Moving agents in geosimulation

Paul M. Torrens

Associate Professor

Geosimulation Research Laboratory

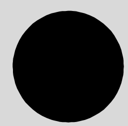
Department of Geography
University of Maryland
College Park, MD, USA

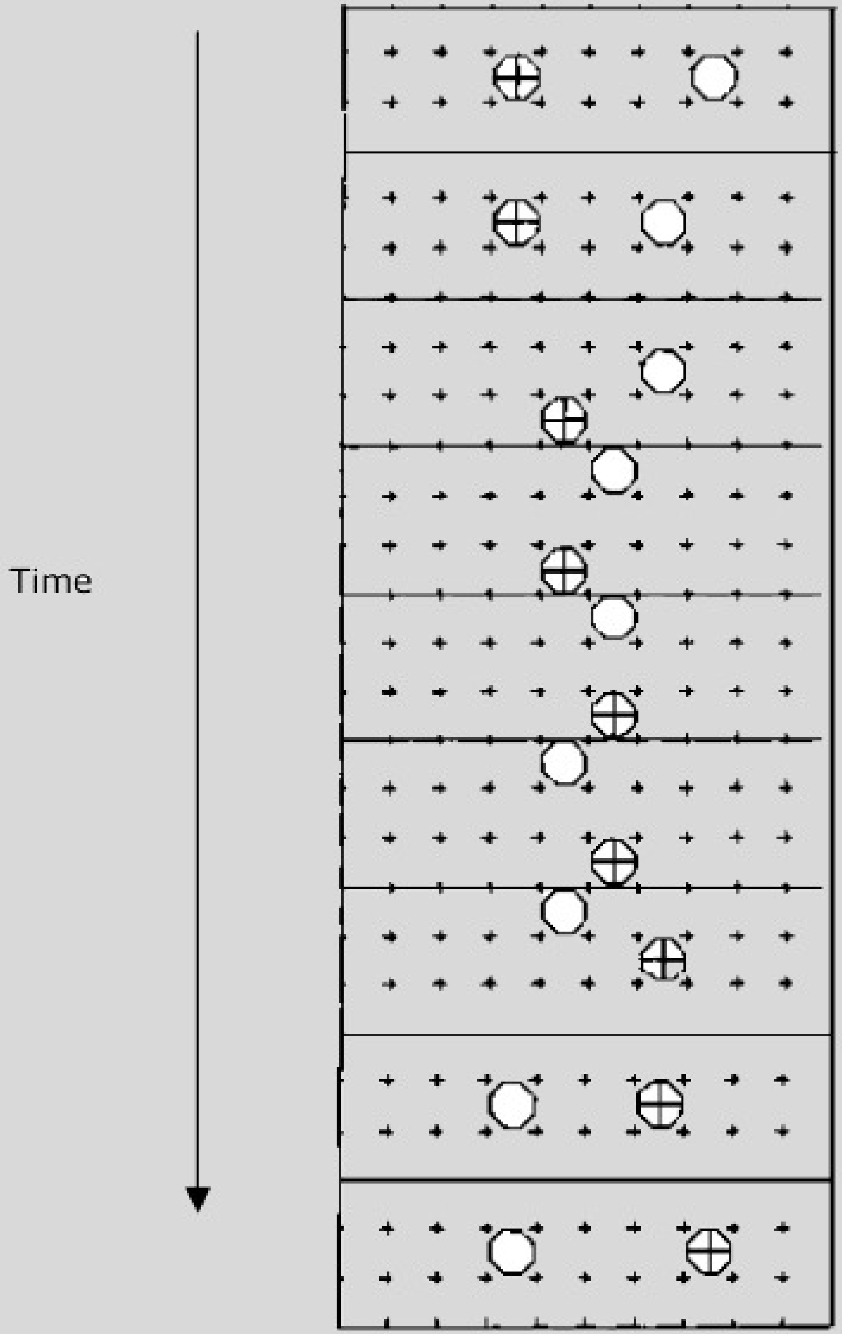
Animating behavioral geography through movement

- Times Square, 1920 (*New York Times*, public domain)

One of the often-advertised features of ABMs is their ability to dynamically map individual behaviors and characteristics to agent-actors in simulation, with incredible levels of detail

Coarse, abstract proxies of behavioral geography





Gipps & Marksjo (1985)

Spatial thinking and behavioral geography underpin much of the information processing we do when moving as pedestrians

How do we catalyze a next generation of geographic information science (and geosimulation)?

BIG

SCIENCE

bom-bom-
bommmm

What relationships exist between models and theory?

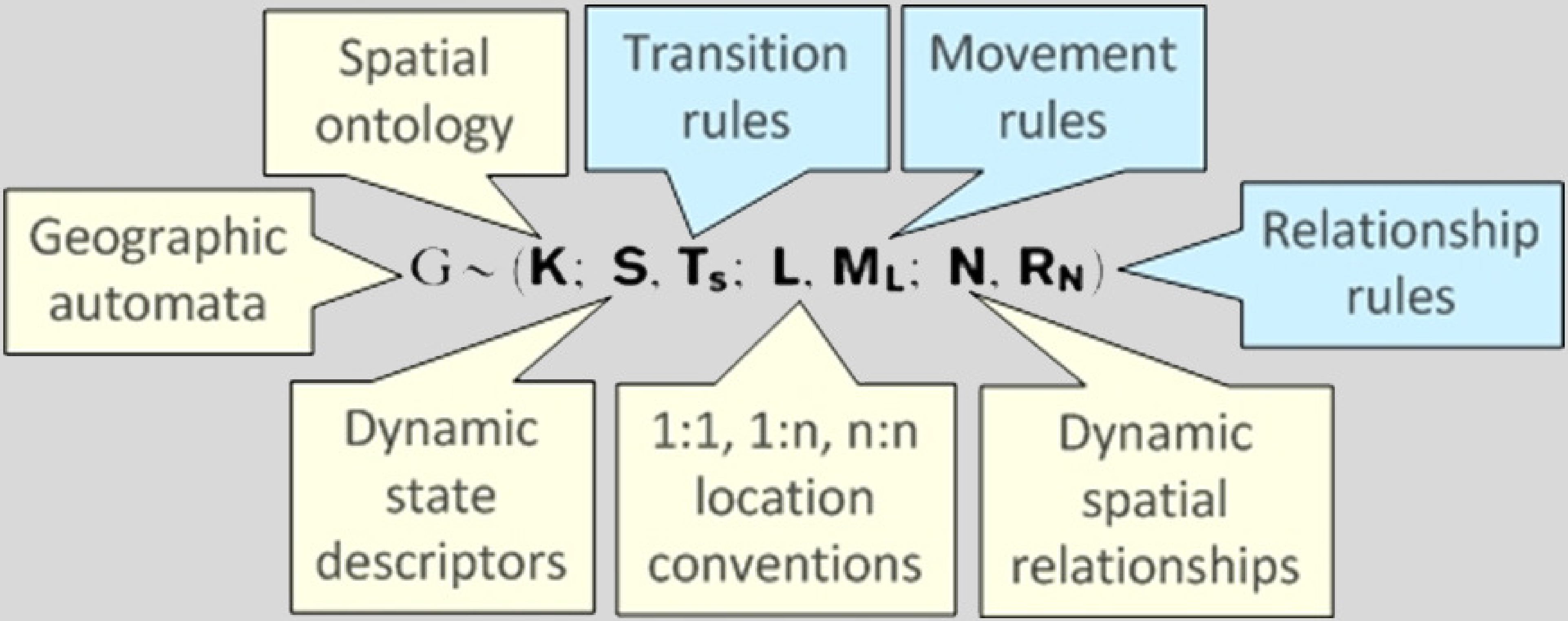
How do we translate human thinking into computable models?

How do phenomena emerge in dynamical systems?

Can we build more intricate representations of complex systems?

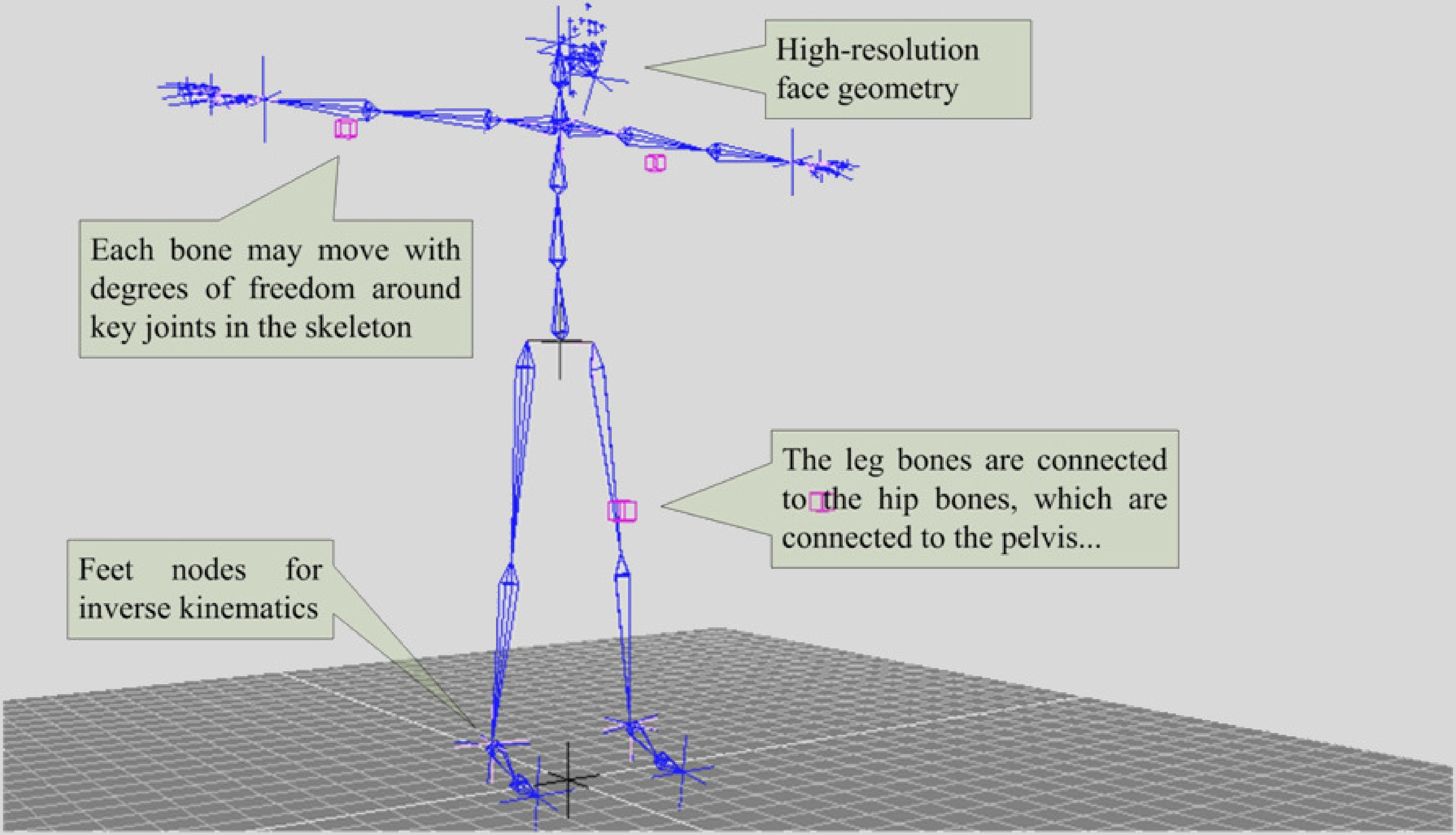
Can big data improve our understanding of the world?

My approach

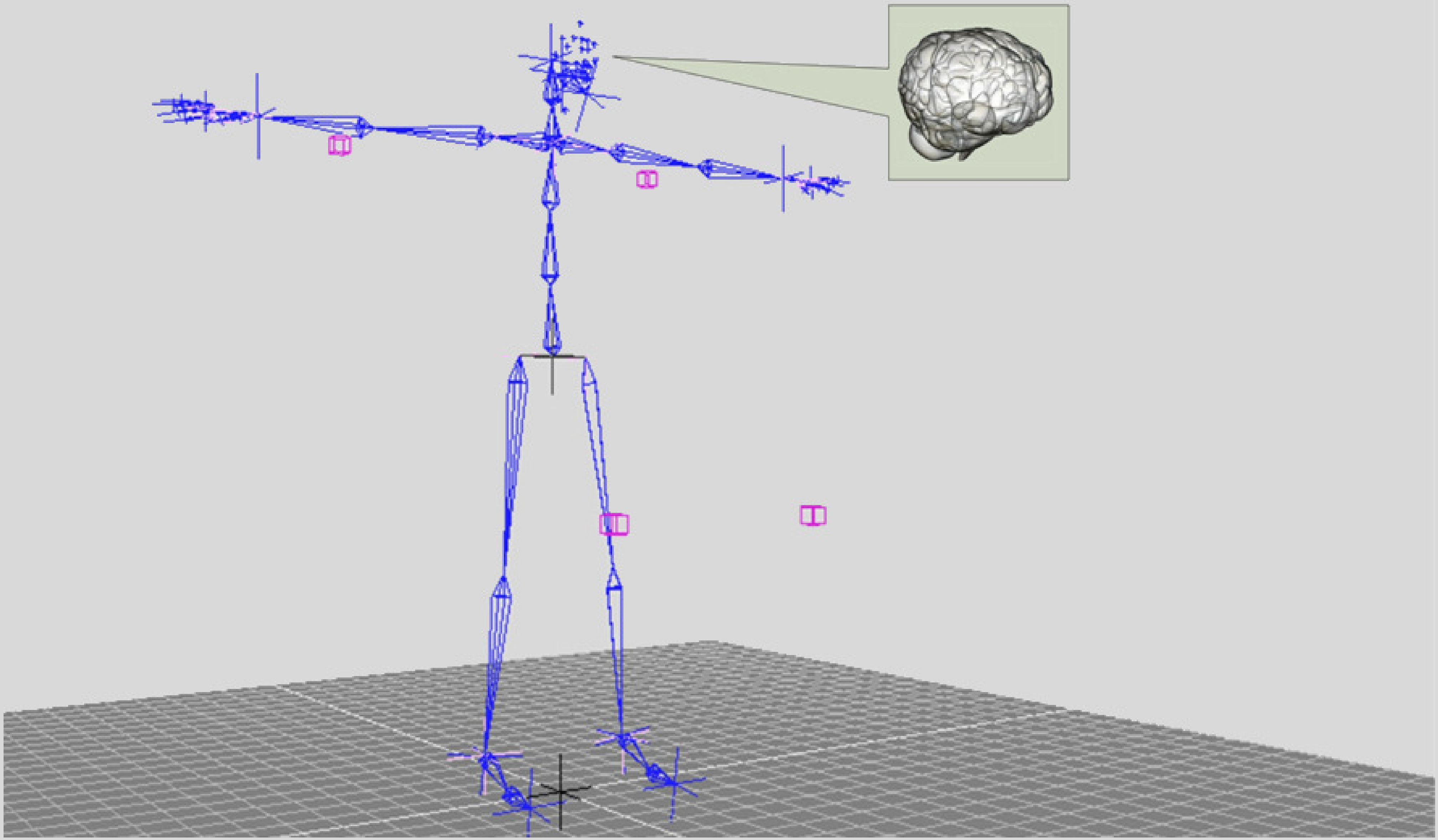


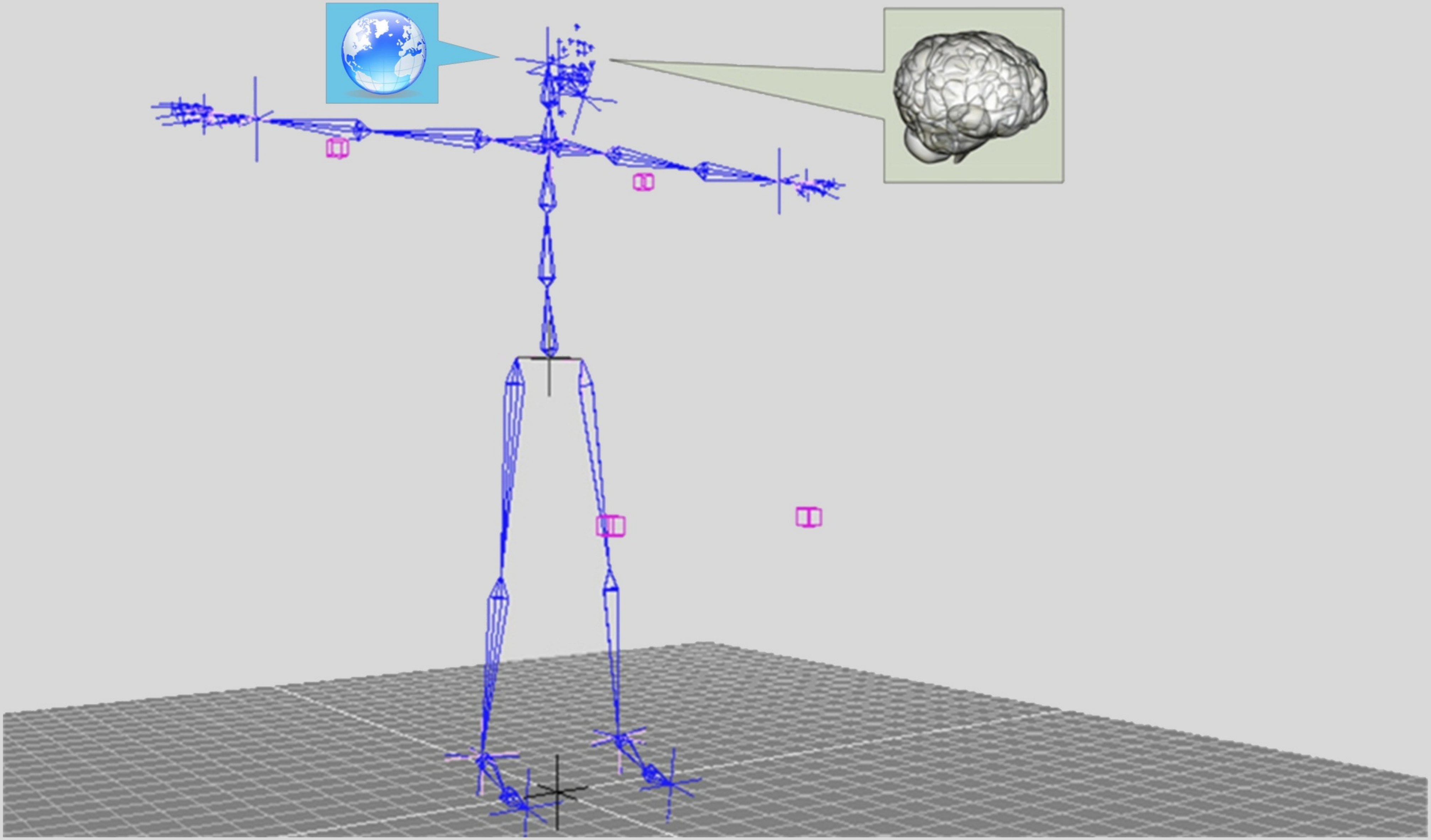
$$\mathbf{T_S}: (S_t, L_t, N_t) \rightarrow S_{t+1}$$
$$\mathbf{M_L}: (S_t, L_t, N_t) \rightarrow L_{t+1}$$
$$\mathbf{R_N}: (S_t, L_t, N_t) \rightarrow N_{t+1}$$

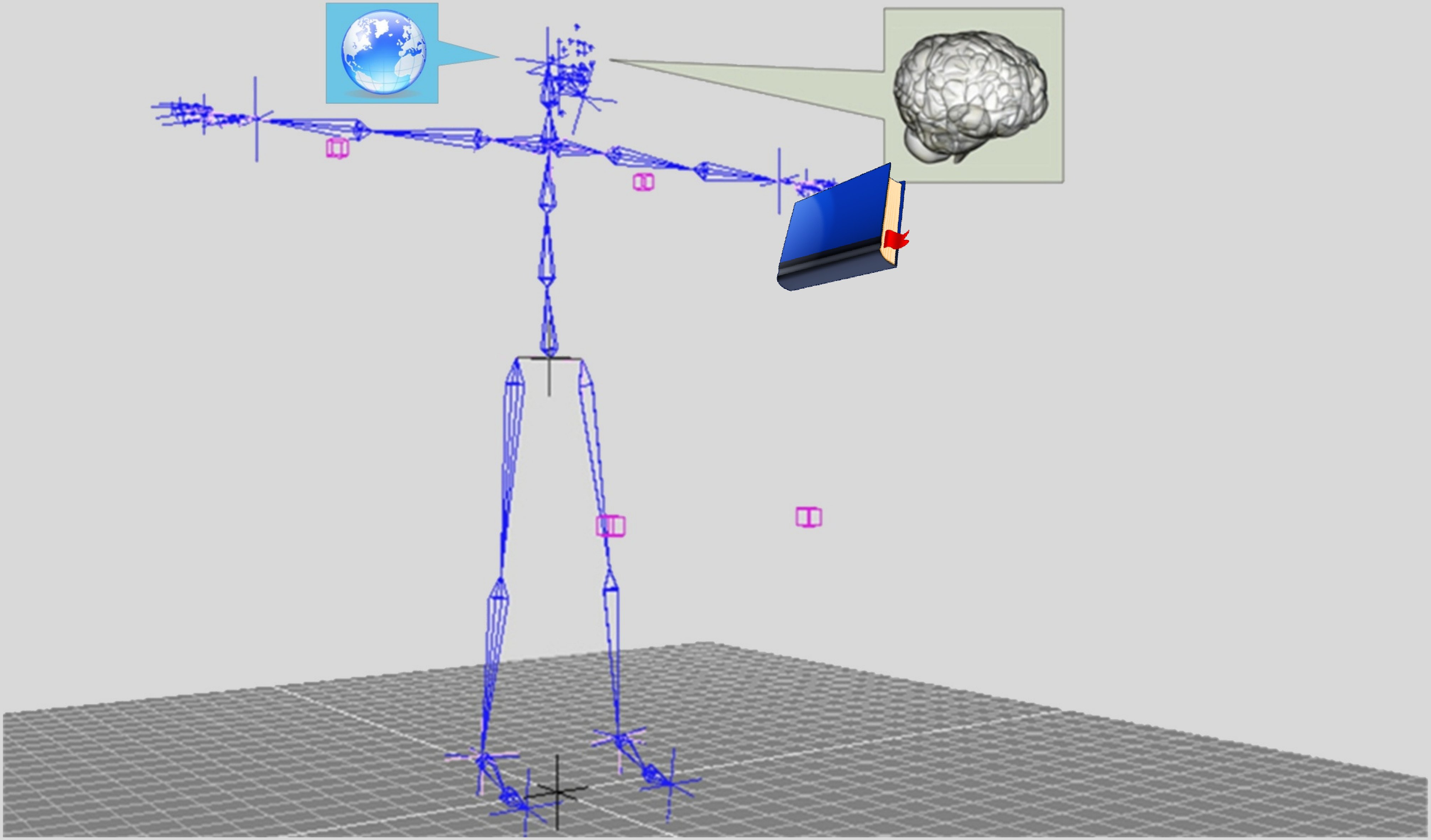
Representing people synthetically

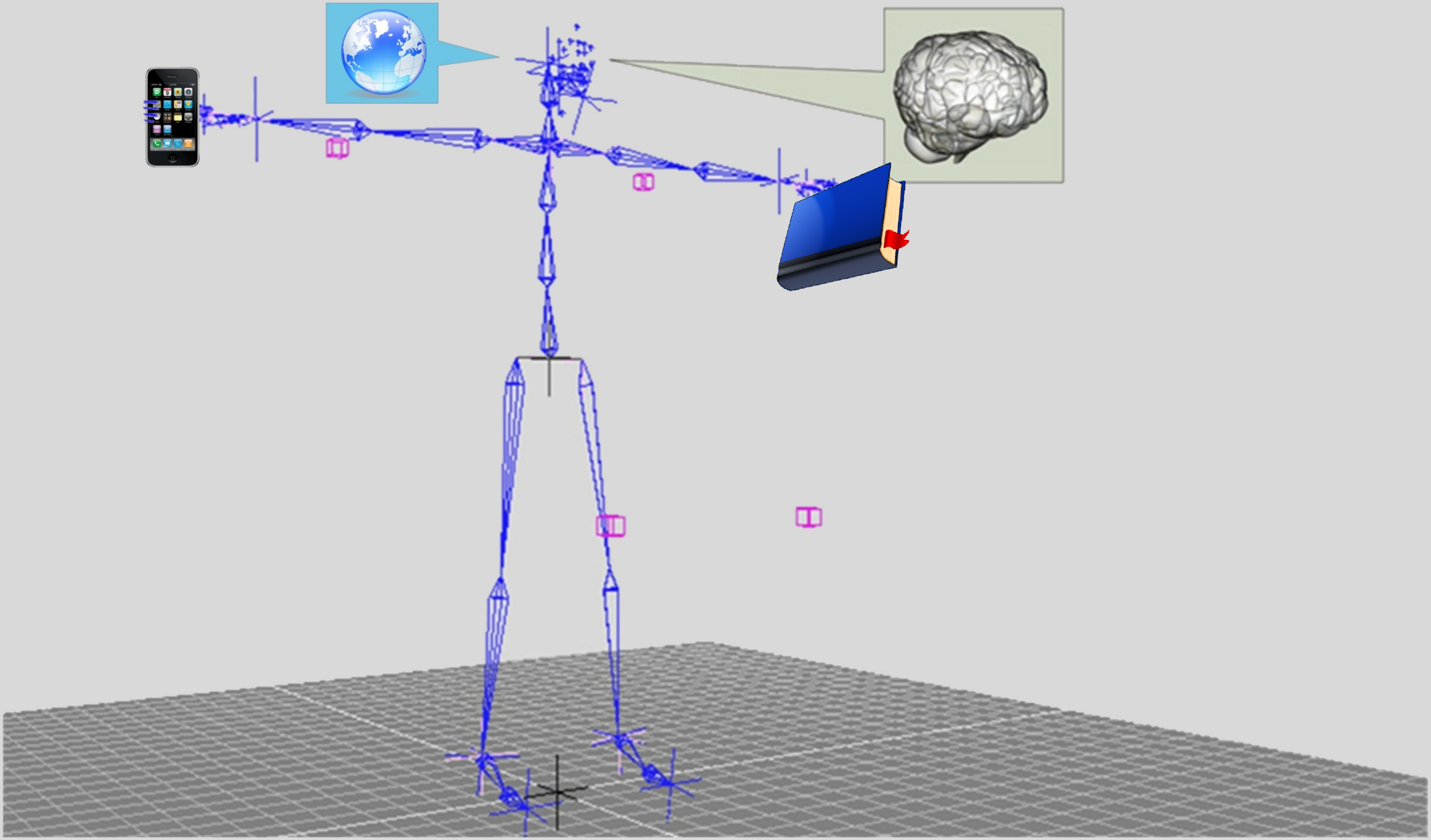

High-resolution face geometry
Each bone may move with degrees of freedom around key joints in the skeleton
The leg bones are connected to the hip bones, which are connected to the pelvis...
Feet nodes for inverse kinematics

