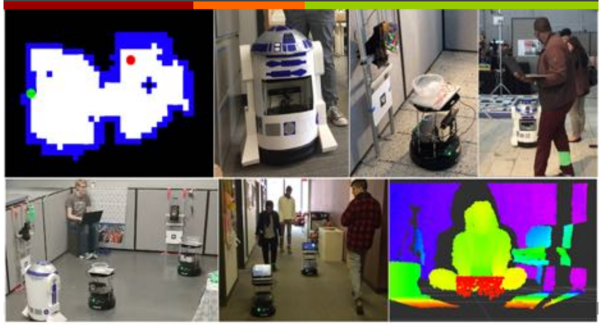
CS 189: Autonomous Robot Systems

Spring 2020, Fridays 9-11:45am, Pierce 301



CS189 Robotics In Review

Robot Reflexes

Motion (Kinematics)

Vision (RGB and Depth)

Feedback Control (PID)

Behavior-based Programming (FSM)

Higher-Level Reasoning

Navigation
- Path Planning
- Localization
- Mapping

Planning
- *CS182*

Machine Learning
- *For Robotics*

Applications

Manipulation

Multi-Robot Systems

Ethical Issues

Human-Robot Interaction

Bio-inspired Robotics

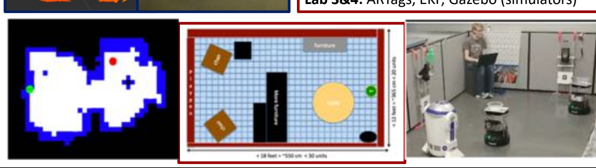
Embodied AI: Real World is unstructured and complex!
(+ Technical Skills: ROS, OpenCV, debugging hardware…..)

CS189 Robotics In Review

LABS AND ASSIGNMENTS

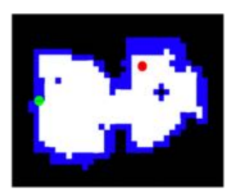
Pset 1: Roomba (behaviors)
Pset 2: Wanderer (depth + collision avoidance)
Pset 3: Follower (RGB and feedback control)
Pset 4: Mapping (Occupancy Grid)
Final Project: *Candy Store! (Applications)*

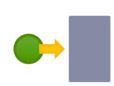
Lab 1&2: Motion and Vision (depth/color)
Lab 3&4: ARTags, EKF, Gazebo (simulators)



Agenda

- Lecture: Multi-Robot Systems
- Before:
  - How is Pset 4a going?
    - Issues?
  - Talk about Pset 4b
    - Marking Obstacles
    - Frontier Nodes
  - Optional: Demo date?





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
- Locating and removing mines from a landmine field
- Searching for survivors after a natural disaster
- Managing an orchard: Picking fruit in an orchard, pesticide application, watering
- Automated factory: Sorting different sized parts or rubble, doing repetitive tasks
- Hospital or hotel delivery robots
- A Fleet of Self-driving Cars!

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 or Highly Structured Environments
- Requires: High Bandwidth/Computation/Sensing (at least for Leader)

Middle Ground: Semi-Centralized
- 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 (peer-to-peer system)
- 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....

Robot Cooperation!

Robot Soccer Competition
Small Size Leagues
Centralized

MAGIC Competition
Semi-Centralized

Kilobot Project
Collective Complexity
Decentralized

Semi-Centralized Cooperation

- Most "Intuitive" Choice for an Application
  - If each robot can do the basic thinking on its own, like a human worker would.
  - Then I can put a supervisor (maybe human) in charge of allocating tasks and adjusting when necessary
  - Intuitive: matches what we already do…

  (e.g. UPS managing a delivery fleet of trucks)
- Robotics Challenges
  - What can we expect a robot to be able to do reliably on its own?
  - How do we communicate with robots?

