

# Flocking and inverse design

these slides:

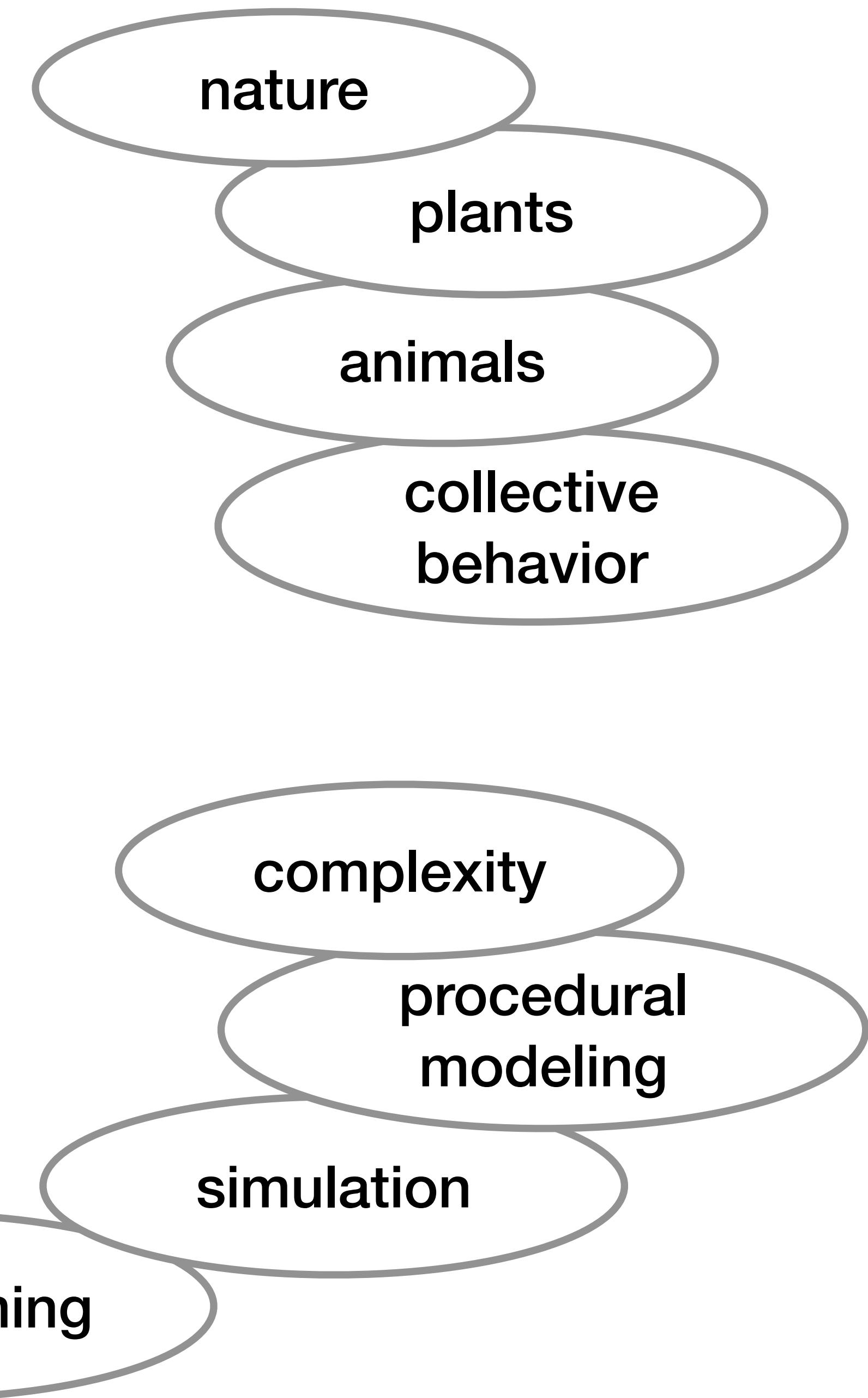
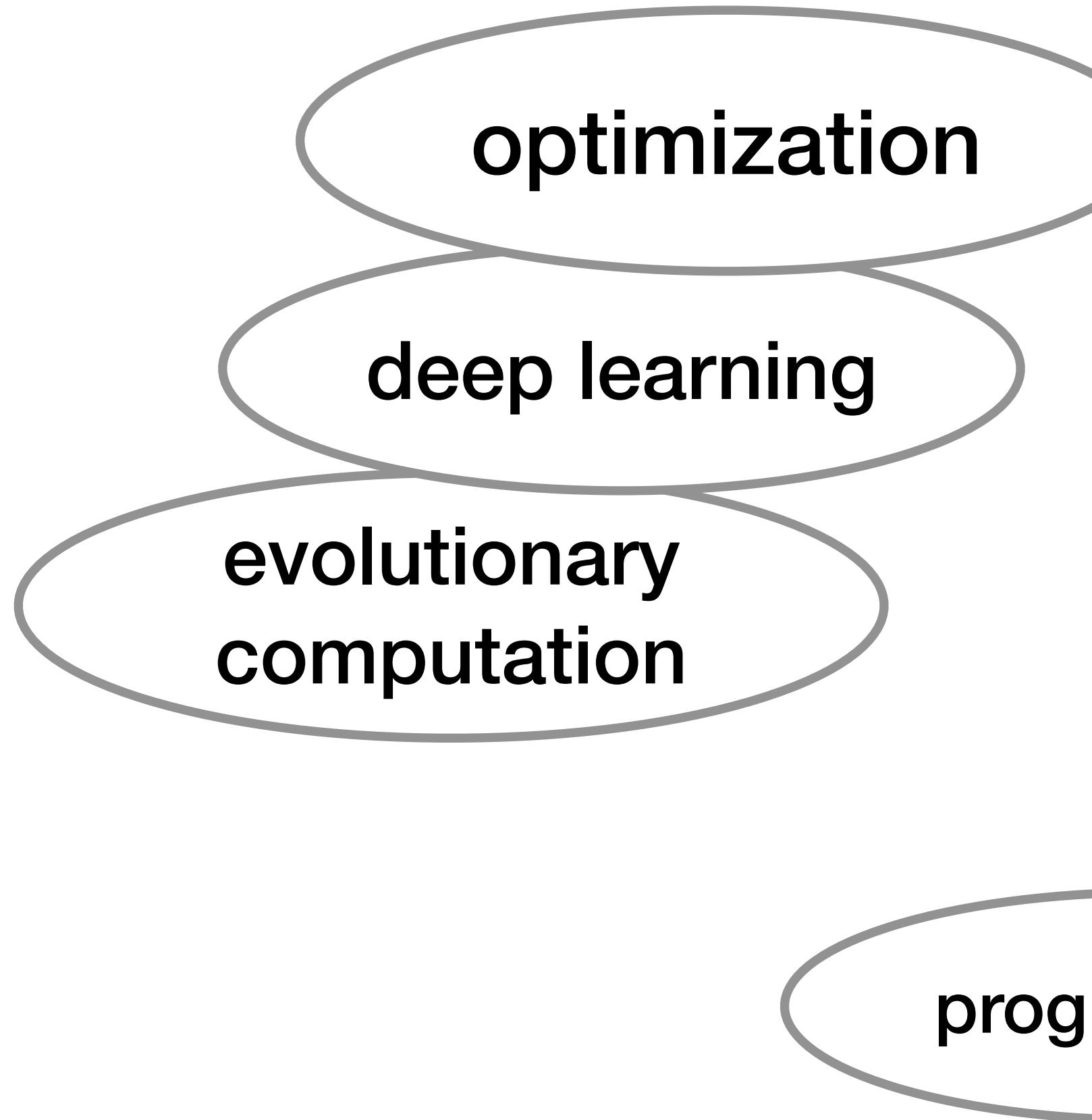
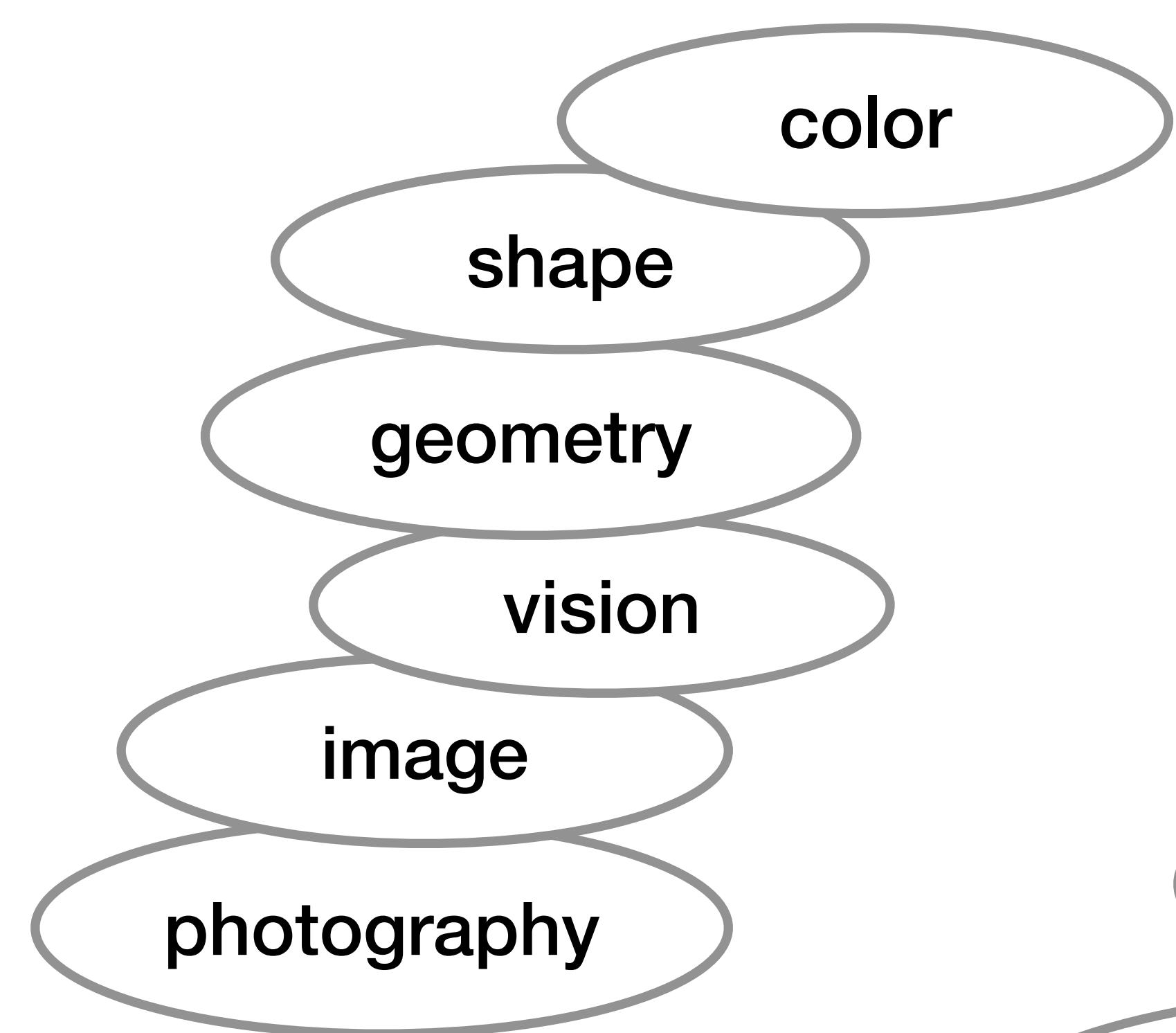


Craig Reynolds – Game AI – UCSC – August 6, 2025

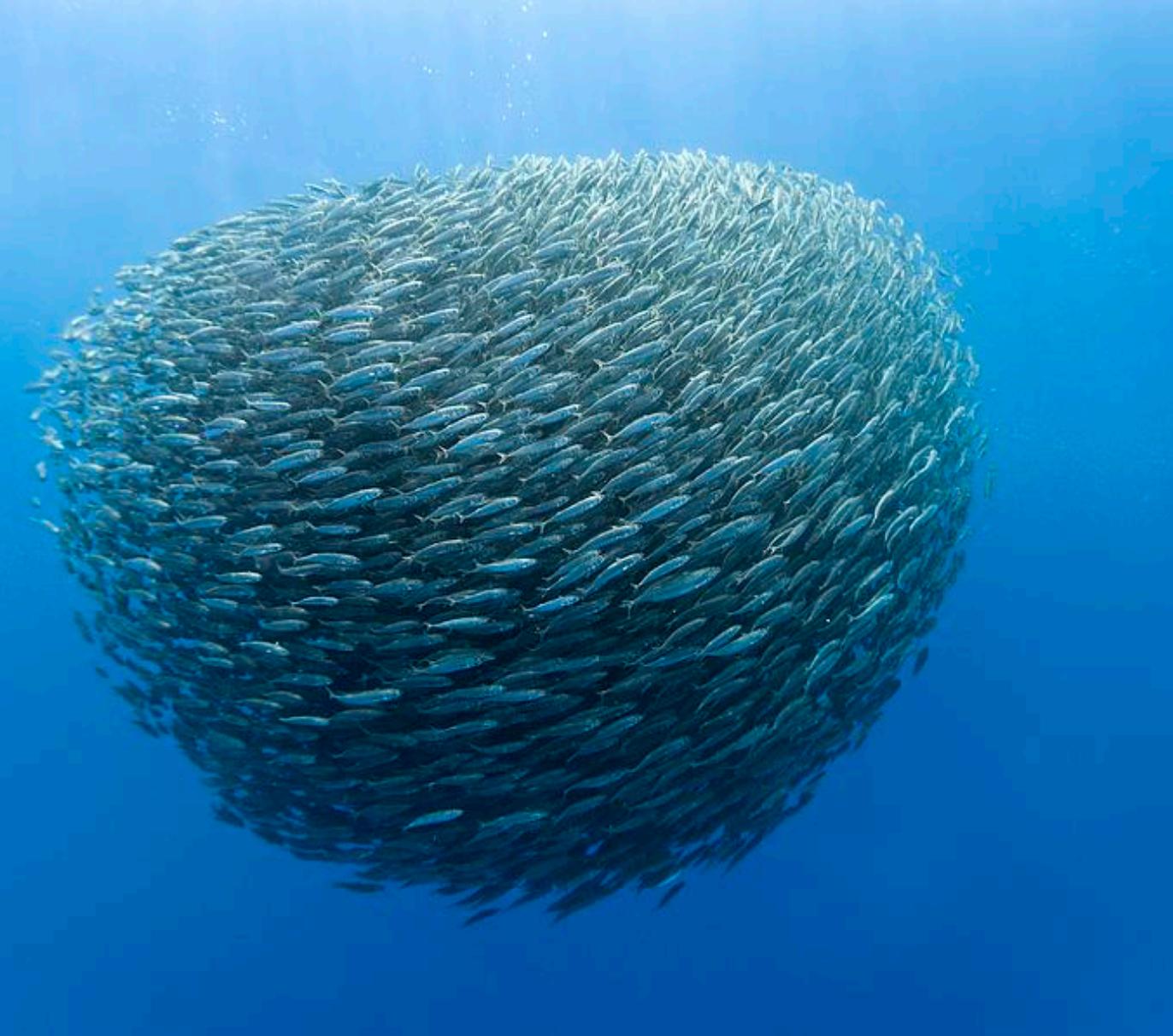
# Background

I am particularly interested  
in procedural models of  
complex natural systems.

- **MIT**: CS, procedural animation (BS 1975, MS 1978)
- **triple-I**: *Juggler*, *Looker*, *TRON*
- **Symbolics**: digital content creation tools, boids, *Breaking the Ice*
- **Electronic Arts**: “non-player characters” for games, steering behaviors
- **SGI Silicon Studio**: DCC tools for games, steering behaviors
- **PlayStation US R&D**: PScrowd, evolution of camouflage
- **UC Santa Cruz (Playable Media)**: human behaviors for “serious games”
- **RightHook**: vehicle/pedestrian behaviors for virtual testing of self-driving
- “**unaffiliated researcher**” retired since 2020