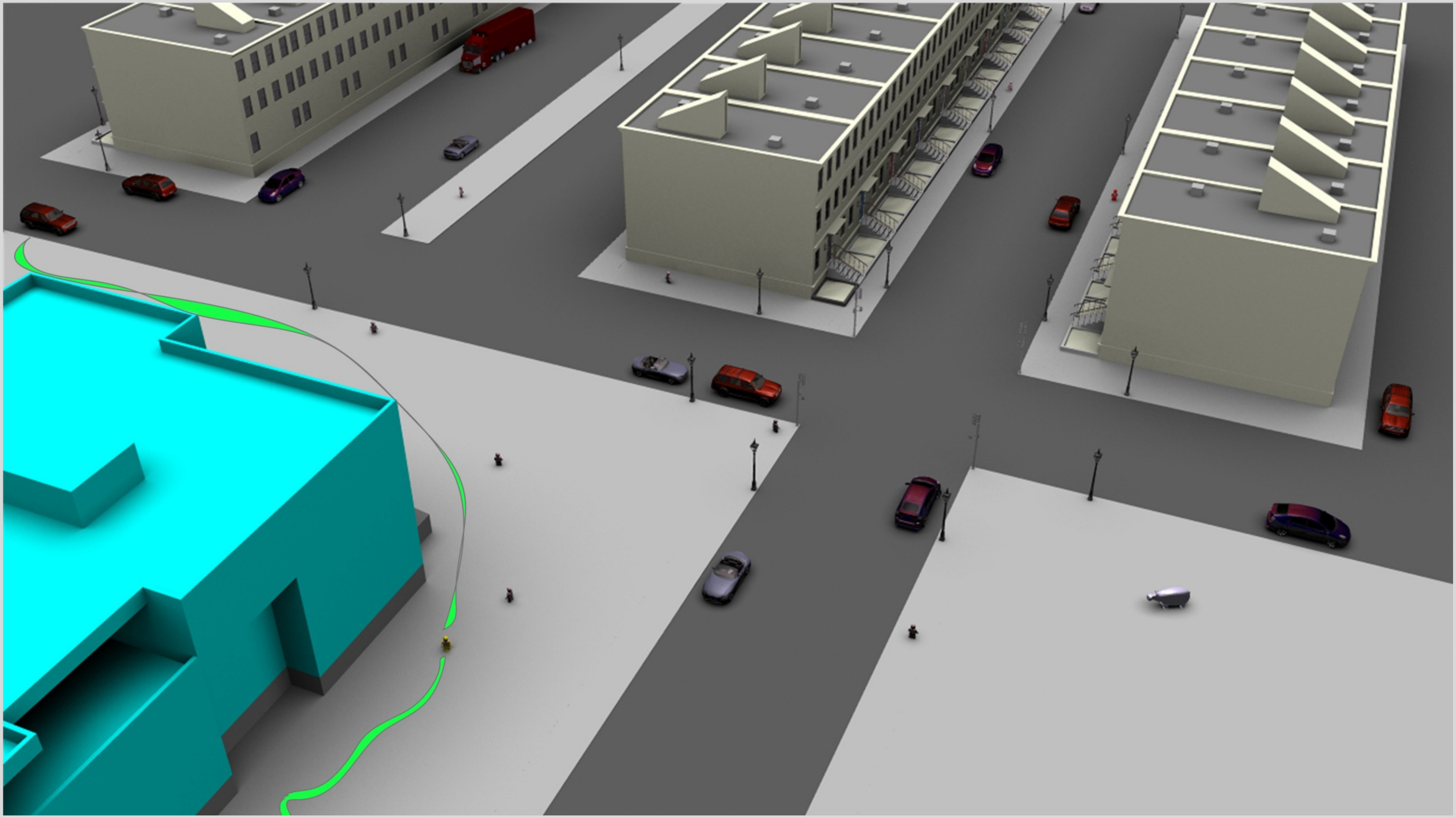
Representing people's behavior



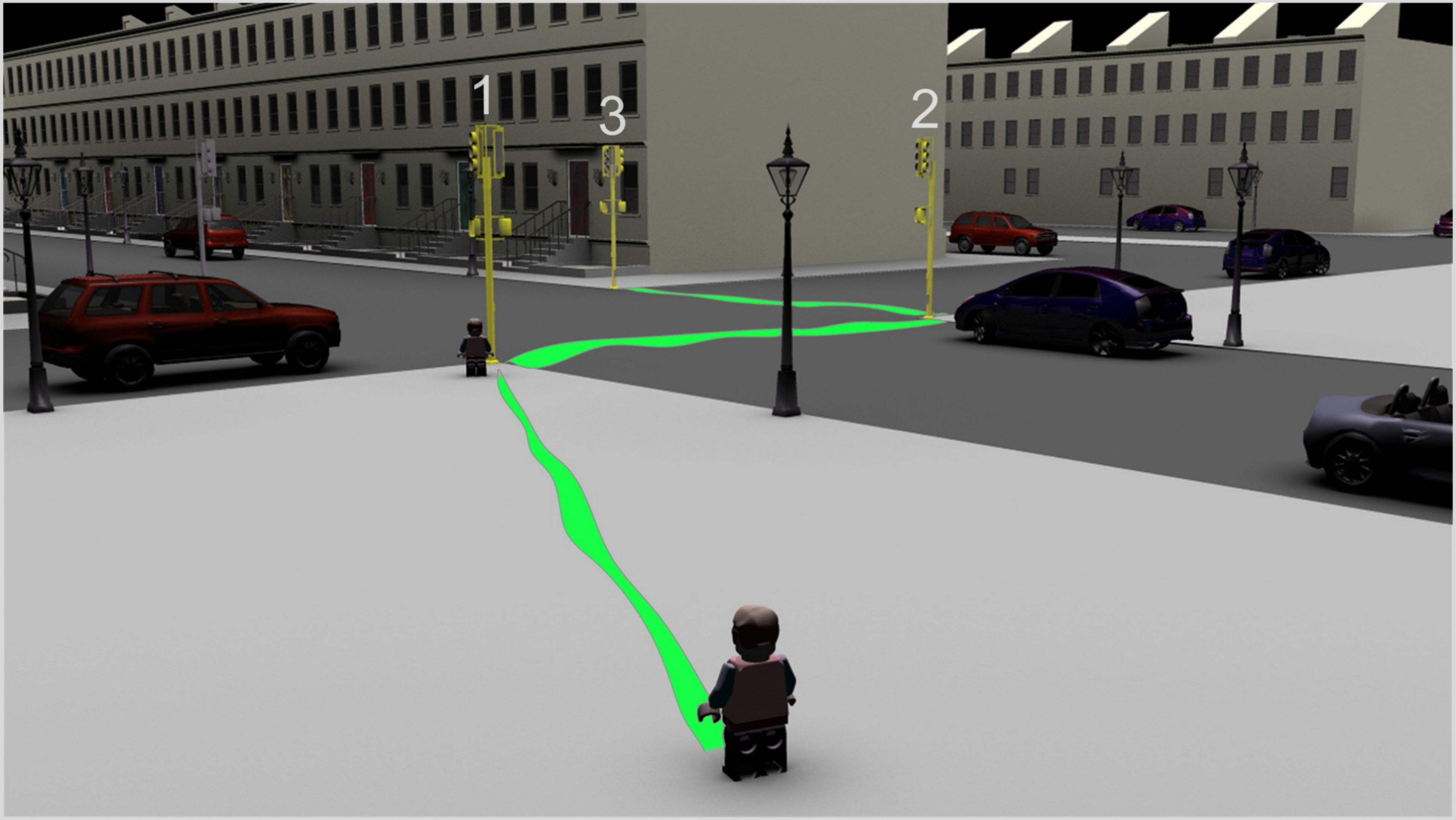




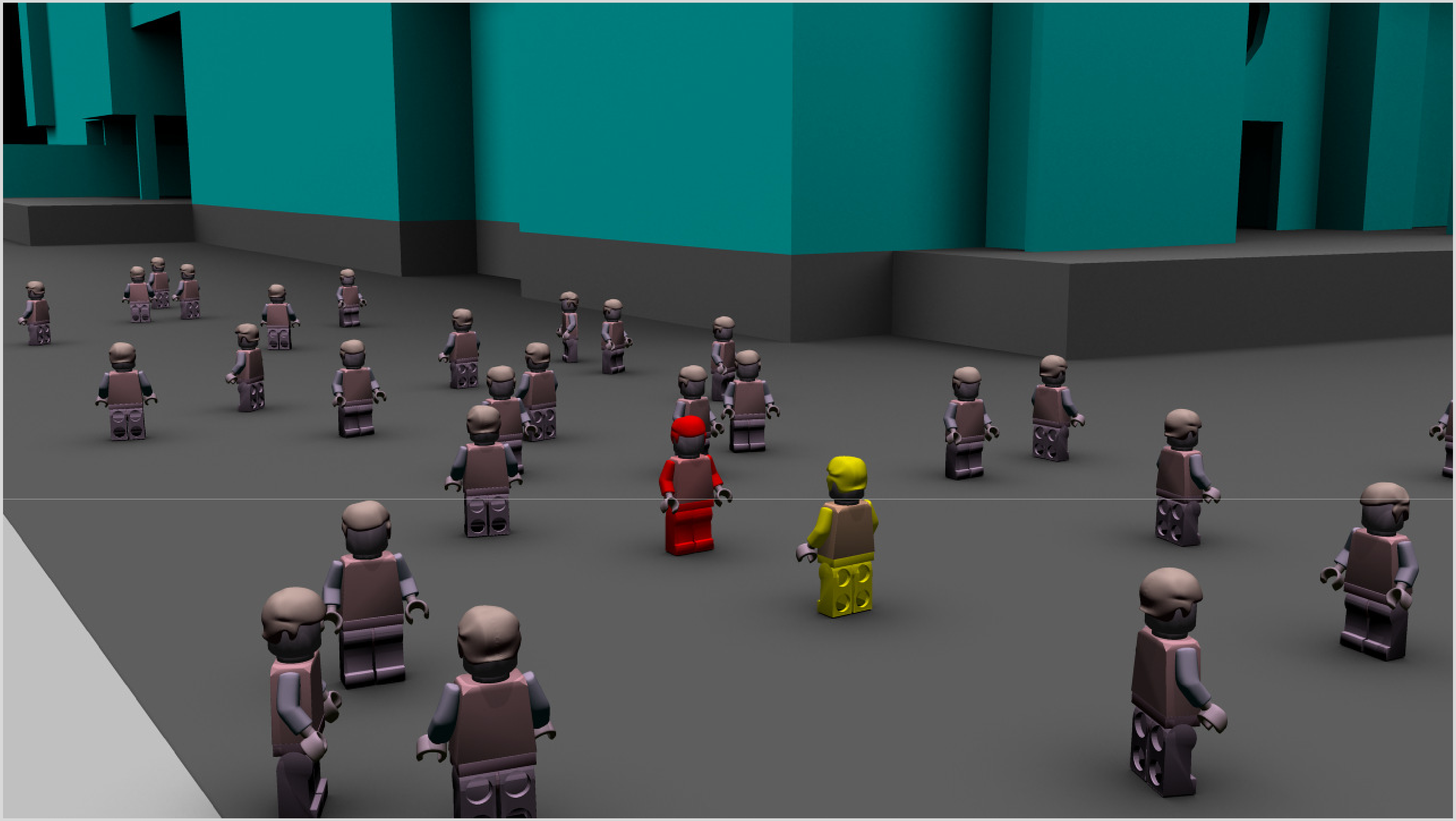




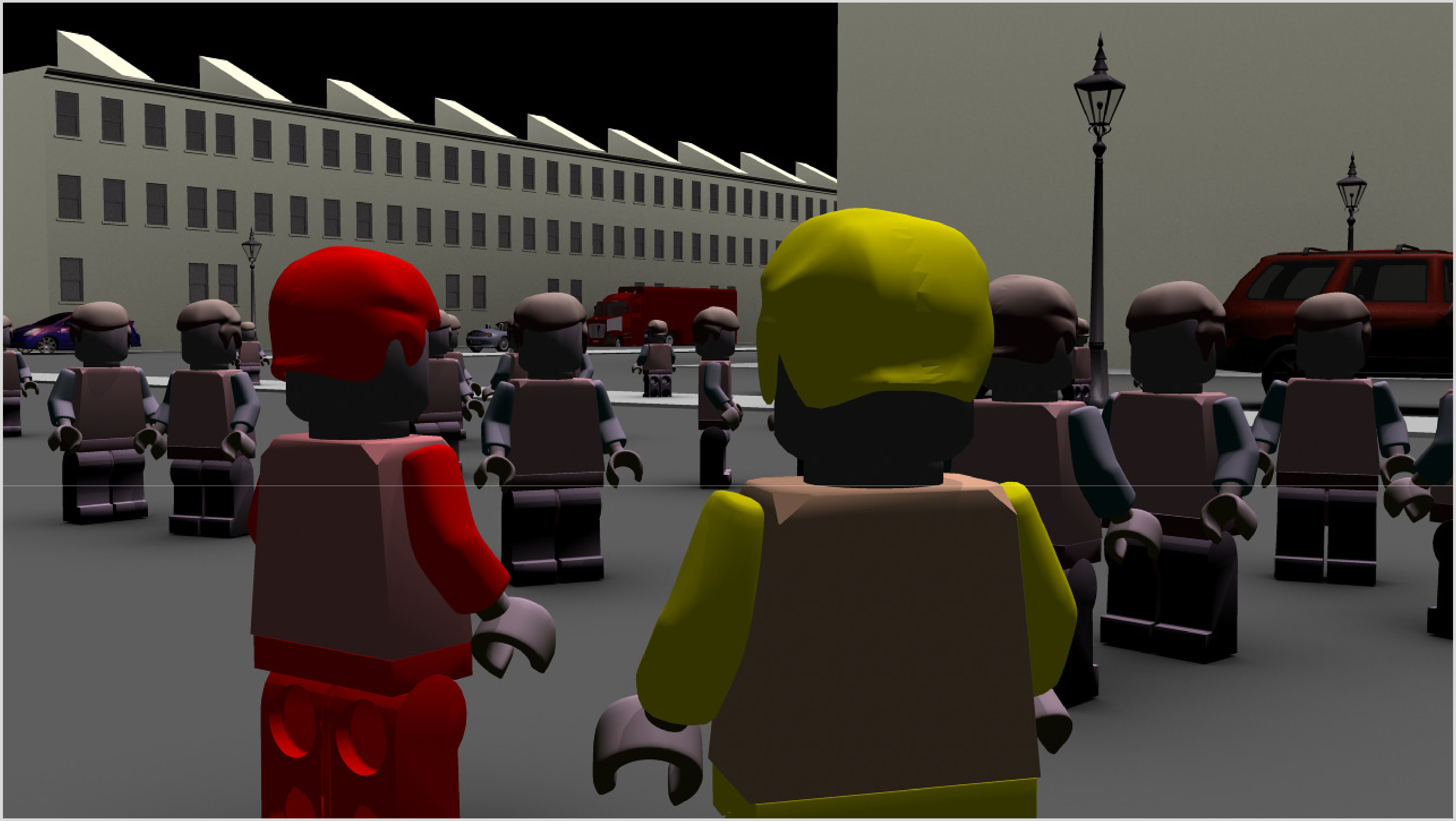
City-scale



Neighborhood-scale



Street-scale

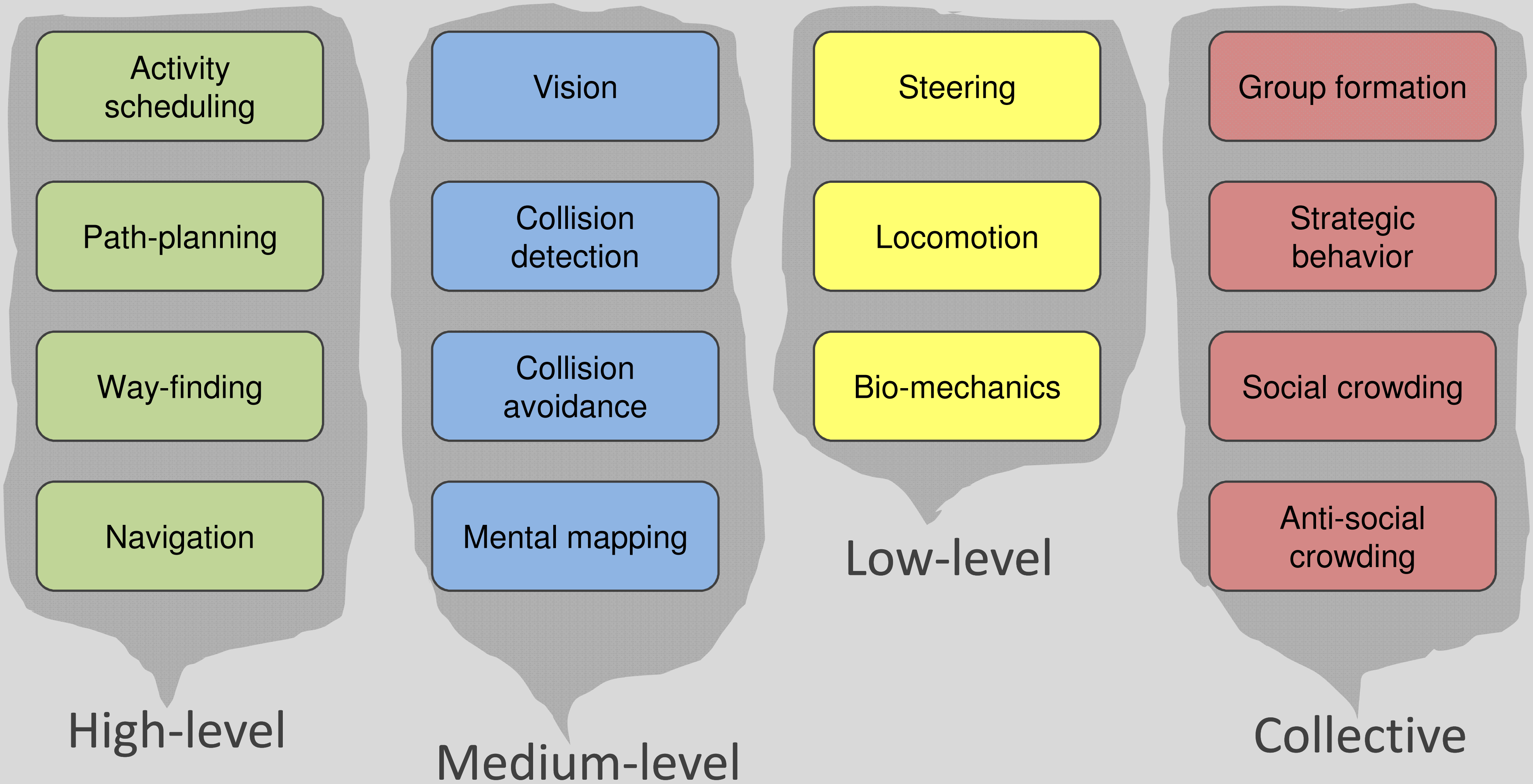


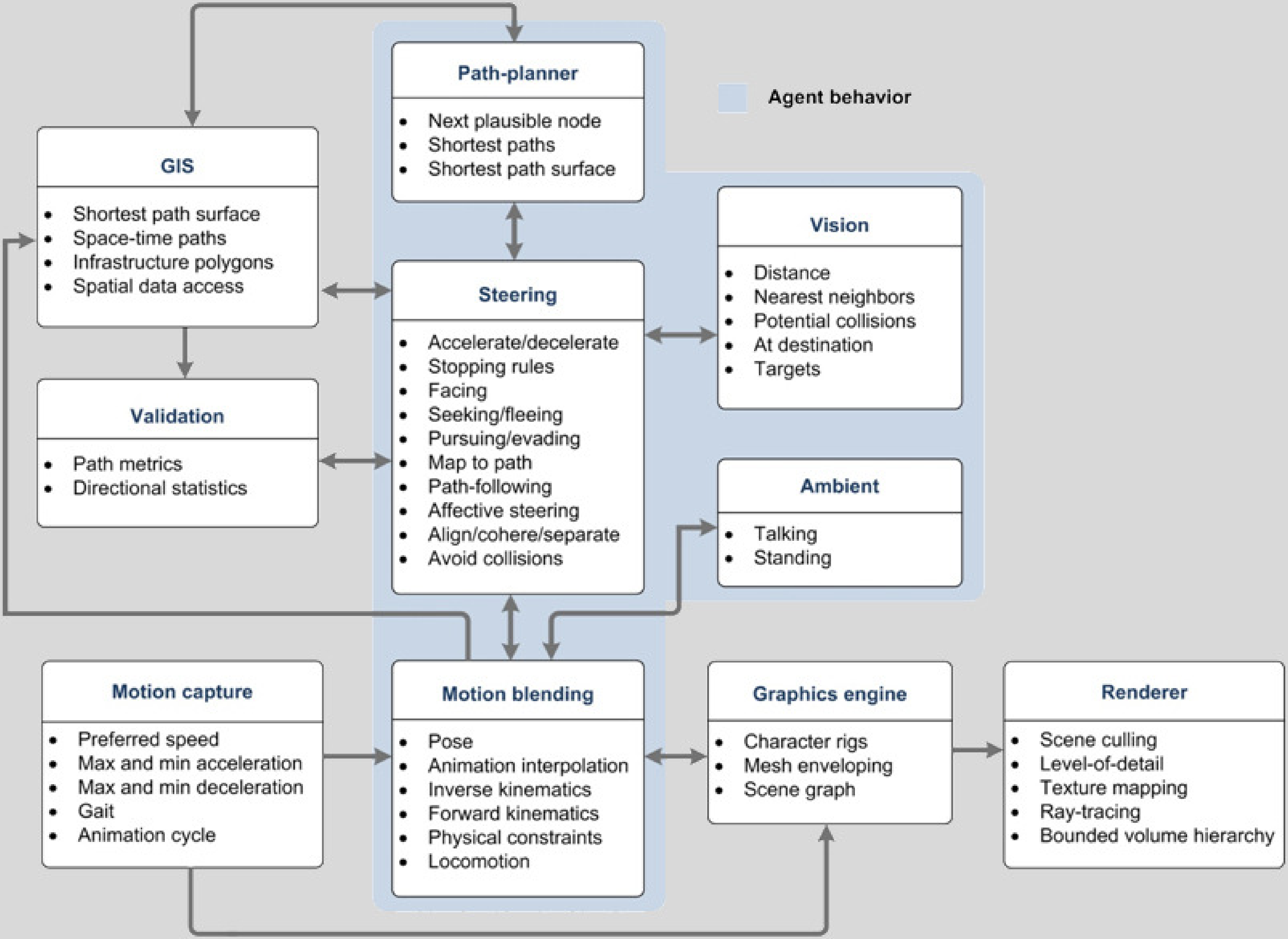
People-scale

Body-scale

Body-scale

Spatial thinking and behavior in computable form




Path-planner
Next plausible node
Shortest paths
Shortest path surface
Agent behavior
GIS
Shortest path surface
Space-time paths
Infrastructure polygons
Spatial data access
Vision
Distance
Nearest neighbors
Potential collisions
At destination
Targets
Steering
Accelerate/decelerate
Stopping rules
Facing
Seeking/fleeing
Pursuing/evading
Map to path
Path-following
Affective steering
Align/cohere/separate
Avoid collisions
Validation
Path metrics
Directional statistics
Ambient
Talking
Standing
Motion capture
Preferred speed
Max and min acceleration
Max and min deceleration
Gait
Animation cycle
Motion blending
Pose
Animation interpolation
Inverse kinematics
Forward kinematics
Physical constraints
Locomotion
Graphics engine
Character rigs
Mesh enveloping
Scene graph
Renderer
Scene culling
Level-of-detail
Texture mapping
Ray-tracing
Bounded volume hierarchy

GIS

- GDAL
- ShapeLib
- ESRI

Rendering

- OpenGL
- XNA (Xbox)
- 3DS
- Mental Ray

Math

- C++
- Bullet Physics

I/O

- PostGIS
- GeosimXML
- GeosimGUI

Path-planning

- A*
- Dijkstra
- GeosimSpaceTime
- GeosimIsovist

NavGraphs

- Boost
- GeosimGraph

Collision detection

- Bullet Physics
- Endorphin
- GeosimSpaceTime
- GeosimRays

Steering

- Opensteer
- Social force model
- GeoSteer
- GeosimMachineLearning

Character rigs

- CAT
- Endorphin
- OpenGL
- GeosimCharacter

Locomotion

- Opensteer
- GeosimLoco

Motion controller

- CAT
- XNA
- Endorphin
- GeosimMotion

GIS

- GDAL
- ShapeLib
- ESRI

Rendering

- OpenGL
- XNA (Xbox)
- 3DS
- Mental Ray

Math

- C++
- Bullet Physics

I/O

- PostGIS
- GeosimXML
- GeosimGUI

Path-planning

- A*
- Dijkstra
- GeosimSpaceTime
- GeosimIsovist

NavGraphs

- Boost
- GeosimGraph

Collision detection

- Bullet Physics
- Endorphin
- GeosimSpaceTime
- GeosimRays

Steering

- Opensteer
- Social force model
- GeoSteer
- GeosimMachineLearning

Character rigs

- CAT
- Endorphin
- OpenGL
- GeosimCharacter

Locomotion

- Opensteer
- GeosimLoco

Motion controller

- CAT
- XNA
- Endorphin
- GeosimMotion

Some (partial) explanation of the nitty-gritty details to demonstrate the intricacy involved in building this infrastructure

High-level behavior

Path-planning

The general form of the search algorithm.

	$\boldsymbol{SEARCH}(G, w, s)$

Initialize a search over directed or undirected graph, with vertices V and edges E, $G = (V, E)$, positive edge lengths $(l_e : e \in E)$, and edge weights w. Begin the search at origin $s \in V$. Record the distance d from the origin s. For each node u that is expanded, record its predecessor π (from an adjacency list $Adj[]$). Maintain a priority list Q as a queue of nodes that remain to be searched at any stage in the algorithm. Also maintain a list S_{list} of the nodes that form the shortest path from origin s to the destination. Initially, mark all nodes as unvisited (WHITE)	for each vertex $u \in V[G]$
	    do $color[u] \leftarrow WHITE$
	    $d[u] \leftarrow \infty$
	    $\pi[u] \leftarrow NIL$
	$Q \leftarrow \emptyset$
	$S_{list} \leftarrow \emptyset$

Begin the search at the point of origin s and mark the node as visited (GRAY)	$color[s] \leftarrow GRAY$
Record its distance from the origin $s = 0$	$d[s] \leftarrow 0$
Record its predecessor node as NIL	$\pi[s] \leftarrow NIL$
Add s to the priority queue by popping stack Q	$ENQUEUE(Q, s)$
As long as more vertices remain to be explored…	while $Q \neq \emptyset$
Remove the previously-visited node from the priority queue by pushing stack Q	do $u \leftarrow DEQUEUE(Q)$

Estimated plausibility of node

Given node

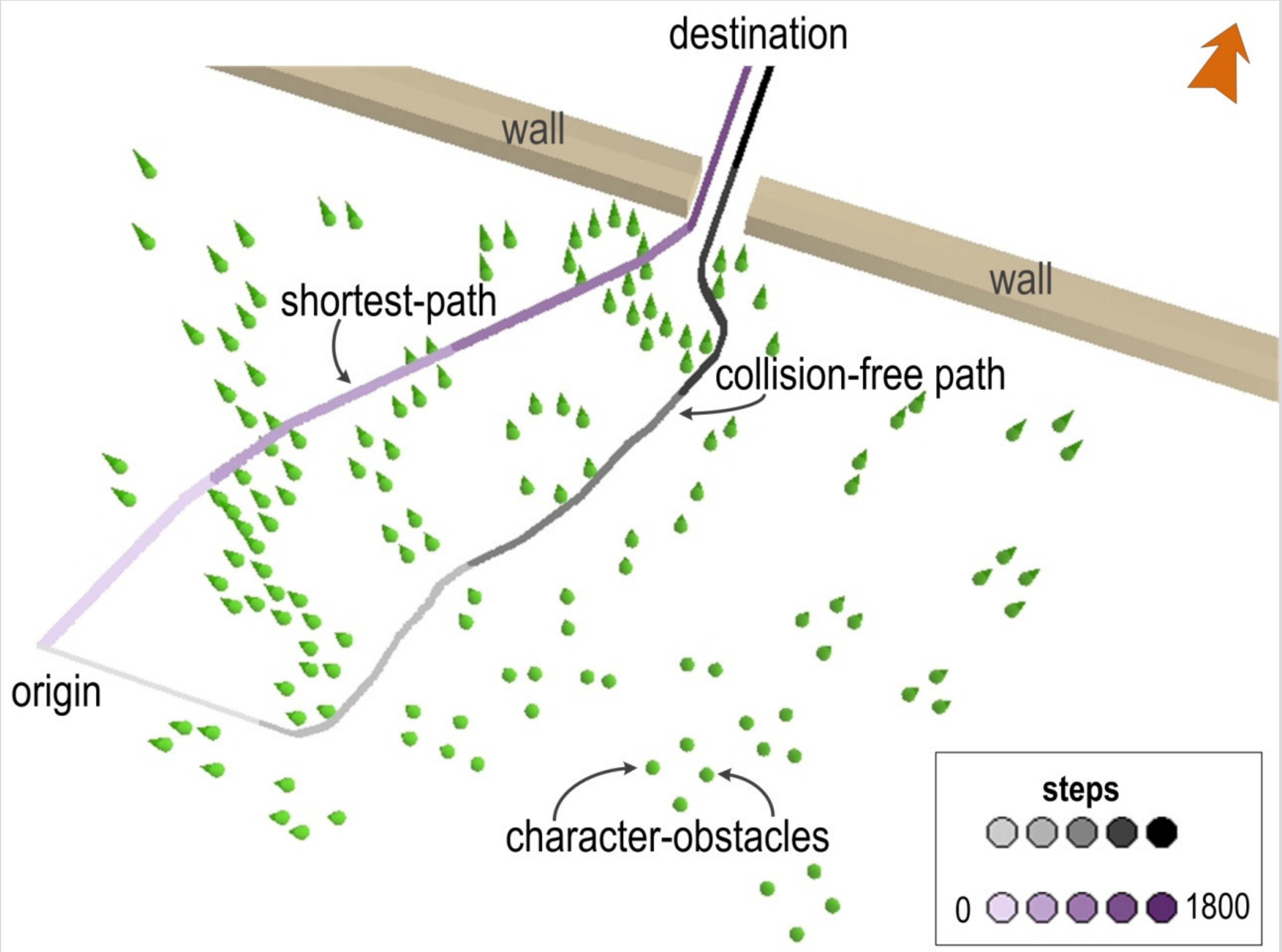
$$f^*(n) = g^*(n) + h^*(n)$$

Estimated progress from the origin *s*

Estimated Euclidean distance to the destination

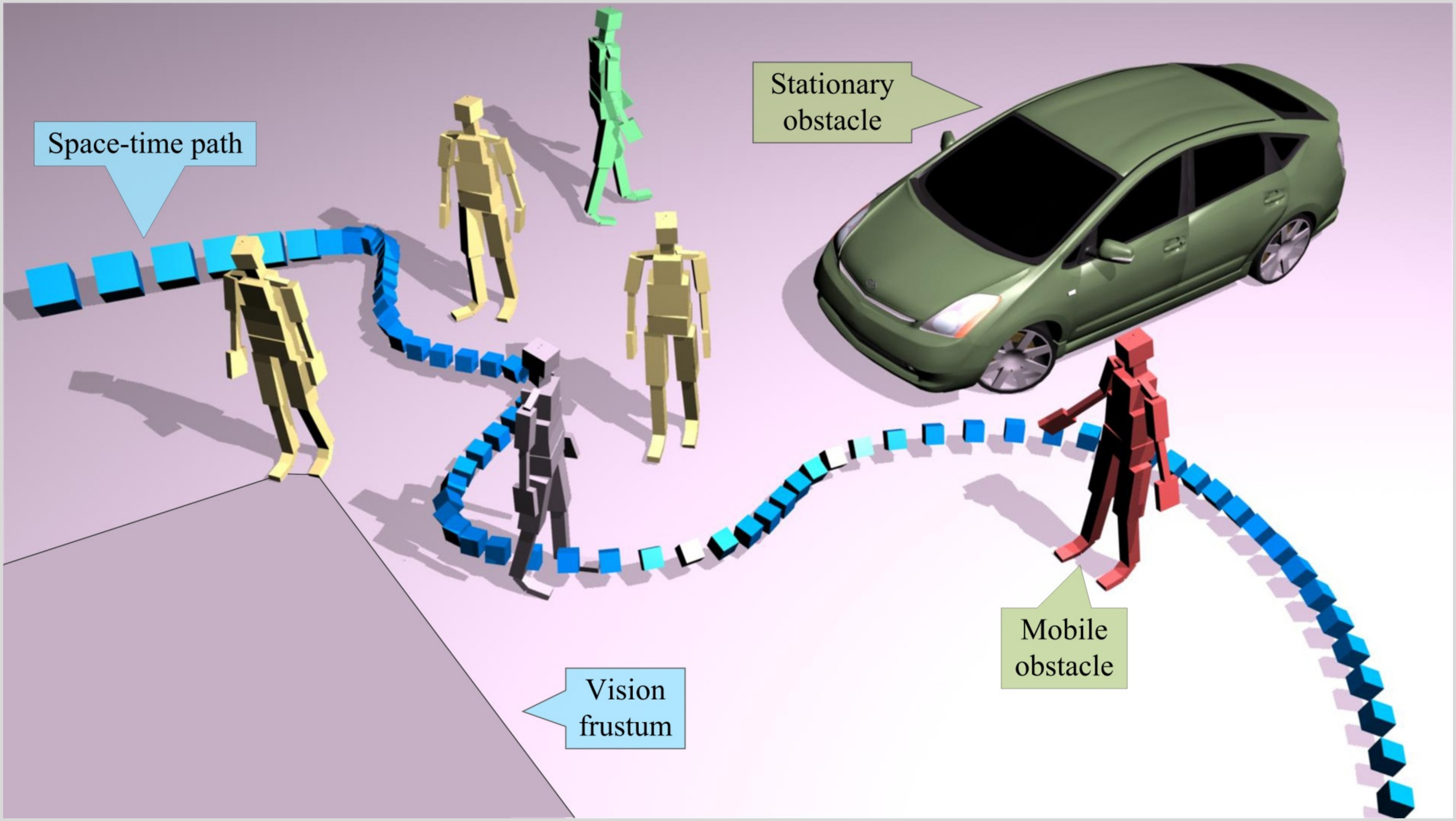
Uh-oh

[Uh-oh]


destination
wall
wall
shortest-path
collision-free path
origin
character-obstacles
steps
0
1800

Medium-level behavior

Collecting ambient geographic information with vision



Proactive vision—steering—locomotion triple

- Speed
- Acceleration
- Mobility
- Radius
- Orientation
- Rotation
- Neighbor
- Distance
- Proximity
- Estimated future vector
- Collision potential
- Potential collision timing

Steering

Basic movements

Facing in the right direction

Direction of target location

x-dimension of vector from agent to target

y-dimension of vector from agent to target

$$facing\,(O_q) = atan2(\overrightarrow{pq}_y, \overrightarrow{pq}_x)$$

$$\text{where, } atan2(y,x)\begin{cases} \text{undefined when } y = 0, x = 0 \\ \pi + arctan\left(\frac{y}{x}\right) \text{ when } y \geq 0, x < 0 \\ -\pi + arctan\left(\frac{y}{x}\right) \text{ when } y < 0, x < 0 \\ -\frac{\pi}{2} \text{ when } y < 0, x = 0 \\ \frac{\pi}{2} \text{ when } y > 0, x = 0 \\ arctan\left(\frac{y}{x}\right) \text{ when } x > 0 \end{cases}$$

Spatio-temporal reaction/interaction

Separate

Seek

Flee

Wander

Arrive

Leave

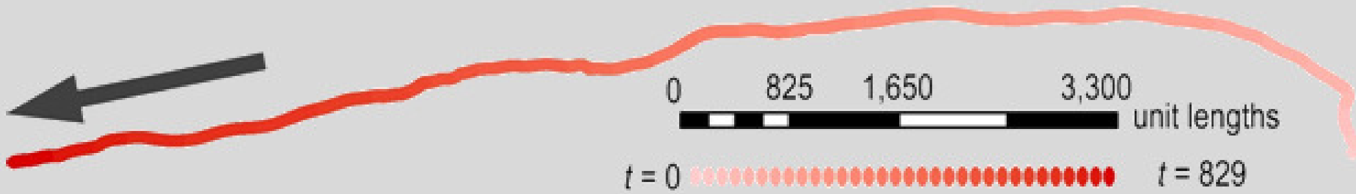
Align

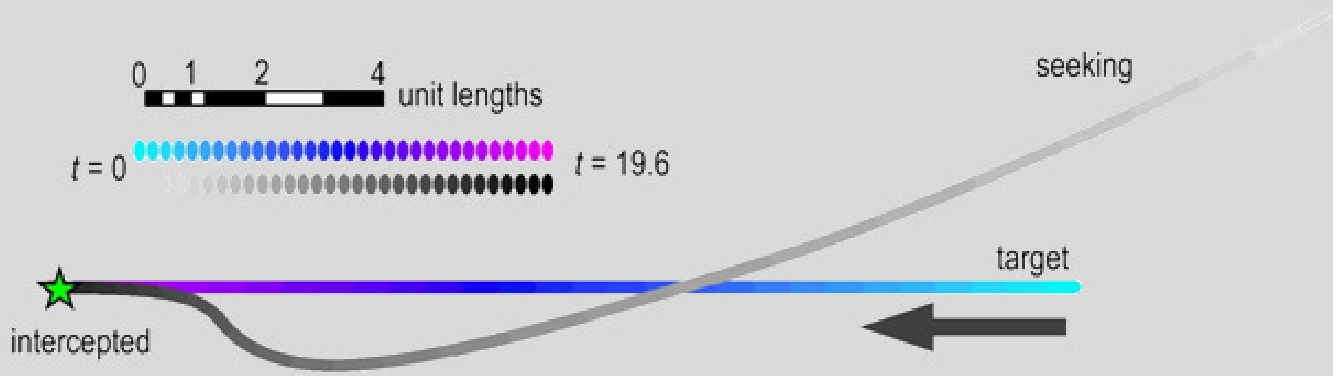
Pursue

Evade

Path-follow

Match velocity





Fleeing and seeking a mobile target

Vector from agent to target

Maximum (preferred) acceleration

$$fleeing = \frac{\overrightarrow{qp}}{|\overrightarrow{qp}|} * a_{max}$$

$$seeking = \frac{\overrightarrow{pq}}{|\overrightarrow{pq}|} * a_{max}$$

Length of vector from target to agent

Arriving (gently) at a target

Maximum (preferred) speed

Velocity of the agent

$$arriving = \begin{cases} \left(s_{max} * \dfrac{|\overrightarrow{qp}|}{r_{slowdown}} \right) * \dfrac{\overrightarrow{qp}}{|\overrightarrow{qp}|} - v_p, & \text{if } |\overrightarrow{qp}| \leq r_{arrival} \\ s_{max} * \dfrac{\overrightarrow{qp}}{|\overrightarrow{qp}|} - v_p, & \text{if } |\overrightarrow{qp}| > r_{arrival} \end{cases}$$

Length of vector from target to agent

Deceleration threshold

Preemptive/proactive spatio-temporal action, interaction, reaction

Pursuing and evading

Estimated future position of target

Position of target

Velocity of target

$$\left.\begin{array}{r} pursuing = seeking(q_{\hat{t}}) \\ evading = fleeing(q_{\hat{t}}) \end{array}\right\}, \text{ where } q_{\hat{t}} = q + v_q * \hat{t},$$

Length of vector from target to agent

$$\text{and where } \hat{t} = min\left(t_{max}, \frac{|\overrightarrow{qp}|}{v_{max}}\right)$$

Estimation cut-off time

Maximum (preferred) velocity

Short estimation time for vector-guessing

Aligning with a target

Rotation of target

Rotation of agent

Maximum angular acceleration

$$aligning = \begin{cases} \left(\dfrac{\frac{\Omega_q - \Omega_p}{t_q}}{\propto} \right) * \propto_{max}, & \text{if } \propto > \propto_{max} \\ \dfrac{\Omega_q - \Omega_p}{t_q}, & \text{otherwise} \end{cases}$$

Time to target

where $\propto = abs(\Omega_q - \Omega_p)$,

Maximum rotation

Slowdown filter

$$\text{where} \begin{cases} \Omega_q = \Omega_{max}, & \text{if } |\Omega_q| > r_{slowdown} \\ \Omega_q = \Omega_{max} * \left(\dfrac{|\Omega_q|}{r_{slowdown}} * \dfrac{\Omega_q}{|\Omega_q|} \right), & \text{if } |\Omega_q| < r_{slowdown} \end{cases}$$

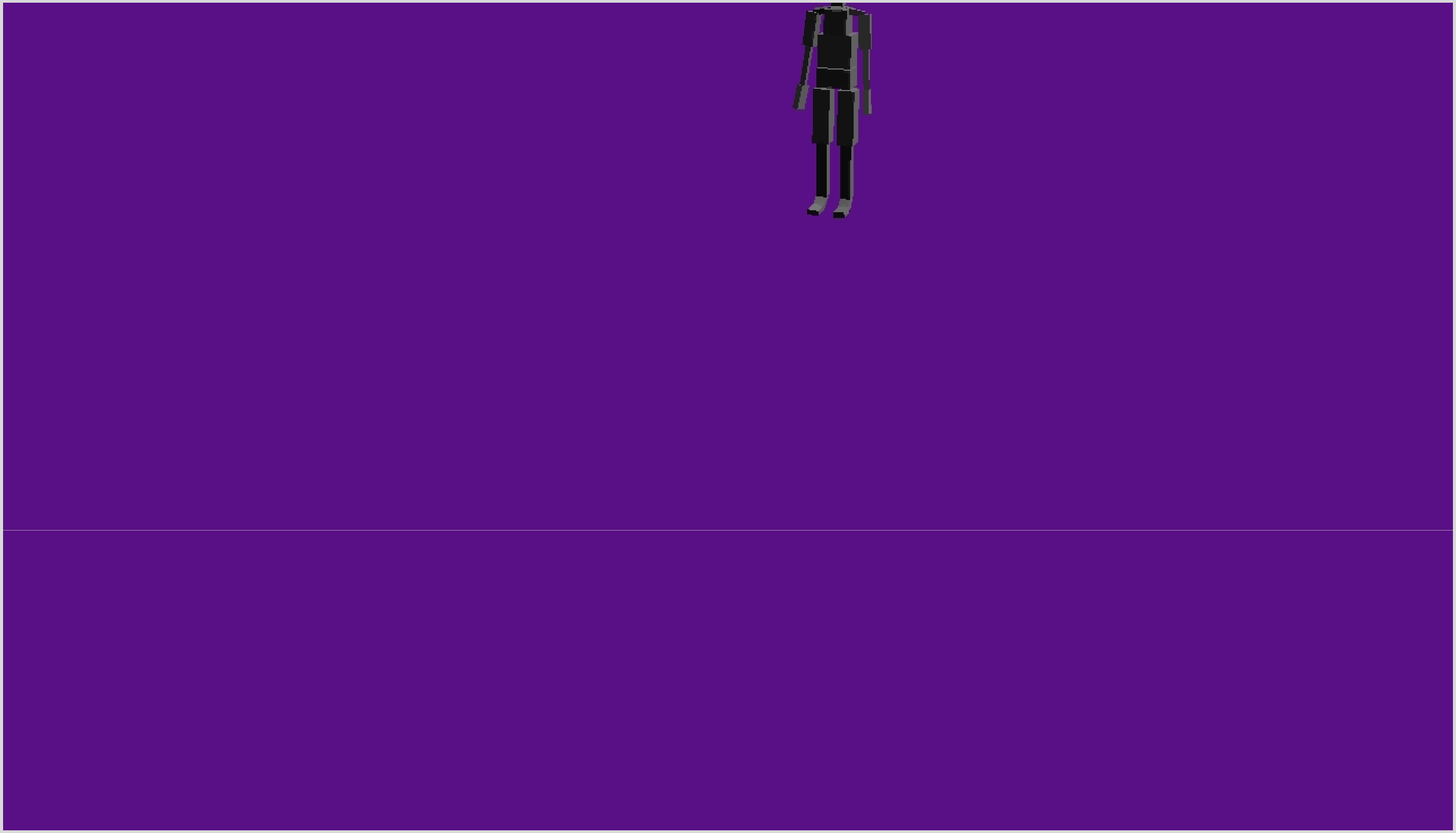
Length of rotation of target

$$\text{and where} \begin{cases} |\Omega_q| = \mathrm{abs}(O_q - O_p) \\ O_q = atan2(\overrightarrow{pq}_y, \overrightarrow{pq}_x) \\ O_p = atan2(\overrightarrow{qp}_y, \overrightarrow{qp}_x) \end{cases}$$

Orientation of target

Orientation of agent

x-dimension of vector from target to agent



uh-oh

A lazy pursuit

Group interaction

Aligning with a group

Position of *i* th target
in neighborhood

$$group\ cohering = seeking\left(\sum_{i \in N} \frac{q_i}{n}\right)$$

Number of neighbors considered

Separating

Monotonic distance-decay damper

Maximum (preferred) acceleration

Vector from agent to target

$$separating = min(\lambda * |\overrightarrow{pq}|^2, a_{max}) * \frac{\overrightarrow{pq}}{|\overrightarrow{pq}|}$$

Position of *i* th target in neighborhood

$$group\ separating = separating\left(\sum_{i \in N} \frac{q_i}{n}\right)$$

Number of neighbors considered

Following paths

Agent's future position

Nudge distance

$$path\text{-}following = \begin{cases} seeking(p' + d_{snap} + p_{offset}), \text{ if } p' \notin path \\ seeking(p' + p_{offset}), \text{ if } p' \in path \end{cases}$$

Position of agent

Small estimate time

where $p' = p + v_p * \hat{t}$,

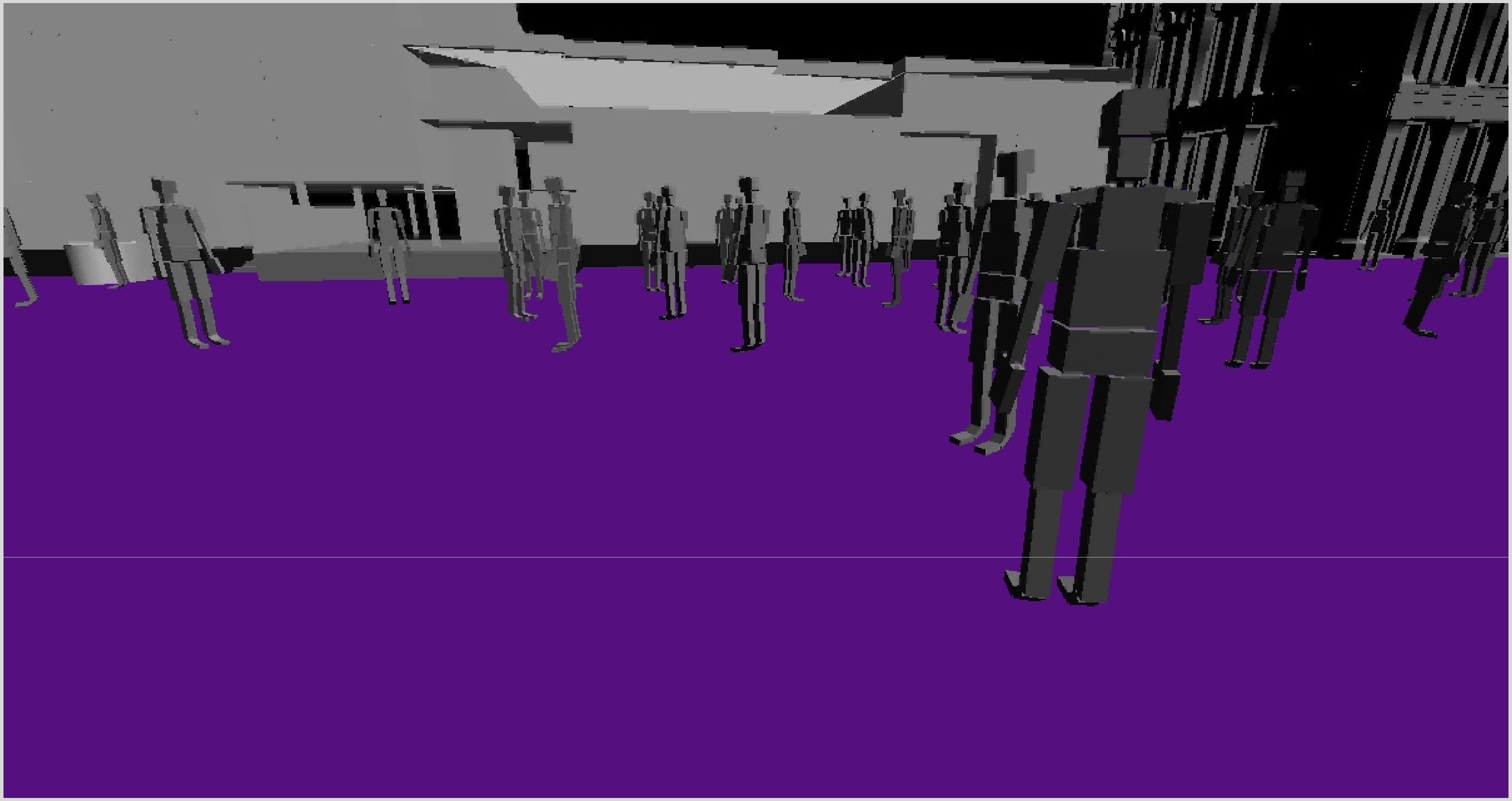
Velocity of agent

and where $d_{snap} = min(q_{path} - p'),\ q_{path} \in path$

Snap distance to path

Future position of agent

Closest line segment on the path



uh-oh

Collision avoidance by steering

Check to see if you are already in collision with the closest target:

