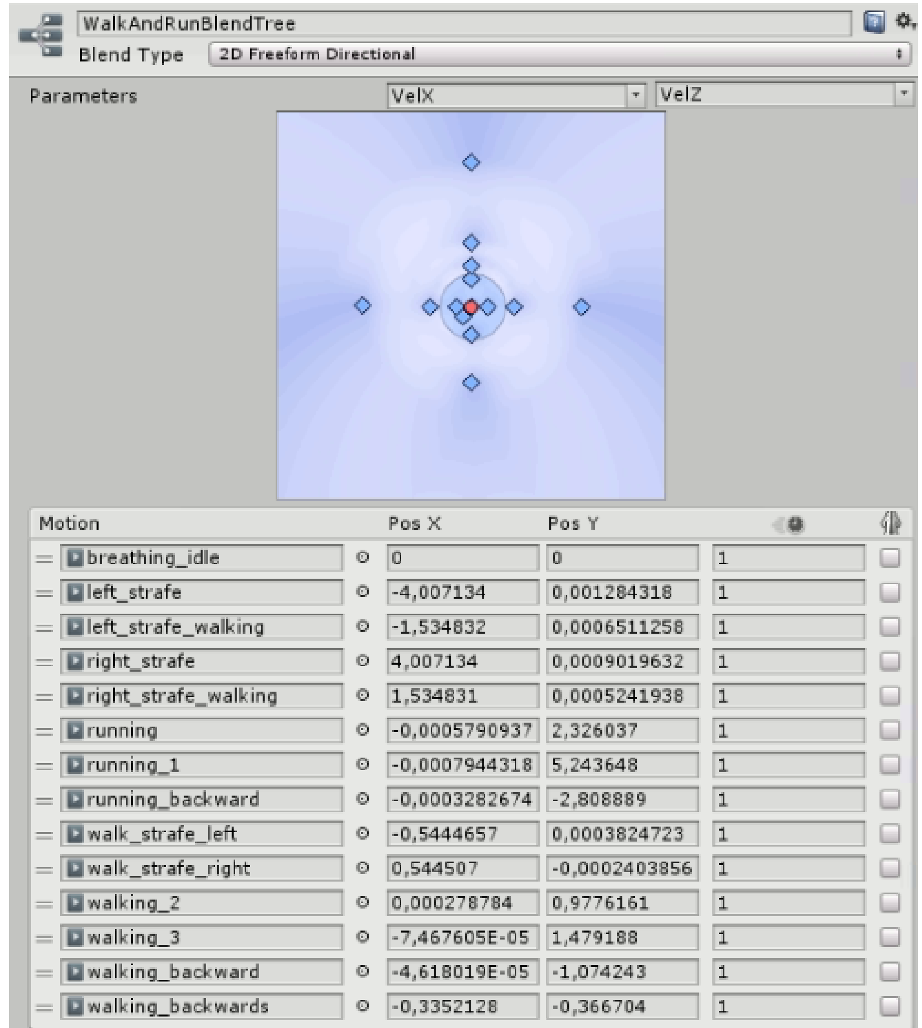
2.3-Parametric Motion

- Motion Control
  - Temporally-dense external signals
    - Usually user-defined
  - → Drive the generation of an animation

- Parametric Motion
  - Interpolate between different existing motions
  - Generate new motions
    - Corresponding to a specific parameter

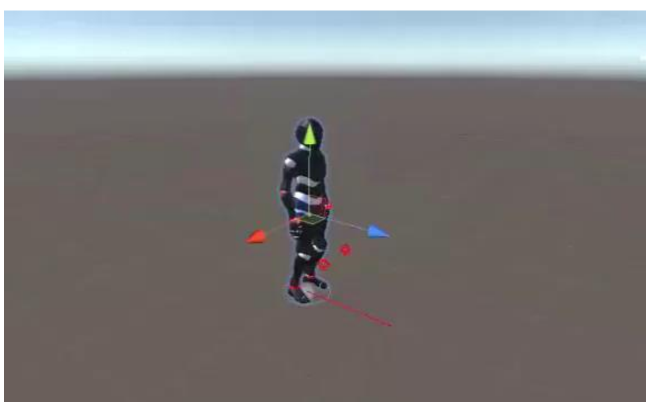
2.3-Parametric Motion

- Parametrize motion by
  - Velocity vector
    - Speed
    - Direction



Motion	Pos X	Pos Y	
breathing_idle	0	0	1
left_strafe	-4,007134	0,001284318	1
left_strafe_walking	-1,534832	0,0006511258	1
right_strafe	4,007134	0,0009019632	1
right_strafe_walking	1,534831	0,0005241938	1
running	-0,0005790937	2,326037	1
running_1	-0,0007944318	5,243648	1
running_backward	-0,0003282674	-2,808889	1
walk_strafe_left	-0,5444657	0,0003824723	1
walk_strafe_right	0,544507	-0,0002403856	1
walking_2	0,000278784	0,9776161	1
walking_3	-7,467605E-05	1,479188	1
walking_backward	-4,618019E-05	-1,074243	1
walking_backwards	-0,3352128	-0,366704	1

- Blend trees



2.3-Parametric Motion
Parametric Space

- Map motion clips → space
  - As many dimensions as parameters to guide your motion

- 1 parameter → 1D space (speed or angle)
- 2 parameteres → 2D space (2D displacement XZ, or speed and angle, …)
- 3 parameters → 3D space (position of an end effector)
- N parameters → N-dimensional space

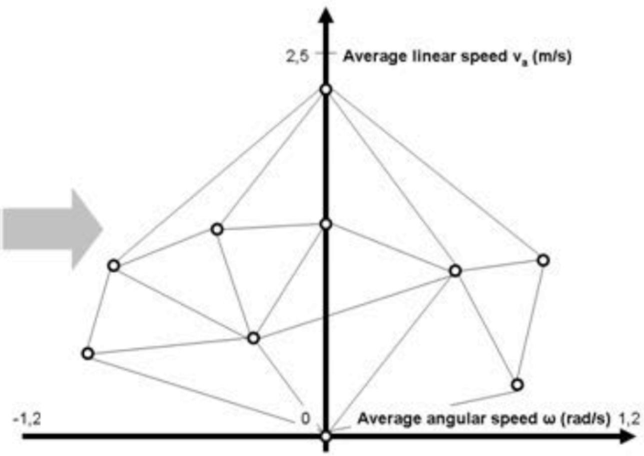
2.3-Parametric Motion Parametric Space

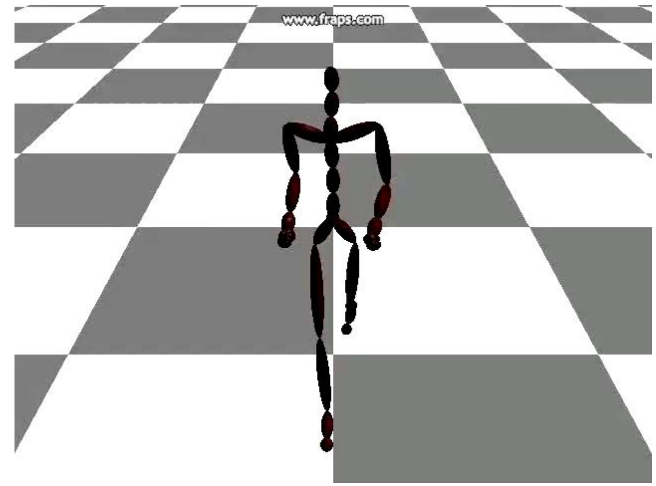
Julien Pettre and Jean-Paul Laumond.
A motion capture-based control-space approach for walking mannequins.
Comp. Anim. Virtual Worlds 2006

CHARACTERISTIC VELOCITIES

LINEAR $\hat{v}(m.s^{-1})$	ANGULAR $\hat{\omega}(rad.s^{-1})$
2.2	0
1.3	0
0	0
1	0.6
1.1	0.75
0.3	0.8
0.5	-0.3
1.3	-0.4
1.2	-0.8
0.4	-0.9

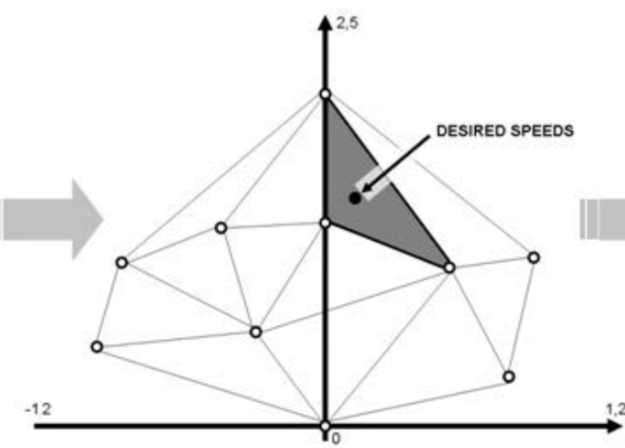
(v, ω) space

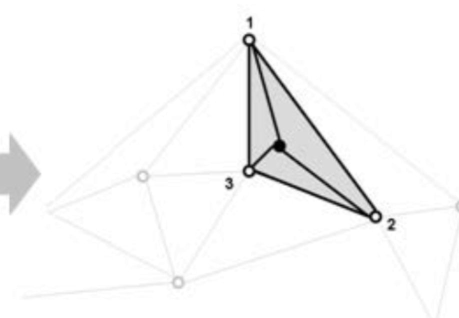




$$\begin{cases} a\hat{v}_1 + b\hat{v}_2 + c\hat{v}_3 = v_d \\ a\hat{\omega}_1 + b\hat{\omega}_2 + c\hat{\omega}_3 = \omega_d \\ a + b + c = 1 \end{cases}$$

(v_d, ω_d)





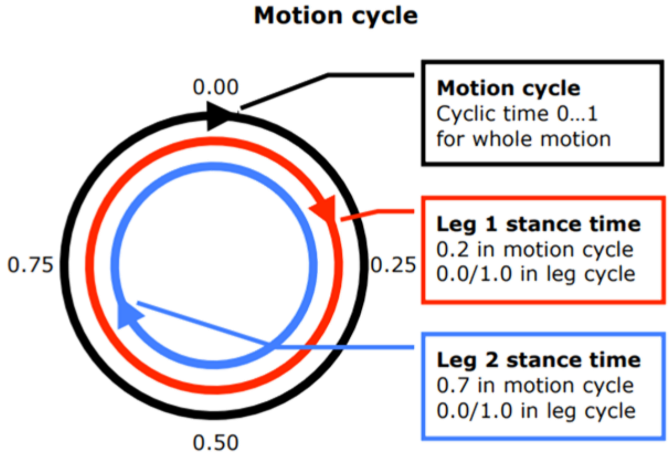
2.3-Parametric Motion Parametric Space

- Clips can be manually tagged or mapped to parametric space
  - Too subjective
  - Too long
  - Too much tweaking

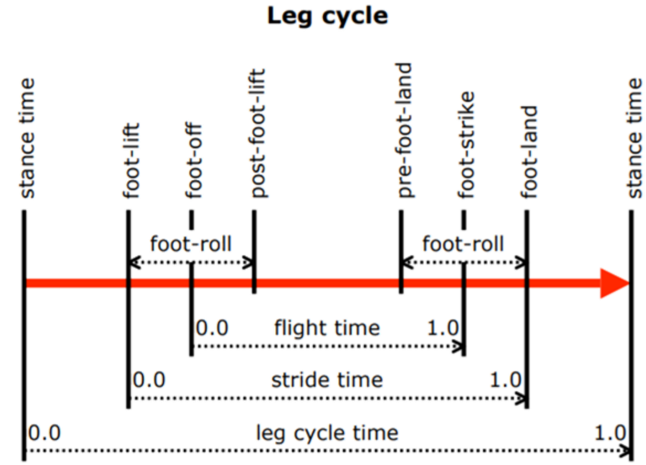
- Clips must be analyzed
- Sampling at some rate
- Parameters extracted
- Clips must also be aligned for proper blending

2.3-Parametric Motion Parametric Space

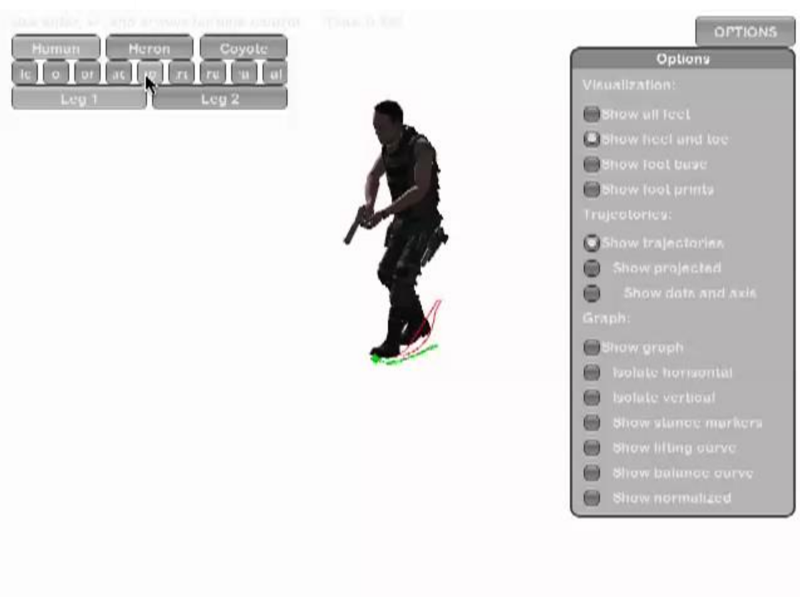
Johansen R. Automated Semi-Procedural Animation. *Master Thesis* (2009). URL Link



(a) *Leg cycles relative to motion cycle. This example shows two legs with stance times opposite each other in the motion cycle.*



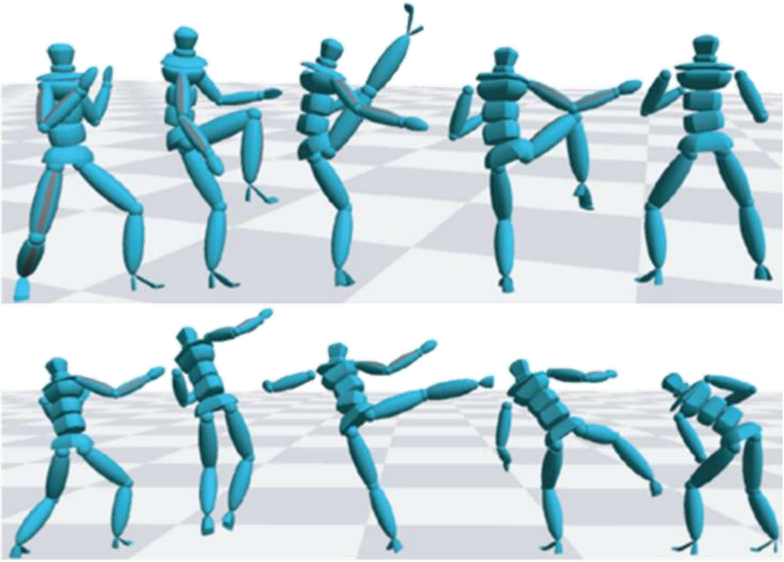
(b) *Keytimes in leg cycle. Each leg has a leg cycle with the leg cycle time going from 0.0 at the stance time to 1.0 at the stance time again.*



2.3-Parametric Motion Parametric Space

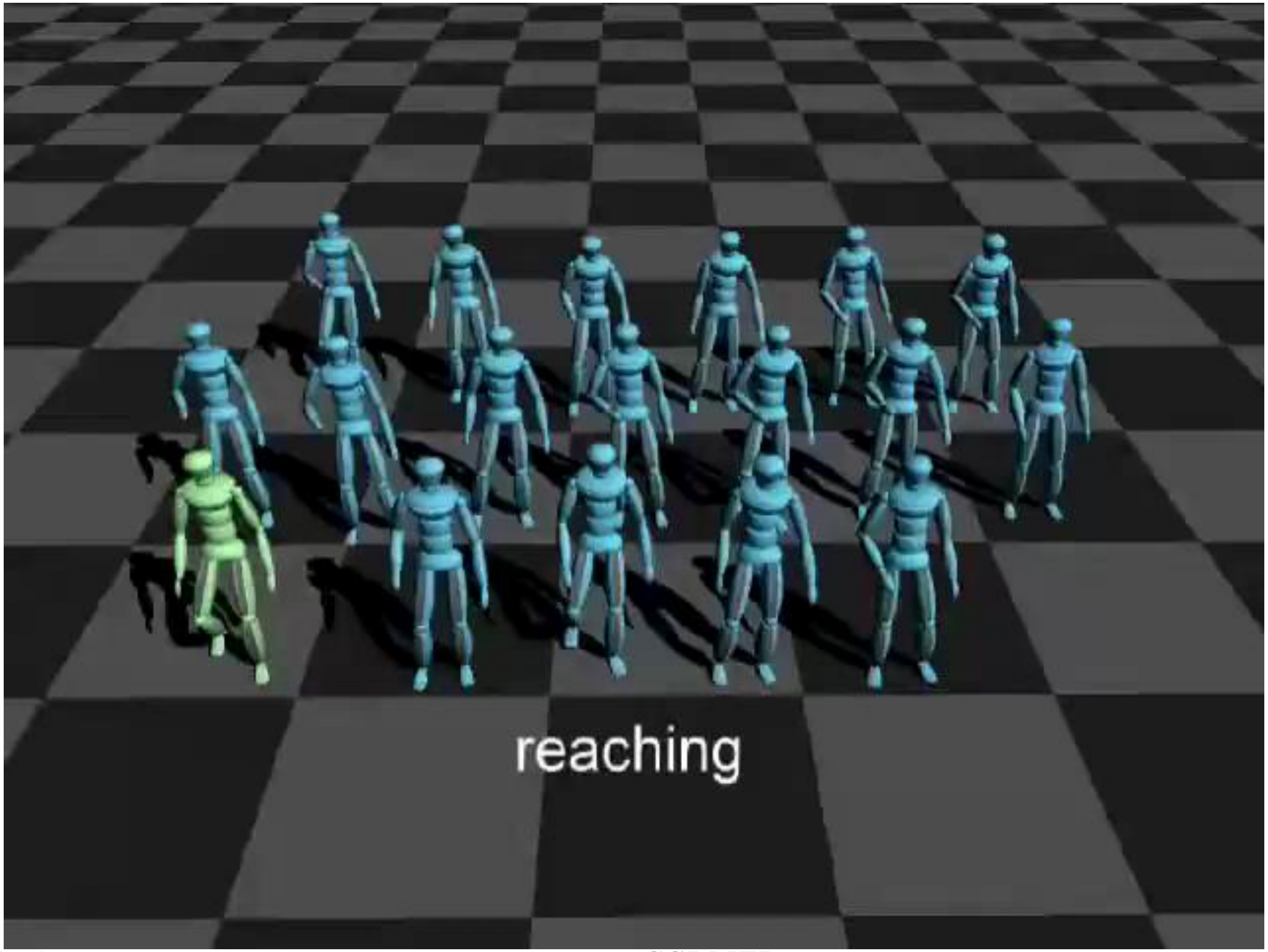
Kovar and Gleicher. **Automated extraction and parameterization of motions in large data sets.** ACM Transactions on Graphics 2014

- Large data sets → many variants of same kind of motion
- Automatically identify logically similar motions
- Use them to build parameterized space of motion
- Find logically similar but numerically similar
  - Distance metric → find “close” motions
  - Intermediaries to find more distant motions
- Related motions
  - Register
  - Blending techniques



2.3-Parametric Motion Parametric Space

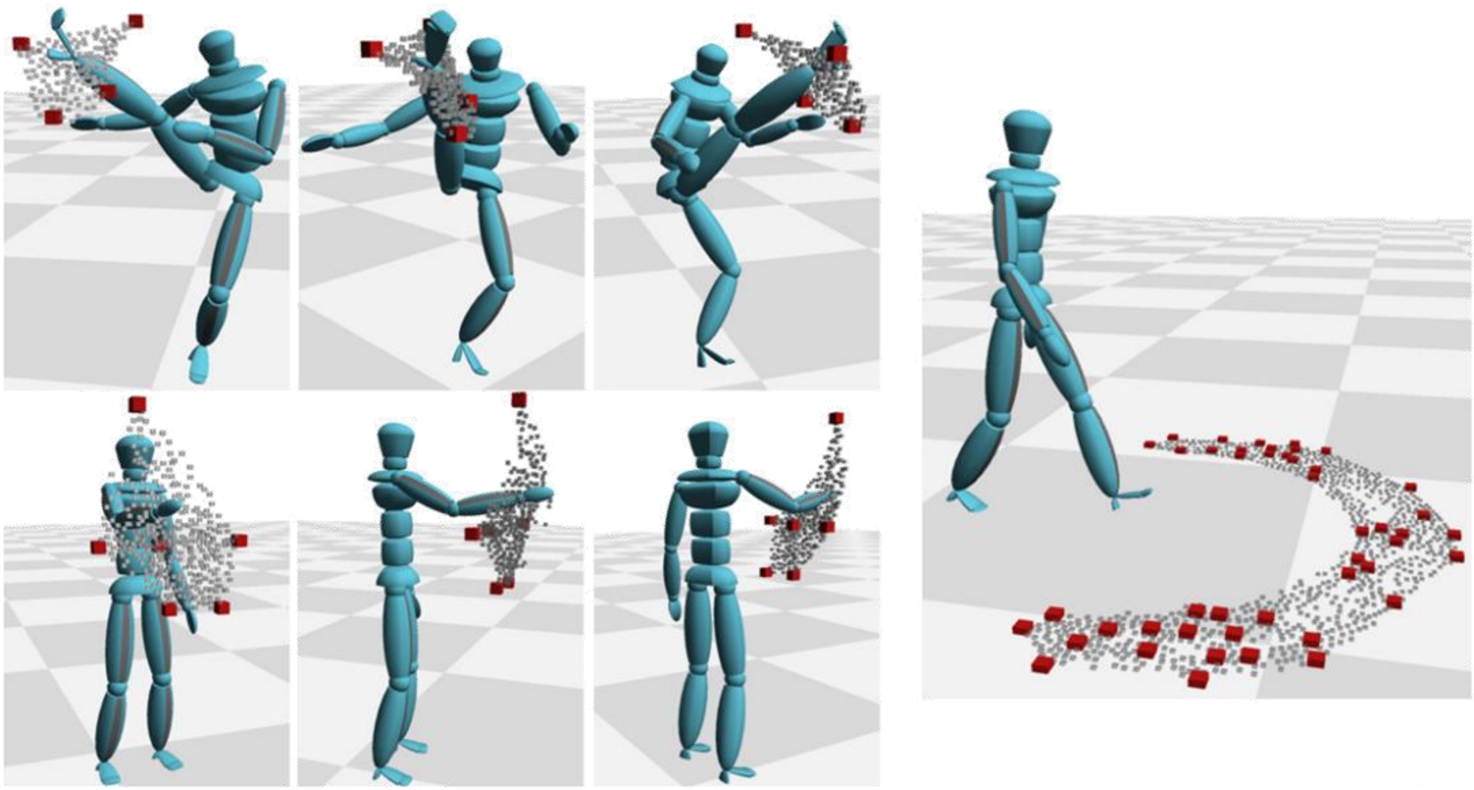
Kovar and Gleicher. **Automated extraction and parameterization of motions in large data sets.** ACM Transactions on Graphics 2014



Kovar and Gleicher. **Automated extraction and parameterization of motions in large data sets.** ACM Transactions on Graphics 2014