# A childhood fascination with natural complexity





bird shaped flock on YouTube

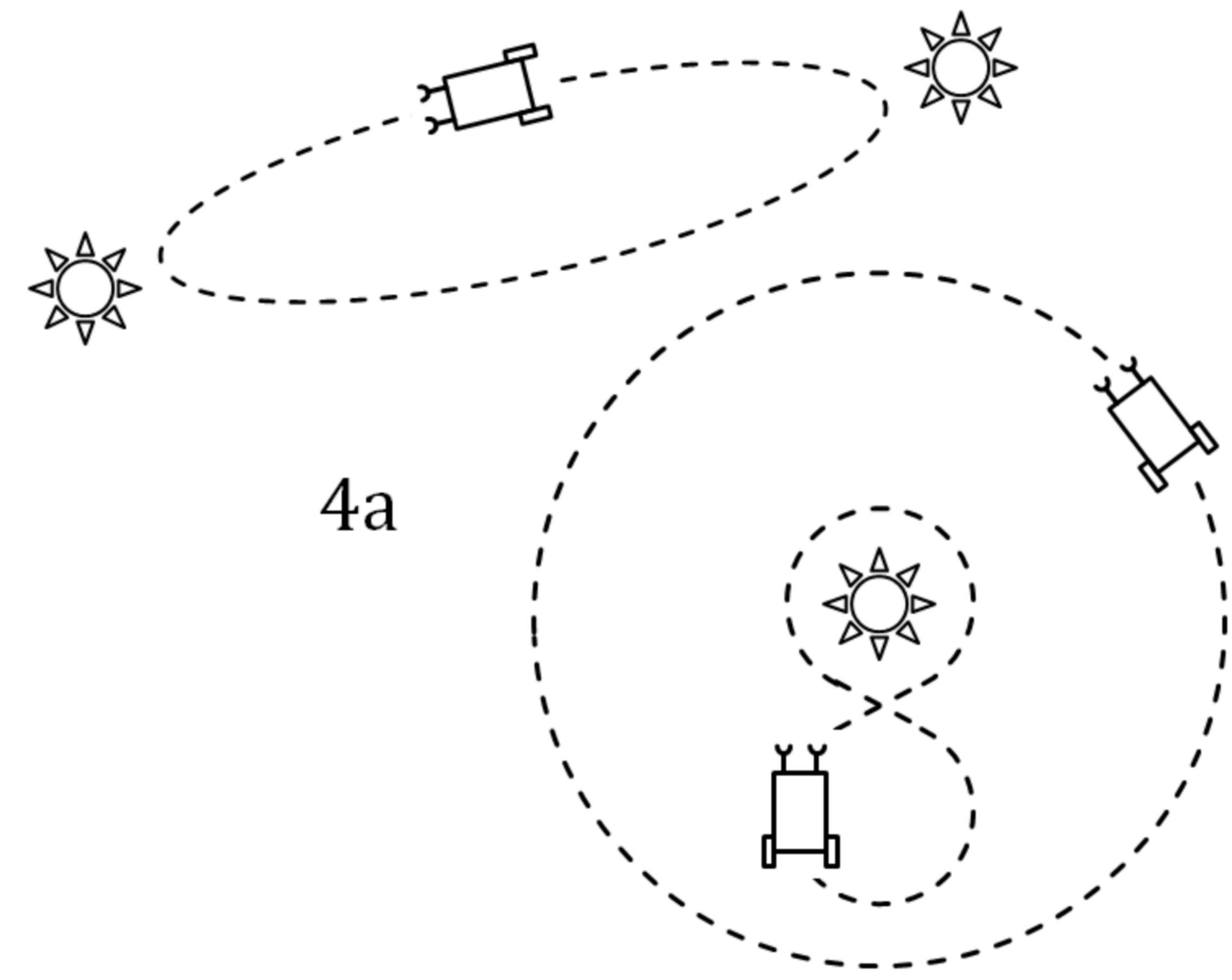
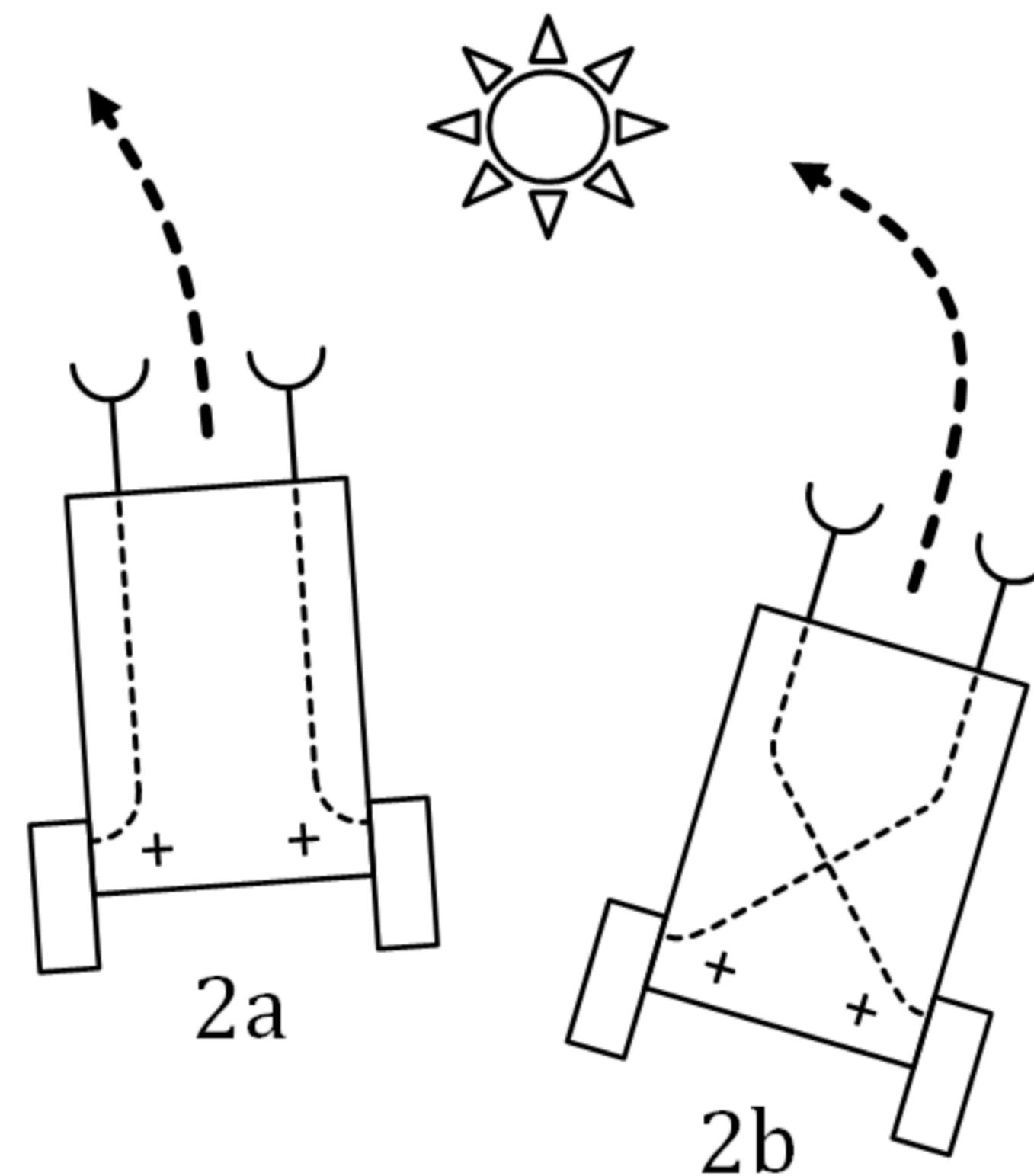
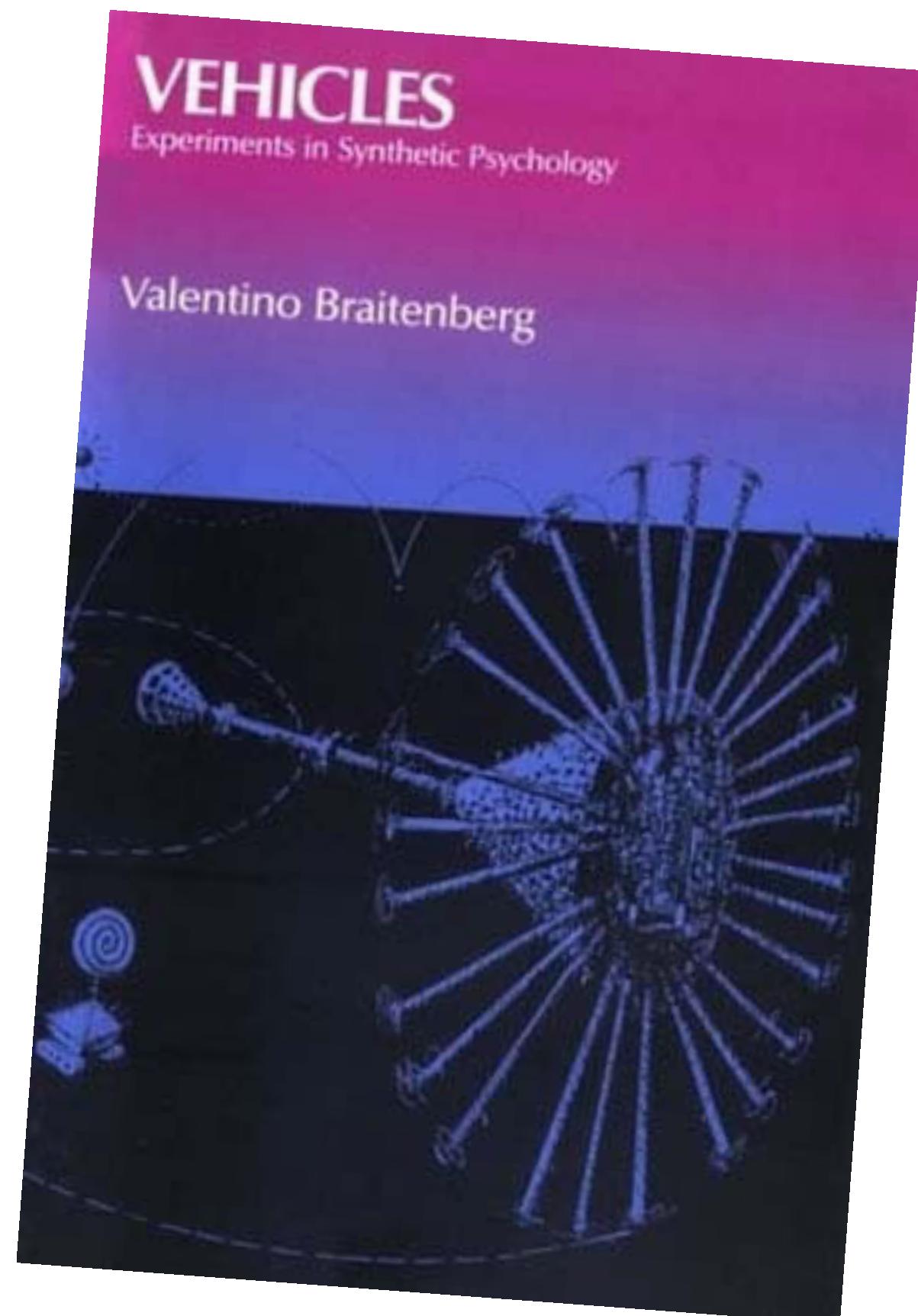
Early boids  
1986-1987

What is it like to be a bird in a flock?



**Key idea:** a shift in perspective  
from an external view of a complex system  
to an internal local view of a few neighbors.

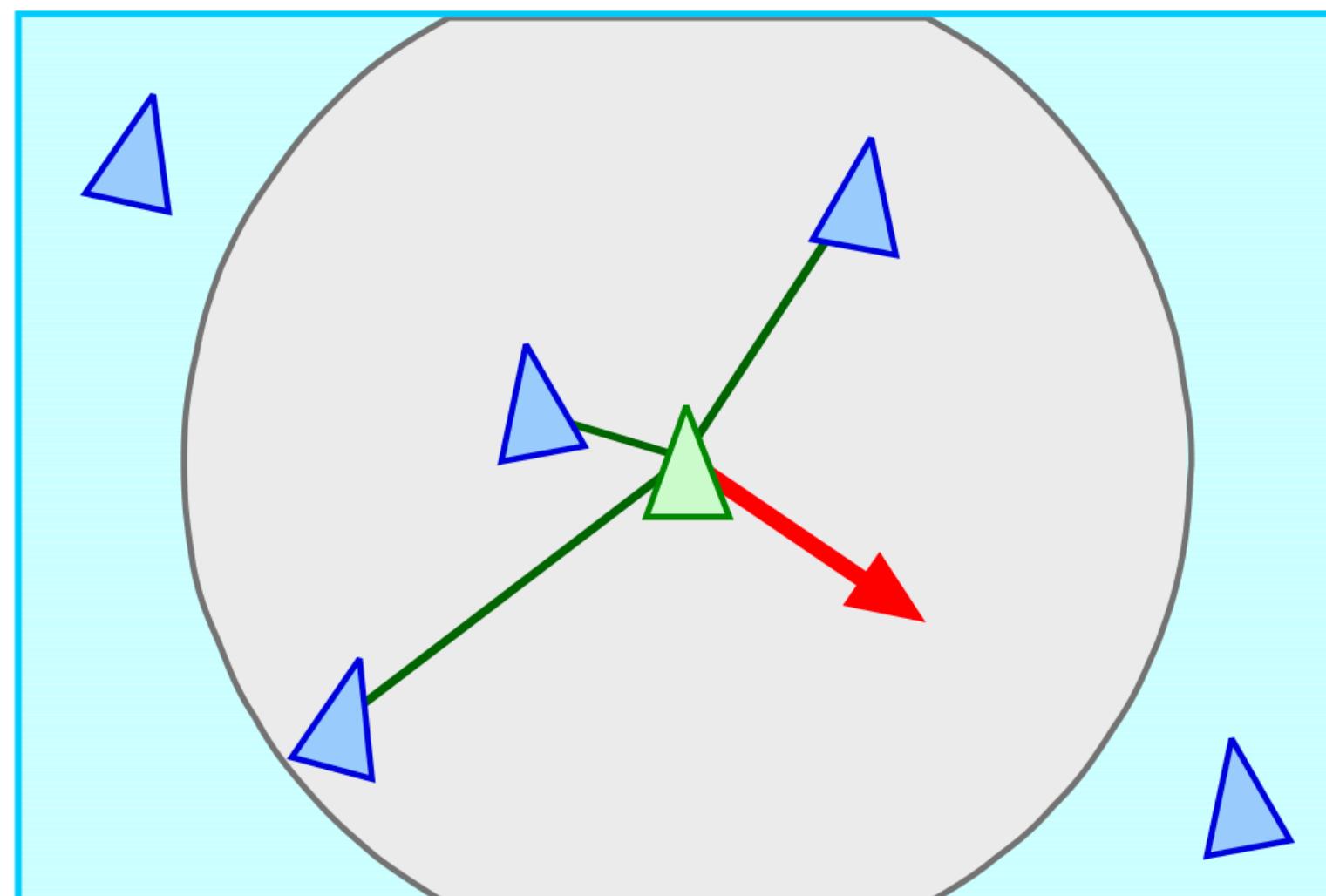
Could the behavior of flocking birds  
be simulated in software?



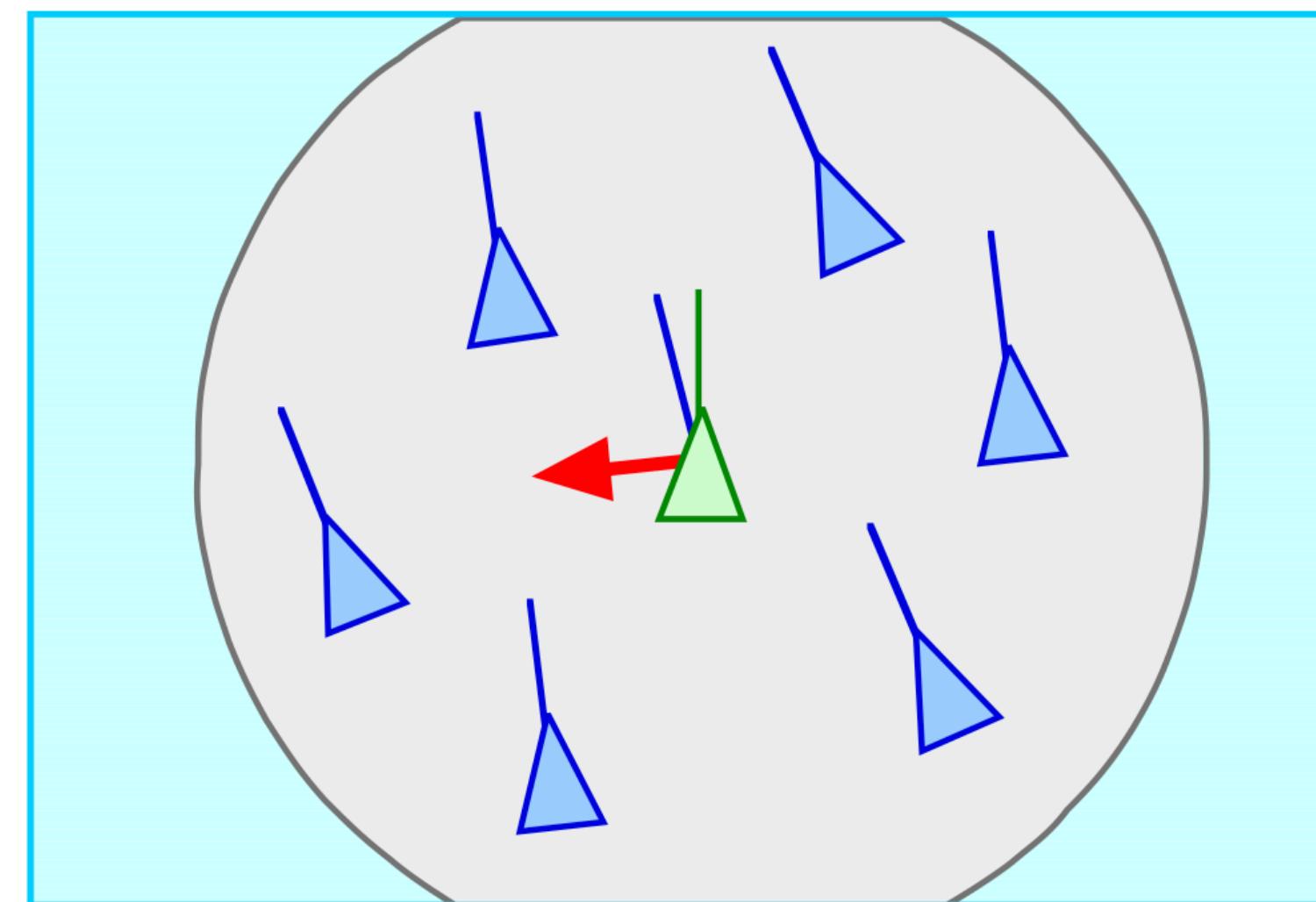


# Three components of flocking

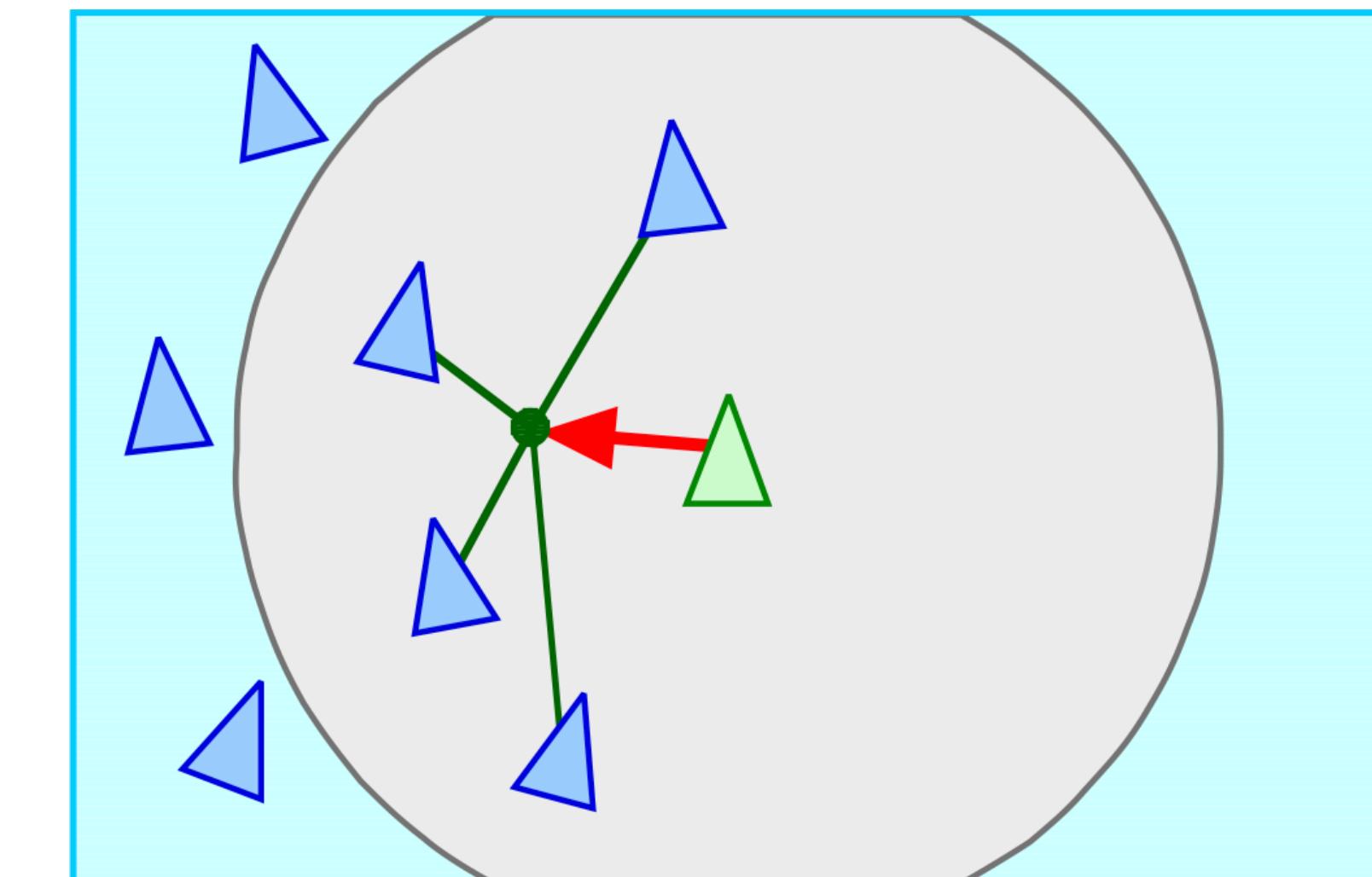
Seemed necessary, but were they sufficient?