Radius of agent

Radius of target

If $d_{minseparation} < (r_p + r_q)$ or $d_{minseparation} \leq 0$, there is already a collision,

where, $d_{minseparation} = |p - q| - |v_q - v_p| * t_{collision}$

Time to potential collision

Position of agent

Position of target

Velocity of target

Velocity of agent

Calculate the time to the nearest collision:

$$\forall q \in N, \qquad t_{collision} = \frac{(p - q) * (v_q - v_p)}{\left| v_q - v_p \right|^2}$$

- Position of agent (p)
- Position of target (q)
- Number of target neighbors (N)
- Time to potential collision ($t_{collision}$)
- Velocity of target (v_q)
- Velocity of agent (v_p)

Steer to avoid a collision:

$$\forall\, q \in N, \qquad seeking\left(\frac{(p-q)'}{|(p-q)'|} * a_{max}\right)$$

$$\text{Where}\begin{cases} (p-q)' = -(p-q), & \text{if there is already a collision} \\ (p-q)' = -(p-q) - \left(\left(v_p - v_q\right) * t_{collision}\right), & \text{if a collision is imminent} \end{cases}$$

Number of target neighbors

Position of agent

Maximum (preferred) acceleration

Position of target

Velocity of agent

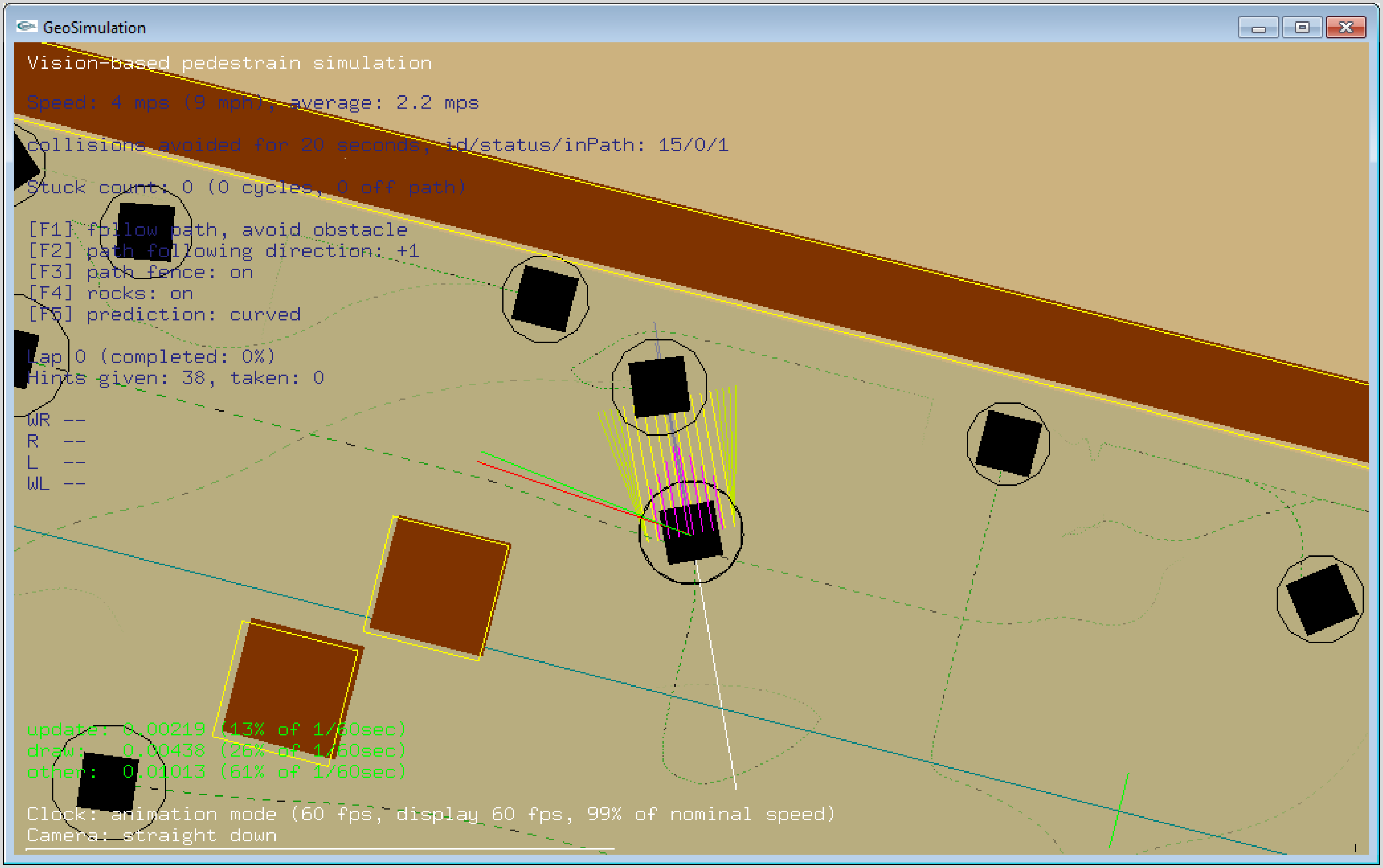
Velocity of target

Time to potential collision

Uh-oh

Using physics to improve collision avoidance

Catalog everything in your visual field, weight it based on focus, digitize it, calculate the potential physics of collision, prioritize your next move

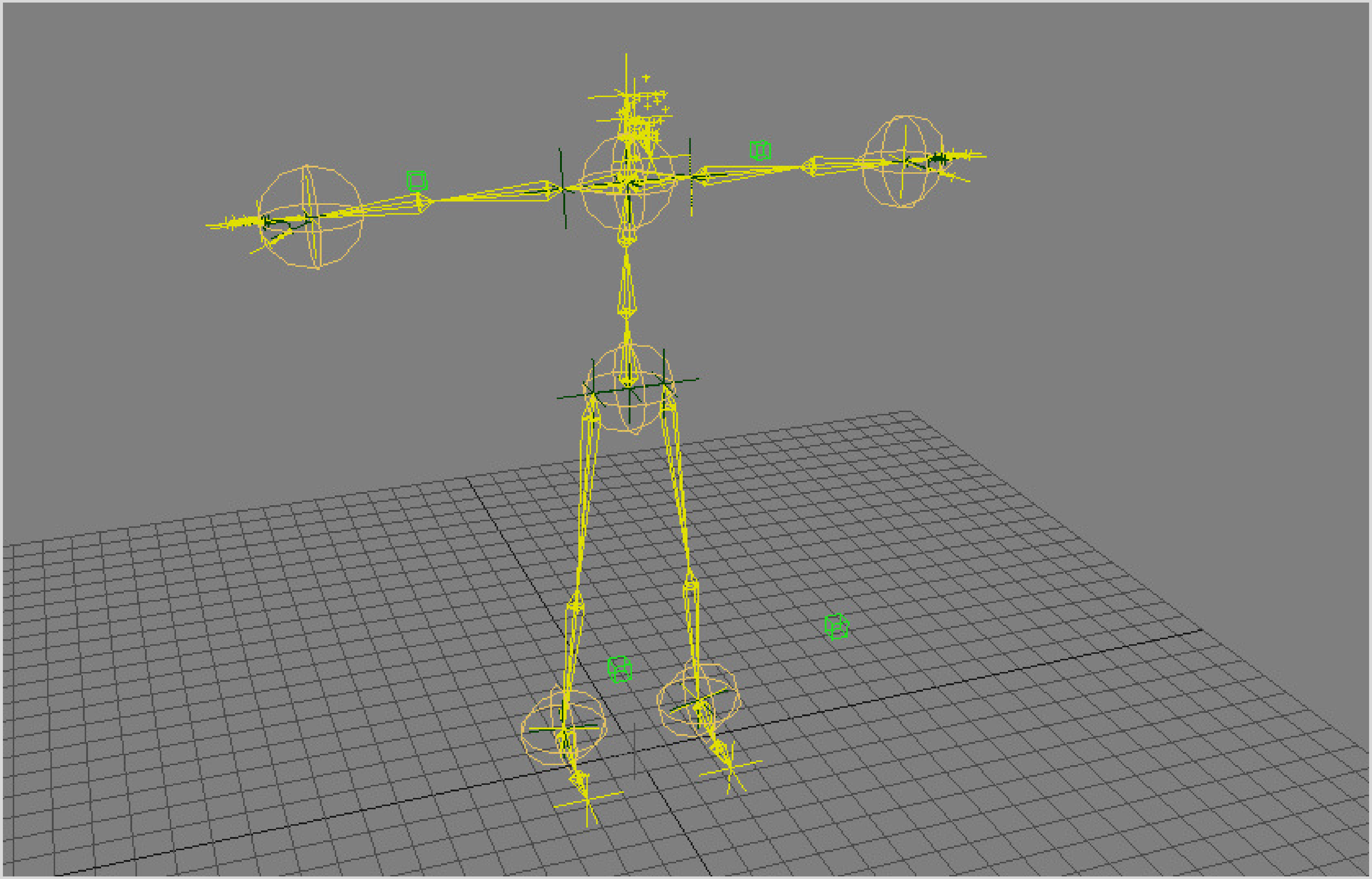


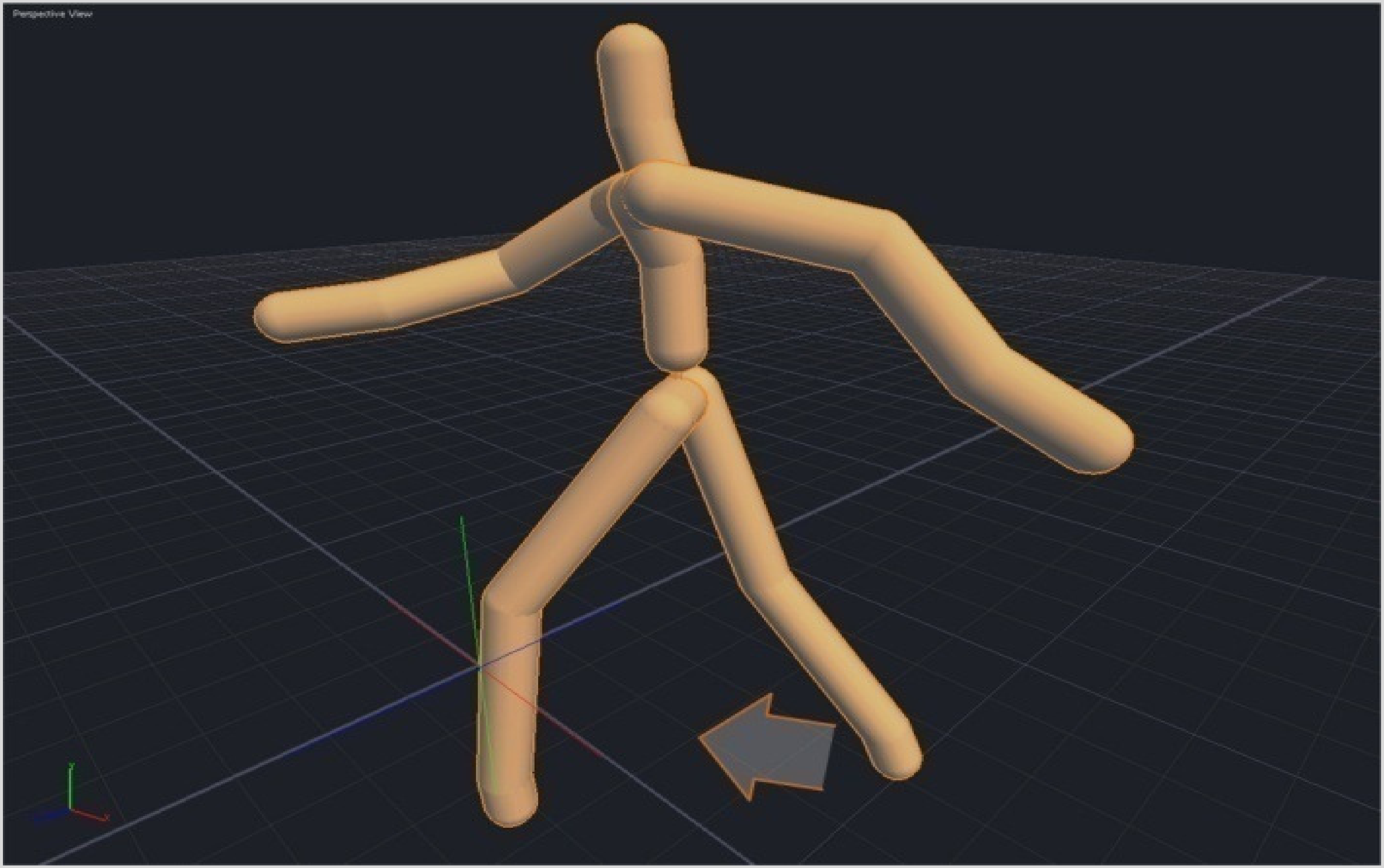
Model

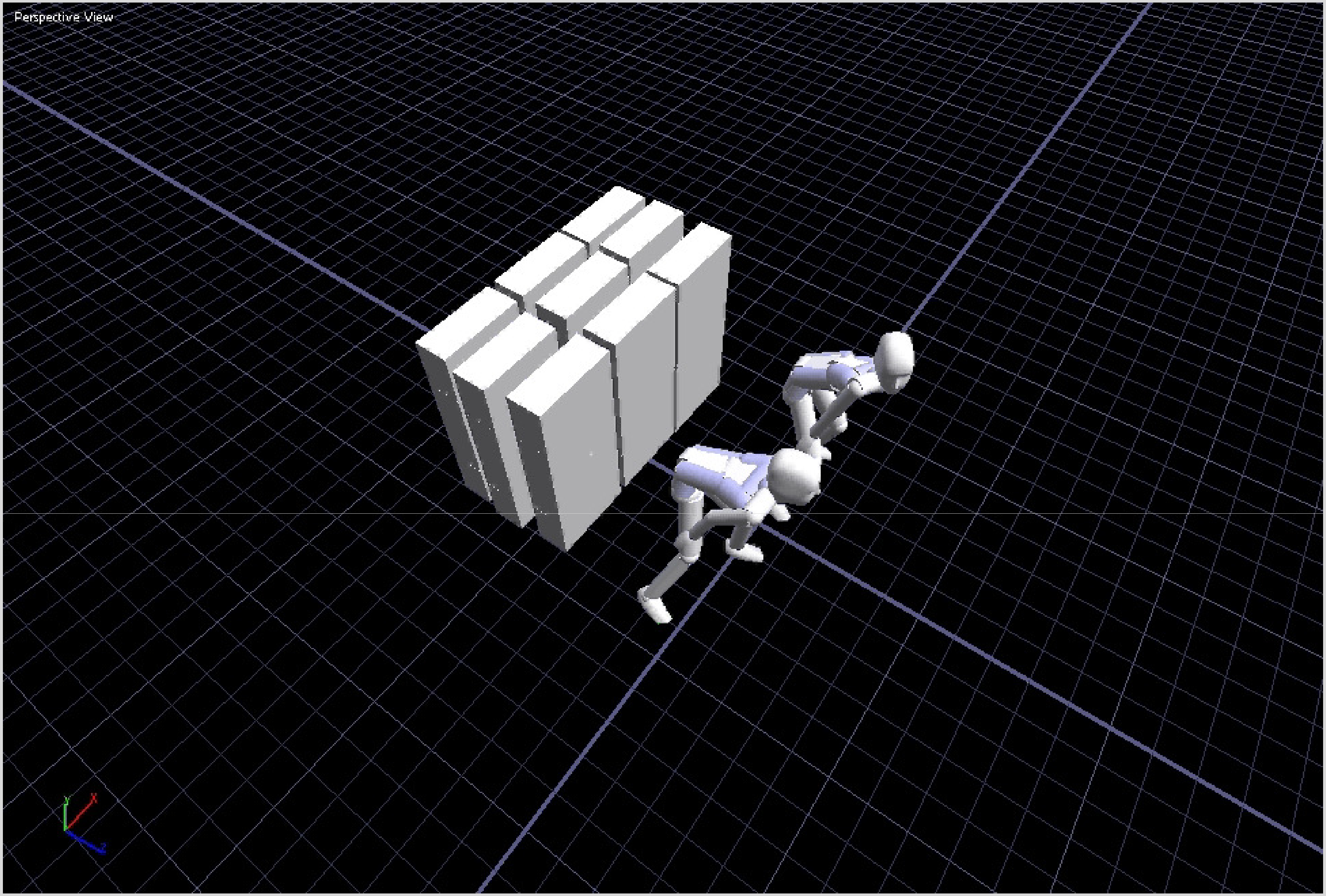
Synthetic vision with adaptive space-time ray-casting

(Work with Haojie Zhu)

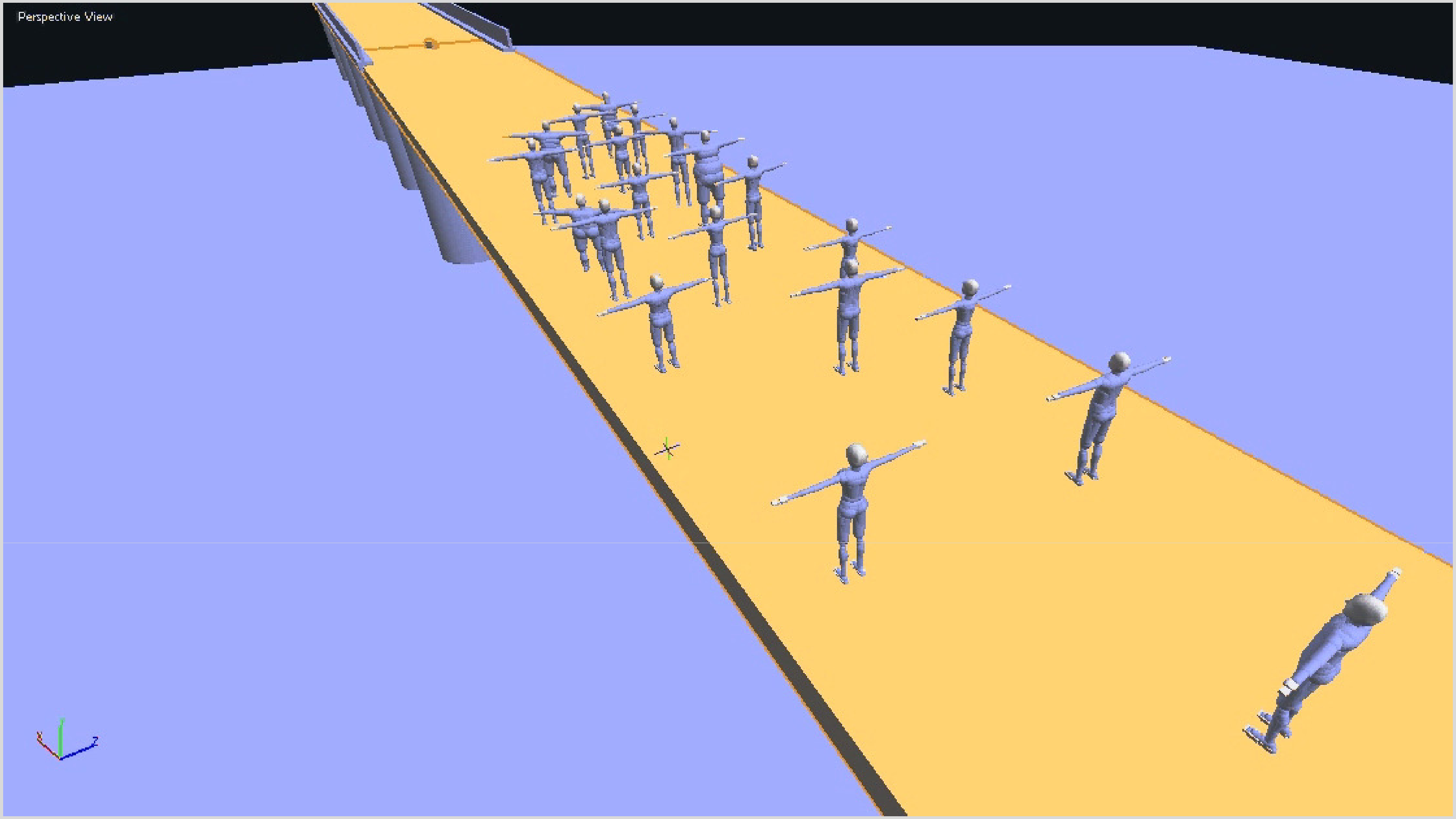
Resolving (very close) physical collisions




Perspective View

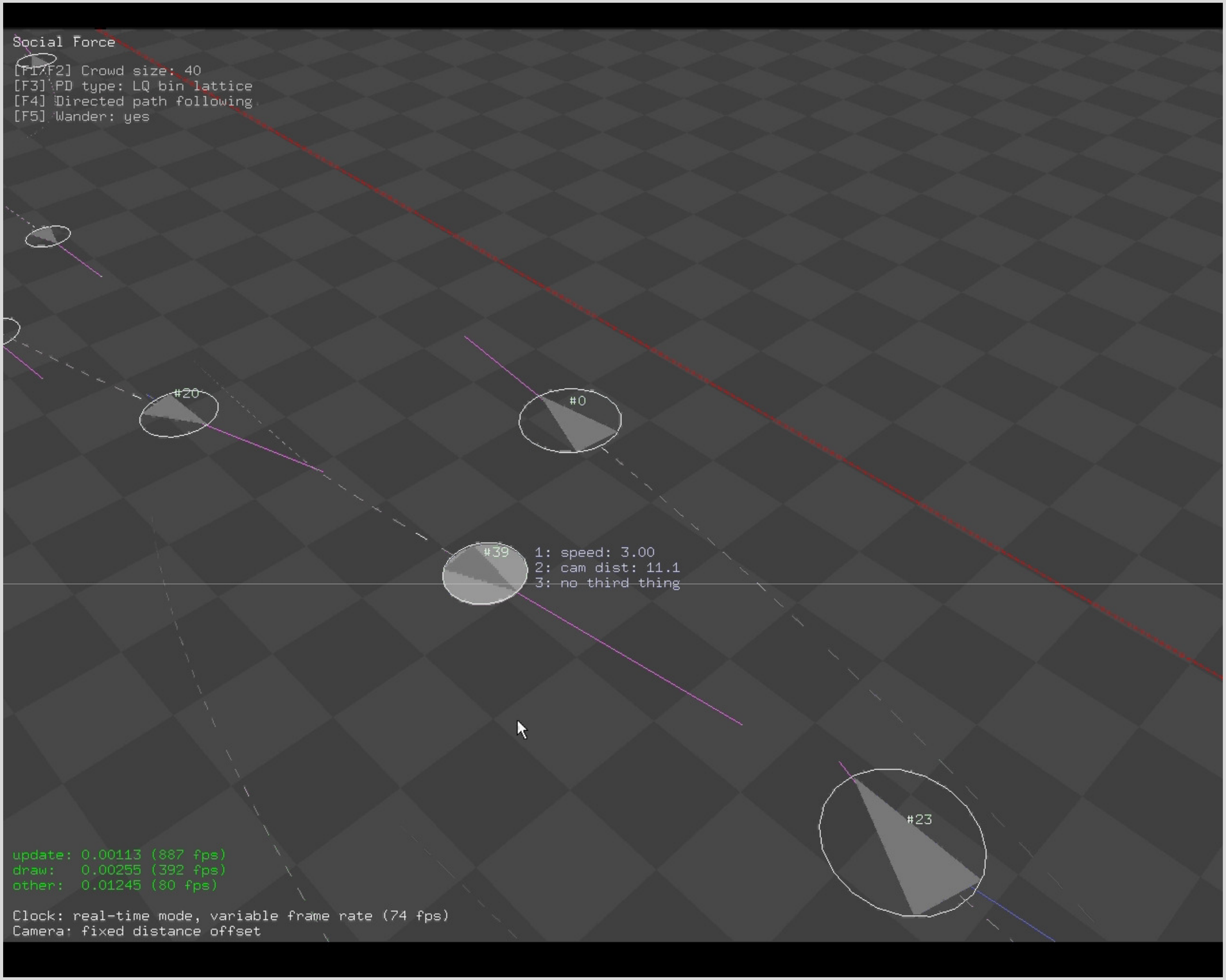


Uh-oh



Uh-oh

Substituting alternative movement schemes



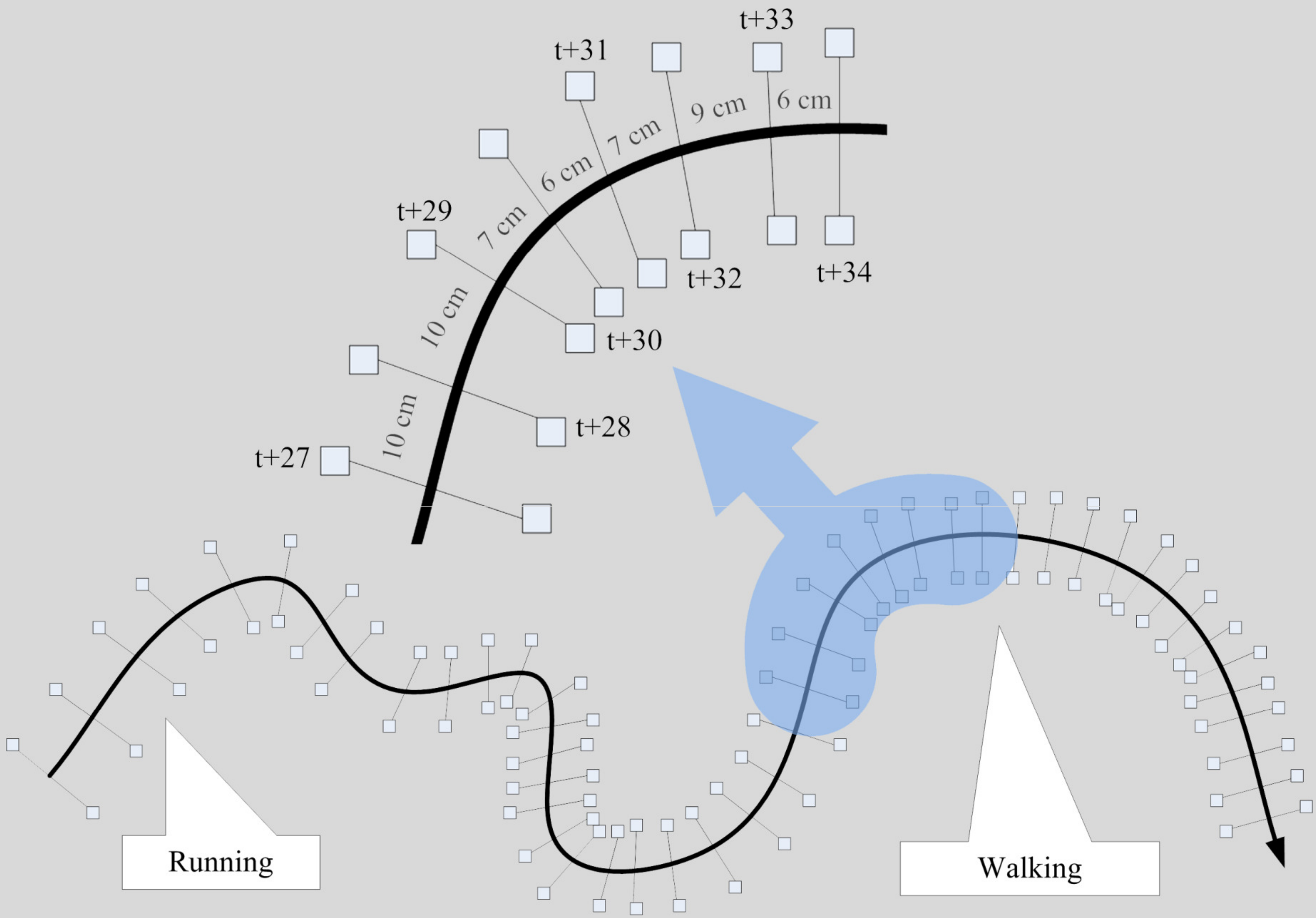
[Model](#) / [uh-oh](#)

Steering by social force

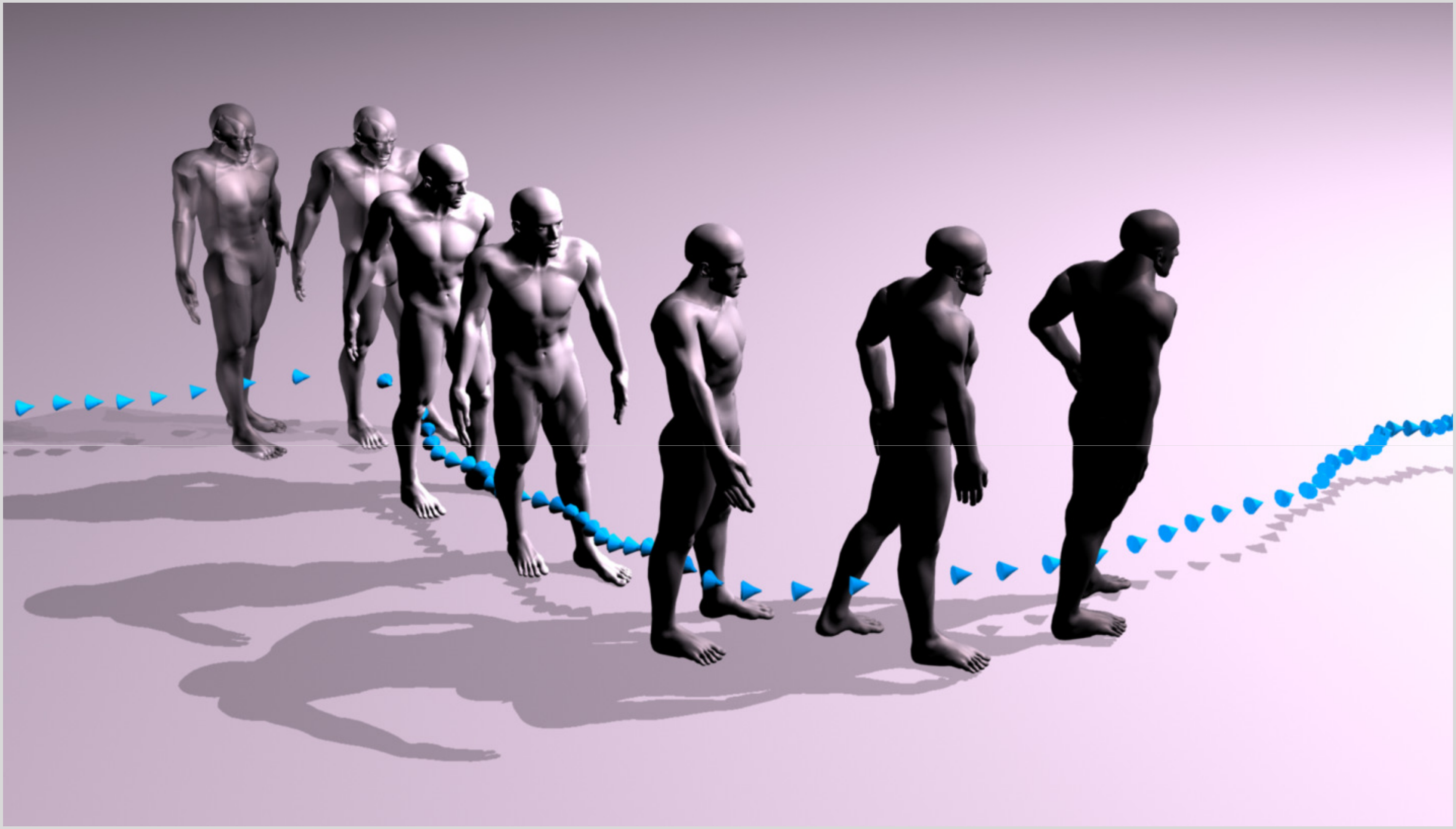
(Work with Haojie Zhu)

Low-level behavior

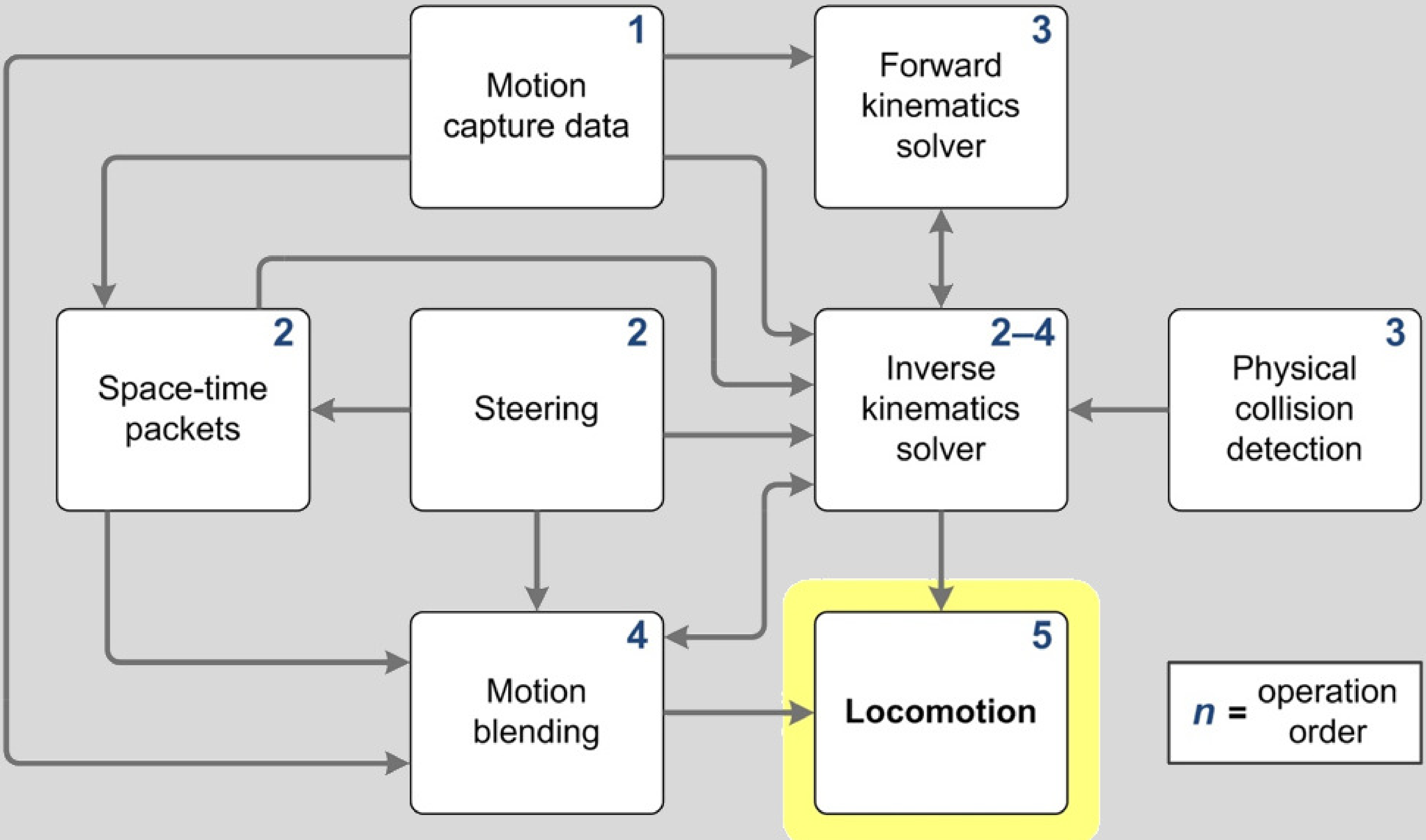
Locomotion



Discrete bundles of space and time as "footsteps"



Inverse/forward kinematics to resolve body locomotion


1
Motion capture data
3
Forward kinematics solver
2
Space-time packets
2
Steering
2–4
Inverse kinematics solver
3
Physical collision detection
4
Motion blending
5
Locomotion
n = operation order

Forward kinematics

Vector for the root node

i th segment

Angle of the joint

$$\overrightarrow{P_n} = \overrightarrow{P_0} + \sum_{j=0}^{i} L_i \left(cos\left(\sum_{j=0}^{i}\theta_j\right), sin\sum_{j=0}^{i}\theta_j \right)$$