2.3-Parametric Motion Parametric Space

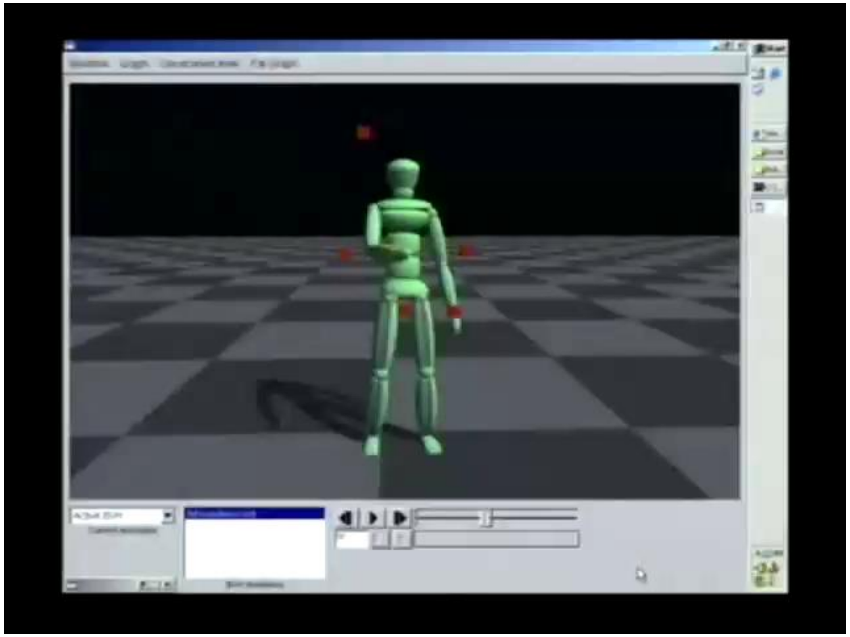
- Sample subsets of blend weights → find sets of example motions nearby in parameter space



Kovar and Gleicher. **Automated extraction and parameterization of motions in large data sets.** ACM Transactions on Graphics 2014

2.3-Parametric Motion Interpolation

- Motion blending or motion interpolation
  - Blend similar motions
  - Weights
  - Parameterize high-level characteristics of interest
- Several methods to compute weights
  - Different advantages
  - Different disadvantages



K-nearest-neighbors interpolation

2.3-Parametric Motion Interpolation

- Motion parameterization depends on
  - Application type
  - Required constraints

- Locomotion
  - Ensure weights vary smoothly
- Reaching motion
  - Precise goal-attainment
  - Less parametric error → accuracy

2.3-Parametric Motion Interpolation

- Barycentric
- Radial Basis Function (RBF)
- K-Nearest Neighbors (KNN)
- Inverse Blending (InvBld)

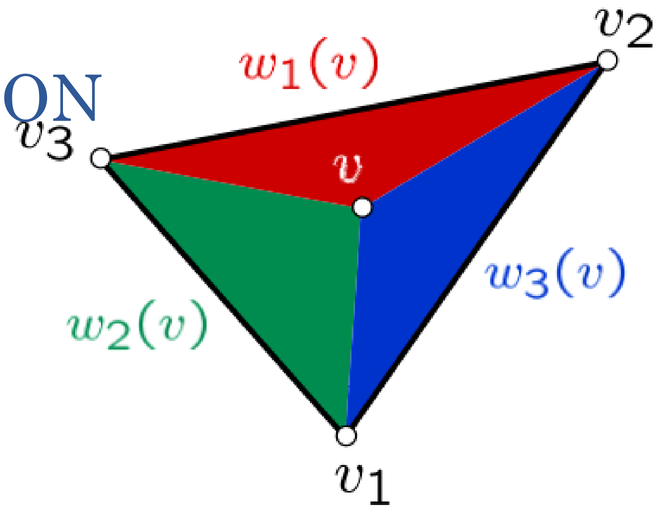
$$M(\boldsymbol{w}) = \sum_{i=1}^{k} w_i M_i$$

2.3-Parametric Motion
Barycentric interpolation [Möbius, 1827]

- Most basic form for motion parameterization
- Assumes that motion parametrization can be linearly mapped to blending weights
  - May not hold for all cases
- 1D → n-D
  - Linear interpolation → barycentric interpolation
- 3D parametric space → Tetrahedralization
  - Manually or Delaunay triangulation
- Given a new motion parameter
  - Search for tetrahedron enclosing it

[Möbius, 1827]

2.3-Parametric Motion
Barycentric interpolation



- Weights → barycentric coordinates inside tetrahedron
- 2D → Triangle
- n-D → n-D simplex
- Can be defined in other abstact spaces
  - Limited ability for extrapolation outside the convex hull

$$b_i(v) = \frac{w_i(v)}{w_1(v) + w_2(v) + w_3(v)}$$

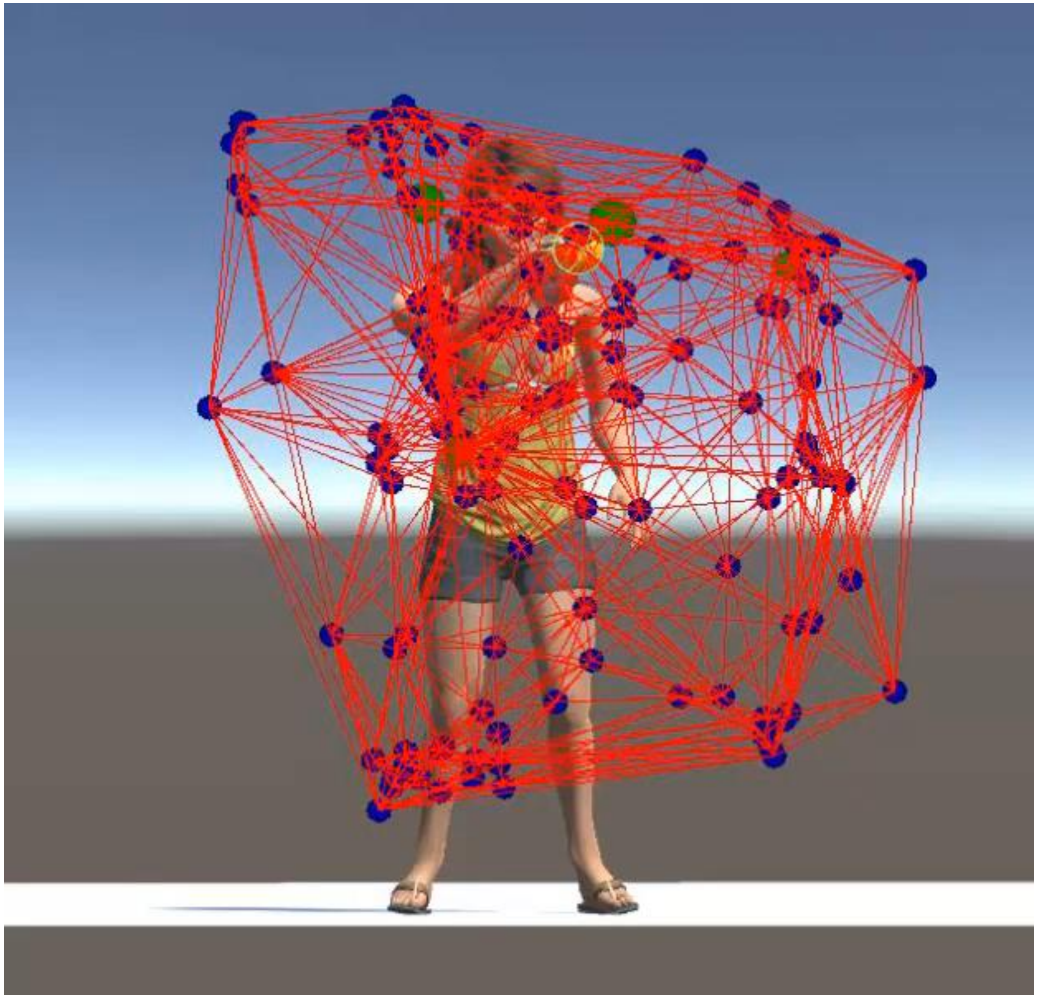
$$\textstyle\sum_i b_i(v) = 1$$

$$\textstyle\sum_i b_i(v) v_i = v$$

$$b_i(v) > 0, \quad v \in \mathring{\triangle}$$

$$b_i(v_j) = \delta_{ij}$$

2.3-Parametric Motion
Barycentric interpolation [Möbius, 1827]



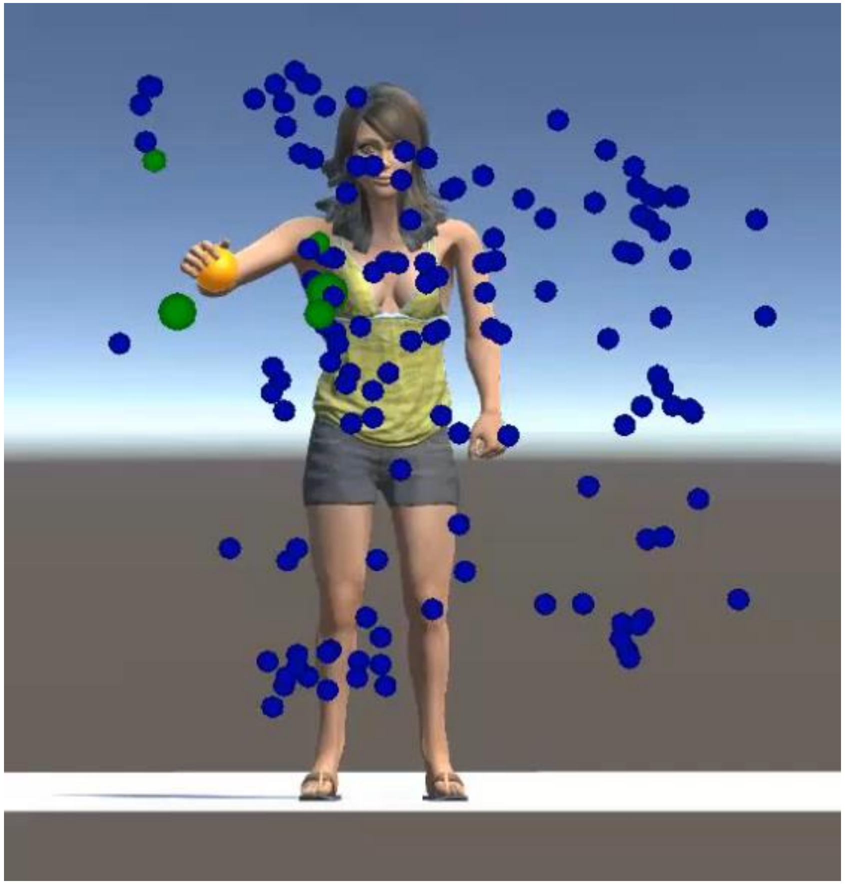
2.3-Parametric Motion
K-Nearest Neighbors (KNN) interpolation

- K-closest examples
- Compute blending weights based on the distance between the input point and nearby examples

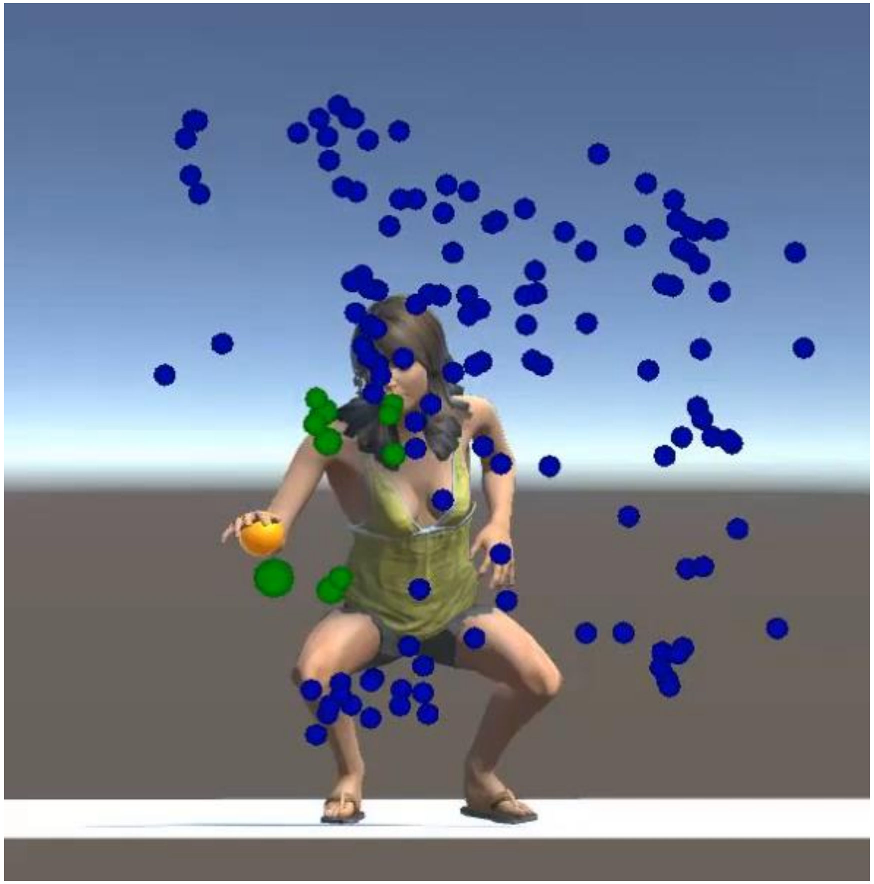
$$w_i = \frac{1}{\|p - p_{n_i}\|} - \frac{1}{\|p - p_{n_k}\|}$$

- Normalized weights so that $w_1 + w_2 + \ldots + w_k = 1.$
- Easy to implemnt
- Works well in dense space
- Sparse points → inaccurate results
  - Pseudo-examples → weighted combination of existing samples → randomly sampling blend weights space
- k-D tree → fast proximity query at run-time

2.3-Parametric Motion
K-Nearest Neighbors (KNN) interpolation



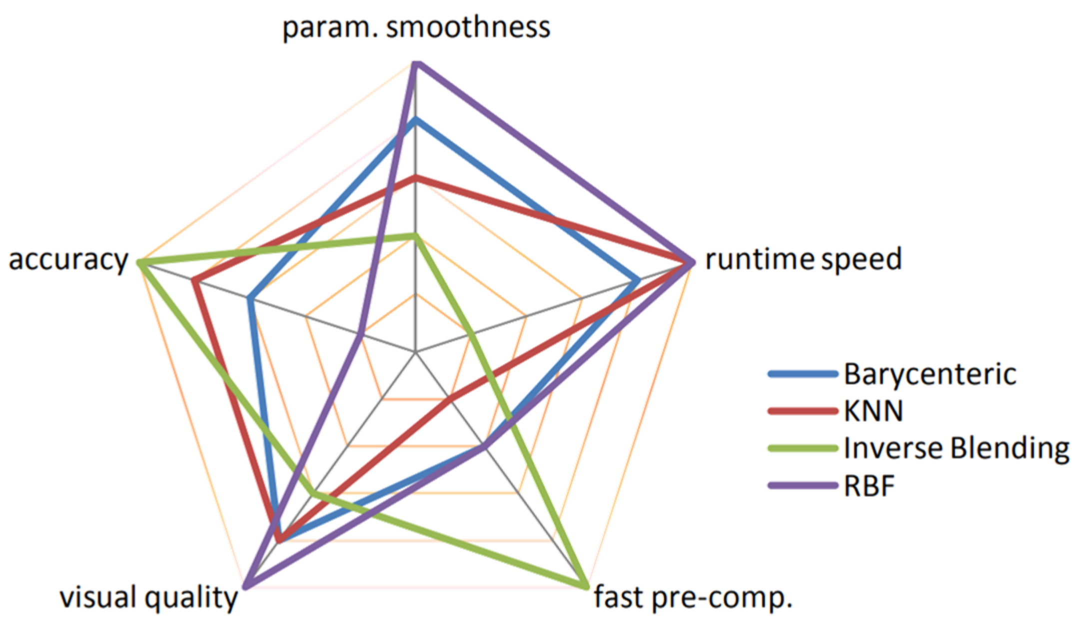
$k = 5$



$k = 10$

2.3-Parametric Motion Interpolation

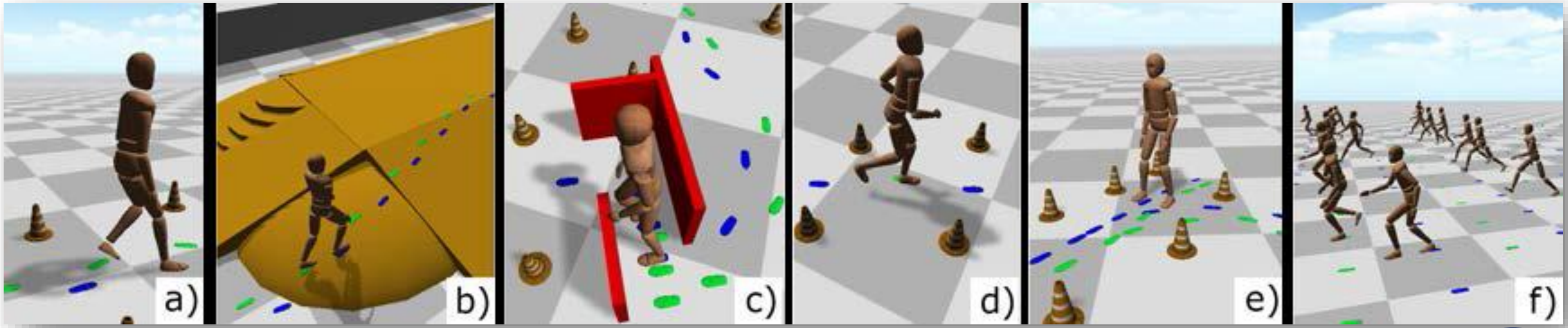
- Good survey comparing interpolation techniques
- Feng et al. **An Analysis of Motion Blending Techniques**. Motion in Games 2012.
- http://www.arishapiro.com/mig12/mig12compare.pdf



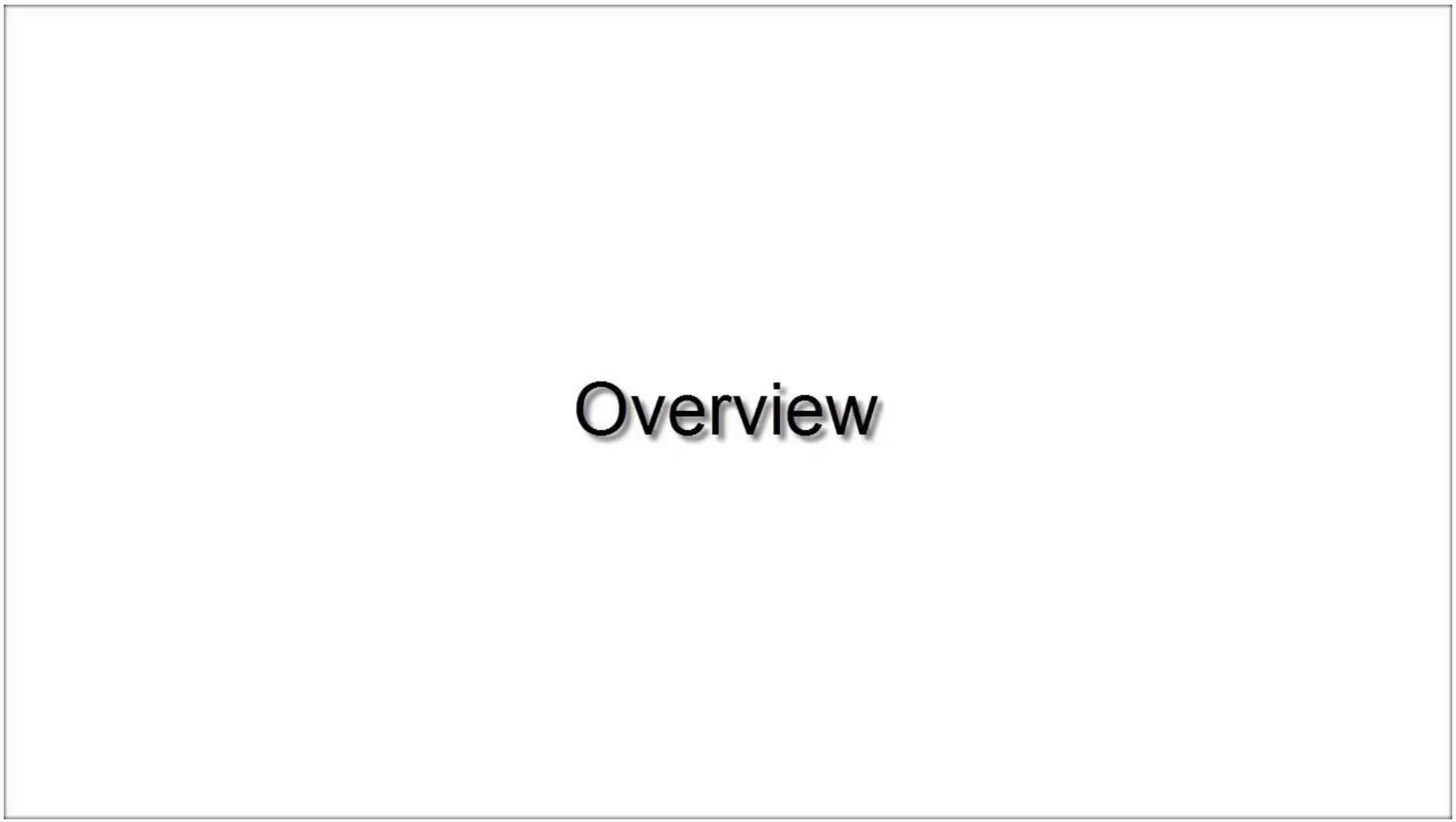
A. Beacco, N. Pelechano, M. Kapadia and N.I. Badler . "Footstep Parameterized Motion Blending using Barycentric Coordinates" Computer & Graphics 2014

2.4-Following Footsteps

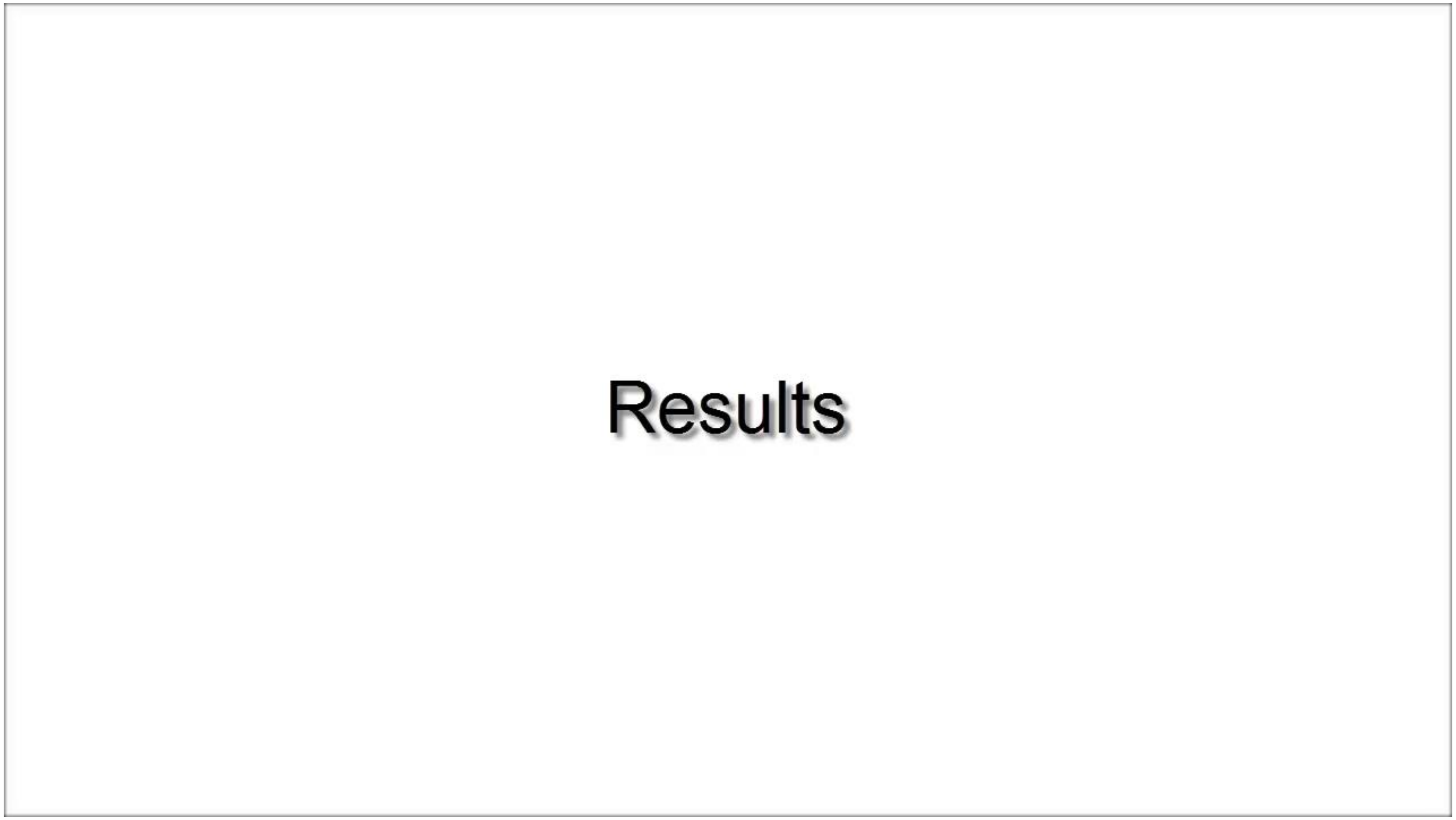
- Online animation synthesis for footsteps simulators
- Satisfy foot placement constraints
- User control over the trade-off between footstep accuracy and root velocity



