Flocking and inverse design

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

these slides:

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

This talk will be a whirlwind tour of a mish-mash of research topics

chronological order:

classic “boids”

coevolution of camouflage

inverse design of boid flocks

This talk will be a whirlwind tour of a mish-mash of research topics

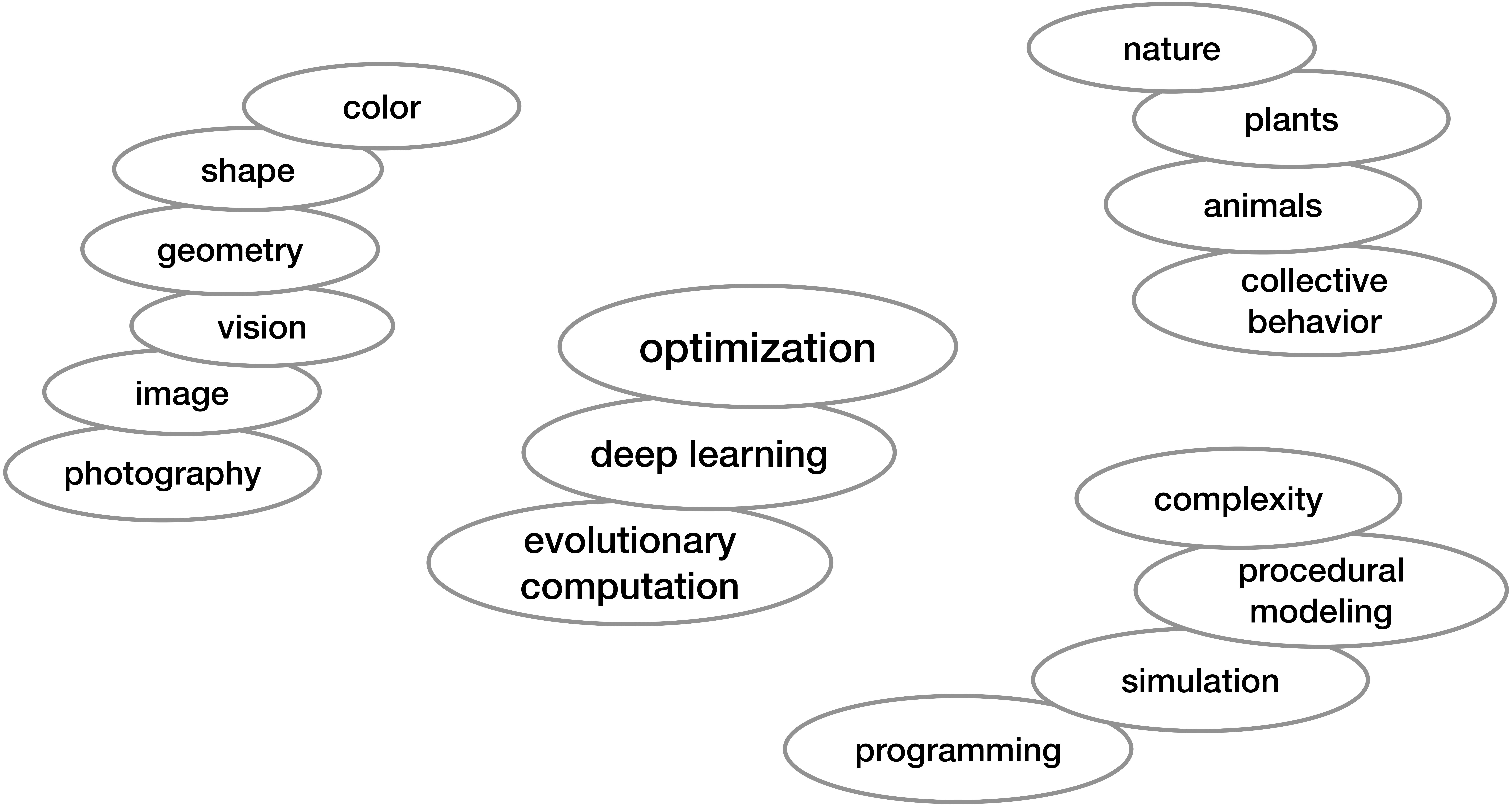
presentation order:

background

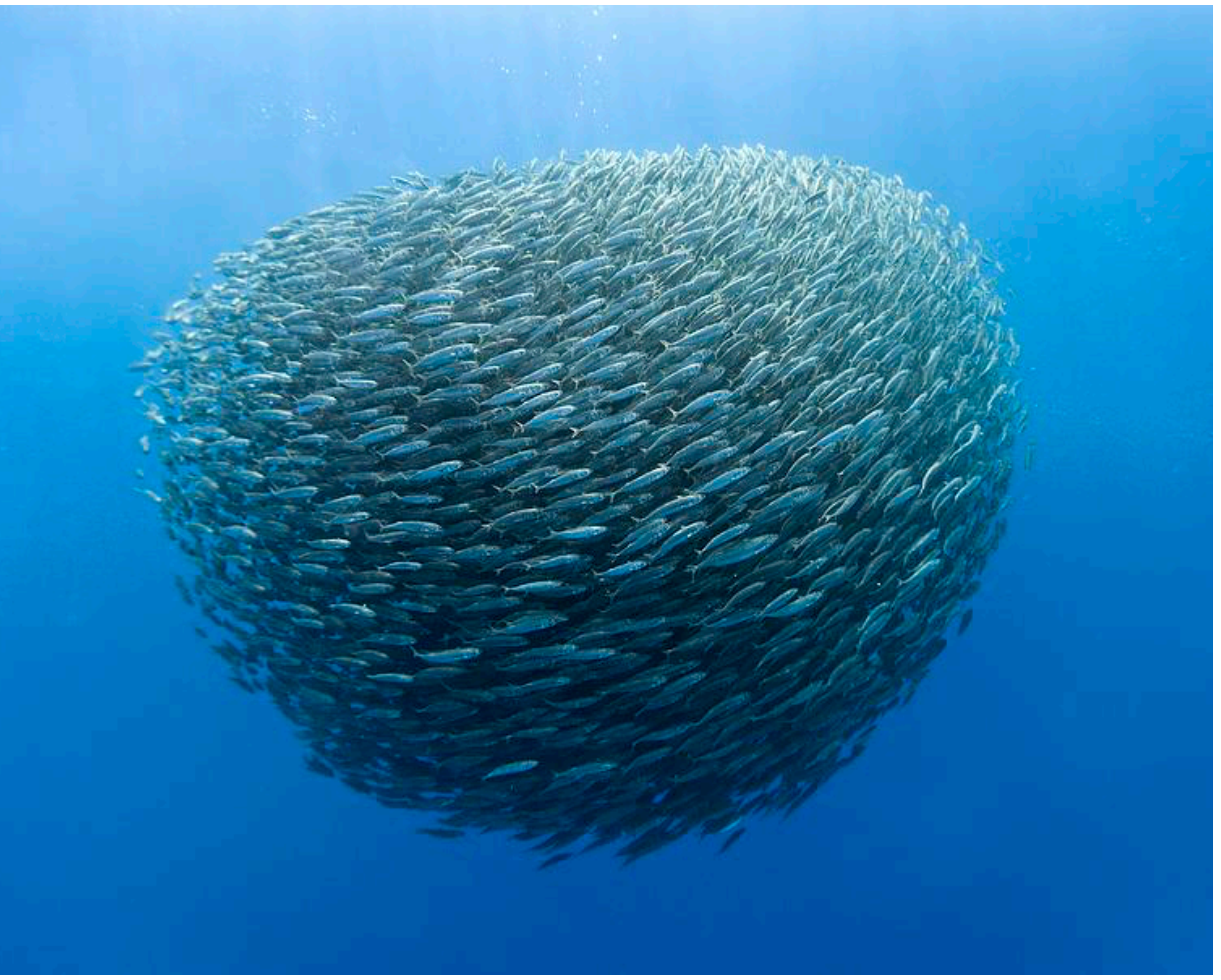
coevolution of camouflage

classic boids

inverse design of boid flocks


color
shape
geometry
vision
image
photography
optimization
deep learning
evolutionary computation
nature
plants
animals
collective behavior
complexity
procedural modeling
simulation
programming