MAGIC 2010 Competition

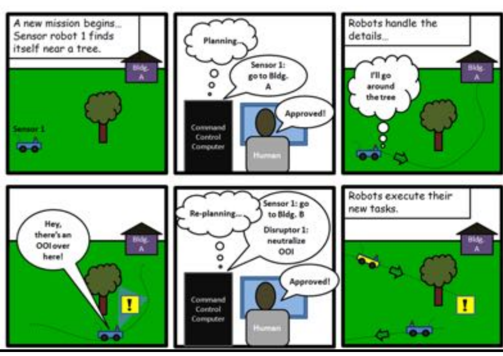
- **M**ulti **A**utonomous **G**round **R**obot **I**nternational **C**hallenge
  - Sponsored by US and Australia, held in Adelaide Nov 2010.
  - Explore and map 500x500m indoor and outdoor environment
  - Success: Build a map, find and neutralize simulated “explosives”, human assistance allowed, multiple robot tasks

(h) Satellite (a) Non-combatant (blue) & mobile OOI (red) (b) Static OOI through Doorway (c) Toxic Vat

Team Michigan (Olsen Group)

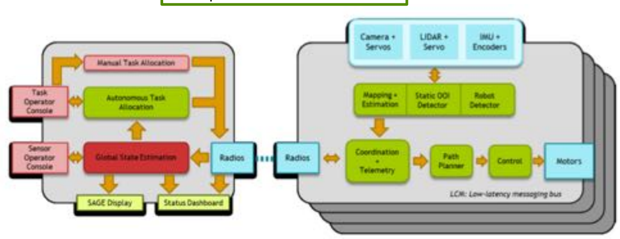
- Winning Team
  - **Robots:** 2 x more robots than any other team (14 robots, 2 humans, low cost self-designed platform)
  - **Architecture:** Decoupled Centralized Architecture:
    - Robots are oblivious to the grand plan, but responsible for their own safe operations (map making, path planning)
    - GCS (ground command) does multi-agent planning and coordination
  - **Humans:** Explicit thought given to cognitive overload
    - Real-time 3D games inspired the human interface design!

Architecture

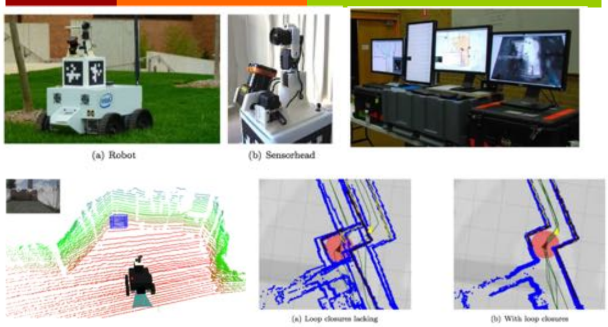


Architecture

Decoupled Centralized Architecture

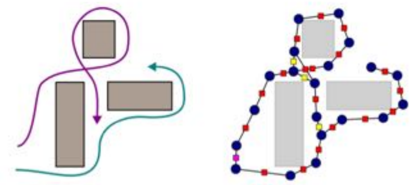


Different Parts of the System



Interesting Mapping Techniques

- Each robot constructs its own map
  - Map is stored as a graph ("factor graph")
    - Roughly speaking: nodes represent positions and edges represent the movement commands that moved the robot between nodes.
    - Add to this probability! (state estimation e.g. kalman filter) so that nodes and edges are estimates rather than fixed values.
    - *Creates many nodes!* 1 node every second



Human Interaction

- What do the humans do?
  - Most coordination is done by high-level planners

    (e.g. exploration planner sends robots to the "frontiers", map constructor does the loop closure, neutralizer planner decides actions for threats)
  - **Humans just do error correction!**
    - Fix mistaken loop closures, or suggest closures
    - Check when robots get into trouble (warning signal)
    - Confirm explosive targets for "neutralization"
  - *In spite of this, still fairly frequent interaction*
    - Estimated 4 interactions per min on average! max 5 min break.
- **Cognitive load of managing multiple robots is high….. Open area**

Robot Cooperation!

Robot Soccer Competition
Small Size Leagues
Centralized

MAGIC Competition
Semi-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, Matl
  - 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
- Single robots
- Static & slow environments
- Limited "strategy" (AI) needed

The Robocup Challenge

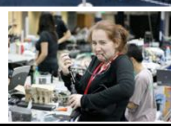
- History
  - 1993 conception, 1997 first tournament (Japan)
  - Goal is to implement full FIFA regulations (even Red Cards!)
- Big Challenge for AI
  Attack by dividing into different *mini-challenges*
  - **Robot design and control**
    - Small-size and build-your-own humanoid leagues
  - **Centralized 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!







