Modelling Complex Systems

Self-propelled particles

This lecture is adapted from Vicsek, T. & Zafeiris, A. (2012) Collective Motion. And slides of David Sumpter

See: arXiv:1010.5017v2




Why do animals move together?

- Increased accuracy (many estimates)
- Increased awareness (many eyes)
- Confuse predators and reduce encounters

How do animals move together?

- Group formation usually seems to be *spontaneous.*
- Based on local interactions
- Phenomenological models
- Can ignore 'first principles' physics!
  e.g. Conservation of momentum
- Use biological principles and limits instead.

Random walk in one dimension

- Run 'RandomWalk1D'

current position

current velocity

future position

$$x_i(t+1) = x_i(t) + v_0\, u_i\,(t)$$

$$u_i(t+1) = a u_i(t) + e_i(t)$$

future velocity

current velocity

stochastic effect

$e_i(t)$ *is a* random number selected uniformly at random from a range $\left[-\eta/2, \eta/2\right]$

Attraction in one dimension

- Run 'Aggregate1D'

future position

current position

current velocity

$$x_i(t+1) = x_i(t) + v_0\, u_i\,(t)$$

$$u_i(t+1) = a u_i(t) + (1\text{-}a)\, s_i(t) + e_i(t)$$

future velocity

current velocity

stochastic effect

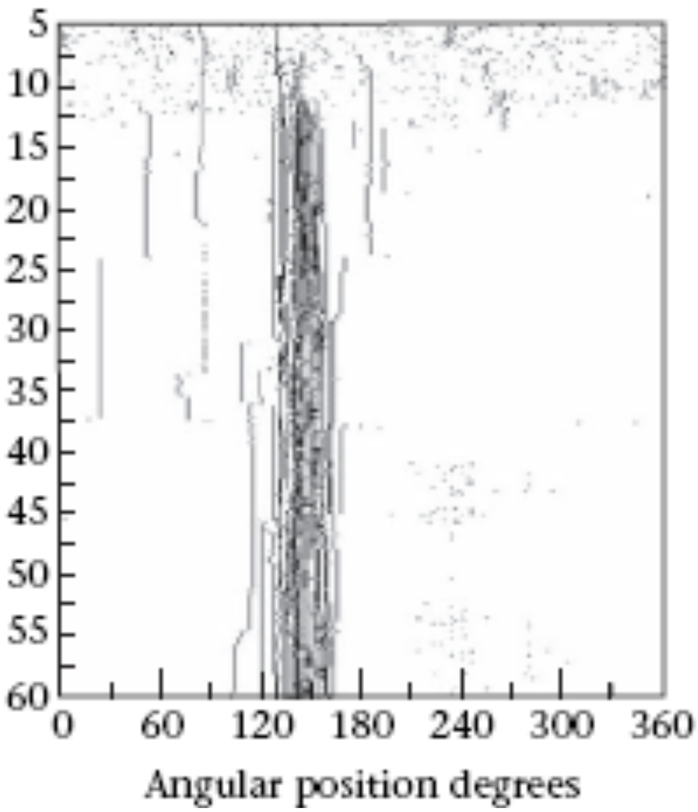
Direction to most neighbours

$$s_i(t) = \frac{1}{\left|R_i\right|} \sum_{j \in R_i} \operatorname{sign}\Big(x_i(t) - x_j(t)\Big)$$

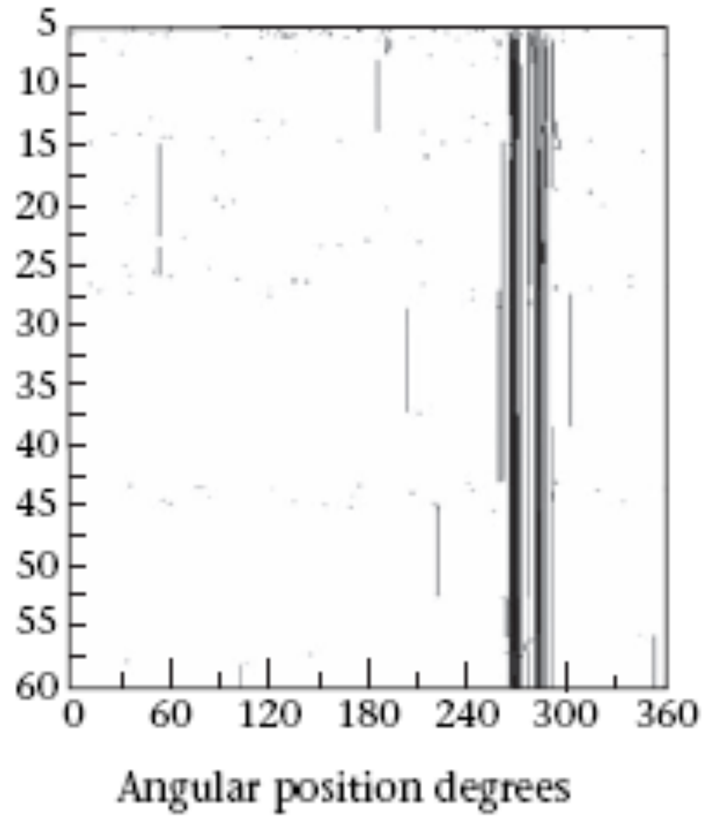
$e_i(t)$ *is a* random number selected uniformly at random from a range $\left[-\eta/2, \eta/2\right]$

Cockroach aggregation

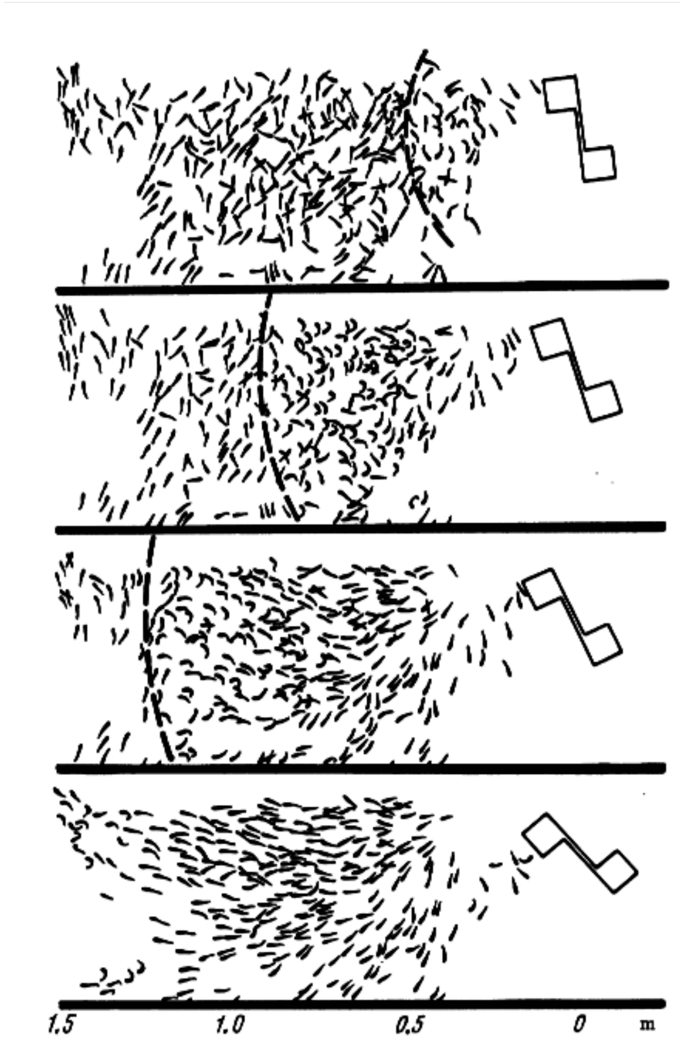
Cockroaches



Model



Radakov's fish



Alignment model in one dimension

- Run 'Align1D'

current position

current velocity

future position

$$x_i(t+1) = x_i(t) + v_0\, u_i\,(t)$$

$$u_i(t+1) = a u_i(t) + (1-a)\, s_i(t) + e_i(t)$$

future velocity

current velocity

stochastic effect

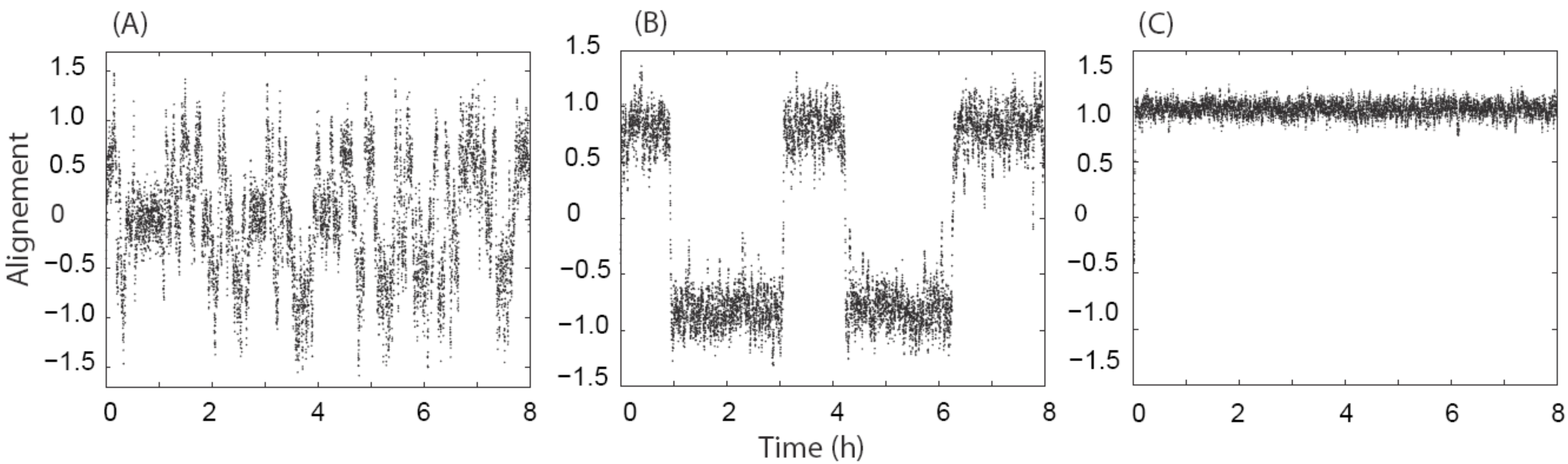
velocity of neighbours

$$s_i = G\left(\frac{1}{|R_i|}\sum_{j\in R_i} u_j(t)\right)$$

$$G(u) = \begin{cases} (u+1)/2 & \mathit{for} \quad u > 0 \\ (u-1)/2 & \mathit{for} \quad u < 0 \end{cases}$$

e is a random number selected uniformly at random from a range $[-\eta/2, \eta/2]$

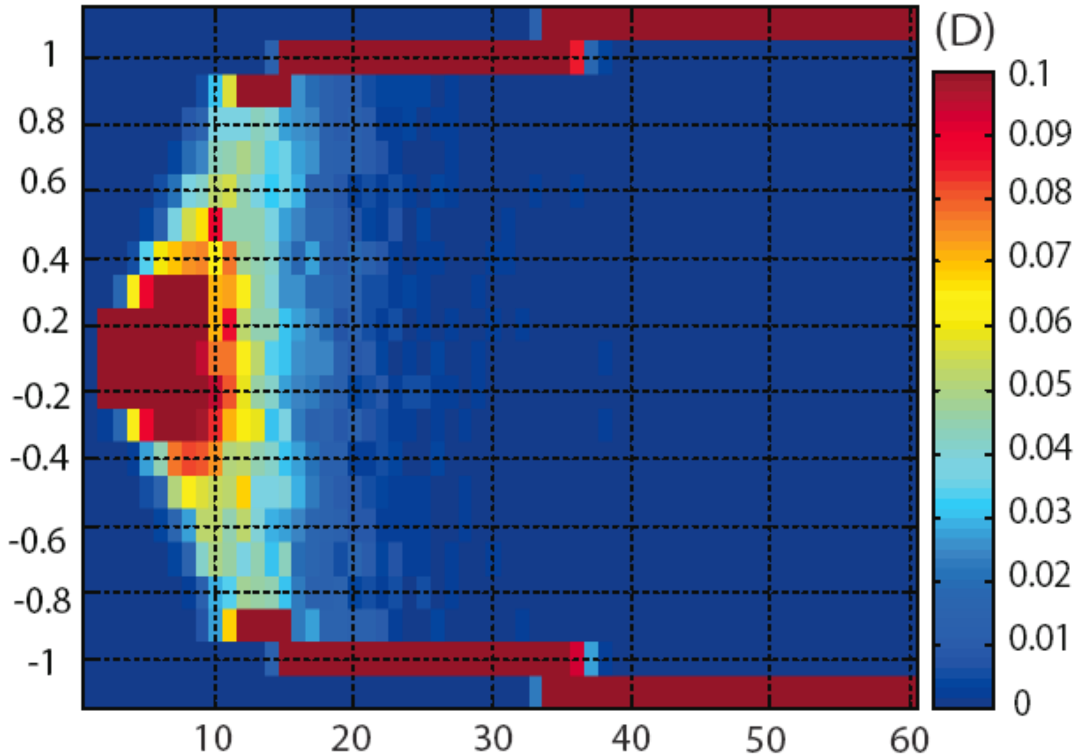
Alignment



$$\phi = \frac{1}{n}\sum_{i=1}^{n} \underline{u}_i(t)$$ measures order in the system.