A childhood fascination with natural complexity



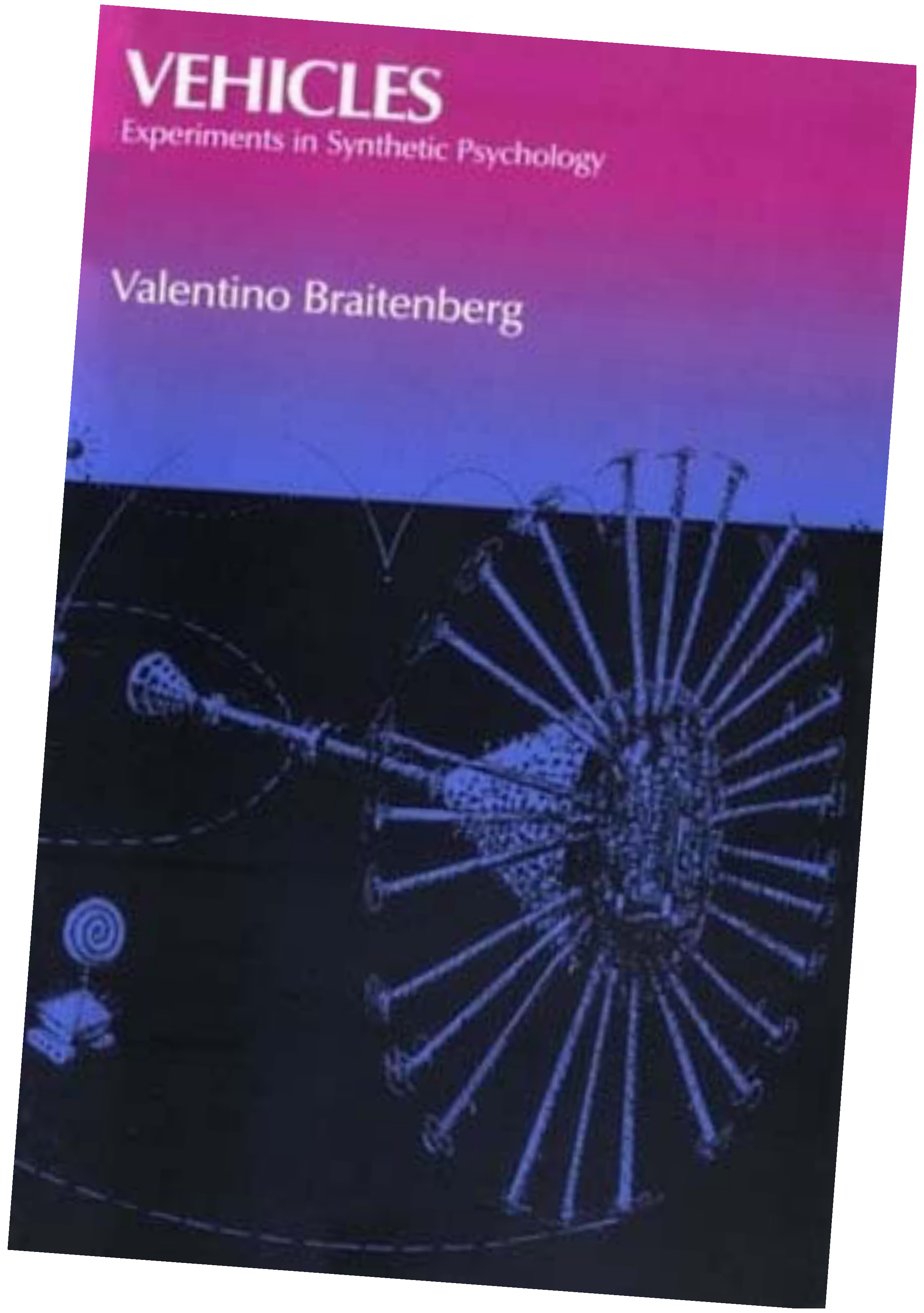
bird shaped flock on YouTube

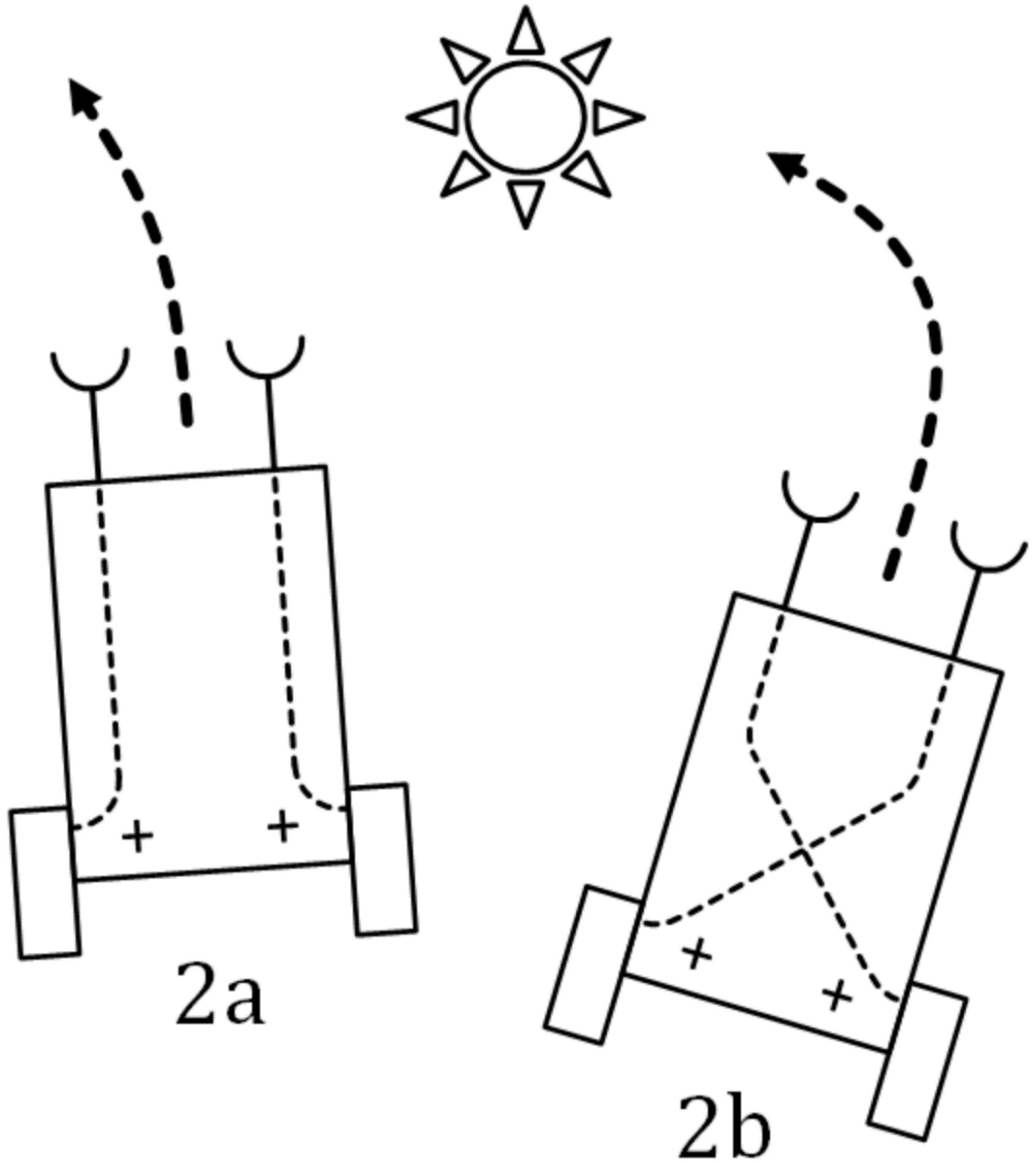
Early boids
1986-1987

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.

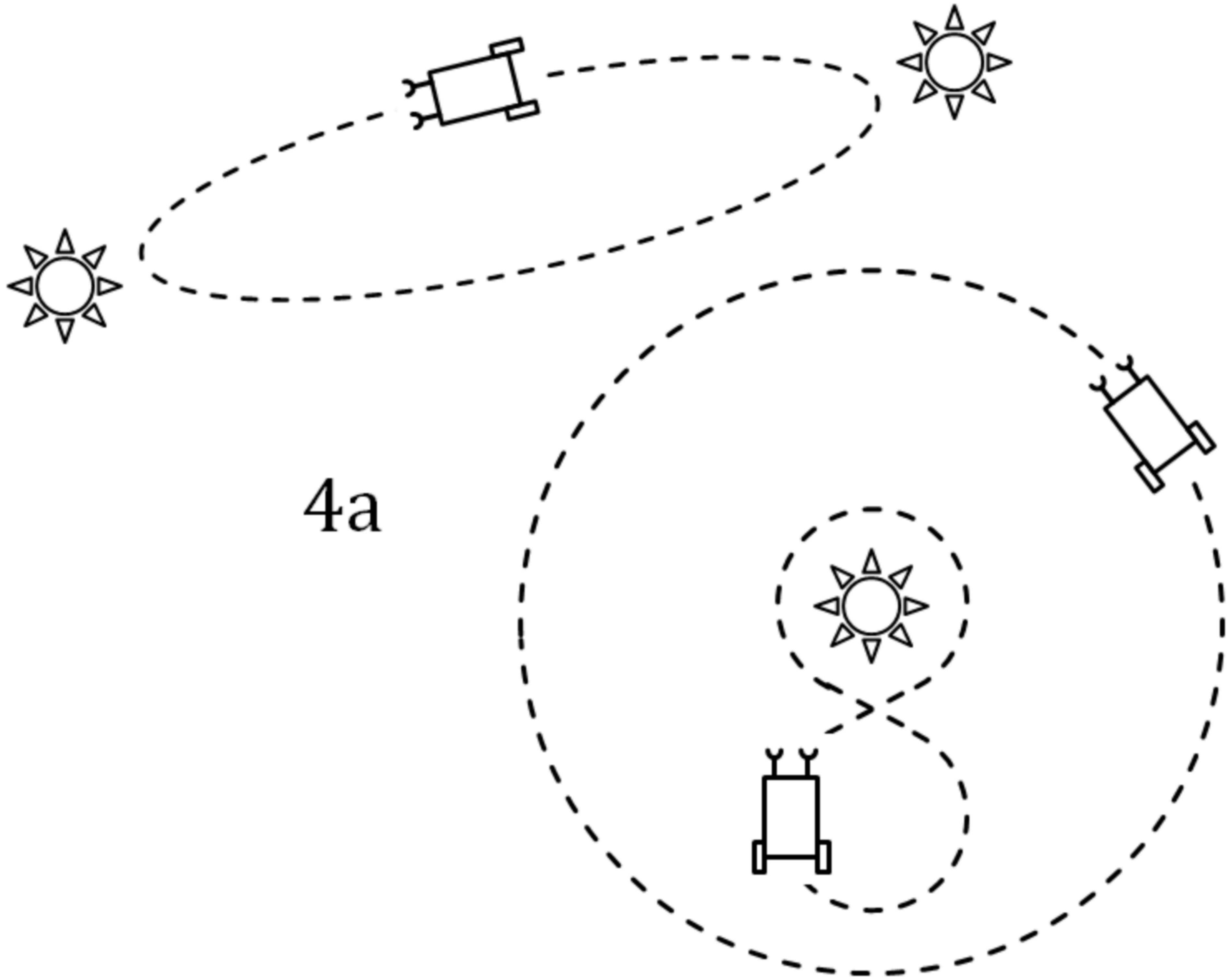
Could the behavior of flocking birds
be simulated in software?