Vector for the effector

Length of the *i* th line segment

j th segment

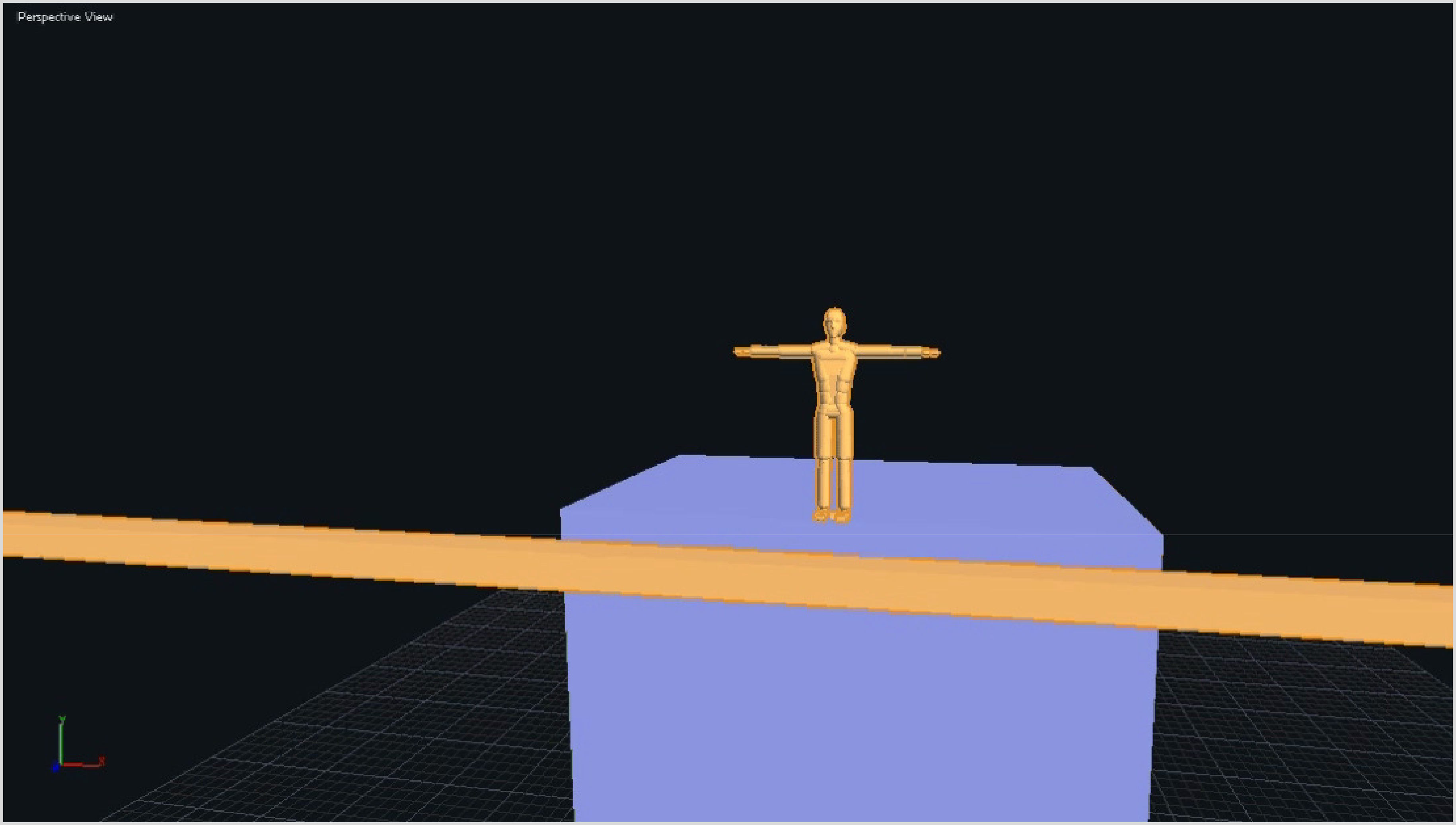
Inverse kinematics

Angle of the joint

(Space-time) goal

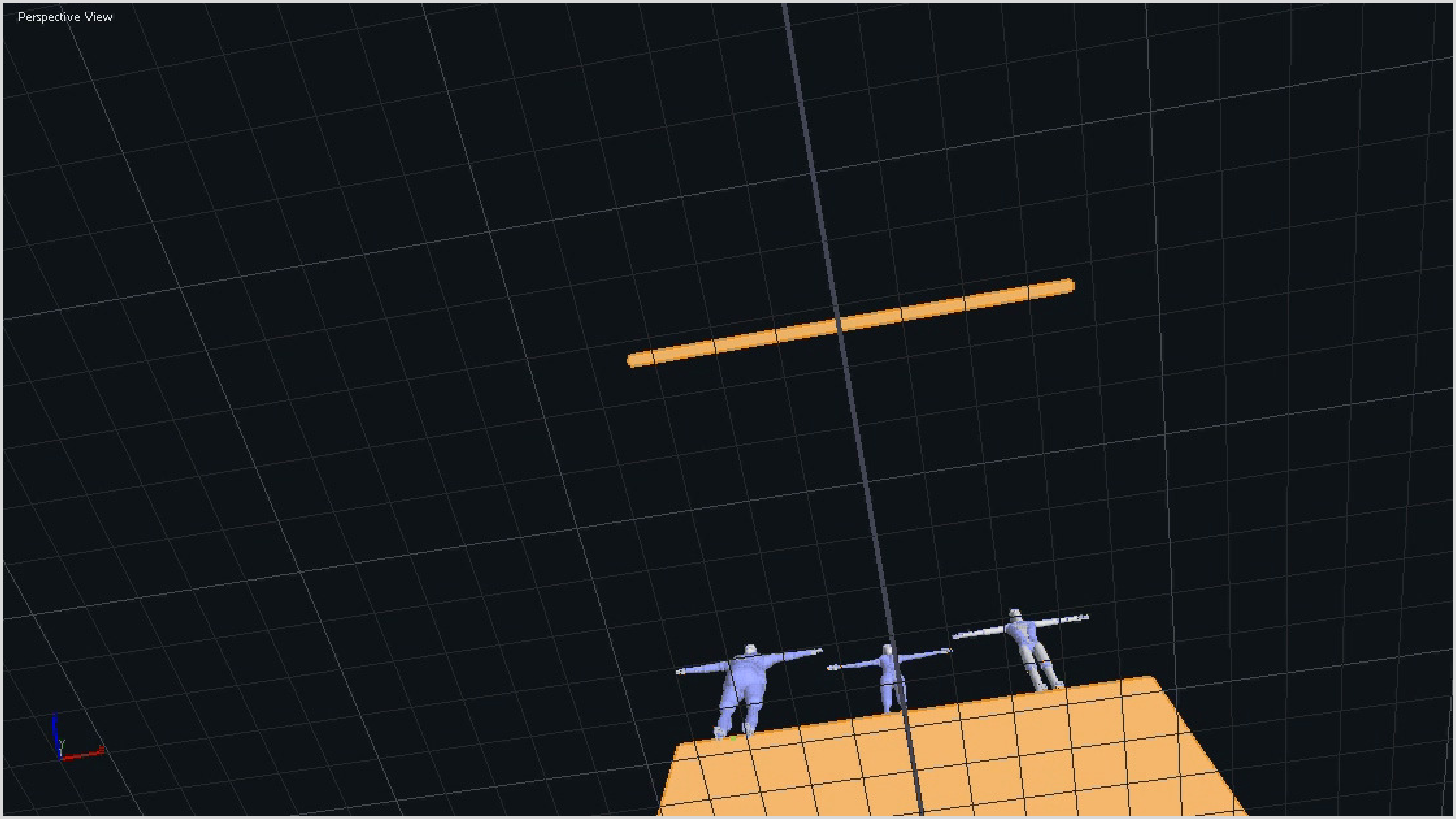
$$\overrightarrow{P_n}(\theta) = G$$

Vector for the effector



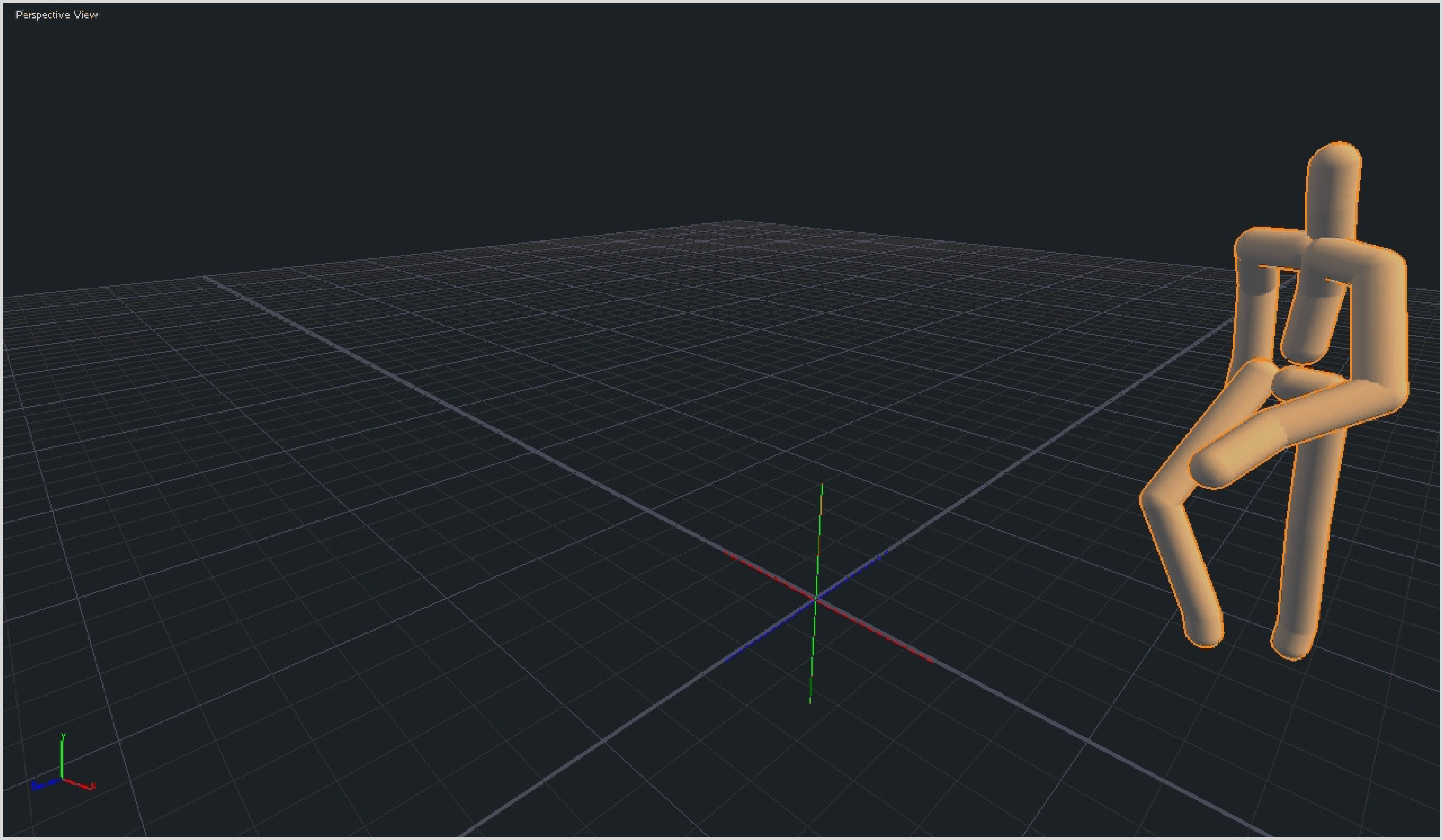
[Uh-oh]

Inverse and forward kinematics



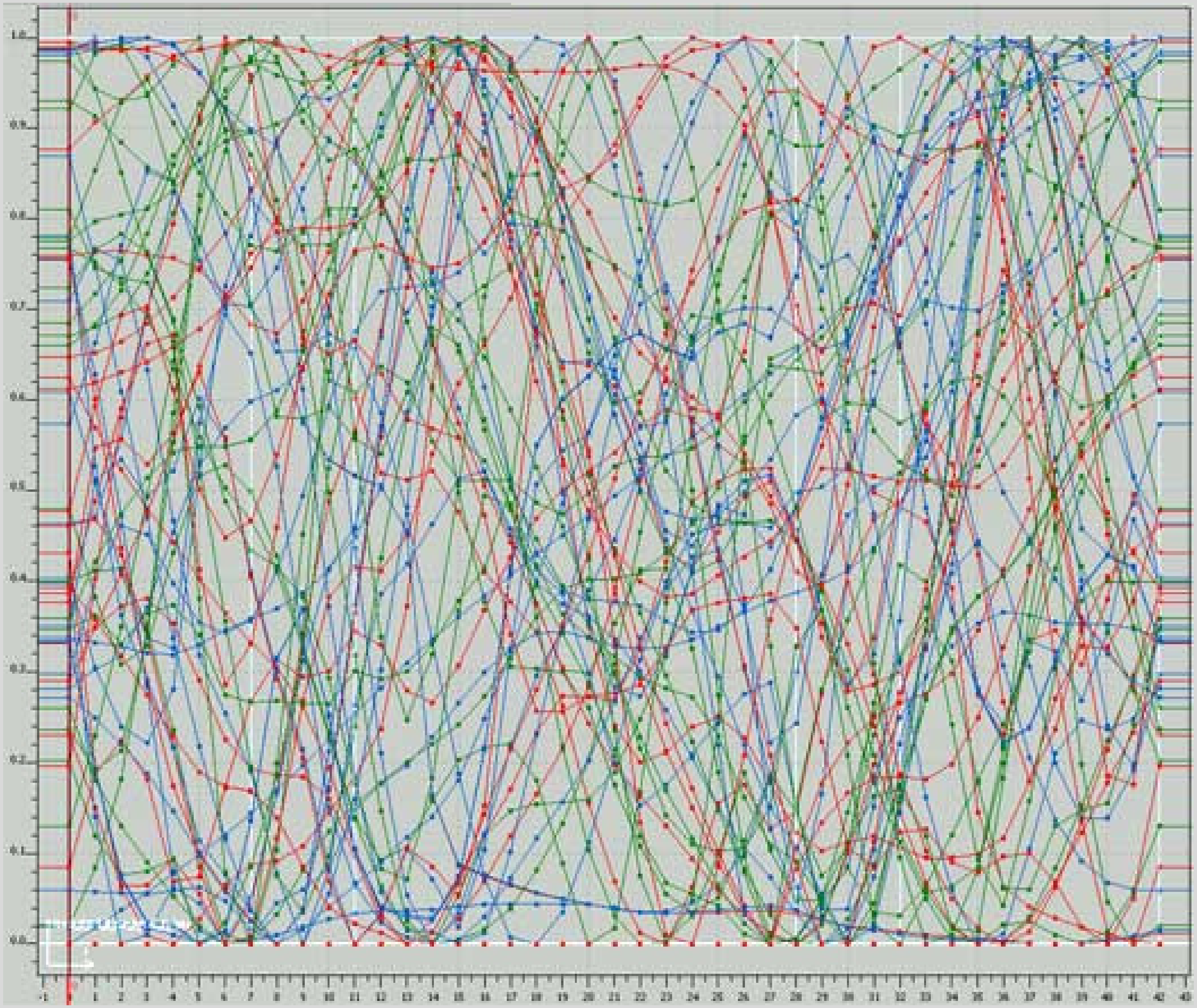
[Uh-oh]

Calibrated choreography with motion capture and editing

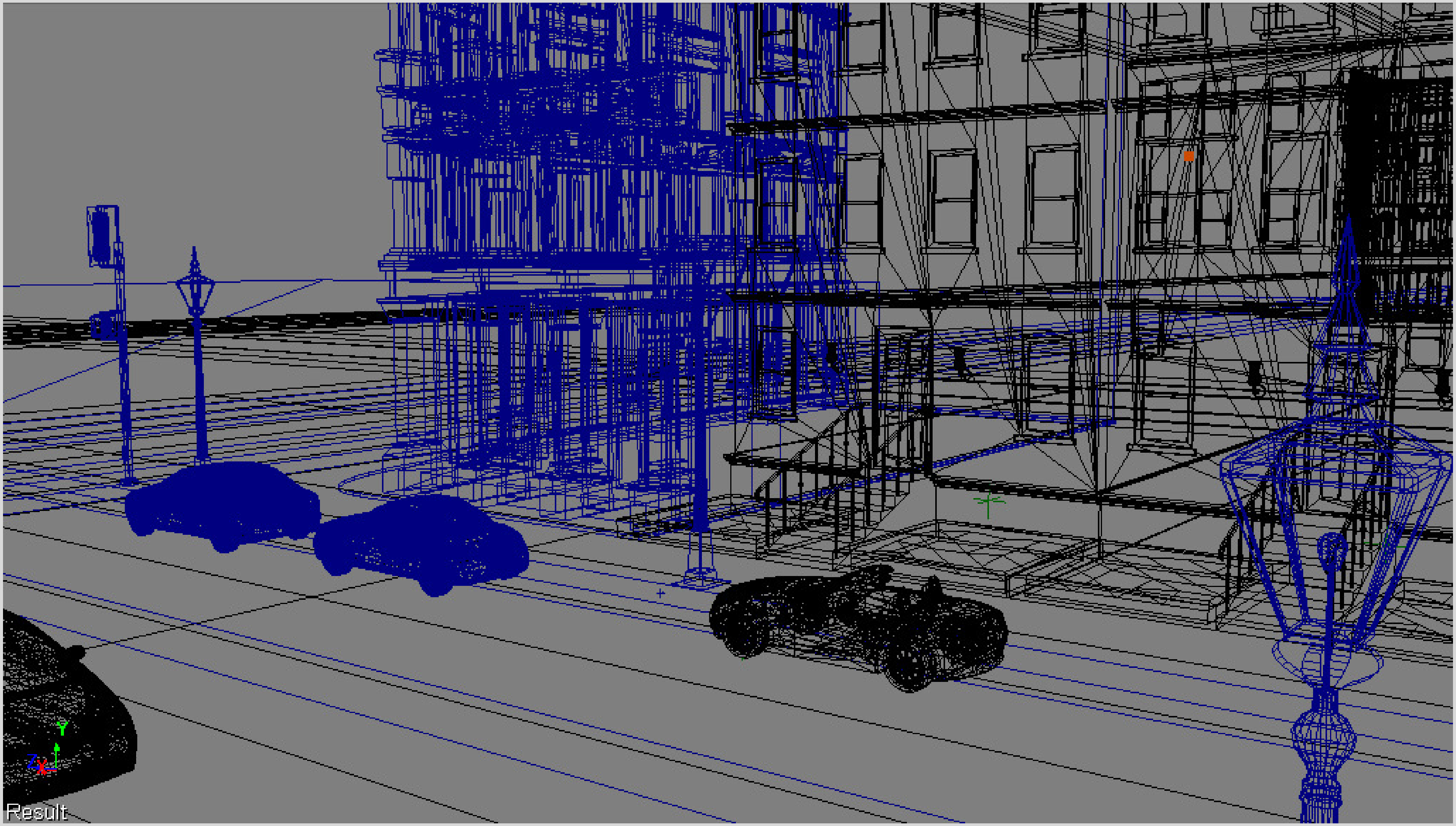


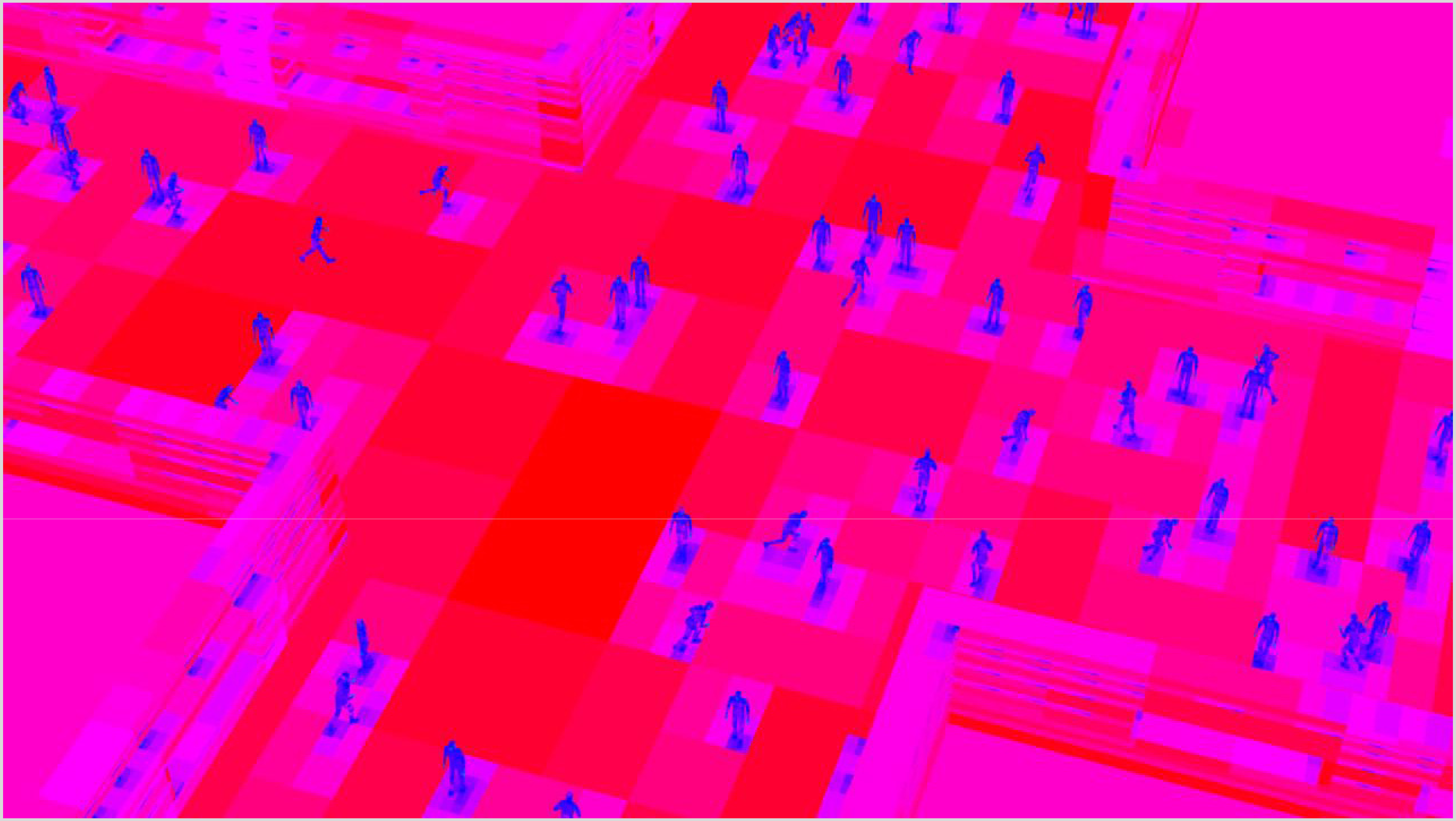
Uh-oh

Motion-captured animation

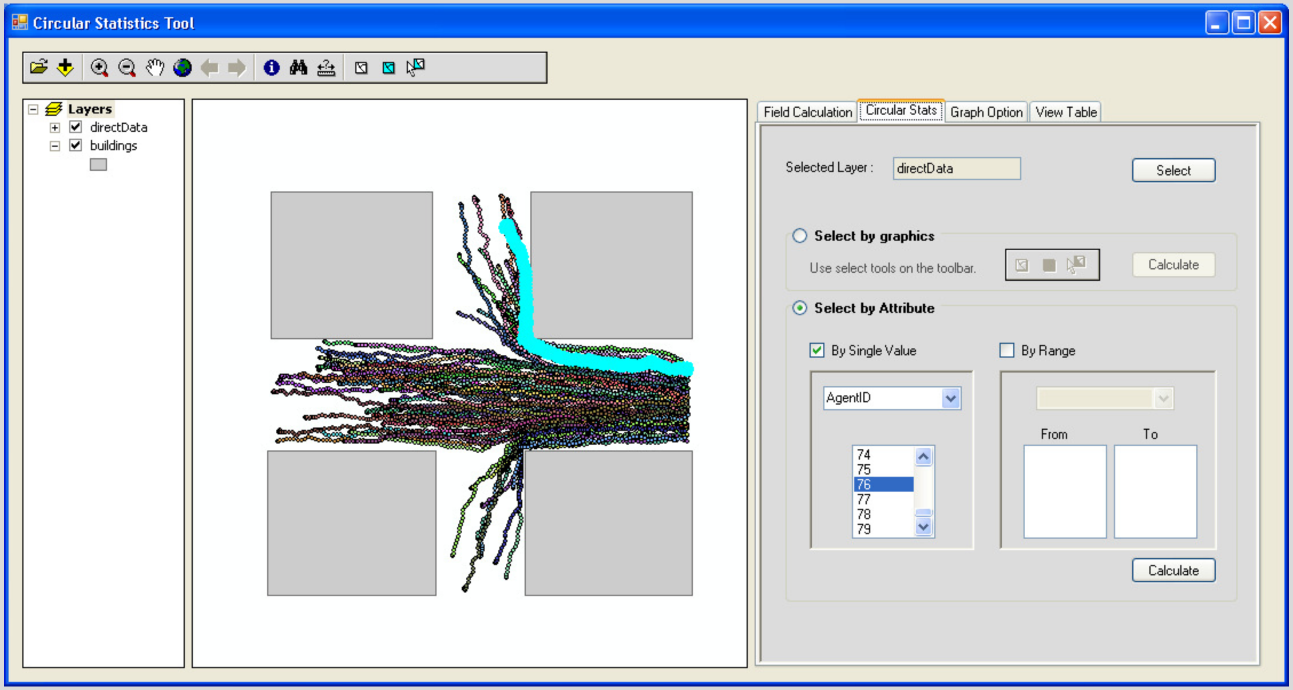

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

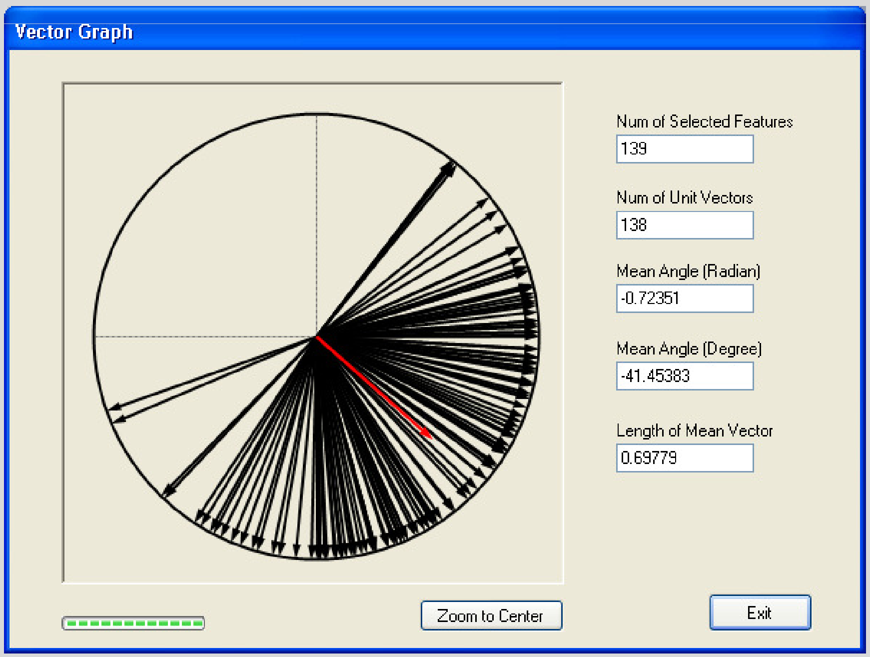
Rendering and animation

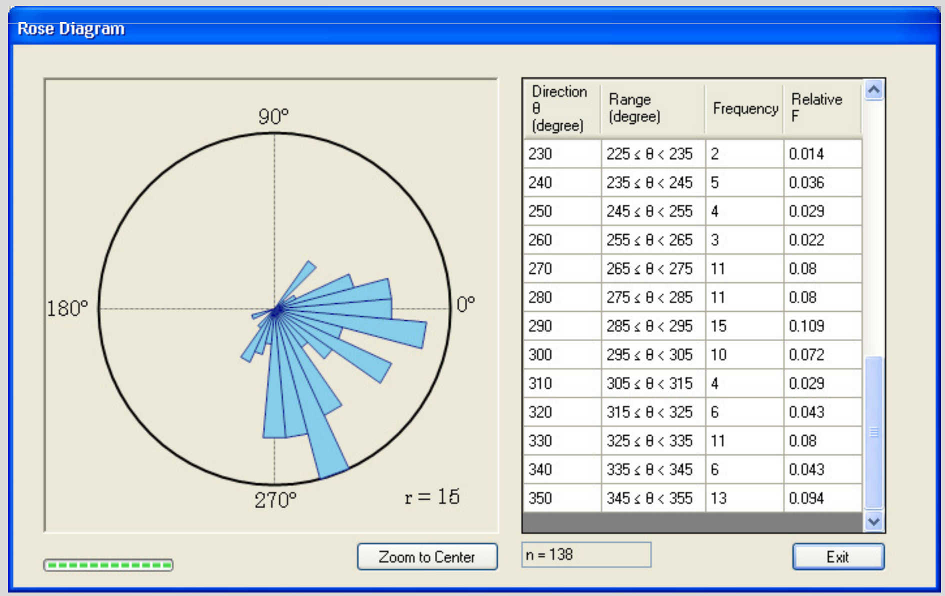

Result



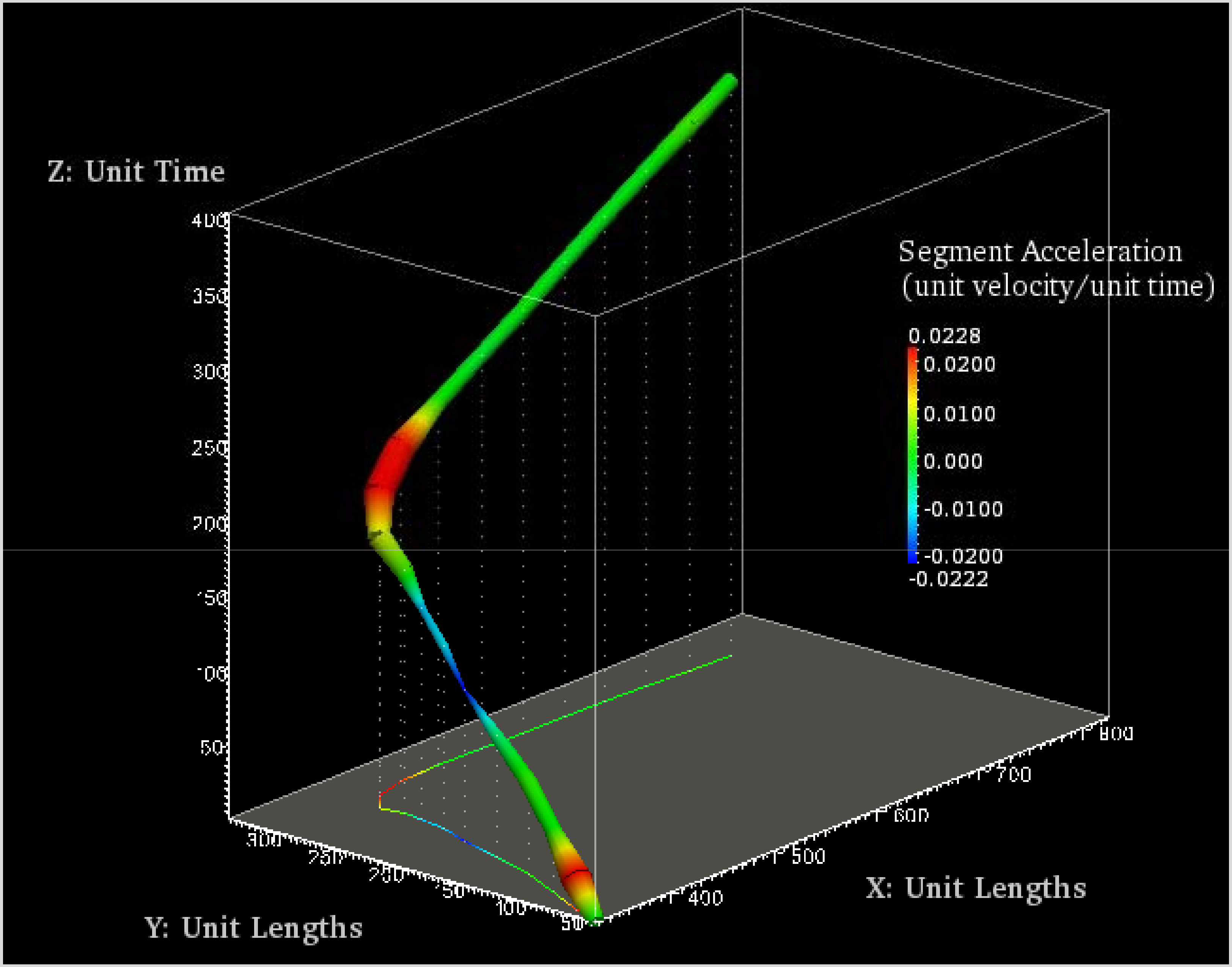
Validation and verification







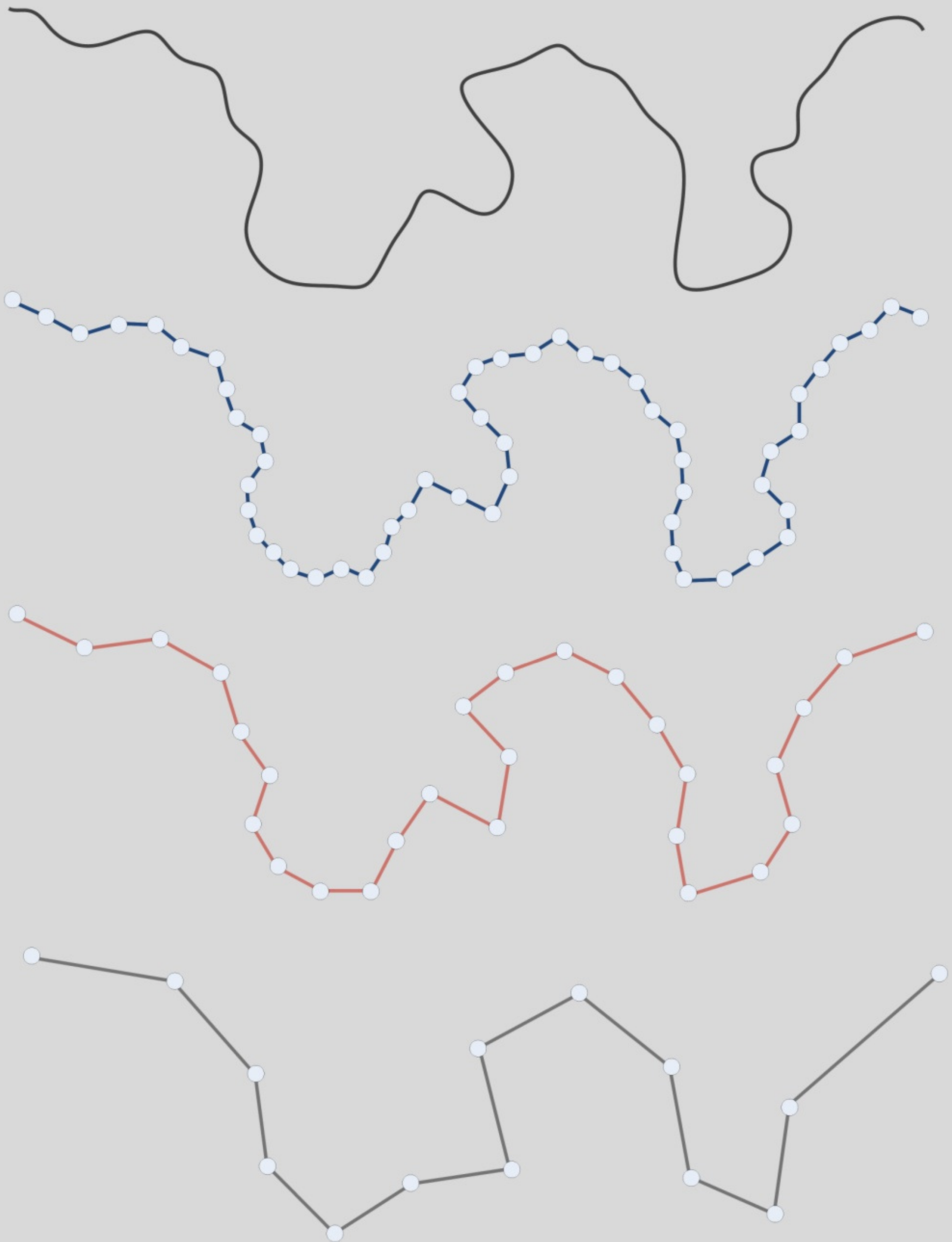
(Work with Atsushi Nara)

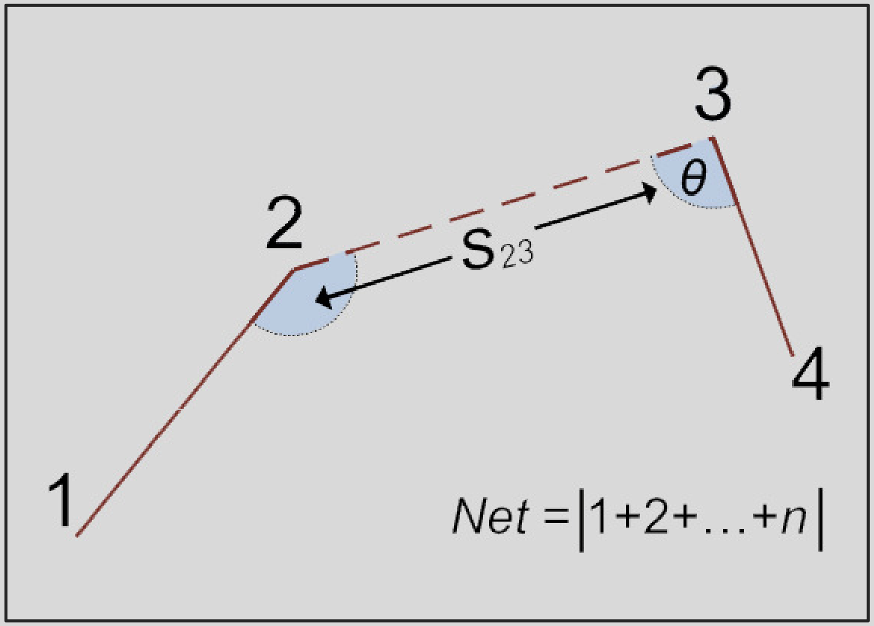


(Work with Atsushi Nara)

- Space-time trajectories
  - Mean cosine of turning angle
  - Probability of turning in the same direction
  - Correlation of the cosine of successive turn angles

- Complexity
  - Mean fractal dimension
  - Approximate entropy





$$d_v = \frac{\log(2)}{\log\left(\frac{Net}{step}\right)} = \frac{\theta}{1 + \log_2(\cos\theta + 1)}$$

Predefined criterion

Data-stream

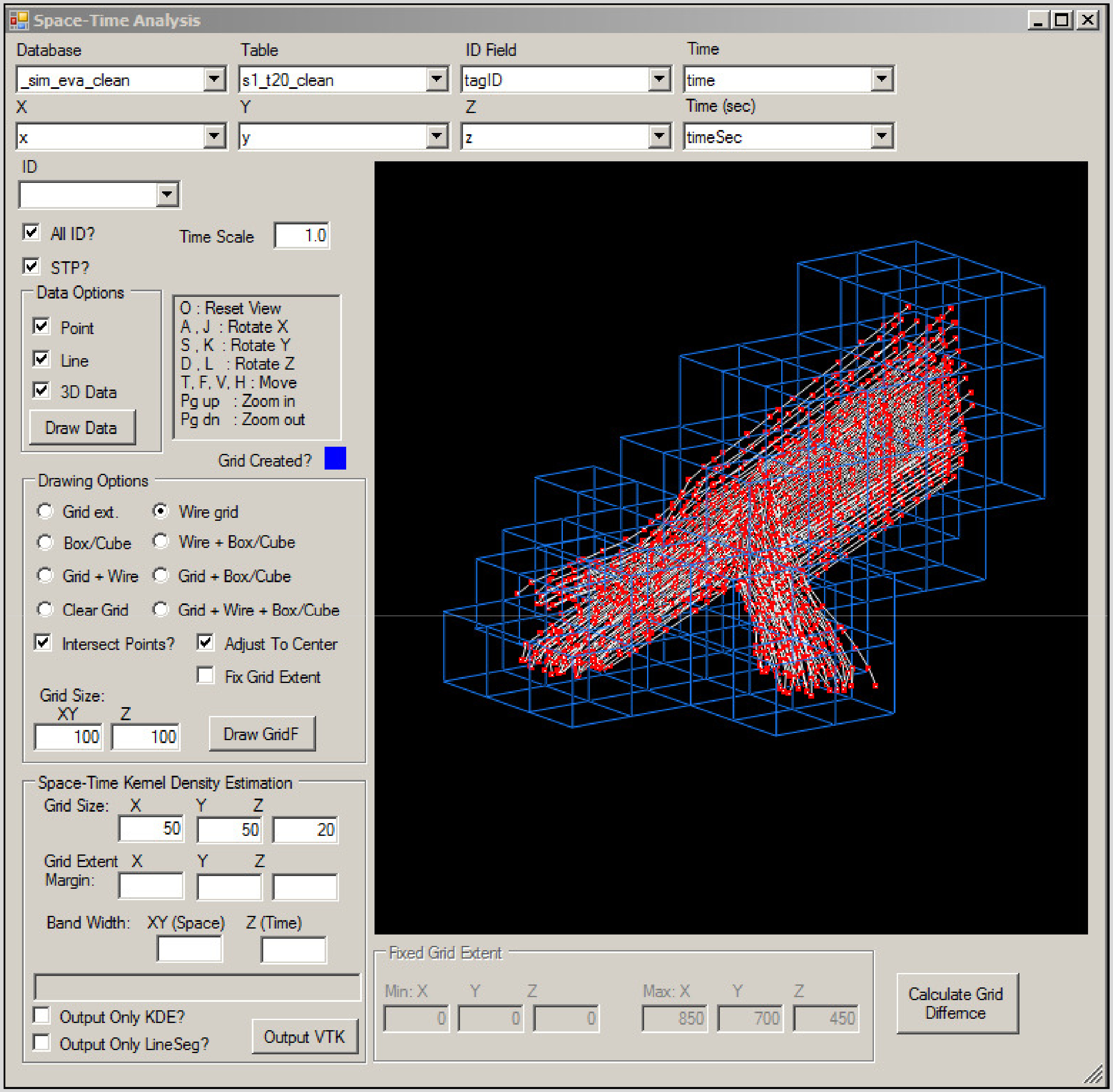
$$ApEn(S_N, m, b) = \ln\left[\frac{C_m(b)}{C_{m+1}(b)}\right]$$

