From: AAAI Technical Report SS-99-05. Compilation copyright © 1999, AAAI (www.aaai.org). All rights reserved.

Algorithms for the Design of Networks of Unmanned Aerial Vehicles

Shankar S. Sastry
University of California, Berkeley CA 94720

At Berkeley we have been interested in design schemes for network of complex networks of semi-autonomous agents. These networks are characterized by interaction between discrete decision making and continuous control. The control of such systems is often organized in hierarchical fashion to obtain a logarithmic decrease in complexity associated with the design, We have used as examples three classes of systems to motivate the design approach:

1. Intelligent Vehicle Highway Systems (IVHS)
2. Air Traffic Management Systems (ATMS)
3. Unmanned Aerial Vehicles

Over the last five years or so, a group of us have developed a set of design approaches which are aimed at designing control schemes which are live, deadlock free, and "safe". Our design methodology is to be considered an alternative to the verification based approaches to hybrid control systems design, and is an interesting blend of game theoretic ideas, fault handling in a probabilistic framework, mathematical and temporal logic and planning ideas from robotics. In this talk, I will focus on design problems involved in coordinating groups of Unmanned Aerial Vehicles (UAVs). Problems to be addressed include:

1. Rapid prototyping of real time control laws: a hybrid systems design and simulation environment.
2. Mode Switching and envelope protection.
3. Vision based control for navigation.
4. Mission planning for multi-UAV missions.

The work is joint with Claire Tomlin, John Lygeros, Datta Godbole, George Pappas, John Koo, David Shim, Joao Hespanha, Omid Shakernia, and Hyoun Jin Kim.

Intelligent Control Architectures for UCAVs & Probabilistic Verification

AAAI Spring Symposium Palo Alto
March 21, 1999

1

Intelligent Control Architectures

- An architecture design problem for a distributed system begins with specified safety, efficiency objectives & aims to characterize communication, sensing & control
- Progress to date
  - developed architecture for multi-UCAV coordination and control
  - implementation on the SHIFT simulation/verification testbed
  - developed control laws for autonomous helicopters
- On-going research
  - multi-UCAV planning & coordination design

ARO DAAH04-96-1-0341 2

Verification and Design Tools

- Research Topics
  - Design Mode Verification
  - Faulted Mode Verification
  - Probabilistic Verification

Perception and Action Hierarchies

- Research Topics
  - Hierarchical Vision
  - Control Around the Vision Sensor
  - Surveillance

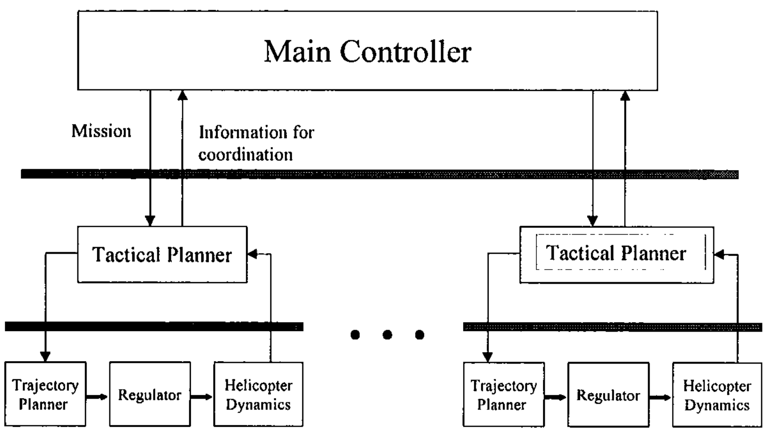
  We are designing a perception and action hierarchy centered around the vision sensor to support the observation and control functions of air vehicles
- Progress to date
  - developed target recognition and tracking algorithms
  - developed trajectory tracking controller using vision
- On going research
  - obstacle detection and avoidance algorithms
  - formation flight using active vision

A Design of Hierarchical Architecture for Multi-UAV Hybrid Systems

Jin Kim, David Shim, Shankar Sastry

5

Hierarchical System Architecture



ARO DAAH04-96-1-0341 6

Application

- **Mission**

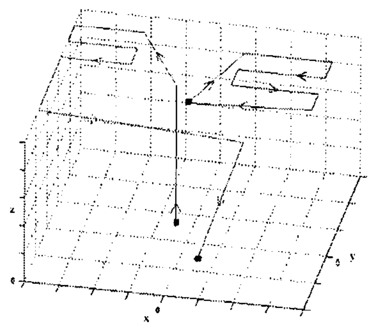
Area is swept by 2 helicopters, which should visit a sequence of targets assigned by main controller.