VEHICLES
Experiments in Synthetic Psychology
Valentino Braitenberg

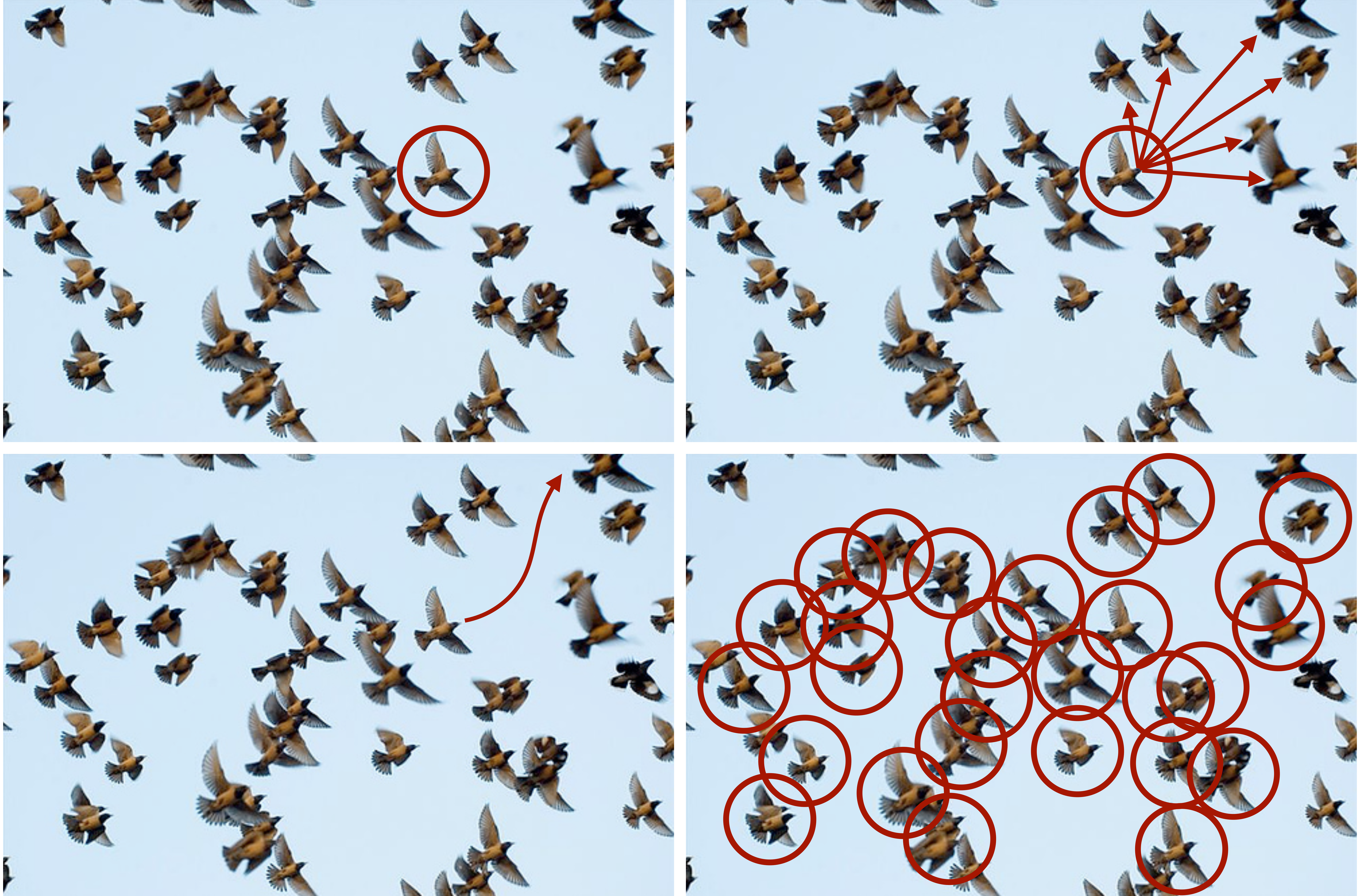


2a

2b

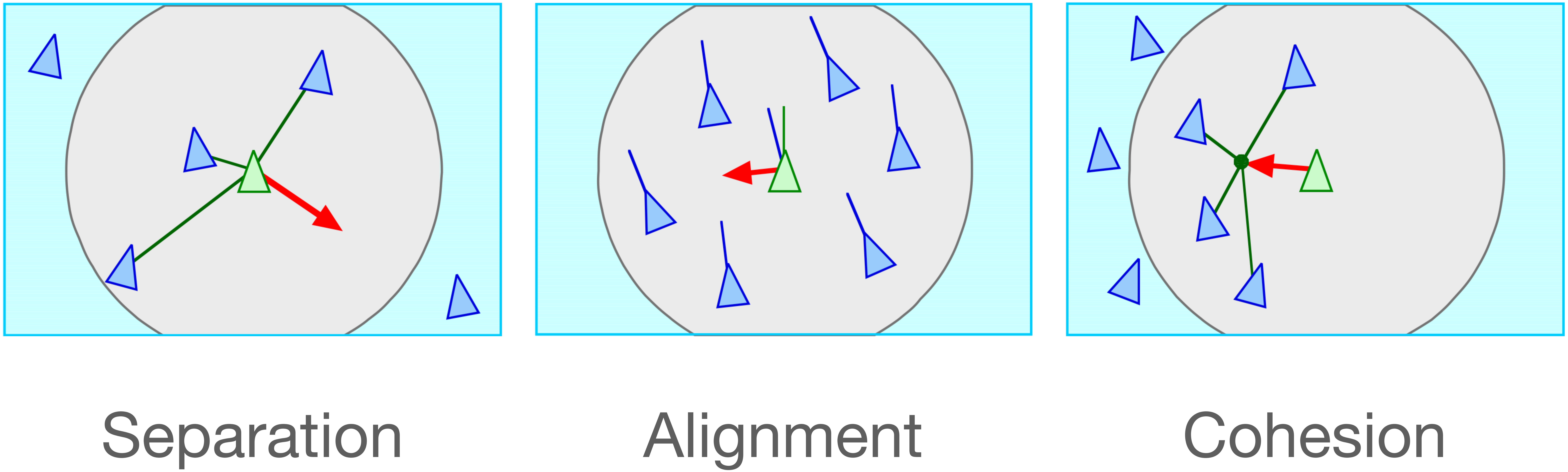


4a



Three components of flocking

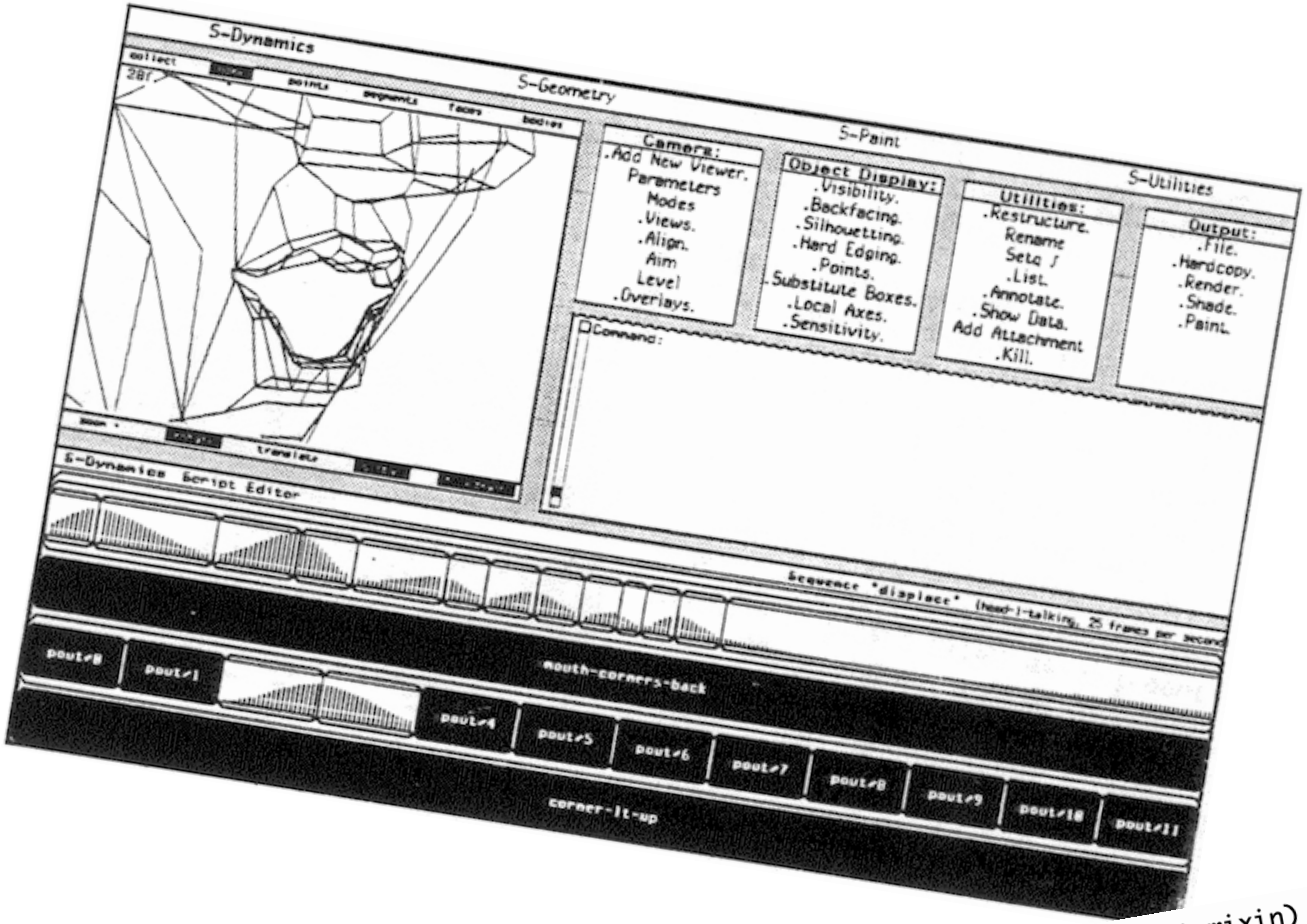
Seemed necessary, but were they sufficient?



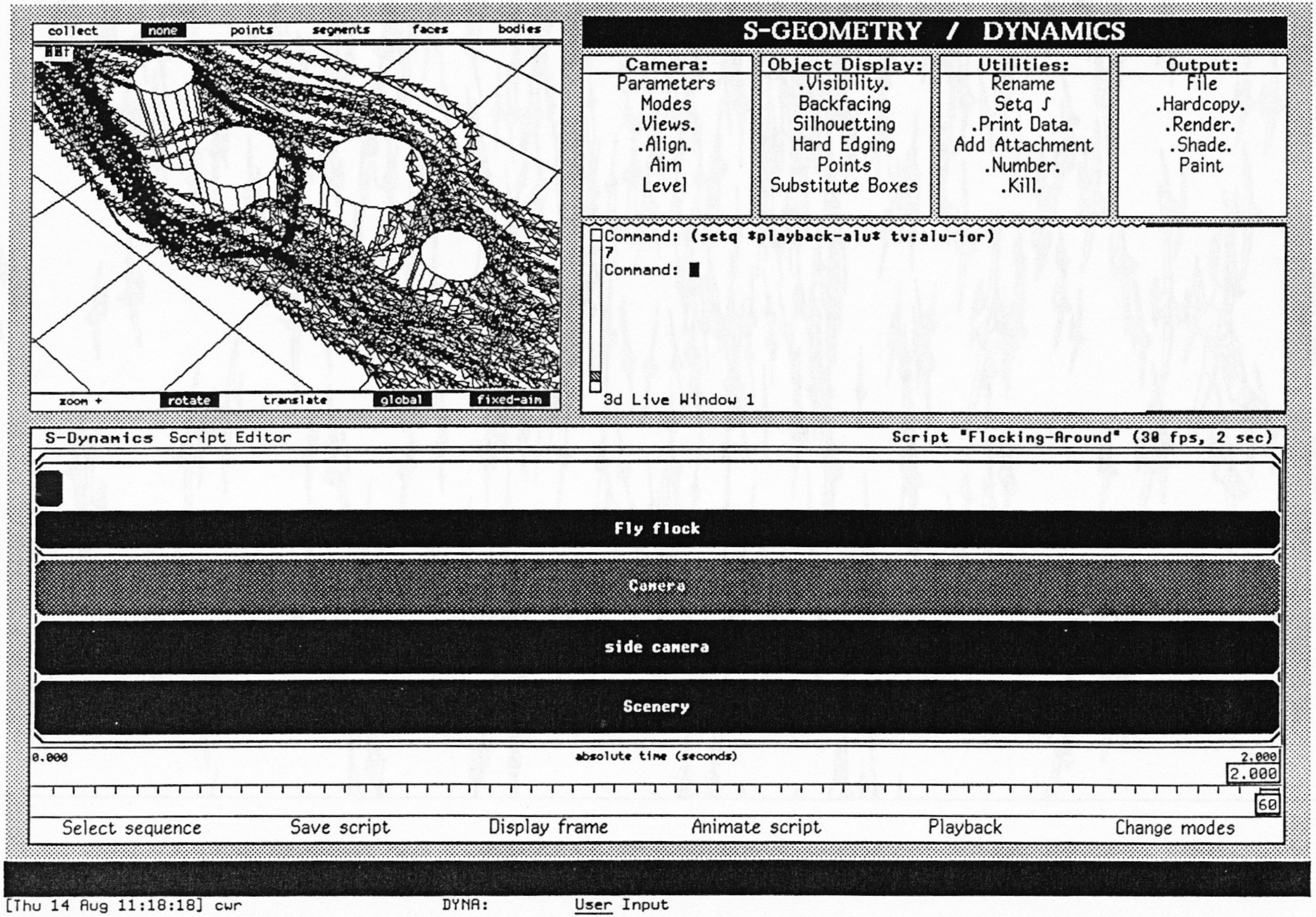
SIGGRAPH 1986:

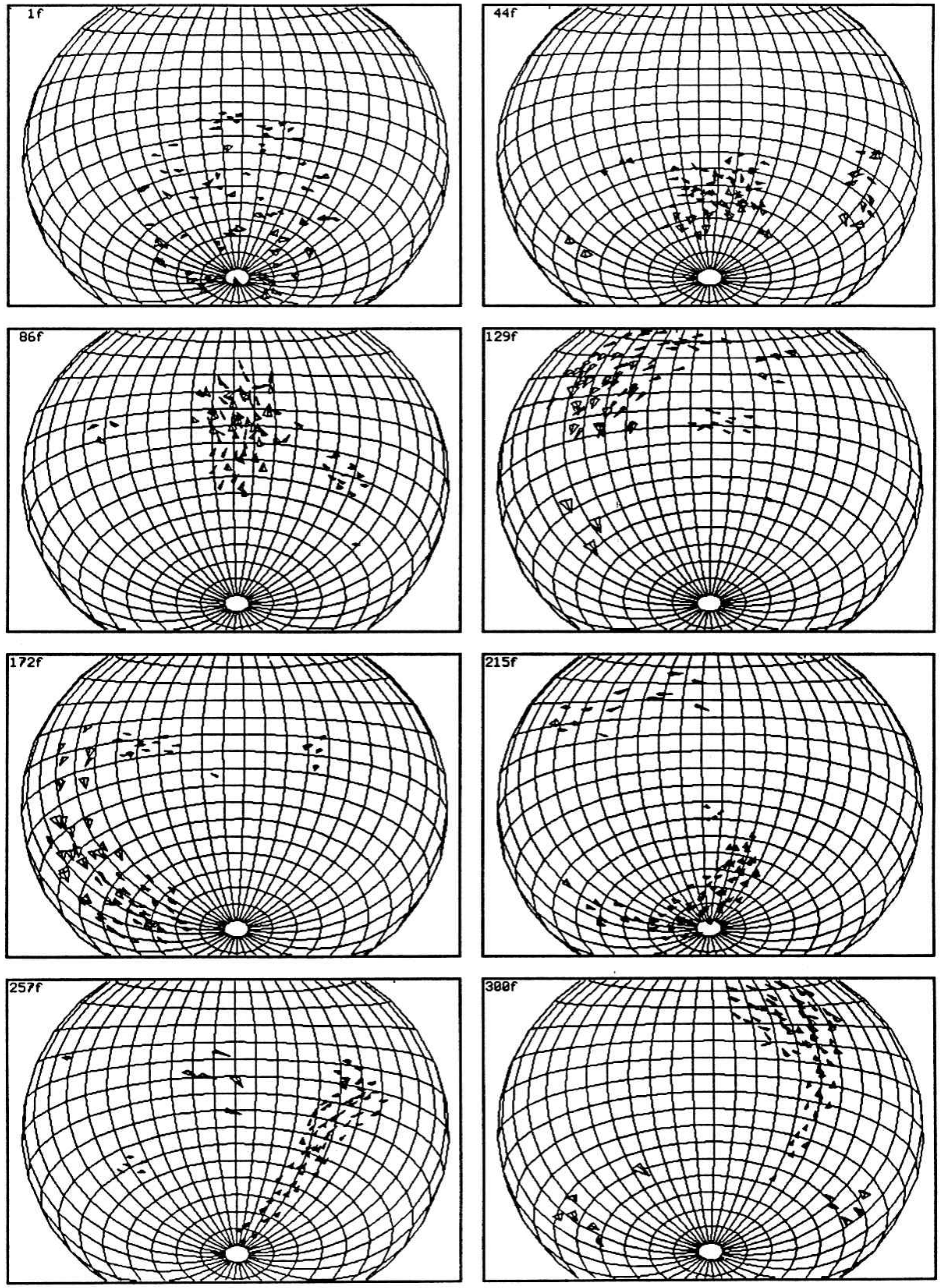
writing tutorial notes:

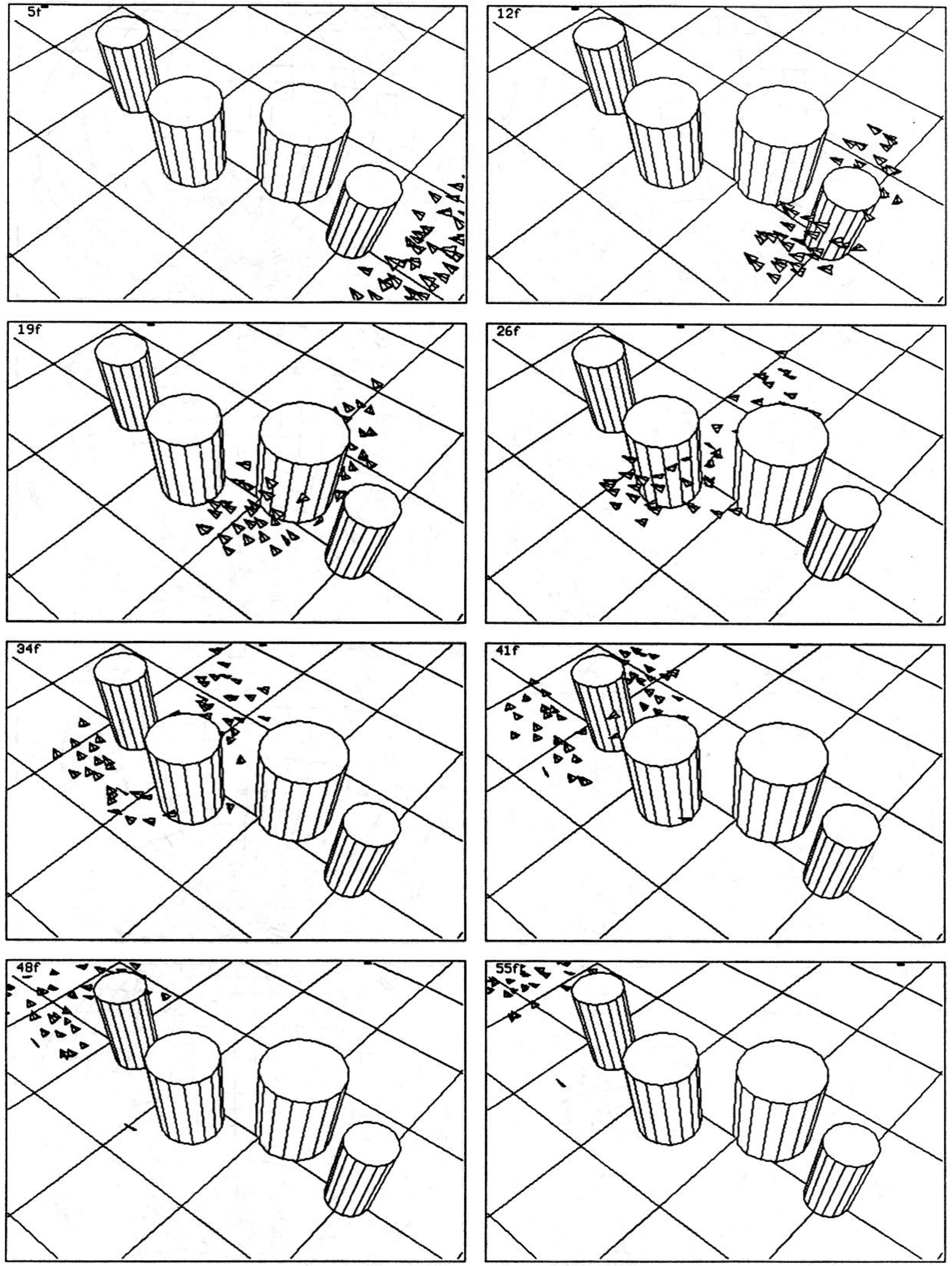
“…it would be easy…”




```
(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-steering-acceleration)))
    (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))))
```


collect
none
points
segments
faces
bodies
zoom +
rotate
translate
global
fixed-aim
S-GEOMETRY / DYNAMICS
Camera:
Parameters
Modes
.Views.
.Align.
Aim
Level
Object Display:
.Visibility.
Backfacing
Silhouetting
Hard Edging
Points
Substitute Boxes
Utilities:
Rename
Setq ʃ
.Print Data.
Add Attachment
.Number.
.Kill.
Output:
File
.Hardcopy.
.Render.
.Shade.
Paint
Command: (setq *playback-alu* tv:alu-ior)
7
Command:
3d Live Window 1
S-Dynamics Script Editor
Script "Flocking-Around" (30 fps, 2 sec)
Fly flock
Camera
side camera
Scenery
0.000
absolute time (seconds)
2.000
2.000
60
Select sequence
Save script
Display frame
Animate script
Playback
Change modes
[Thu 14 Aug 11:18:18] cwr
DYNA:
User Input