Separation

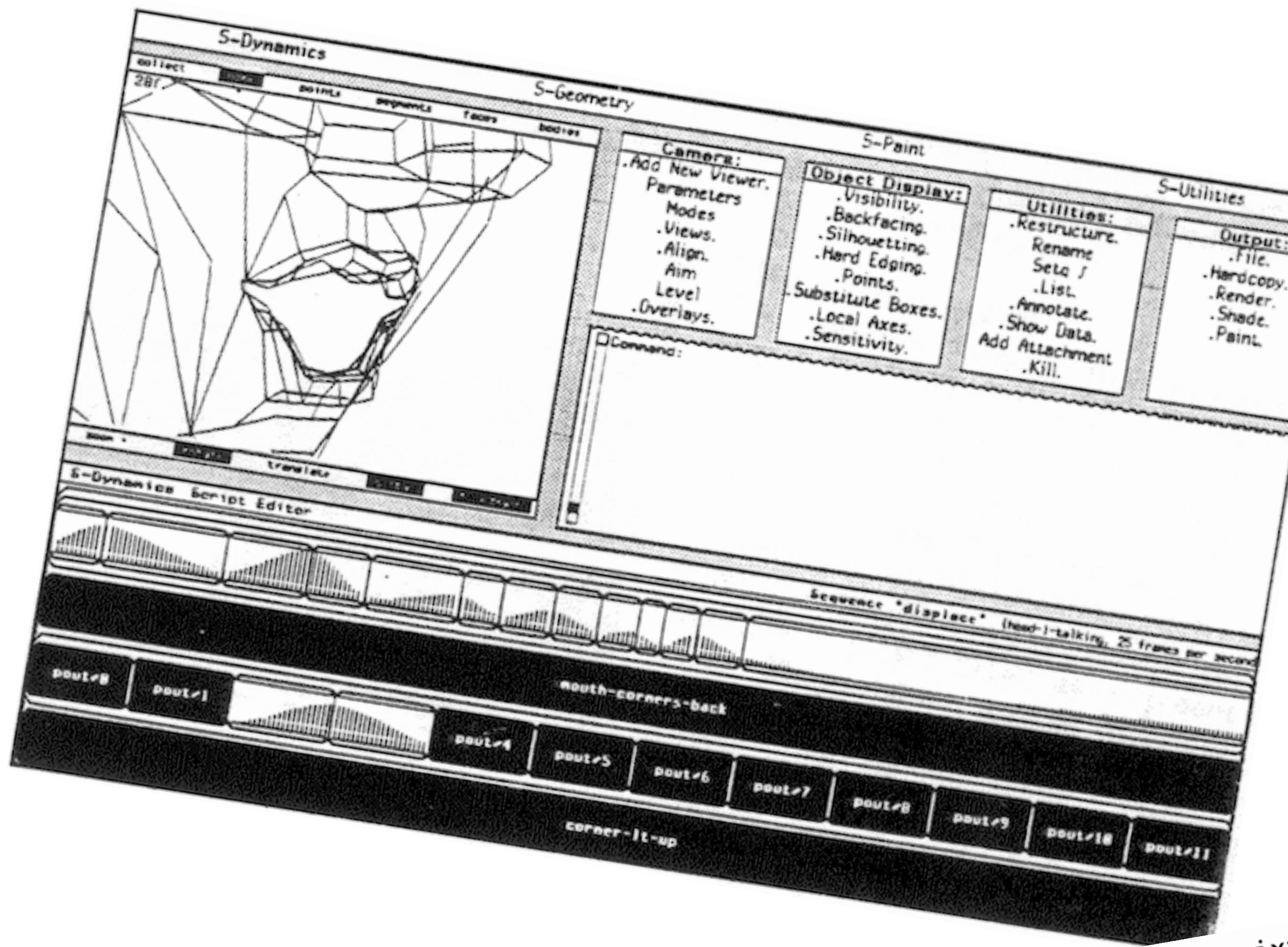


Alignment



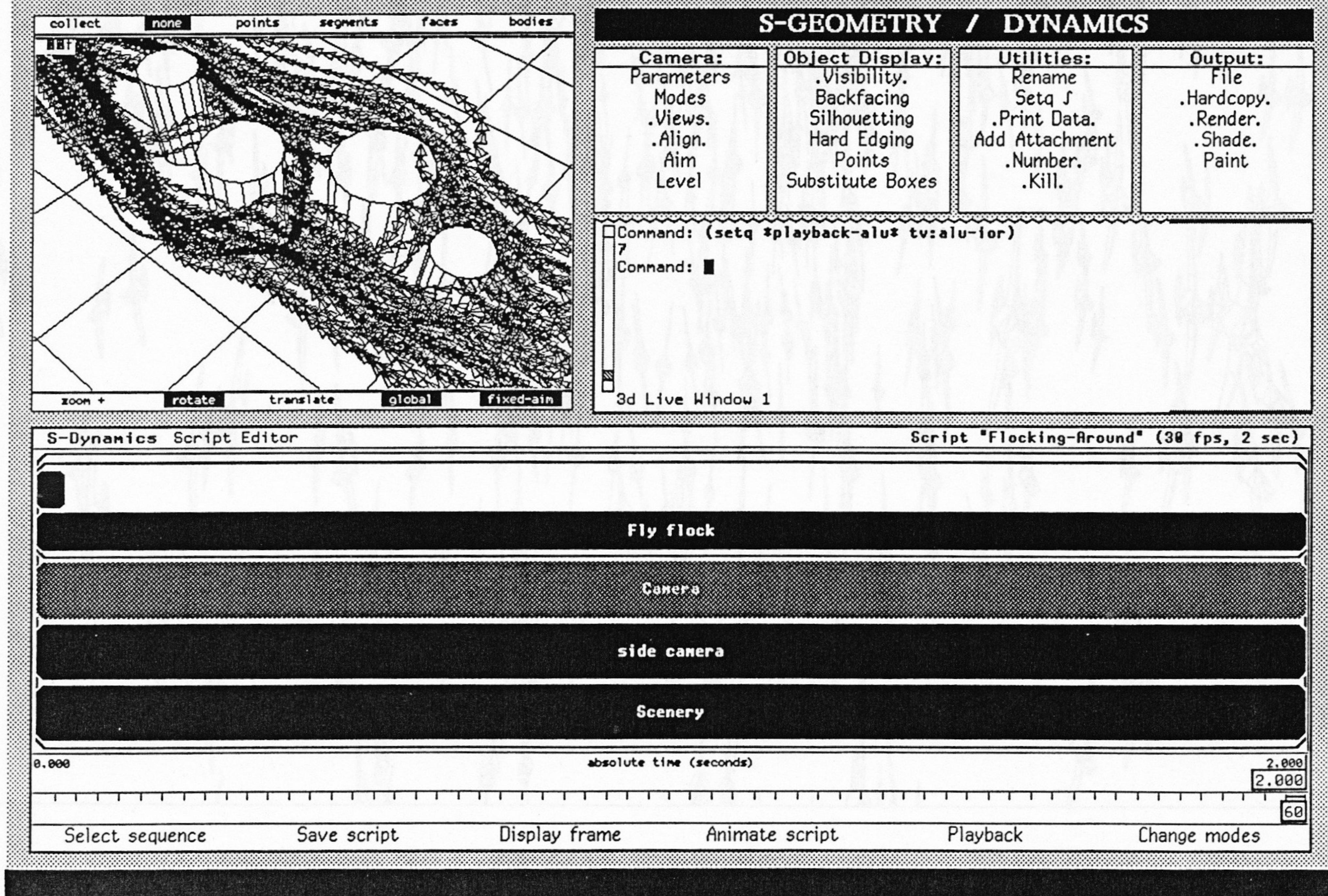
Cohesion

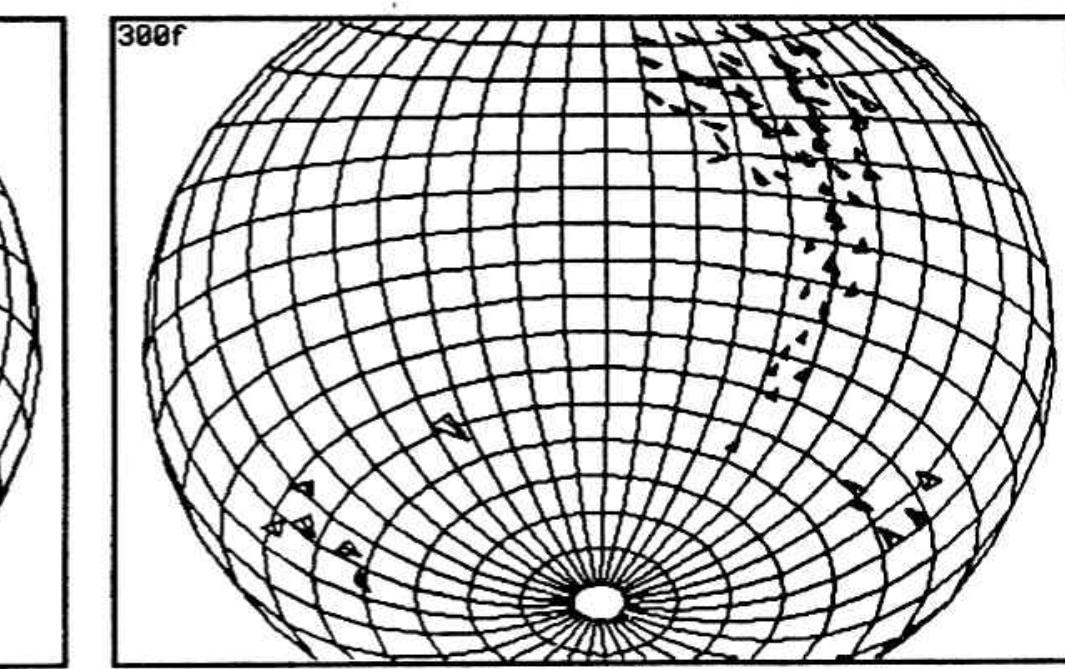
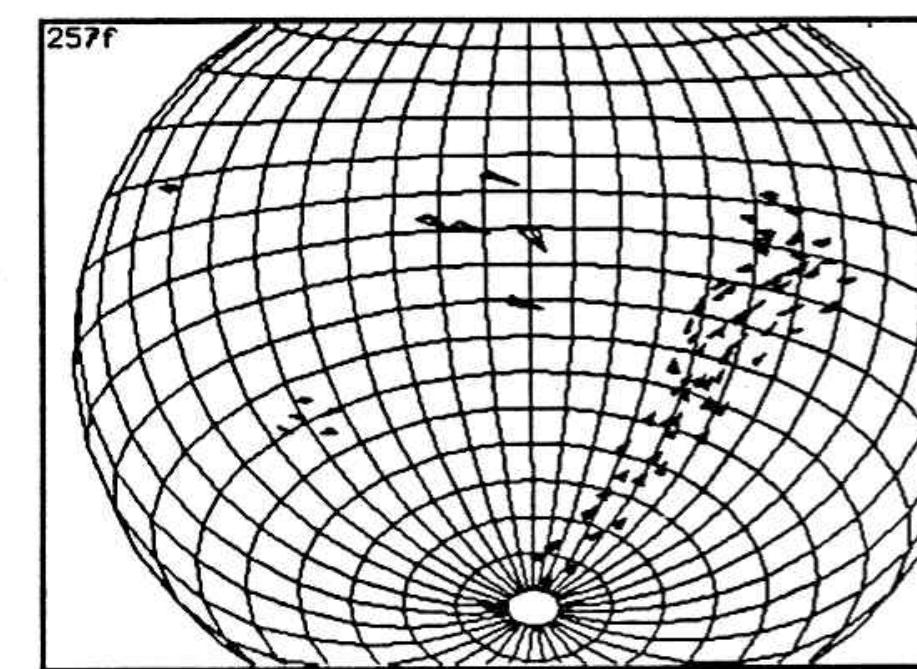
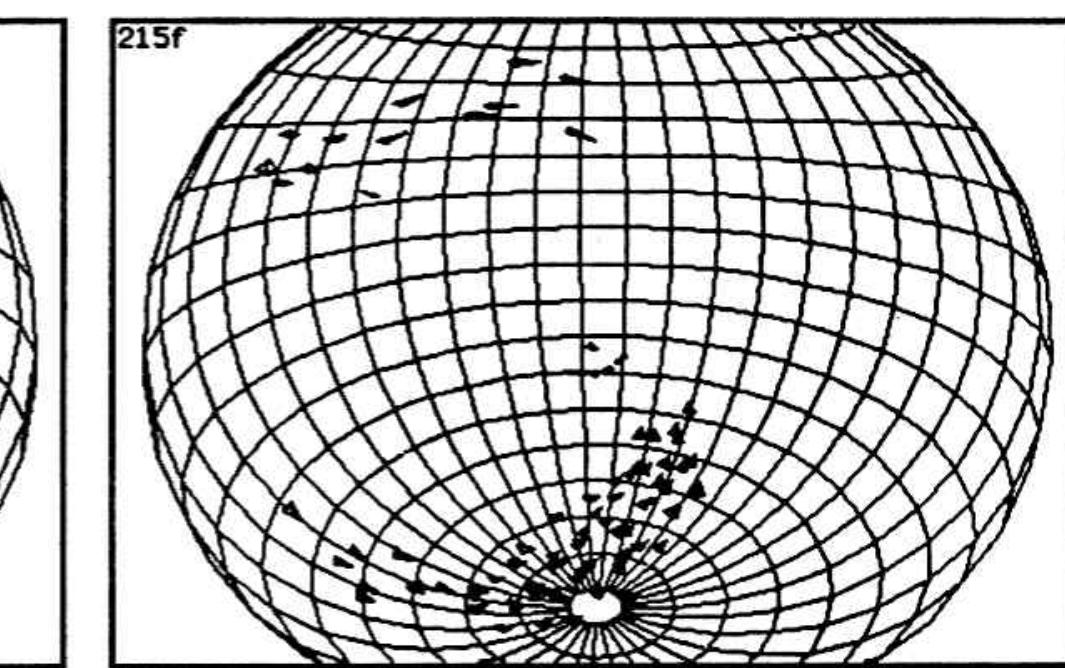
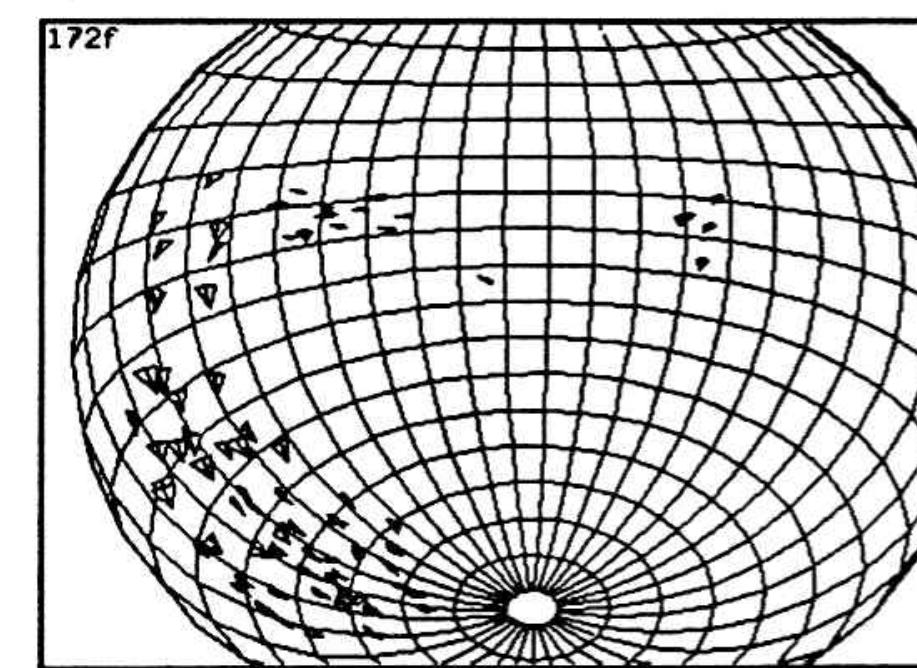
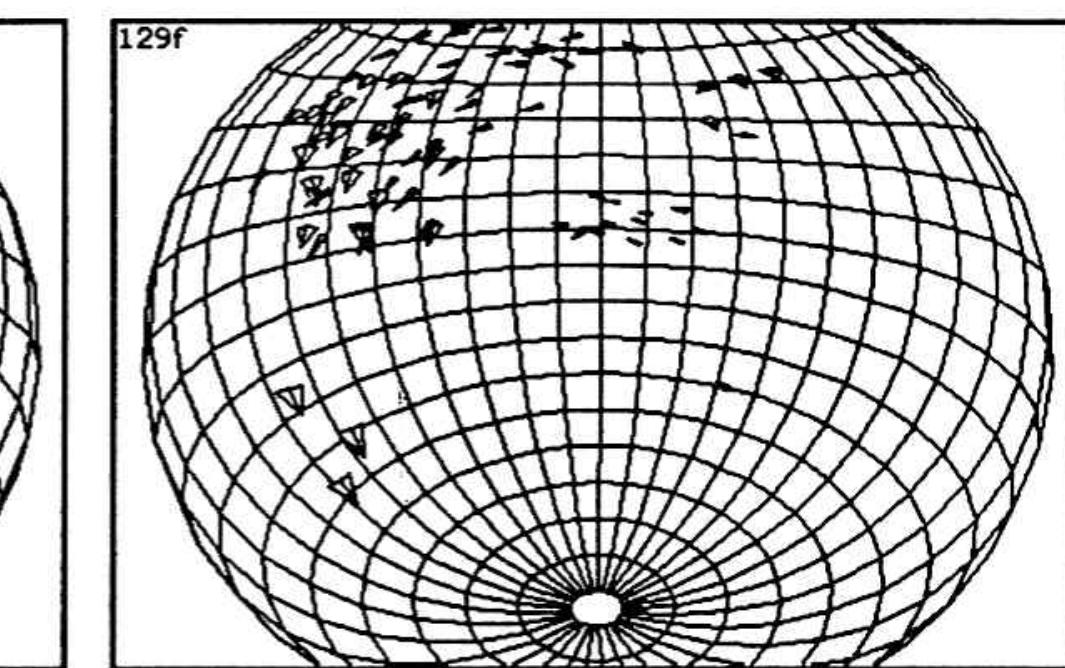
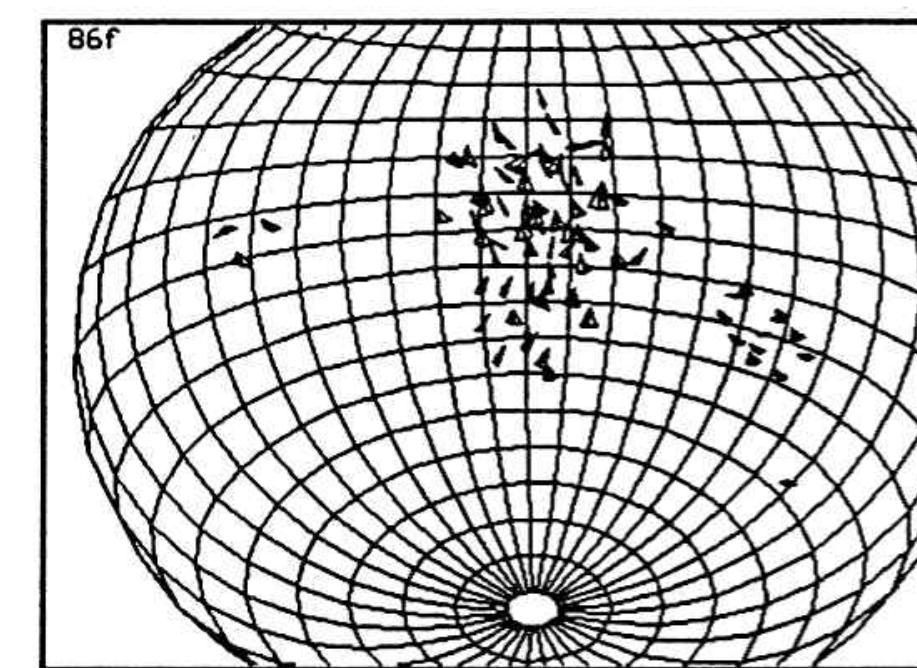
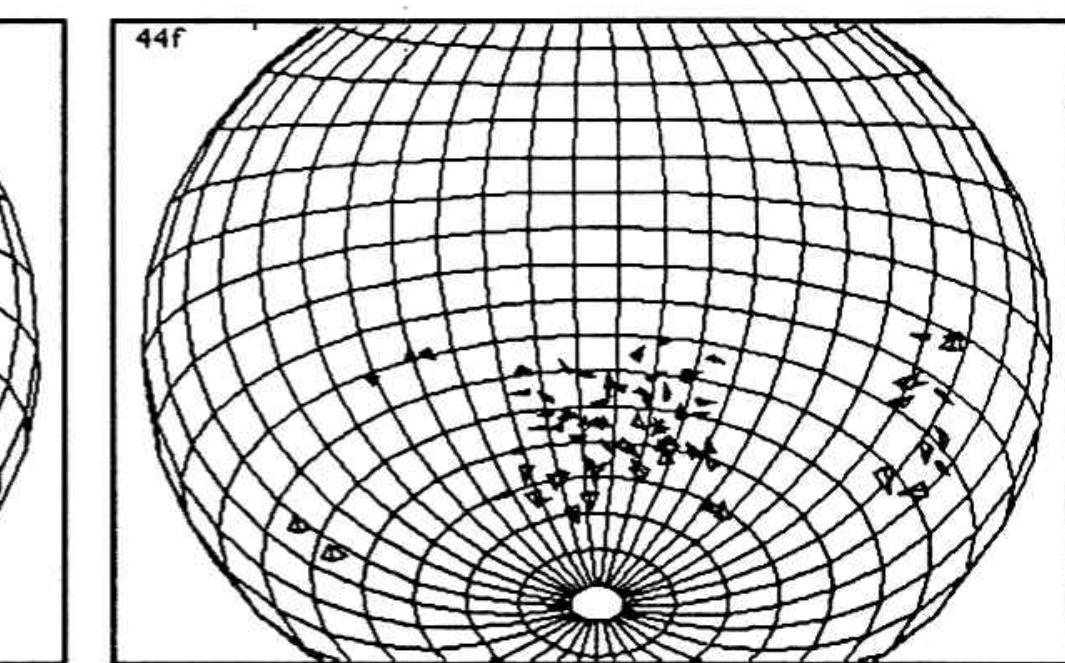
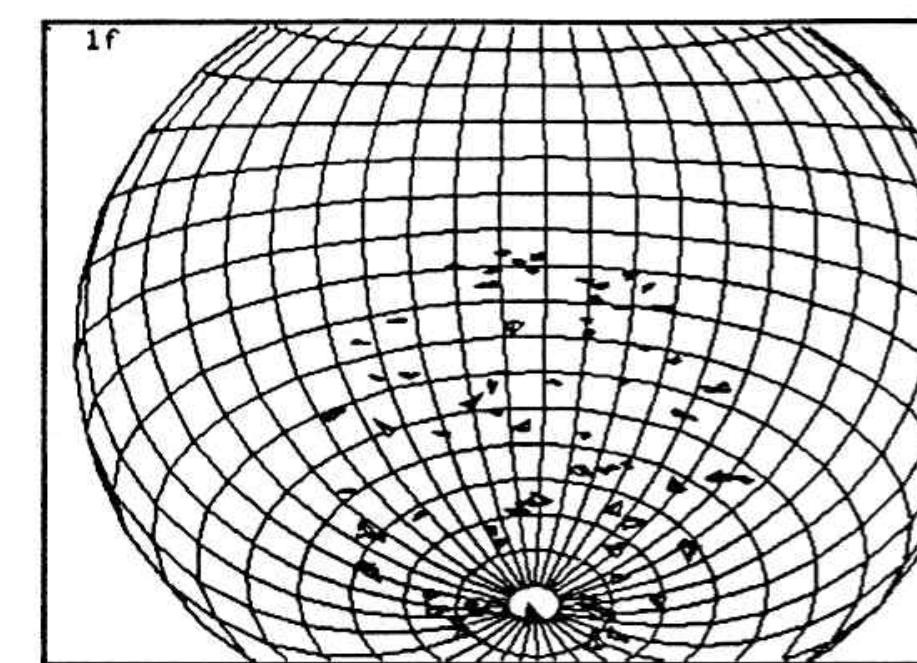
**SIGGRAPH 1986:**  
writing tutorial notes:  
**“...it would be easy...”**