2.4-Following Footsteps



2.4-Following Footsteps

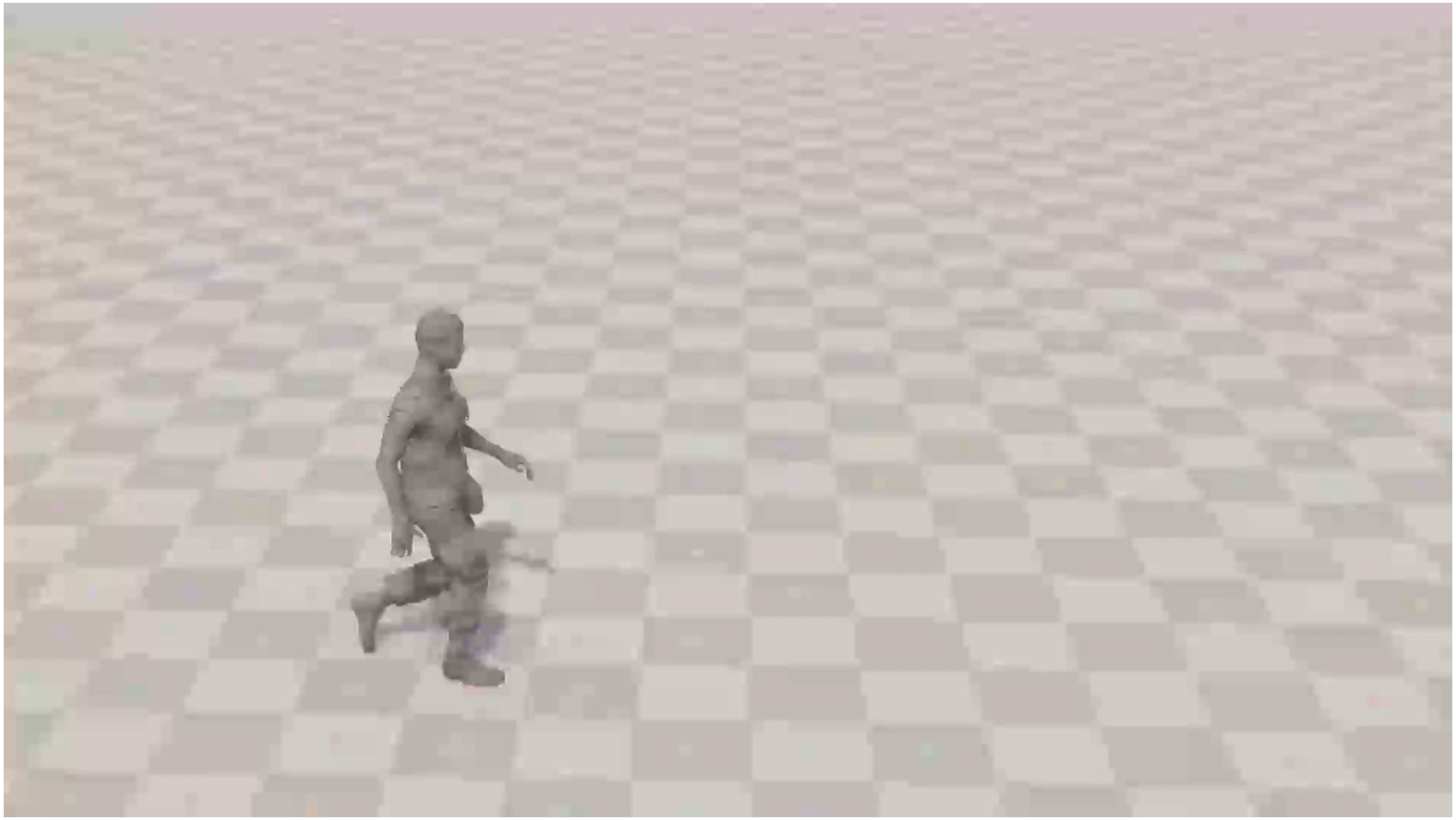


2.5-Motion Matching

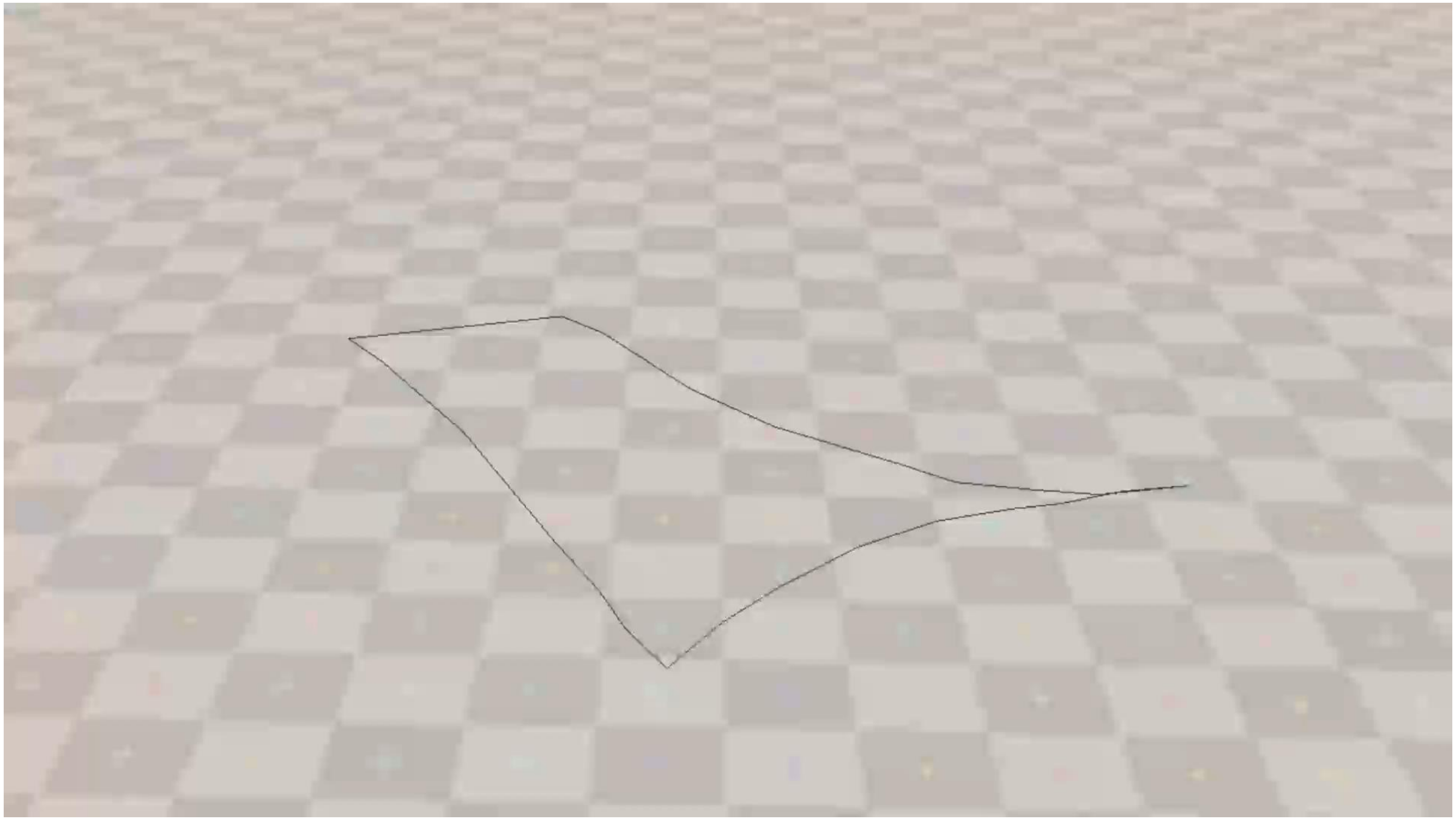
- A ridiculously brute-force approach to animation selection
- Algorithm: Every frame, look at all mocap data and jump at the best place
- No manual work
  - No graphs
  - No alignment
  - No explicit transitions
- Works best with big database
  - Large memory

Ubisoft – Simon Clavet – 2015 – For Honor – Game Developpers Conference
https://www.gdcvault.com/play/1023280/Motion-Matching-and-The-Road
https://www.gdcvault.com/play/1026472/ML-Tutorial-Day-From-Motion