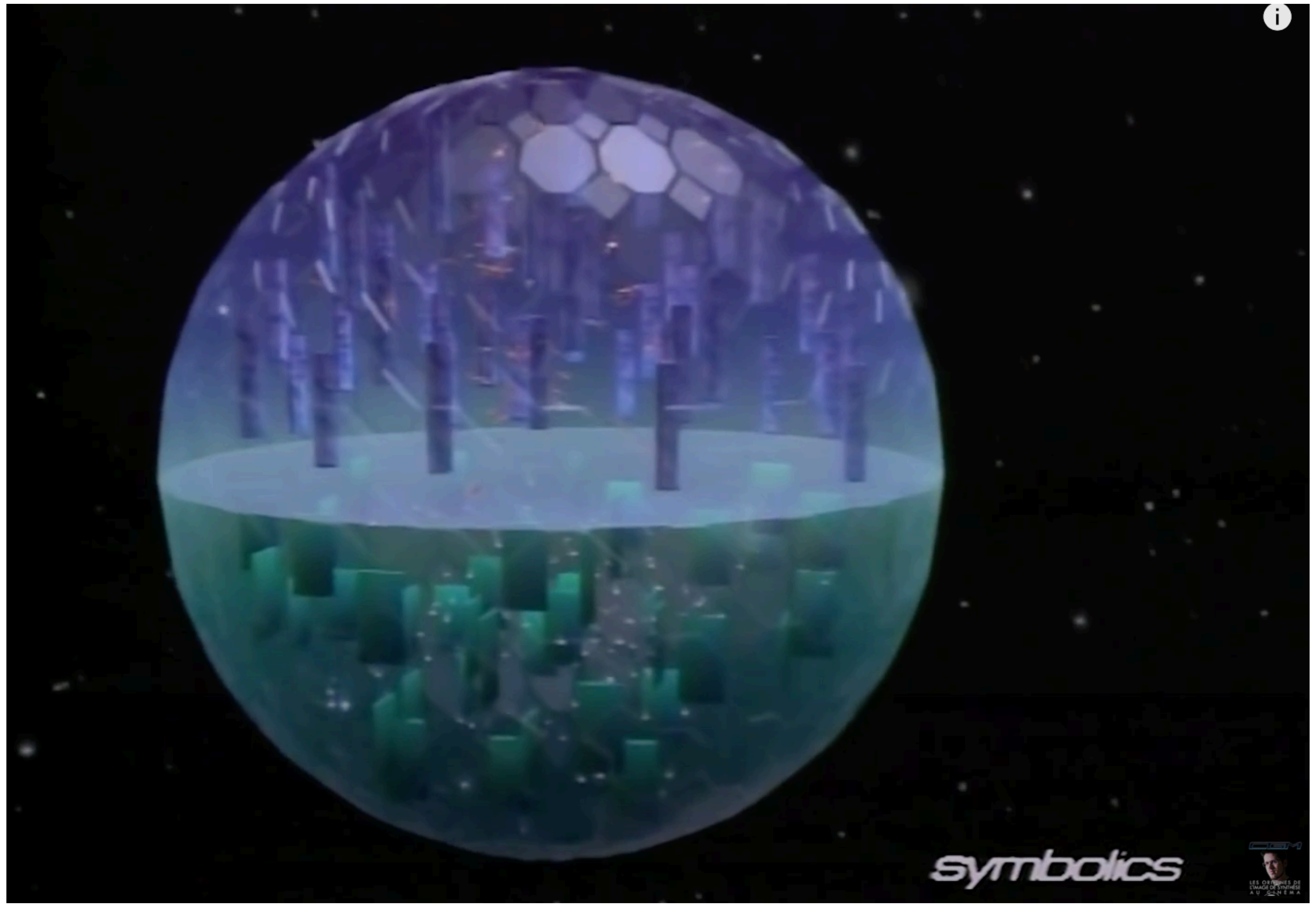

1f
44f
86f
129f
172f
215f
257f
300f


5f
12f
19f
26f
34f
41f
48f
55f

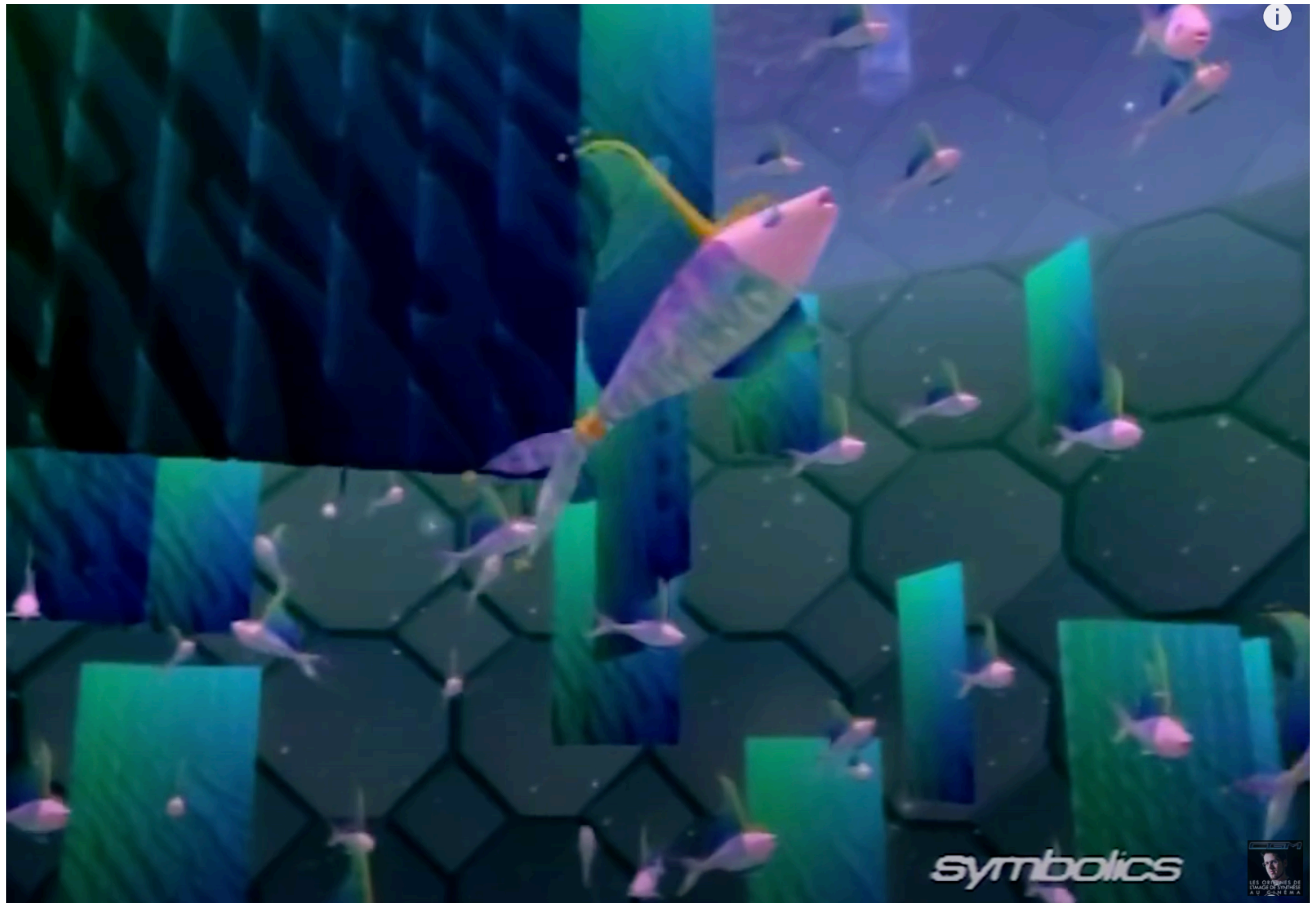
SIGGRAPH 1987

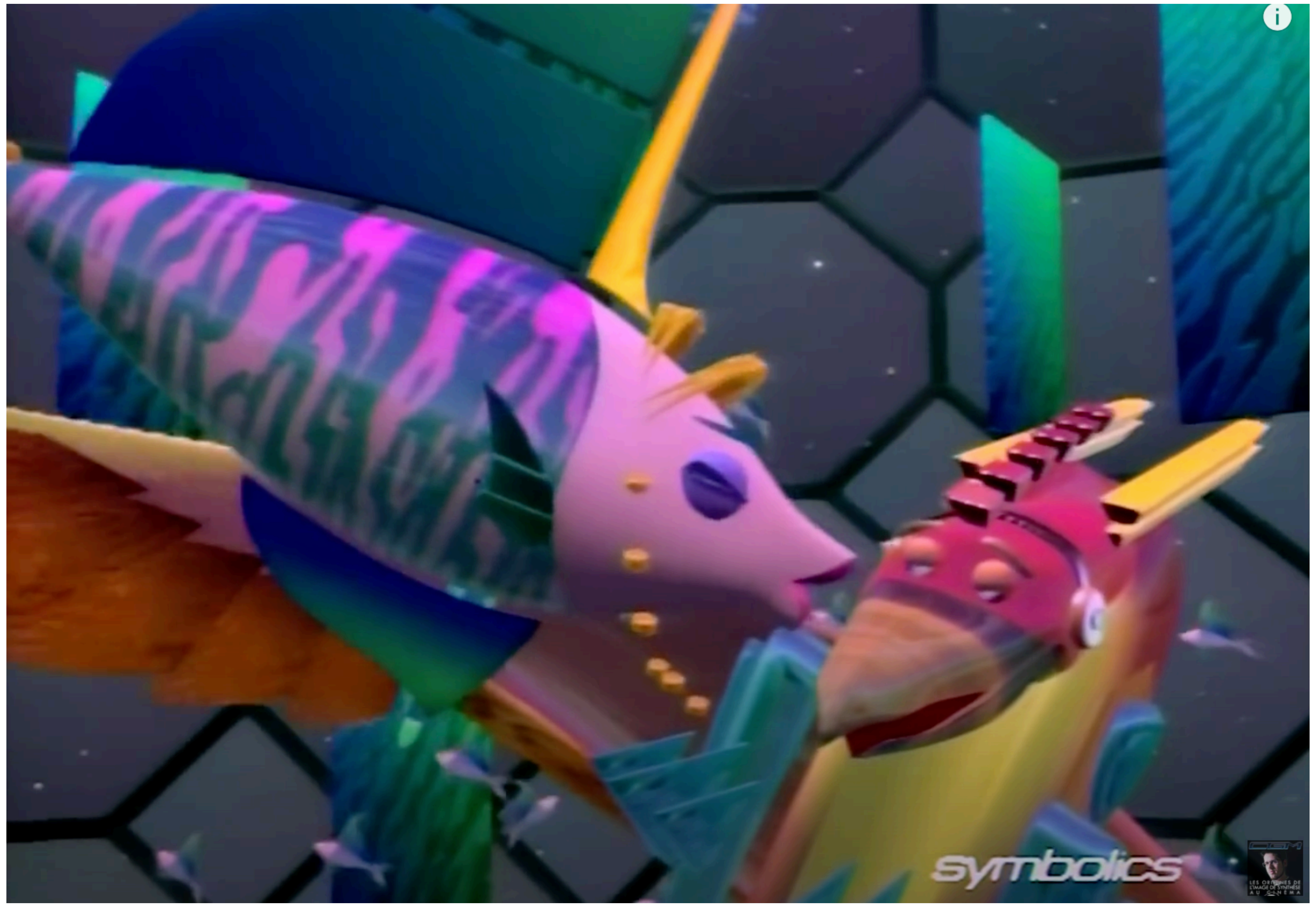
Breaking the Ice

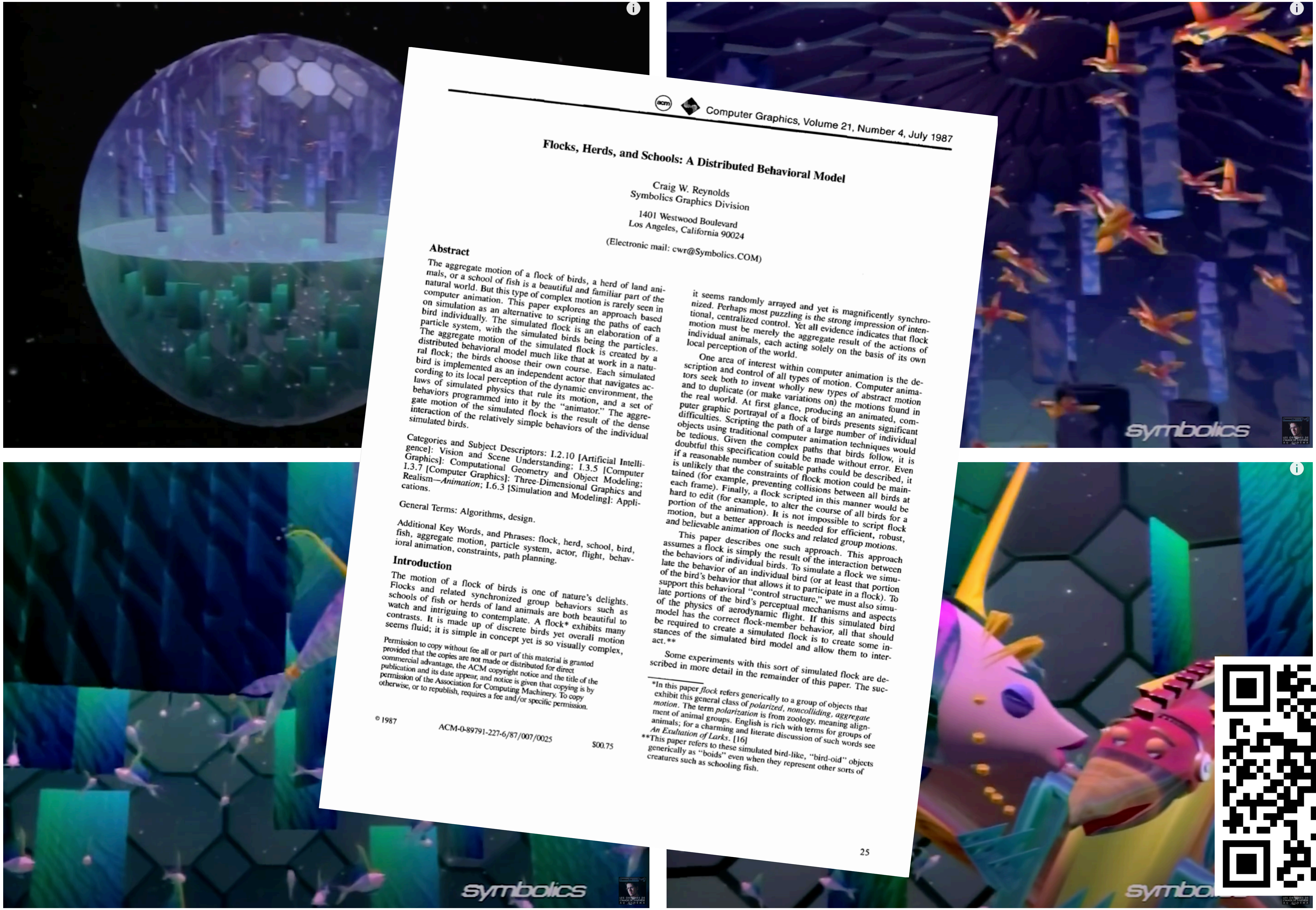
Flocks, Herds, and Schools:
A Distributed Behavioral Model


symbolics


symbolics


symbolics


symbolics



Computer Graphics, Volume 21, Number 4, July 1987

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 so visually complex, it seems randomly arrayed and yet is magnificently synchronized. Perhaps most puzzling is the strong impression of intentional, centralized control. Yet all evidence indicates that flock motion must be merely the aggregate result of the actions of individual animals, each acting solely on the basis of its own local perception of the world.

One area of interest within computer animation is the description and control of all types of motion. Computer animators seek both to invent wholly new types of abstract motion and to duplicate (or make variations on) the motions found in the real world. At first glance, producing an animated, computer graphic portrayal of a flock of birds presents significant difficulties. Scripting the path of a large number of individual objects using traditional computer animation techniques would be tedious. Given the complex paths that birds follow, it is doubtful this specification could be made without error. Even if a reasonable number of suitable paths could be described, it is unlikely that the constraints of flock motion could be maintained (for example, preventing collisions between all birds at each frame). Finally, a flock scripted in this manner would be hard to edit (for example, to alter the course of all birds for a portion of the animation). It is not impossible to script flock motion, but a better approach is needed for efficient, robust, and believable animation of flocks and related group motions.