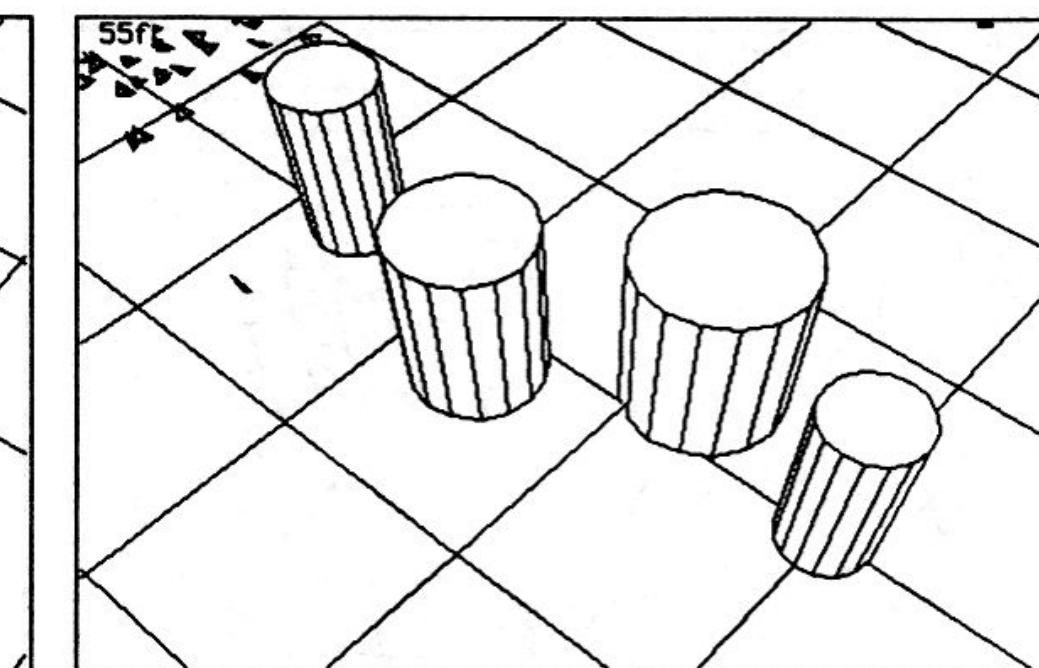
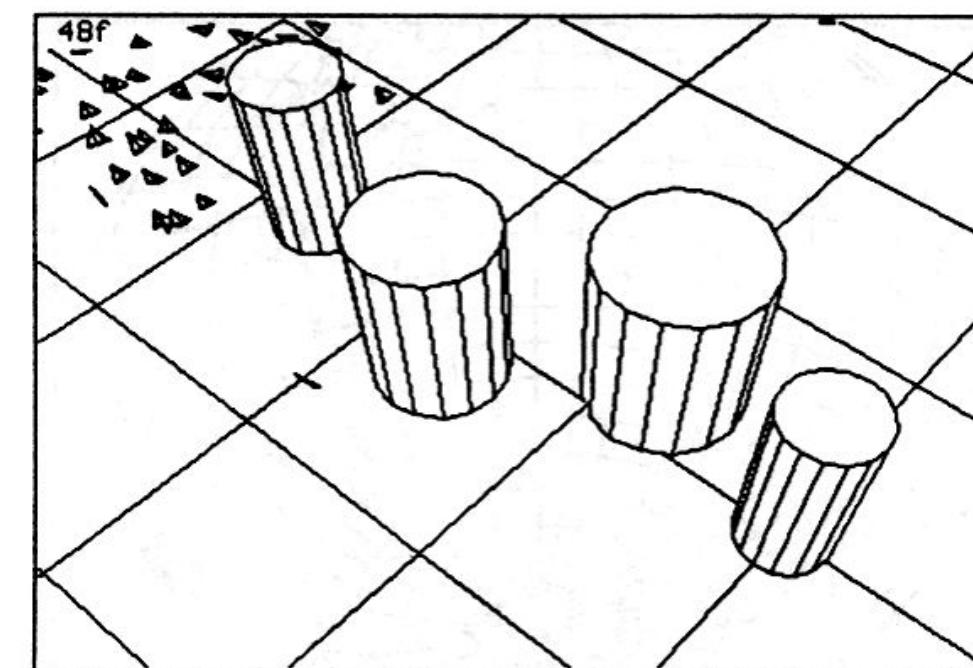
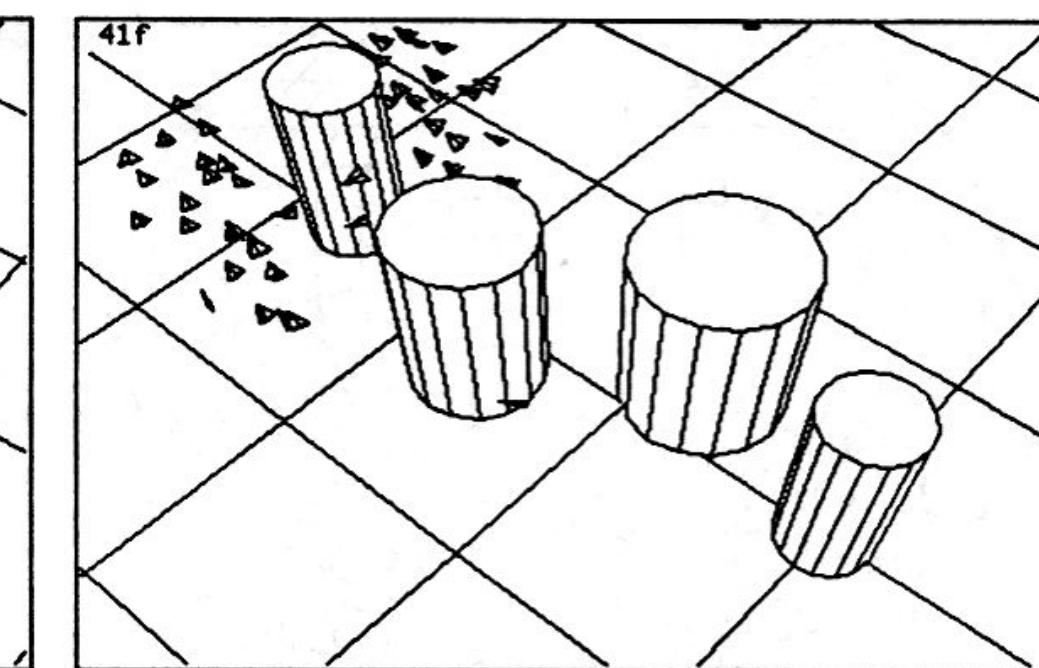
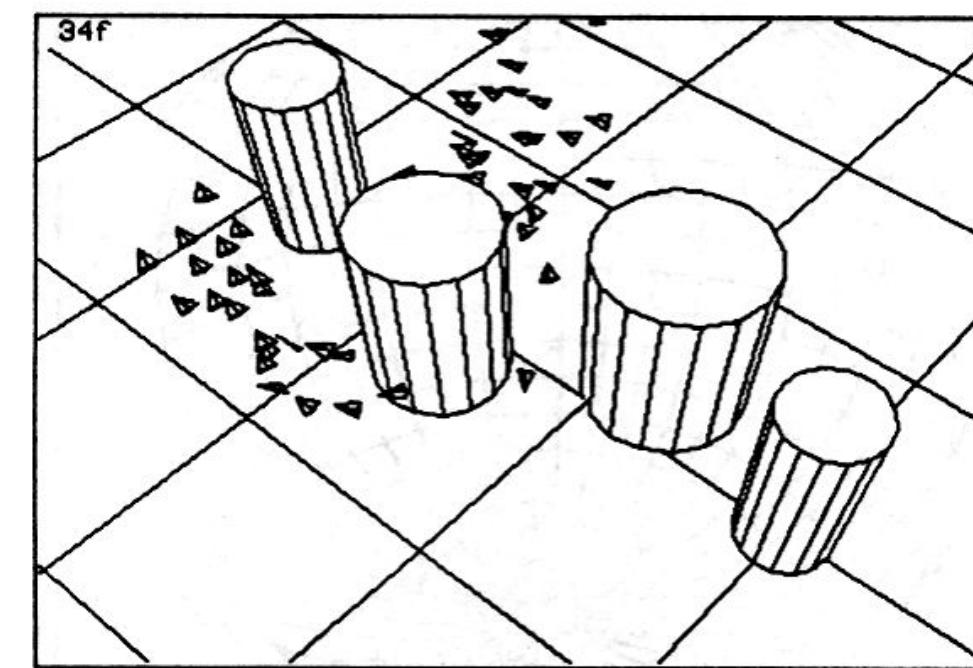
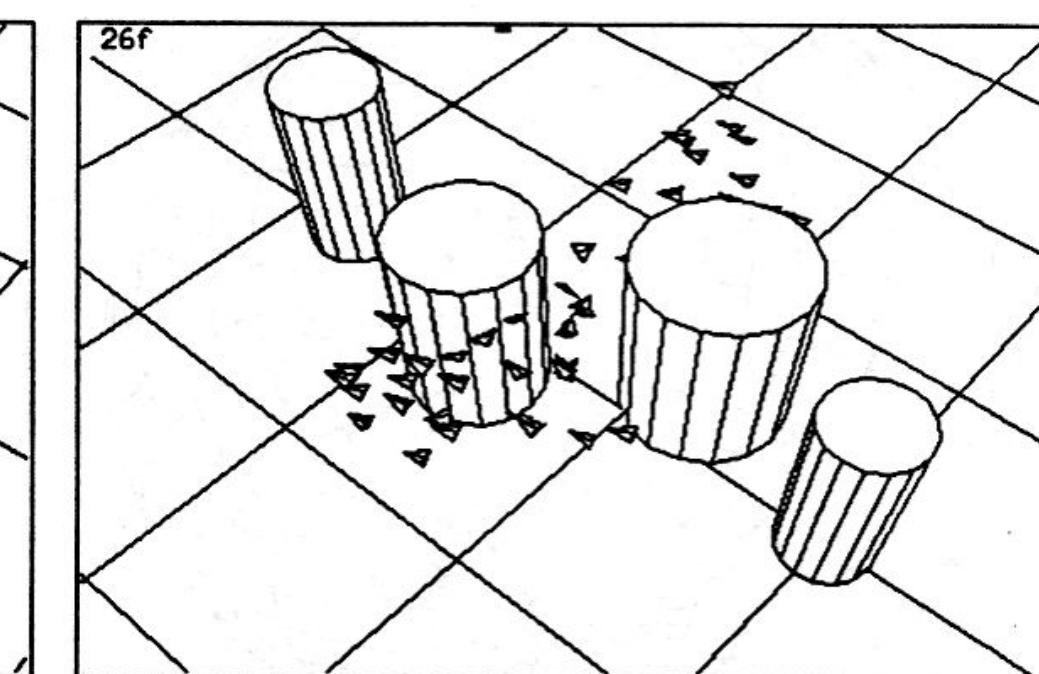
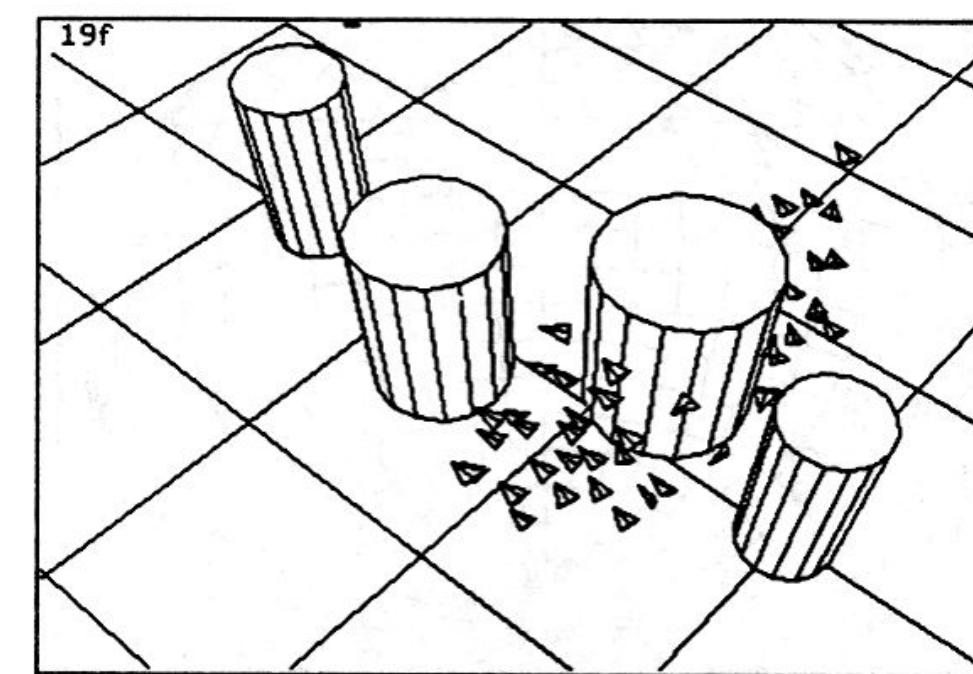
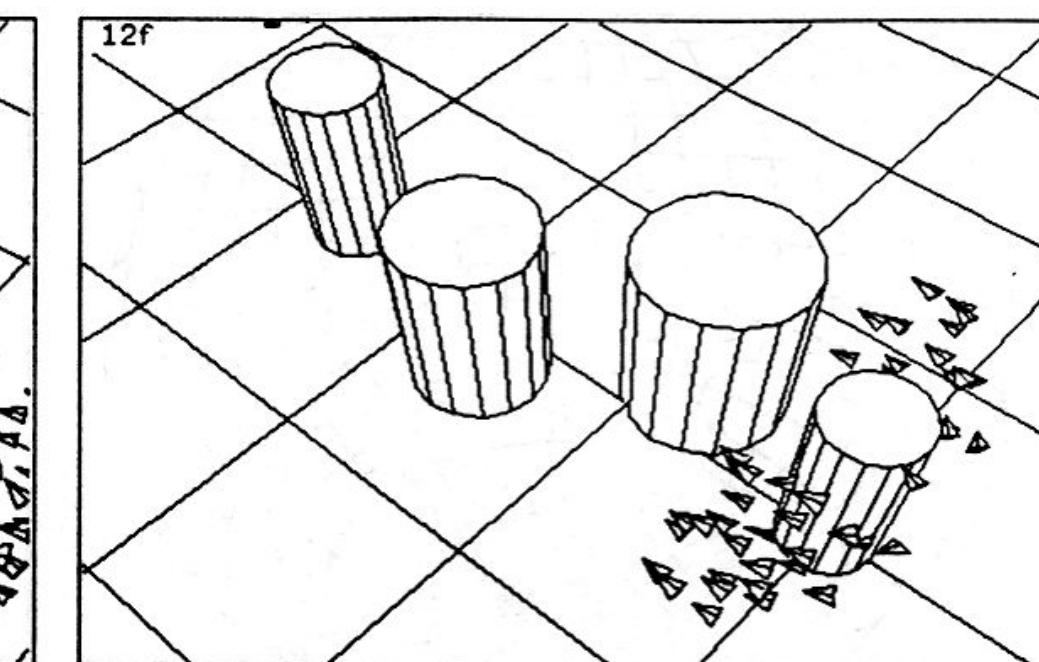
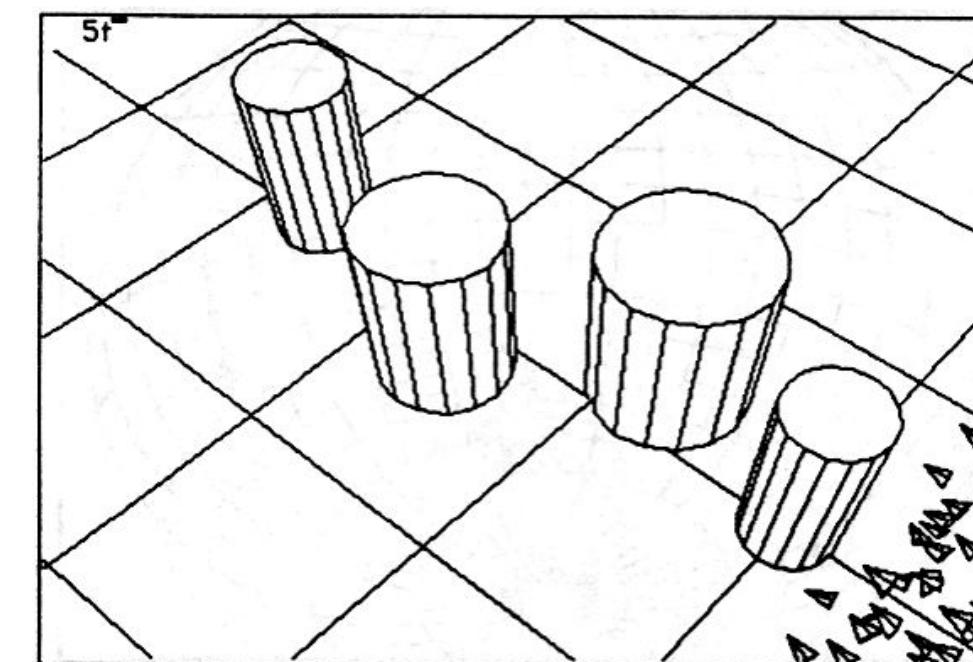




```
(defmethod (:steer flight-mixin) (old-velocity new-velocity)
  (vlet* ((new-adjusted-velocity (send self :constrain-steering new-velocity))
          (new-adj-acceleration (3d-vector-sub new-adjusted-velocity old-velocity))
          (actual-steering-acceleration (truncate-magnitude new-adj-acceleration
                                                               (send self :max-acceleration)))
          (actual-new-velocity (3d-vector-add old-velocity actual-new-velocity)))
    (when (plusp (magnitude-squared actual-new-velocity))
      ;; PITCH and YAW the object (local X and Y rotations) to align with the new local velocity.
      (send self :align-to-local actual-new-velocity)
      ;; Adjust the roll for a "coordinated turn".
      (send self :bank actual-steering-acceleration))))
```