Observations

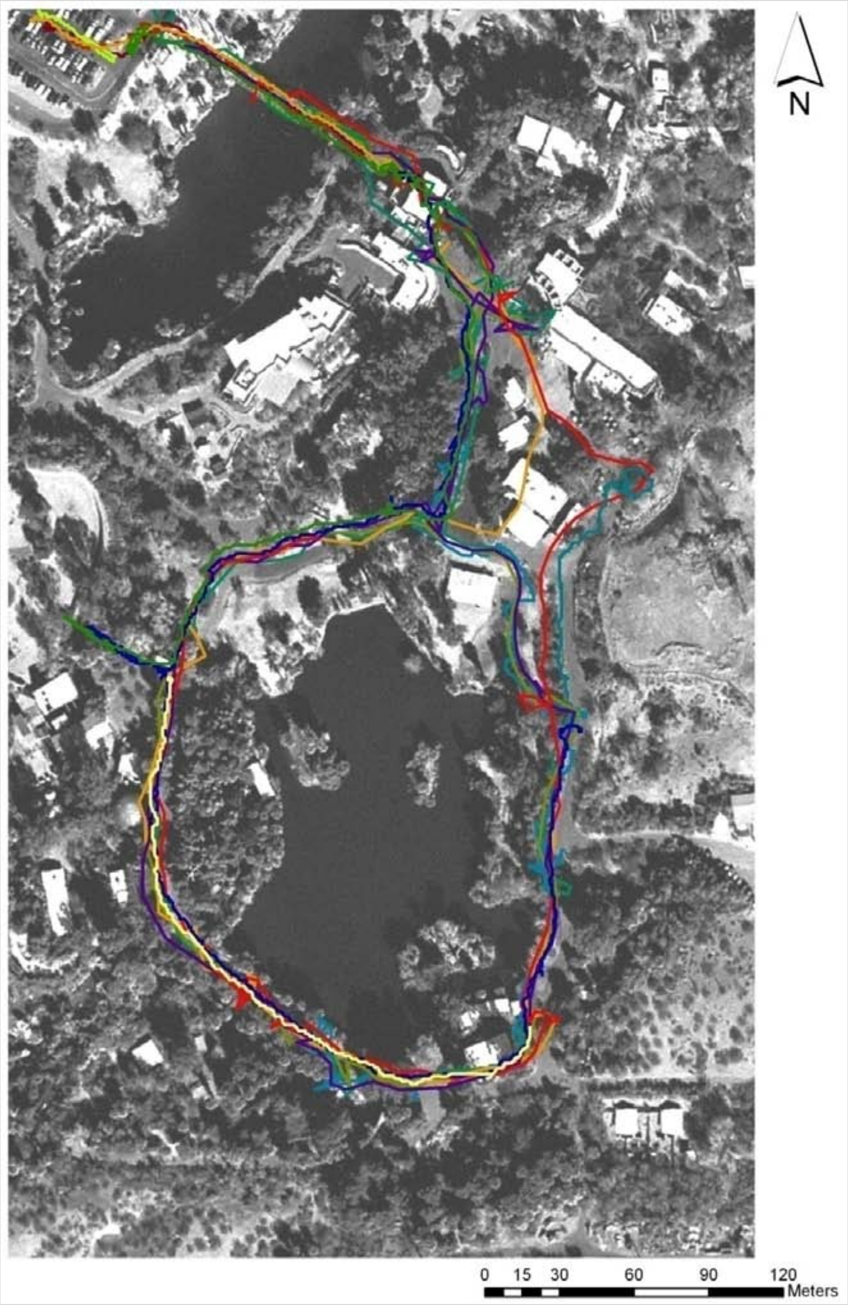
Sequence

$$C_m(b) = \sum_{i=1}^{N-m+1} n_{im}(b) \Big/ (N-m+1)^2$$

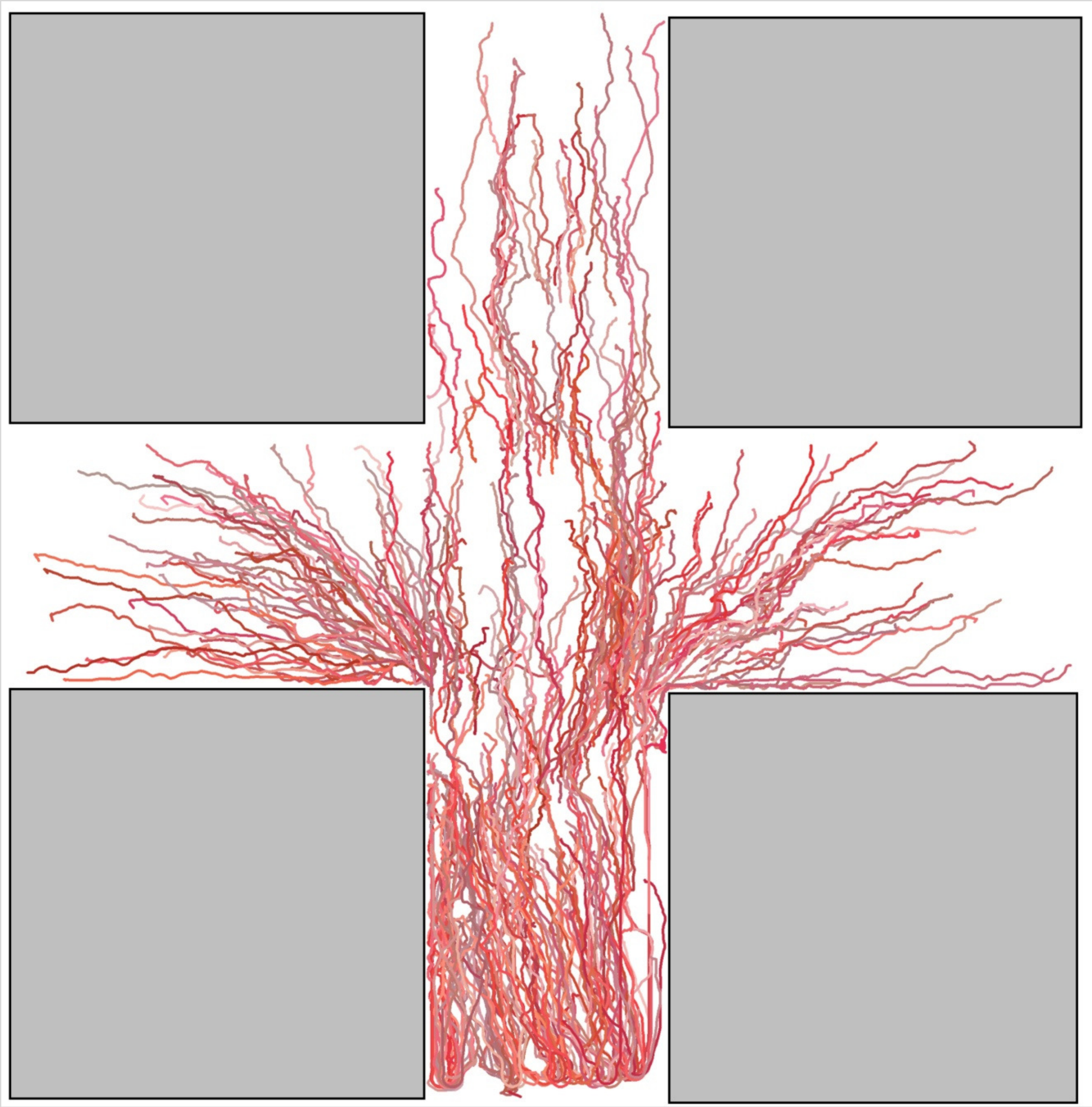
The number of patterns in P_m that are similar to $p_m(i)$



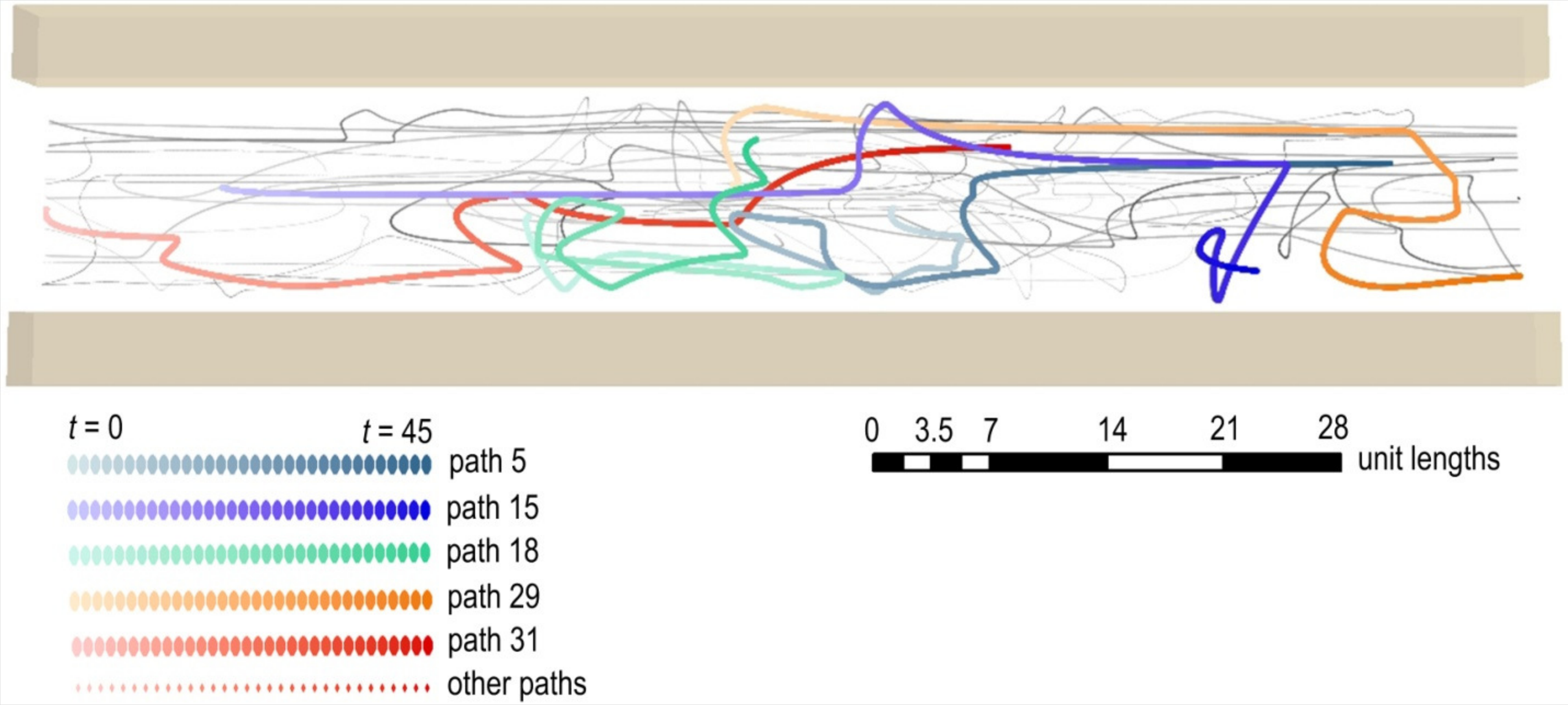
(Work with Atsushi Nara)



Phoenix Zoo Lights festival (FD =1.23; MFD =1.2)



Behavioral model (male, walking) (FD =1.1; MFD =1.08)



Social force model (FD =1.53; MFD = 1.34)

Some applications

Simulating downtown streetscapes

Uh-oh

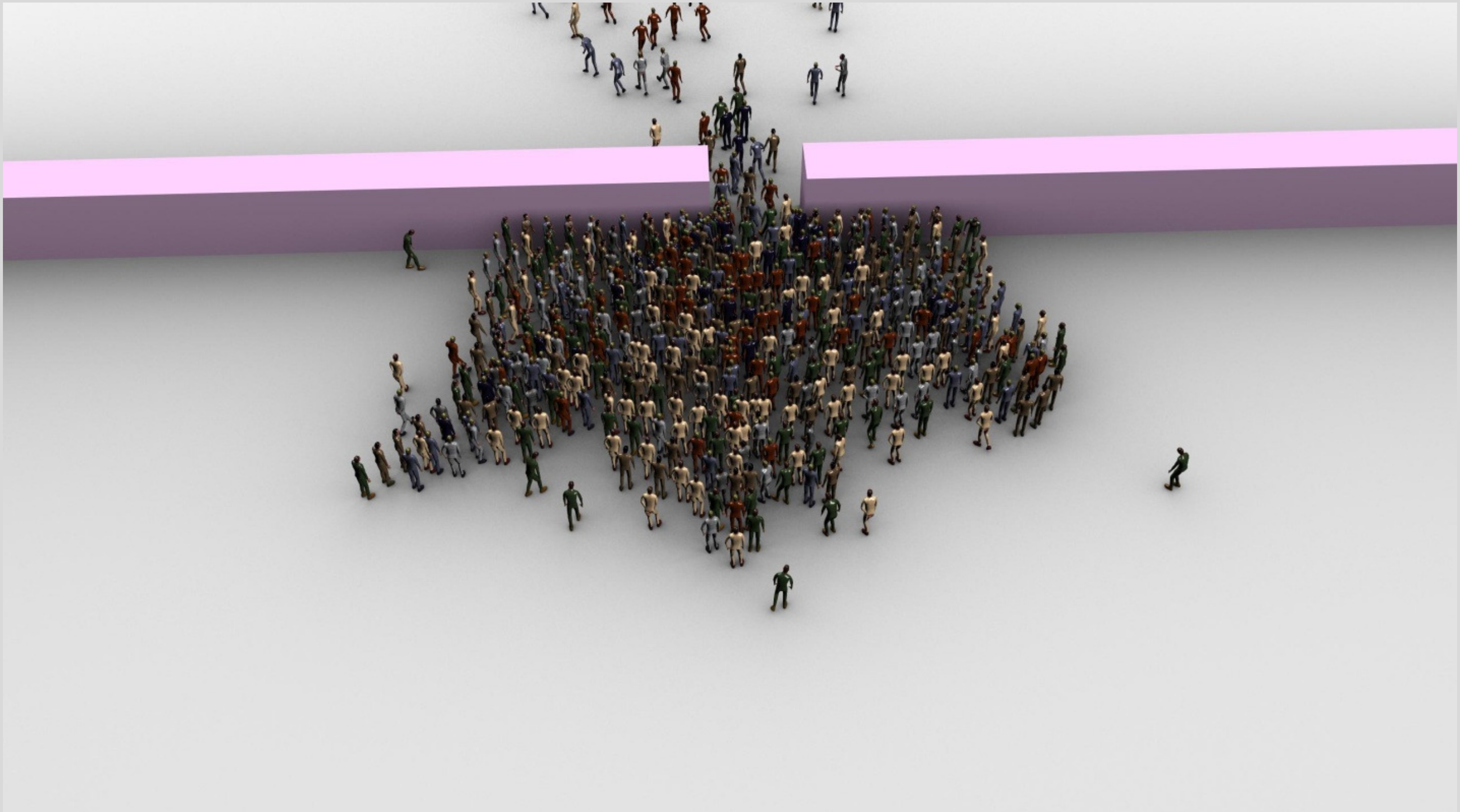
Simulating street overloading

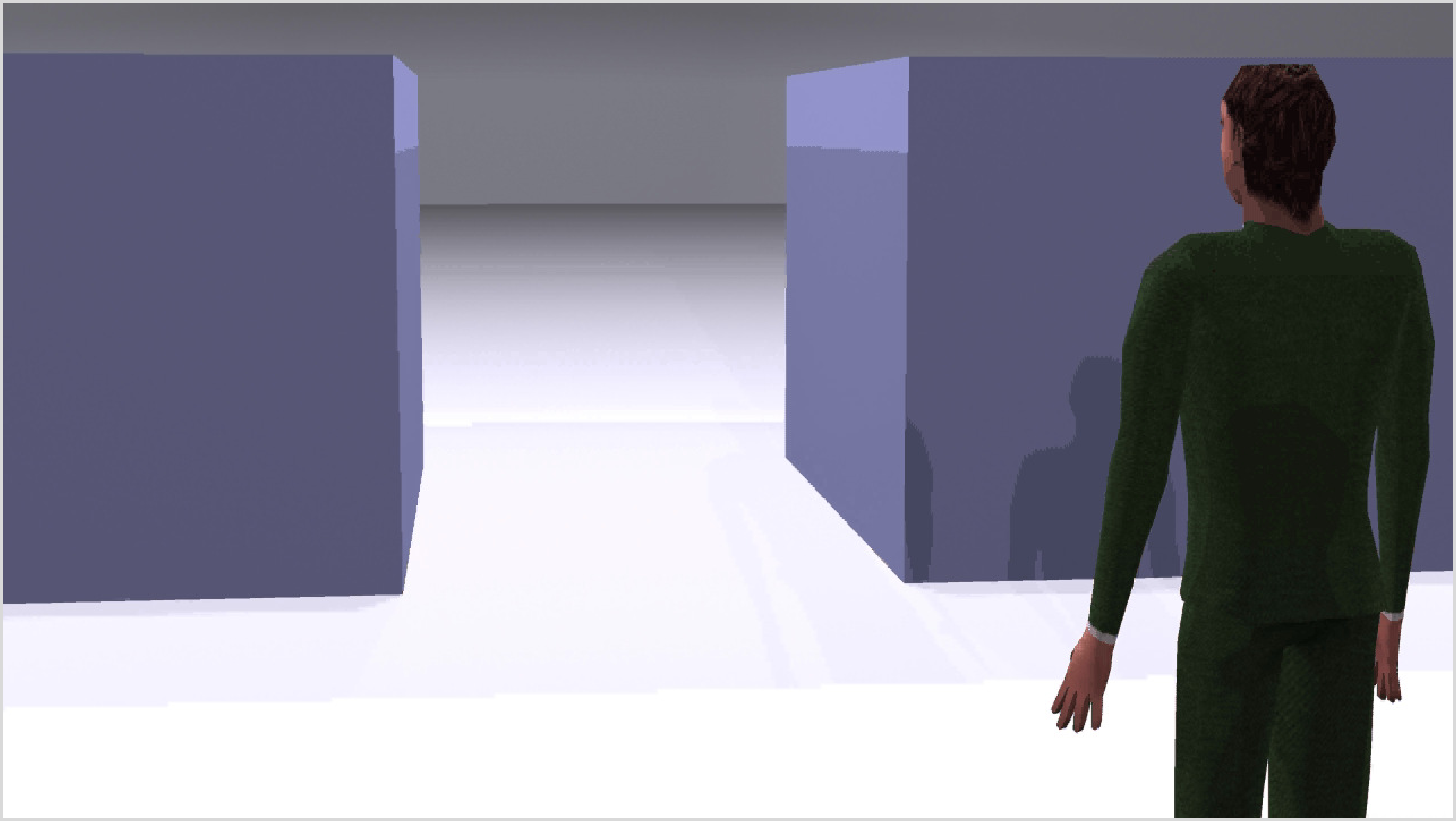
Street crowd, New York City blackout (August 15, 2003)

(Photo: John Wehr)

[Uh-oh]

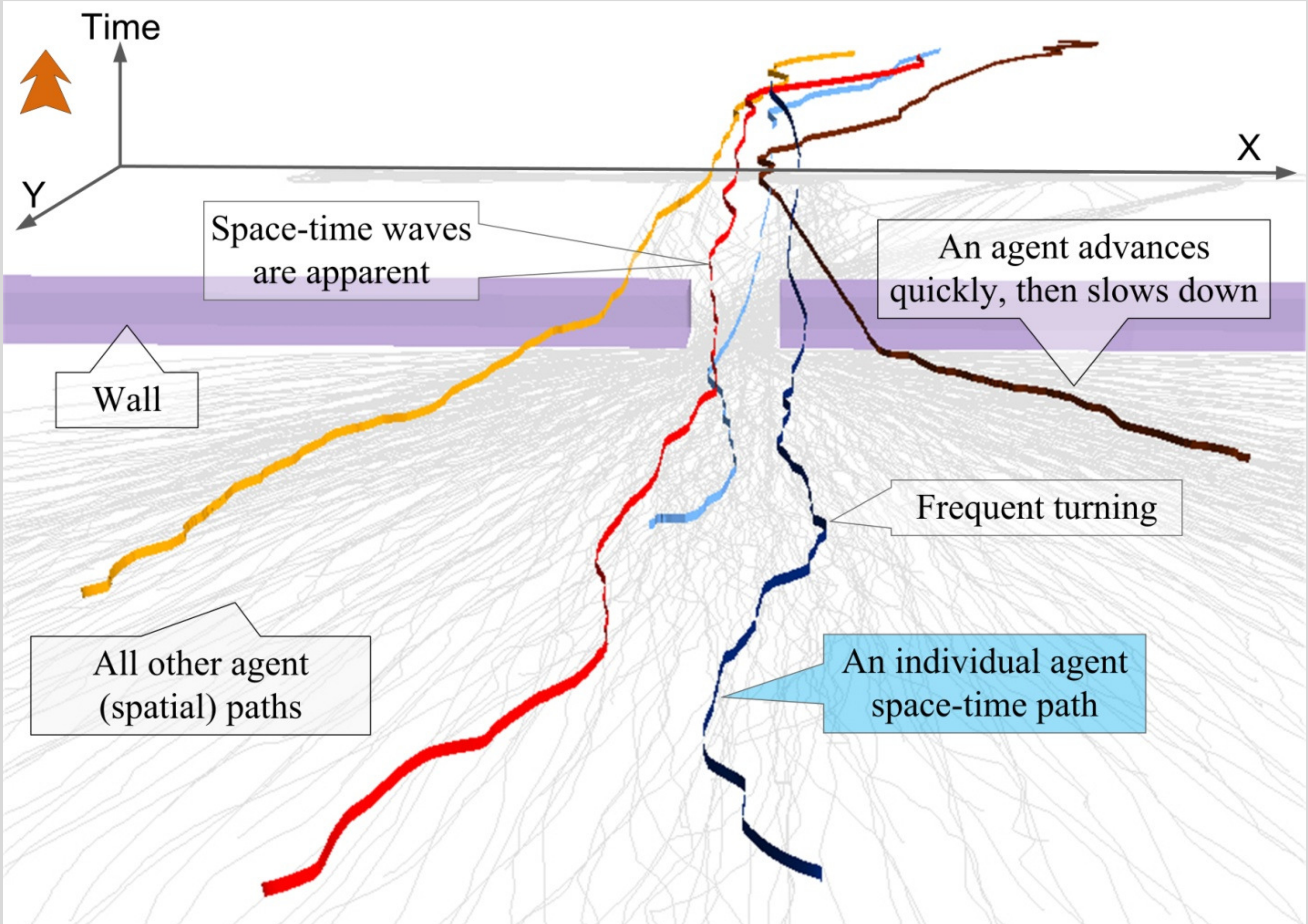
Simulating emergency egress

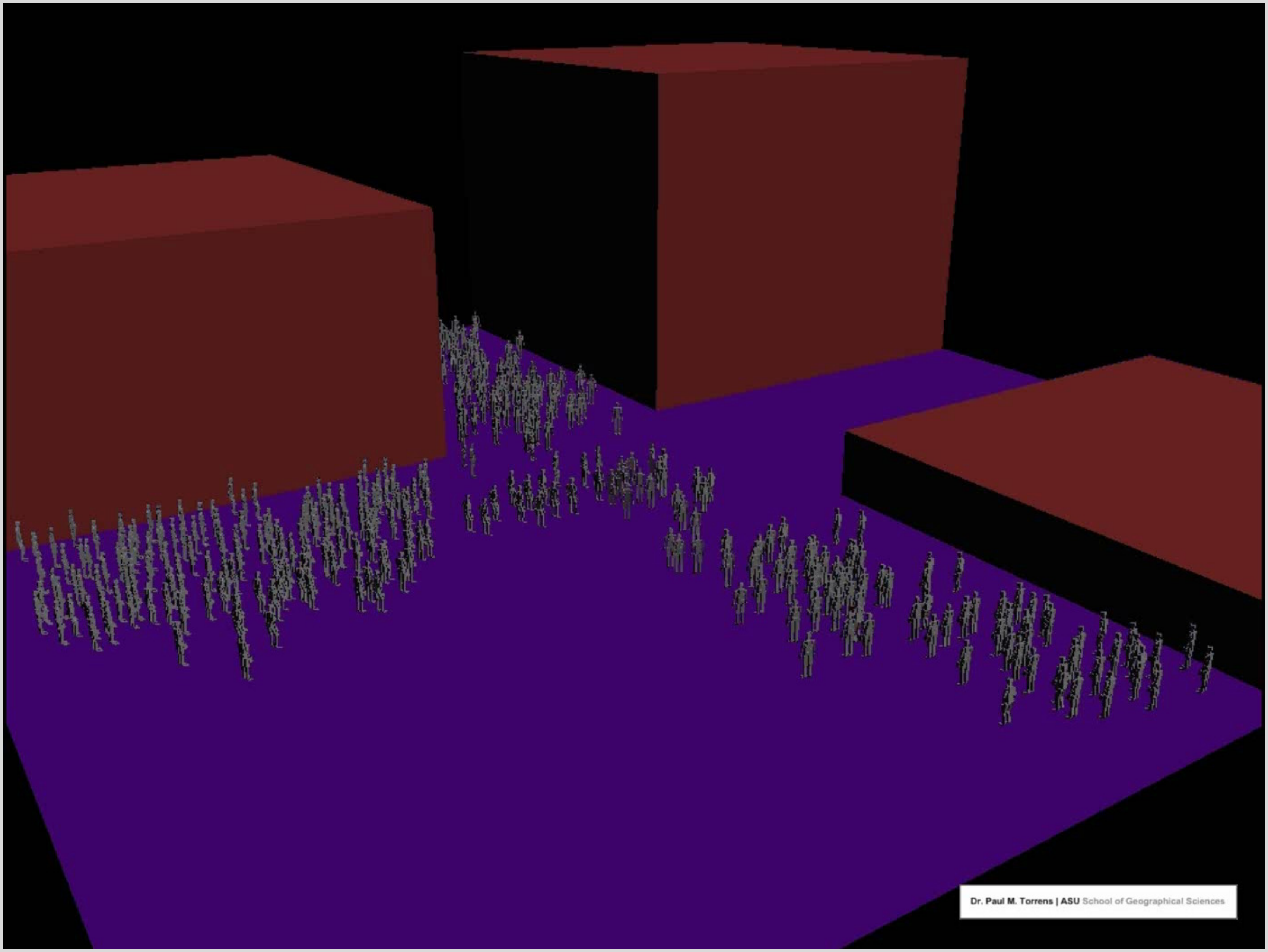




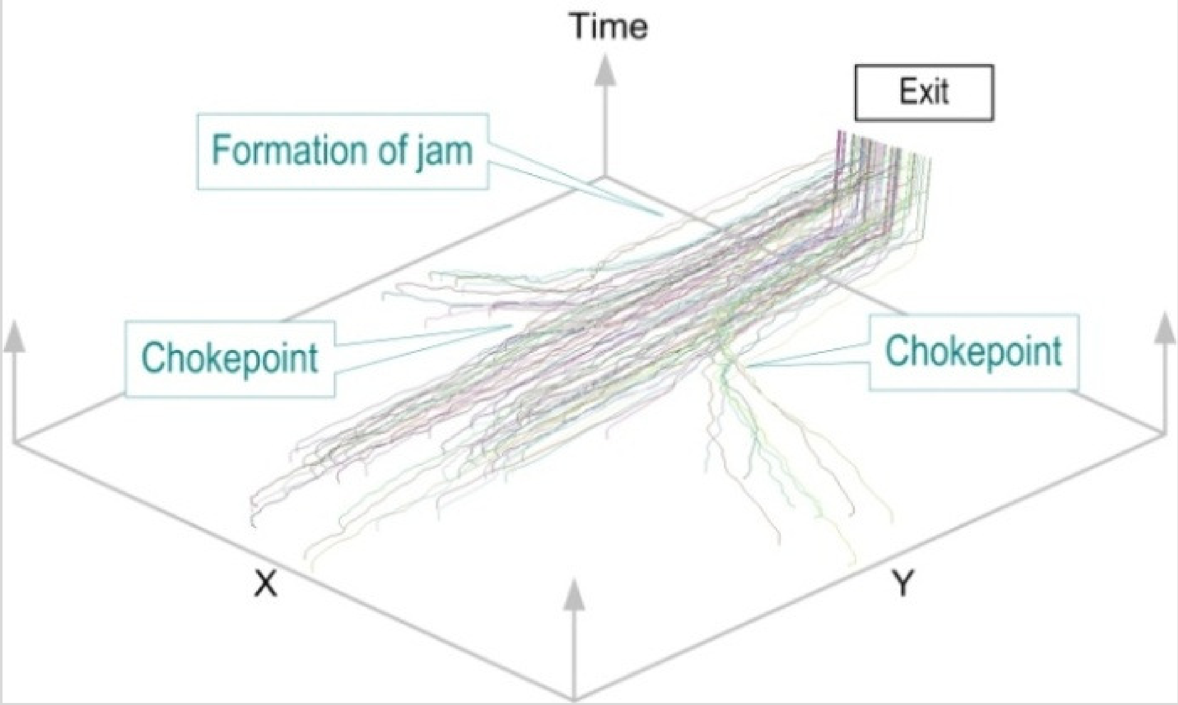
Limited degrees of freedom in mobility;

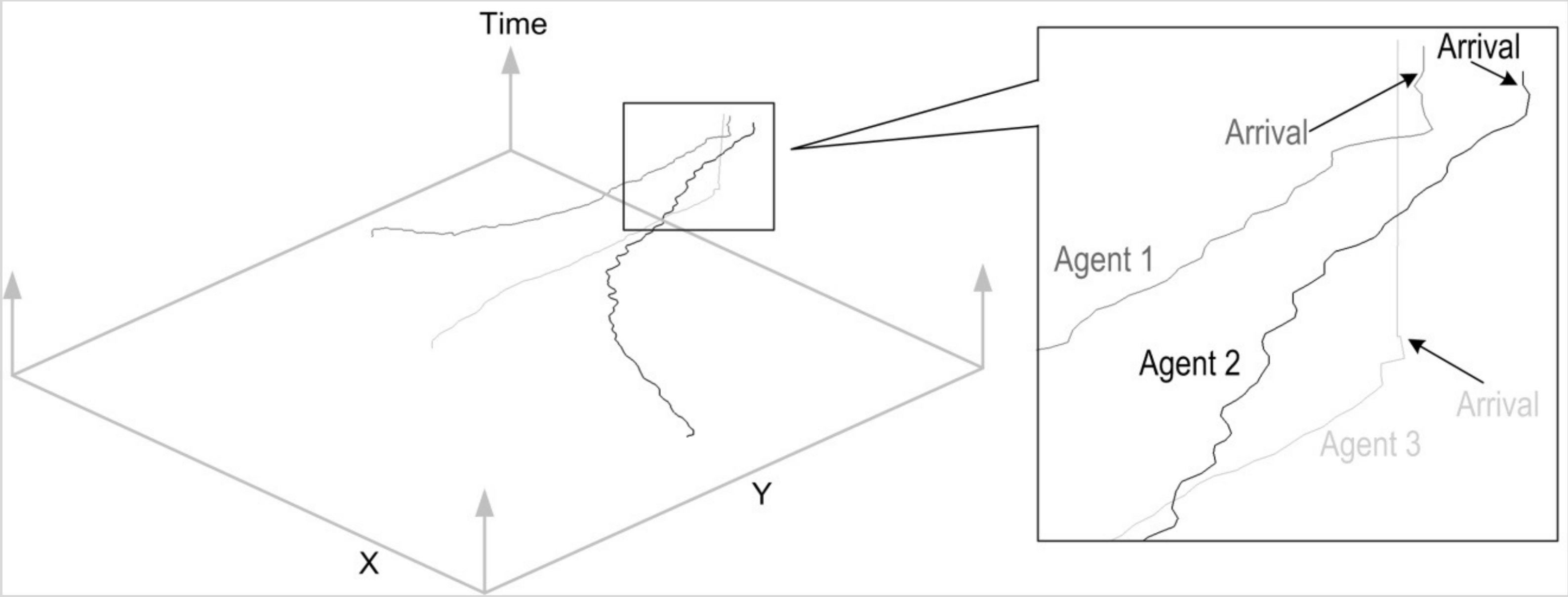
Limited geographic information due to crowding

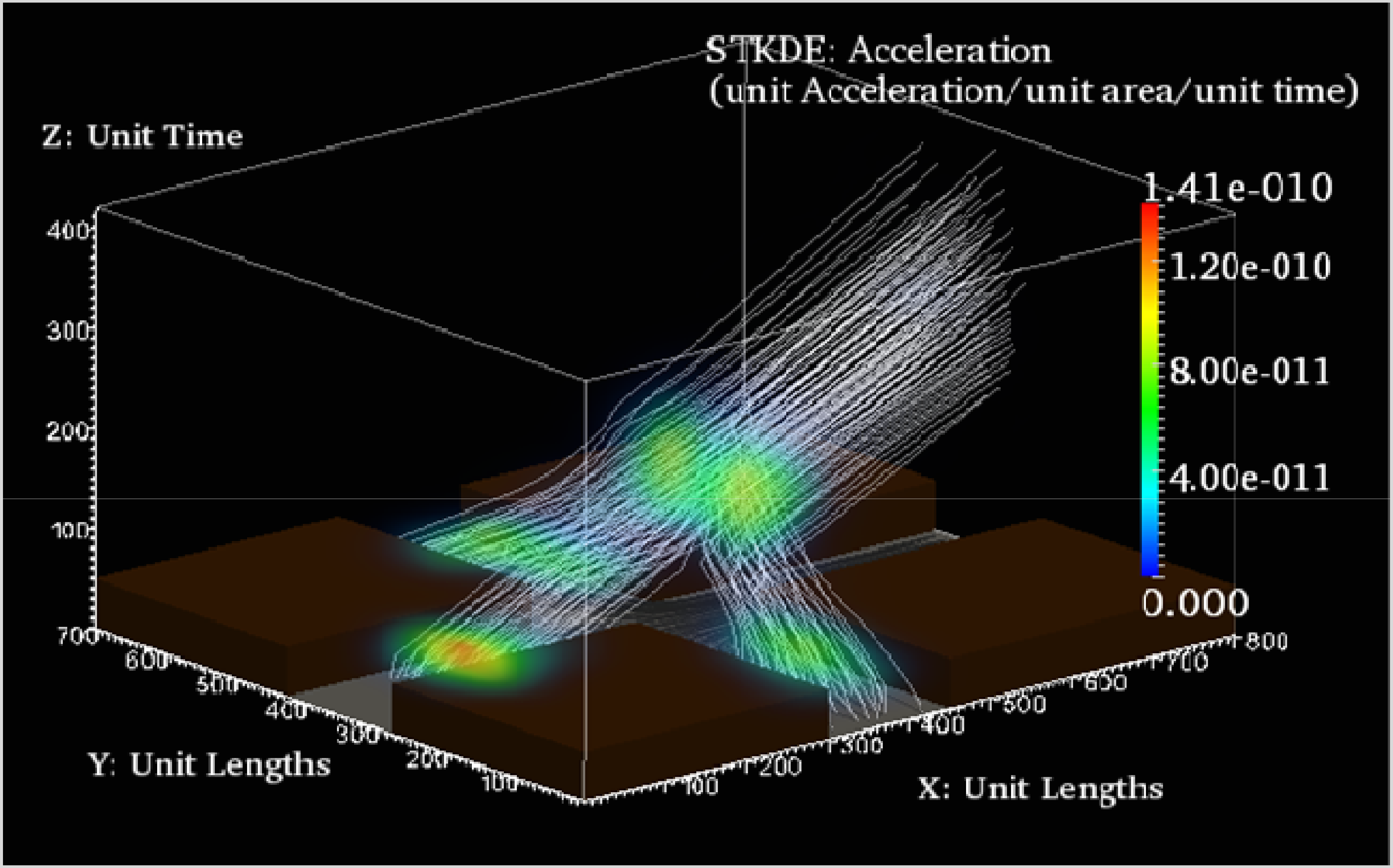

Time
X
Y
Space-time waves are apparent
An agent advances quickly, then slows down
Wall
Frequent turning
All other agent (spatial) paths
An individual agent space-time path