This paper describes one such approach. This approach assumes a flock is simply the result of the interaction between the behaviors of individual birds. To simulate a flock we simulate the behavior of an individual bird (or at least that portion of the bird's behavior that allows it to participate in a flock). To support this behavioral "control structure," we must also simulate portions of the bird's perceptual mechanisms and aspects of the physics of aerodynamic flight. If this simulated bird model has the correct flock-member behavior, all that should be required to create a simulated flock is to create some instances of the simulated bird model and allow them to interact.**

Some experiments with this sort of simulated flock are described in more detail in the remainder of this paper. The suc-

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 Exultation 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”

Evoflock: evolved inverse design of multi-agent motion

Craig Reynolds
unaffiliated researcher
cwr@red3d.com

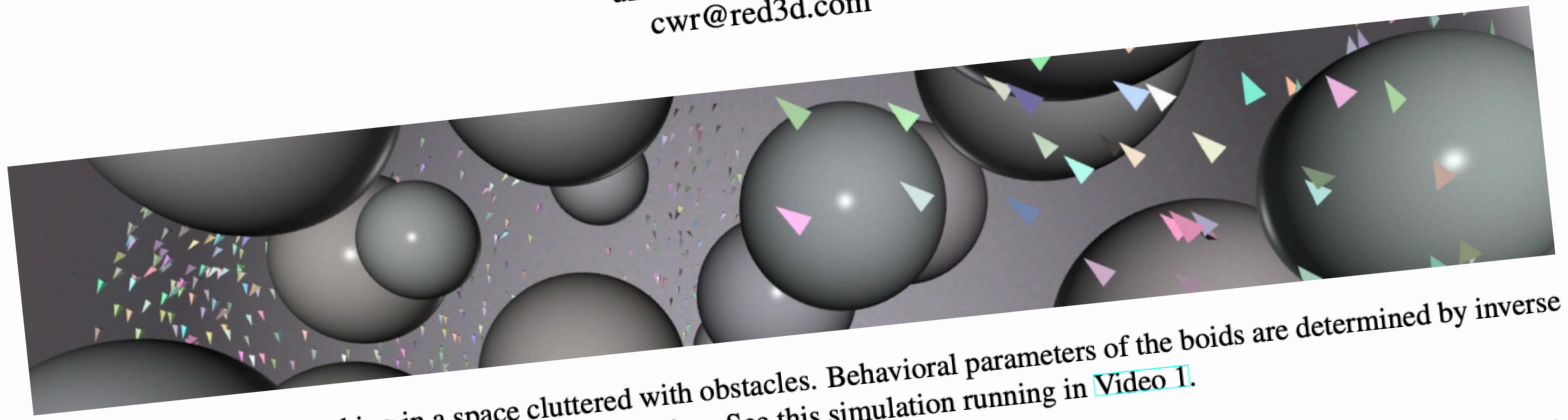


Figure 1: 1000 boids flocking in a space cluttered with obstacles. Behavioral parameters of the boids are determined by inverse design, using multi-objective evolutionary optimization. See this simulation running in Video 1.

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 Related Work). These allow creating simulations of flocks and other group motions, which most observers will recognize as some form of flocking. This paper will refer to bird 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.

