CS 189: Autonomous Robot Systems

Spring 2018, Fridays 1-4pm, Pierce 301

More is Better!

Introduction to Multi-Robot Systems

- Why Multiple Robots?
  - **Parallelism:** Many robots can accomplish the task faster
  - **Redundancy:** Hazardous environment with chances of losing robots
  - **Required:** Too difficult to do with a single size robot
  - **Complex Tasks:** Need several specialized robots
  - **Real-time Requirements:** Monitor large areas, respond quickly

Introduction to Multi-Robot Systems

- Why Multiple Robots?
  - **Parallelism:** Many robots can accomplish the task faster
  - **Redundancy:** Hazardous environment with chances of losing robots
  - **Required:** Too difficult to do with a single size robot
  - **Complex Tasks:** Need several specialized robots
  - **Real-time Requirements:** Monitor large areas, respond quickly

Example Applications ***(which aspect do they focus on?)***
- Exploration of a abandoned mine to construct a map
- Searching for survivors and bringing them back to safety
- Locating and removing mines from a landmine field
- Managing an orchard: Picking fruit in an orchard, pesticide application, watering
- Sorting different sized parts or rubble, doing tasks in an automated factory
- Tracking and capturing an intruder
- Automated warehouse

Introduction to Multi-Robot Systems

- Why Multiple Robots?
  - **Parallelism:** Many robots can accomplish the task faster
  - **Redundancy:** Hazardous environment with chances of losing robots
  - **Required:** Too difficult to do with a single size robot
  - **Complex Tasks:** Need several specialized robots
  - **Real-time Requirements:** Monitor large areas, respond quickly

- **How do we make Robots Cooperate Effectively?**

Centralized ⟵ Semi-Centralized ⟶ Decentralized

Architectures for Coordination

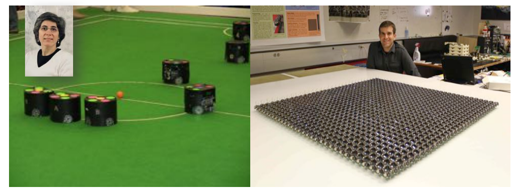
Centralized
- Global Controller with Global View
- *Good for* Tightly-coupled tasks, Efficiency, Adversarial
- *Good for* Small Teams (exception: Kiva!)
- Requires: High Bandwidth/Computation/Sensing (at least for Leader)

Middle Ground
- Try to approximate the effect of a centralized system
  - Supervisor and Team (supervisor acts as global controller)
  - Hive-based (homebase or rendevous to deposit information)
  - Role-based coordination (pre-decide responsibilities)
- When? Communication is available but slow or limited range.

Decentralized
- No one has a full world view
- Independent acting robots (purely local or no communication)
- Good for large distributed teams (no centralized bottleneck/failure)
- Often biologically-inspired (swarm intelligence)

Goes Beyond Robots….

Two Example Systems



Robot Soccer Competition
Small Size Leagues
Centralized

Kilobot Project
Collective Complexity
Decentralized

Soccer as a New Grand Challenge

- By the year 2050,
  Develop a team of *fully autonomous humanoid robots* that can win against the human world soccer champion team

What makes Soccer Different From Chess?

- It is a Game!
  - **Dynamic** and **Adversarial**
- But lots of differences too
  - Not Symbolic (In AI, Math is easier than Vision)
  - Not turn taking (harder for Game heory)
  - Distributed and Multi-agent (cooperation)
- Embodied Intelligence
  - We still understand very little about how to make "physical" systems that operate in our world
  - *Moravec's Paradox*

What makes Soccer Different From Chess?

- It is a Game!
  - **Dynamic** and **Adversaria**
- But lots of differences too
  - Not Symbolic (In AI, Mat
  - Not turn taking (harder f
  - Distributed and Multi-ag
- Embodied Intelligence
  - We still understand very little about how to make "physical" systems that operate in our world
  - *Moravec's Paradox*

But Also Different from
Other Robotics Challenges
e.g. DARPA Challenges
- Static & slow environments
- Limited "strategy" (AI) needed
- Single robots

The Robocup Challenge

- History
  - 1993 conception, 1997 first tournament (Japan)
  - Goal is to implement full FIFA regulations (even Red Cards!)
- Big Challenge for AI
  Attack by dividing into different *mini-challenges*
  - **Robot design and control**
    - Small-size and build-your-own humanoid leagues
  - **Centralized Strategy in highly dynamic environments**
    - Small-size league = very fast-paced!
  - **Distributed perception and strategy**
    - AIBO (robot dog) and Nao (humanoid Robot)
  - **Playing against humans**
    - Segway leagues!