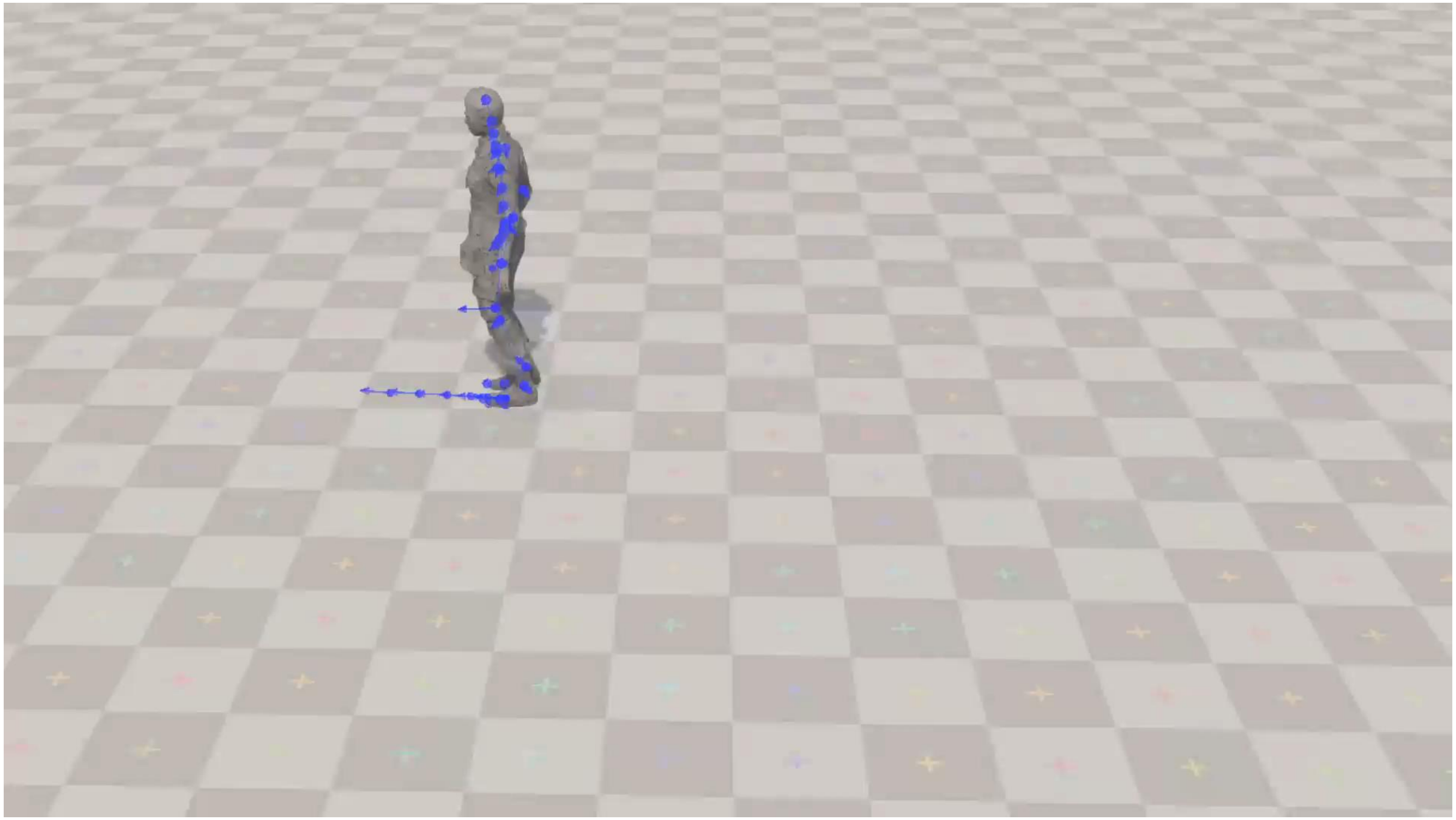
2.5-Motion Matching



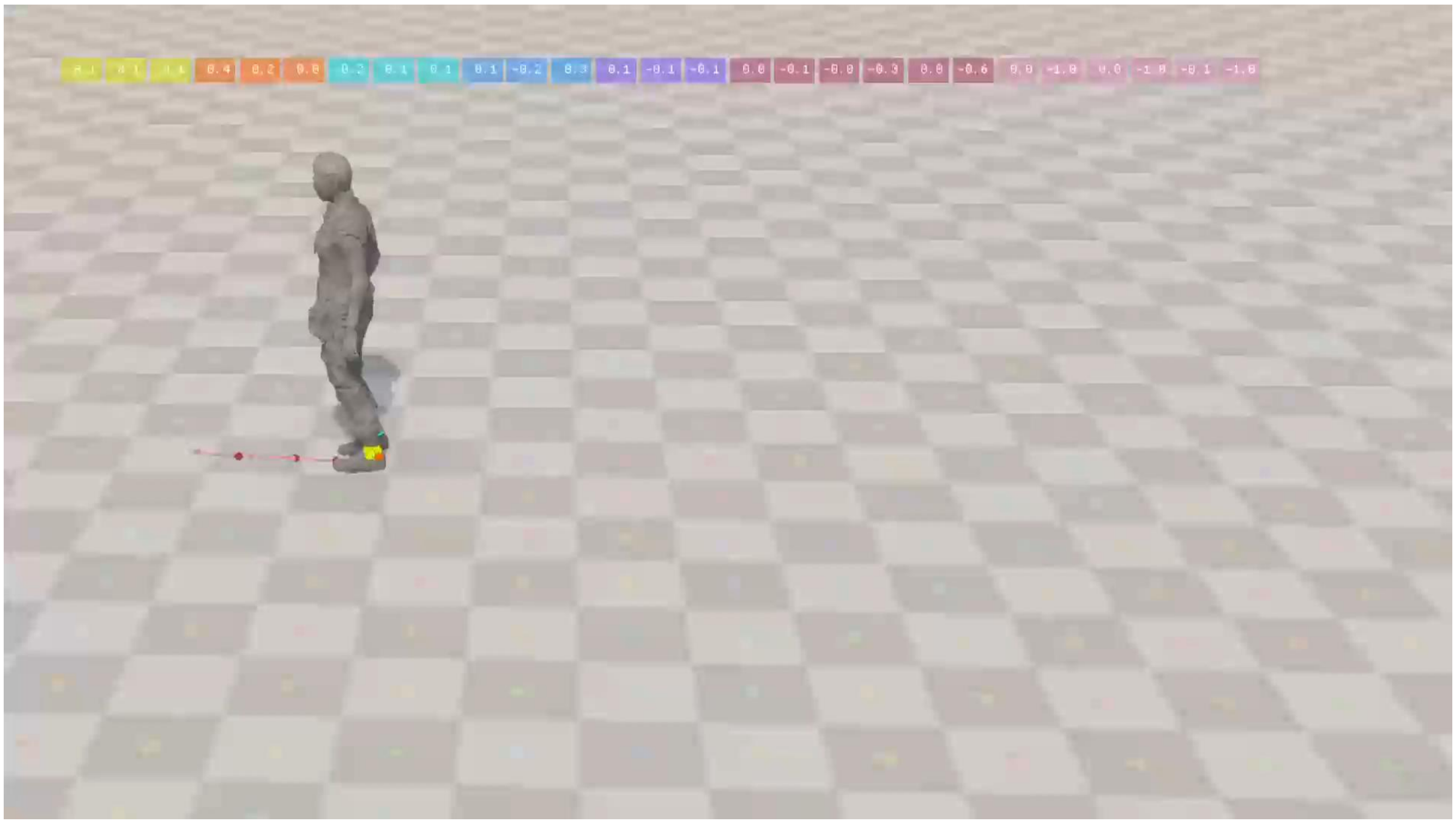
2.5-Motion Matching



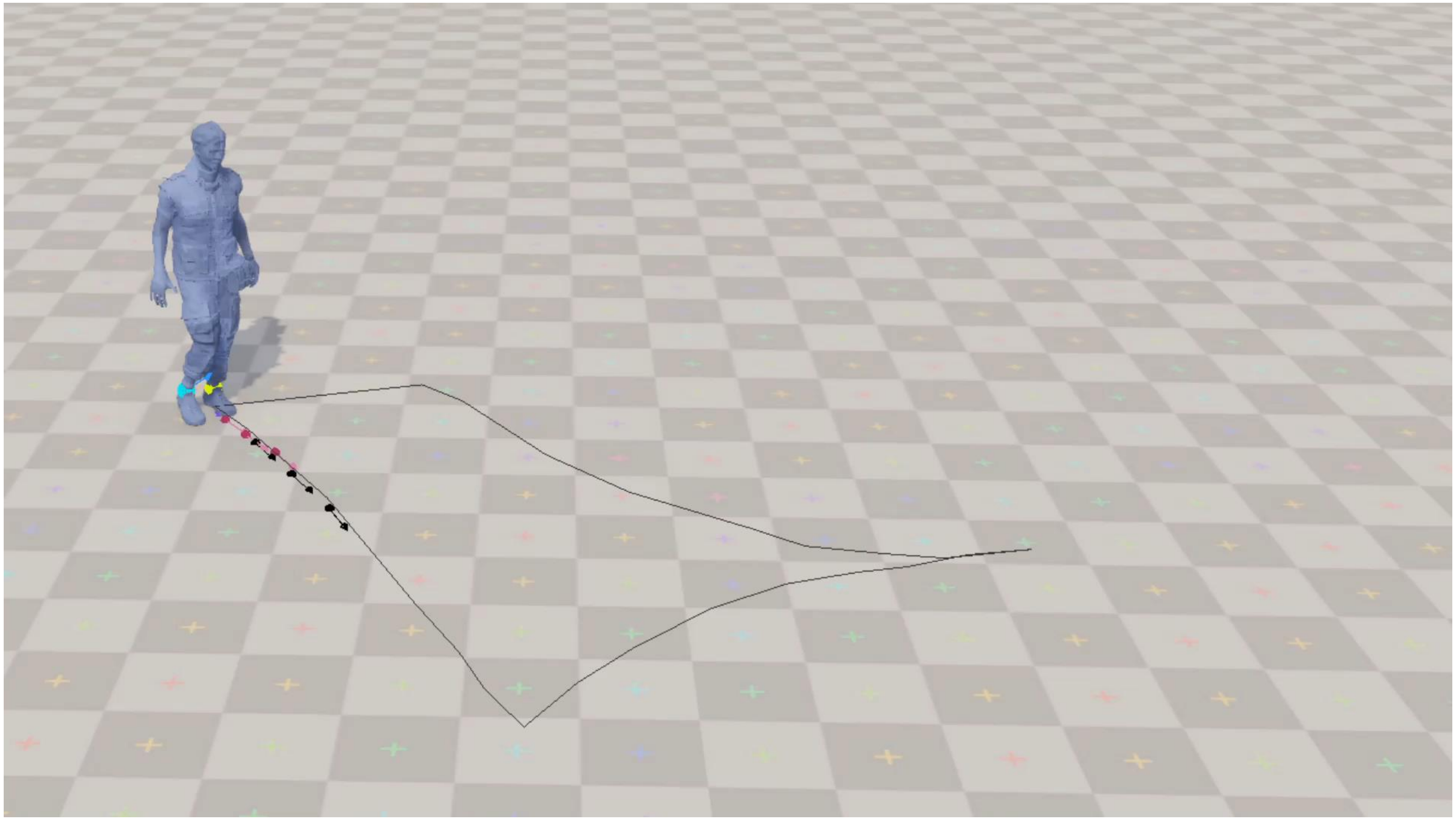
2.5-Motion Matching



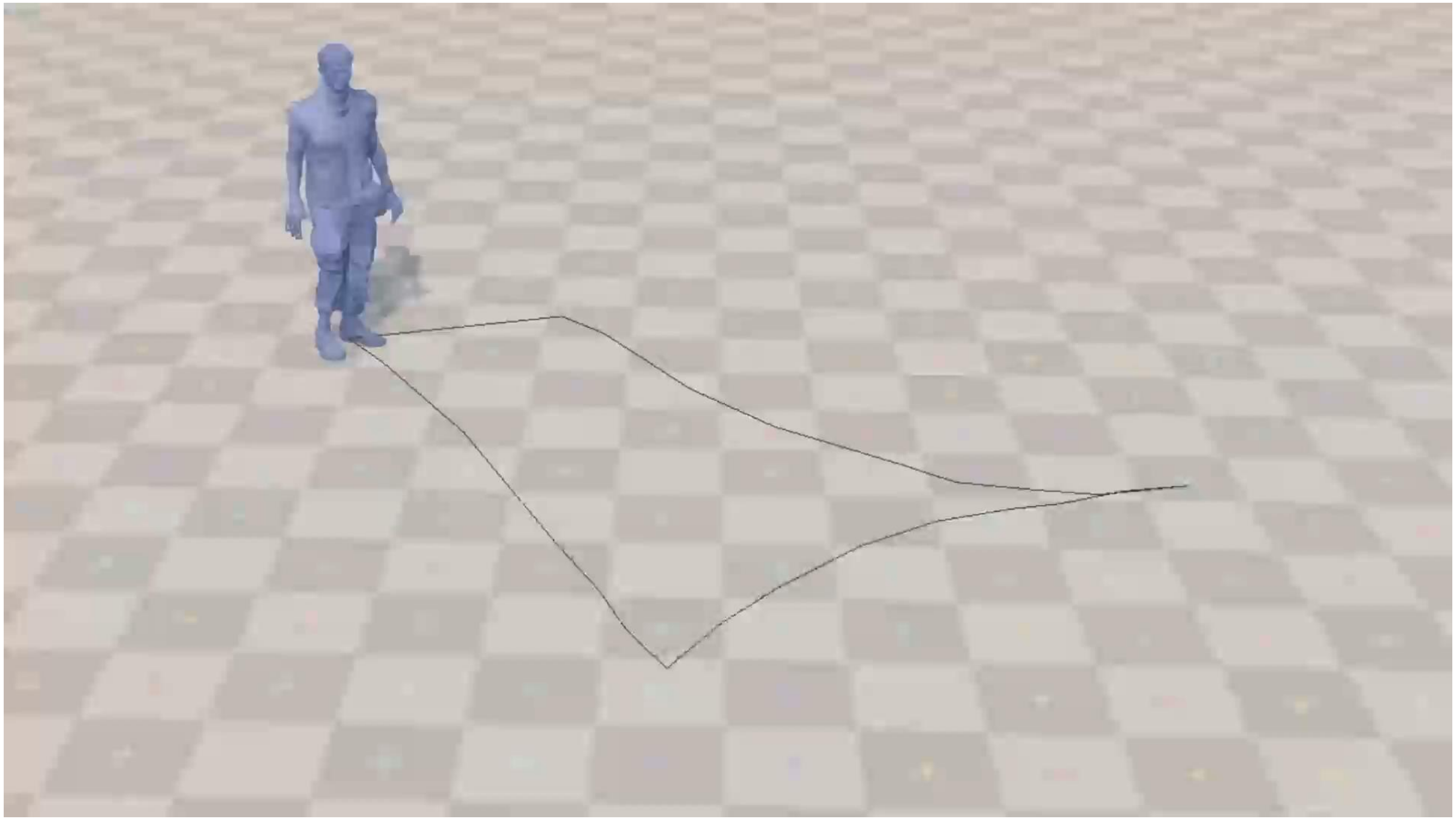
2.5-Motion Matching



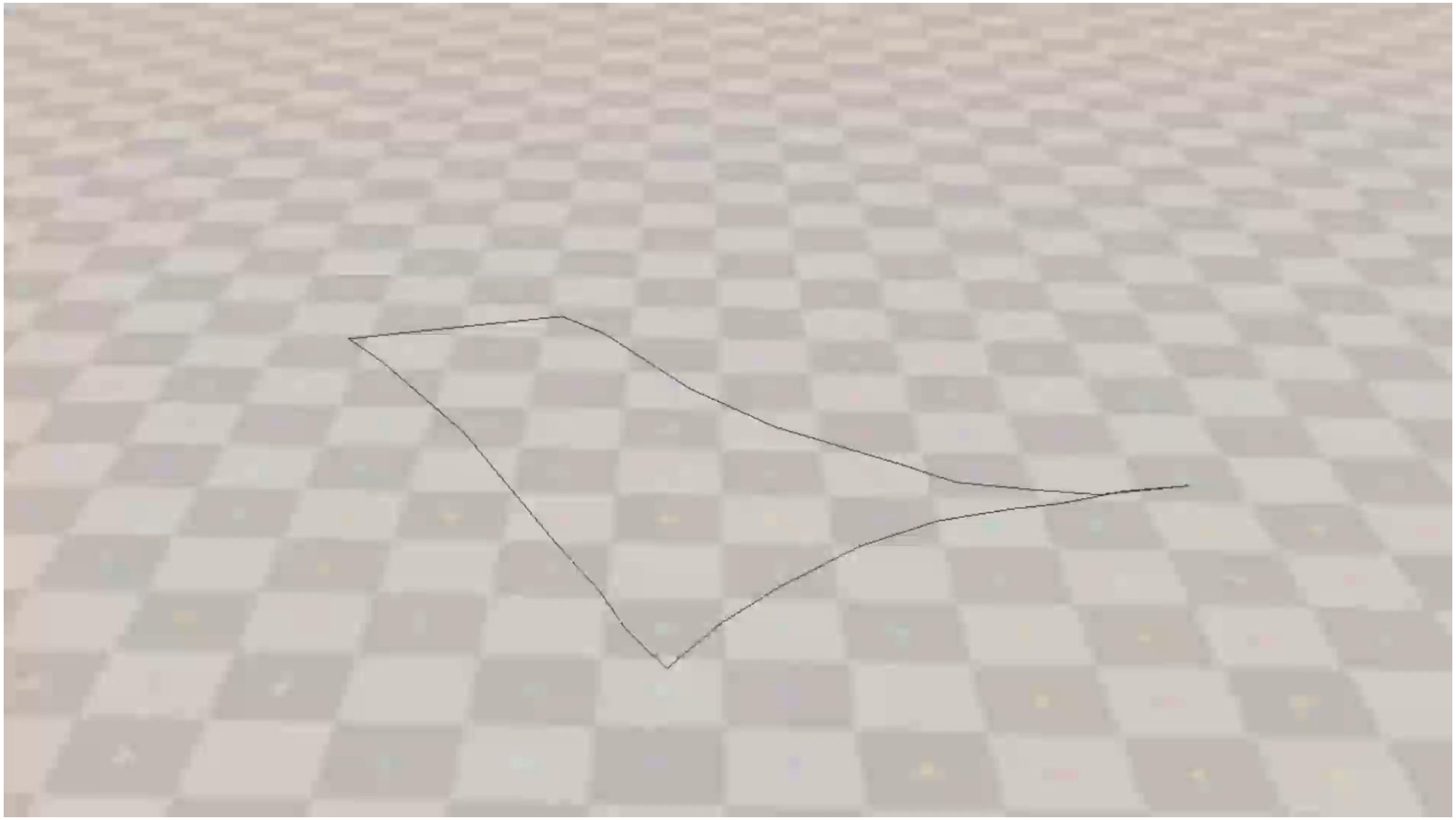
2.5-Motion Matching



2.5-Motion Matching



2.5-Motion Matching



3-Crowds and Navigation

- 3.1-Introduction to Navigation
- 3.2-Navigation Meshes
- 3.3-Pathfinding
- 3.4-Steering

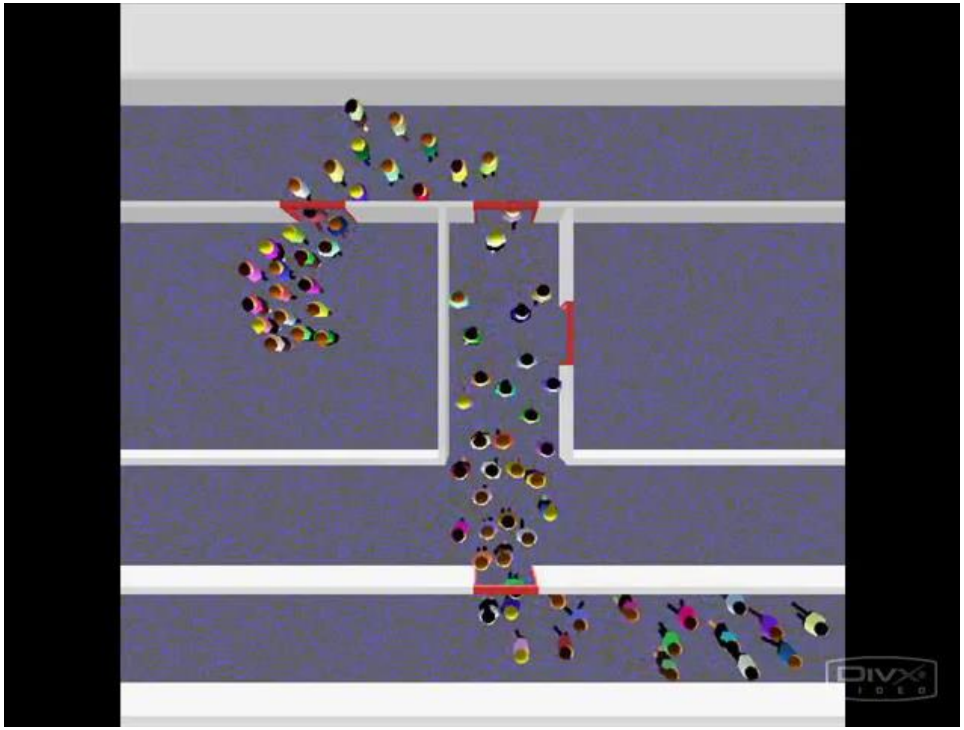
3.1-Introduction to Navigation

- Crowd simulation
  - Macroscopic simulations
    - Similar to particles or fluid simulations
    - Simulation of the crowd as a whole
  - Microscopic simulations

Rahul Narain, Abhinav Golas, Sean Curtis, and Ming C. Lin, 2009. **Aggregate Dynamics for Dense Crowd Simulation.** In *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, vol. 28, no. 5, pp. 122:1–122:8.

3.1-Introduction to Navigation

- Crowd simulation
  - Macroscopic simulations
    - Similar to particles or fluid simulations
    - Simulation of the crowd as a whole
  - **Microscopic simulations**
    - Individual motion per agent
    - Each agent has its goal and parameters



Controlling Individual Agents in High-Density Crowd Simulation.
N. Pelechano, J. Allbeck and N. Badler.
ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA'07) August 3-4, San Diego (USA). 2007

3.1-Introduction to Navigation

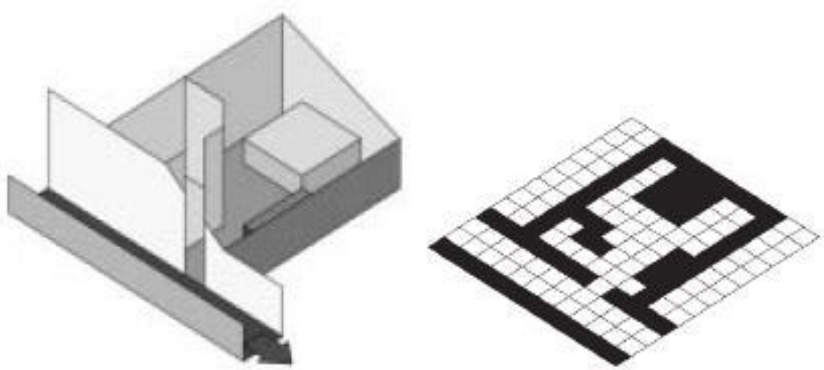
- AI → action → Goal position in world
  - Go to work = **exiting house** → get to metro station A → take metro → stop at station B → walk to work
- Different Levels of planning / navigation
  - **High Level planning** → a path of places to traverse in order to reach the goal
    - Exiting house = exit bedroom → traverse corridor → exit entrance → get down stairs → exit building entrance
    - **Navigation meshes + pathfinding**
  - Local movement → how to navigate locally
    - Avoiding obstacles (dynamic and static)
      - Plants, chairs, table, cars, …
    - Avoiding other agents
    - Steering

3.2-Navigation Meshes

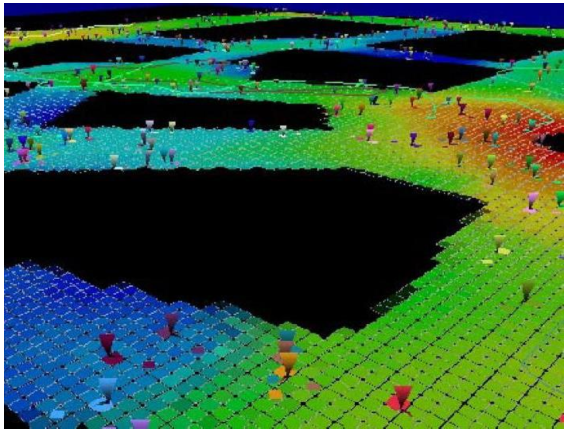
- What is a Navigation Mesh?
  - An abstract representation of the walkable space
  - Represented as a graph, where nodes or cells are convex obstacle free regions, and edges are portals between those regions

3.2-Navigation Meshes
Regular Grids

- 2D grid
- All cells have the same size
- value per cell representing empty space or obstacle (binary) or can store other info: density, %of occupancy, …



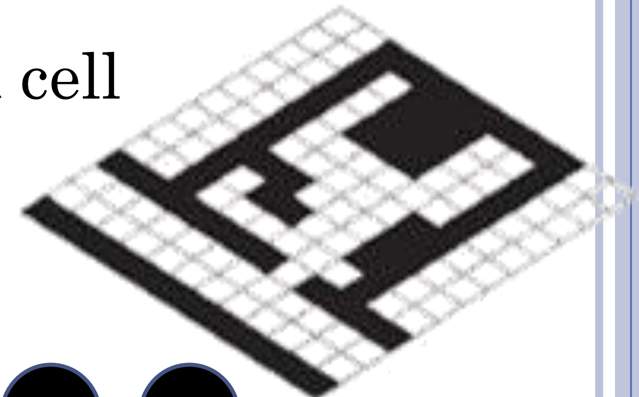
indoors

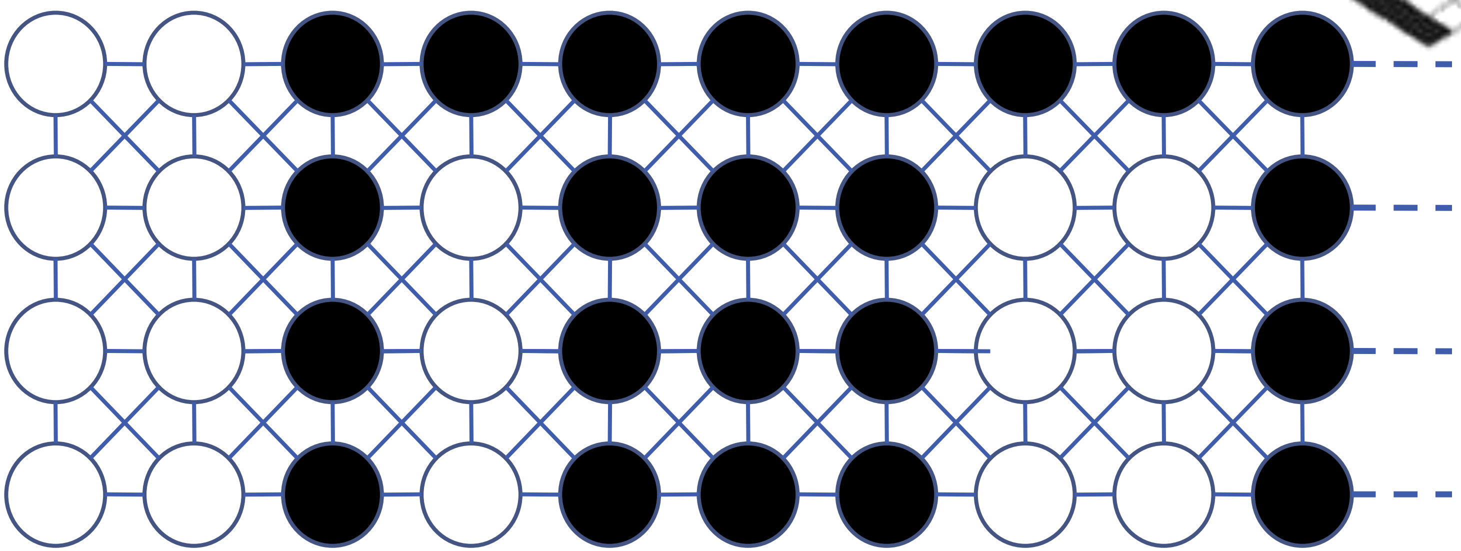


outdoors

3.2-Navigation Meshes
Regular Grids

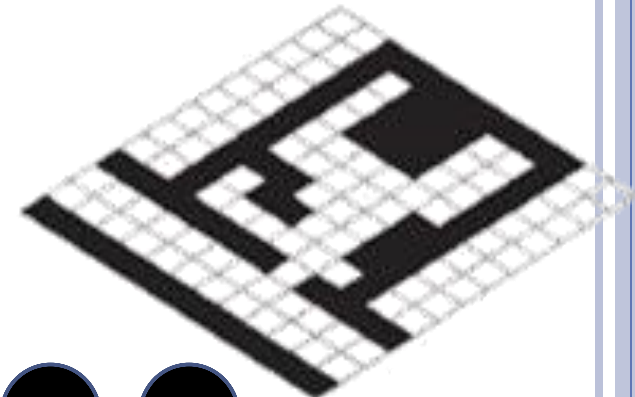
- Each node in the graph represents a cell
- White/Black = empty/obstacle
- Edges represent the connectivity

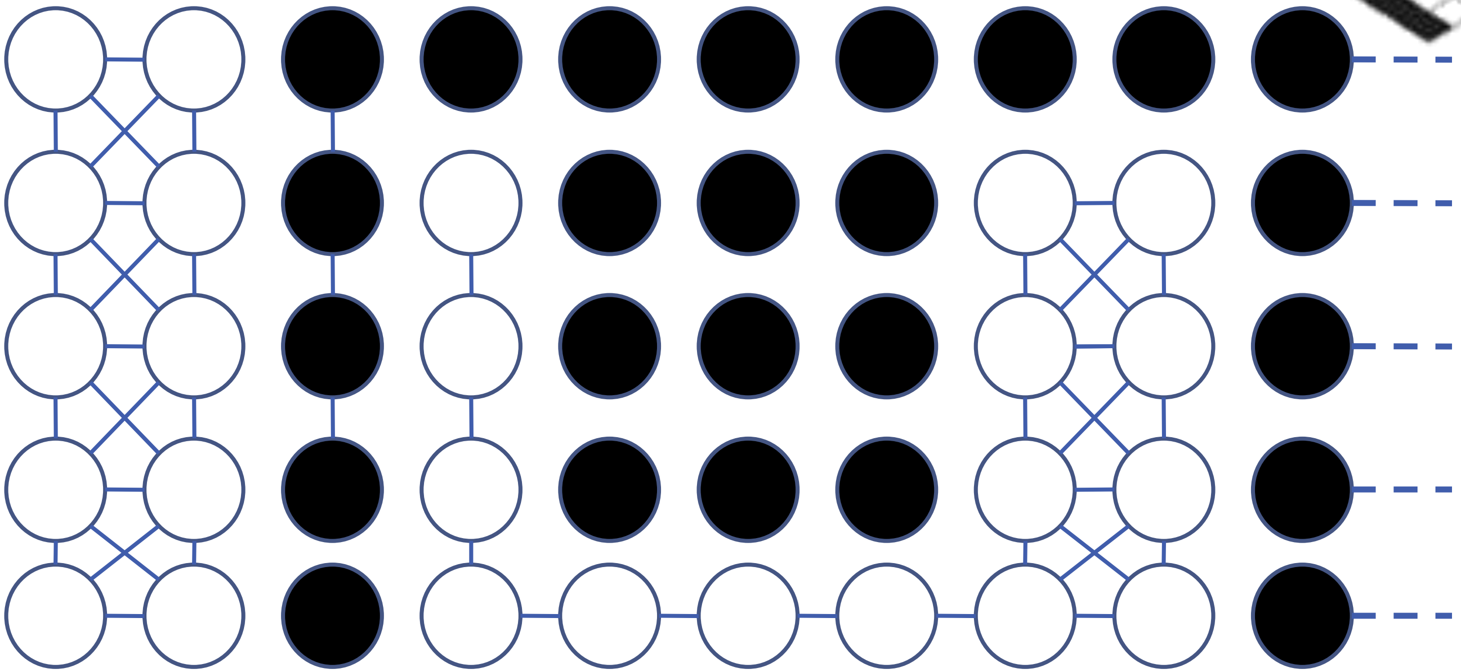




3.2-Navigation Meshes Regular Grids

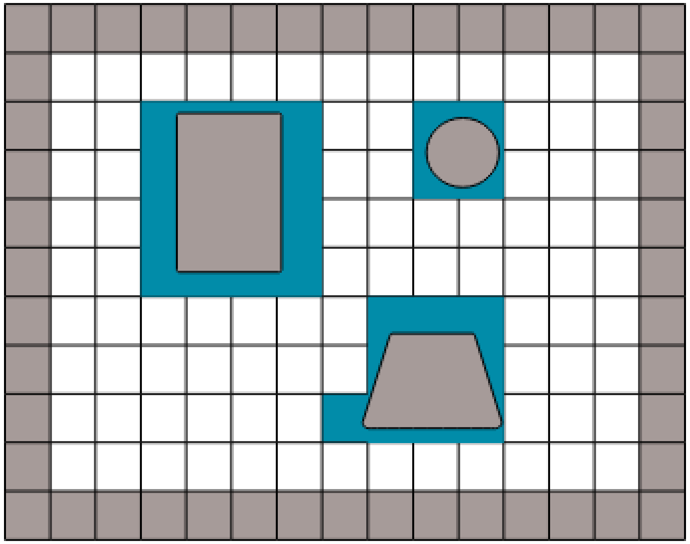
- Remove edges when there is no connection



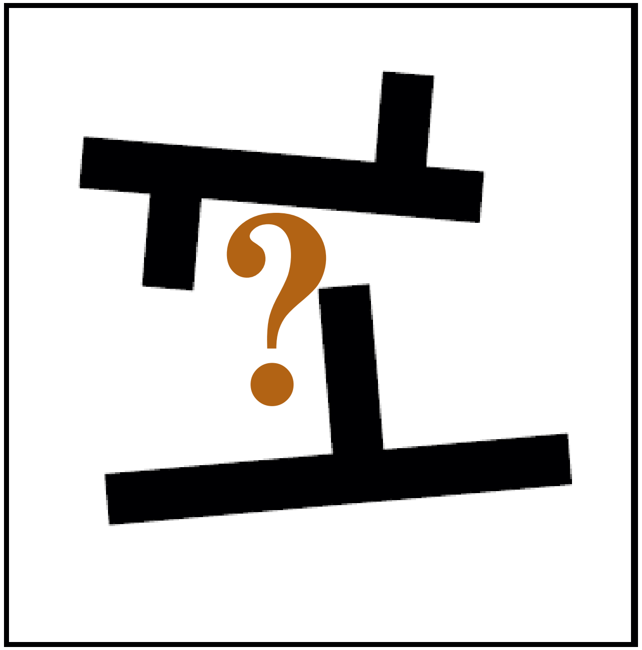


3.2-Navigation Meshes
Limitations of grid methods

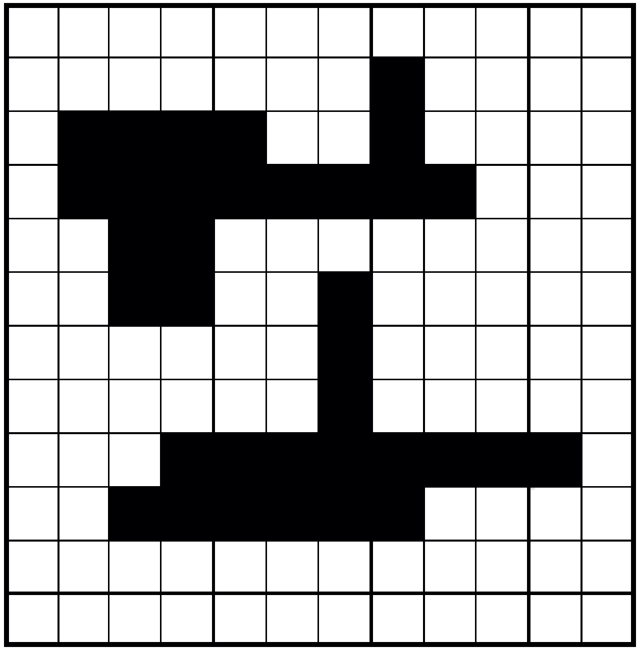
- Coarse coverage of underlying terrain
- Restricted movement (fixed angles)
- Paths don’t look realistic
- Often need to apply smoothing
- Not very useful in 3D spaces



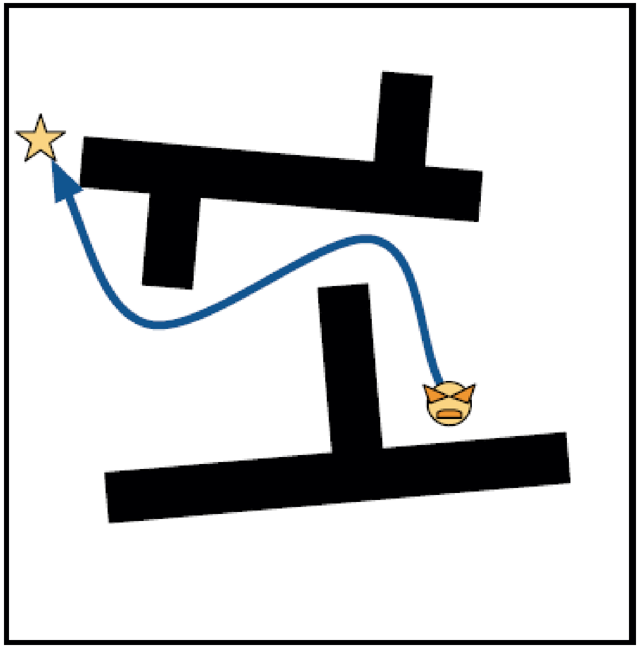
3.2-Navigation Meshes
Limitations of grid methods



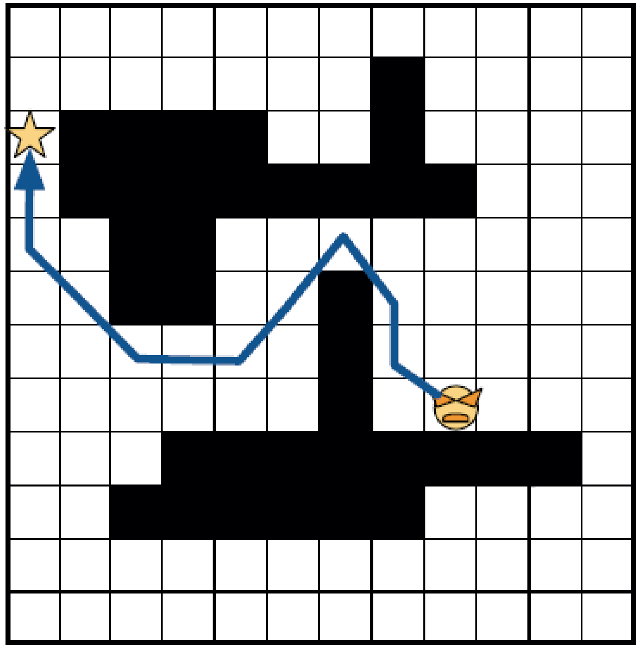
vs.