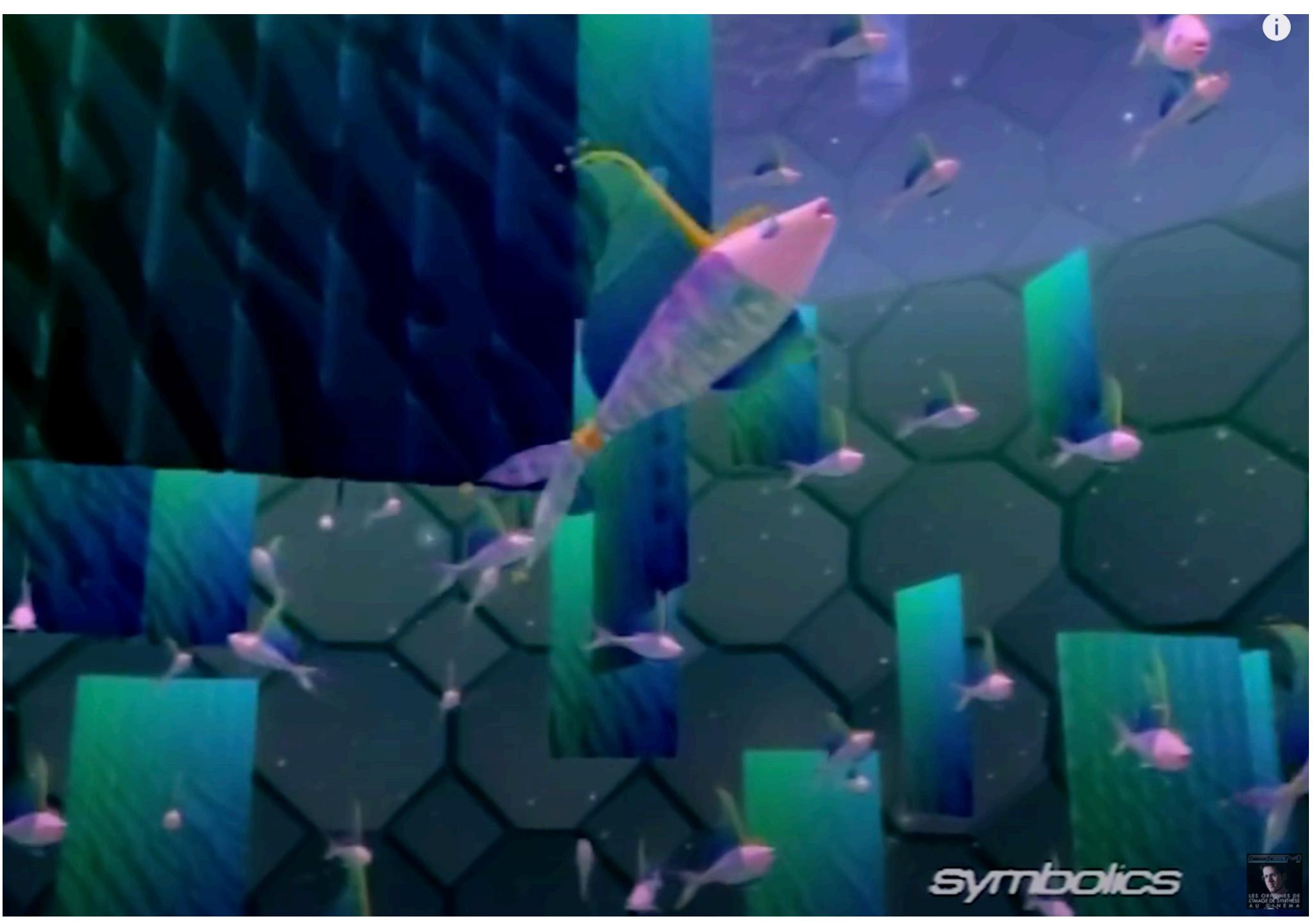
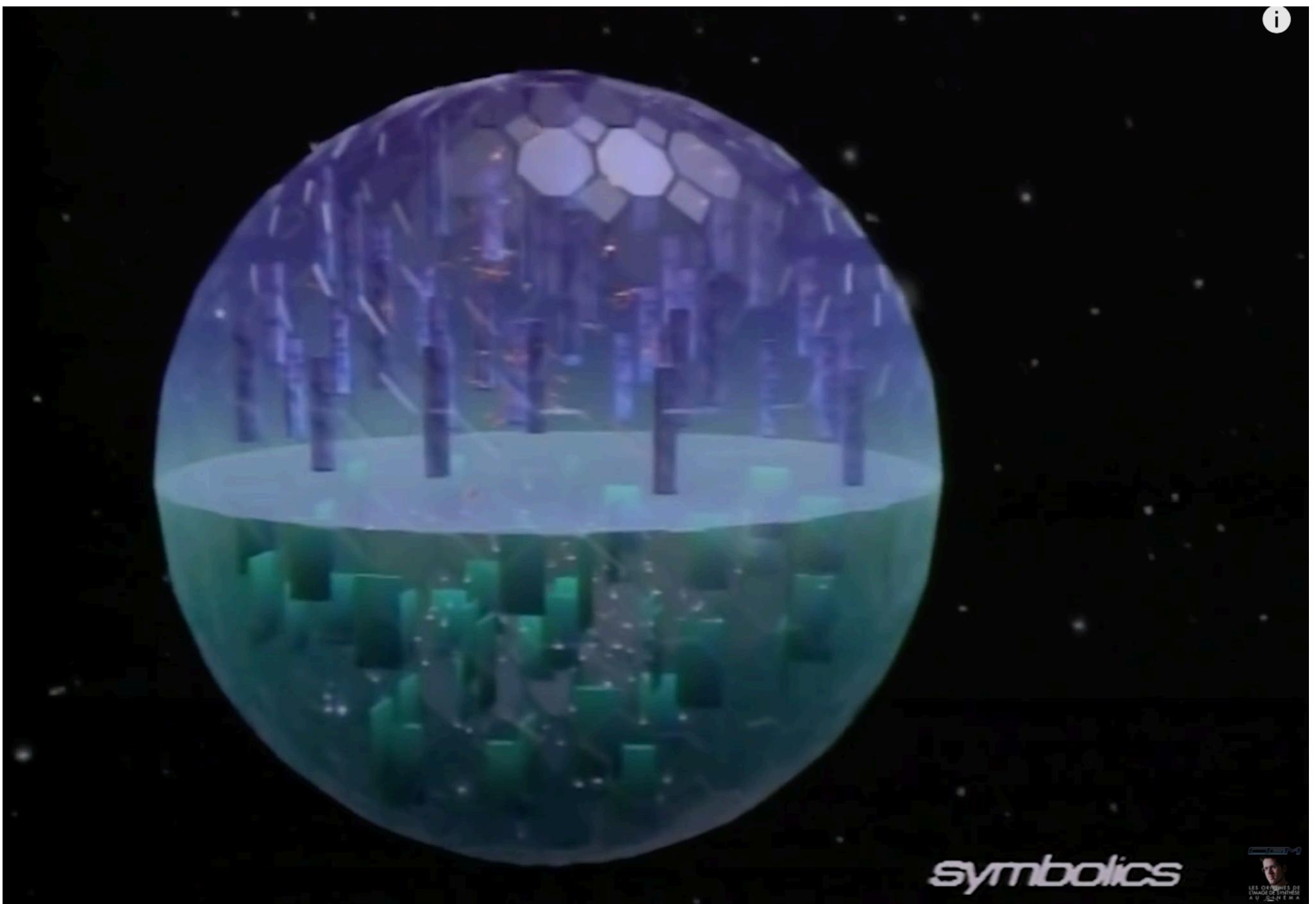