Prof. Manuela Veloso (CMU)
Prof. Minoru Asada (Japan)
and others

Incredible Experience!

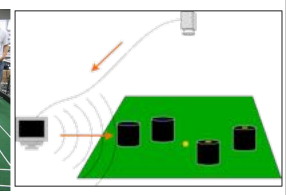
Today: How do Robots Play Soccer?

- Contrast the AI Architectures from two different leagues
  - **RoboCup small-size league**
    - **Skills, Tactics, and Plays**
    - Centralized intelligence, very fast paced
    - Ability to generate and respond to opportunities

  - **RoboCup four-legged league (briefly)**
    - **Distributed Centralized Systems**
    - Fully distributed perception and intelligence
    - Low reliability of communication
    - Emulate central control + decoupled strategies

RoboCup Small-Sized League

- Competition between two teams of 5 robots each
  - Overhead vision, single computer controller, wireless comms to robots
  - Small robot design size (20cm diam) and large field (6x4meters)
  - Very fast-paced! (robots 2m/s, ball speeds 4m/s)
  - Soccer-like Rules and Soccer-like Behavior! Video2010 (2007)



The "AI" of SSL Robot Soccer

General architecture for robotics

Vision System → world-view (format)

Strategy-System:
Based on current and opponents position
game state and past decisions,
what should each robot do?

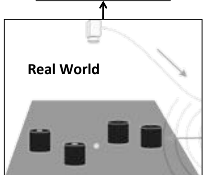
Individual robot commands

Single Robot Control:
How the robot actually accomplishes
things (move to position, avoid
obstacles, position and execute a kick)

For each robot,
wheel speeds,
discharge kicker, etc

Wireless Comms

Real World



The "AI" of KIVA too

General architecture for robotics

Wireless System → world-view (format)

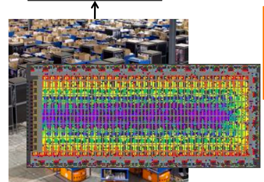
Strategy-System:
Based on current robot/rack positions
And current pending/incoming orders
What should each robot do?
[PLANNING, OPTIMIZATION, CS182!]

Individual robot commands

Single Robot Control:
Path planning, Waypoint
navigation, Localization
("GPS" flooring)

For each robot,
wheel speeds,
discharge kicker, etc

Wireless Comms



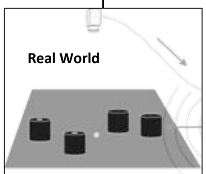
The "AI" of SSL Robot Soccer

General architecture for robotics

Vision System

world-view (format)

Strategy-System:
Based on current and opponents position game state and past decisions, *what should each robot do?*



Individual robot commands

Single Robot Control:
How the robot actually accomplishes things (move to position, avoid obstacles, position and execute a kick)

Wireless Comms

For each robot, wheel speeds, discharge kicker, etc

The "AI" of SSL Robot Soccer

General architecture for robotics

SIMULATED WORLD
Obeys the same interfaces!
Simulates robots, ball, referee
(give it an opponent controller)

world-view (format)

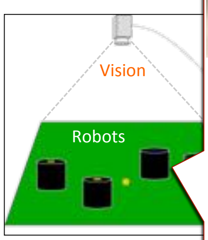
Strategy-System:
Based on current and opponents position game state and past decisions, *what should each robot do?*

Individual robot commands

Single Robot Control:
How the robot actually accomplishes things (move to position, avoid obstacles, position and execute a kick)

For each robot, wheel speeds, discharge kicker, etc

How SSL Robot Soccer is Played

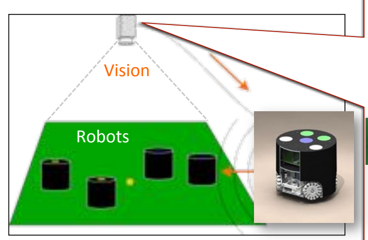


Robot Hardware

Robots must fit specs, but each team designs the whole inside.