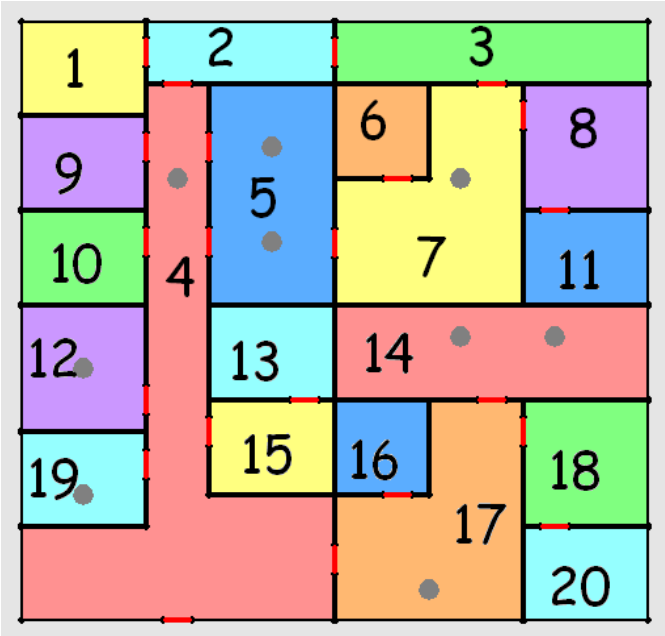
3.2-Navigation Meshes
Limitations of grid methods

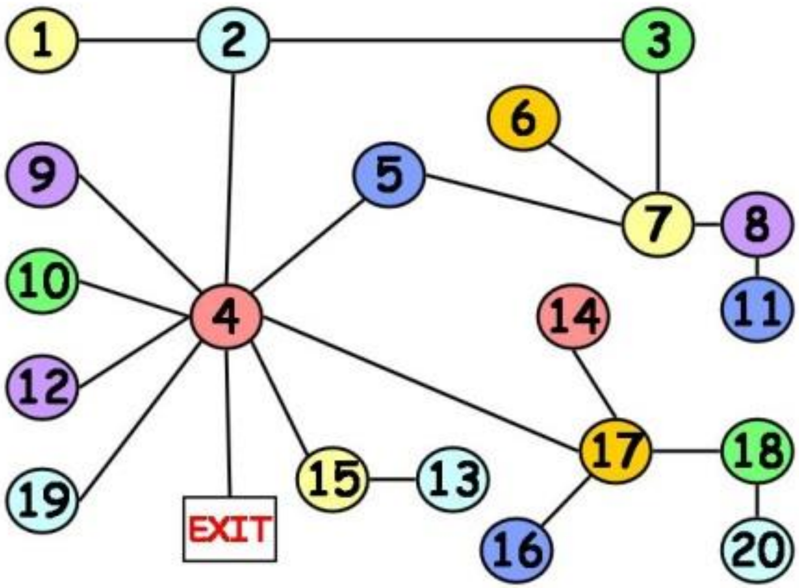


vs.



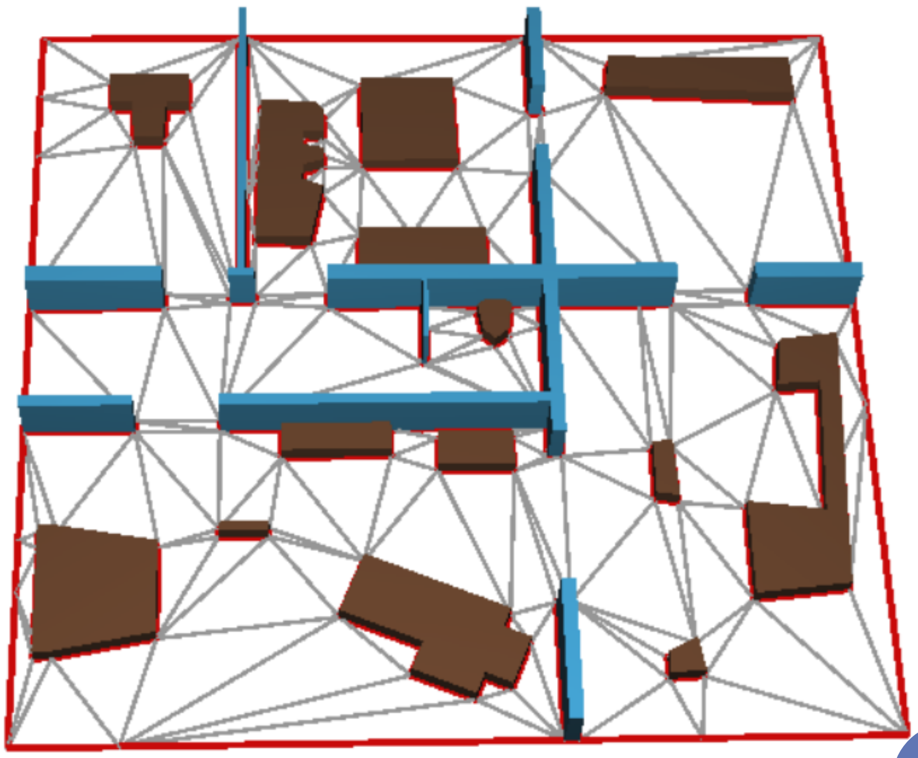
3.2-Navigation Meshes
Cell and Portal Graphs (CPG)





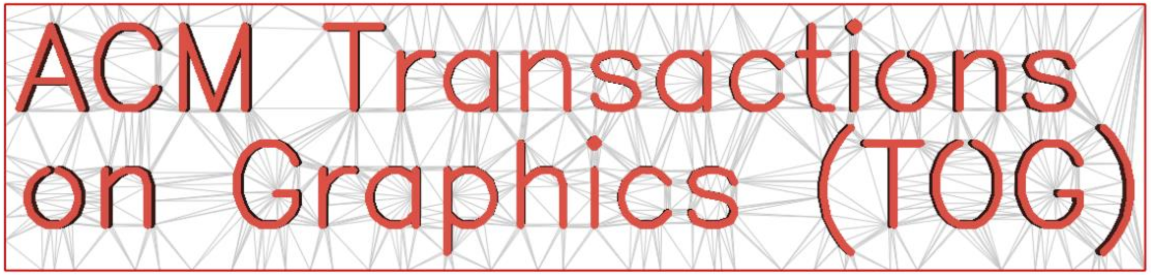
3.2-Navigation Meshes
Triangle Meshes

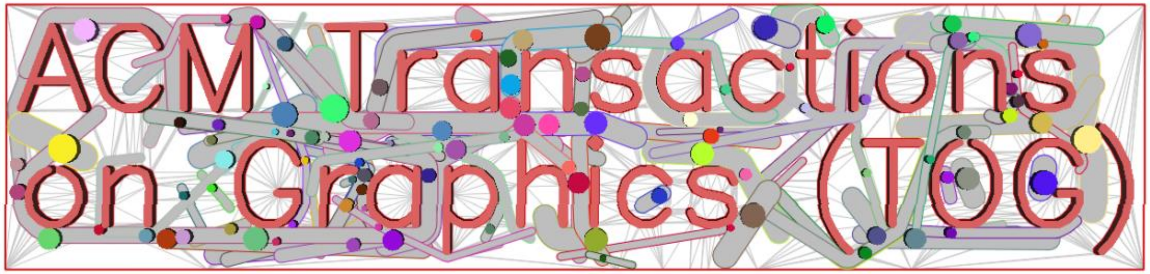
- Triangles forming the navmesh cover the entire walkable geometry



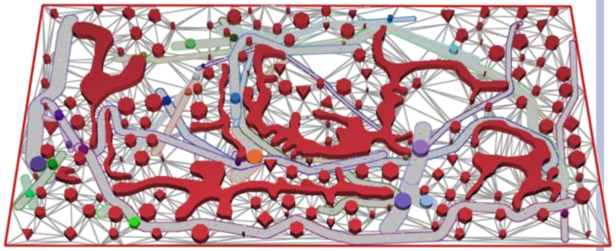
3.2-Navigation Meshes
Triangle Meshes

- Delaunay triangulation speeds up clearance calculations
- Clearance = determines if an agent of a specific size / radius can fit through a cell / portal





Local Clearance Triangulation (LCT) (Kallmann 2010)

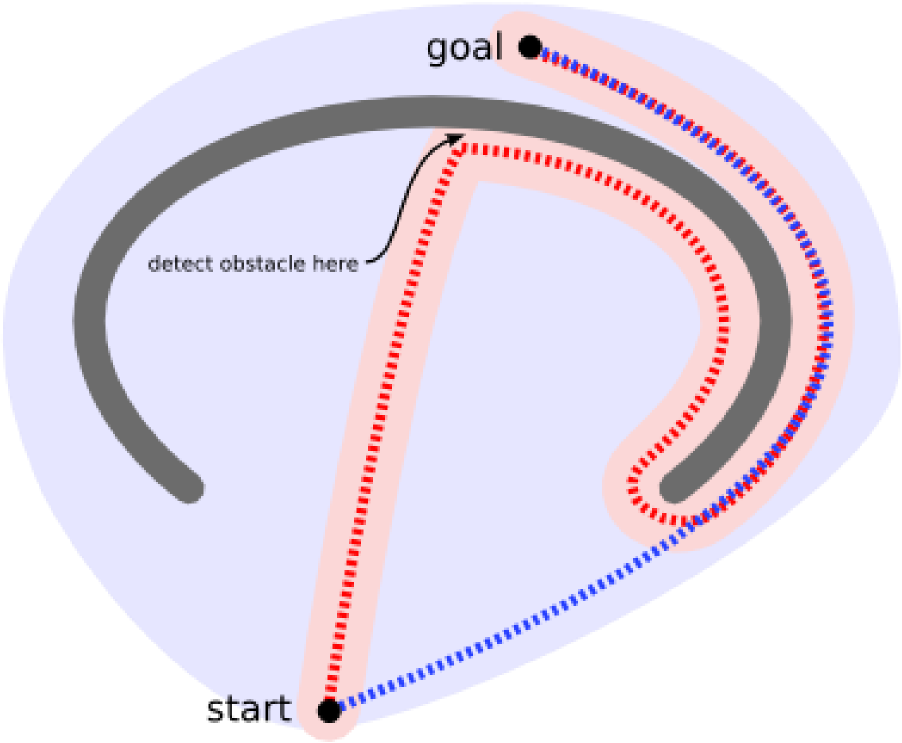


3.2-Navigation Meshes
Triangle Meshes

- Limitations:
  - Still too many cells…
- You could merge triangles together into convex regions

3.3-PATHFINDING

- Why do we need pathfinding?

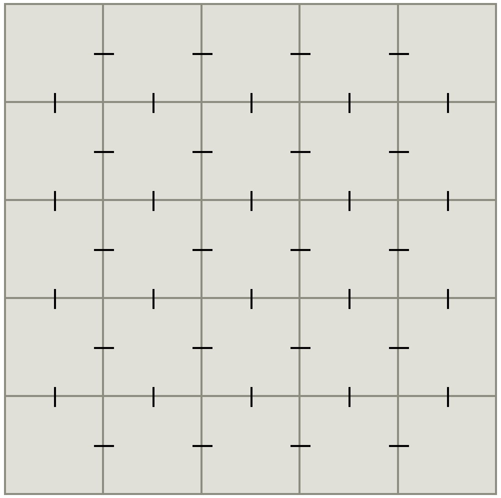


3.3-PATHFINDING

- Look ahead and make plans
  - Not waiting until the last moment to discover there's a problem.
- Tradeoff between planning with pathfinders and reacting with movement algorithms.
  - Planning → generally slower → gives better results
  - Movement → generally faster →can get stuck
- Game world changing often → planning ahead is less valuable.
- Recommended:
  - pathfinding for big picture, slow changing obstacles and long paths
  - movement for local area, fast changing, and short paths.

3.3-PATHFINDING

- Algorithms
  - Work on graphs
  - We will start with a 2D grid
    - But algorithms work with any navigation meshes
  - Consider each cell a node of the graph and each edge between cells an arch (portal, connection,…)

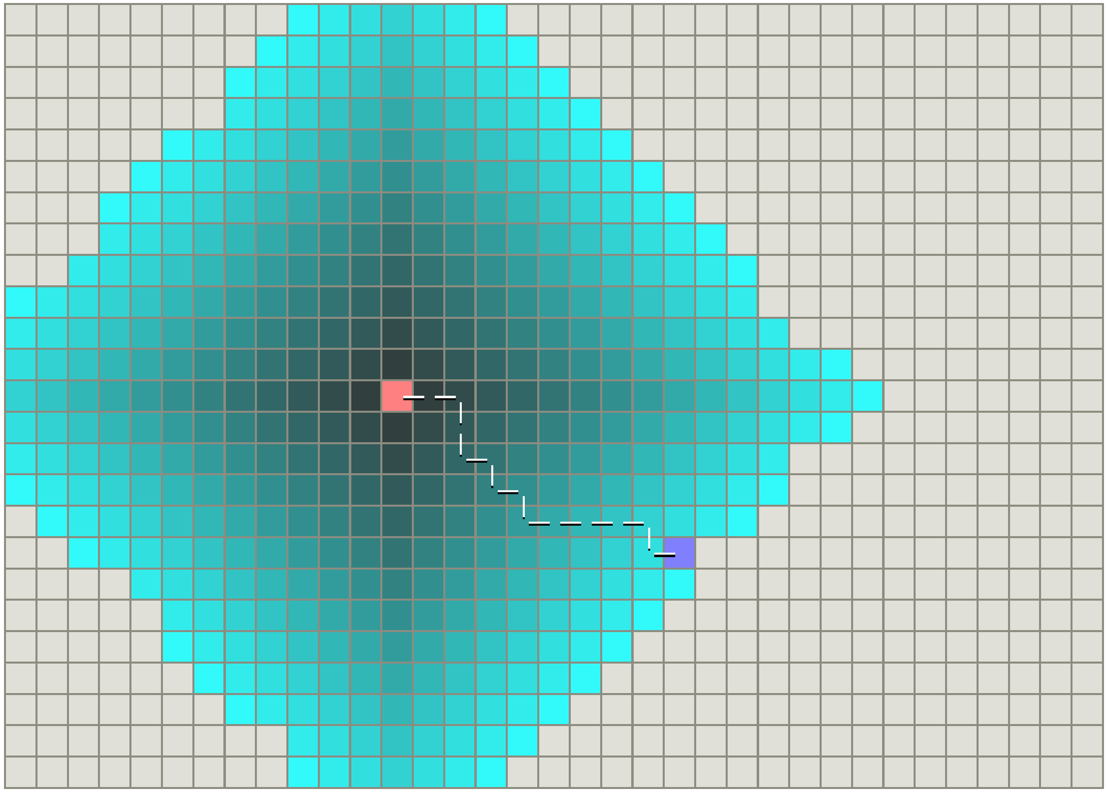


3.3-Pathfinding Dijkstra's

- Starting node: S
- It iteratively visits the closest not-yet-expanded node, adding its nodes to the set of nodes to be expanded
- It expands outwards from S until it reaches the goal node G
- It is guaranteed to find the shortest path
- But… it is very inefficient

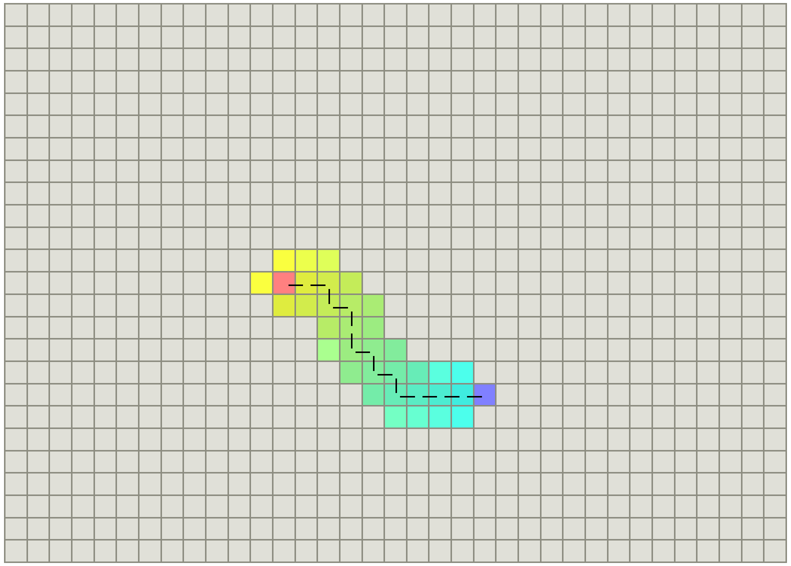
3.3-Pathfinding Dijkstra's

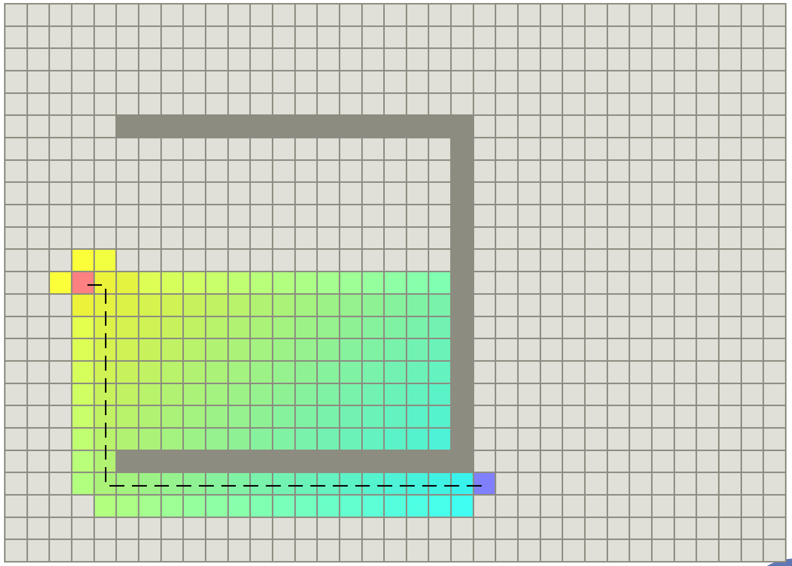
- S: pink node
- G: blue node
- Lightest nodes are those farthest from S



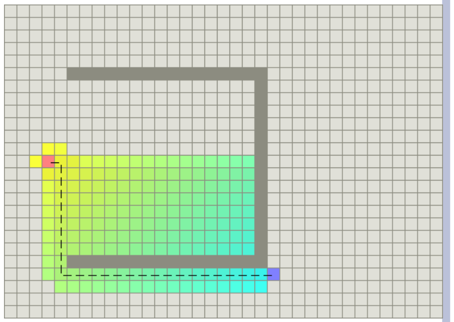
3.3-PATHFINDING
A*

- It can find a shortest path since it is based on Dijkstra, and it is fast as BFS because it is driven by heuristics.



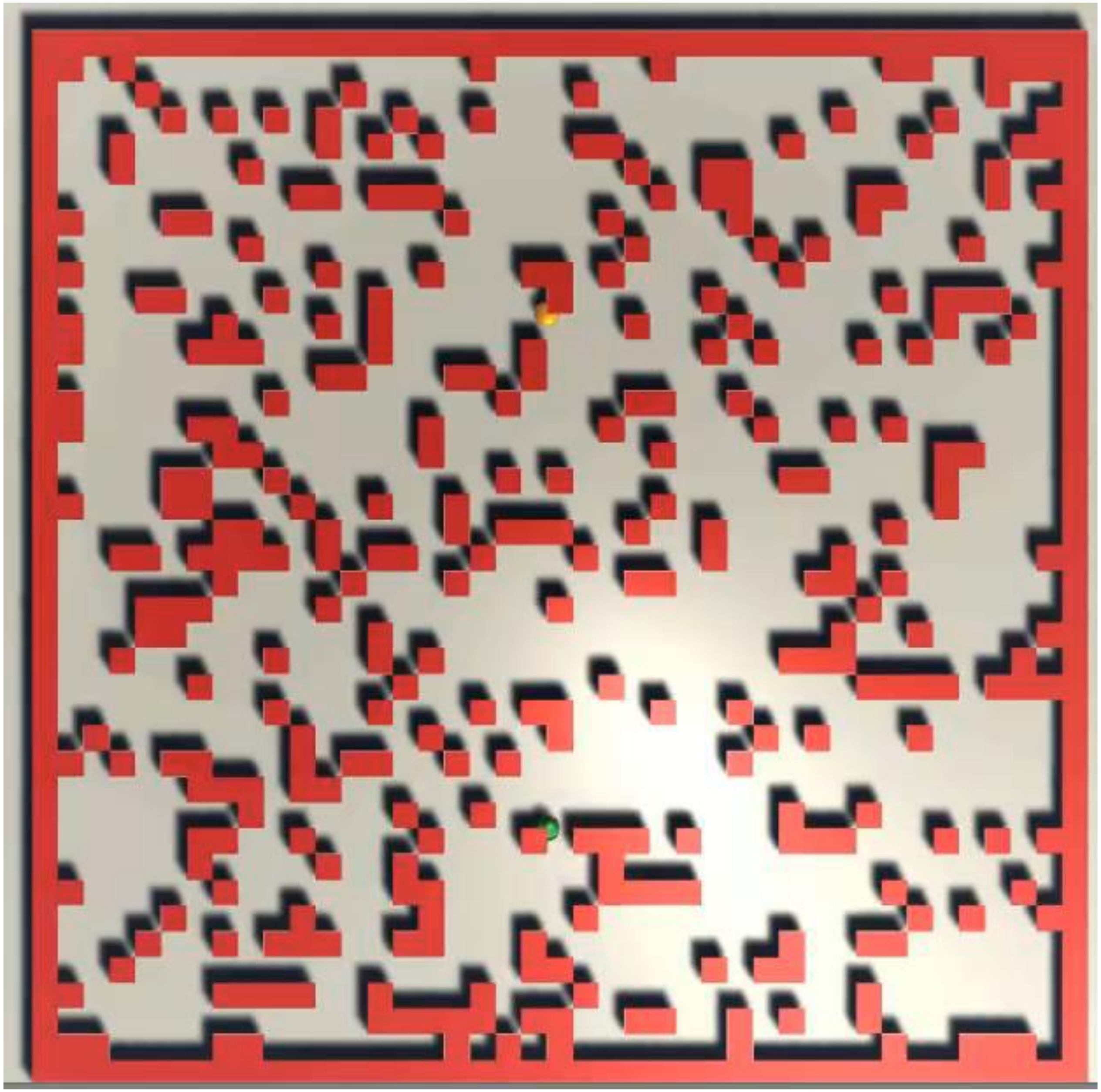


3.3-Pathfinding
A*



- In A*, *g(n)* represents the exact cost of the path from *S* to any node *n*, and *h(n)* represents the heuristic (estimated cost) from node *n* to *G*
- *A** balances the two as it moves from S to G
- At each iteration it expands the node *n* with the lowest $f(n)=g(n)+h(n)$

3.3- Pathfinding A*



3.3-Pathfinding
A* References

- http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html

3.3-Pathfinding Variants of A*

- Why do we need variants?
  - Efficient path planning in very large environments or for large number of agents or different sizes or dynamic environments is NOT SOLVED!