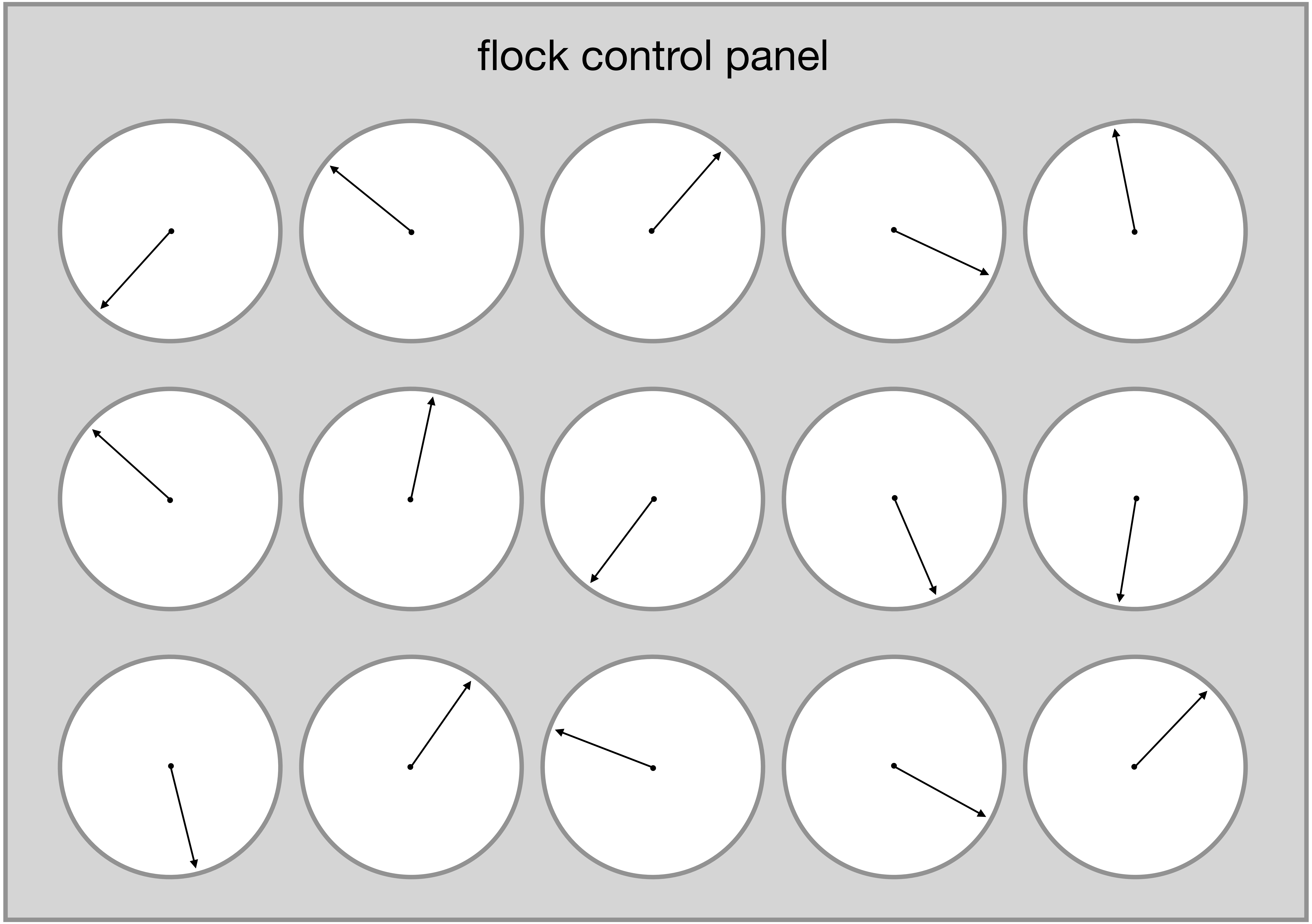
For a user of this optimization framework, it is convenient that the flock model is treated as a black box. Almost no restrictions are imposed on a preexisting model by the optimization framework. There are no restrictions on model design, source code availability, or the programming language used. (As could be an issue if optimization required language tools such as automatic differentiation (Baydin et al., 2018).)

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, Jennifer Luo at UW.
- 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 programing: steering programs form scratch

Difficult to find gradient of chaotic system

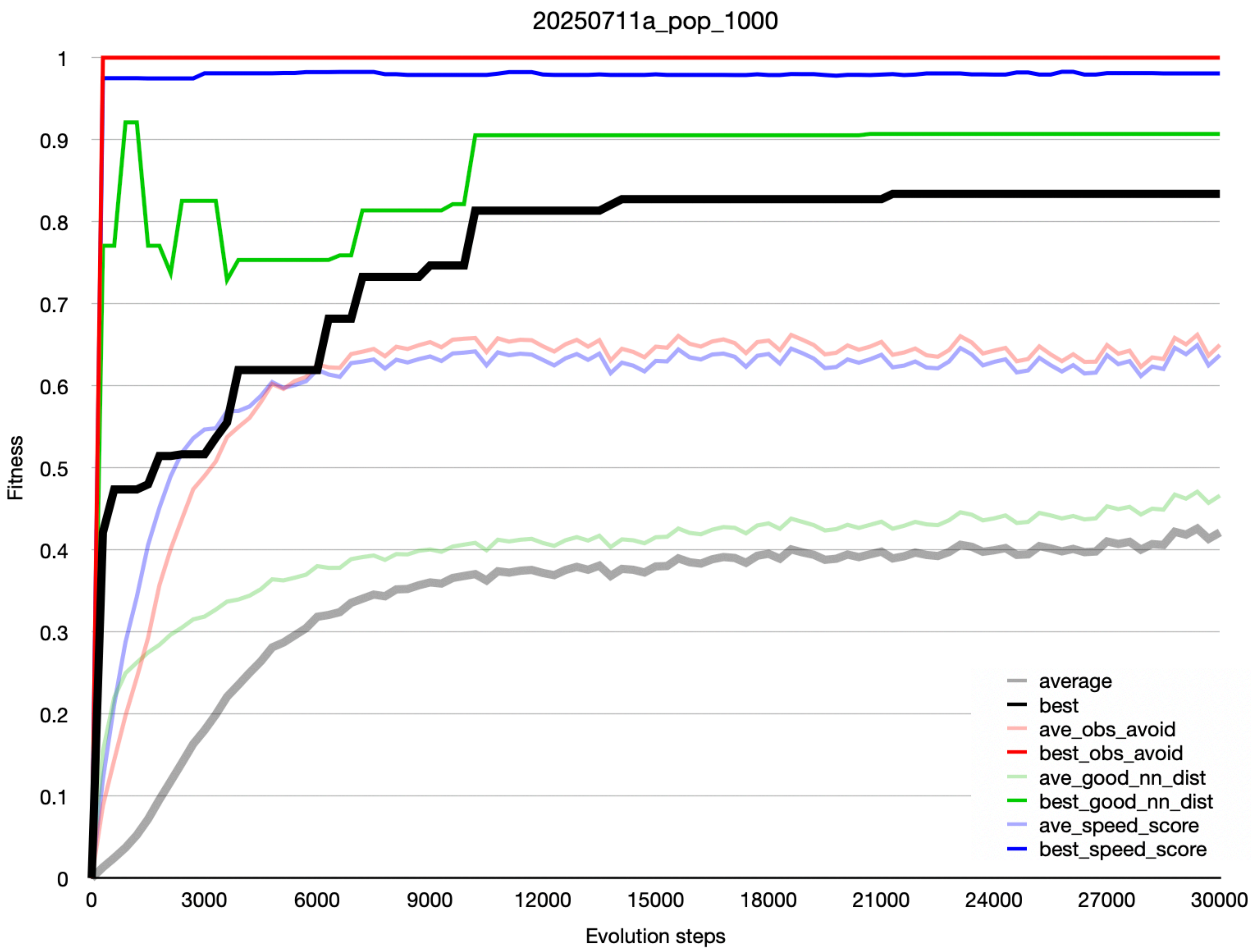
Multi-agent systems are chaotic.
Finding a principled gradient is hard.
That analysis is ongoing.

In parallel, using evolutionary optimization

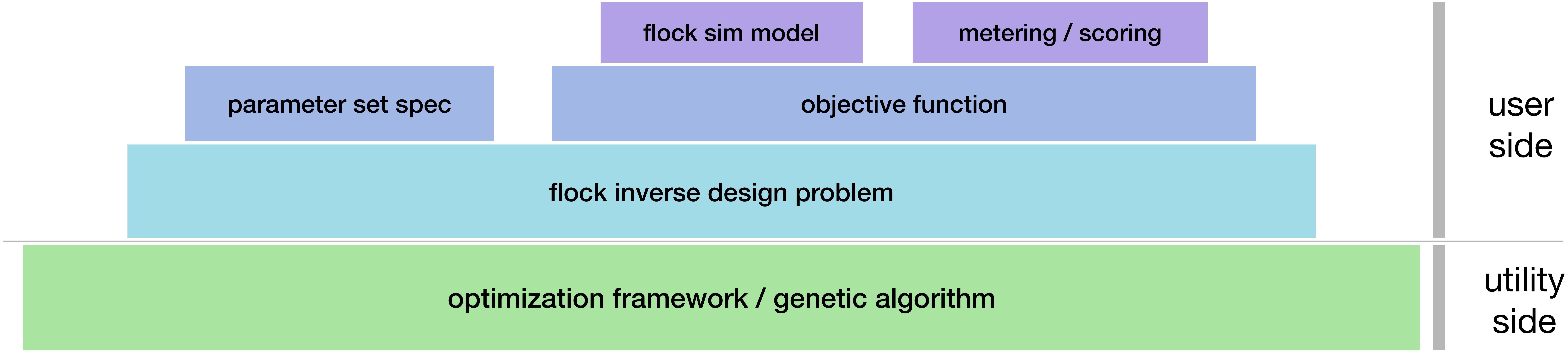
Genetic algorithm works moderately well.
Genetic programming *really not* working, so far.


strong
weak
expensive
affordable
Brooklyn
Firth of Forth
Natchez Trace Parkway
popsicle sticks
SIMPLE
EASY
BRIDGE
Tacoma Narrows
Tay
metal tubes and wood deck
rickety wood bridge

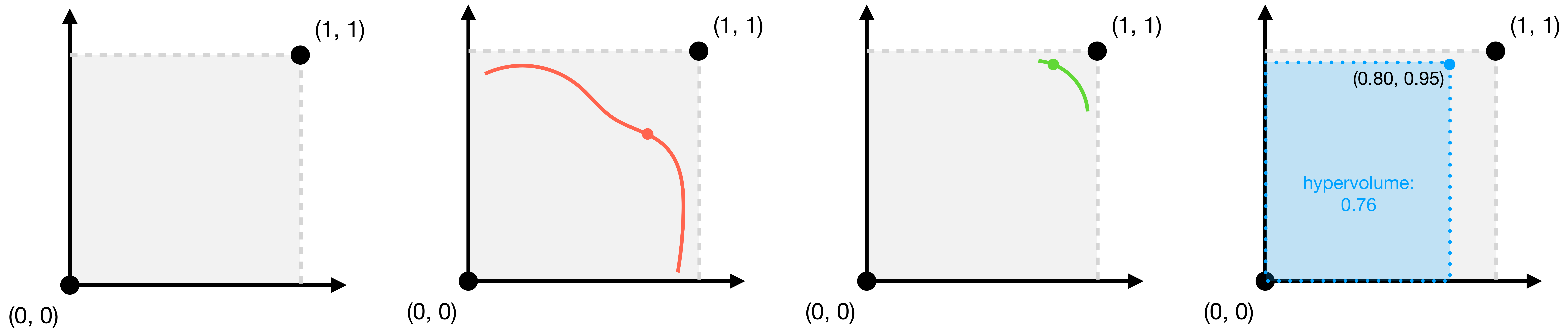
EvoFlock fitness (and multiple objectives)



EvoFlock components



Normalized multi-objective fitness space.