1D self-propelled particles

Alignment

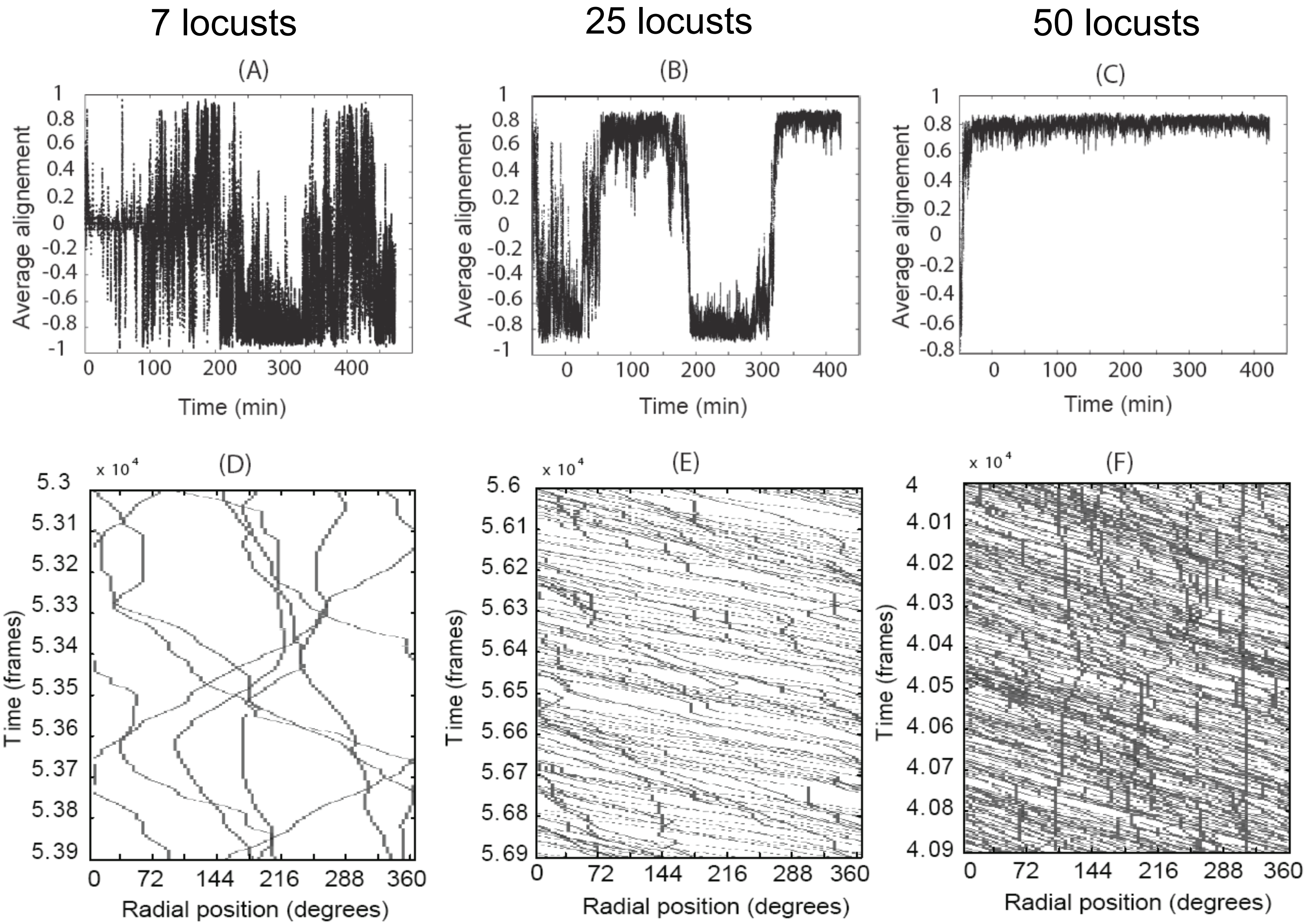


Number of particles

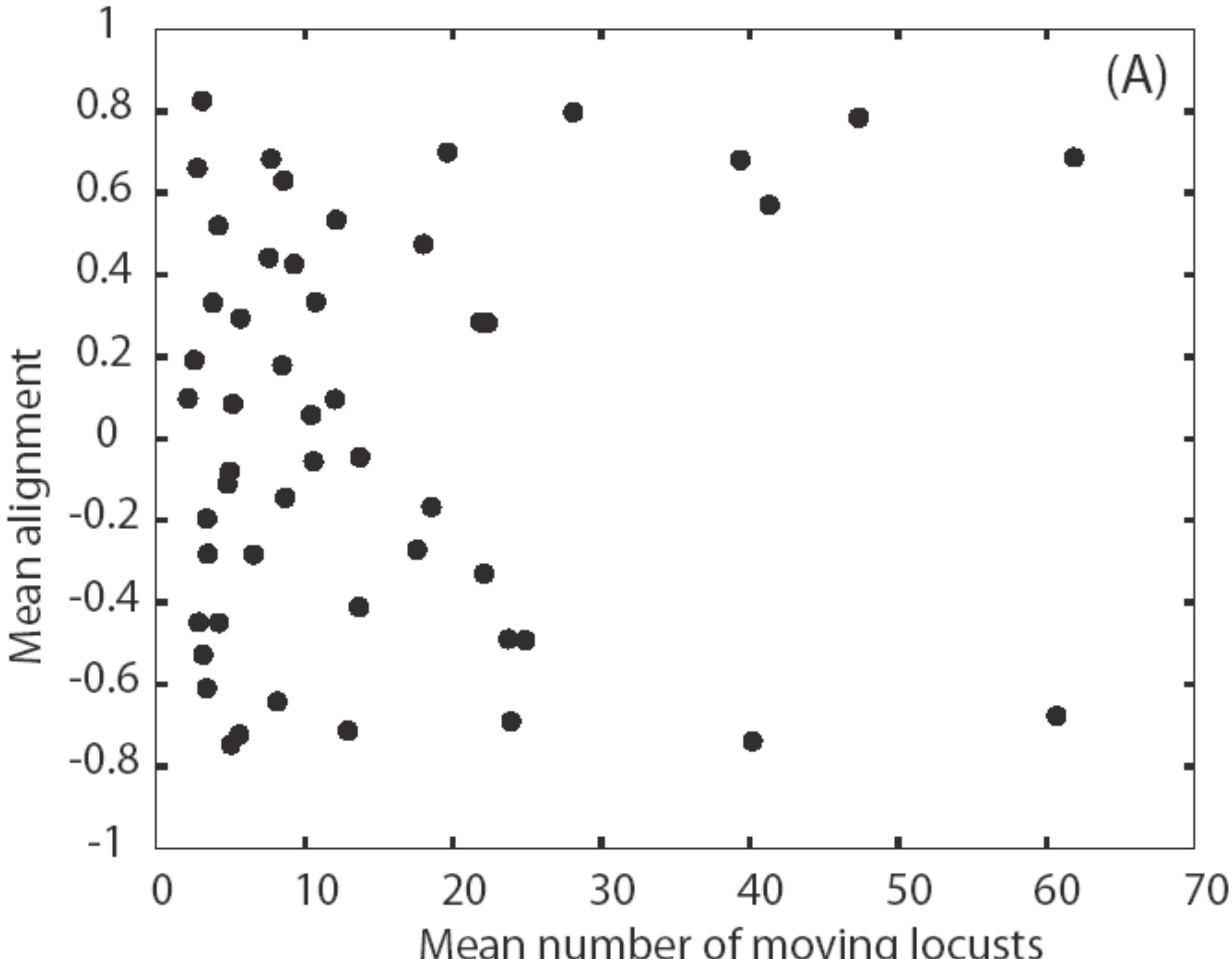
$$\phi = \frac{1}{n}\sum_{i=1}^{n} \underline{u}_i(t)$$ measures order in the system (alignment).

Buhl et al. (2006), *Science*
Yates et al. (2009), *PNAS*

Buhl et al. (2006), *Science*
Yates et al. (2009), *PNAS*

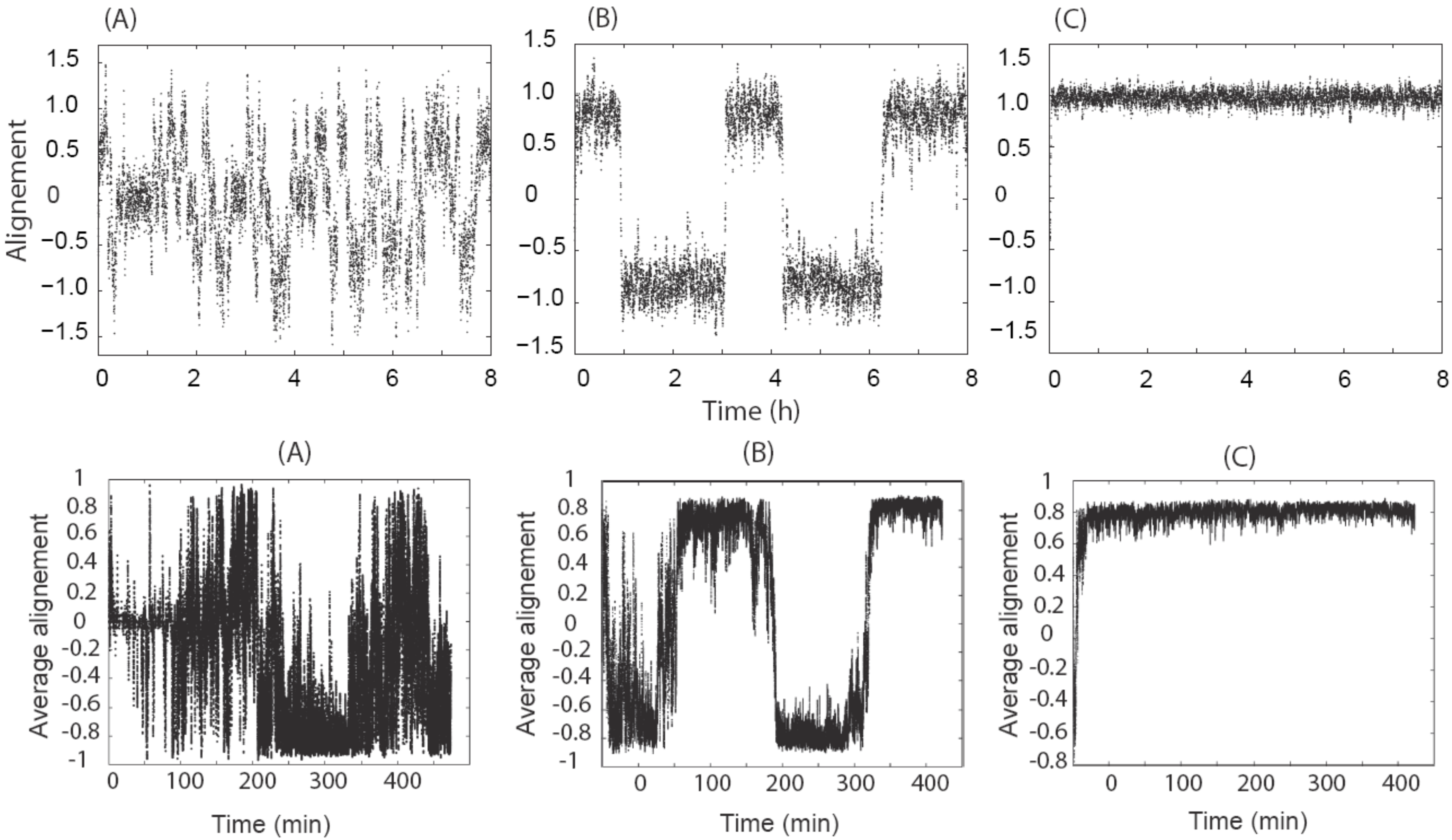


Buhl et al. (2006), *Science*

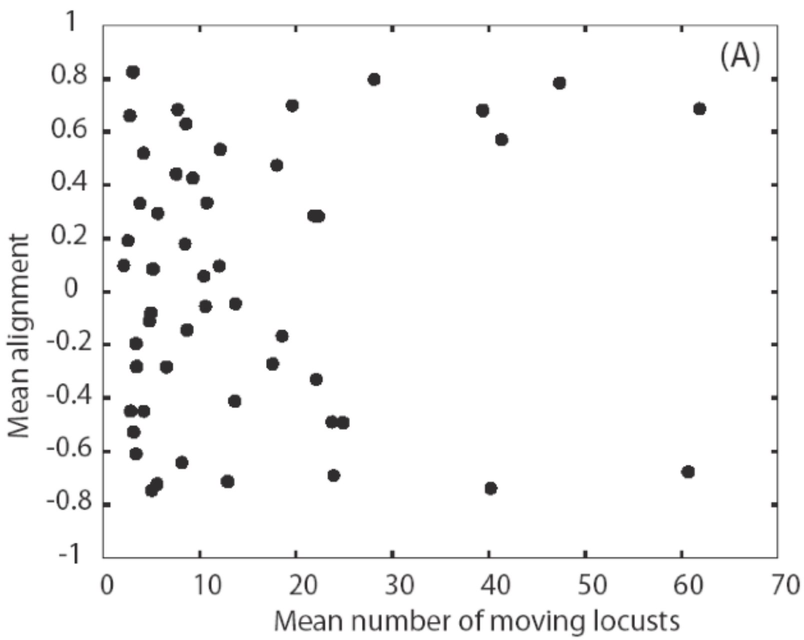


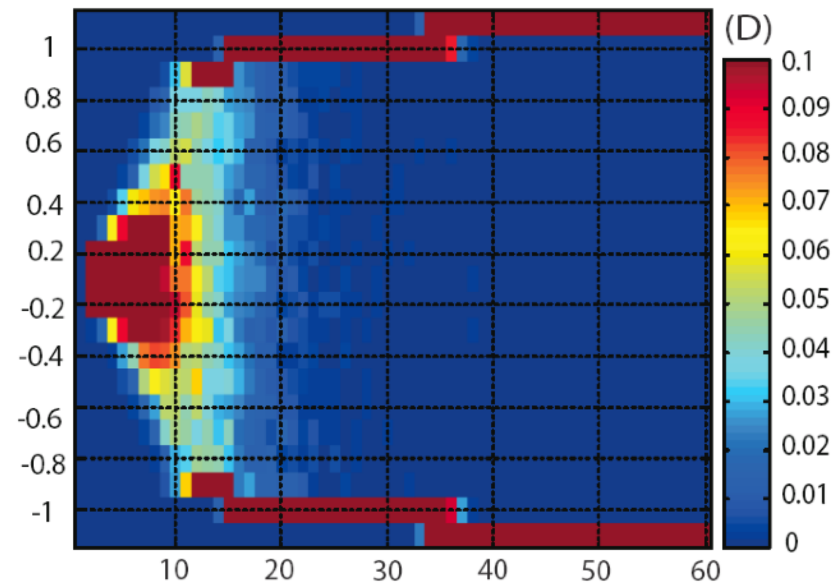
Buhl et al. (2006), *Science*

Model vs Experiment

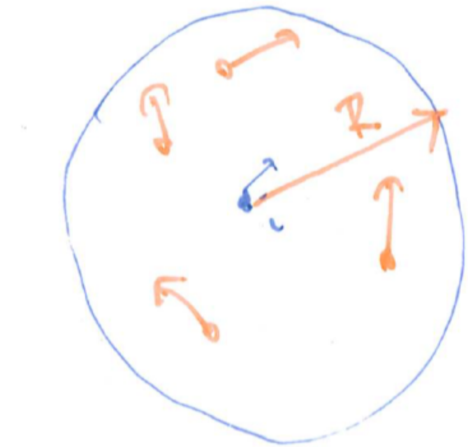


Model vs Experiment





Vicsek Model



N: number of particles
η: noise parameter
L: size of domain
R: radius of interaction
v: speed