Heritage of Kings (Blue Byte Software)

Company of Heroes (Relic Entertainment)

Red Alert 3 (Electronic Arts)

3.3-Pathfinding Variants of A*

- Bidirectional Search
- Incremental Planners
- Jump Point Search
- ARA* (Anytime A*)
- Hierarchical path finding (HPA*, HNA*)

3.3-Pathfinding Recommended Tutorial/courses

Siggraph 2014:

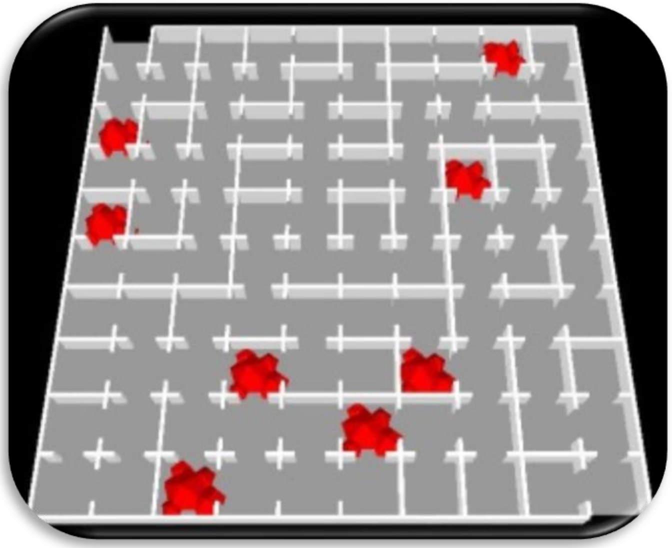
Navigation Meshes and Real-Time Dynamic Planning for Interactive Virtual Worlds

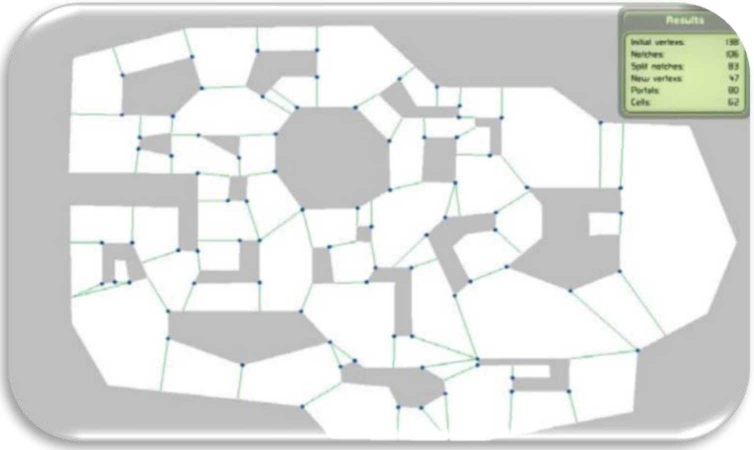
Marcelo Kallmann and Mubbasir Kapadia

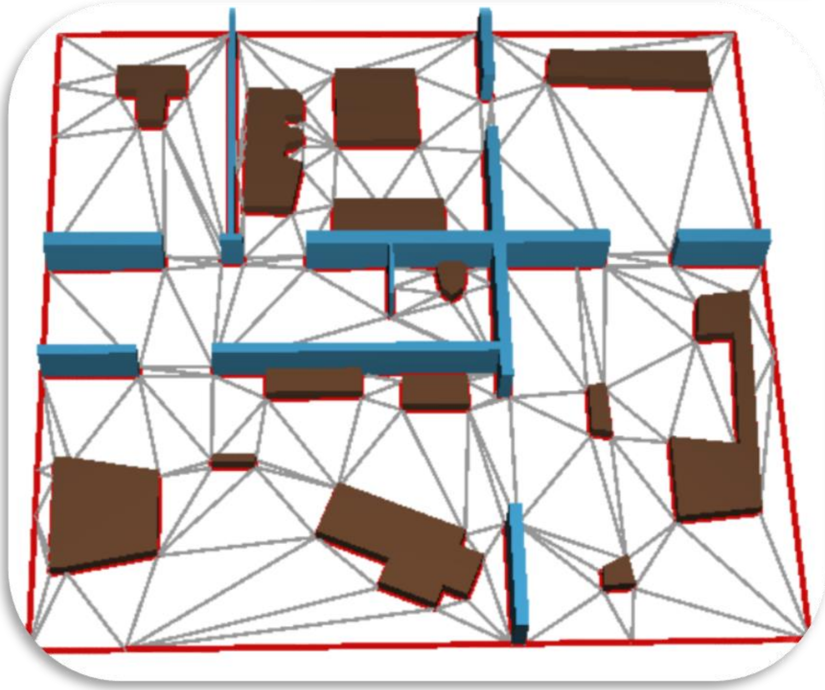
http://www.cs.rutgers.edu/~mk1353/2014-siggraph-course-navigation.html

3.4-Steering

- Microscopic models
- How to move within cells of the navigation mesh
- Local movement

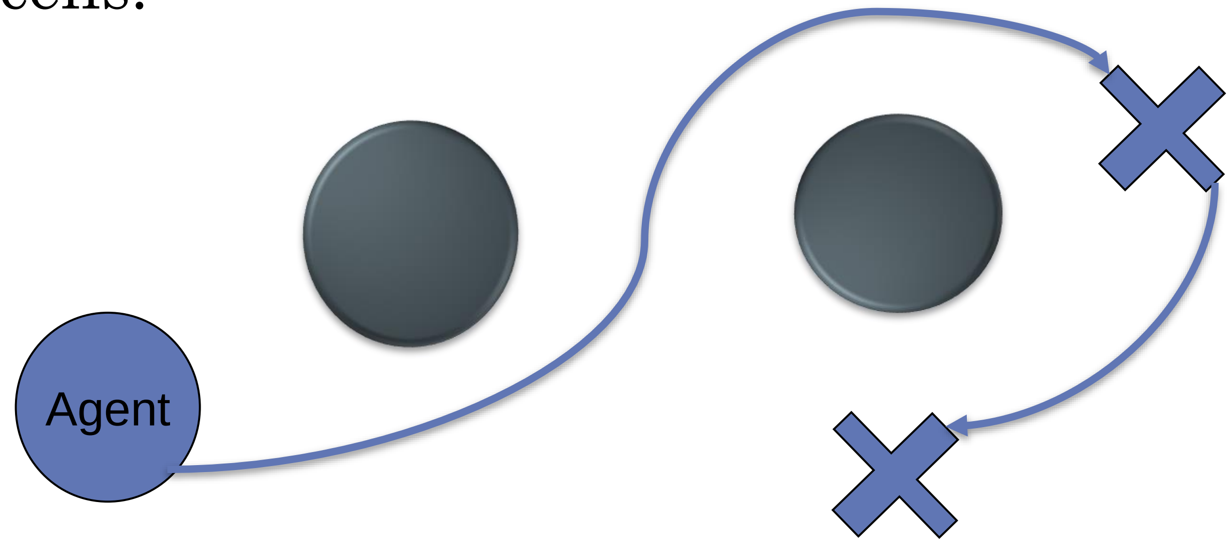




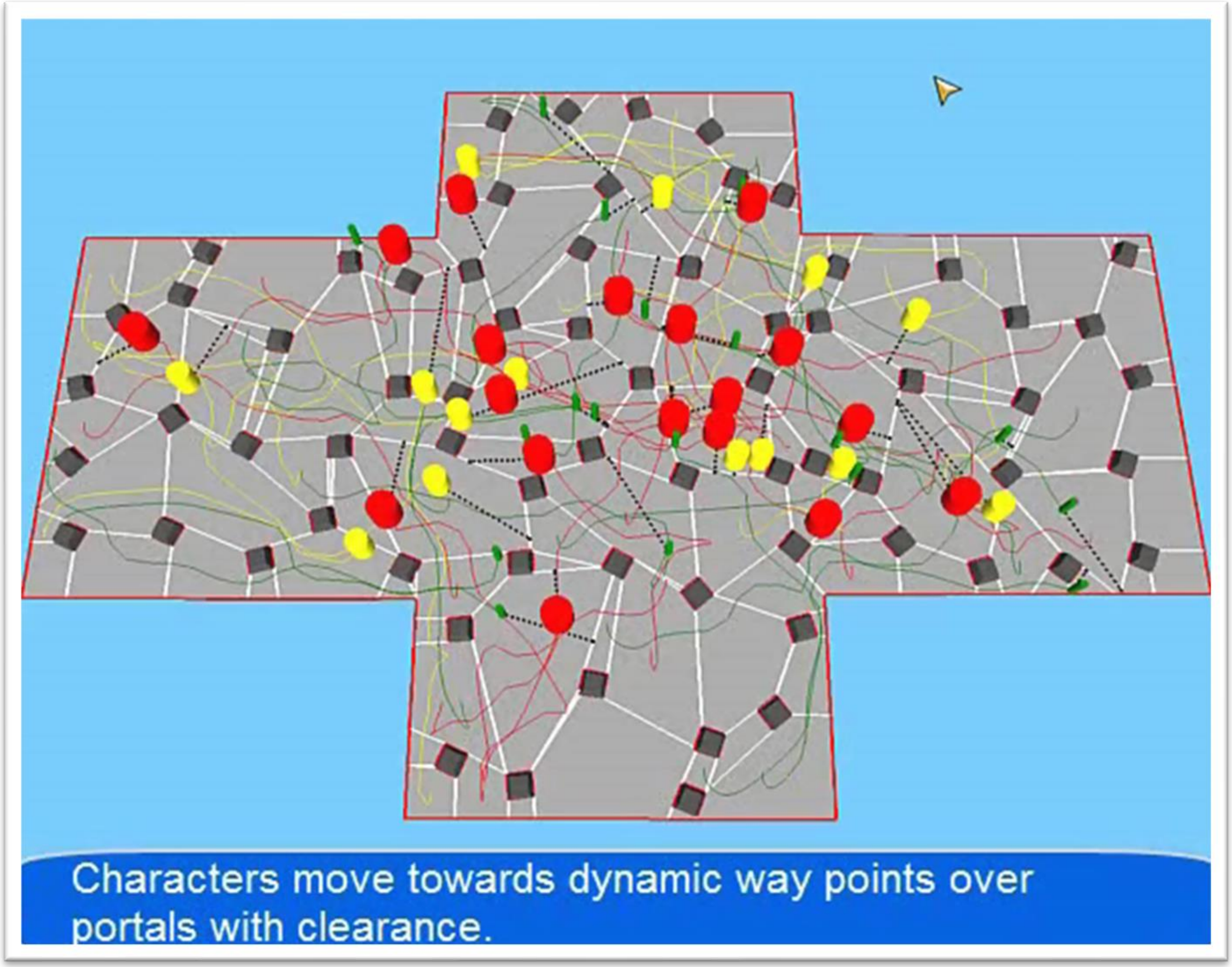


3.4-Steering
Local Movement

- How to move locally avoiding obstacles and other agents?
- How to move between waypoints?
- Usually waypoints set at portals, or at center of cells.



3.4-Steering
Local Movement

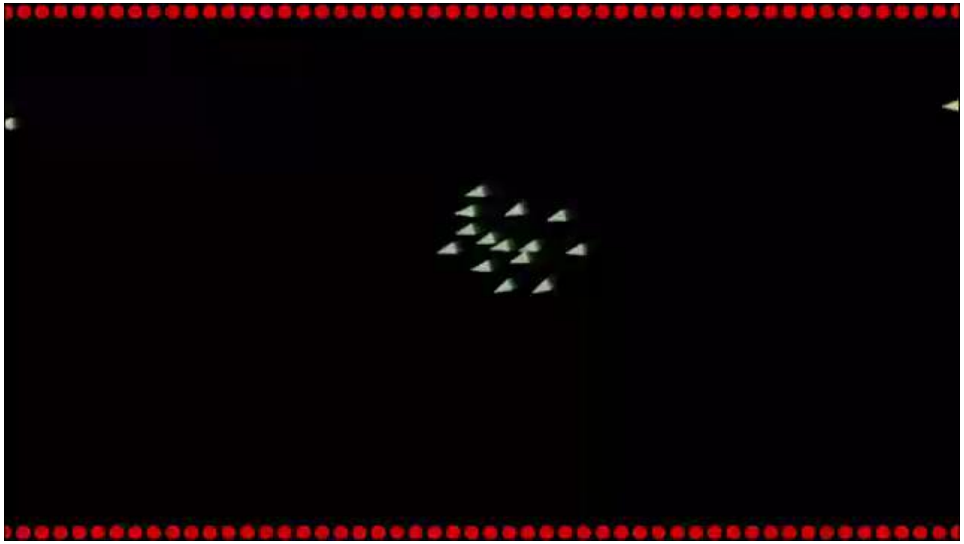


3.4-Steering

- Steering behaviors: influence the character's movement by adding forces → *steering forces*
- Depending on those forces, the character will move in one or another direction.
- Why? It looks more natural than just straight trajectories

https://www.youtube.com/watch?v=QbUPfMXXQIY&t=73s

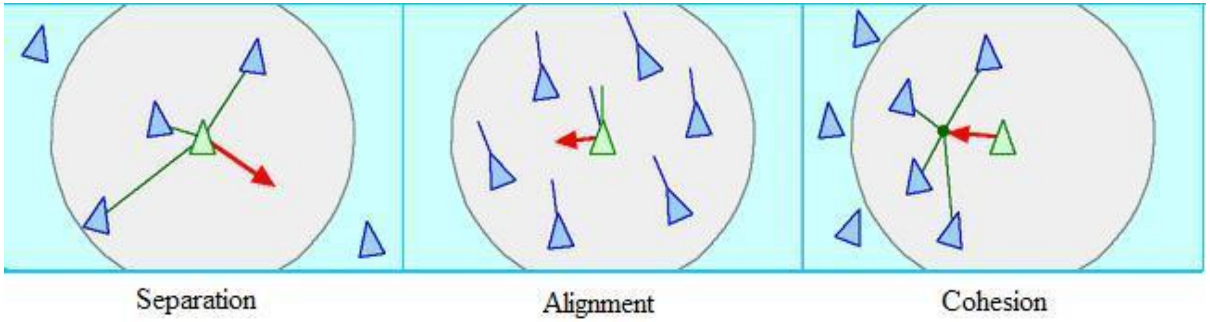
3.4-Steering Reynolds (Rule Based models)



Reynolds C.: *Flocks, herds, and schools: A distributed behavior model.* SIGGRAPH 1987 https://www.youtube.com/watch?v=QbUPfMXXQIY&t=73s

3.4-Steering Reynolds (Rule Based models)

- Reynolds's local rules model (boids) (1987):
  - Distributed behavioral model, where each agent is simulated as an independent actor that navigates according to its local perception of the dynamic environment, the laws of simulated physics that rule its motion and a set of behaviors.

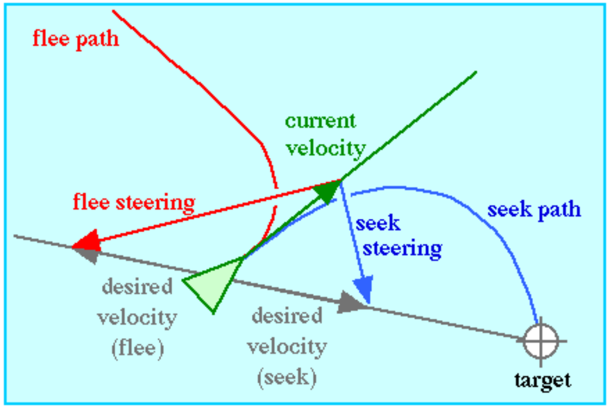


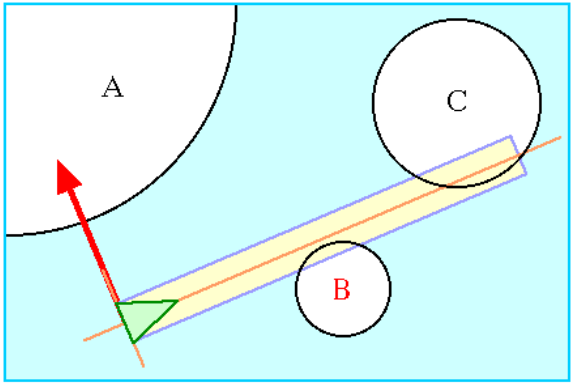
Reynolds C.: *Flocks, herds, and schools: A distributed behavior model.* SIGGRAPH 1987

3.4-Steering Reynolds (Rule Based models)

http://www.red3d.com/cwr/steer/gdc99/

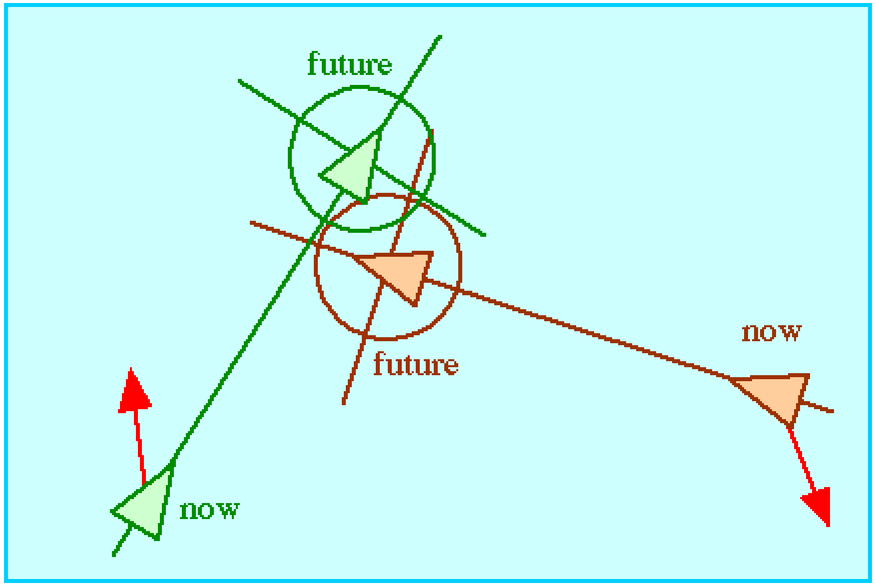
- Simple behaviors for individuals and pairs:
  - Seek and Flee
  - Pursue and Evade
  - Wander
  - Arrival
  - Obstacle Avoidance
  - Containment
  - Wall Following
  - Path Following
  - Flow Field Following





3.4-Steering
Reynolds (Rule Based models)

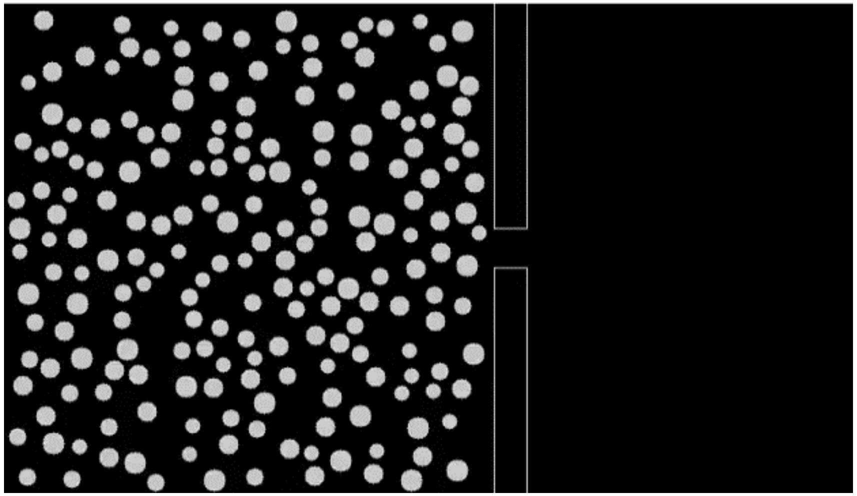
- Combined behaviors and groups:
  - Crowd Path Following
  - Leader Following
  - Unaligned Collision Avoidance
  - Queuing (at a doorway)
  - Flocking (combining: separation, alignment, cohesion)



Simulating dynamical features of escape panic
D Helbing, I Farkas, T Vicsek. Nature 407 (6803), 487-490

3.4-Steering
Helbing (social Forces)

- Social forces model describe human crowd behavior with a mixture of socio-psychological and physical forces.
  - Repulsive interaction, friction, dissipitation, fluctuation
- Not discretization in time or space.
- Newton's equation for each individual:

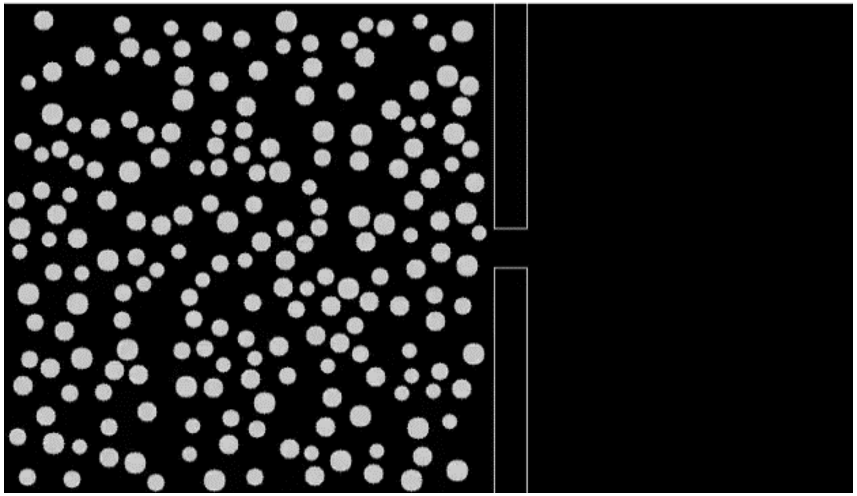


$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \frac{v_i^0(t)\mathbf{e}_i^0(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_{W} \mathbf{f}_{iW}$$

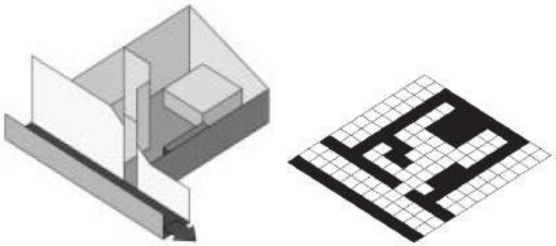
[Simulating dynamical features of escape panic]
D Helbing, I Farkas, T Vicsek. Nature 407 (6803), 487-490

3.4-Steering Helbing (social Forces)

- Limitations
  - Similar to particle animations
  - Agents appear to shake unnaturally in high-density crowds



3.4-Steering Cellular Automata



- Cellullar Automata models (CA)
- Regular uniform grid with a discrete variable at each cell.
- Evolution takes place in discrete time steps
- In a CA a finite set of local rules are used to describe the intelligent decision-making behavior.
- The emergent group behavior is a result of the interactions of the local rules as each pedestrian searches the available cell in its neighborhood.