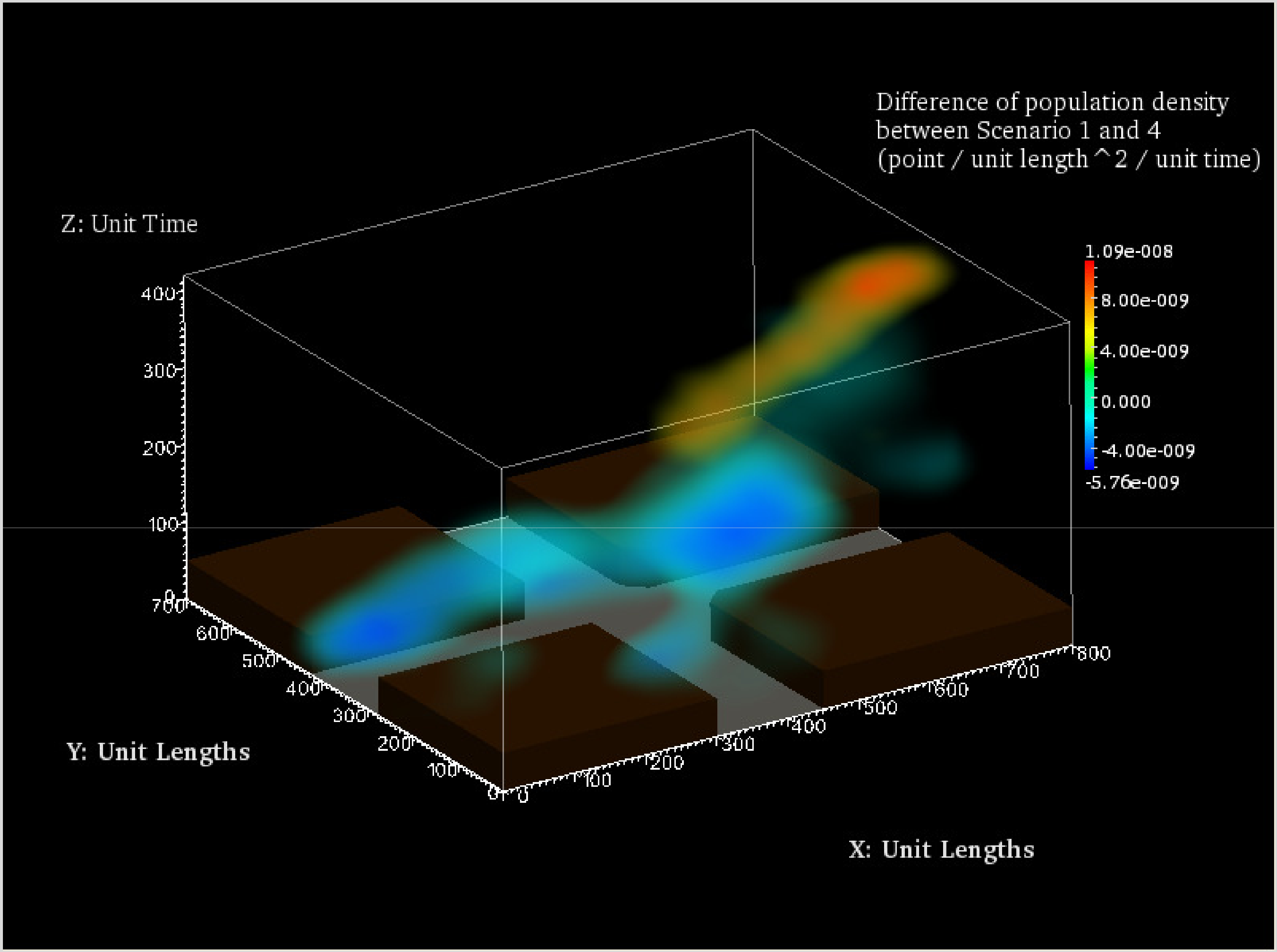
[Uh-oh]


Time
Exit
Formation of jam
Chokepoint
Chokepoint
X
Y


Time
Y
X
Arrival
Arrival
Agent 1
Agent 2
Arrival
Agent 3



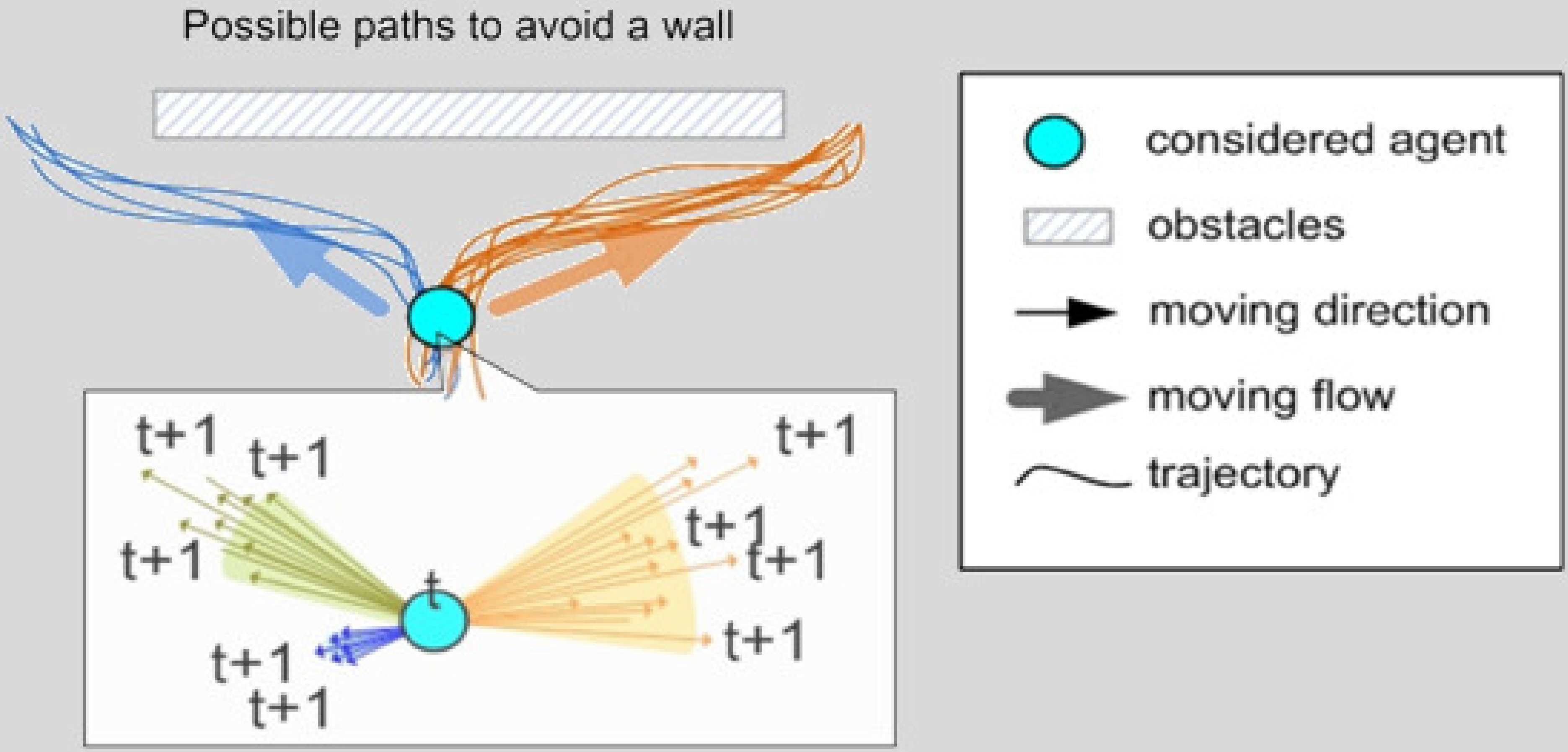
(Work with Atsushi Nara)



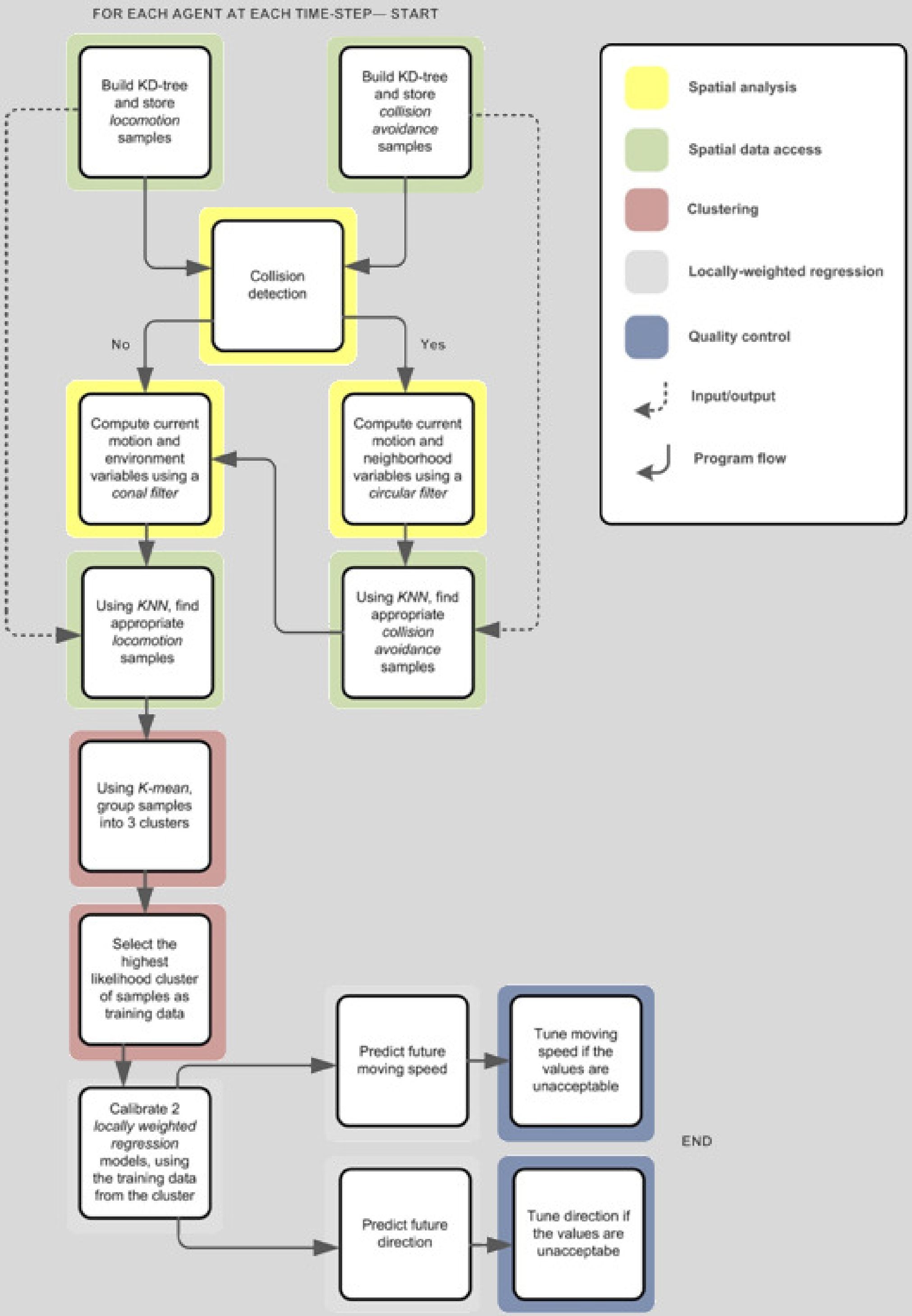
(Work with Atsushi Nara)

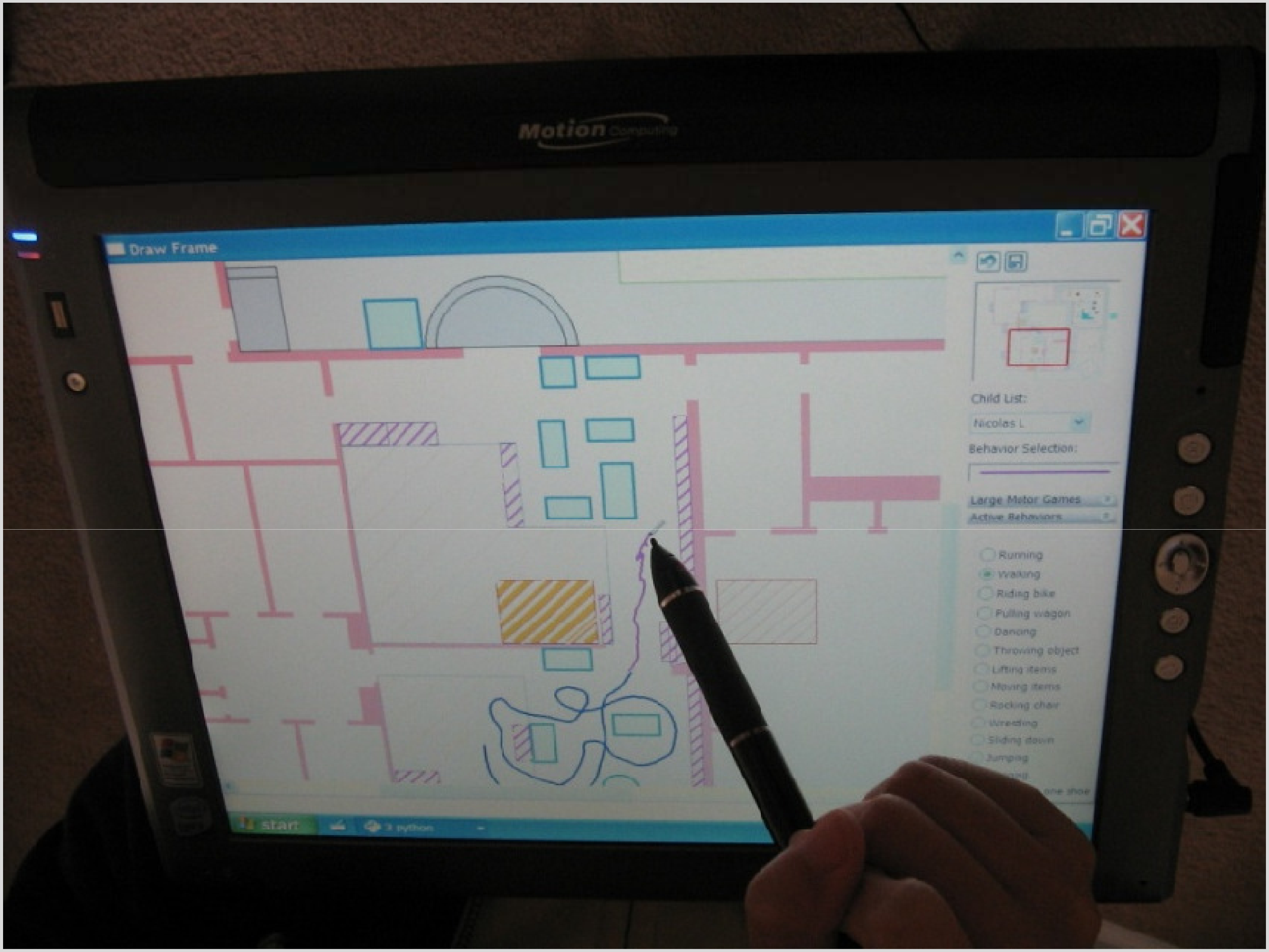
Can we automate construction of synthetic behavioral geography in simulation?

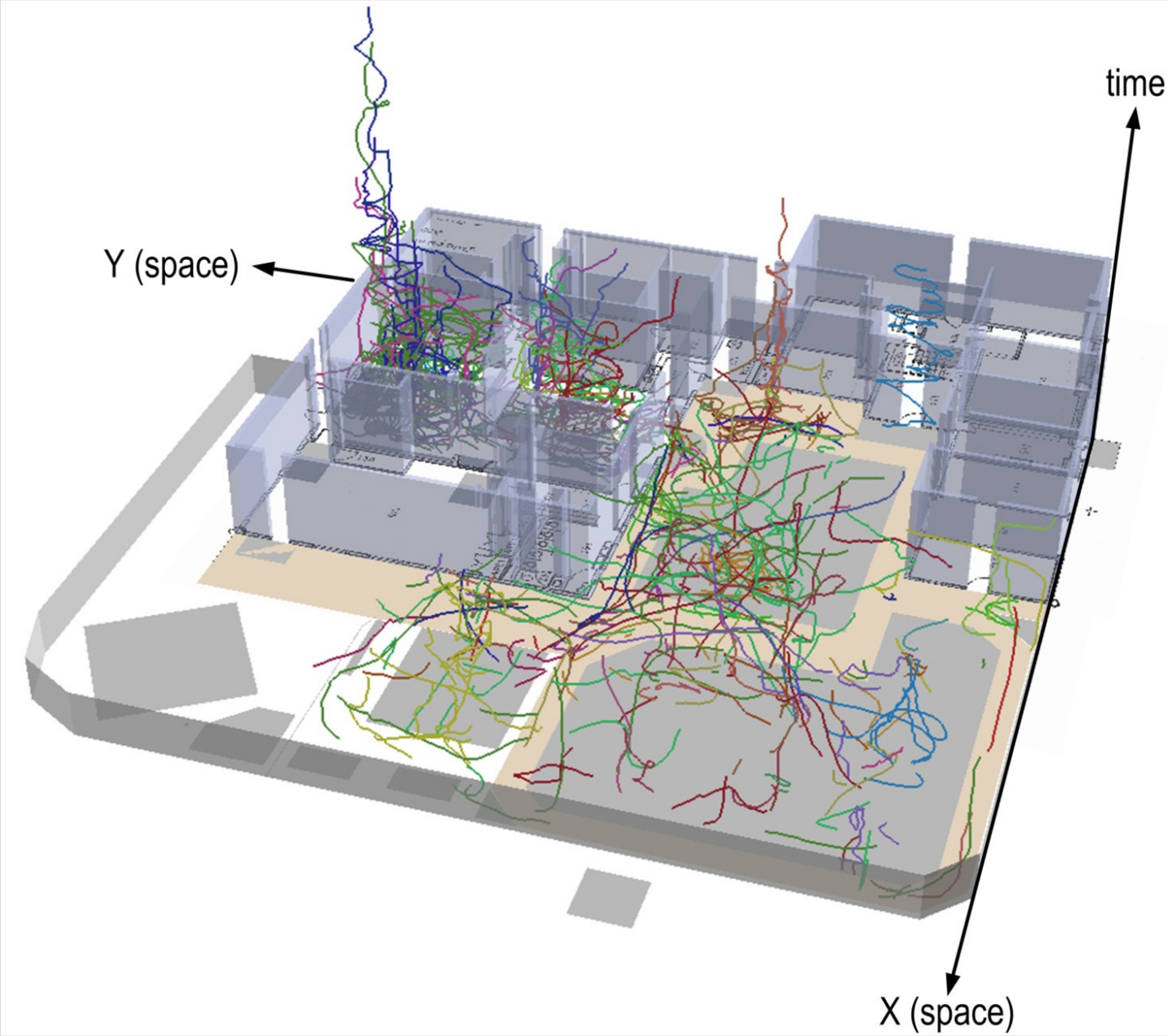
Machine-learning behavior



The three clusters of trajectory samples for that point in space and time

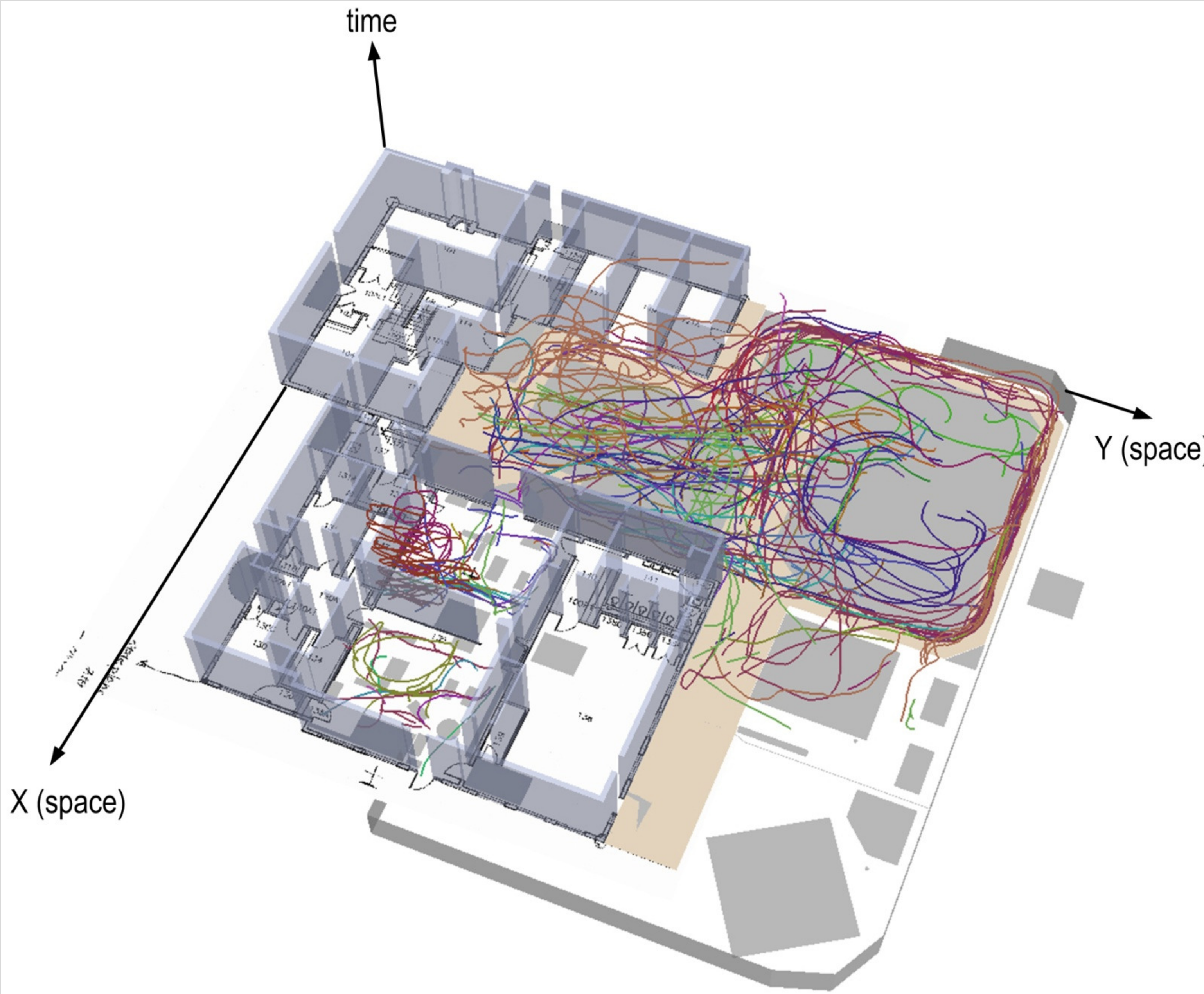

FOR EACH AGENT AT EACH TIME-STEP— START
Build KD-tree and store *locomotion* samples
Build KD-tree and store *collision avoidance* samples
Collision detection
No
Yes
Compute current motion and environment variables using a *conal filter*
Compute current motion and neighborhood variables using a *circular filter*
Using *KNN*, find appropriate *locomotion* samples
Using *KNN*, find appropriate *collision avoidance* samples
Using *K-mean*, group samples into 3 clusters
Select the highest likelihood cluster of samples as training data
Calibrate 2 *locally weighted regression* models, using the training data from the cluster
Predict future moving speed
Tune moving speed if the values are unacceptable
Predict future direction
Tune direction if the values are unacceptabe
END
Spatial analysis
Spatial data access
Clustering
Locally-weighted regression
Quality control
Input/output
Program flow


Motion
Draw Frame
Child List:
Nicolas L
Behavior Selection:
Large Motor Games
Active Behaviors
Running
Walking
Riding bike
Pulling wagon
Dancing
Throwing object
Lifting items
Moving items
Rocking chair
Sliding down
Jumping
start



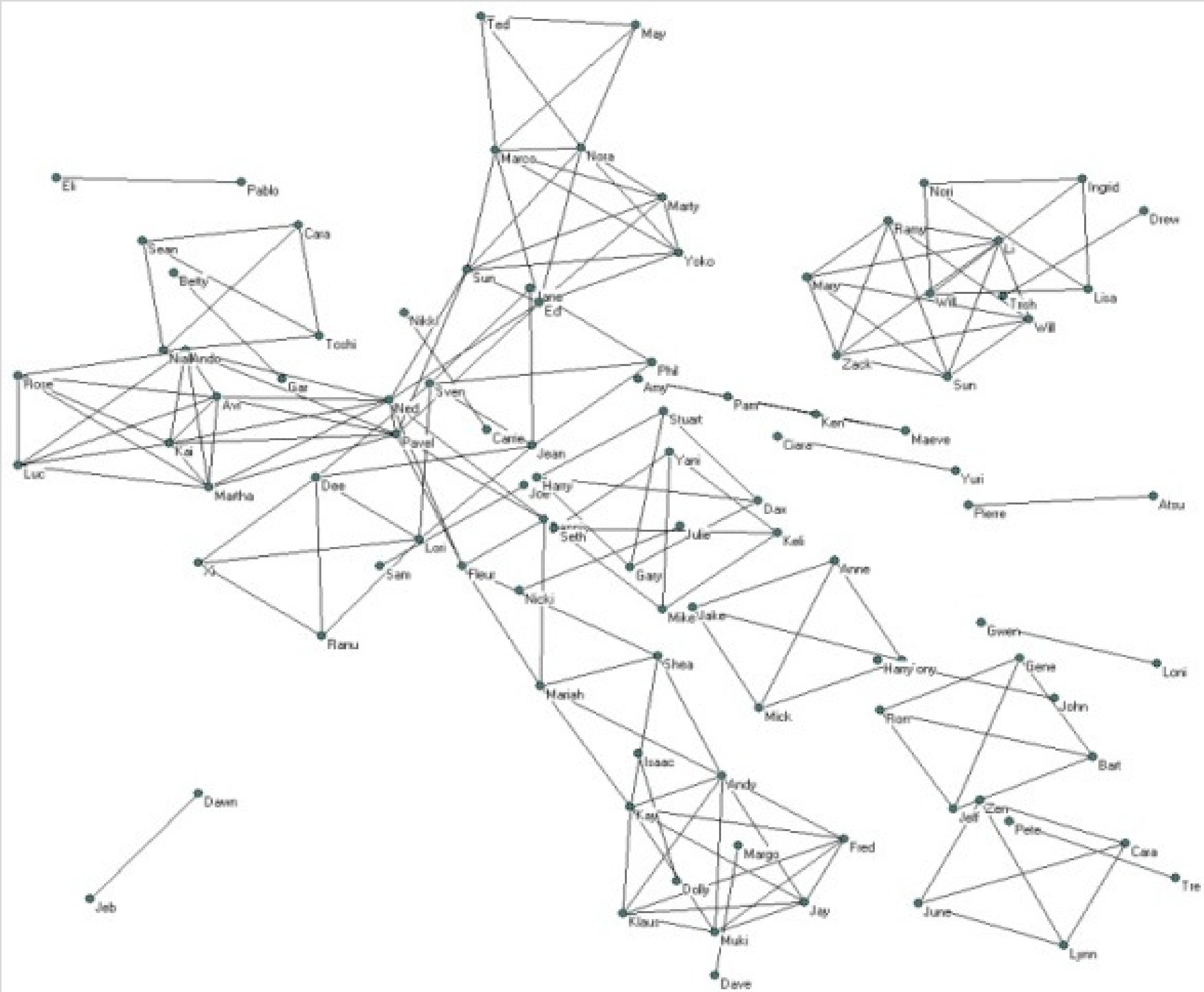
Observed behaviors

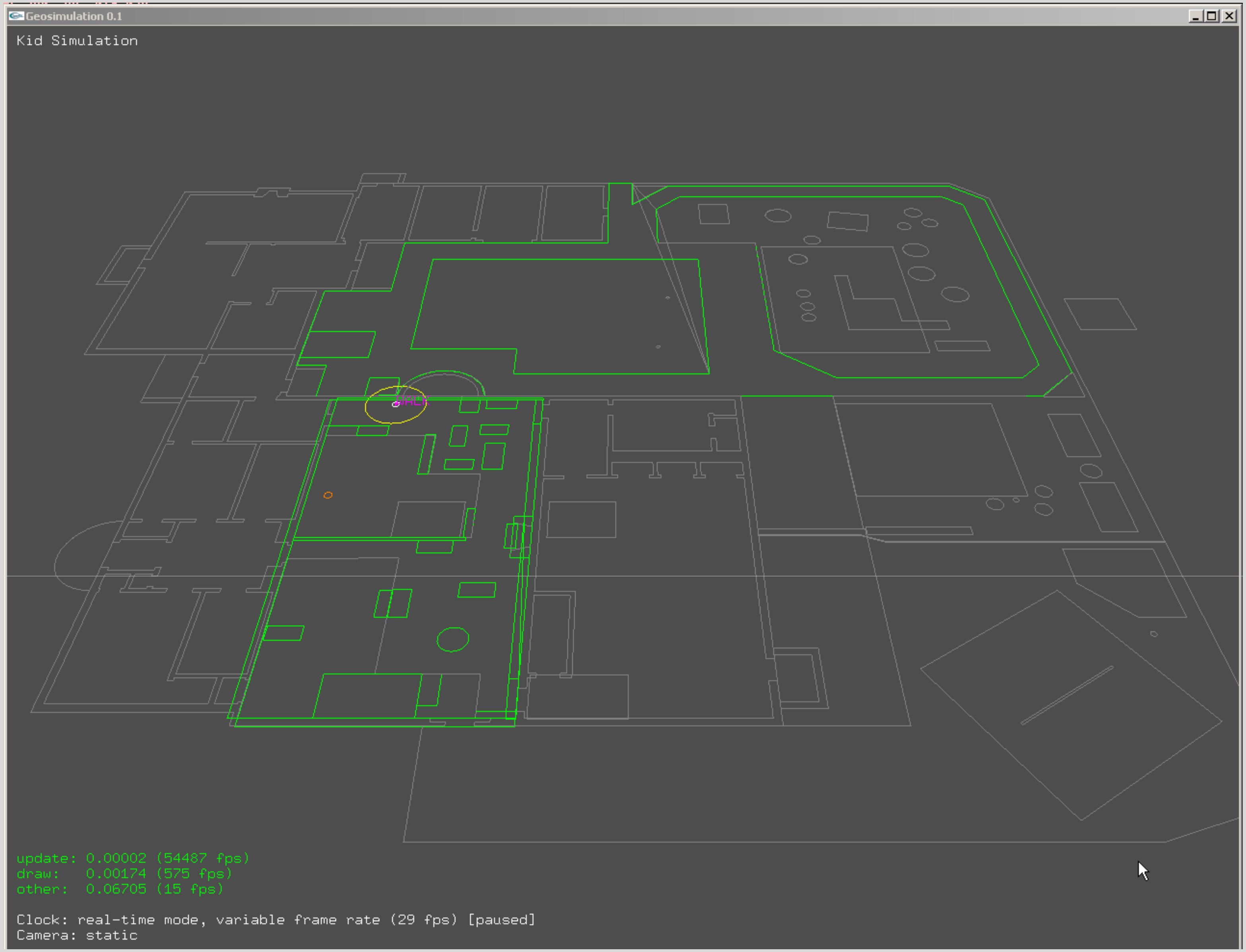
Walking



Observed behaviors

Running

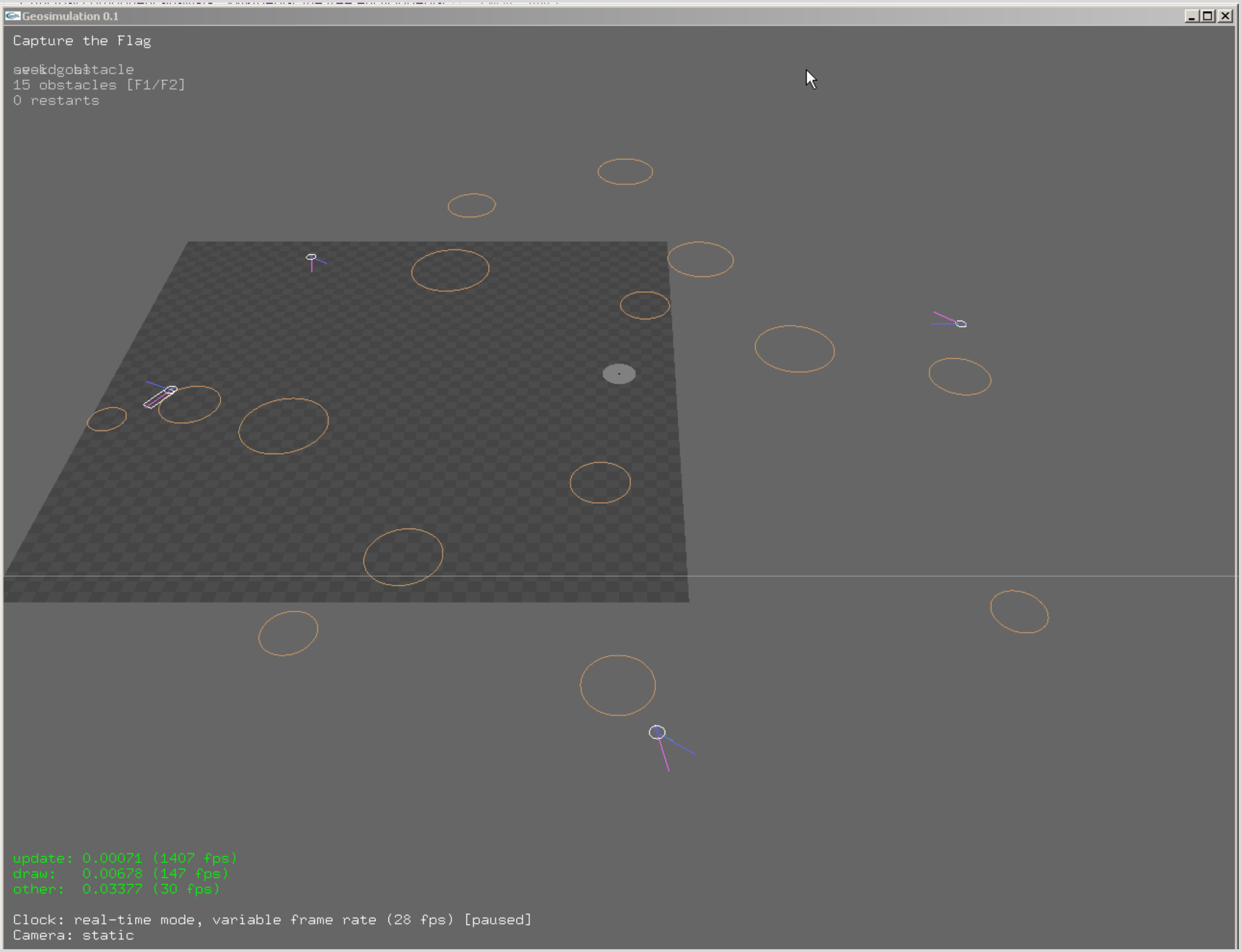

Ted
May
Marco
Nora
Marty
Yoko
Eli
Pablo
Sean
Cara
Betty
Toshi
Sun
Jane
Ed
Nikki
Phil
Amy
Pam
Ken
Maeve
Ciara
Yuri
Pierre
Atsu
Rose
Gar
Avi
Kai
Luc
Martha
Sven
Ned
Pavel
Carrie
Jean
Stuart
Yani
Harry
Joe
Dee
Dax
Kelli
Seth
Julie
Lori
Sam
Fleur
Gary
Nicki
Mike
Jake
Anne
Ranu
Shea
Mariah
Mick
Gwen
Loni
Gene
John
Ron
Bart
Isaac
Andy
Kay
Margo
Fred
Dolly
Jay
Klaus
Muki
Dave
Dawn
Jeb
Zen
Jeff
Pete
Cara
Tre
June
Lynn
Nori
Ingrid
Drew
Ramy
Li
Mary
Will
Tosh
Lisa
Zack
Sun



Model / link

Learning to walk with real trajectory data

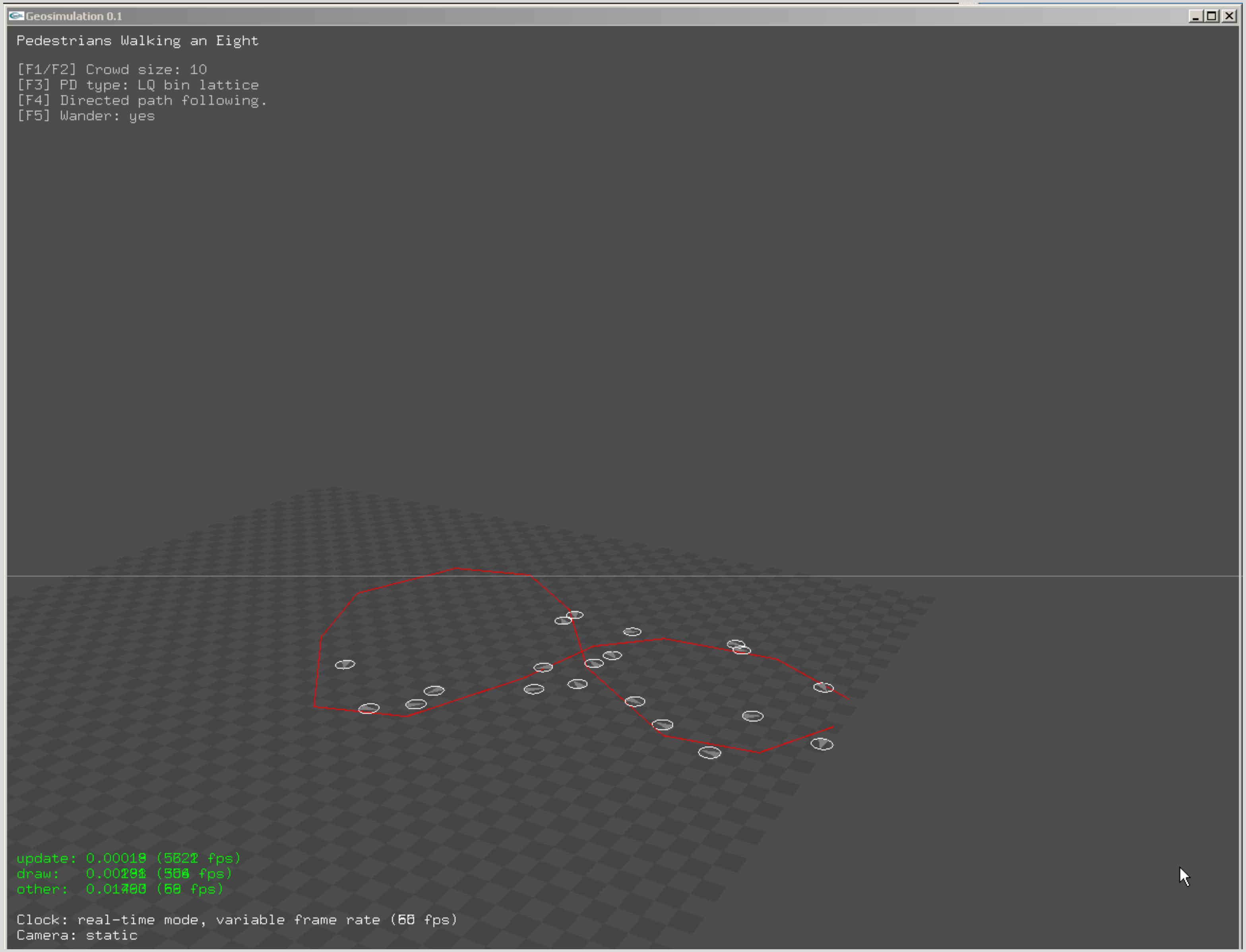
(Work with Xun Li)