Many COOL features
Omni wheels; kicker, dribbler, wireless
Challenges
Complex control. High performance, fault-tolerance, Modularity,

How SSL Robot Soccer is Played



Vision System
Robots, Ball, and Field all have special markings. Vision system identifies the elements and locations and transmits a "world-view"

Challenges
Large field! (2 cameras, high distortion), Small objects! (ball), Noise and ambiguity (95%), High speed (60Hhz)

STP Architecture: Skills, Tactics, Plays

- Single Robots: Behavior-based
  - **Skills:** low-level action primitives
    - Navigation, kicking, simple "behaviors"
  - **Tactics:** single-robot behaviors
    - More complex decision making
- **Plays:** team-level behaviors
  - "Pre-packaged" plans
  - Coordinate tactics of each team member
  - Select, Execute, and Monitor Plays
  - *Later: "Learn" the weights for plays*

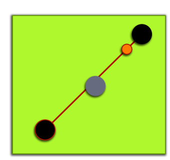
Single Robot Control:
How the robot actually accomplishes things (move to position, avoid obstacles, position and execute a kick)

Strategy-System:
Based on current and opponents position game state and past decisions, *what should each robot do?*

"STP: Skills, Tactics and Plays for multi-robot control in adversarial environments." Browning, Bruce, Bowling, and Veloso, 2005.

What is a Skill?

- Basic "Behaviors"
  - GotoBall, ApproachBall, PullBall
  - FaceTarget , DriveToGoal
  - Kick, Dribble
  - SpinAtBall, ReceiveBall
  - GotoPoint, NavToPoint
- Skills are implemented as state machines
  - e.g. GotoPoint
- Note that there is still stuff below skills!
  - e.g. Obstacle free navigation, control for kicker and dribbler



What is a Tactic?

- Top-level Single Robot behavi
  - Tactics call skills to generate commands
- Can be quite complex!
  - e.g dribble_to_position <co or defend_line <x1 y1 x2 y2
  - A robot continues executing otherwise

Active tactics (involve "ball" possession):
- shoot
- steal <x,y>
- clear
- dribble_to_region <region>
- Etc....

Non-active tactics:
- position_for_loose_ball <region>
- position_for_rebound <region>
- position_for_pass <region> <region>
- defend_line <x1,y1,x2,y2, ...>
- block <min,max,side>

Example Tactic: Shoot!

```
SHOOT TACTIC
bestscore = 0

(score,target) = evaluation.AimAtGoal()
bestscore = score
setSkillCommand(Kick, target)

Foreach teammate j do
    (score, target) = evaluation.DeflectOffTeammate(j)
    if (score > bestscore) then
        setSkillCommand(Kick, target)
        bestscore = score

If (bestscore < THRESHOLD) then failed
else sendSkillCommand
```

Choose best option
= shoot directly on goal
= OR deflect off playmate!

Evaluator
= Does the geometry calculations to decide what is a good option

Skill
= Kick at a target may first involve repositioning relative to the ball

Plays: Multi-Robot Plans

- **Plays = Multi-Robot Coordination**
  - Skills+ Tactics = Strong Suite of Single Robot Behaviors
  - But the world moves very fast........(traditional AI planning too slow)
  - Plays provide strategic control of the entire team
    - Simple language for describing plays, including "set plays"
    - *Can think of plays as prepackaged "plans"*
- What constitutes a Play?