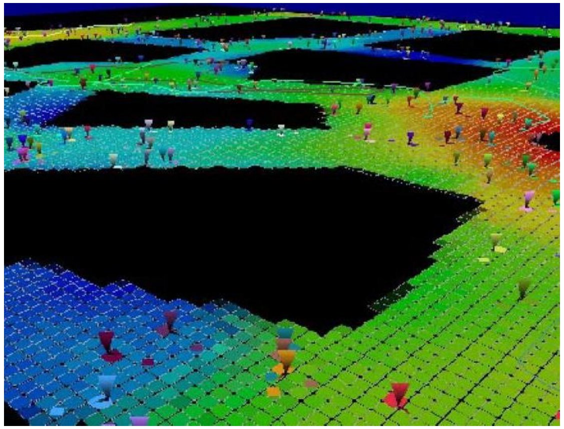
Blue and Adler 2000 (Cellular Automata Model Of Emergent Collective Bi-Directional Pedestrian Dynamics, Artificial Life VII. The Seventh International Conference on the Simulation and Synthesis of Living Systems)

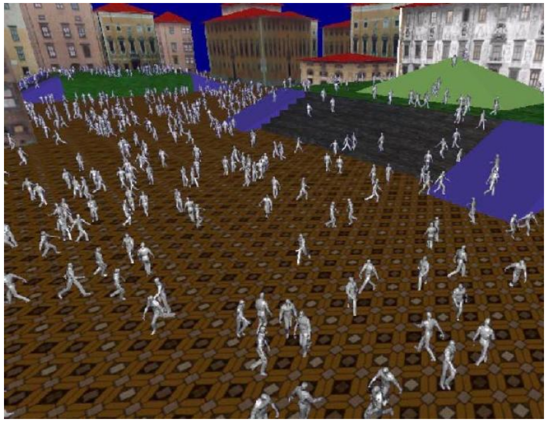
Other examples: Kirchner (2003), Dijkstra (2000), Chenney (2004), Tecchia (2001)

3.4-Steering Cellular Automata

Intuitive crowd behavior in dense urban environments using local laws. C Loscos, D Marchal, A Meyer. Proc. Theory and Practice of Computer Graphics, 2003., 122-129

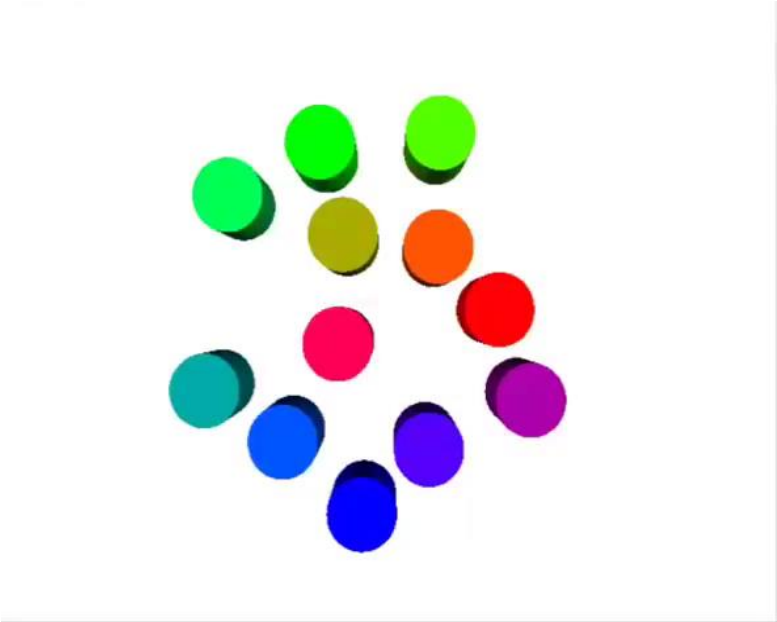
- Discretized space → agents only moves to adjacent free cells
  - No contact between agents
  - Relistic for low density crowds only
- Paths in the grid can be obtained by pre-computing paths towards goals and storing them within the grid (Loscos, Marchal et al. 2003).



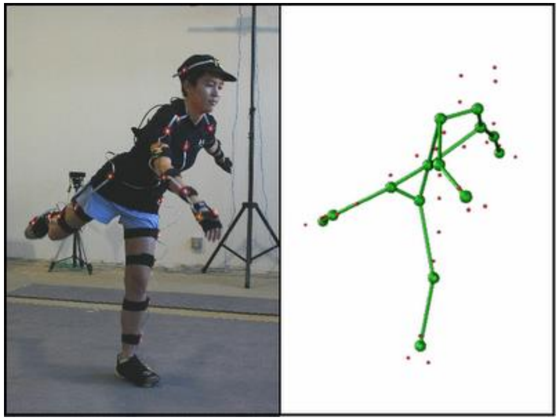


3.4-Steering
Velocity-Based Models

- Reciprocal Velocity Obstacles (RVO)
- Choose a velocity which satisfies ALL pairwise constraints
  - Efficient O(n) implementation with Linear Programming
- Library Online: http://gamma.cs.unc.edu/RVO2/



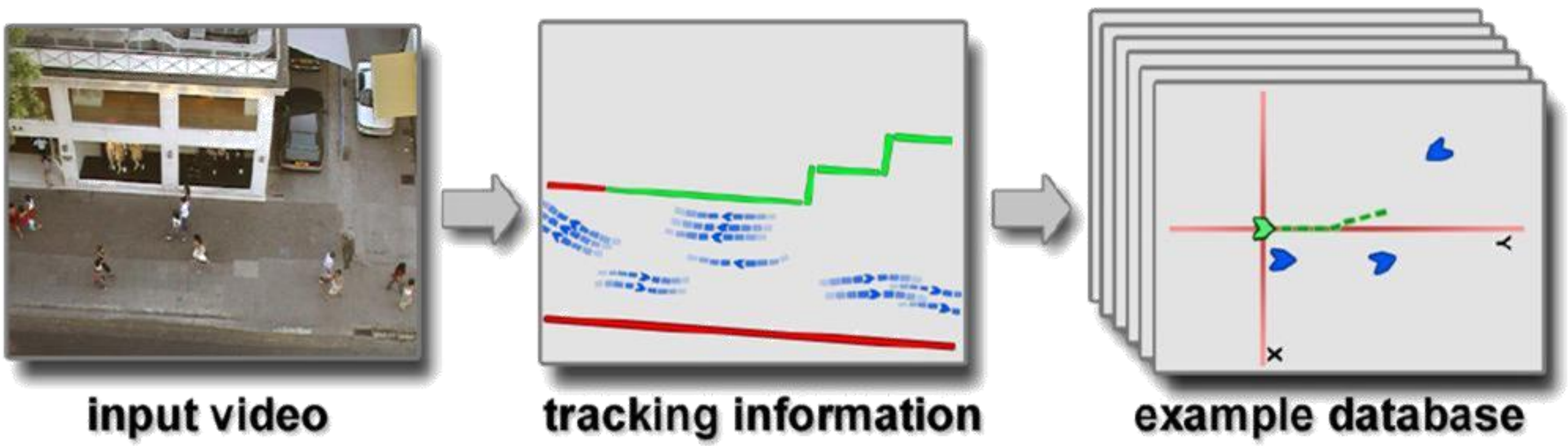
3.4-Steering
Data Driven Crowds



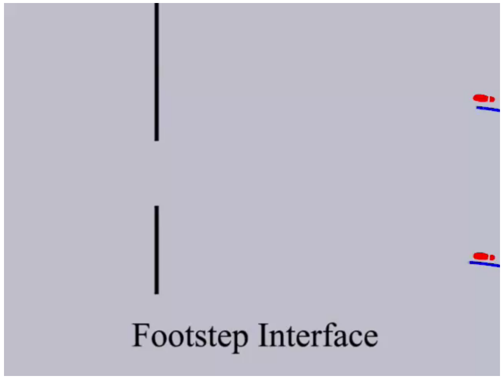
- Data driven algorithms have been used in Computer Animation
  - Motion capture single characters
  - Facial motion capture
  - Even trees…
- In crowds the emphasis is on multi-character interactions
  - Capture subtle actions

3.4-Steering
Data Driven Crowds

- Crowds by Example
- Data-driven approaches use learning algorithms based on locally weighted linear regression to simulate crowd behavior (Lee, Choi et al. 2007).



3.4-Steering Footstep-Driven Approaches



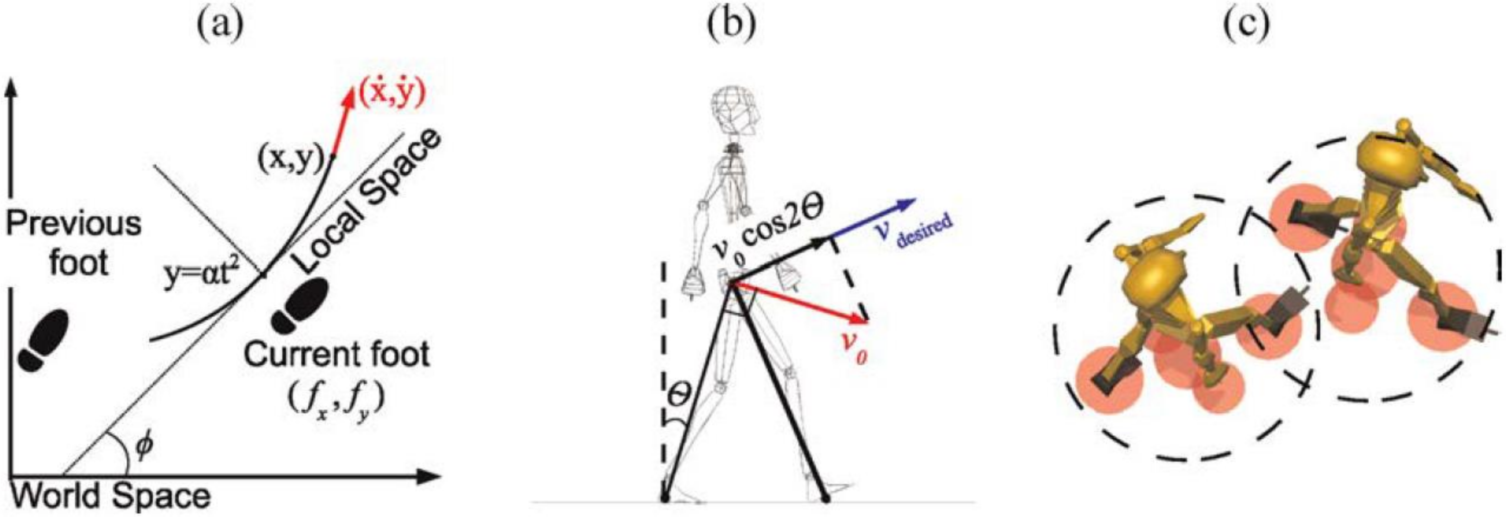
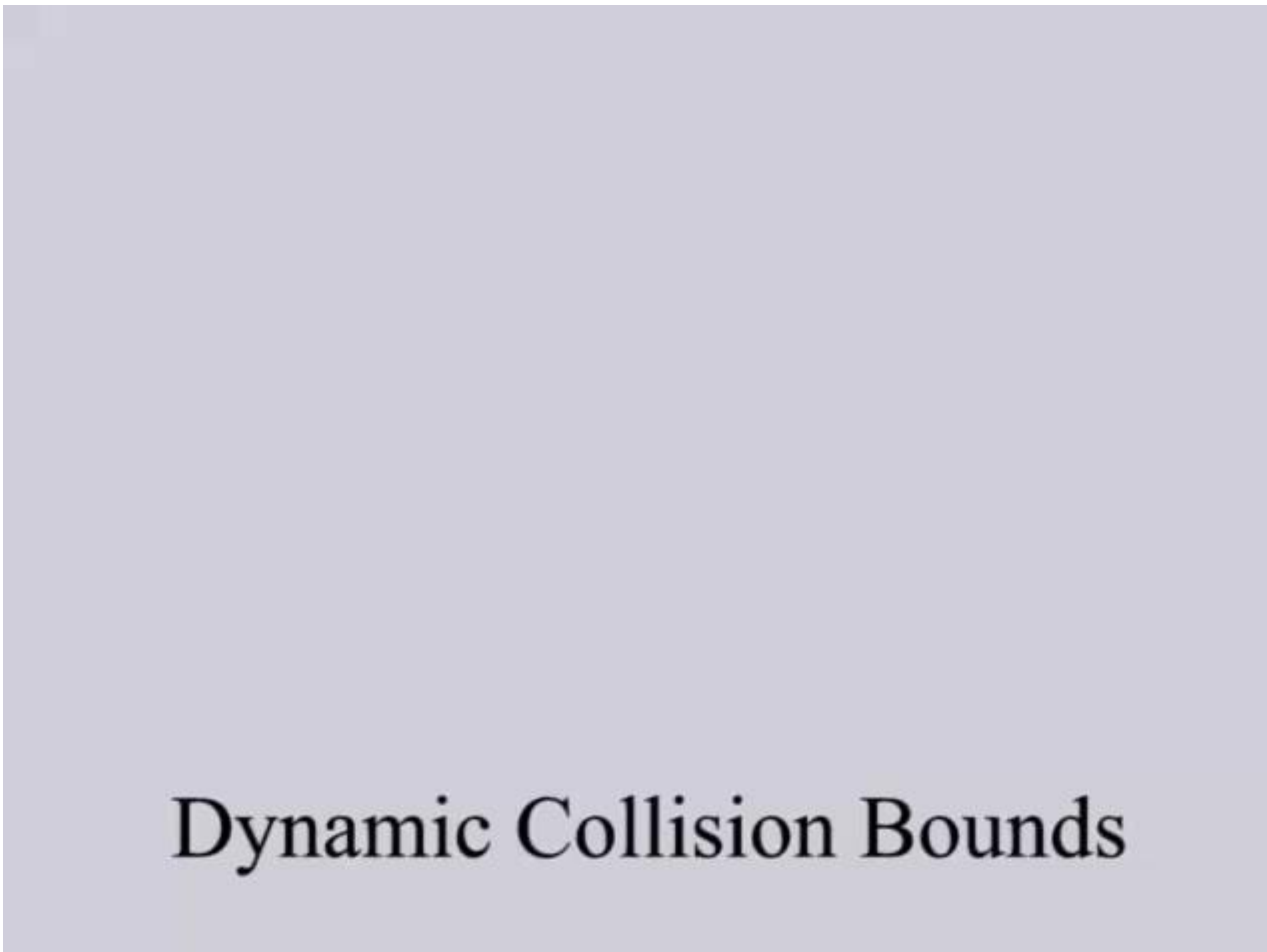


FIGURE 3.8: A footstep navigation model. (a) Depiction of state and action parameters. (b) A sagittal view of the pendulum model used to estimate energy costs. (c) The collision model uses five circles that track the torso and feet over time, allowing tighter configurations than a single coarse radius. [Singh et al., 2011]

Singh et al. Footstep navigation for dynamic crowds. Computer Animation and Virtual Worlds. 2011

3.4-Steering
Footstep-Driven Approaches



3.4-Steering
Recommended Tutorial/courses

Eurographics 2014:

Simulating heterogeneous crowds with interactive behaviors.
N. Pelechano, M. Kapadia, J. Allbeck, Y. Chrysantyou, S. Guy, N. Badler

http://www.cs.rutgers.edu/~mk1353/2014-eg-crowds-course.html

5-Perception

- 5.1-Perceiving Animations
- 5.2-The Uncanny Valley
- 5.3-Level of Detail
- 5.4-Realism
- 5.5-Artifacts
- 5.6-Variety

5.1-Perceiving Animations

- Animated avatars

- How do we perceive them?
  - When whatching a movie
  - When playing a videogame
  - When inmersed in a VR simulation

- What is perception?
  - Latin *Perceptio*: Gathering or receiving
  - Organization, identification, and interpretation of sensory information in order to represent and understand the presented information or environment
- Illusions

5.1-Perceiving Animations

- Are you that avatar? Is it your representation?
- Is the avatar trying to tell you something?
- Narrative
  - Explicit or passive

- Cartoon vs photorrealistic
- Same for animations

- Can we evaluate animations?
- Are they pleasant enough?
- Do they achieve their purpose?
- Should/Can we compare them to something real? Objective measure?

5.1-Perceiving Animations

- Effects of realism
  - In animation too
- Too much realistic animation → bad?
- Sometimes cartoonish better

- Effects of bad skinning and/or bad skeletal animation quality
- Blending, synthesizing artifacts
- Interpenetrations, collisions, etc…

- Effects of non-smooth, non-fluid animation
- Effects of not enough frames
  - But artistically can work → Spiderverse Movie
- Asynchronous motion → less body ownership

5.1-Perceiving Animations

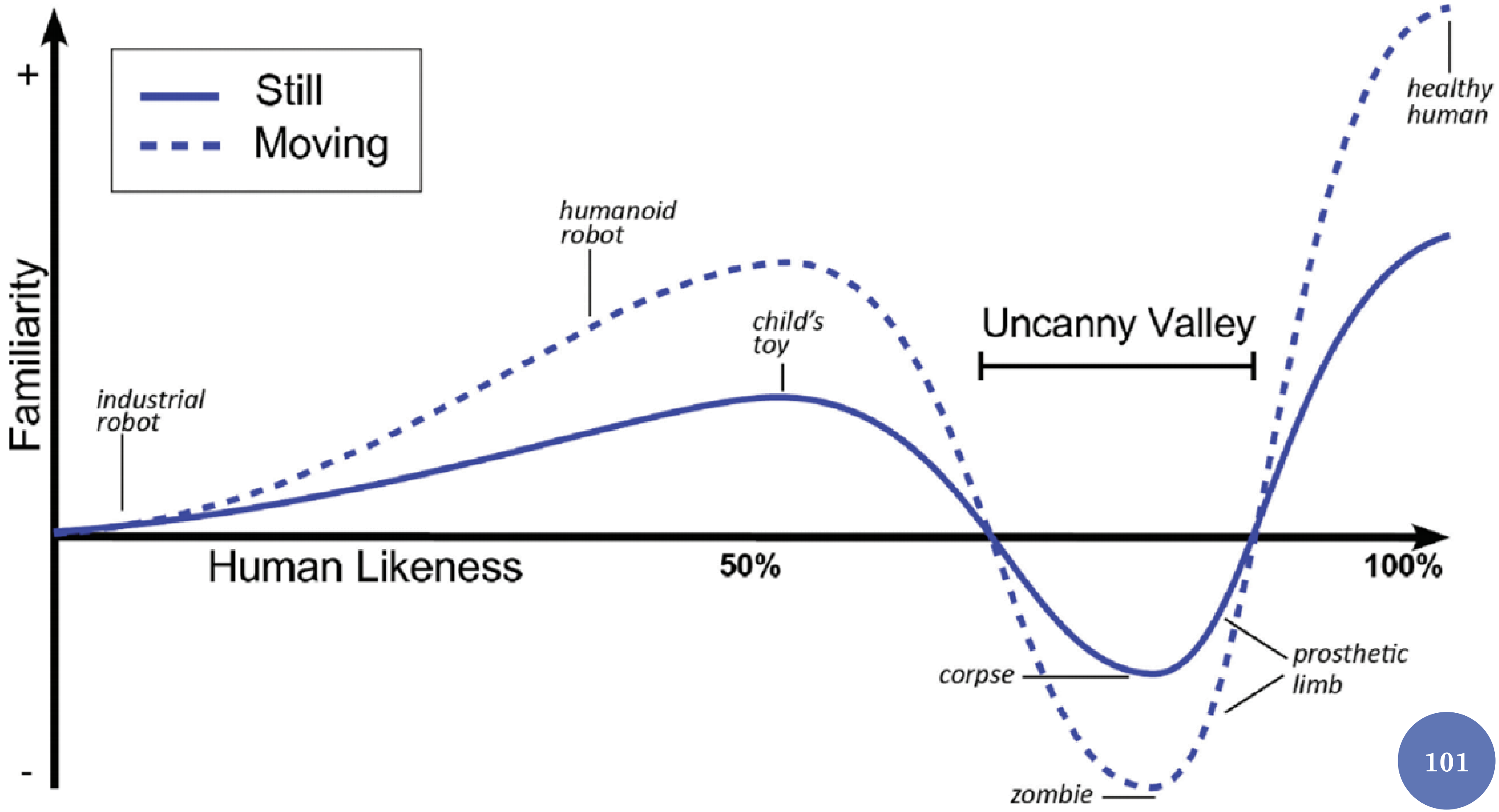
- Animations as behavior perception
- Narratively
- Emotions → living characters
- Animations repetitions, loops → robots, not alive
- Gazing behavior, look ats

- Simulations / crowds realism
- Simulations can be evaluated, compared to real life video data