Angular update rule:

$$\theta_i(t+1) \quad = \quad \tan^{-1}\left(\frac{\sum_{j\in R_i}\sin(\theta_j(t))}{\sum_{j\in R_i}\cos(\theta_j(t))}\right) + e(t)$$

e(t) is a random number selected uniformly at random from a range $\left[-\eta/2, \eta/2\right]$

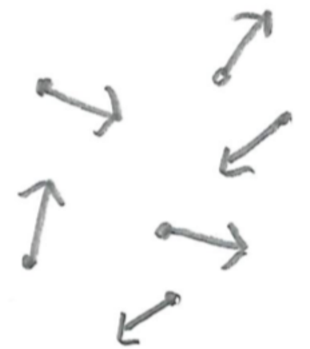
Vicsek et al., PRL 75 (1995)

2D Alignment

- Run ‘Align2D’

Measure of Alignment: Polarisation

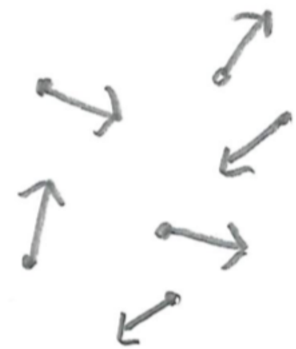
High polarisation



Low Polarisation

Measure of Alignment: Polarisation

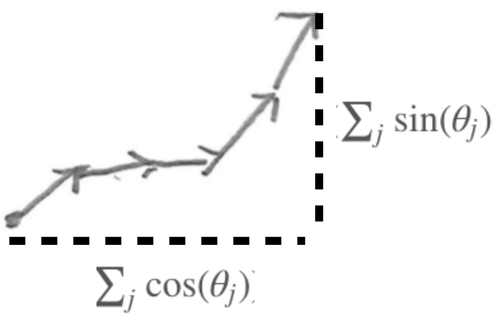
High polarisation

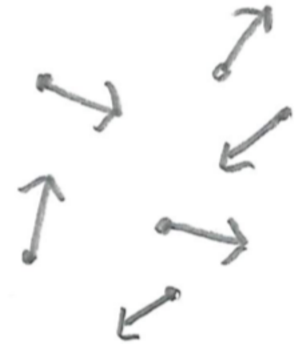


Low Polarisation

Measure of Alignment: Polarisation

High polarisation

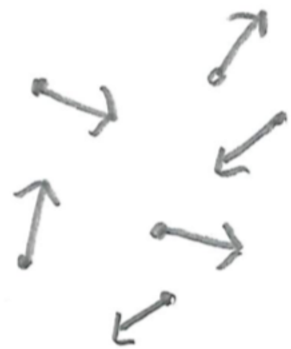




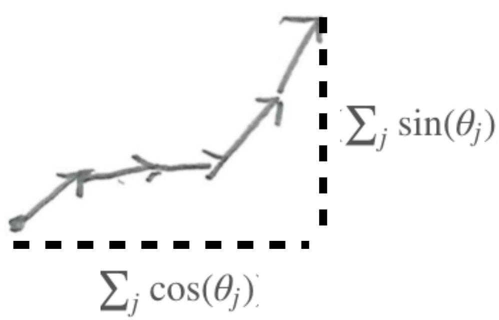
Low Polarisation

Measure of Alignment: Polarisation

High polarisation



Low Polarisation

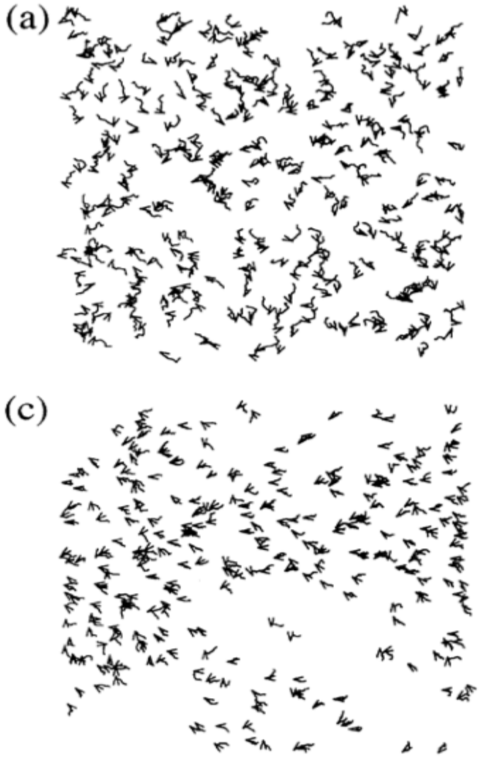


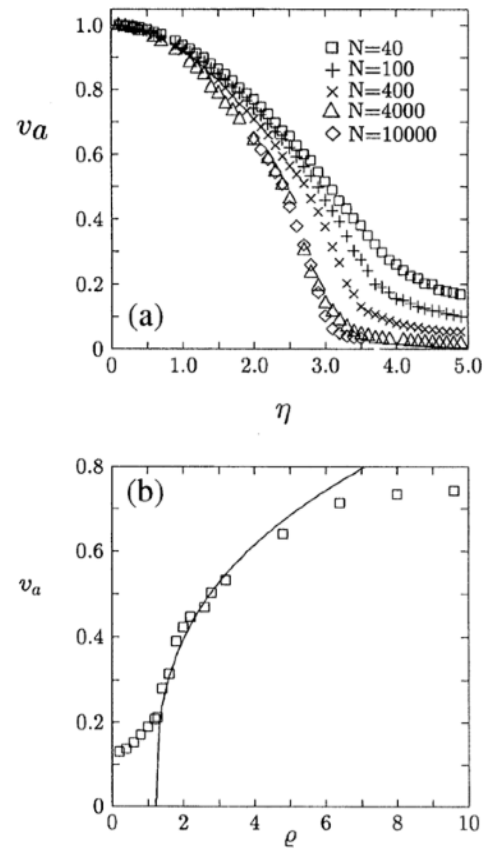
Polarisation of:

$\theta_1, \theta_2, \ldots, \theta_N$

$$= \frac{1}{N}\sqrt{(\sum_j \sin(\theta_j))^2 + (\sum_j \cos(\theta_j))^2}$$

Vicsek Model

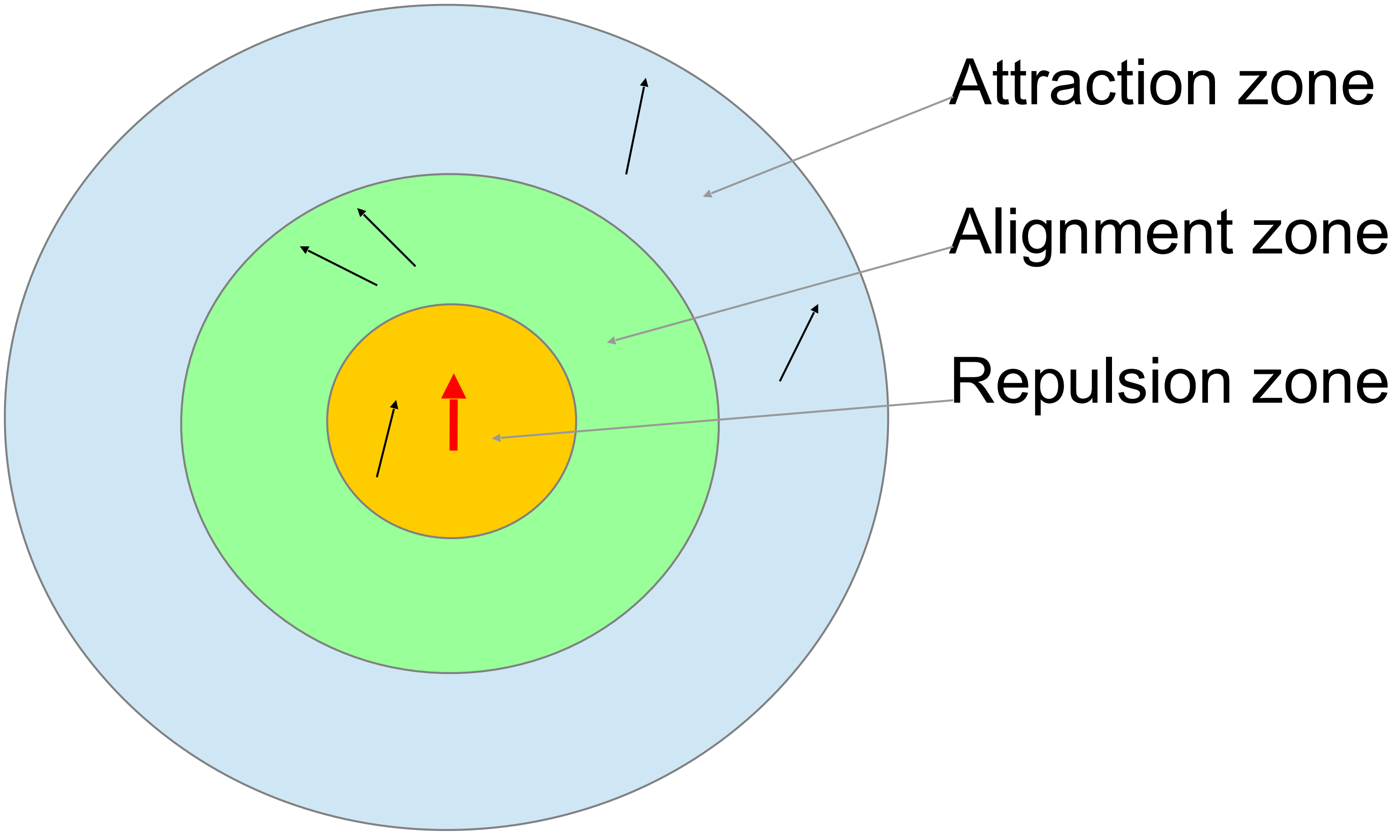




Vicsek et al., PRL 75 (1995)