- **Approach**

- Tactical Planner sets a series of suitable modes to ensure the current target is reached.

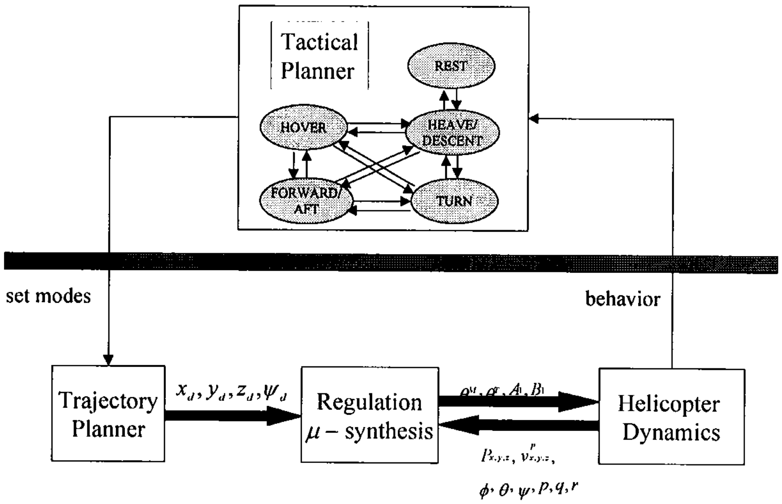
- Trajectory Planner generates desired path for helicopter.

- Regulator with suitable techniques stabilizes the helicopter system with enhanced robustness.



Desired Trajectory

System Structure for each UAV



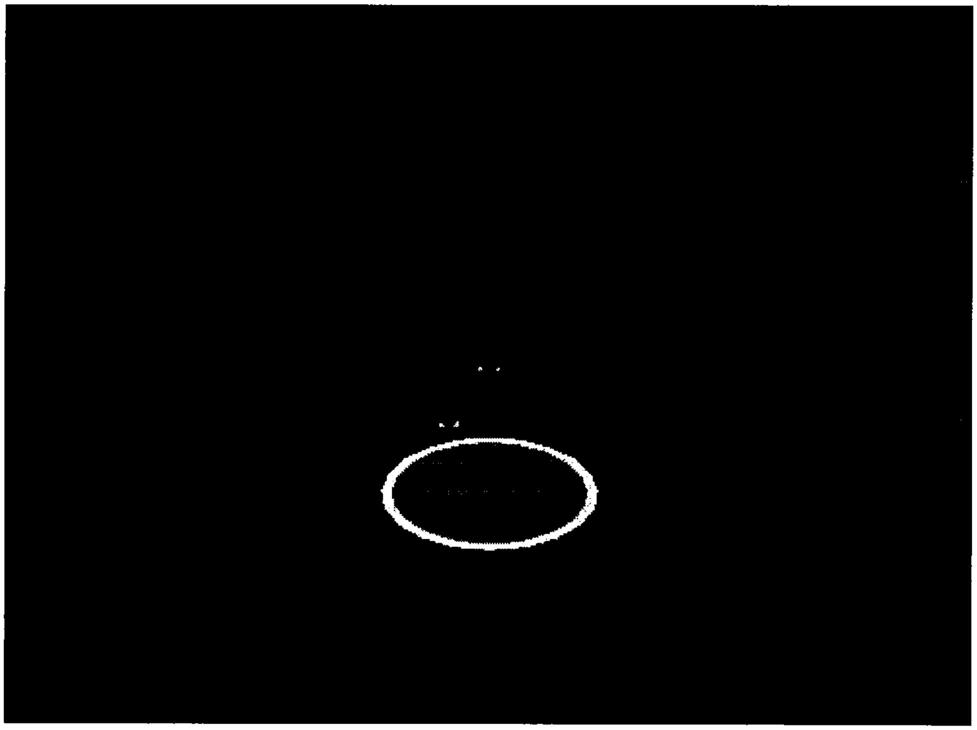
Techniques for Regulator Design

- **Nonlinear Tracking Controller**
  - nice transient and steady-state response with nominal case only
  - very sensitive to model change and disturbance
  - computationally expensive
- μ – **Synthesis Controller**
  - tradeoff between the robustness and the tracking performance
  - satisfactory robustness to model perturbation
  - models the noise characteristics of the sensor system
  - specifies the performance objective quantitatively