Roles:
- Provides four roles, which are assigned to robots on initiation
- Each role is a sequence of tactics with implicit synchronization ("plan")

Applicability conditions (~ PRECOND)
- Specify when the play can be initiated

Termination conditions (~ EFFECTS)
- Specify when the play should stop
- Four types: succeeded, failed, completed, aborted

Example Play 1

- PLAY Naive Offense
  - APPLICABLE offense
  - DONE aborted !offense
  - ROLE 1
    - Shoot
  - ROLE 2
    - defend_point (-1400 250) 0 700
  - ROLE 3
    - defend_lane (B 0 -200) (B 1175 200)
  - ROLE 4
    - defend_point (-1400 -250) 0 1400

Example Play 2

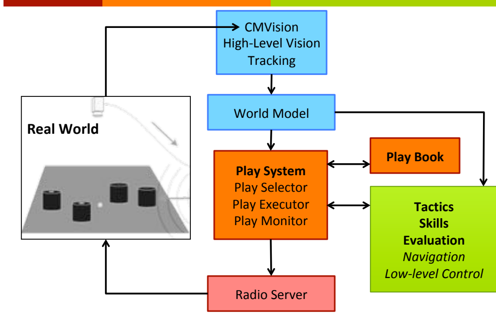
PLAY Two Attackers, Pass
APPLICABLE: offense in_their_corner (predicates)
DONE: abort !offense
OROLE 0 closest_to_ball (opponent)

- ROLE 1
  - pass 3
  - Mark-opponent o from_shot
- ROLE 2
  - block 320 900 -1
- ROLE 3
  - position_for_pass (R (1000 0) (700 0) 500) (implicit sync w passer)
  - receive_pass
  - shoot
- ROLE 4
  - defend_line (-1400 1150) (-1400 -1150) 1000 1400

Play Book and Play Executor

- **Play Book**
  - Library of plays available to the team *(must be easy to change!)*
  - Each play can be given a weight *(can learn the weights! Use a simulator)*
- **Play Selection**
  - Find all applicable plays
  - Choose plays according to their weights
    - Choose the highest-weight play? Choose probabilistically?
    - Adapt play weights based on past success/failure!
- **Play Executor and Monitor**
  - "Interprets" the play by turning it into real robot commands
  - Monitors how well things are going (e.g. termination conditions)
  - "Hysteresis" (switch to take advantage of sudden opportunities, but not too often)

Wrapping it up: STP Architecture



RoboCup Four-legged League

- Different Model: Fully Distributed Team
  - 2 teams of 5 Sony AIBO robots, Field size: 7.5 x 5m
  - On-board perception, cognition, and action
  - Wireless networking used for communication
  - 208x160 camera, 60 degree FOV (Video)

Movies

- The Game
- The Dog's perspective

Distributed Playbook

- Challenges: Coordination is Difficult
  - Each robot has only limited view of the world (distributed perception)
  - Communication is low reliability and low bandwidth/high latency
  - Robots are slow and less reliable
- But: Coordination is still Essential
  - Example: all robots going for the ball or leaving the goal undefended
  - Conflicting decisions, lack of knowledge on opponent

Distributed Playbook

- Challenges: Coordination is Difficult
- But: Coordination is still Essential
- Distributed PlayBook: The Team Leader chooses the Play
  - Play Selection runs on one robot arbitrarily chosen as leader
  - Leader chooses the highest-weight applicable play and broadcasts periodically
  - Plays tend to be longer in duration (minutes instead of seconds)
  - Plays depend on roles – loosely coupled behaviors of different robots (much like real soccer)
- *This requires a world model beyond what the leader can see!*
  - Use communication to share world views amongst all robots
  - Leader uses it to decide play, others use it to localize
  - But world view is now uncertain
    - attach a confidence and priority level to every object

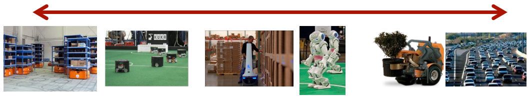
From AIBOs to Humanoids

- New Platform
  - Similar to AIBO in perception and coordination challenge
  - But locomotion and manipulation are huge challenges (biped)
    - Video 2013

Introduction to Multi-Robot Systems

- Why Multiple Robots?
  - **Parallelism:** Many robots can accomplish the task faster
  - **Redundancy:** Hazardous environment with chances of losing robots
  - **Required:** Too difficult to do with a single size robot
  - **Complex Tasks:** Need several specialized robots
  - **Real-time Requirements:** Monitor large areas, respond quickly
- **How do we make Robots Cooperate Effectively?**