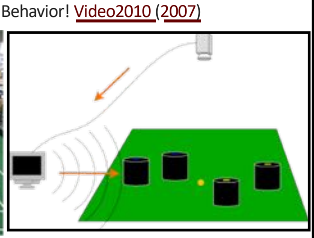

Today: How do Robots Play Soccer?

RoboCup small-size league

- **Skills, Tactics, and Plays**
- Centralized intelligence, very fast paced
- Ability to generate and respond to opportunities

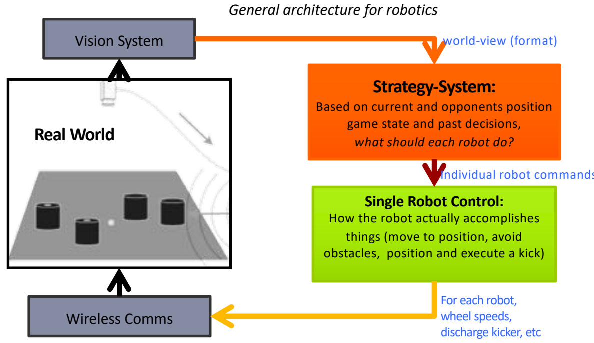
RoboCup Small-Sized League

- 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)

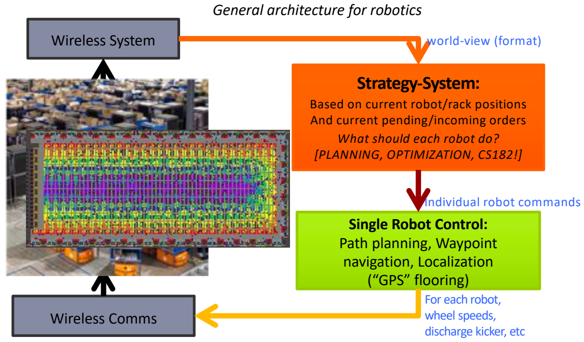
The "AI" of SSL Robot Soccer

General architecture for robotics

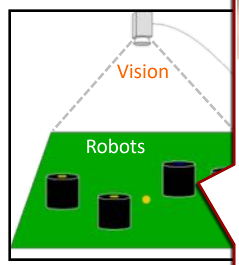


The "AI" of KIVA too

General architecture for robotics



How SSL Robot Soccer is Played



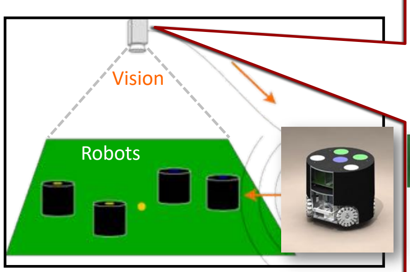
Robot Hardware

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, basic "behaviors"
  - **Tactics:** single-robot behaviors ("FSM")
    - More complex decision making

  i.e. what we covered in class!
- **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.

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/game-players 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