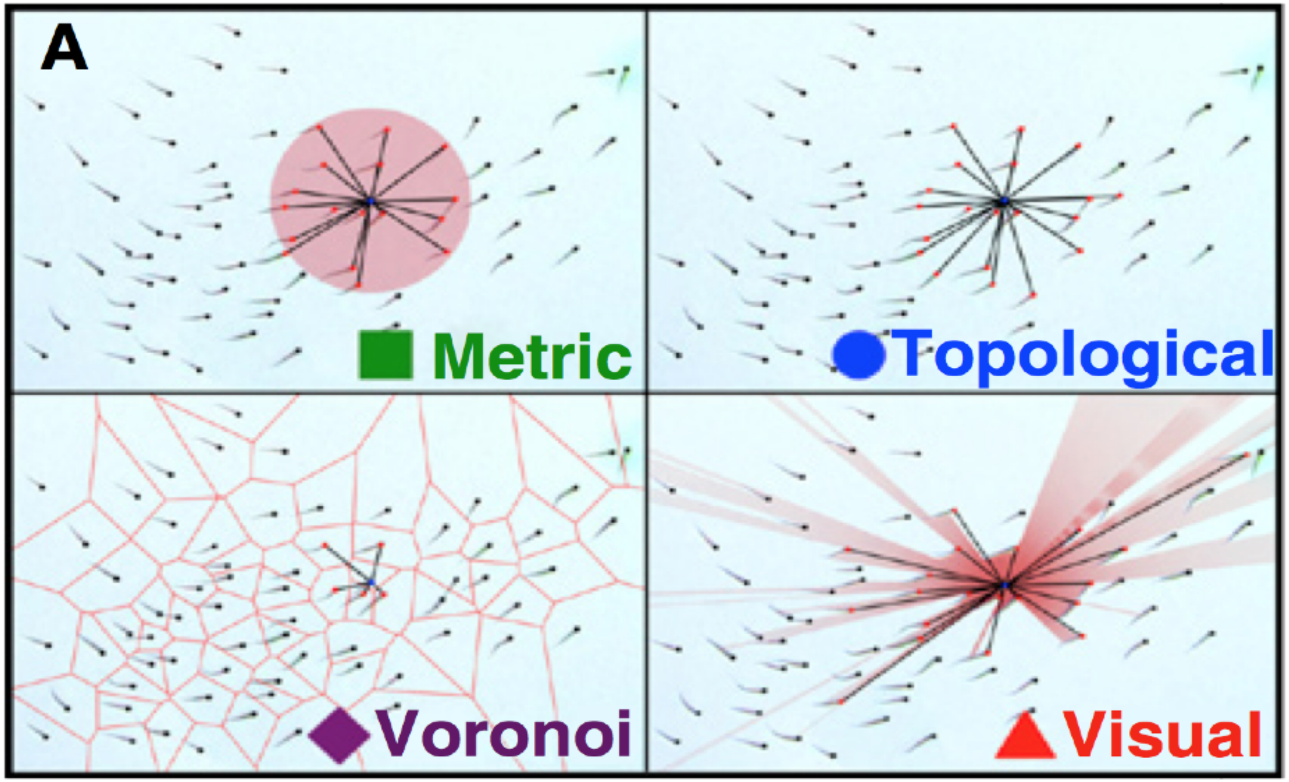
Attraction/Repulsion

"Boids" model



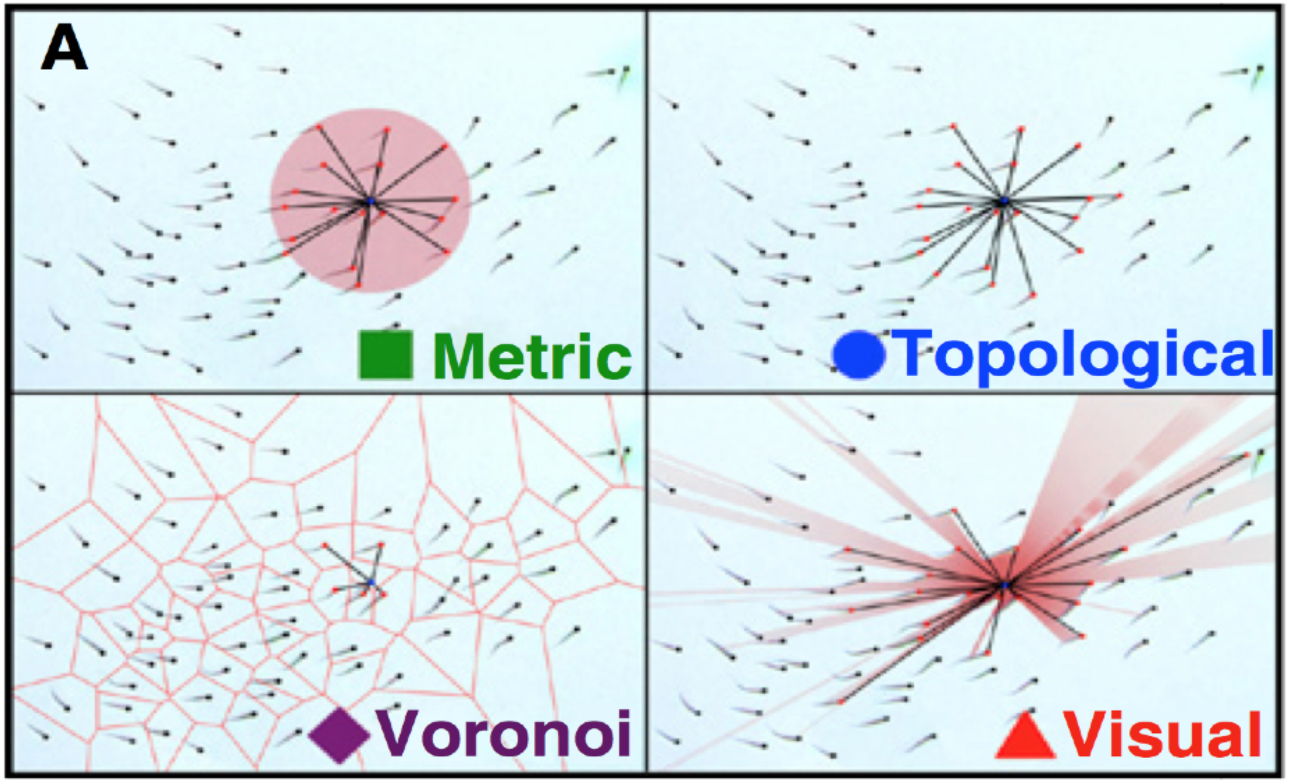
See: Couzin et al., J. Theor. Bio. (2002)

Alternative distance measures



Strandburg-Peshkin et al., Current Biology (2013)

Alternative distance measures



Metric: all individuals within a certain distance.
Topological: a fixed number of nearest neighbors.
Voronoi: those individuals sharing a boundary in a Voronoi tes-sellation of the group.
Visual: all individuals that occupy an angular area on the retina of the focal fish that is greater than a threshold value.

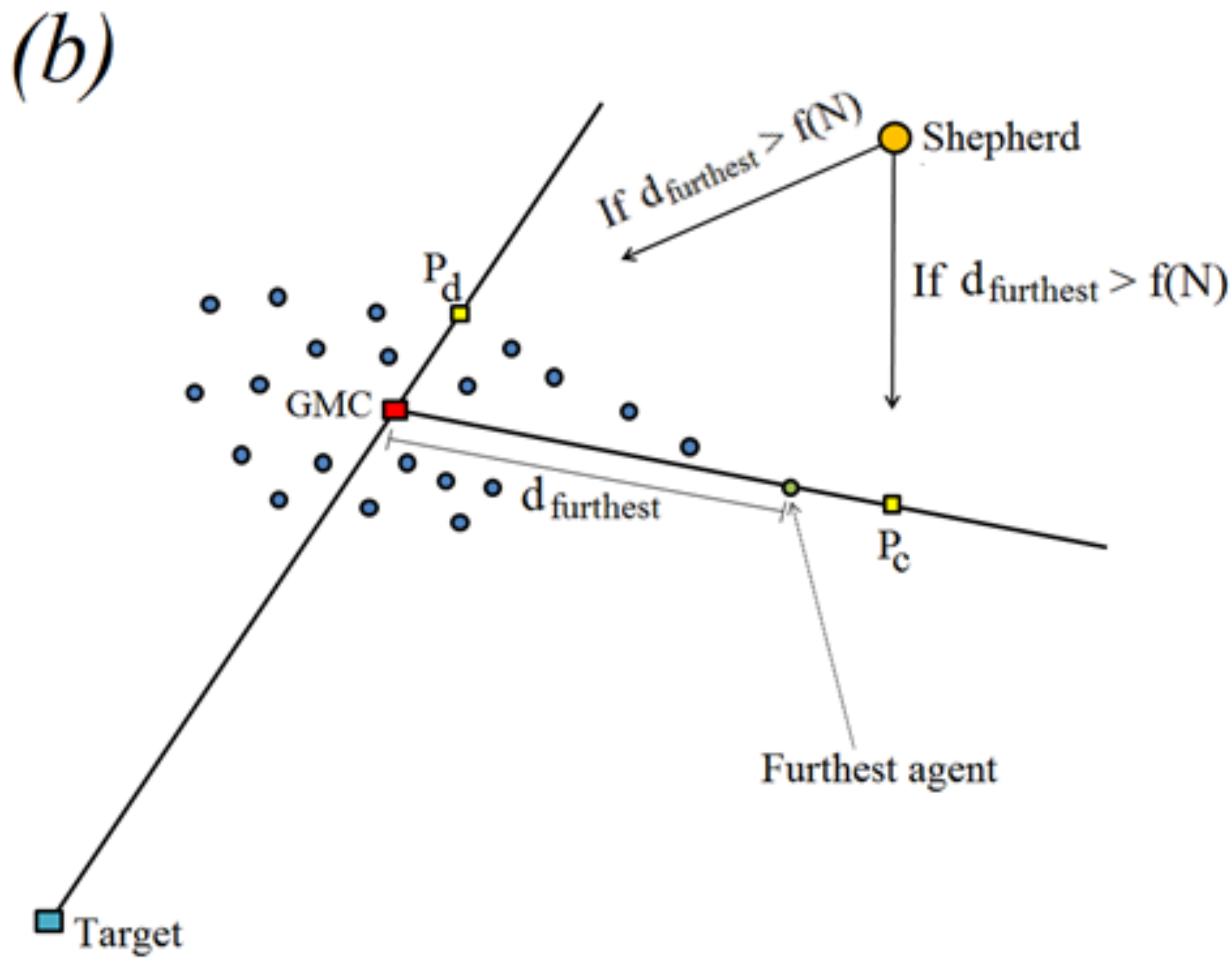
Strandburg-Peshkin et al., Current Biology (2013)

Even more options

- Maximum turning angles
- Blind angles
- Attraction/repulsion potentials
- Reaction times
- Wall interactions
- Variable speed
- Variation in individuals
- Pheromone trails
- Etc....

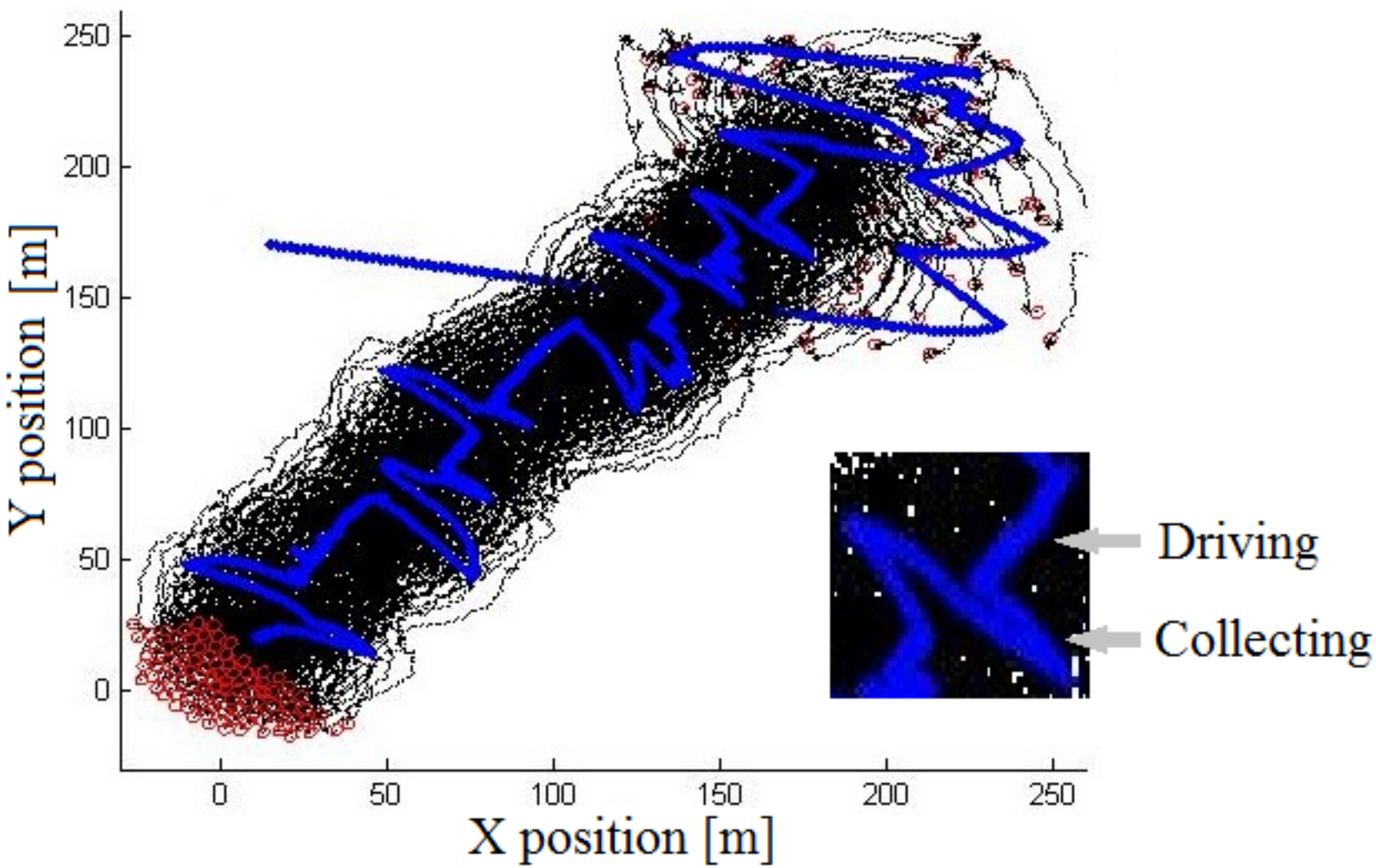
King et al. (2012) *Current Biology*

Sheepdog model



Strömbom et al., J. Royal Soc. Interface (2014)

Drive and collect



Next: when humans go ballistic

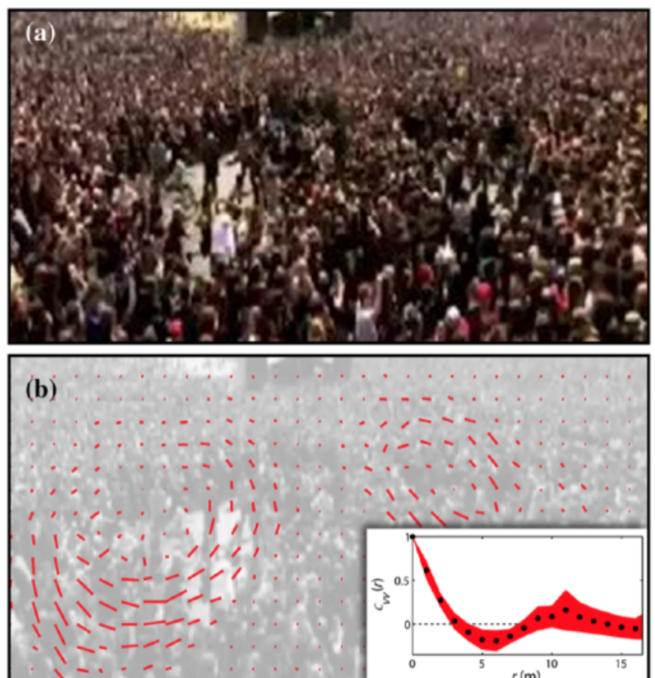
PRL **110,** 228701 (2013) PHYSICAL REVIEW LETTERS week ending 31 MAY 2013