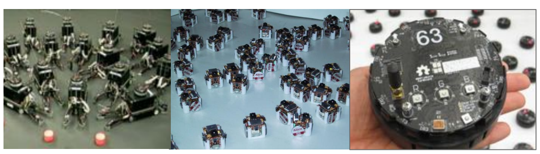
Centralized — Semi-Centralized — Decentralized



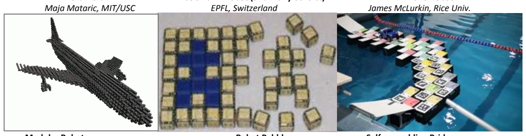
Swarm Intelligence

3 KEY FEATURES
Individuals << Collective
No Leaders
Simple Local Rules of Interaction

Swarm Robotics!

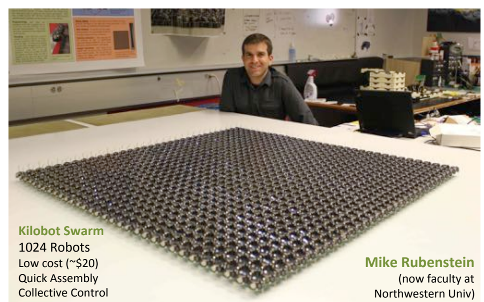


Nerd Herd *Maja Mataric, MIT/USC*
Alice Swarm Robot (and many others!) *EPFL, Switzerland*
R-one Robot *James McLurkin, Rice Univ.*



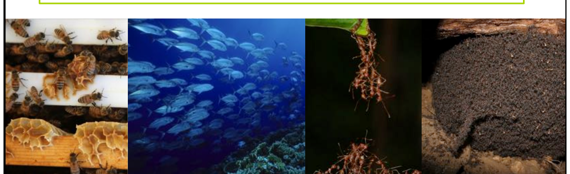
Modular Robots *Kasper Stoy, SDU/ITU*
Robot Pebbles *Daniela Rus, MIT*
Self-assembling Bridges *Yim/Kumar, UPenn*

The Kilobot Project



Towards a "Kilo" of Robots

- What would it take to create (build and program) our own artificial collectives of the scale and complexity that nature achieves?



Animal groups with tens, to thousands, to millions of individuals

Building the Swarm

- What would it take to create (**build** and program) our own artificial collectives of the scale and complexity that nature achieves?

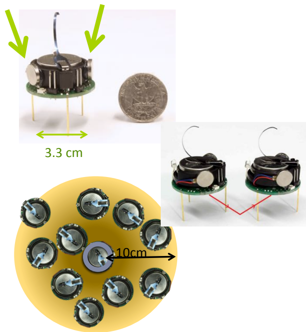
Challenges of Scaling Up

Manufacturing
- *What is a "minimal" swarm robot? (open question)*
- Simple computation, locomotion, sensing, communication
- Cost $1 → $1000
- Assembly 1 min → 17 hours

Operations
- *Need "hands-off operation" (charging, programming)*
- Individual operations no longer possible
- A Power Switch: 4 seconds → > 1 hour!

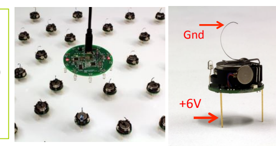
A Single Bot

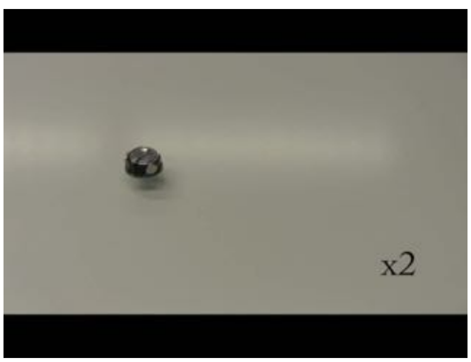
- **Computation**
  - Microprocessor
  - 32K, 8 mhz, C programming
  - Battery 3-24 hours
- **Locomotion**
  - Vibration (cell phone)
  - Low cost!
  - But slow speed (1 cm/s)
- **Communication**
  - Reflection off surface
  - IR Receiver/Transmitter
  - 30 kb/s upto 3 robots away
  - Distance, but not bearing
- **Sensing**
  - Ambient light sensor



A Single Bot => Swarm

Scalable
- Charge .. As a group
- Programming .. As a group
- Turn on .. As a group
- Build .. As a group





Programming the Swarm

- What would it take to create (build and **program**) our own artificial collectives of the scale and complexity that nature achieves?