- Experiments → useful to study the influence of specific factors → tune them

Masahiro. *Bukimi no tani (不気味の谷)*
The Uncanny Valley. Energy 1970

5.2-The Uncanny Valley



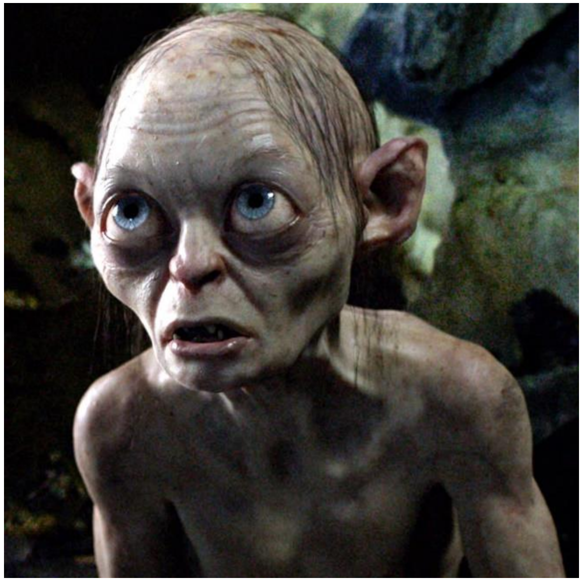
5.2-The Uncanny Valley

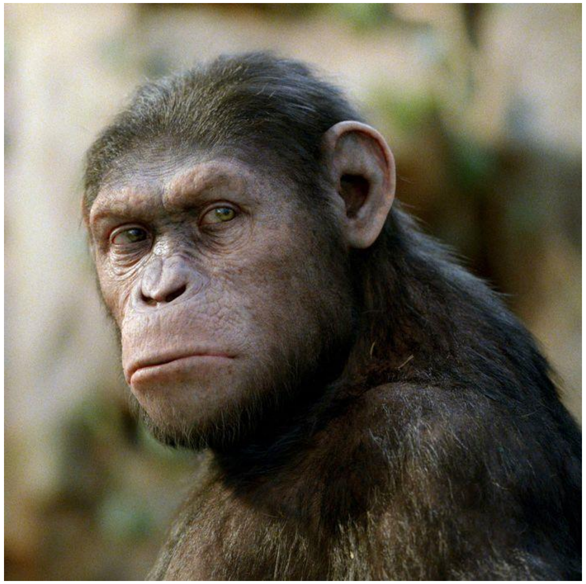
5.2-The Uncanny Valley

5.2-The Uncanny Valley

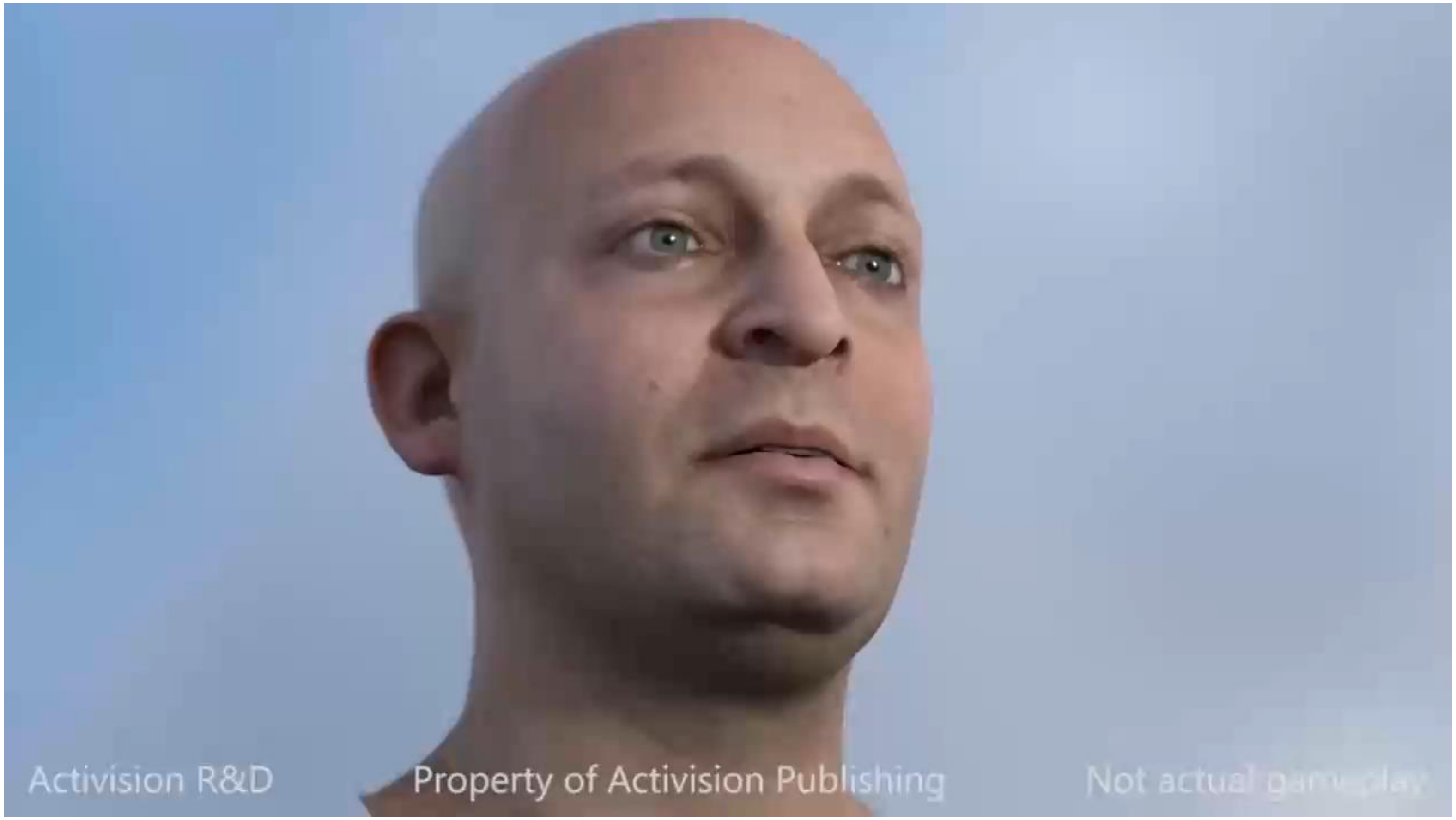
5.2-The Uncanny Valley

5.2-The Uncanny Valley





5.2-The Uncanny Valley



5.2-The Uncanny Valley

5.2-The Uncanny Valley

5.3-Level of Detail

- In crowds with many agents, focus doesn't have to be on all agents at the same time
- We will see better closer agents
- Level of Detail
  - Different geometries for different distances
    - Lower polycount
    - Impostors
  - Can also be applied to motion and behaviour

5.3-Level of Detail

- ALOHA (Adaptive Level of Detail for Human Animation)
  - LODs for geometry, motion, behaviour (conversational and social)

- Far characters
  - Random animations
  - Changed at different intervals
  - → varied activity
- Focused characters (closer or interacting)
  - Meaningful actions chosen
  - More sophisticated behaviour
  - More realistic motions → more accurate

O'Sullivan et al. *Levels of Detail for Crowds and Groups*. Computer Graphics Forum 2002

5.3-Level of Detail

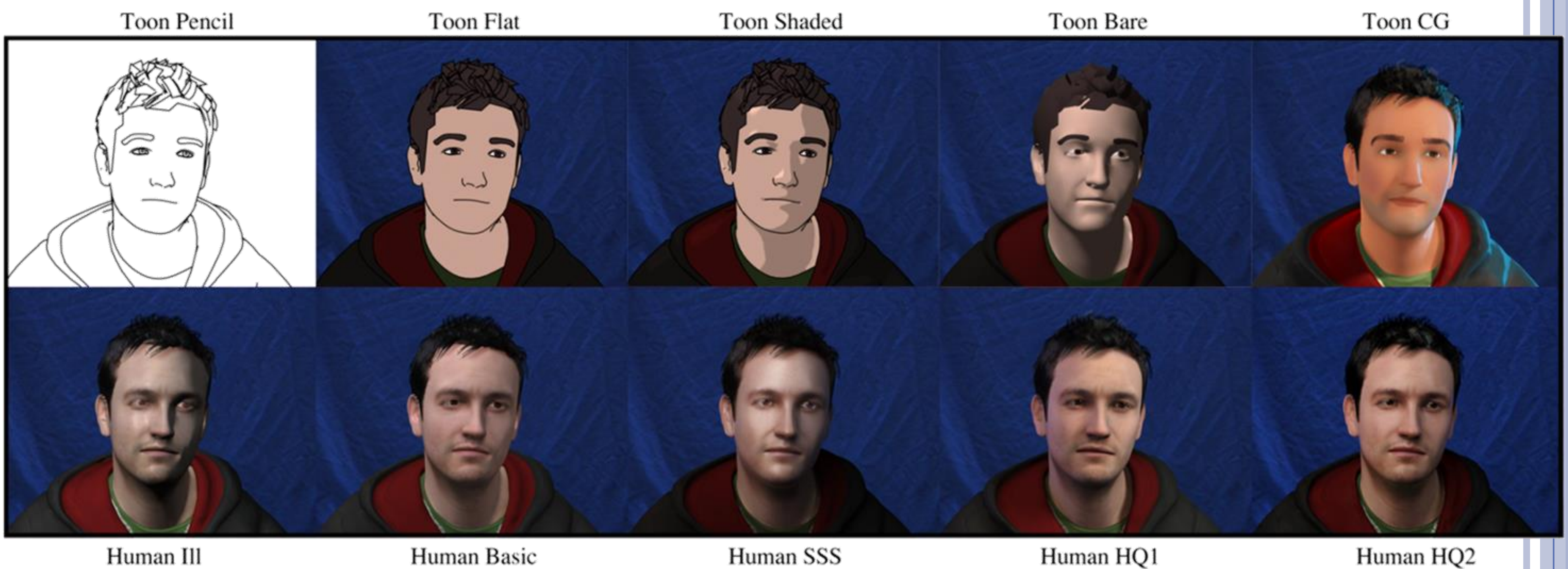
- Behaviour LODs
  - Conversation levels
  - LOD AI → model more general behaviour and motivations at multiple LODs
  - Based on visual and functional properties

- Behaviour generation rules
- Filter Rules → remove generated behaviours based on LOD criteria

- Animation → each behaviour annoates visual salience
  - → selectively drop behaviours as the character moves further away or out of focus

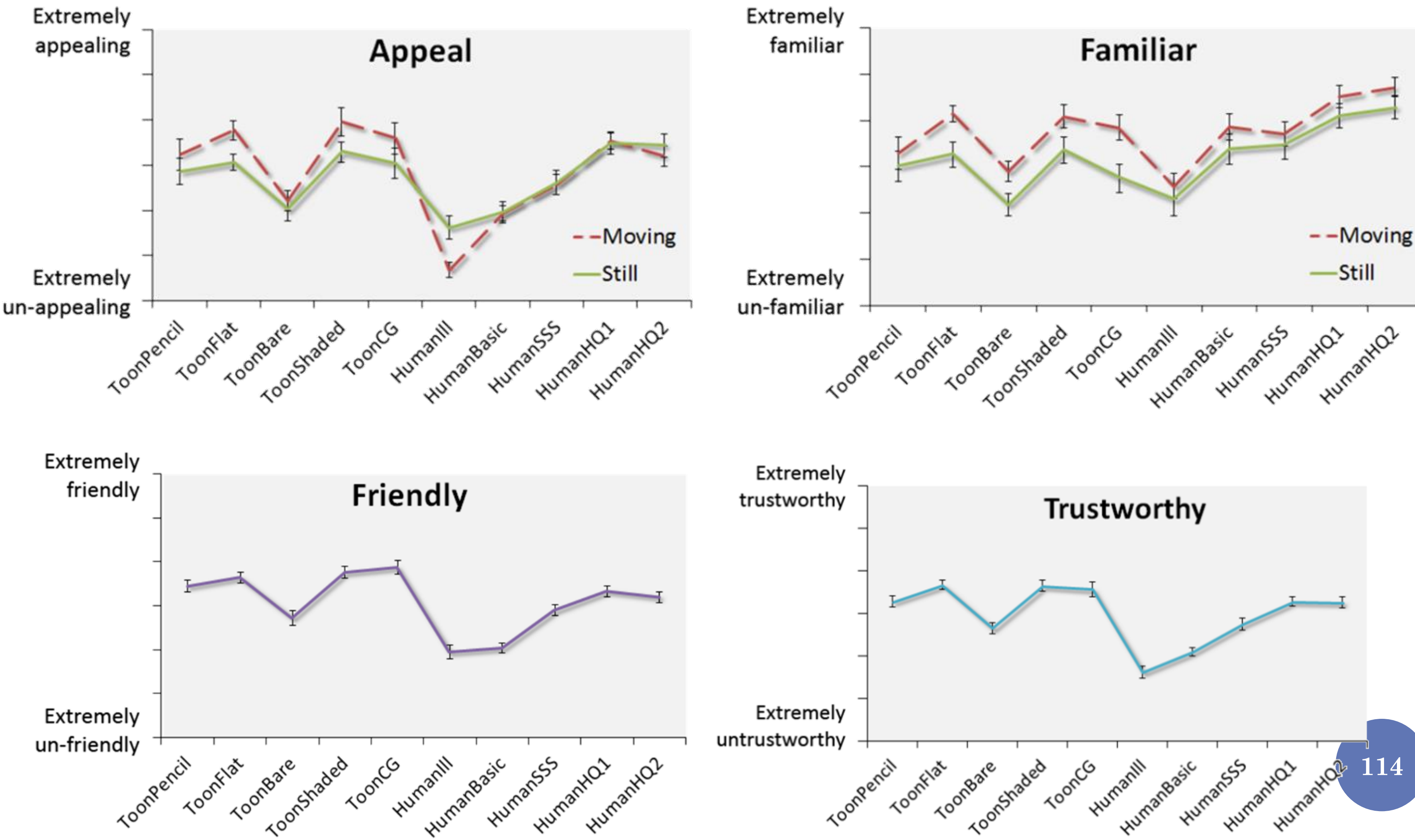
5.4-Realism Render Style

McDonnell et al. *Render me real? investigating the effect of render style on the perception of animated virtual humans*. ACM Transactions on Graphics 2012



5.4-Realism Render Style

McDonnell et al. *Render me real? investigating the effect of render style on the perception of animated virtual humans*. ACM Transactions on Graphics 2012



Zibrek et al. *The Effect of Realistic Appearance of Virtual Characters in Immersive Environments - Does the Character's Personality Play a Role?* IEEE Trans Vis Comput Graph. 2018

5.4-Realism Personality

- Different styles influence appeal?
- Or personality more important?
- Perceptual metrics
  - Subjective ratings
  - Proximity
  - Attribution bias
- Affinity → complex interaction between appearance and personality
- Realism → positive choice

- Actor: react to different emotional situations

Zibrek et al. *Is Photorealism Important for Perception of Expressive Virtual Humans in Virtual Reality?* ACM Transactions on Applied Perception 2019

5.4-Realism Expression

- Realistic vs stylized
- 3 Attitudes: Friendly, Unfriendly or Sad
- Render style → no effect on confort or presence
- Photorealistic character → changed emotional response
- Preference for realism in VR → affinity and place illusion

Zibrek and McDonnell. *Social presence and place illusion are affected by photorealism in embodied VR.* MIG 2019

5.4-Realism
Social Presence

- Embodiment → photorealistic vs simplified
- Photorealism
  - → increases social presence and place illusion in VR
  - → no effect on behaviour (proximity)
- Possible link between empathy and render style



Hodgins et al. *The saliency of anomalies in animated human characters*. ACM Trans. Appl. Percept. 2010

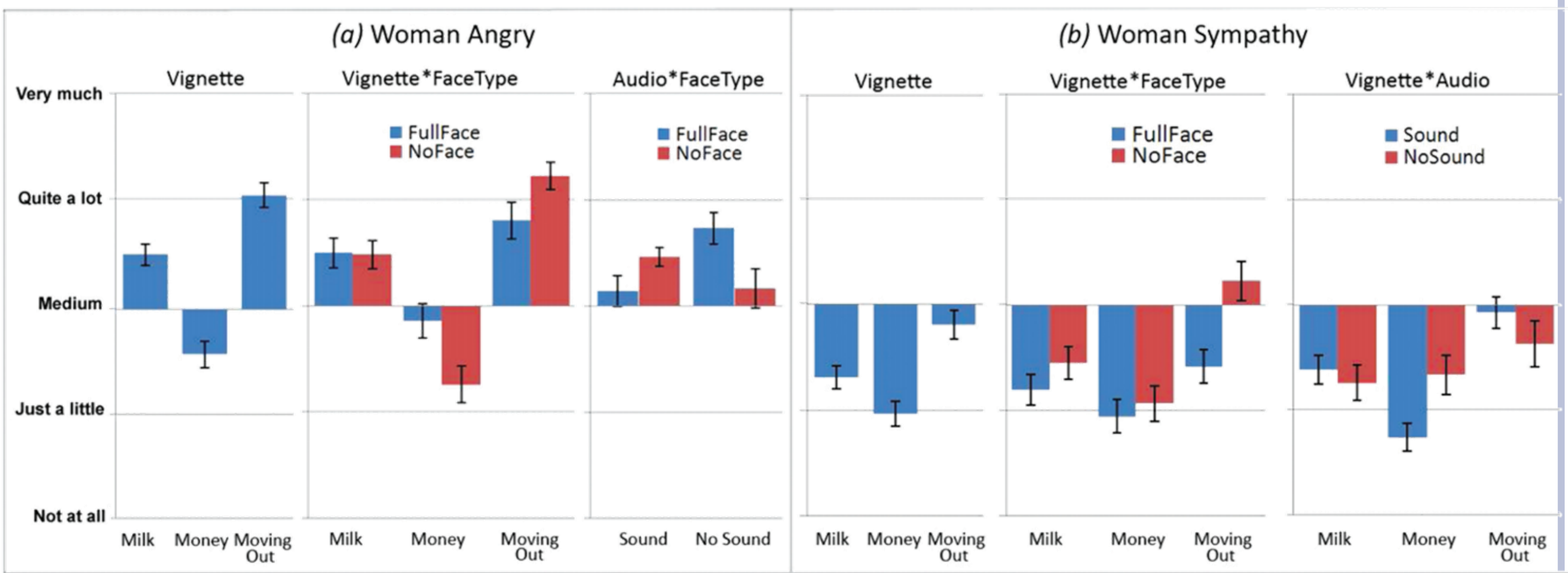
5.5-Artifacts Anomalies

- 3 animated vignettes of an arguing couple with detailed motion for face, eyes, hair and body
- Importance of different anomalies
  - Questionnaire → emotional response to full-length vignettes
    - With/without facial motion, with/without audio
  - 2AFC → compare performance in short clips
    - Range of different facial and body anomalies

Hodgins et al. *The saliency of anomalies in animated human characters*. ACM Trans. Appl. Percept. 2010

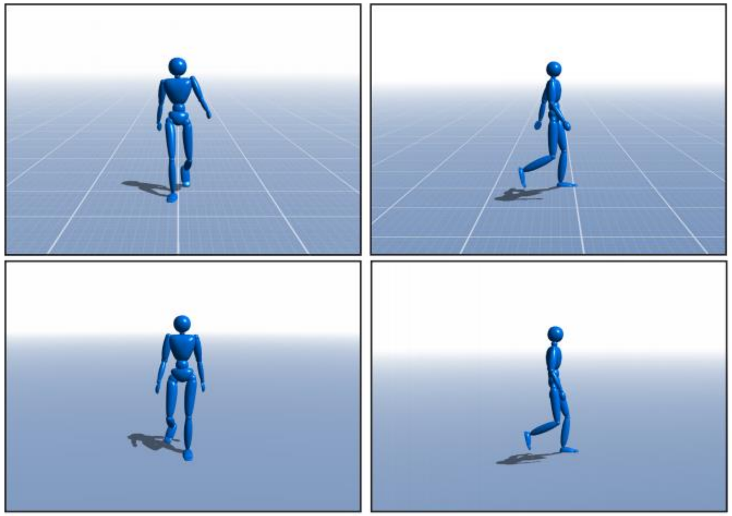
5.5-Artifacts Anomalies

Facial anomalies particularly salient
(even with body anomalies)



Pražák et al. *Perceptual evaluation of footskate cleanup*. SCA 2011

5.5-Artifacts Foot Sliding



- 2 Blocks
  - Grid texture on the ground →
    - Guides for footskate detection
  - Neutral environment
- Each block → worst-case scenarios
  - Side view
  - Front view
- Trials: 1 random motion (out of 10) with introduced amount of footsliding
  - Determine if the character is footsliding (yes/no)
  - Previous description
  - No explicit time limit

Pražák et al. *Perceptual evaluation of footskate cleanup*. SCA 2011

5.5-Artifacts Foot Sliding

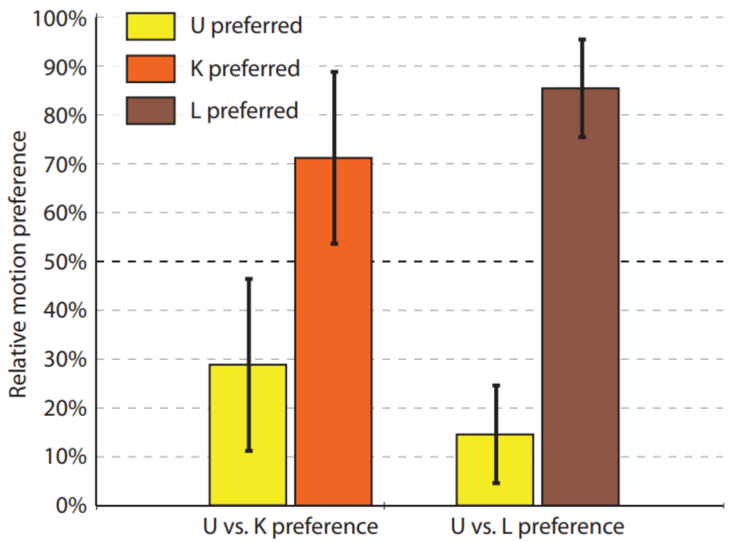


Figure 6: *The main effect of comparison factor on trials where corrected motions (K or L) were presented together with uncorrected ones (U).*

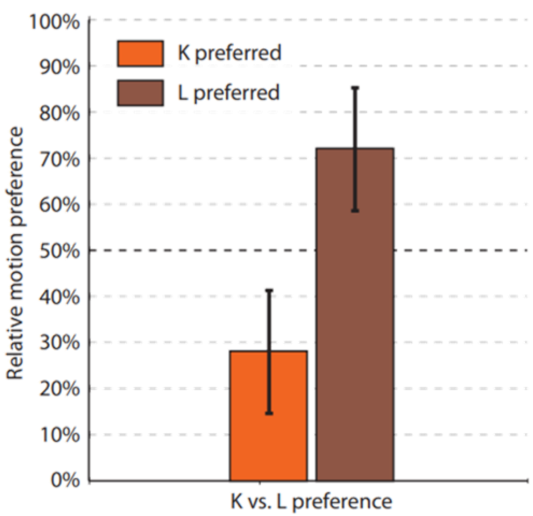


Figure 8: *Lengthening correction method is prefered over Kovar's when displayed simultaneously.*

McDonnell et al. *Clone attack! Perception of crowd variety* SIGGRAPH 2008

5.6-Variety Clones

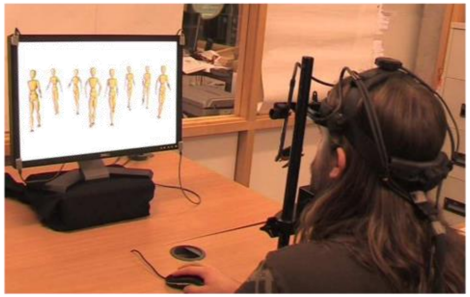
https://www.youtube.com/watch?v=rFExhAfkED0

- Appearance clones were easier to detect than motion clones
- Increasing clone multiplicity reduced variety significantly
- No appearance model was more easily detected than others
- Certain gaits were more distinctive than others
- Color modulation and spatial separation effectively masked appearance clones
- Combined appearance/motion clones were only harder to find than static appearance clones when their cloned motions were out-of-step
- Appearance clones were also harder to find when combined with random motions
- Motion clones were not affected at all by appearance, even with random appearances

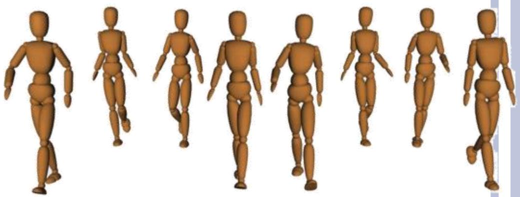
Exposure	# Appearance clones	# Motion Clones
5 seconds	8	10
10 seconds	4	10
15 seconds	2	9
20 seconds	none	7

McDonnell et al. *Eye-catching crowds: saliency based selective variation*. Transactions on Graphics. 2009

5.6-Variety Saliency Parts



- Which body parts are most looked at
  - Scenes containing clones
- Eye-tracking device
  - Fixations on body parts
  - Indicate if clones or not



- Head and upper torso attrack the majority of first fixations, and are atended the most
  - Regardles of orientation, presence or absence of motion, sex, age, size and clothing style
- Selective color variation → as effective as full color variation
- Head accessories, top texture and face texture variation → equally effective
- Facial geometry alterations → less effective

Pražák and O'Sullivan. *Perceiving human motion variety* APVG 2011

5.6-Variety
Motion Cloning

- Motion cloning: same motion animates more than one character
- No effect of either gender or position
- Infomal questionnaire → participants reported they were searching for pairs of similar motions → dominant factor = number of cloned motions
- Crowd size and speed → weak and upredictable effect
- Maximum number of individual motions successfully detectable → 2
- Minimum number of individual motions needed for a crowd to look varied → 3

5-Perception
More References

- Trinity College – Dublin
  - Carol O'Sullivan
  - Rachel McDonnell
    - https://www.youtube.com/watch?v=dC8Q78vAyR0

SGI-MEI
Sistemas Gráficos Interactivos
Animación de Personajes (2)

Motion synthesis, Crowds and Perception

Alejandro Beacco Porres

alejandro.beacco@upc.edu