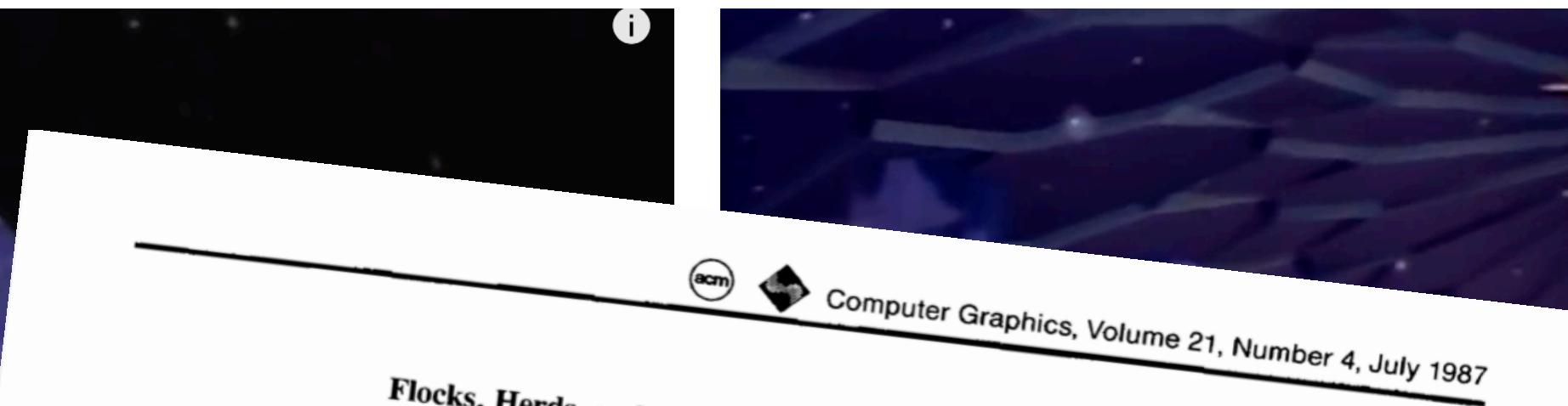
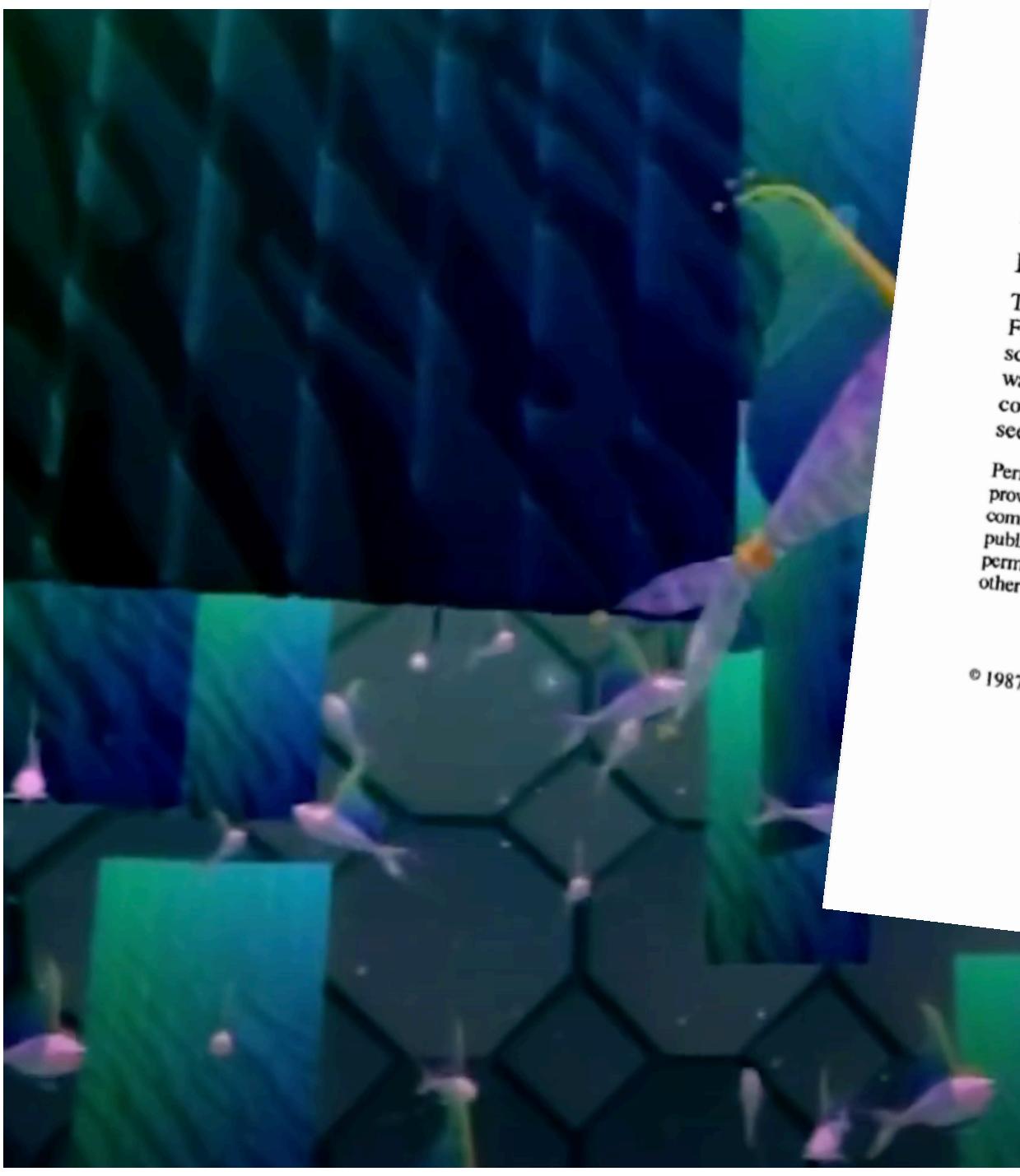
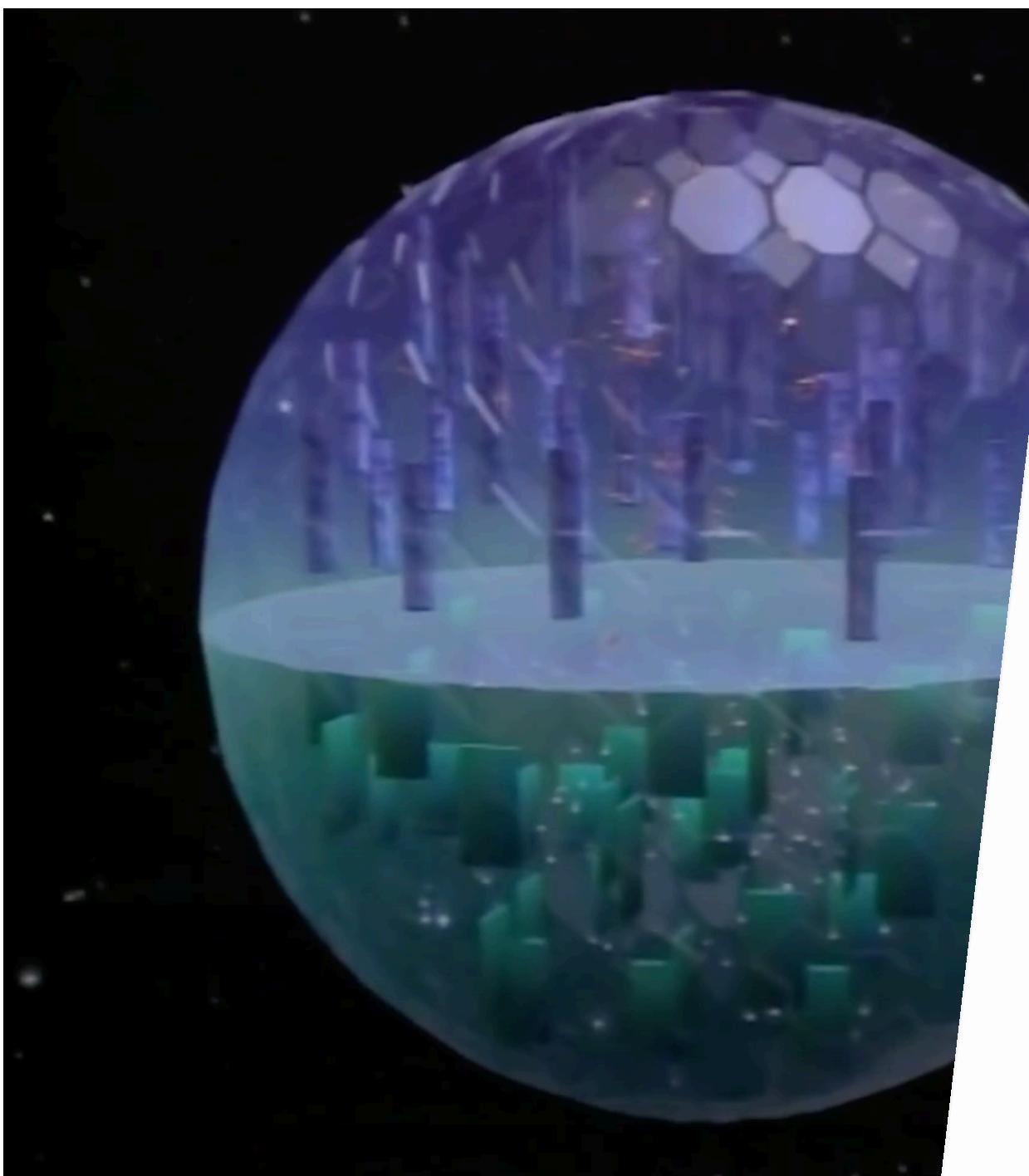


**SIGGRAPH 1987**

Breaking the Ice

Flocks, Herds, and Schools:  
A Distributed Behavioral Model





## Flocks, Herds, and Schools: A Distributed Behavioral Model

Craig W. Reynolds  
Symbolics Graphics Division

1401 Westwood Boulevard  
Los Angeles, California 90024

(Electronic mail: cwr@Symbolics.COM)

### Abstract

The aggregate motion of a flock of birds, a herd of land animals, or a school of fish is a beautiful and familiar part of the natural world. But this type of complex motion is rarely seen in computer animation. This paper explores an approach based on simulation as an alternative to scripting the paths of each bird individually. The simulated flock is an elaboration of a particle system, with the simulated birds being the particles. The aggregate motion of the simulated flock is created by a distributed behavioral model much like that at work in a natural flock; the birds choose their own course. Each simulated bird is implemented 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 programmed into it by the "animator." The aggregate motion of the simulated flock is the result of the dense interaction of the relatively simple behaviors of the individual simulated birds.

Categories and Subject Descriptors: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.3 [Simulation and Modeling]: Applications.

General Terms: Algorithms, design.

Additional Key Words, and Phrases: flock, herd, school, bird, fish, aggregate motion, particle system, actor, flight, behavioral animation, constraints, path planning.

### Introduction

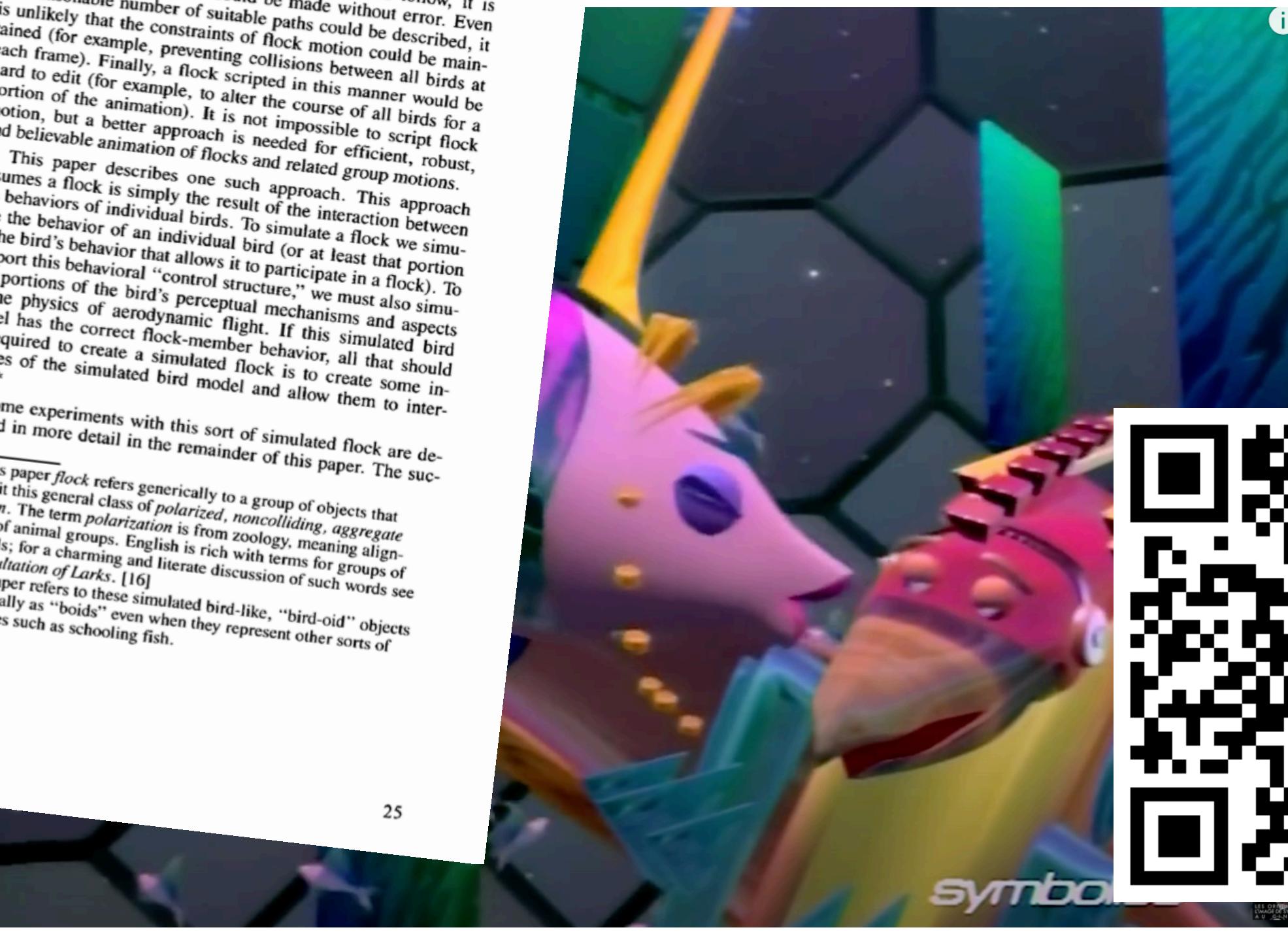
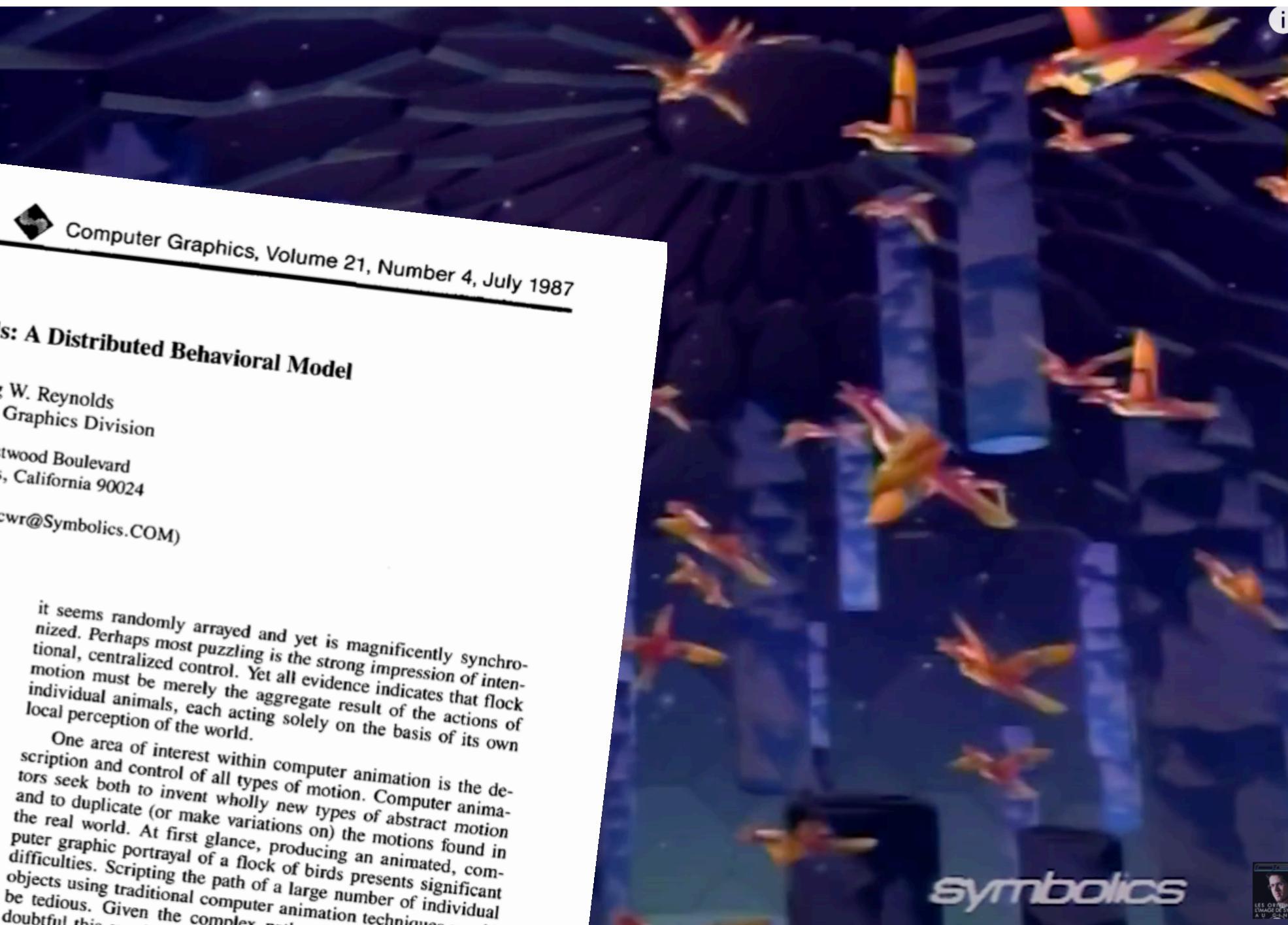
The motion of a flock of birds is one of nature's delights. Flocks and related synchronized group behaviors such as schools of fish or herds of land animals are both beautiful to watch and intriguing to contemplate. A flock\* exhibits many contrasts. It is made up of discrete birds yet overall motion seems fluid; it is simple in concept yet is visually complex,

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1987

ACM-0-89791-227-6/87/007/0025

\$00.75



\*In this paper *flock* refers generically to a group of objects that exhibit this general class of *polarized*, *noncolliding*, *aggregate* motion. The term *polarization* is from zoology, meaning alignment of animal groups. English is rich with terms for groups of animals; for a charming and literate discussion of such words see *An Exaltation of Larks*. [16]

\*\*This paper refers to these simulated bird-like, "bird-oid" objects generically as "boids" even when they represent other sorts of creatures such as schooling fish.