Challenges of Scaling Up

Programming
- *What global behaviors are possible from local interactions?*
- Bio-inspired: Decentralized, Robust, Scalable
- But how to generalize? *Compile complex behavior?*

Mathematical Models
- How do we prove things about collective behavior?
- Simple algorithms → Complex analysis

  (Control theory, Distributed Computing, Graph Theory & Geometry)

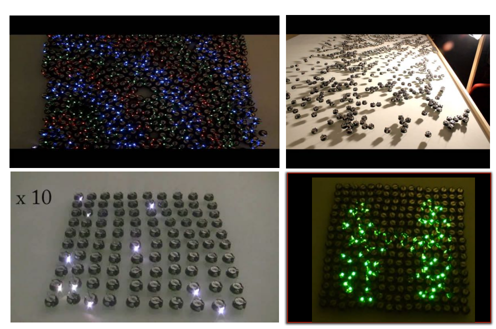
Programming the Swarm

- What would it take to create (build and **program**) our own artificial collectives of the scale and complexity that nature achieves?

Simple Behaviors => Complex Collectives

Simple Collective Behaviors

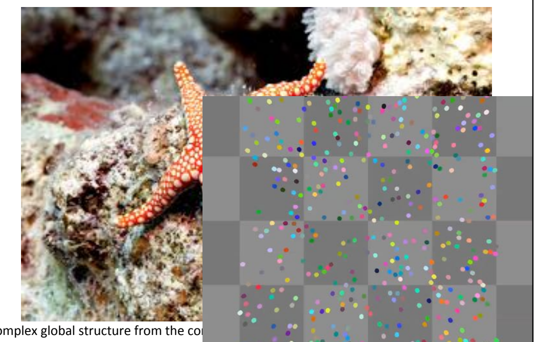
Gradient Patterns, Synchronization, Light Following, Coordinate Systems, and more…



Shape Self-Assembly

Complex global structure from the composition of many simpler collective behaviors

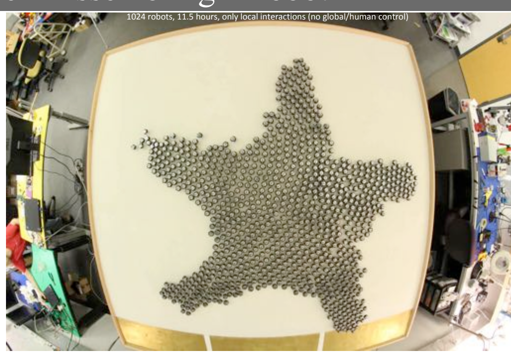
Shape Self-Assembly



Complex global structure from the co

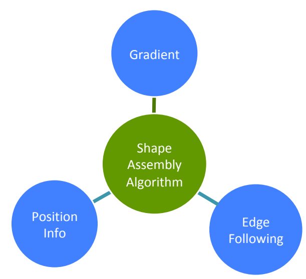
Mike Rubenstein, Wei-Min Shen (USC), 2010

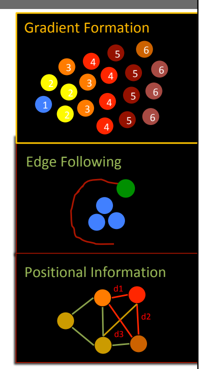
Self-Assembling Kilobot



Mike Rubenstein, Alex Cornejo, Radhika Nagpal, *Science*, Aug 2014

Simple => Complex Collective Behavior





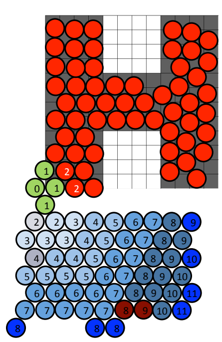
Self-Assembly Algorithm

A Robot Pool and a Seed

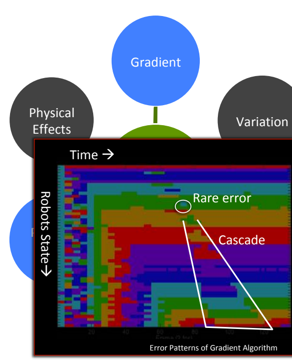
1. Seed creates a **Gradient**
2. Any robot at the ***edge*** can move
3. **Edge-follow** and constantly compute **positional information**