Collective Motion of Humans in Mosh and Circle Pits at Heavy Metal Concerts

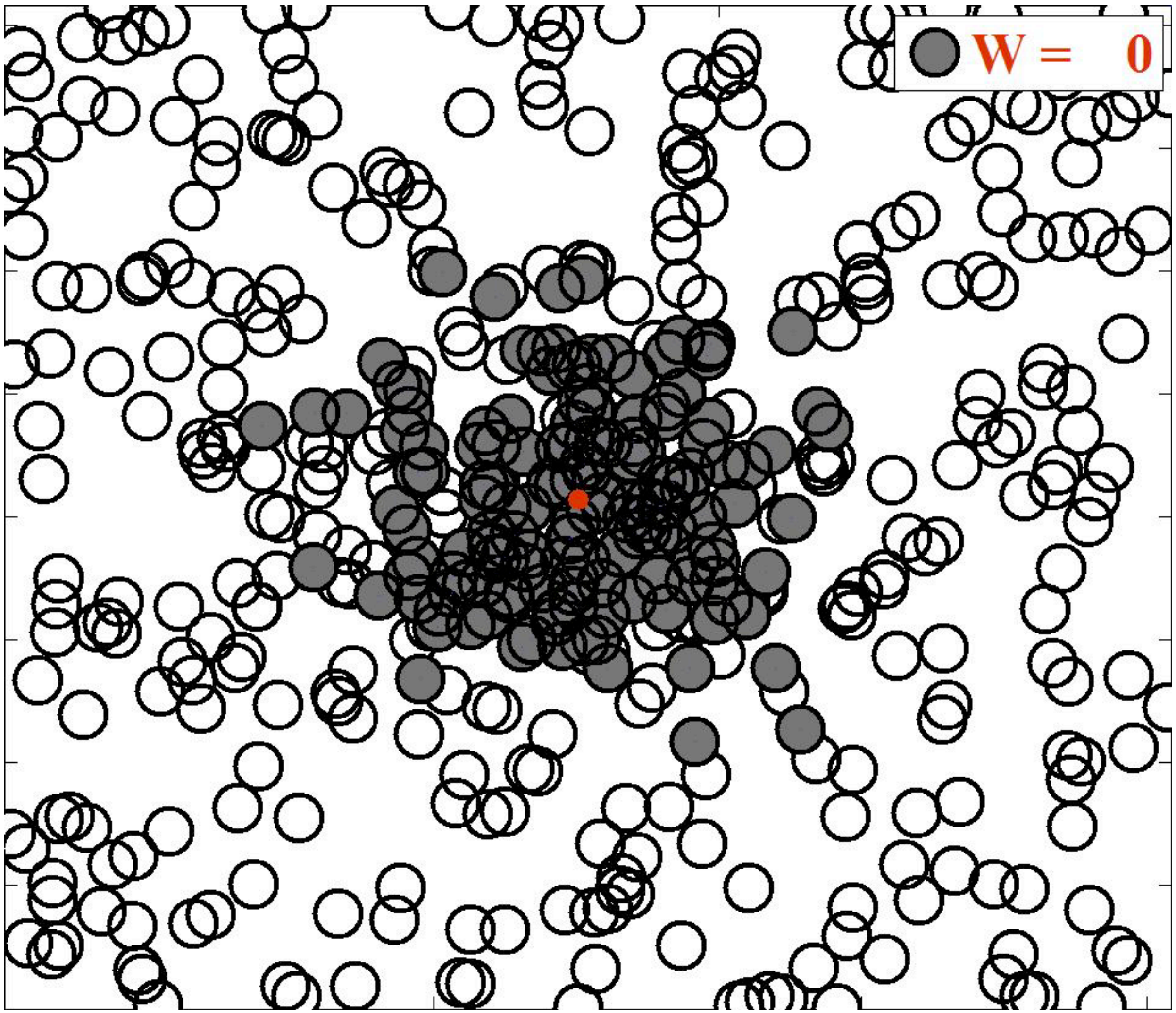
Jesse L. Silverberg,* Matthew Bierbaum, James P. Sethna, and Itai Cohen

Department of Physics and Laboratory of Atomic and Solid State Physics, Cornell University, Ithaca, New York 14853, USA

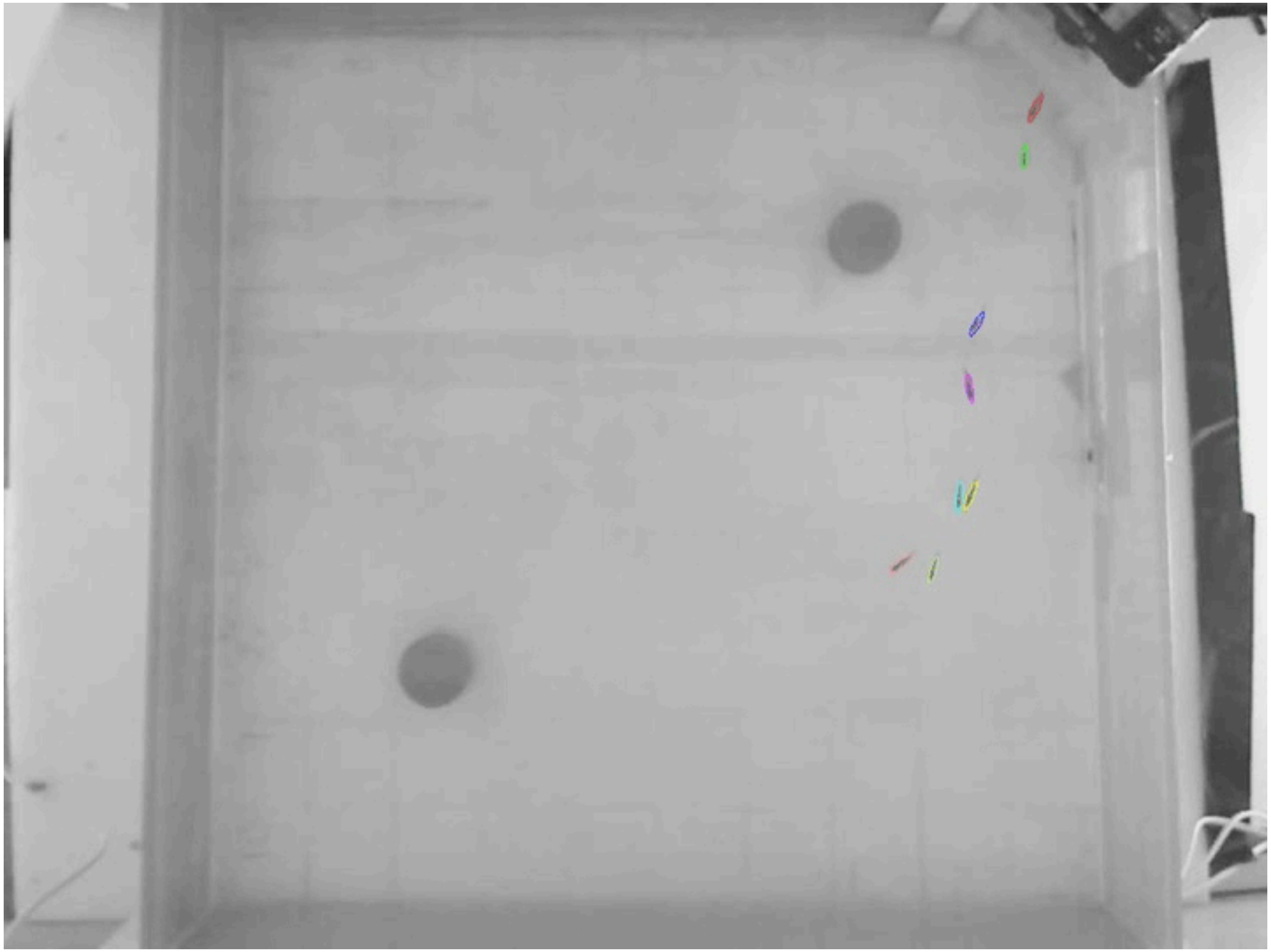
(Received 13 February 2013; published 29 May 2013)