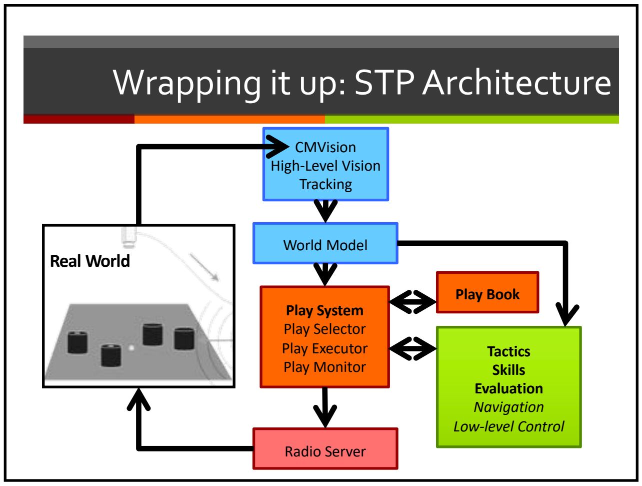
Example Play 2

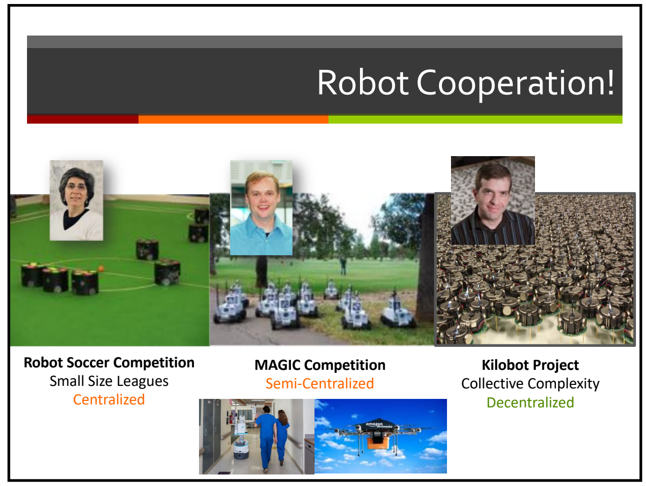
PLAY Two Attackers, Pass
APPLICABLE: offense in_their_corner (predicates)
DONE: abort !offense
OROLE 0 closest_to_ball (opponent)

- 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

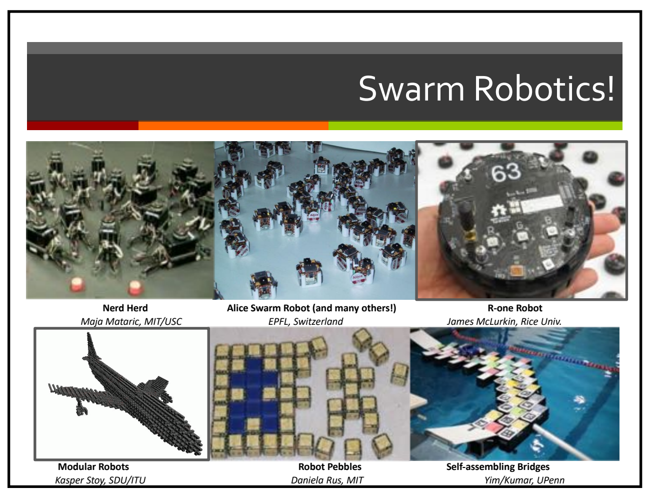
Play Book and Play Executor

- **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
CMVision
High-Level Vision
Tracking
World Model
Real World
Play System
Play Selector
Play Executor
Play Monitor
Play Book
Tactics
Skills
Evaluation
Navigation
Low-level Control
Radio Server

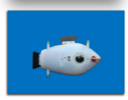

Robot Cooperation!
Robot Soccer Competition
Small Size Leagues
Centralized
MAGIC Competition
Semi-Centralized
Kilobot Project
Collective Complexity
Decentralized


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

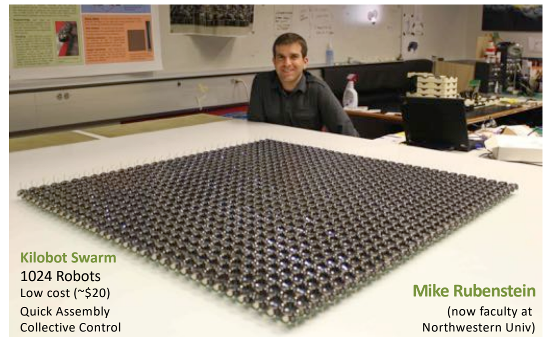
Towards Collective A.I

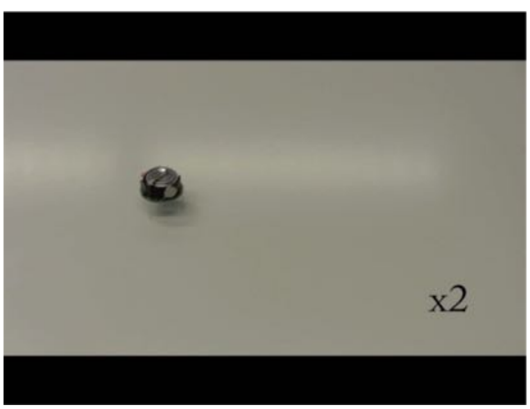
- What would it take to create (build and program) our own artificial collectives of the scale and complexity that nature achieves?