Robots join the Shape

1. Edge follow until
   About to exit the shape
   Or close to shape neighbors
   with hop counts of N
   (Shape forms in layers)
2. Positional Information
   Compute position in the shape
   Transmit to moving robots



Simple => Complex Collective Behavior



Mathematically Provable

The idealized algorithm can form *any simply connected shape*

But scaling up to a 1000 robots requires more!

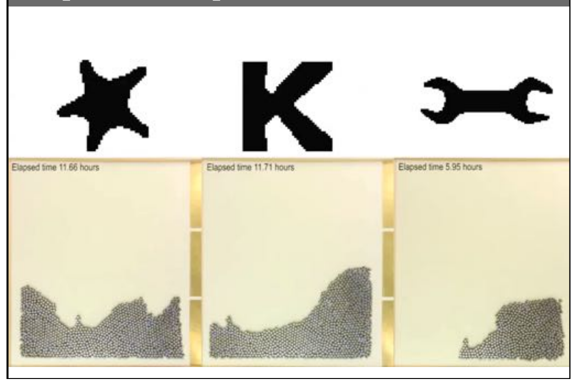
- **Physical effects** (e.g. pushing)
- **High variation** (e.g. speed)
- **Rare events** (e.g. sensors)
- *=> Cascades*

Need new algorithmic strategies *e.g. cooperative error-detection*

Also, need better understanding *of emergent properties.*

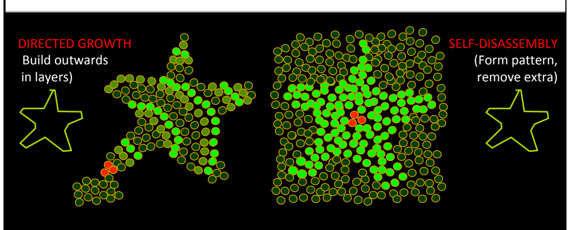
Close up shots

Simple => Complex Collective Behavior



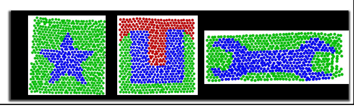
"Synthesis" of Self-Assembly

Two Methods

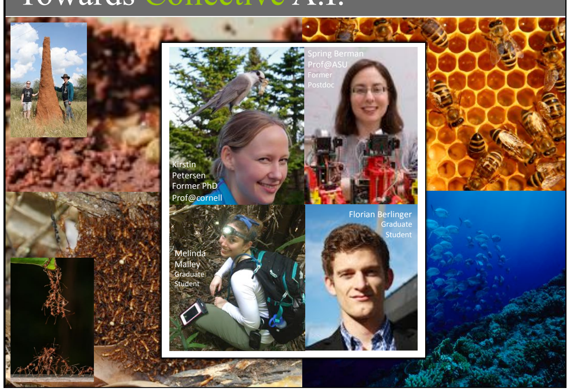


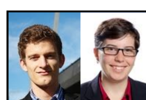
GRADIENTS | POSITIONAL INFORMATION | EDGE FOLLOWING | PHOTOTAXIS | CONSENSUS

Melvin Gauci, Mike Rubenstein, Radhika Nagpal, DARs 2016



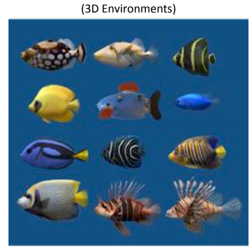
Towards Collective A.I.





Some New Directions

Underwater Swarms
(3D Environments)



Heterogeneous Swarms
(Collective Perception)