Tool for Simulation - SHIFT

- object-oriented : accommodates structural design -> modularity
- precise interpretation as dynamic networks of interacting hybrid systems
- can serve as language for system specification and simulation, and as means of team coordination
- Prototype-based description

```
type Helicopter {
    input ... what we feed to it                 ⎫
    output ... what we see on the outside        ⎪
    state ... what is internal                   ⎬ data model
    discrete ... discrete modes of behavior      ⎪
    export ... event labels                      ⎭
    flow ...        ⎫
    transition ...  ⎭ continuous and discrete evolution
    setup ... actions executed at create time
}
```

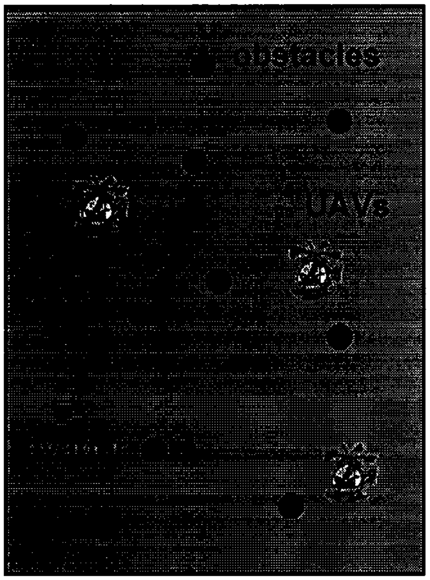


Probabilistic Pursuit-Evasion Games With Cooperating UCAVs

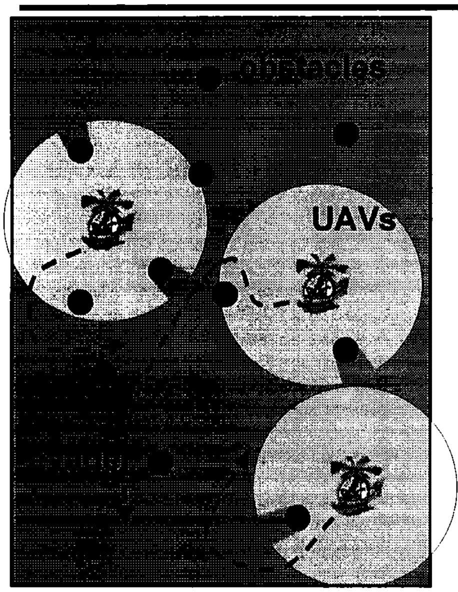
João Hespanha, Jin Kim,
Shankar Sastry,

12

The "rules" of the game



The "rules" of the game



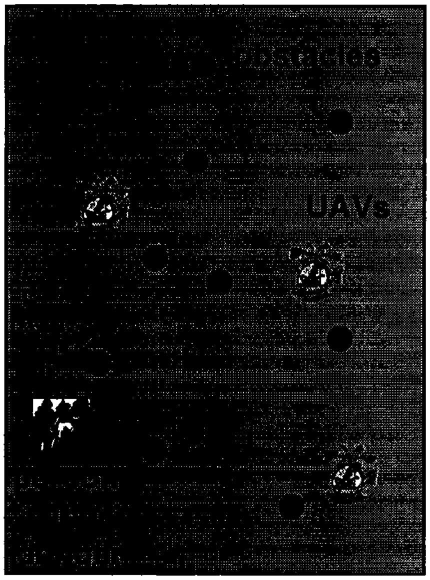
Terrain:
* with **fixed obstacles** (tree trunk like)
* not accurately mapped

UAVs (pursuers) capable of:
* **flying** between obstacles
* **seeing** a region around them (limited by the occlusions)

Evader capable of:
* **moving** between obstacles (possibly actively avoiding detection)

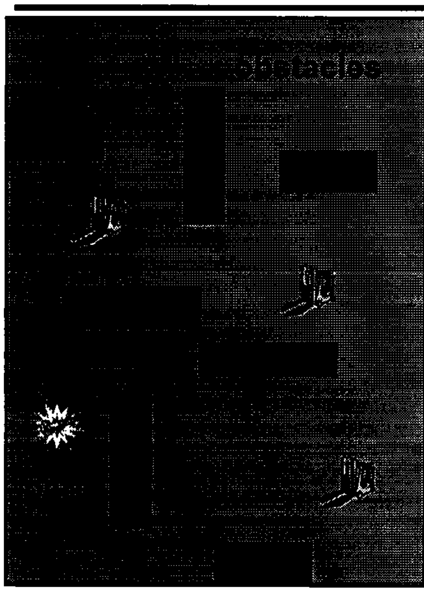
Objective:
find the evader in minimum time