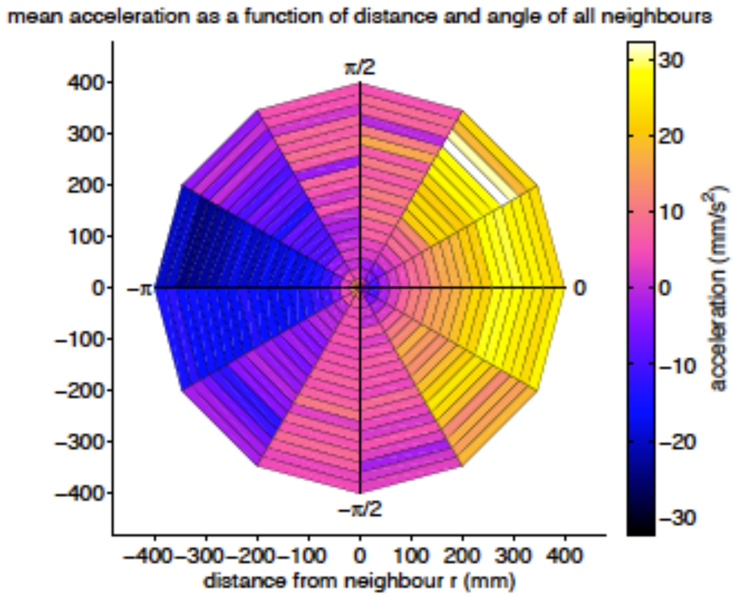
Moshpit model

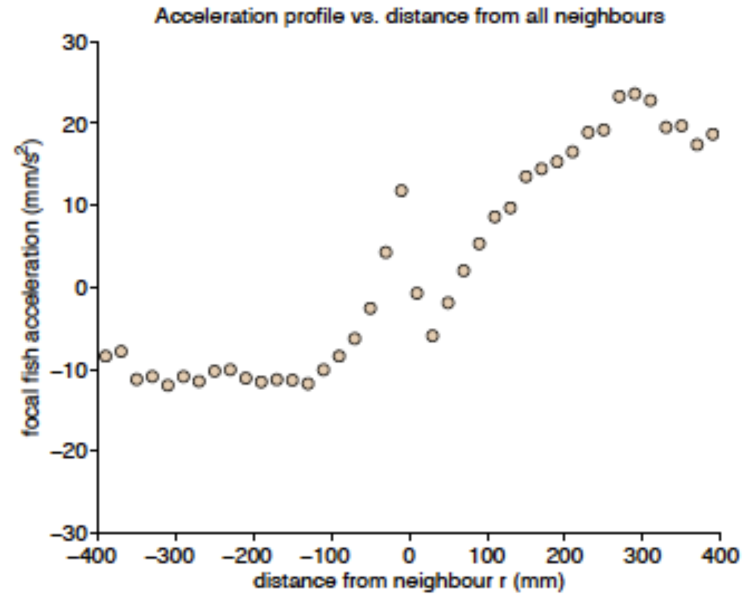


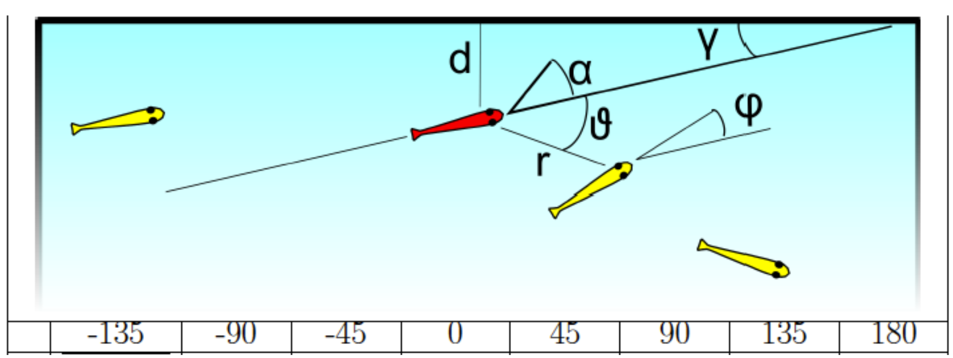
Rules of motion



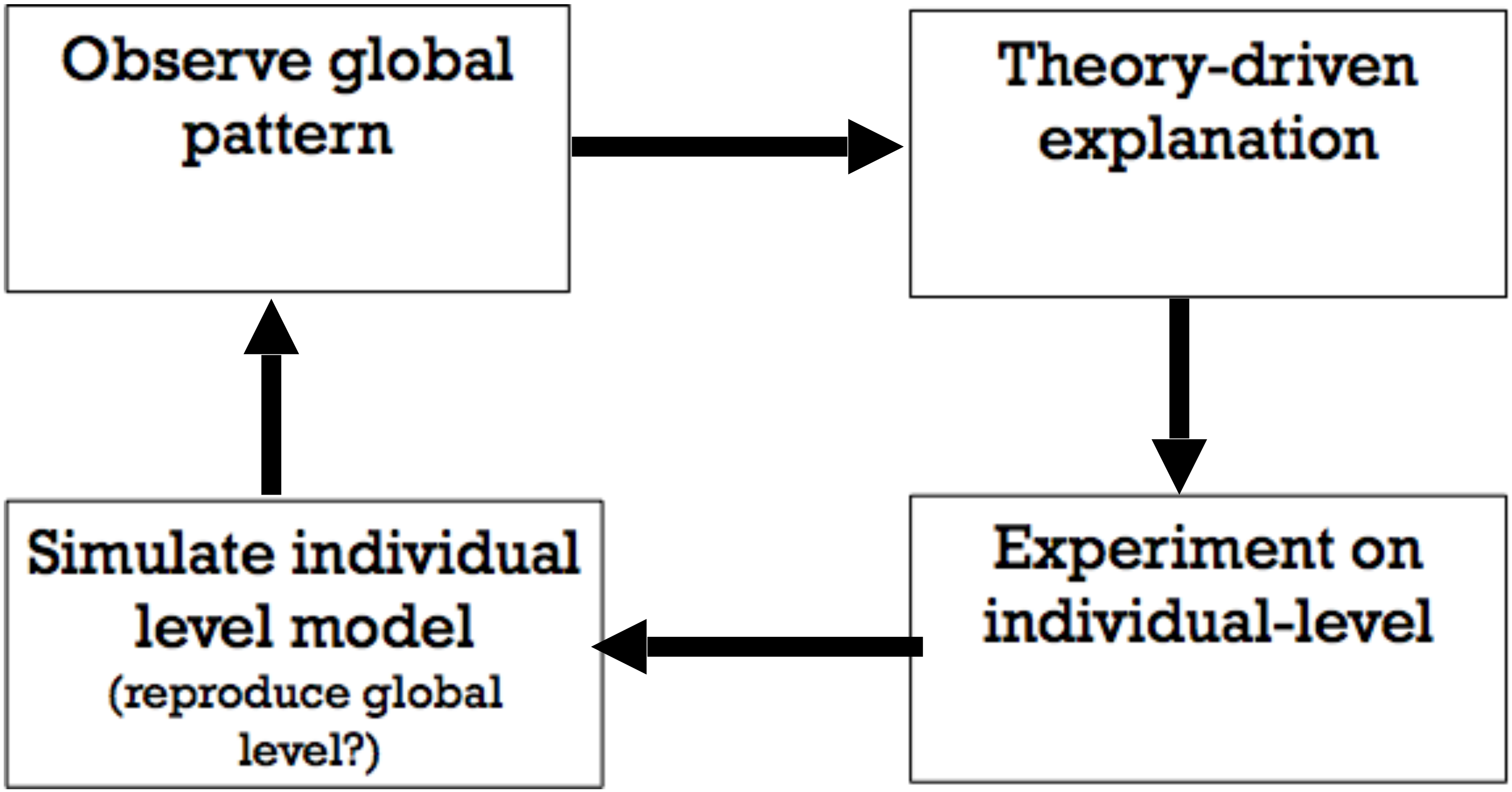
Using data to fit models



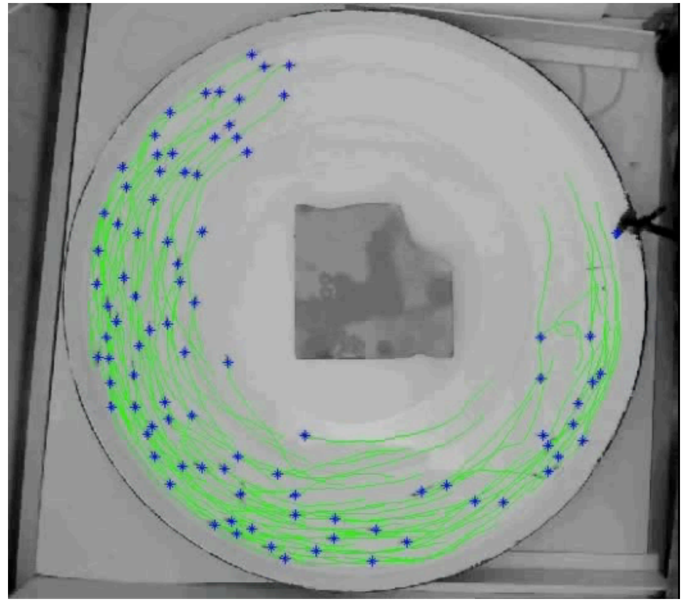


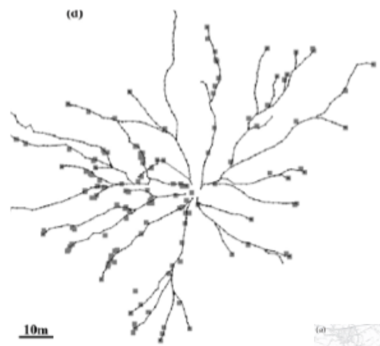


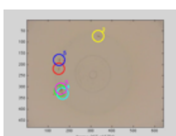
The modelling cycle

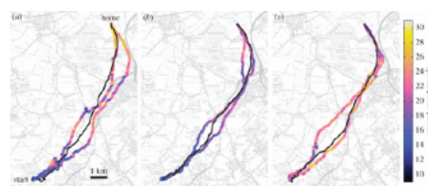



© Andrea Perna 2010



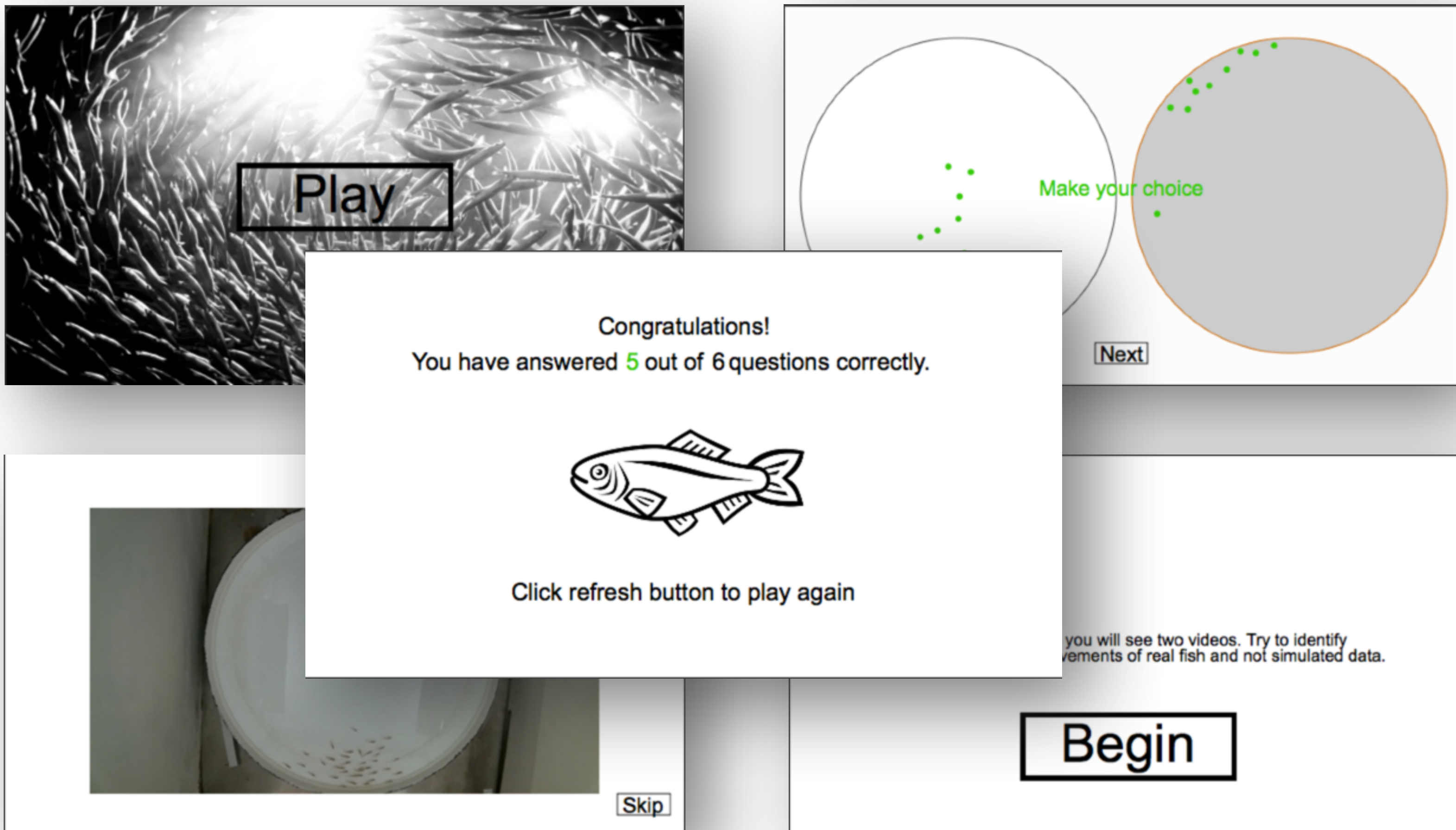

(d)
10m





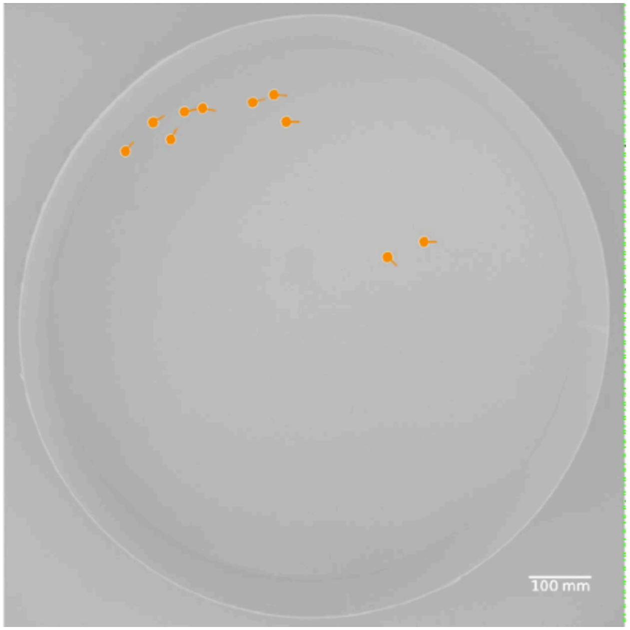
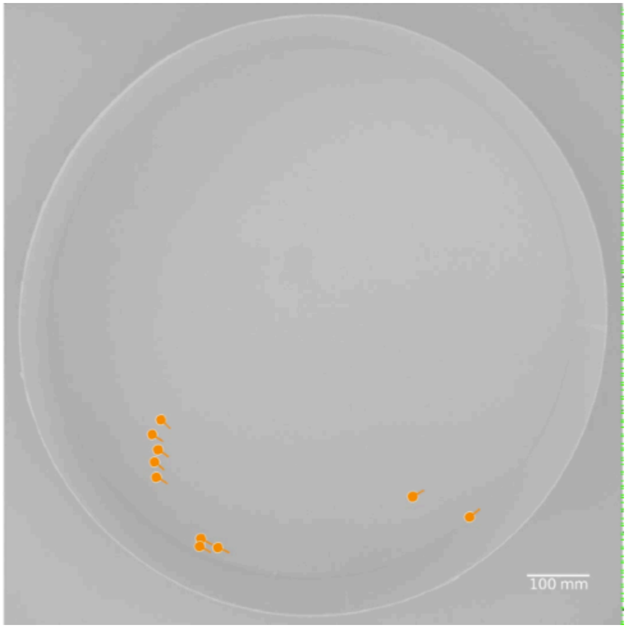
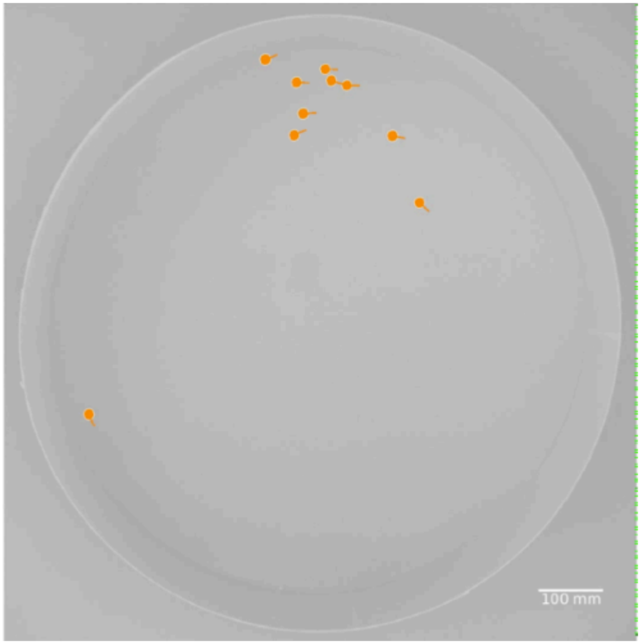

COLLECTIVE BEHAVIOUR

Can you tell the difference between real and simulated fish?