# **SYMBOLICS**

## **GRAPHICS DIVISION**

# Boids and inverse design

## 2024-2025

**Boid models are hard to “tune”**

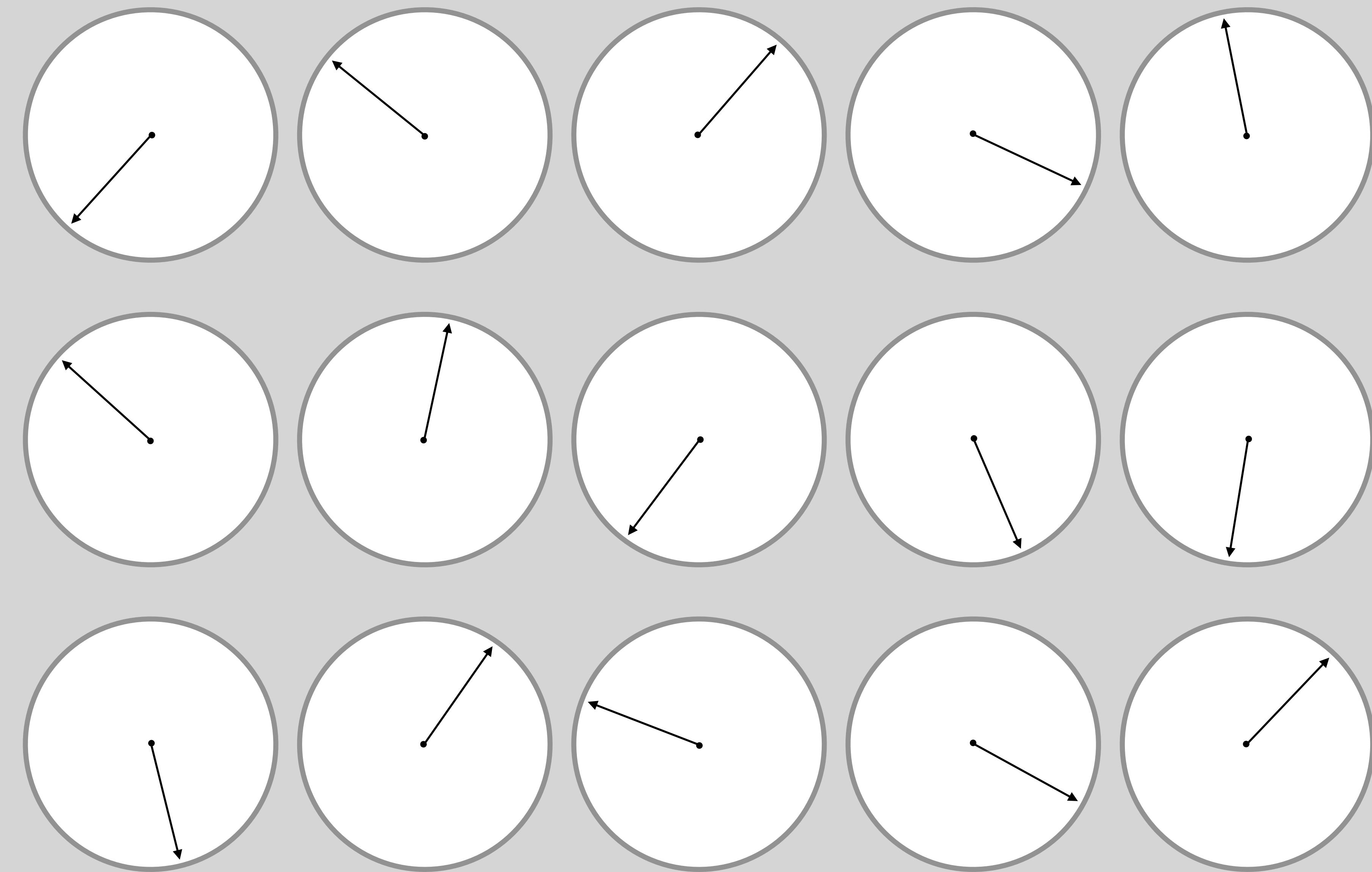
# **Boid models are hard to “tune”**

They have lots of parameters,  
which have nonlinear effect,  
and all interact with each other.

# **Boid models are hard to “tune”**

adjusting one knob requires  
adjusting others to compensate

# flock control panel



**How can we use optimization to tune  
boid flocks to get a desired result?**

# **How can we use optimization to tune boid flocks to get a desired result?**

Biological example: finch versus crow.  
Animation example: happy versus afraid.

# **How can we use optimization to tune boid flocks to get a desired result?**

“Inverse design.”

Create a metric/loss/fitness function.

Optimize toward design goal.

# Preliminary work in progress

- Joint work with Gilbert Bernstein, Matthew Shang, and Jennifer Luo at University of Washington.
- Made reference boids model, first in Python then in C++.
- Two approaches being pursued:
  - Gradient descent with differentiable programming.
  - Gradient-free evolutionary optimization.
    - **Genetic algorithm: parameters for a fixed boid model**
    - Genetic programming: steering programs from scratch

# Preliminary work in progress

- Joint work with Gilbert Bernstein, Matthew Shang, and Jennifer Luo at University of Washington.
- Made reference boids model, first in Python then in C++.
- Two approaches being pursued:
  - Gradient descent with differentiable programming.
  - Gradient-free evolutionary optimization.
    - **Genetic algorithm: parameters for a fixed boid model**
    - Genetic programming: steering programs from scratch



# **Difficult to find gradient of chaotic system**

Multi-agent systems are chaotic.  
Finding a principled gradient is hard.  
That analysis is ongoing.

**In parallel, using evolutionary optimization**

# Evoflock: evolved inverse design of multi-agent motion

Craig Reynolds  
unaffiliated researcher  
cwr@red3d.com

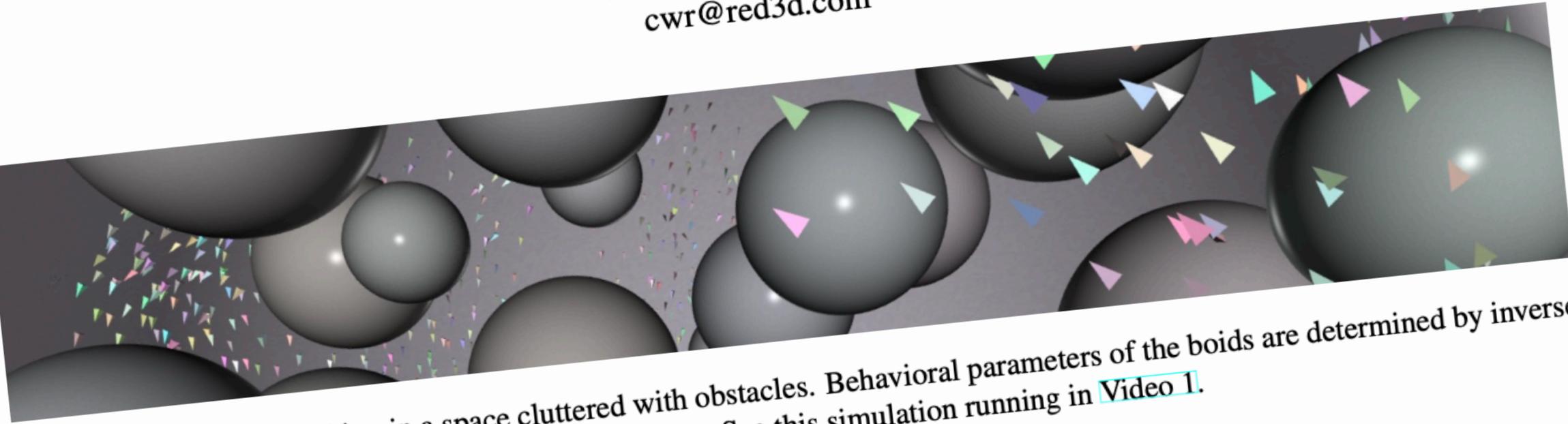


Figure 1: 1000 boids flocking in a space cluttered with obstacles. Behavioral parameters of the boids are determined by inverse design, using multi-objective evolutionary optimization. See this simulation running in [Video 1](#).

## Abstract

This paper describes an automatic method for *adjusting* or *tuning* models of multi-agent motion. Simulating the motion of bird flocks, human crowds, vehicle traffic, and other multi-agent systems is a widely used technique. These simulations model the behavior of a single group member (bird, human, or vehicle). The group behaviors (flock, crowd, traffic) *emerge* from interactions between group members. These models typically have many numeric control parameters. Although each parameter may be intuitive in isolation, their interaction can be complex and nonlinear. It can be difficult to know how to adjust which parameters to make a desired change in the group behavior. Changing one aspect of group behavior often causes other aspects to change, leading to a long frustrating process of incremental changes. In this work, the desired group behavior is measured with an objective(fitness/loss) function and optimized with a genetic algorithm. The objective function used here combines scoring that a flock's birds maintain proper spacing with neighbors, fly near a given speed, and avoid obstacles. Interestingly, the vivid alignment seen in bird flocks appears to emerge from maintaining proper spacing between flockmates.

**Keywords:** multi-agent, inverse design, optimization, evolutionary computation, genetic algorithm, boids, flocks, herds, schools, crowds, traffic

## Introduction

Simulation models of multi-agent motion are used in many fields. These include animation, games, biology, robotic swarms, urban planning, training autonomous vehicles, and other applications. Since the 1980s, several models of group motion have been developed, including *boids* and others (see

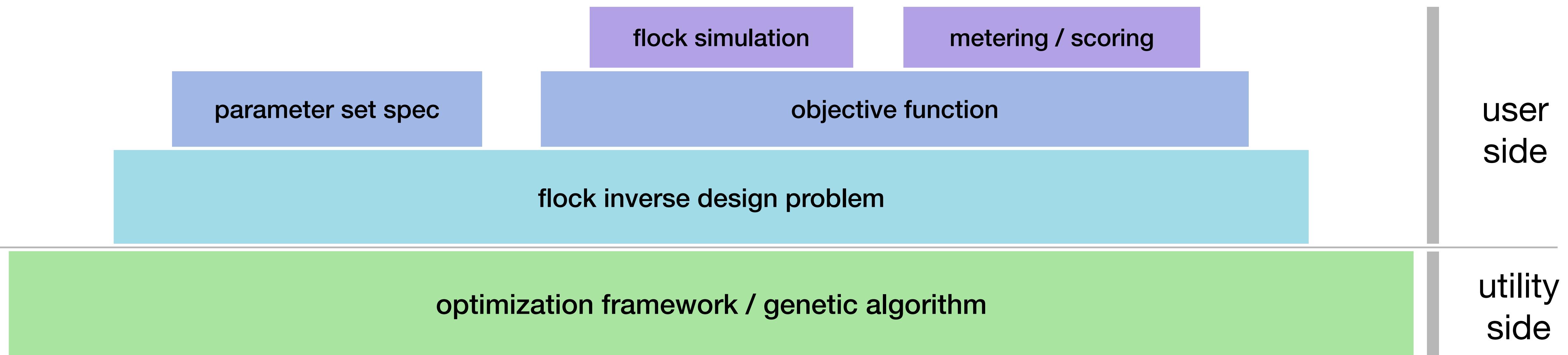
flocks, with the assumption that other types of group motion (herds, schools, crowds, traffic, drone swarms) can be similarly modeled.

This project addresses the issue of *adjusting* or *tuning* multi-agent motion models, toward a given behavioral goal. For example, modifying an existing model of bird flocks to instead portray fish schools. Or starting from a model of flocking crows and change it to represent flocking sparrows. Similarly, starting from a generic abstract flock model and fitting it to field observations of a particular species of real birds in nature. Or to take a plausibly realistic model of a natural bird flock, and change it, say for storytelling purposes, to convey a flock of birds that are happy, or angry.

A boids-like simulation model typically has a collection of numeric parameters (“knobs”) that control its action. EvoFlock is a software framework that automatically finds a set of near-optimal parameters for a simulation-based multi-agent system. The flock model used for these experiments is a predefined hand-written “black box” model whose input is a *parameter set* (consisting of 15 numbers, see Table 2). The user provides a *objective* (also known as *fitness* or *loss*) function. It takes a candidate parameter set, runs a simulation, and returns a *score* reflecting how well the behavioral goals were met. The optimization process runs (for about two hours on a laptop) and produces a high quality parameter set, as measured by the objective function.

For a user of this optimization framework, it is convenient that the flock model is treated as a black box. Almost no restrictions are imposed on a preexisting model by the optimization framework. There are no restrictions on model design, source code availability, or the programming language used. (As could be an issue if optimization required language tools such as automatic differentiation (Baydin et al.,

# EvoFlock components





Brooklyn



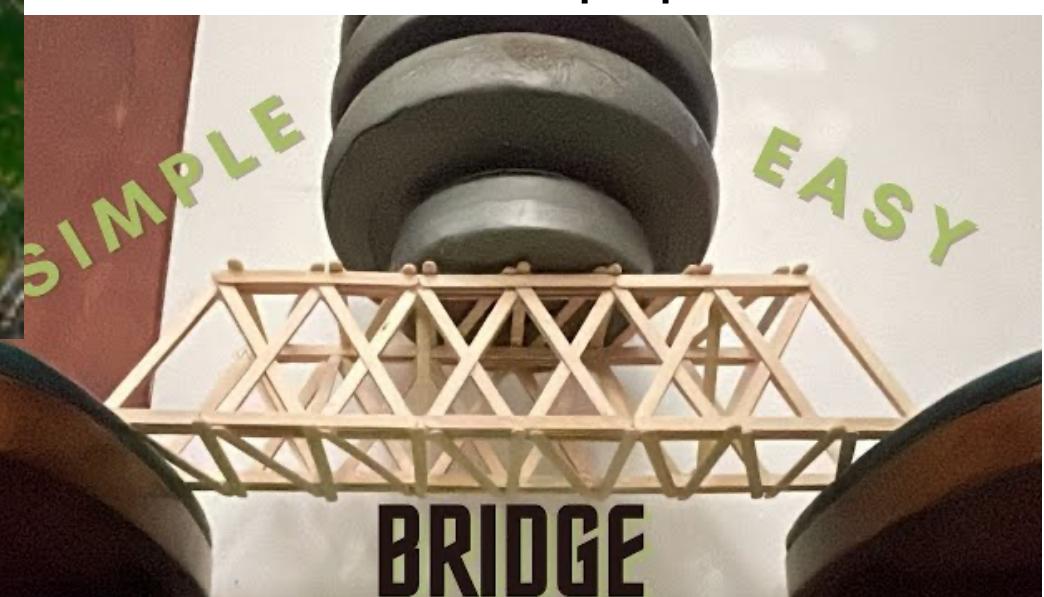
Firth of Forth

expensive

Strong



Natchez Trace Parkway



popsicle sticks

cheap

Tacoma Narrows



Tay

weak

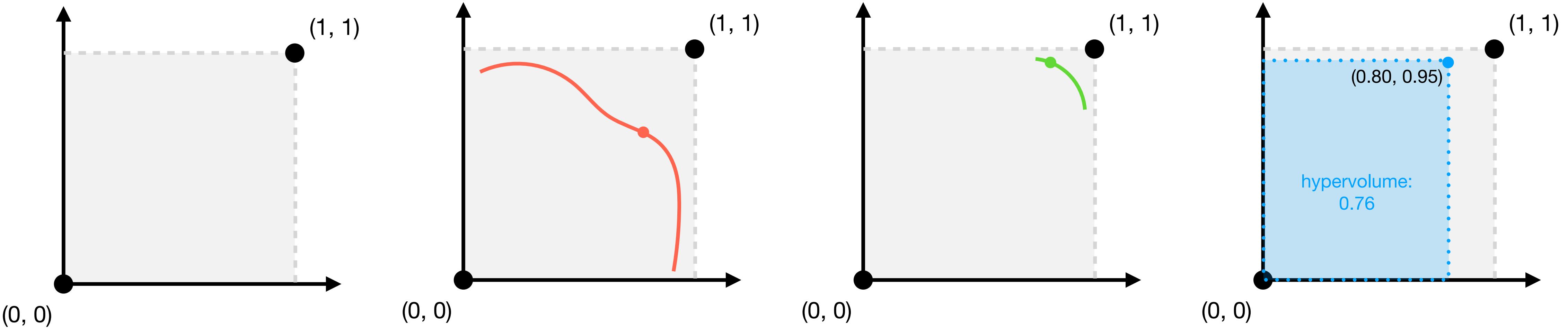


metal tubes and wood deck



rickety wood bridge

# Multi-objective optimization – normalized fitness space.

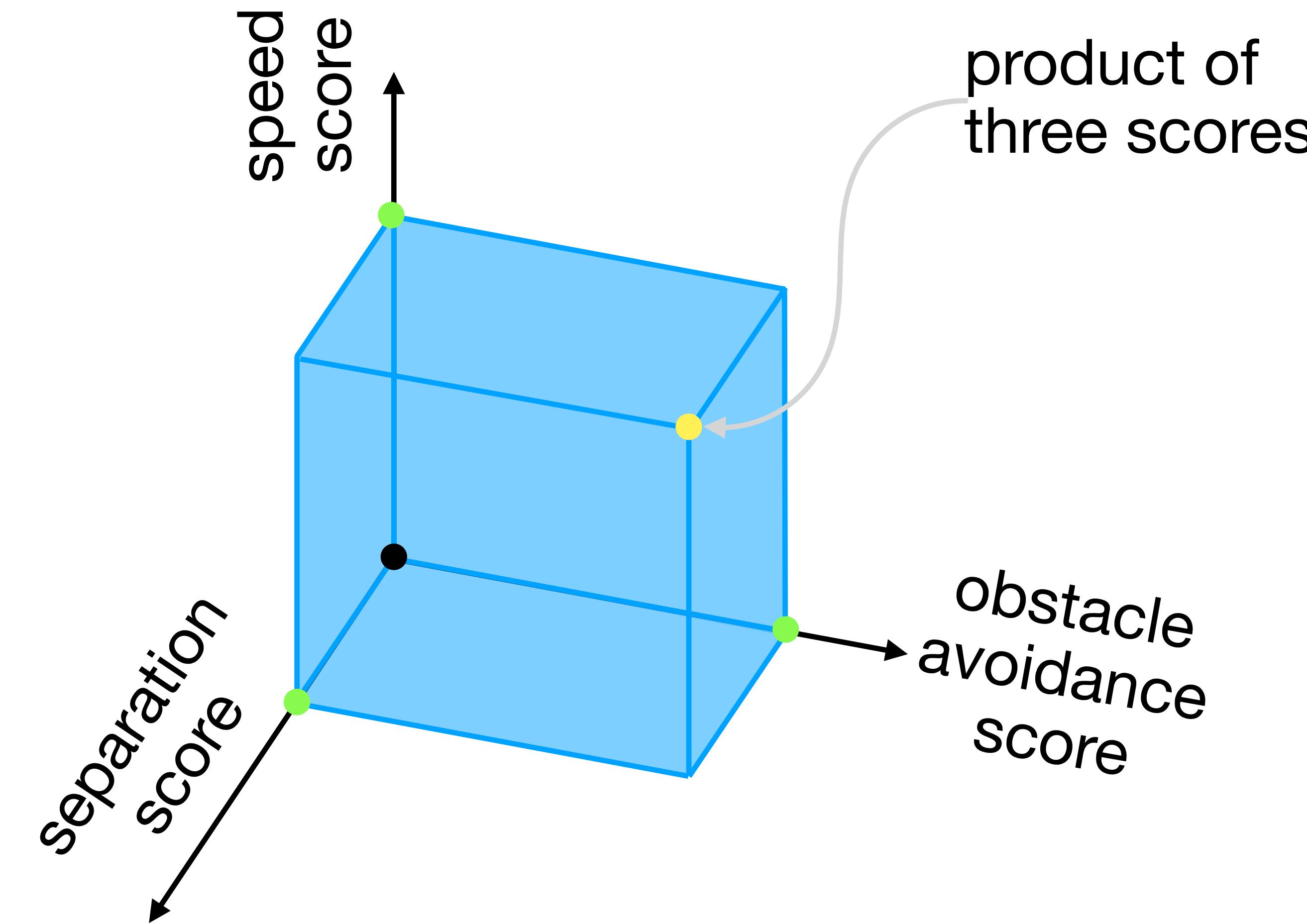


*Pareto front* for two  
mutually conflicting  
objectives.