[Uh-oh]

Learning to walk with simulated data: obstacle avoidance

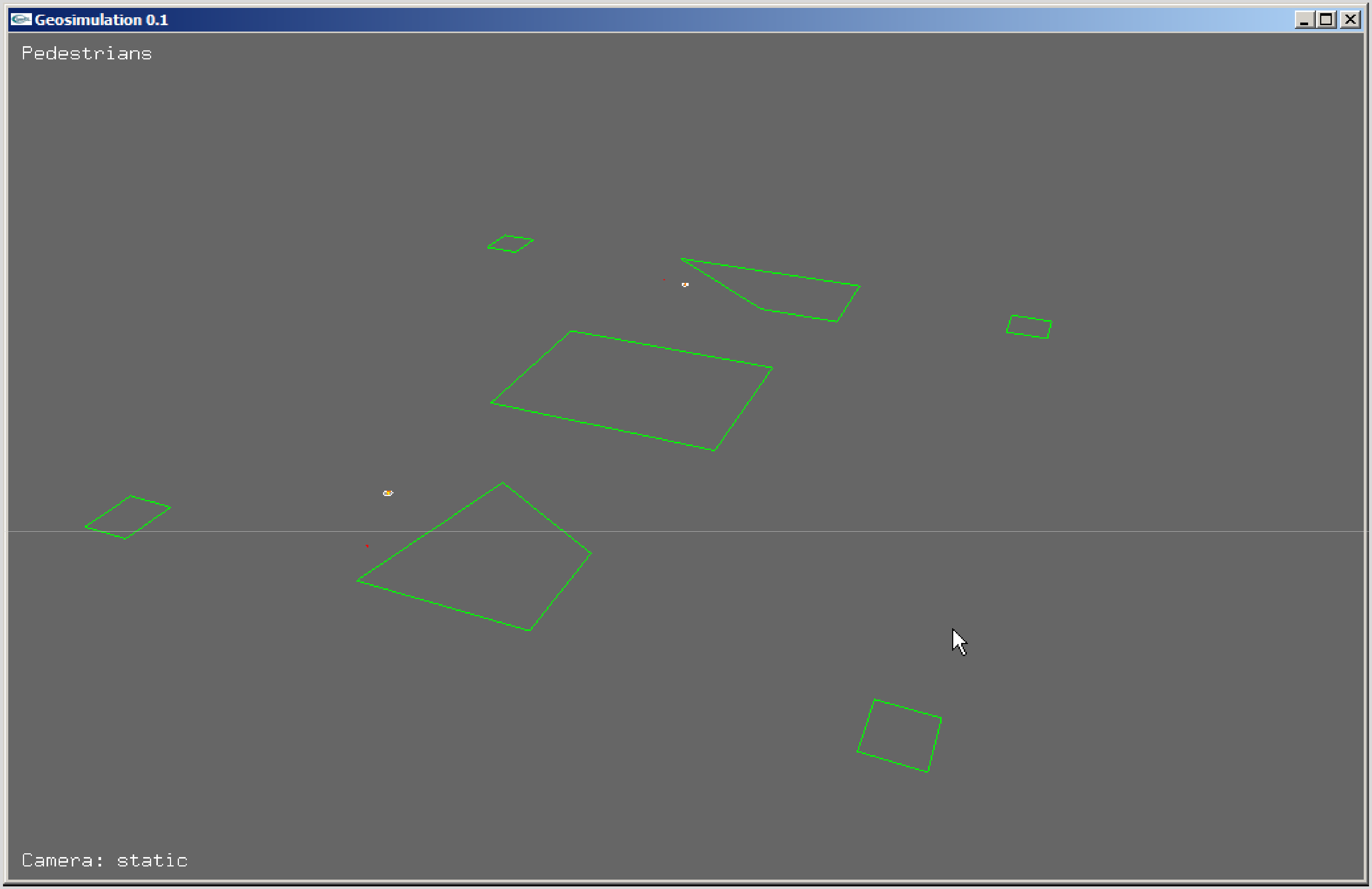
(Work with Xun Li)



Uh-oh

Learning to walk with simulated data: people avoidance

(Work with Xun Li)



Model

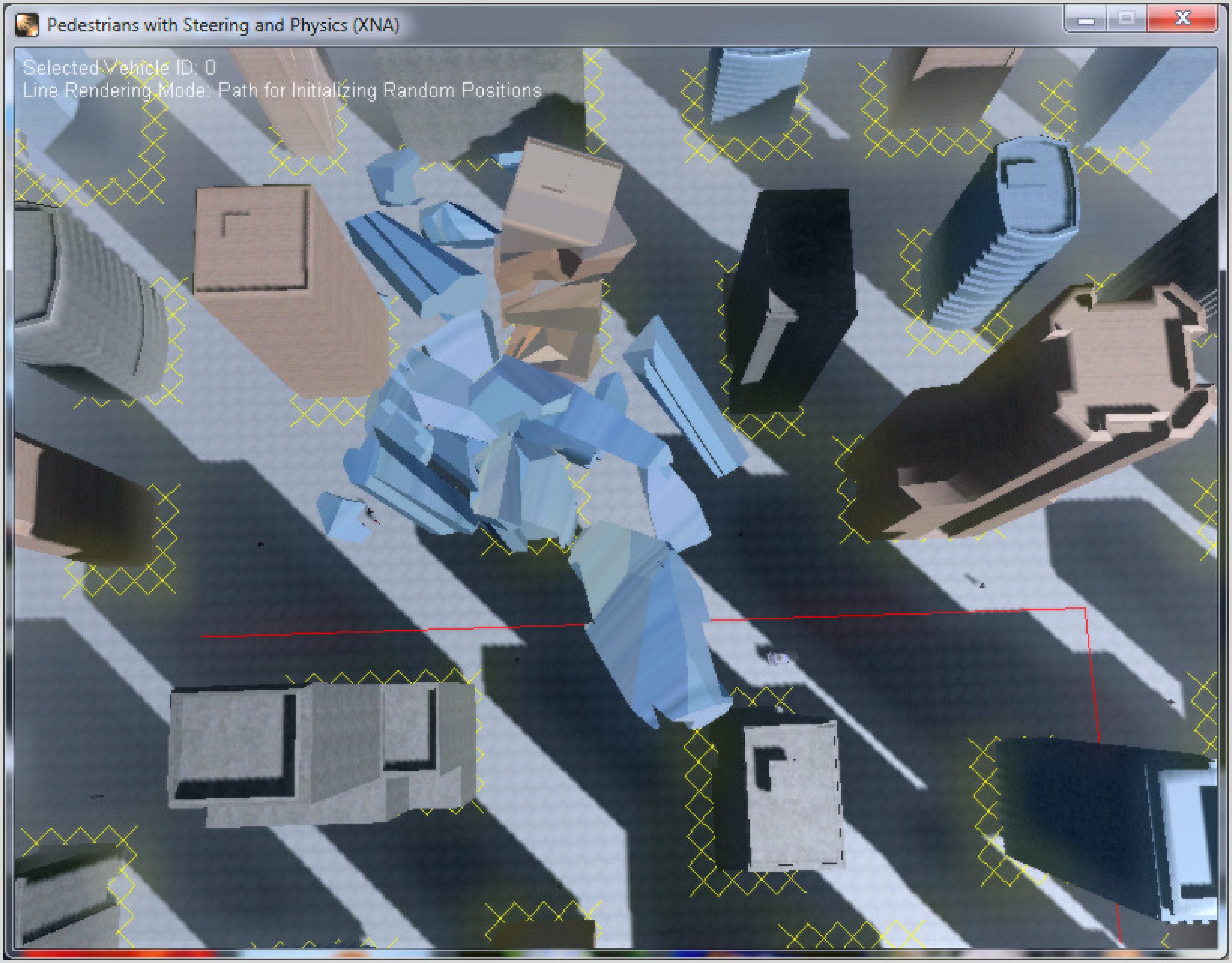
Learning to walk with a combined data library

(Work with Xun Li)

Next steps

STEM learning on the Xbox

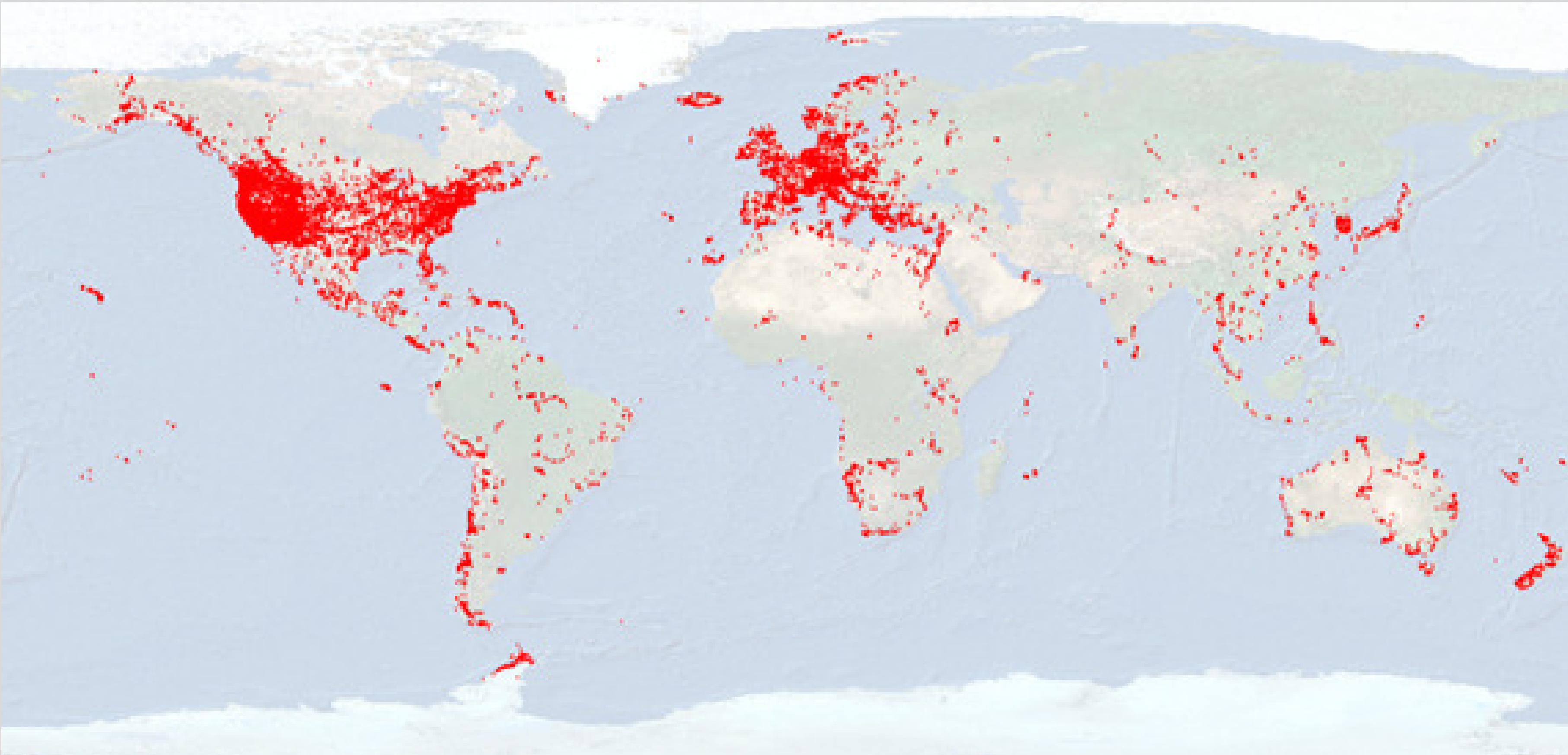
(Work with Haojie Zhu)



model

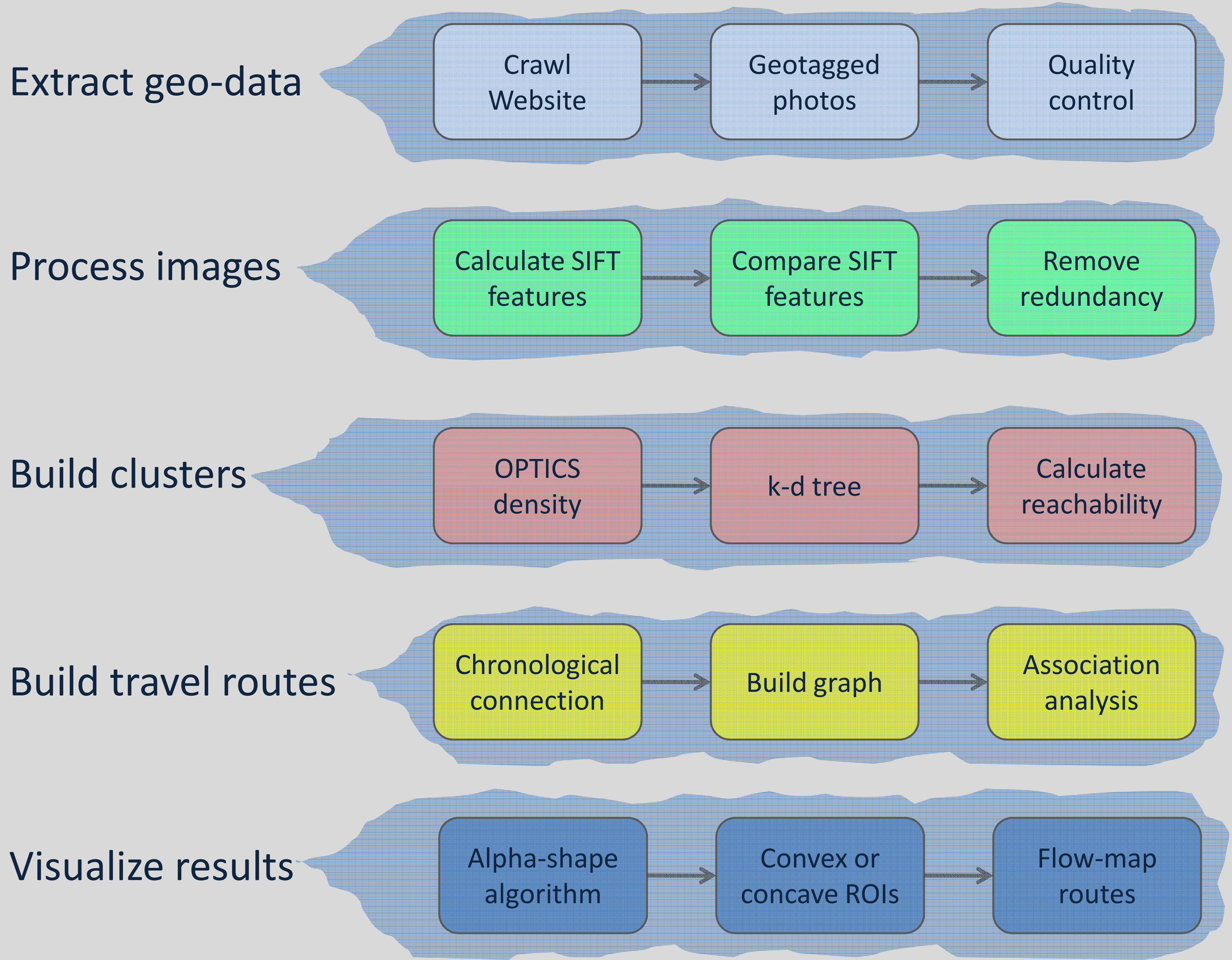
(Work with Haojie Zhu, Joshua Epstein, Tak Igusa)

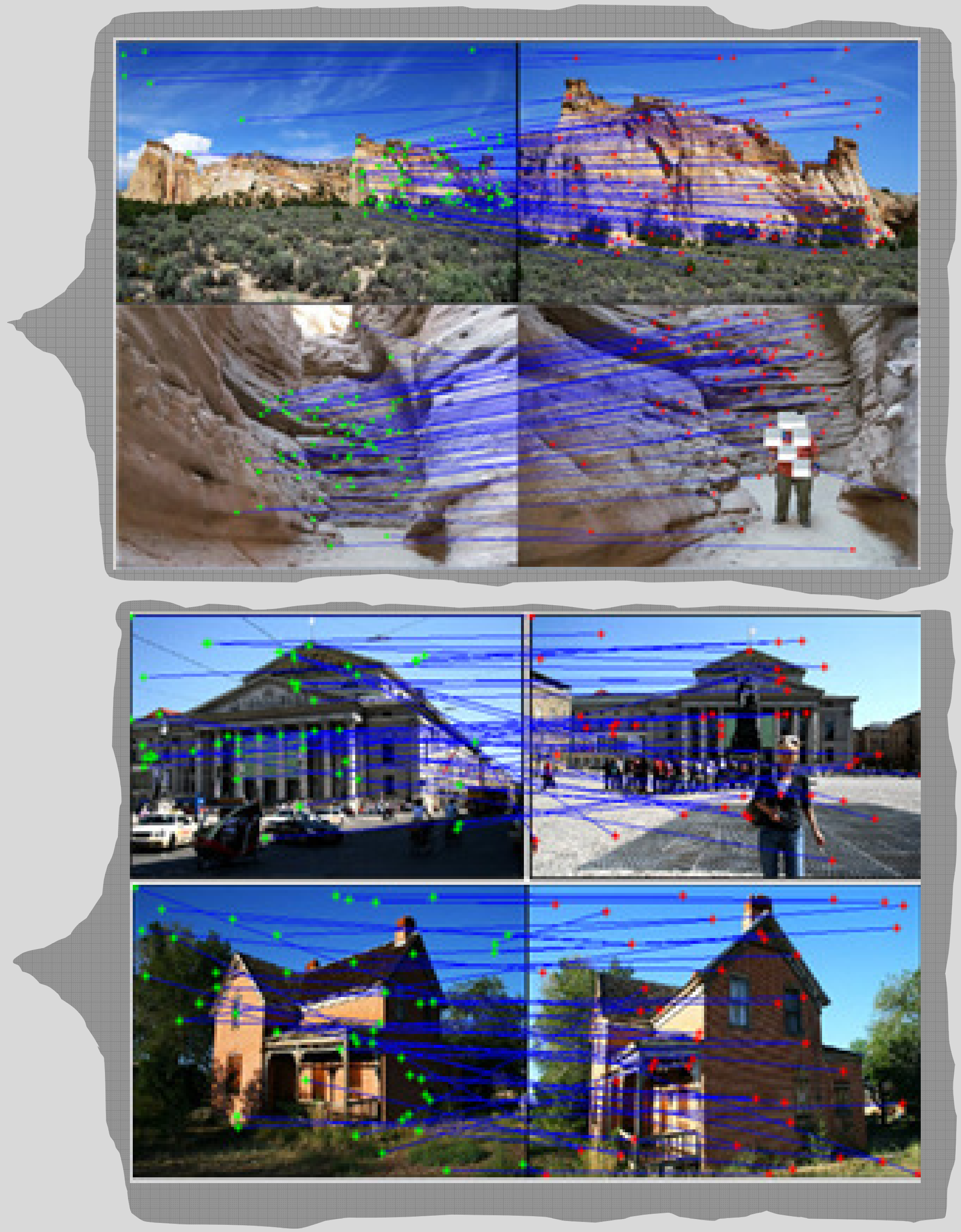
Machine-learning behavioral geography from Online silos of trajectory data



Distribution of photographs on *panoramio.com* (*n*=20 million)

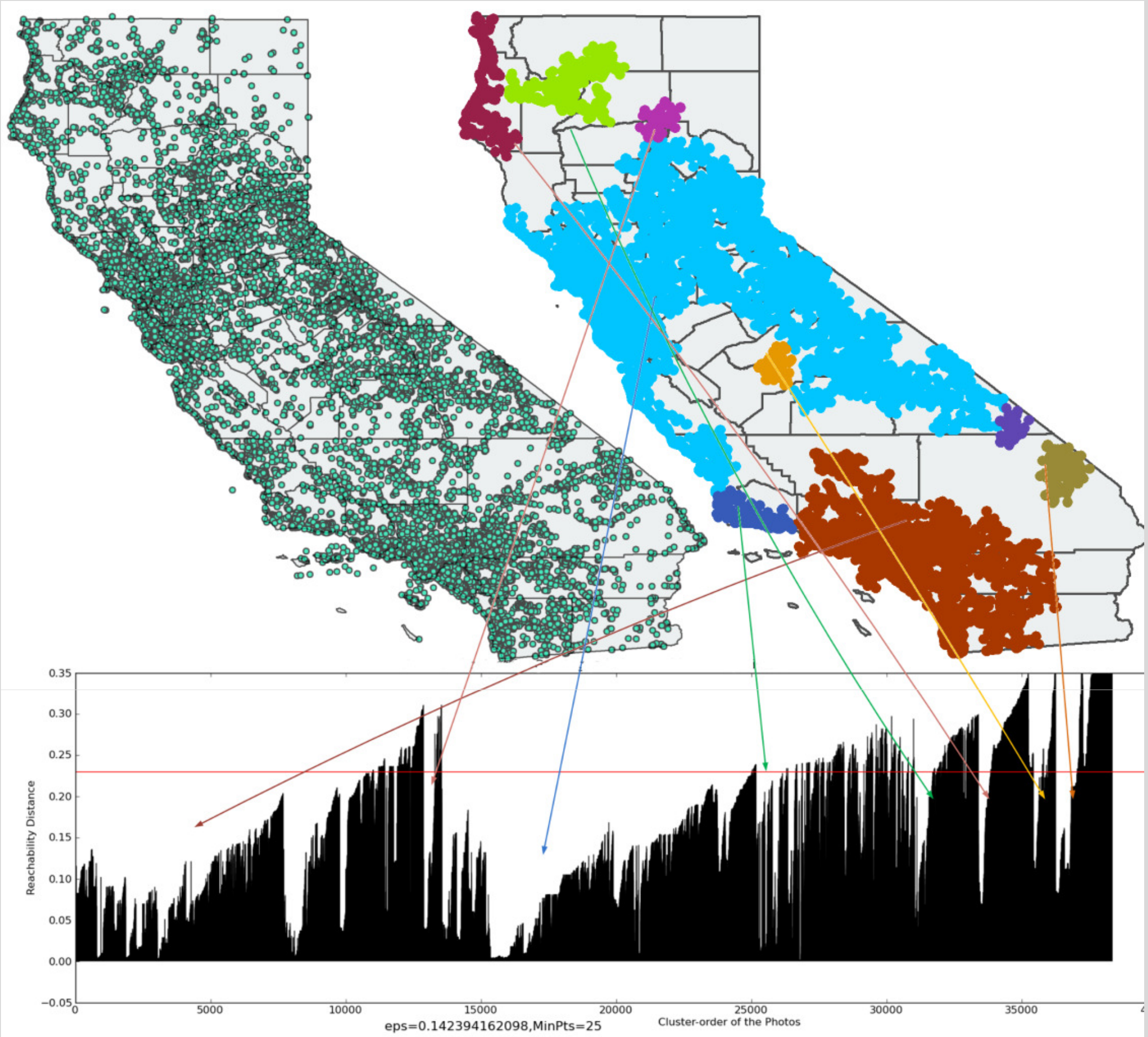
(Work with Xun Li)


Extract geo-data
Crawl Website
Geotagged photos
Quality control
Process images
Calculate SIFT features
Compare SIFT features
Remove redundancy
Build clusters
OPTICS density
k-d tree
Calculate reachability
Build travel routes
Chronological connection
Build graph
Association analysis
Visualize results
Alpha-shape algorithm
Convex or concave ROIs
Flow-map routes



Scale-invariant feature transform (SIFT) to remove redundant images

(Work with Xun Li)



Reachability Plot of OPTICS on Geo-tagged Photos

(Work with Xun Li)

158,753 users with trips to CA

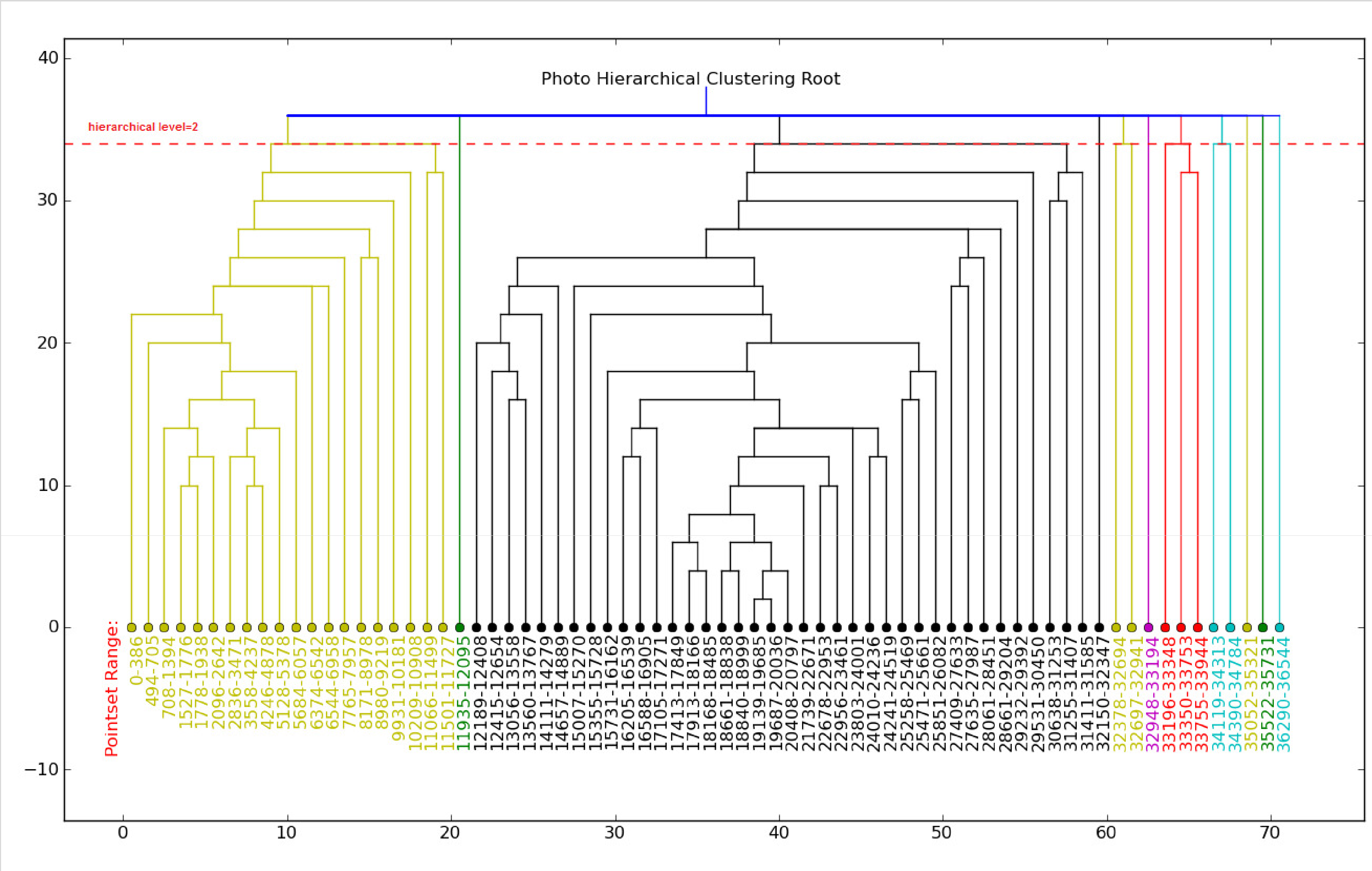
44,764 photographs in CA

Removed 672 duplicates

44,092 photographs remaining

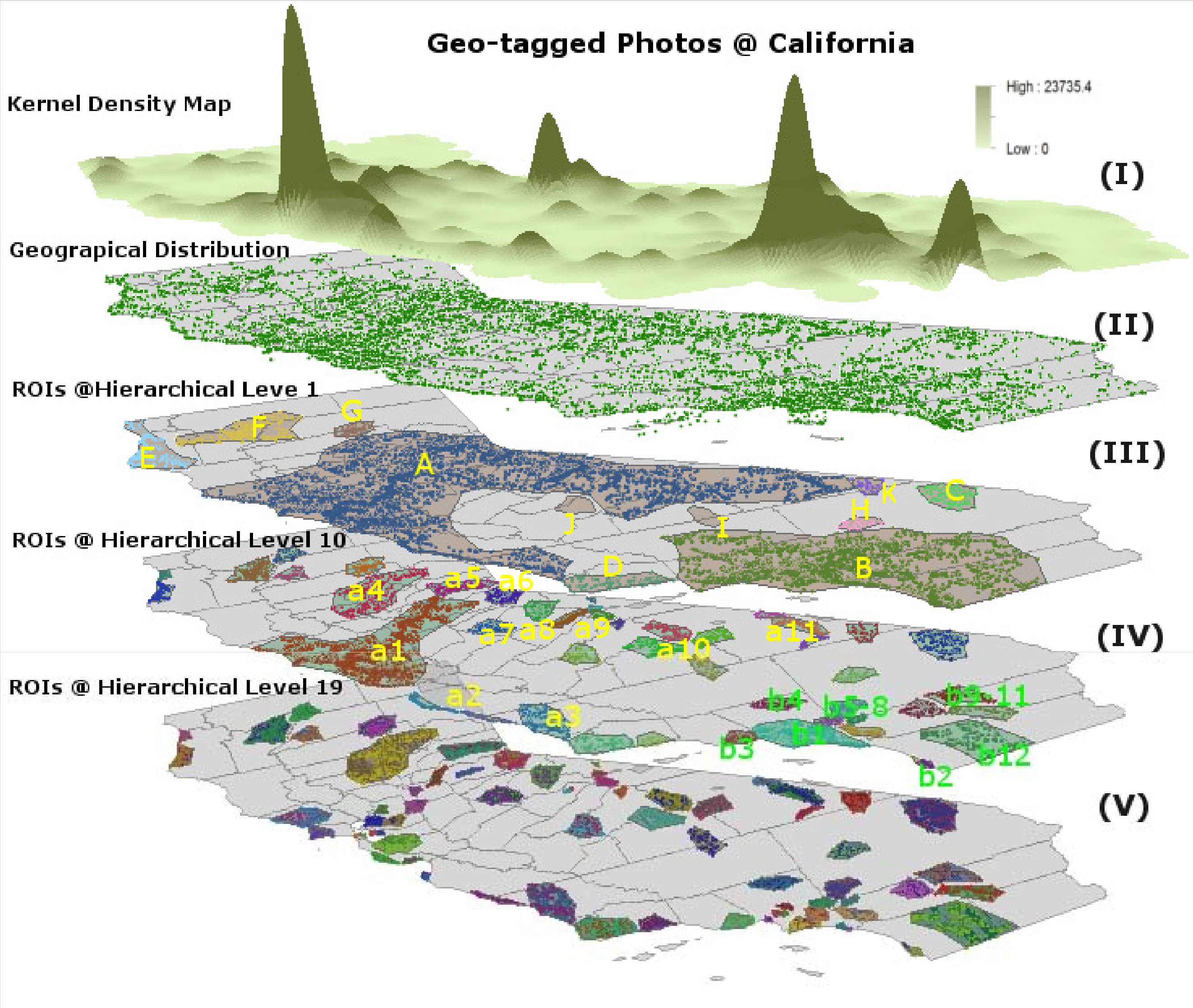
Ordering points to identify the clustering structure (OPTICS)

Upper: LHS photo locations; RHS colored by cluster; Lower: reachability plot



Hierarchical dendrogram (leaves are smallest regions of interest)

(Work with Xun Li)



Regions of interest @ hierarchy 1

A Northern California

B Southern California

C Mojave National Preserve

D Santa Barbara

E Redwood National Park

F Trinity National Forest

G Lassen National Park

H Calico Ghost Town

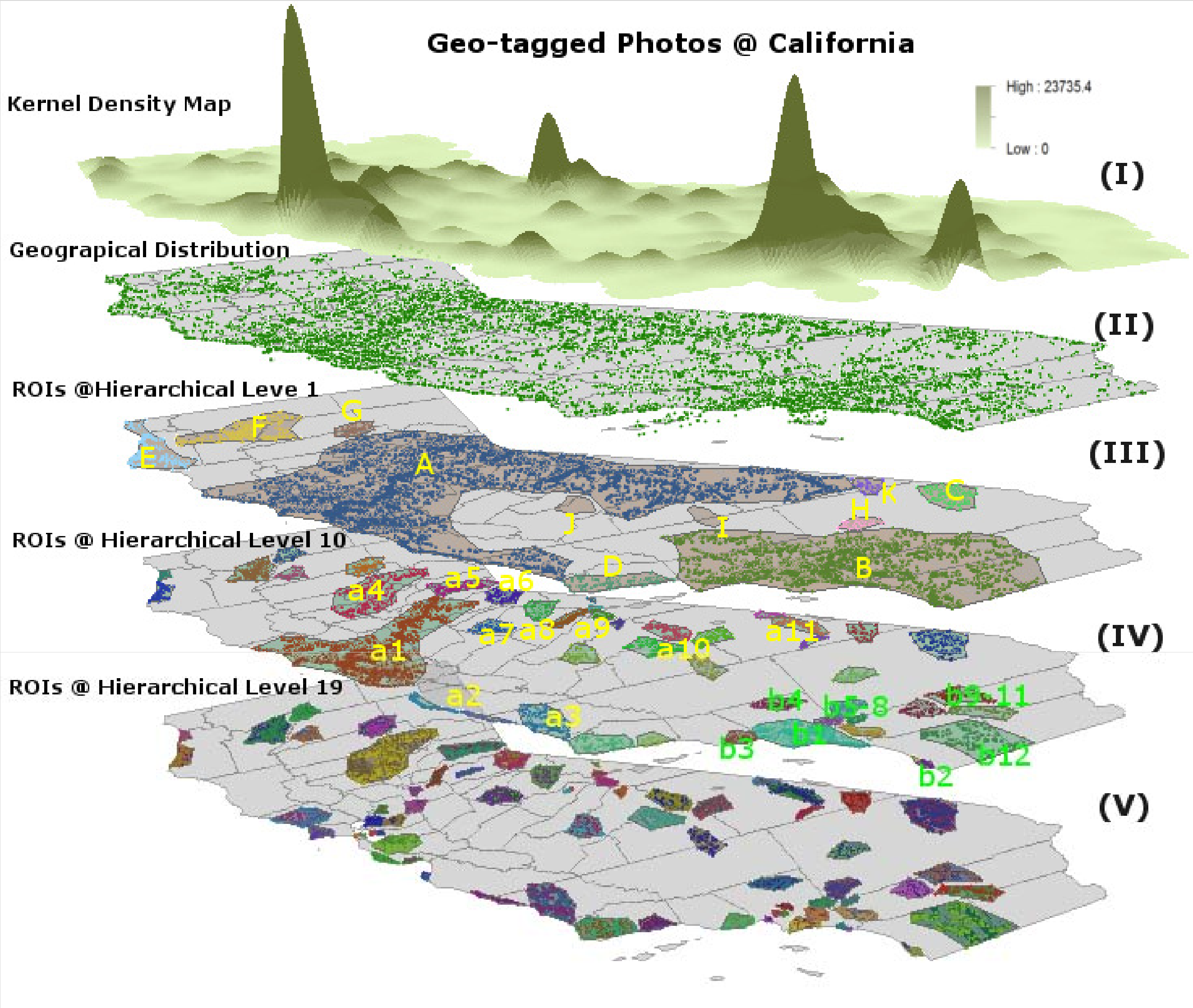
I Sequoia National Forest

J Fresno

K Mojave National Park

Learned regions of interest at different scales

(Work with Xun Li)



Regions of interest @ hierarchy 10

A1 San Francisco

A2 Highway 1

A3 San Luis Obispo

A4 Plumas National Forest

A5 Tahoe National Forest

A6 Lake Tahoe

A7 New Meloes Lake

A8 Stanislaus National Forest

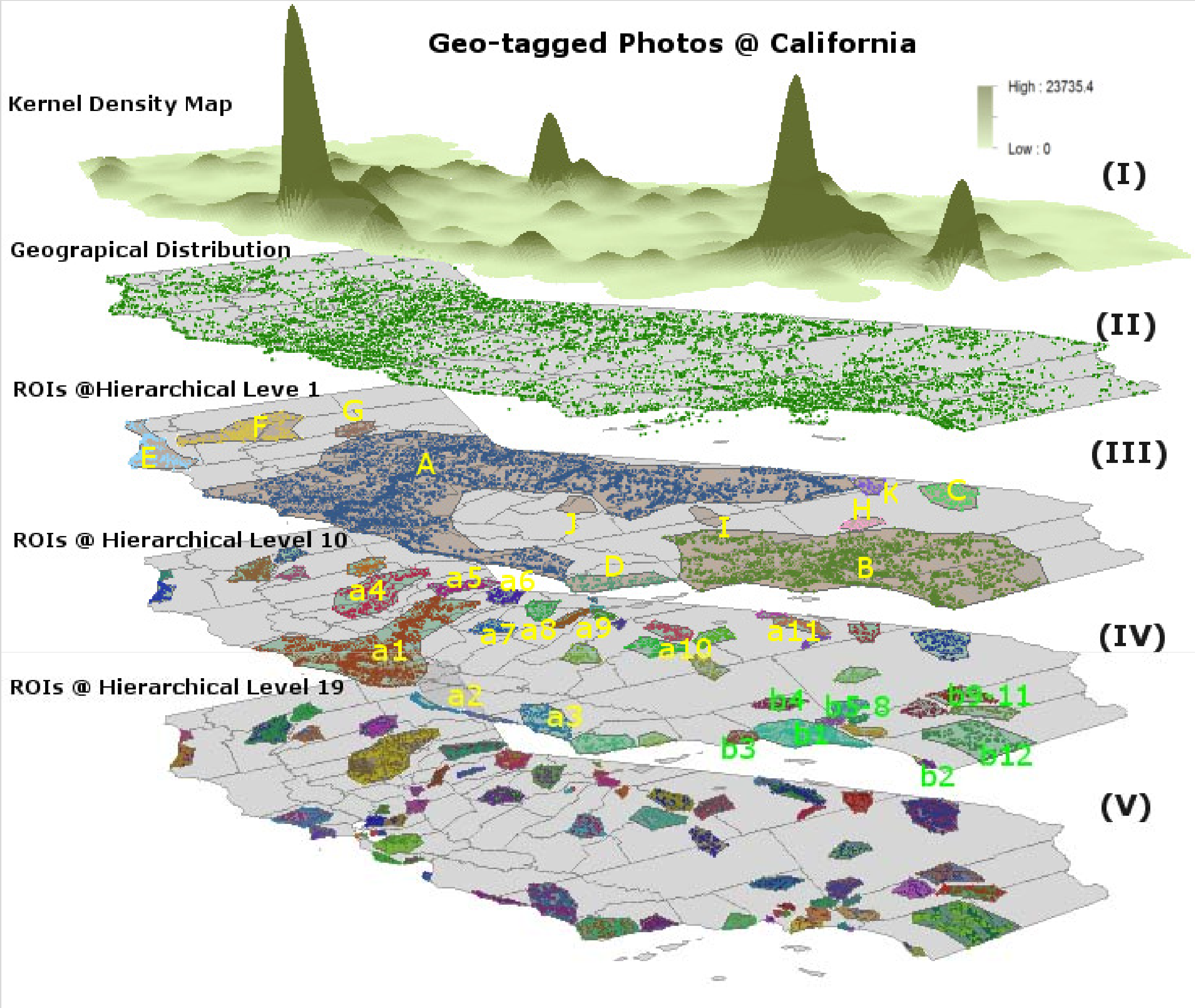
A9 Yosemite National Park

A10 Sequoia National Park

A11 Death Valley National Park

Learned regions of interest at different scales

(Work with Xun Li)



Regions of interest @ hierarchy 10

B1 Los Angeles

B2 San Diego

B3 Malibu

B4 Palmdale

B5 Ontario

B6 San Bernardino National Forest

B7 Riverside

B8 Corona

B9 Palm Desert

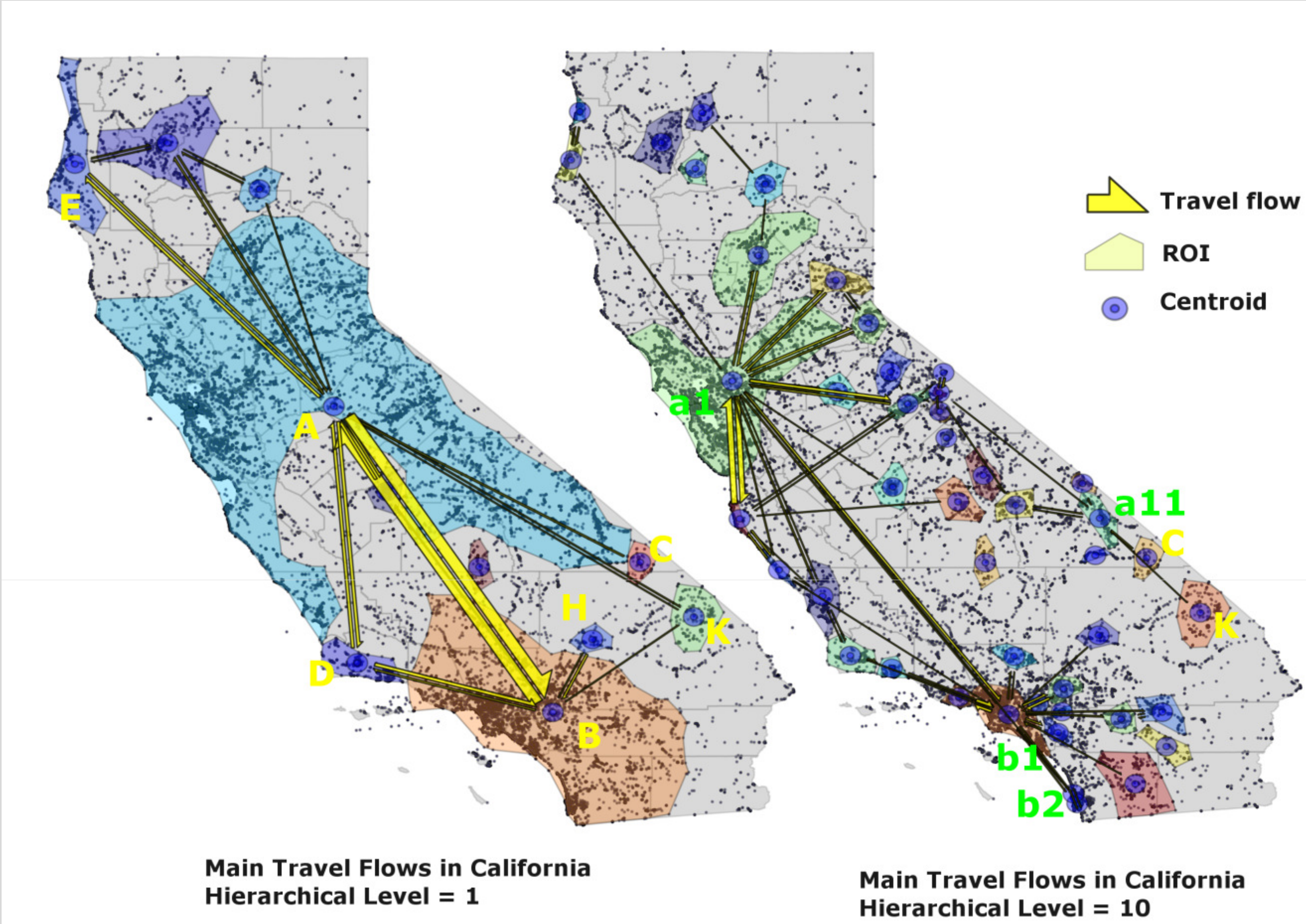
B10 Joshua Tree National Park

B11 Salton Sea

B12 Anza-Borrego Desert State Pk.