Self-organizing Systems Research Lab

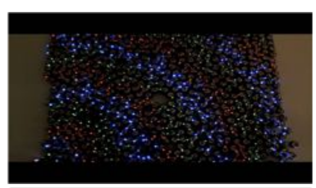
The Kilobot Project

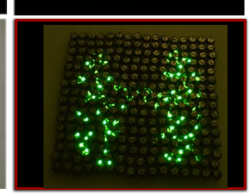




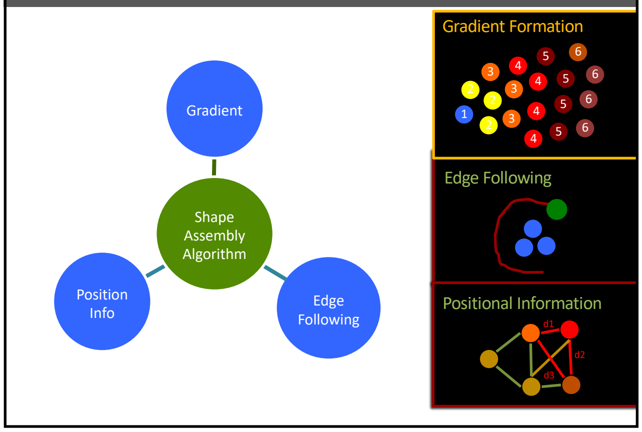
Simple Collective Behaviors

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





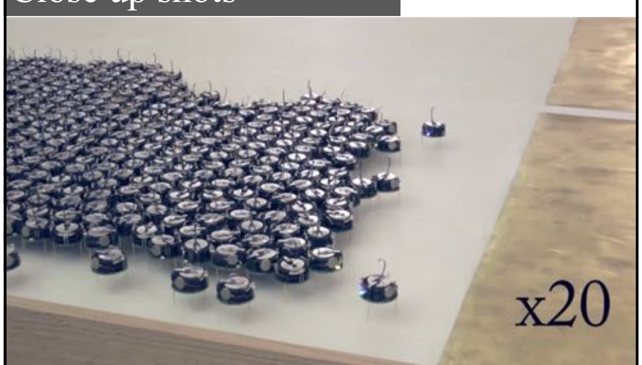
Simple => Complex Collective Behavior



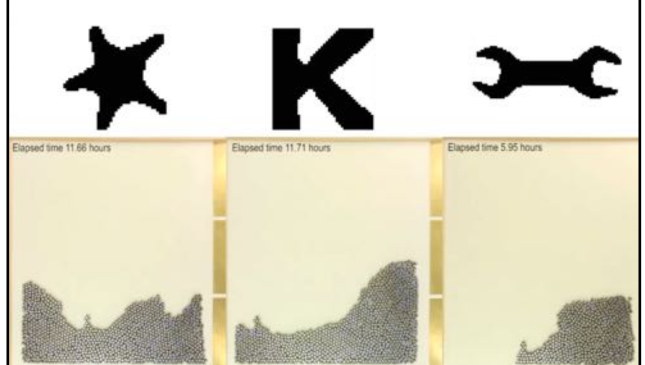
Self-Assembling Kilobot

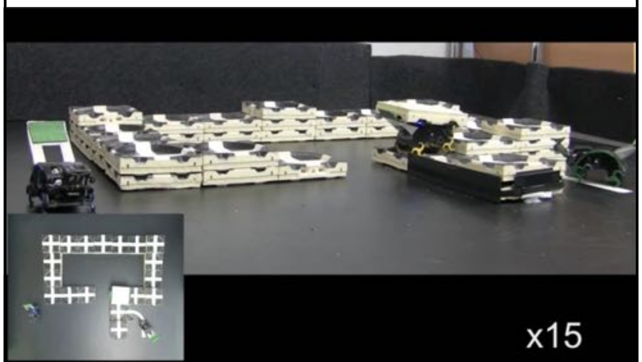
Mike Rubenstein, Alex Cornejo, Radhika Nagpal, *Science*, Aug 2014

Close up shots

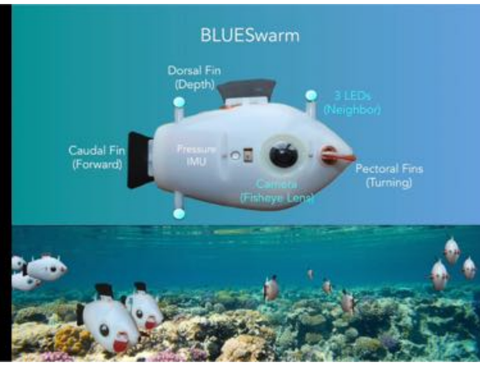


Simple => Complex Collective Behavior





Justin Werfel, Kirstin Petersen, Nagpal, Science 2014



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