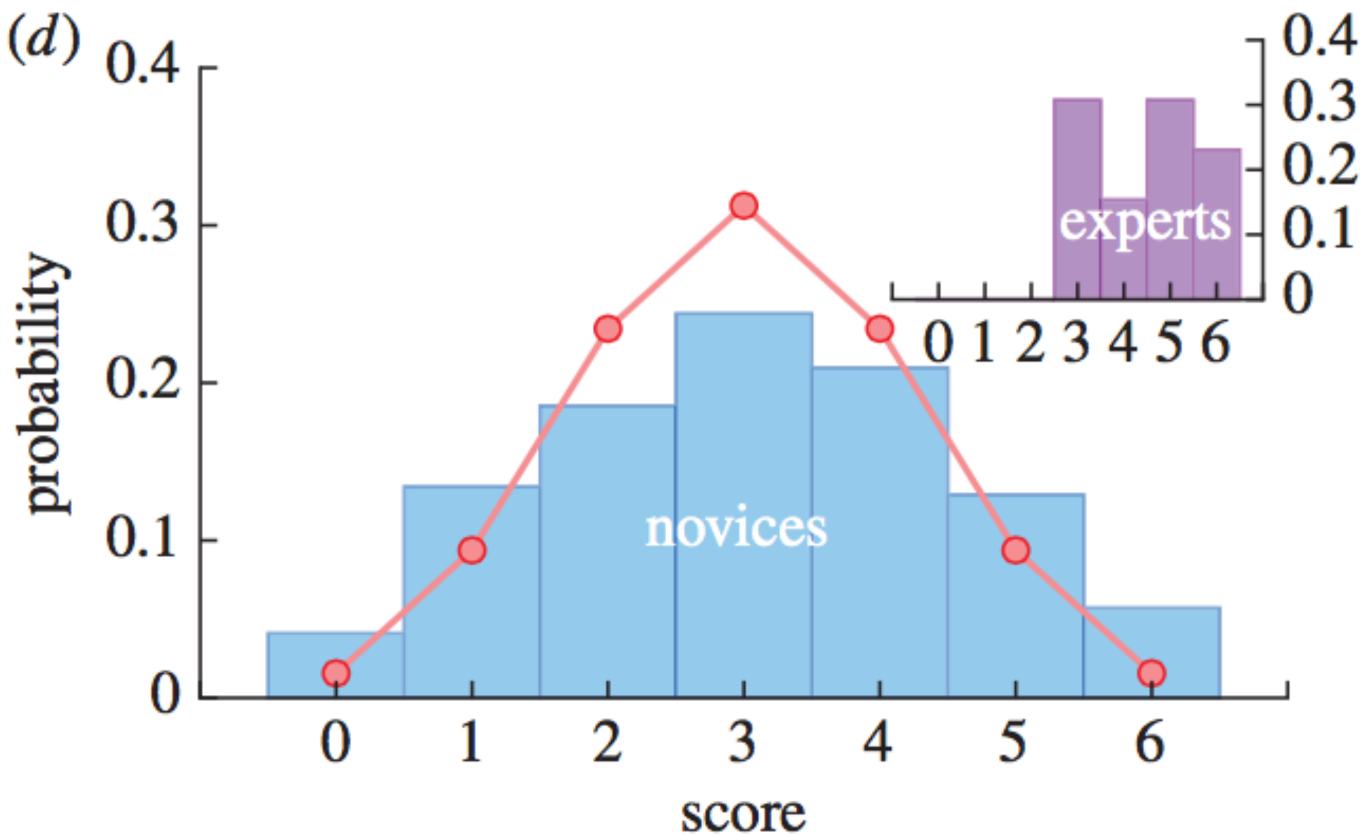


Get playing!

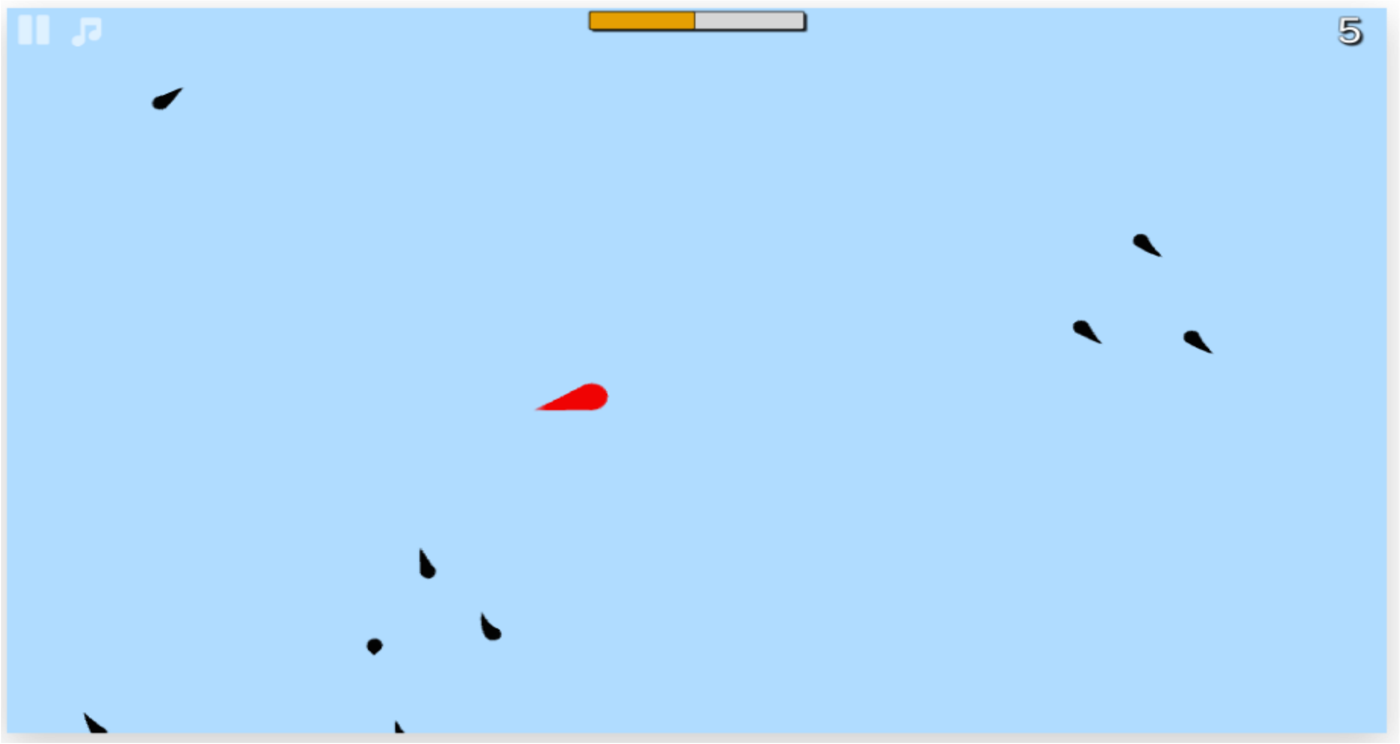
http://www.collective-behavior.com/apps/


100 mm

100 mm

100 mm

100 mm

Can people tell the difference between real and simulated fish?



Evolving prey

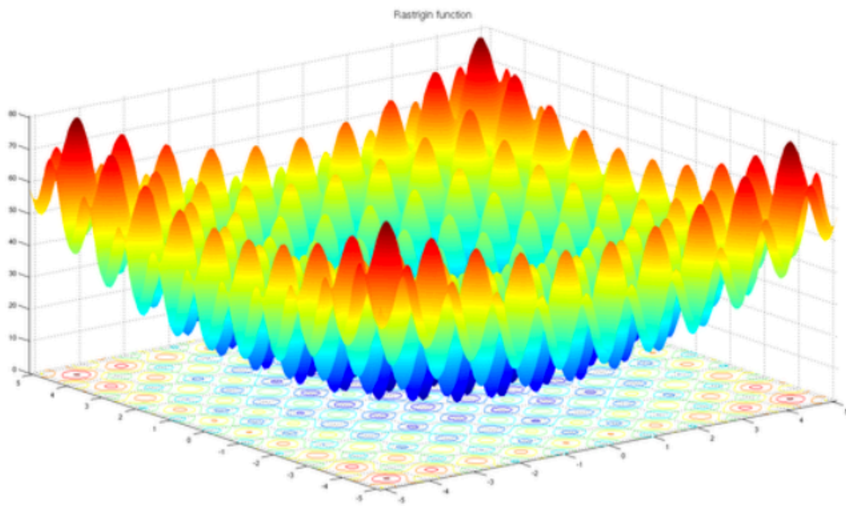


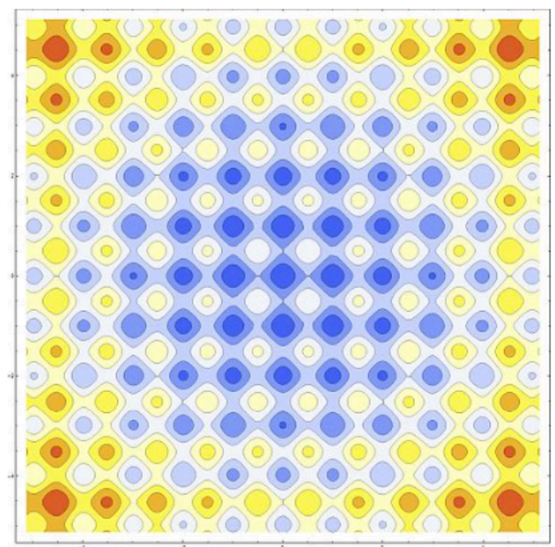
http://collective-behavior.com/apps/fishindanger/webgl

Project - Particle Swarm Optimisation

Optimisation problem.

Find global minimum/local minima.





Benchmark: Rastrign function

$$F(x_1, x_2) = 10n + \sum_{i=1}^{2} x_i^2 - 10\cos(2\pi x_i) \qquad x_i \in [-5.12, 5.12]$$

Miranda, Lester. *PySwarms: a research toolkit for particle swarm optimization in Python*. Journal of Open Source Software (2018)

Recall: attraction in one dimension

current position

current velocity

future position

$$x_i(t+1) = x_i(t) + v_0\, u_i\,(t)$$

$$u_i(t+1) = a u_i(t) + (1-a)\, s_i(t) + e_i(t)$$

future velocity

current velocity

stochastic effect

Direction to most neighbours

$$s_i(t) = \frac{1}{|R_i|} \sum_{j \in R_i} \text{sign}\Big(x_i(t) - x_j(t) \Big)$$

$e_i(t)$ *is a* random number selected uniformly at random from a range $\left[-\eta/2, \eta/2\right]$

Extension: Particle swarm optimisation

future position

current position

current velocity

$$x_i^{(t)} = x_i^{(t-1)} + v_i^{(t)}$$

$$v_i^{(t)} = v_i^{(t-1)} + c_1 U(0,1) \odot \left(p_i - x_i^{(t-1)}\right) + c_2 U(0,1) \odot \left(p_g - x_i^{(t-1)}\right)$$

future velocity

current velocity

stochastic effect

N particles. $p_1, \ldots, p_N$ best positions of each particle.

p_g - best position of particles in neighbourhood

Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95- International Conference on Neural Networks. Vol. 4. IEEE, 1995.

Jonas Olsson

Extension: Particle swarm optimisation

$$\underbrace{x_i^{(t)}}_{\text{future position}} = \underbrace{x_i^{(t-1)}}_{\text{current position}} + \underbrace{v_i^{(t)}}_{\text{current velocity}}$$

$$\underbrace{v_i^{(t)}}_{\text{future velocity}} = \underbrace{v_i^{(t-1)}}_{\text{current velocity}} + c_1 U(0,1) \odot \underbrace{\left(p_i - x_i^{(t-1)}\right)}_{\text{cognitive}} + c_2 U(0,1) \odot \underbrace{\left(p_g - x_i^{(t-1)}\right)}_{\text{social}}$$

N particles. $p_1, \ldots, p_N$ best positions of each particle.

p_g - best position of particles in neighbourhood

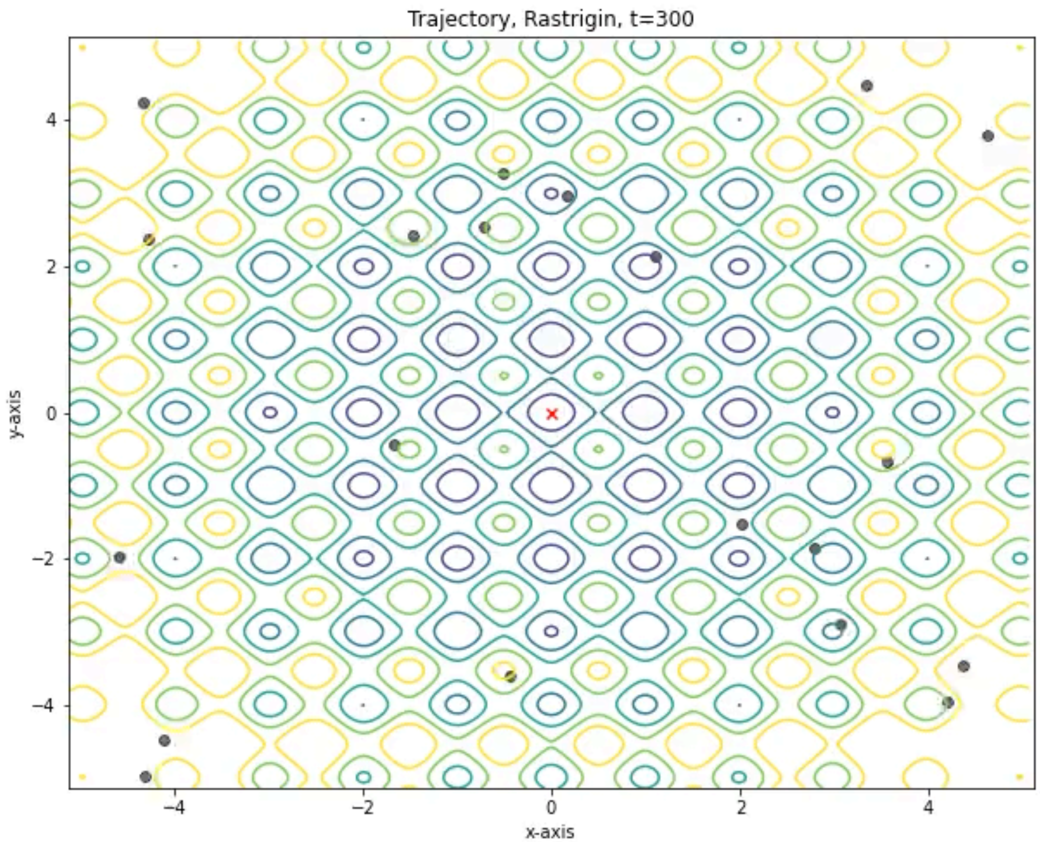
c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

```
x = rand(N, d)      # positions
v = rand(N, d)      # velocities
p = rand(N, d)      # previous best position
pbest = infinity(N) # best function value
g = 0               # index of best in neighborhood

# Run some amount of iterations
for t in range(iter):

    # Update all particles
    for i in range(N):

        # Check if F at current x is better than previous and update pbest, p.
        if F(x[i]) < pbest[i]:
            pbest[i] = F(x[i])
            p[i] = x[i]

        # Get neighbors and get index of best performing particle
        neighbors = get_neighbors(i)
        g = best_performer(neighbors)

        # Update velocity and position
        v[i] += mult_elem(c1*rand(d), (p[i] - x[i])) +
                mult_elem(c2*rand(d), (p[g] - x[i]))
        x[i] += v[i]
```