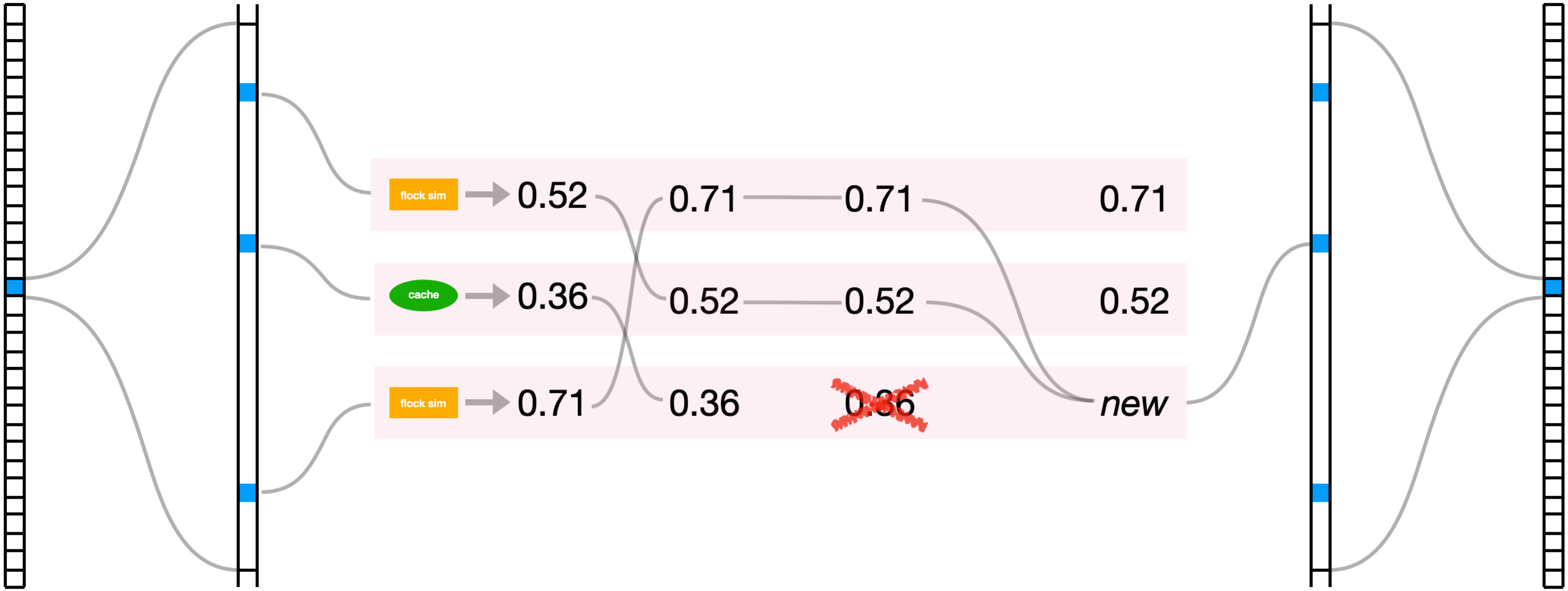


Pareto front for two mutually conflicting objectives.

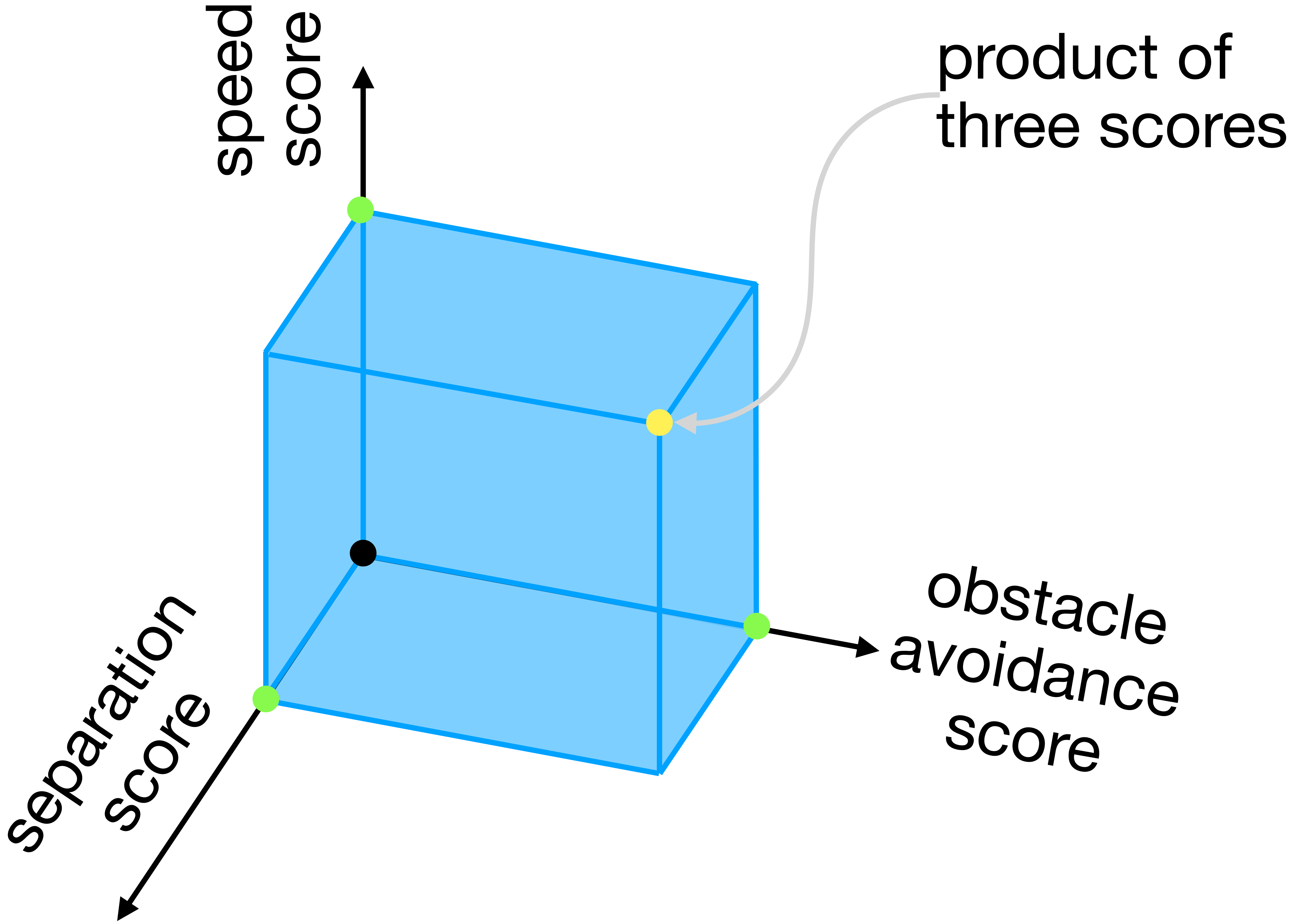
Pareto front for two "mostly compatible" objectives.

Scalarization (area) for two objectives.

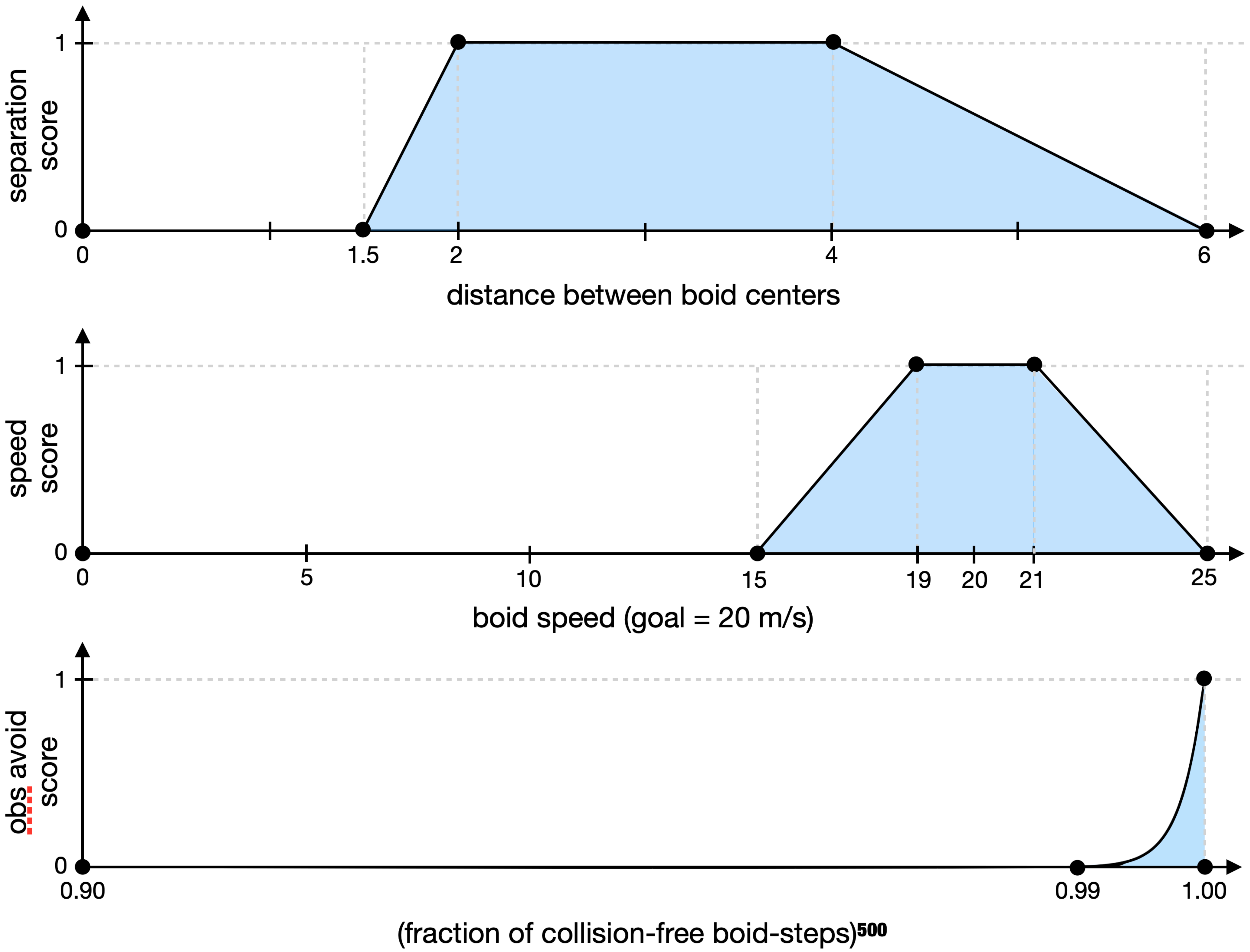
Single step of steady-state genetic algorithm (SSGA)



Hypervolume of three objectives



Score weighting functions: separation, speed, avoid



Thank you!

contact:

cwr@red3d.com

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

these slides:

~~https://www.red3d.com/cwr/presentations/20250221_UCSC_camo_flocks.pdf~~