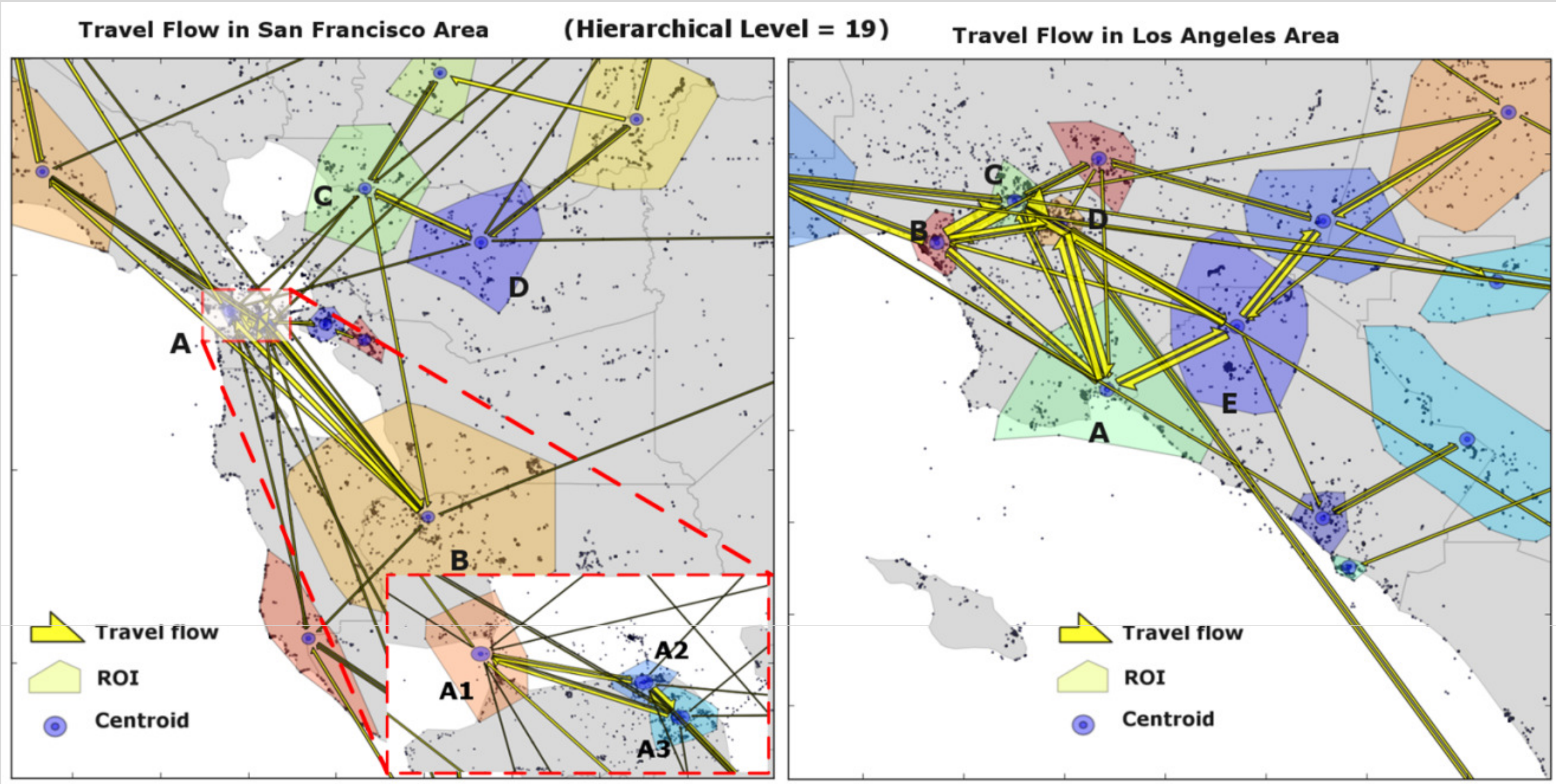
Learned regions of interest at different scales

(Work with Xun Li)



Learned travel flows for California at different scales

(Work with Xun Li; Tobler flow mapping)



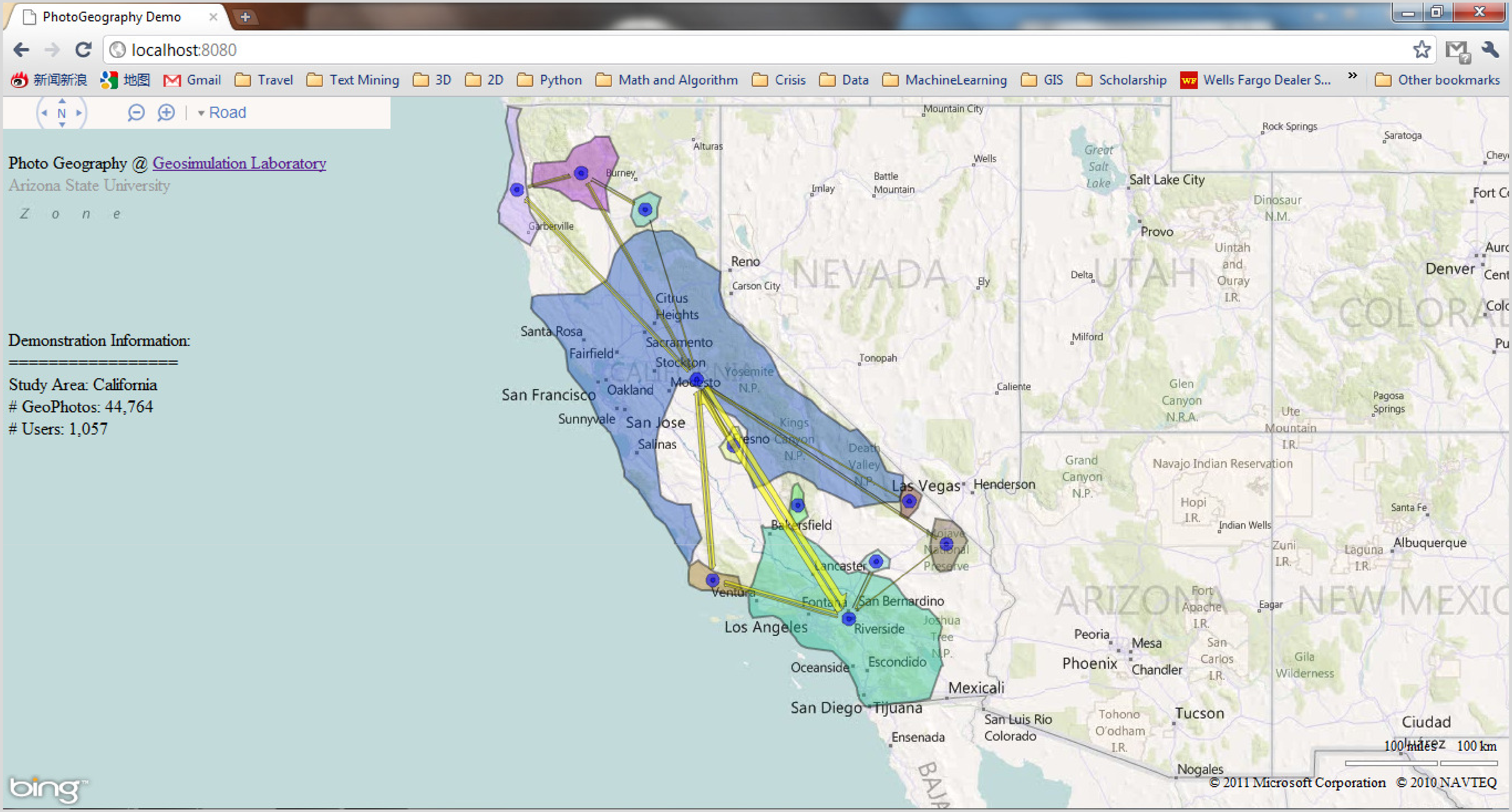
SF: A1 Golden Gate bridge; A2 Fisherman's Wharf; A3 Mission; B Palo Alto; C Vallejo; D Mt. Diablo

LA: A Long Beach; B Santa Monica; C Hollywood; D downtown; E Anaheim

Learned travel flows for San Francisco and Los Angeles (2 day limit)

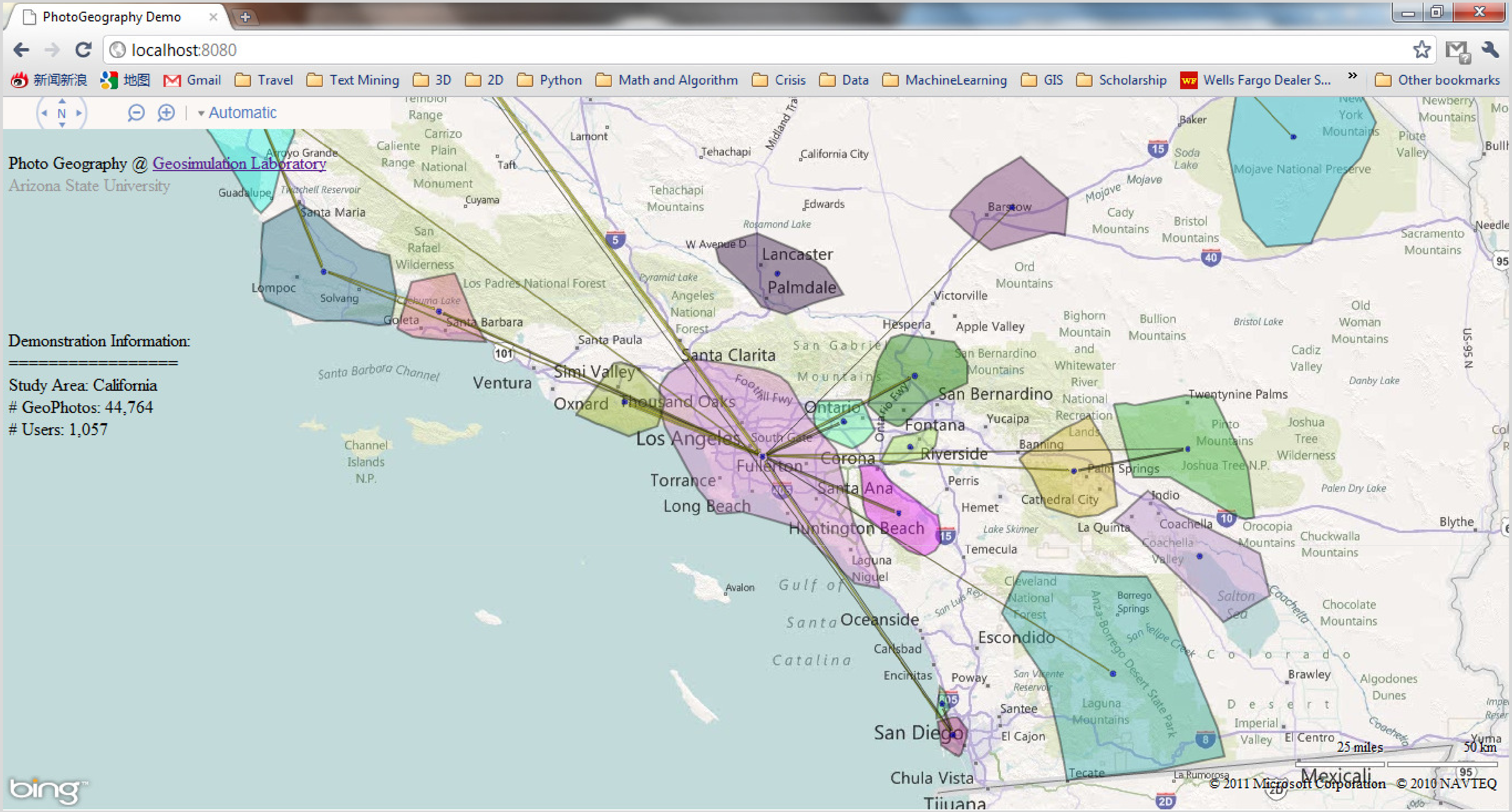
(Work with Xun Li)

Server-side mining and visualization with *Bing Maps* zoom-scaling



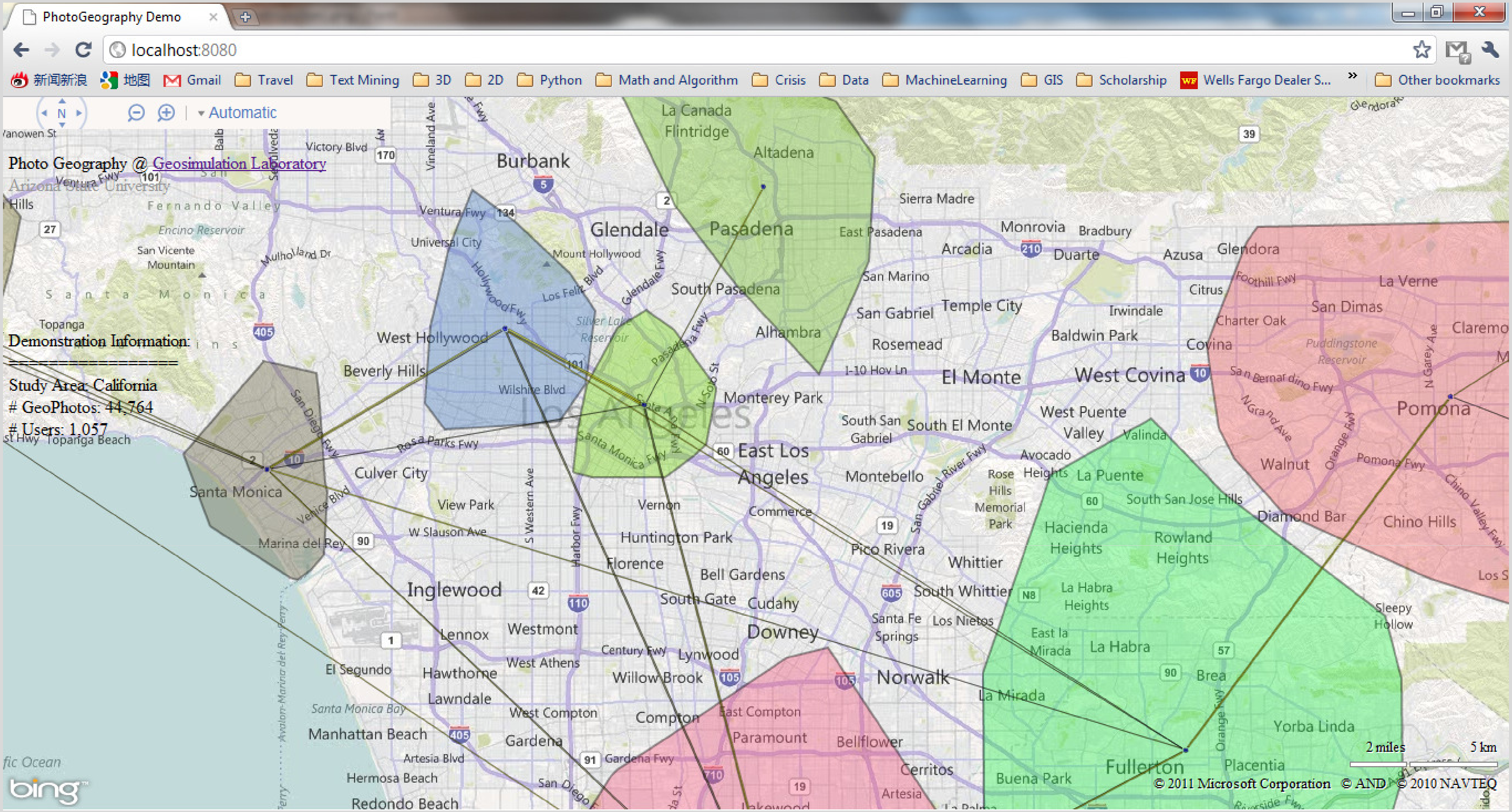
California, zoom-level 1 (state)

(Work with Xun Li)



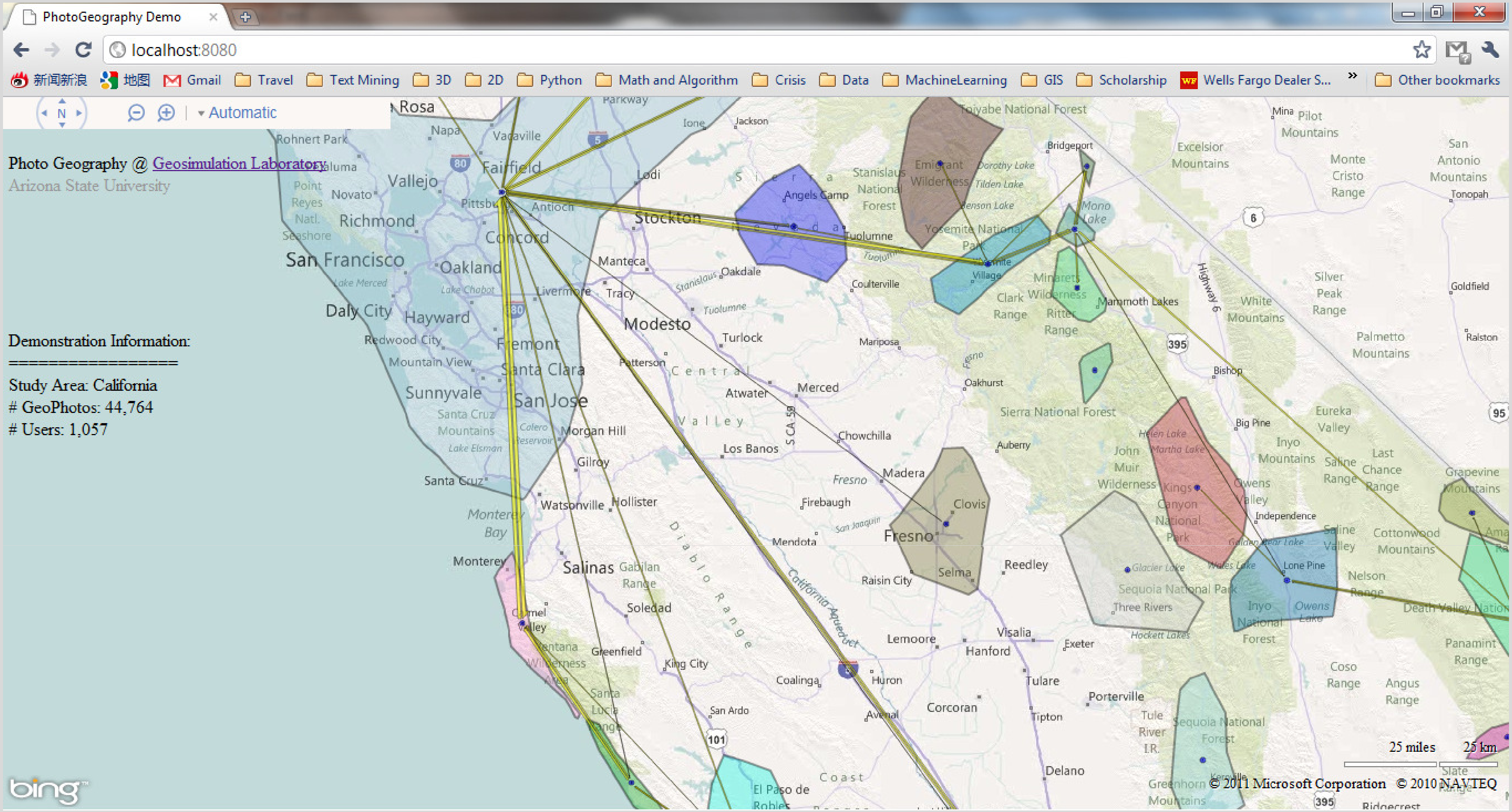
Los Angeles, zoom-level 9 (county)

(Work with Xun Li)



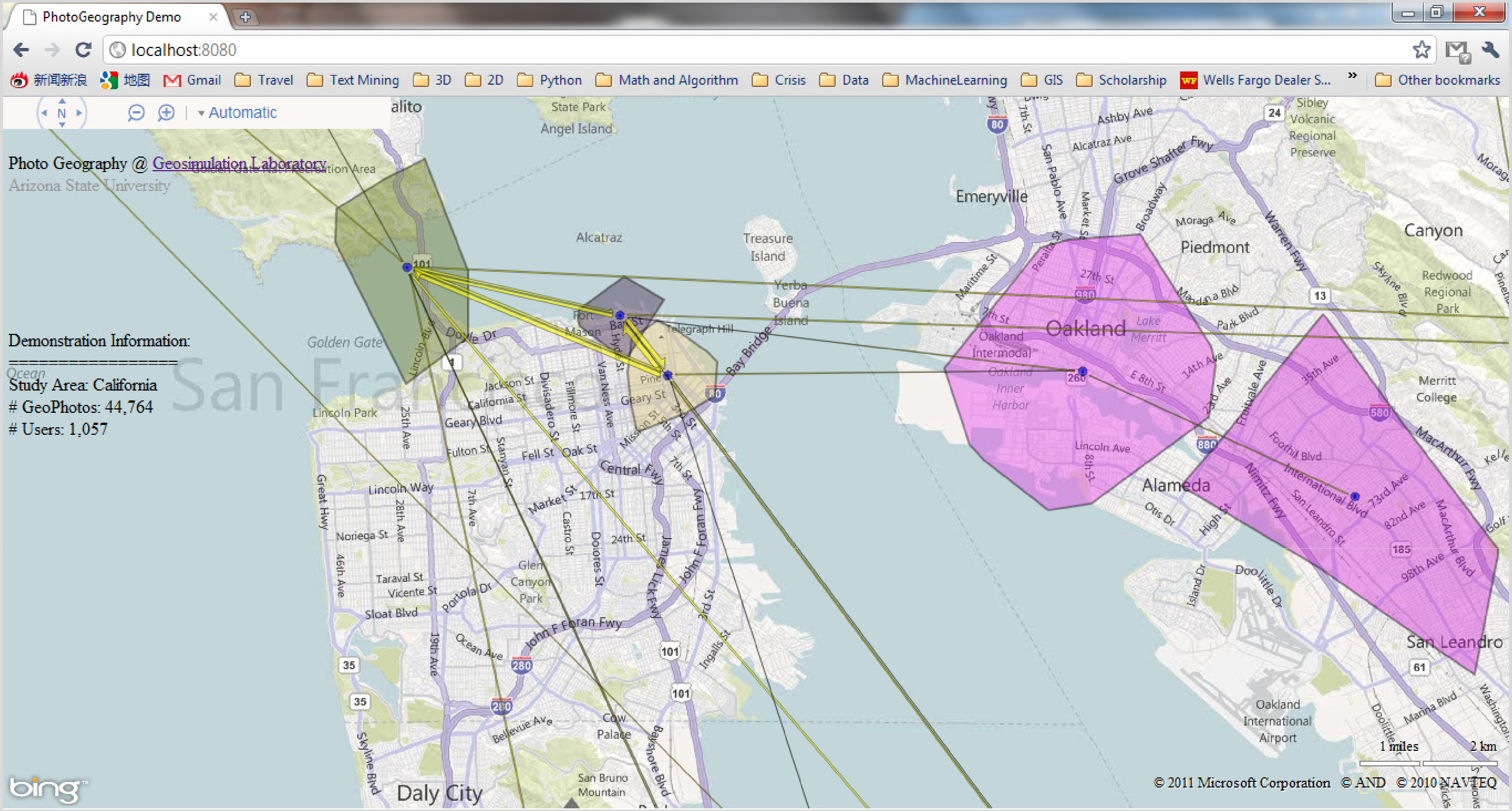
Los Angeles, zoom-level 19 (metropolitan)

(Work with Xun Li)



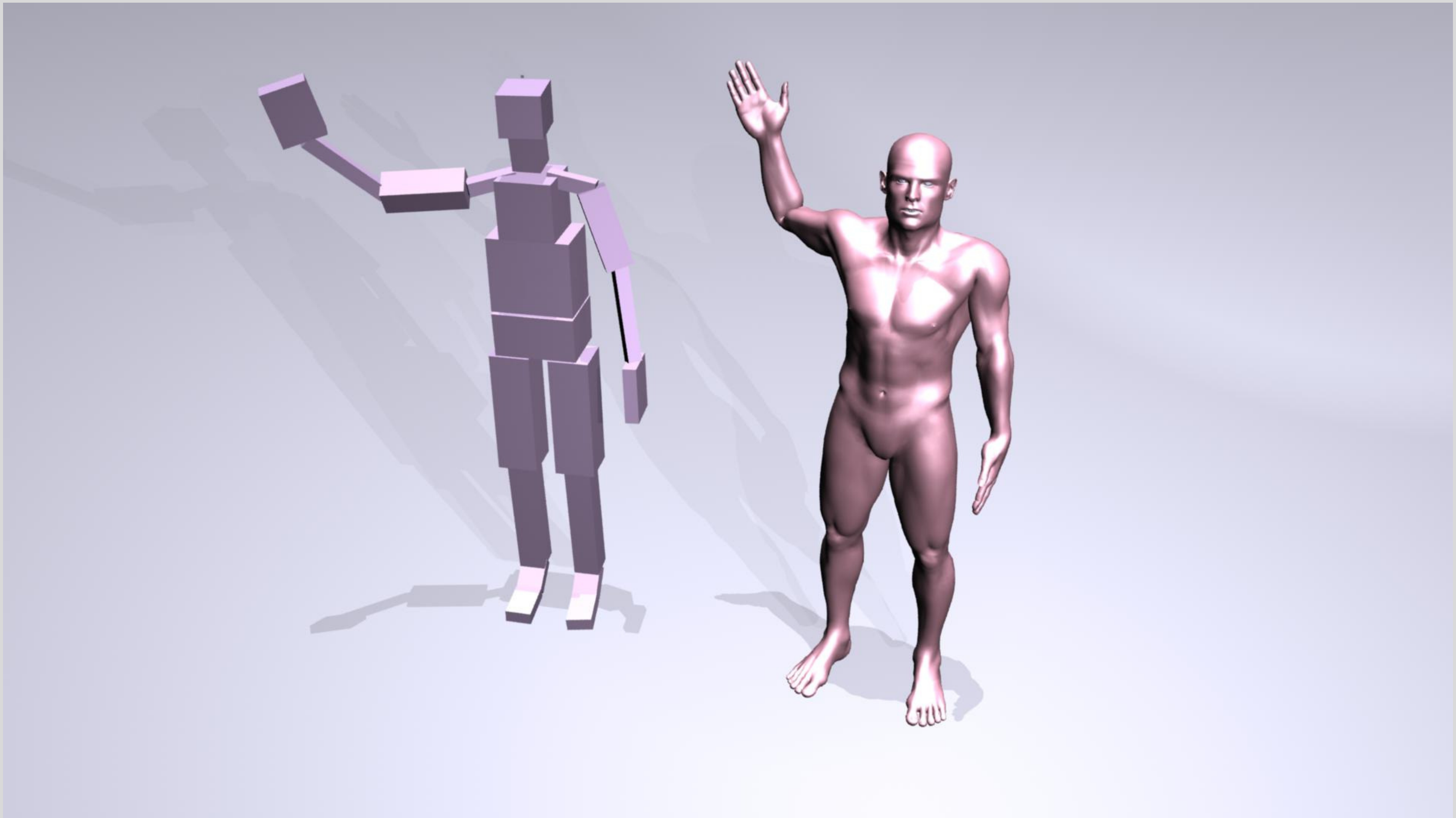
San Francisco, zoom-level 9 (county)

(Work with Xun Li)



San Francisco, zoom-level 19 (metropolitan)

(Work with Xun Li)



Thanks!

Presidential Early Career Award for Scientists and Engineers

NSF CAREER Award
NSF Civil and Mechanical Systems
NSF Human Dynamics of Social Change
NSF Geography & Spatial Sciences
NSF Methodology, Measurement, and Statistics

Science Foundation Arizona

Autodesk, Inc.

Alias, Inc.

The Johns Hopkins University School of Medicine

Geosimulation

Home | Background | Book | Research | Publications | Bio | Press | Geosimulation Labs

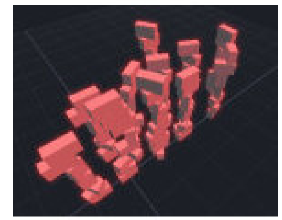
Dr. Paul M. Torrens, Department of Geography, University of Maryland, torrens at geosimulation dot com

News | New publications | New grants | Contact details

Presidential Early Career Award for Scientists and Engineers >>

I was awarded the *Presidential Early Career Award for Scientists and Engineers* by President Bush in a ceremony at the White House on December 19, 2008. The award was for my work on computer models of human behavior in critical situations. **Press releases:** White House; Executive Office of the President, Office of Science and Technology Policy; National Science Foundation; Arizona State University; The Association of American Geographers; UCL Centre for Advanced Spatial Analysis (CASA); UCL Department of Geography; *Engineering News Record*; *The Irish Times*. (Photo by Chris Greenberg.)

Projects >>



Dynamic physics for built infrastructure

Moving agents through space and time

http://geosimulation.org

V fractal dimension

Turning angle between steps

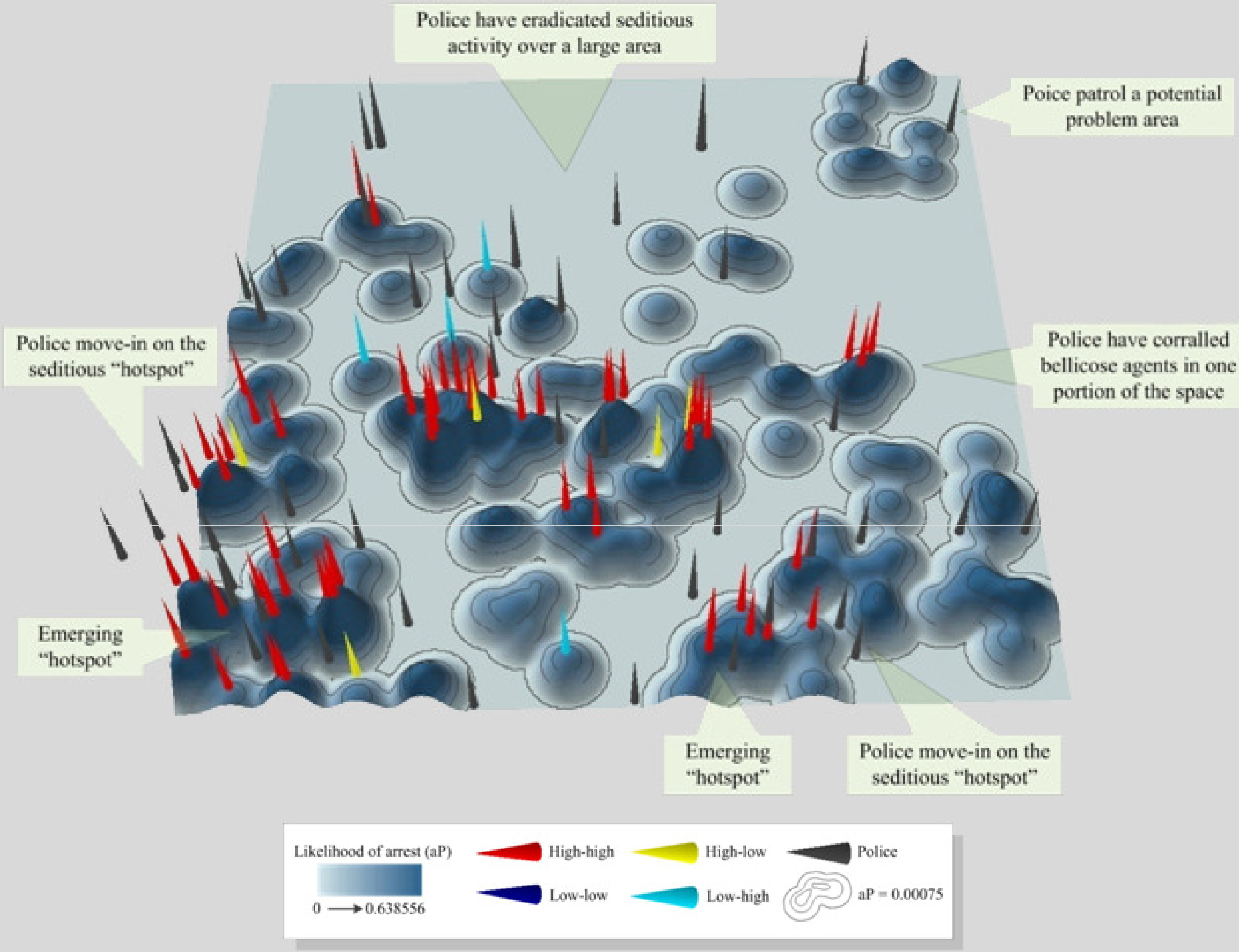
$$d_v = \frac{\log(2)}{\log\left(\frac{Net}{step}\right)} = \frac{\theta}{1 + \log_2(\cos\theta + 1)}$$

Consecutive step

Net length of step-pairs

Simulating riot geography

Uh-oh


Police have eradicated seditious activity over a large area
Poice patrol a potential problem area
Police move-in on the seditious "hotspot"
Police have corralled bellicose agents in one portion of the space
Emerging "hotspot"
Emerging "hotspot"
Police move-in on the seditious "hotspot"
Likelihood of arrest (aP)
0 → 0.638556
High-high
Low-low
High-low
Low-high
Police
aP = 0.00075