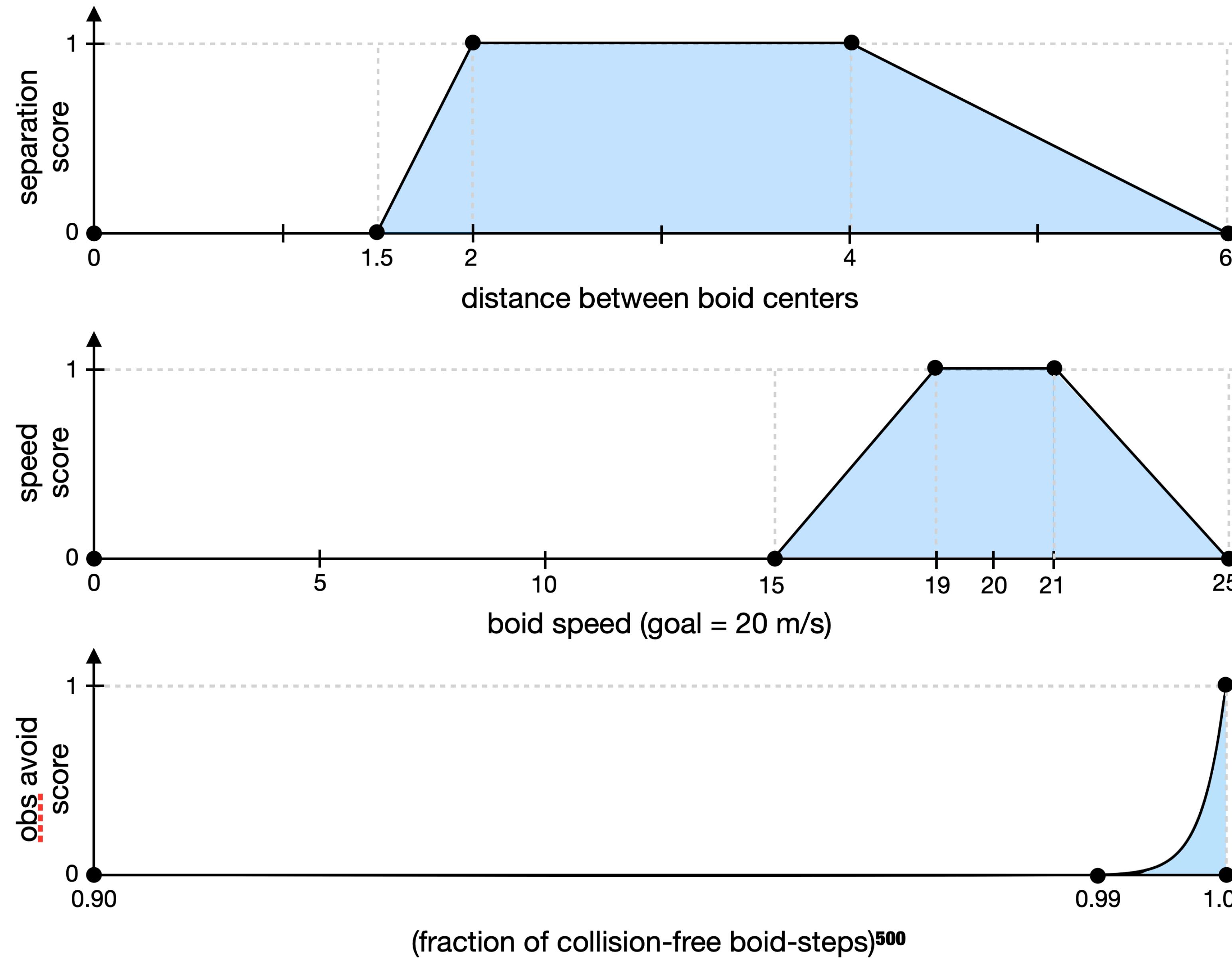
*Pareto front* for two  
“mostly compatible”  
objectives.

*Scalarization (area)*  
for two objectives.

# Hypervolume of three objectives

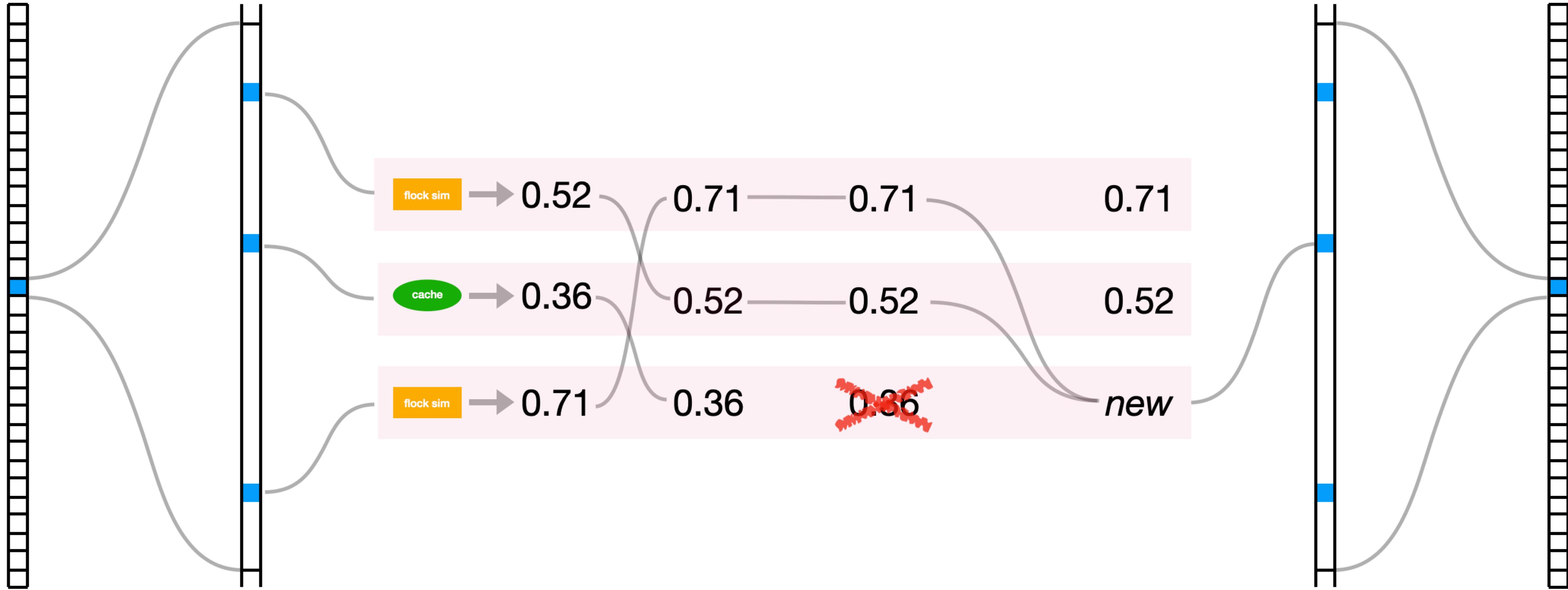


# Score weighting functions: separation, speed, avoid

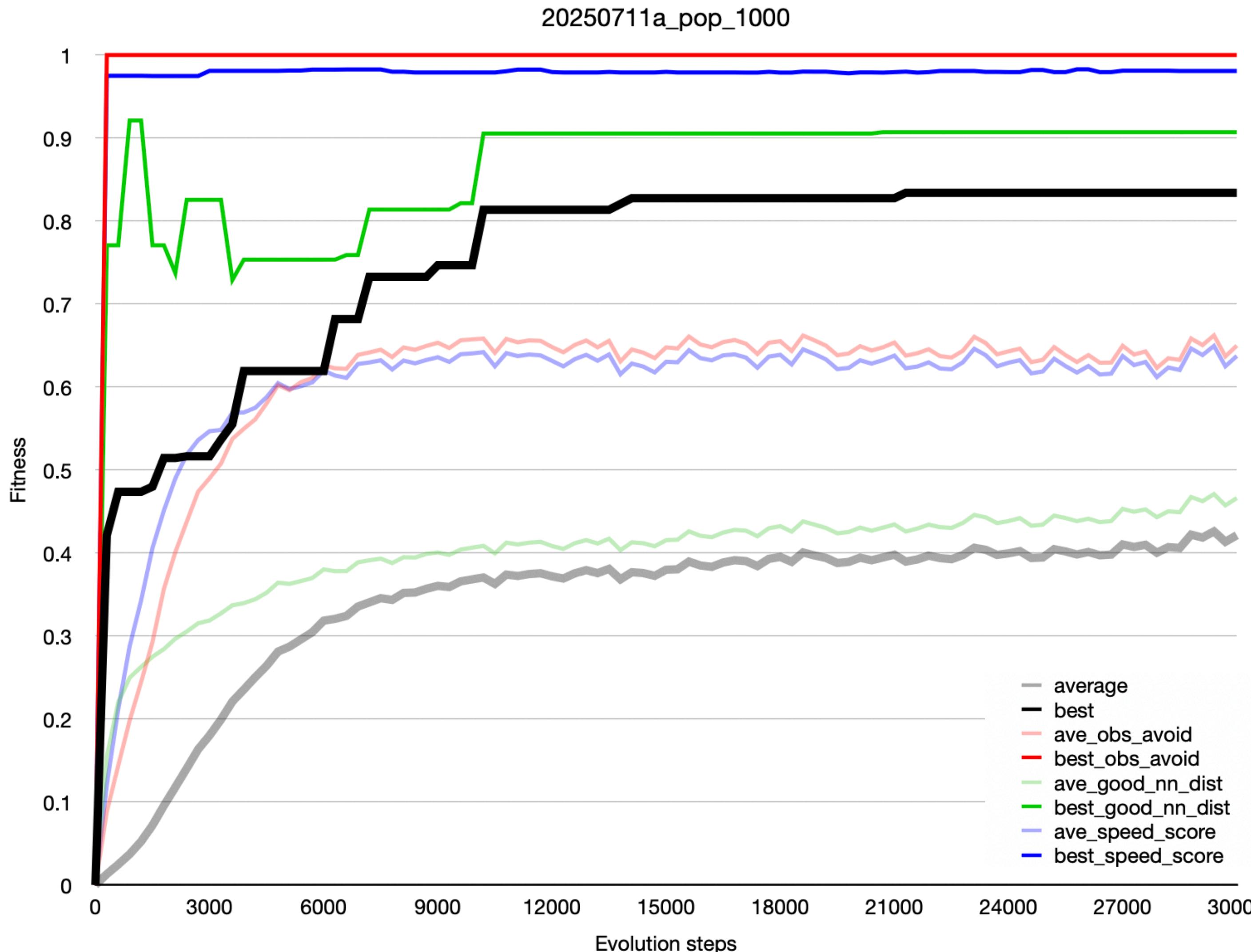


# Single step of steady-state genetic algorithm (SSGA)

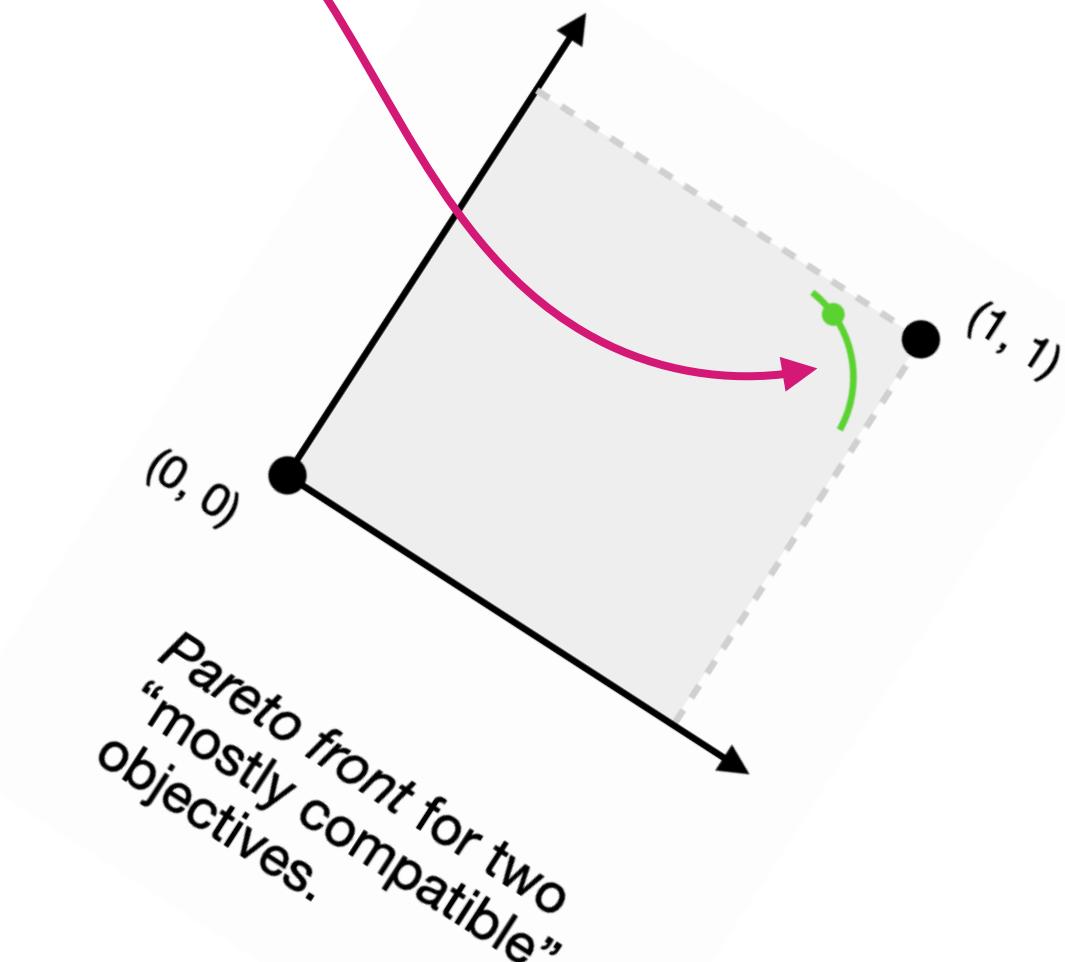
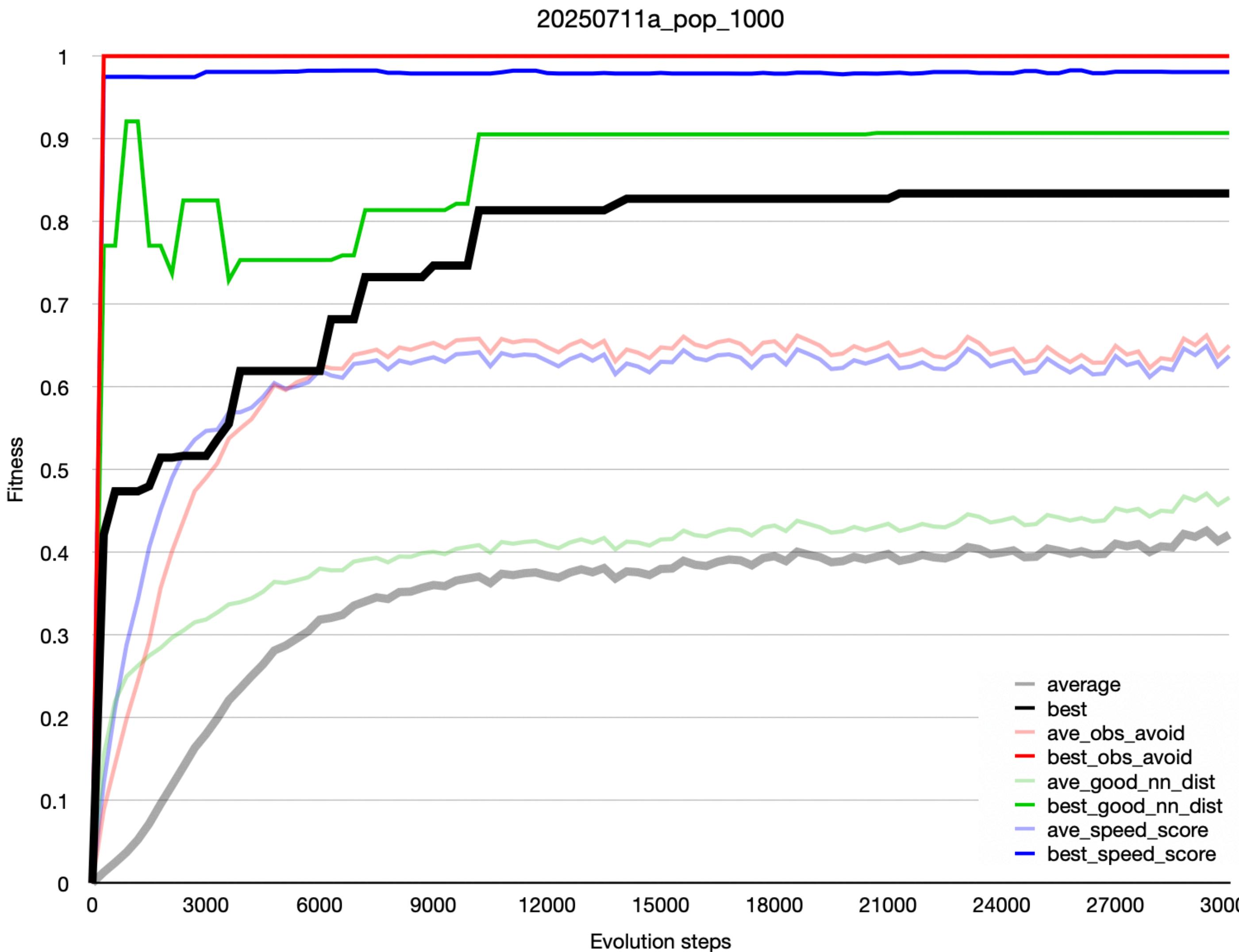
3 way tournament, negative selection, preserves diversity



# EvoFlock fitness (and multiple objectives)



# EvoFlock fitness (and multiple objectives)



video

# Adding an objective: curvature

video

# Thank you!

contact:

[cwr@red3d.com](mailto:cwr@red3d.com)

<https://www.red3d.com/cwr/>

these slides:

~~<https://www.red3d.com/cwr/presentations/2025001-HOSC-sample-slides.pdf>~~