Extension: Particle swarm optimisation

Jonas Olsson



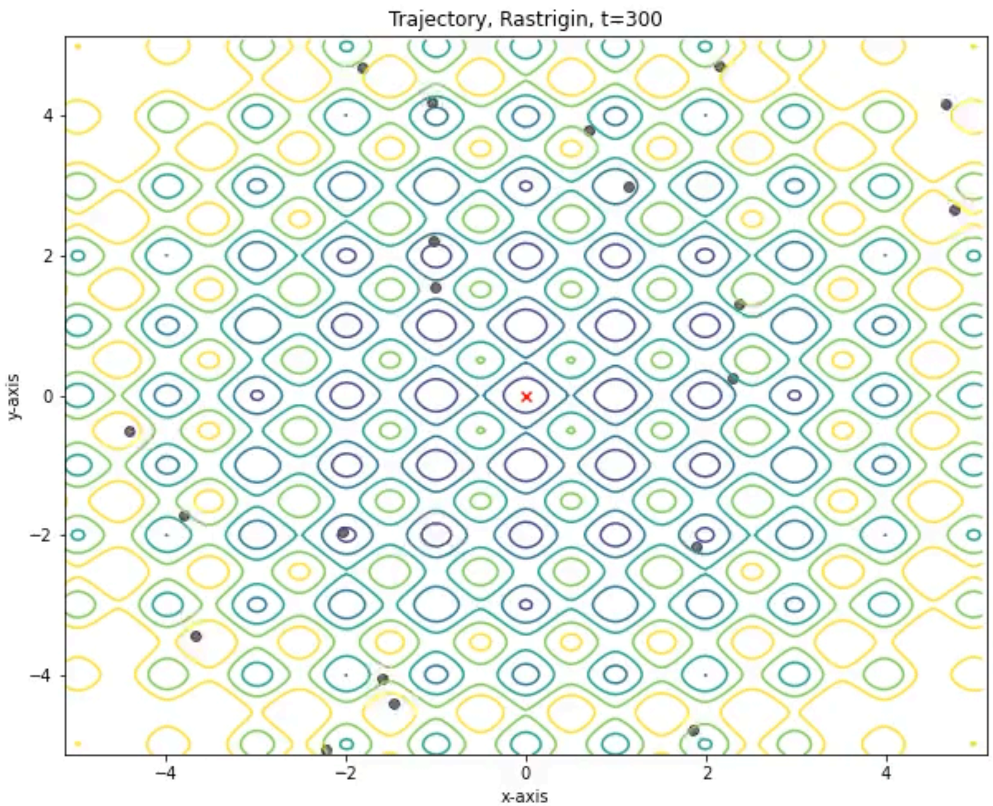
$c_1 = 1.49618$, $c_2 = 1.49618$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

Extension: Particle swarm optimisation

Jonas Olsson



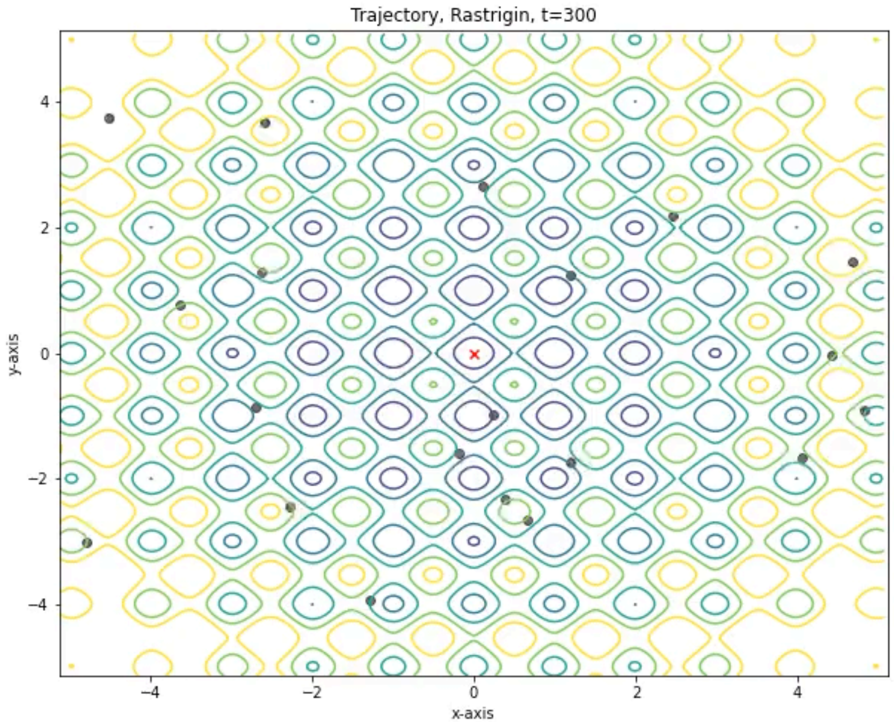
$c_1 = 0, c_2 = 1.49618$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

Extension: Particle swarm optimisation

Jonas Olsson



$c_1 = 1.49618, c_2 = 1.49618, w = 0.7298$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

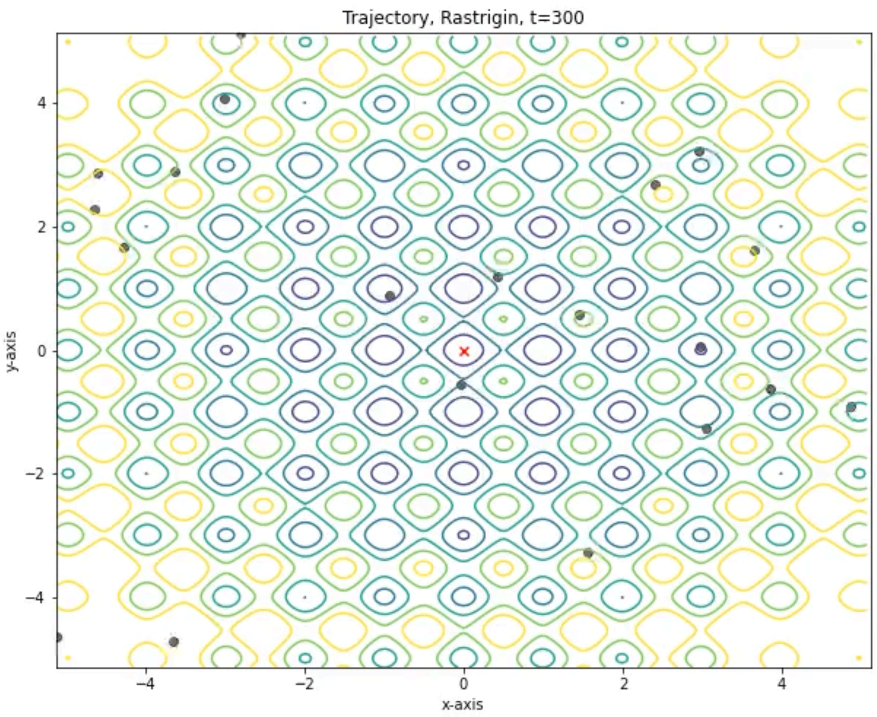
w inertia/constriction/viscosity,

large w - viscosity low, particles move easily - favours global min

Small w - viscosity high particles move slower - favours local min

Extension: Particle swarm optimisation

Jonas Olsson



$c_1 = 0,\ c_2 = 1.49618,\ w = 0.7298$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

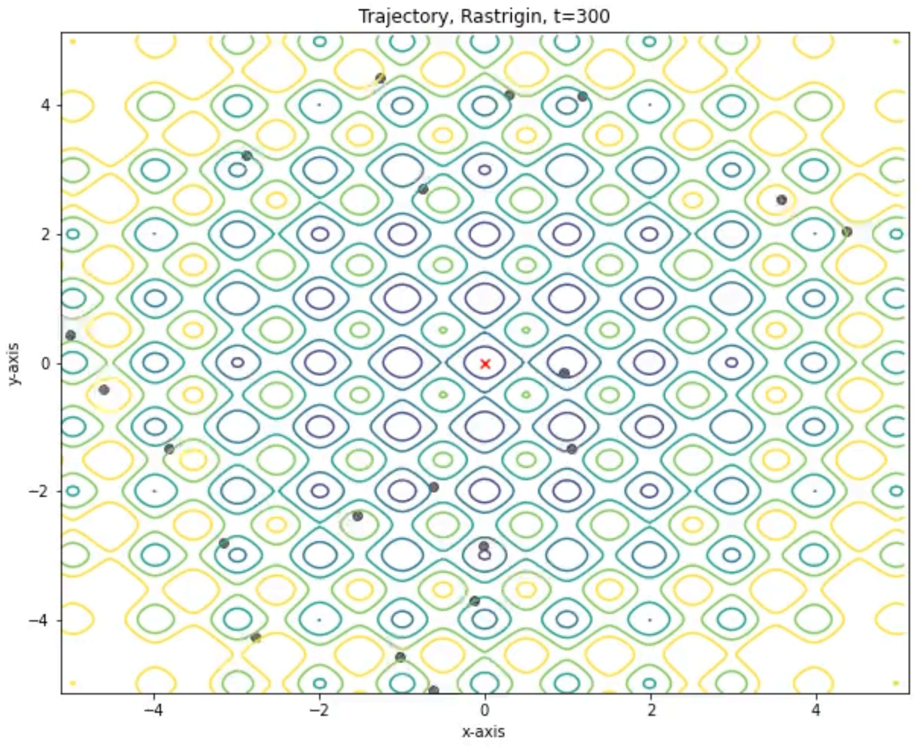
w inertia/constriction/viscosity,

large w - viscosity low, particles move easily - favours global min

Small w - viscosity high particles move slower - favours local min

Extension: Particle swarm optimisation

Jonas Olsson



$c_1 = 1.49618$, $c_2 = 0$, $w = 0.7298$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

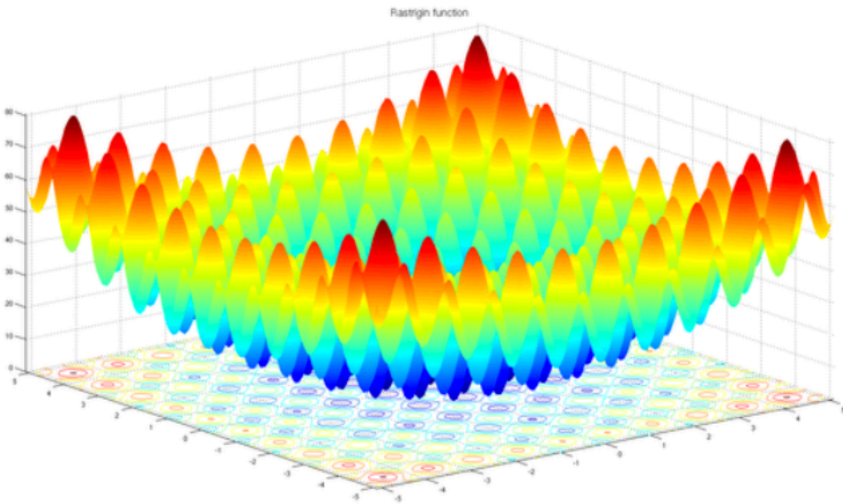
w inertia/constriction/viscosity,

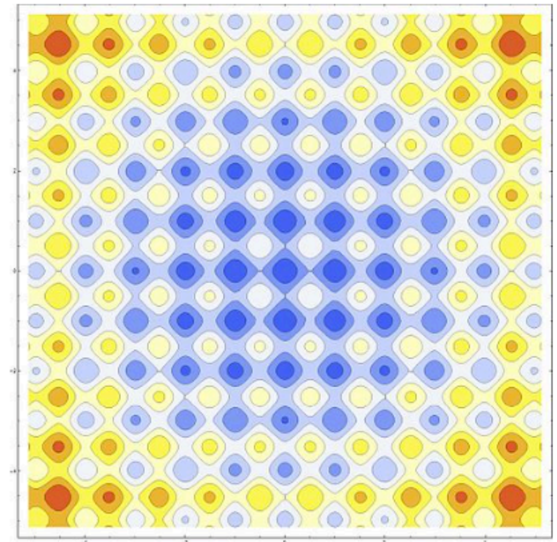
large w - viscosity low, particles move easily - favours global min

Small w - viscosity high particles move slower - favours local min

Prc

future





$$u_i(t+1) = a u_i(t) + (1\text{-}a)\, s_i(t) + e_i(t)$$

future velocity

current velocity

Direction to most neighbours

stochastic effect

$$s_i(t) = \frac{1}{|R_i|} \sum_{j \in R_i} \mathrm{sign}\Big(x_i(t) - x_j(t) \Big)$$

$e_i(t)$ *is a* random number selected uniformly at random from a range $[-\eta/2, \eta/2]$