Scenarios



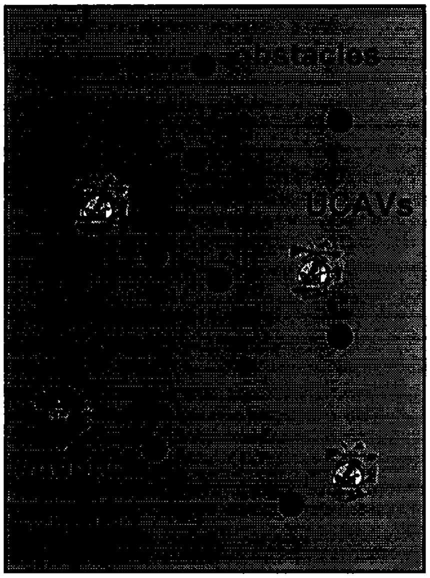
- search and rescue operations

The "rules" of the game



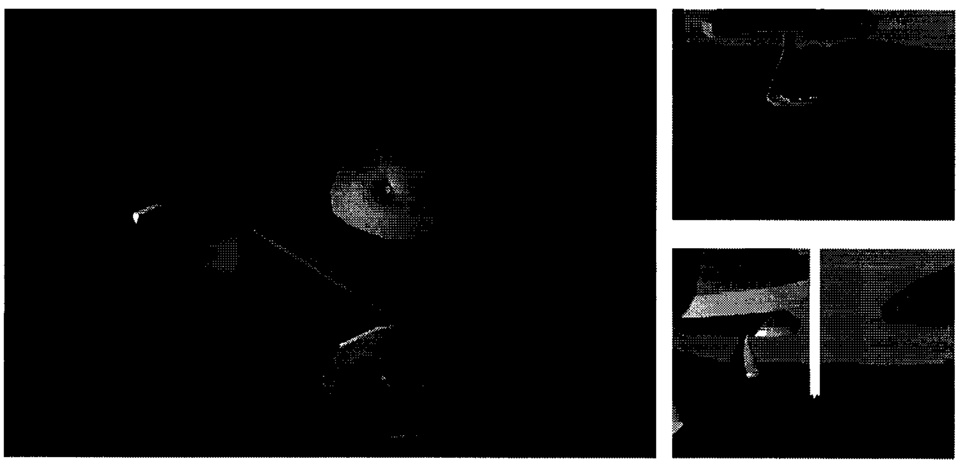
- search and rescue operations
- finding parts in a warehouse

The “rules” of the game



- search and rescue operations
- finding parts in a warehouse
- search and capture operations

The goal…



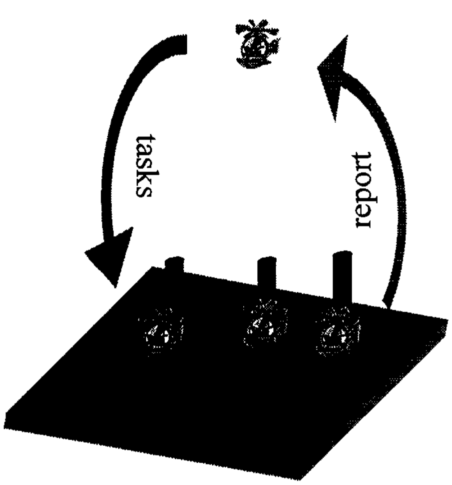
Divide and conquer...

The design of a swarm of UAVs capable of winning the game requires the solution of several problems:

- Developing **strategies** to win multi-agent pursuit-evasion games
- **Map building** by a network of agents:
  "How to combine local maps acquired by the individual agents into a global map?"
- **Navigation** (trajectory tracking & obstacle avoidance)
  "How to move the UAV from point A to point B keeping a minimum distance from neighboring obstacles?"
- **Sensing and actuation**
  "How to find the evader with the available sensors?"
  "How to model and control a single UAV?"

ARO DAAH04-96-1-0341 19

Hierarchical architecture



A **high-level planner** capable of winning the game with certain probability...

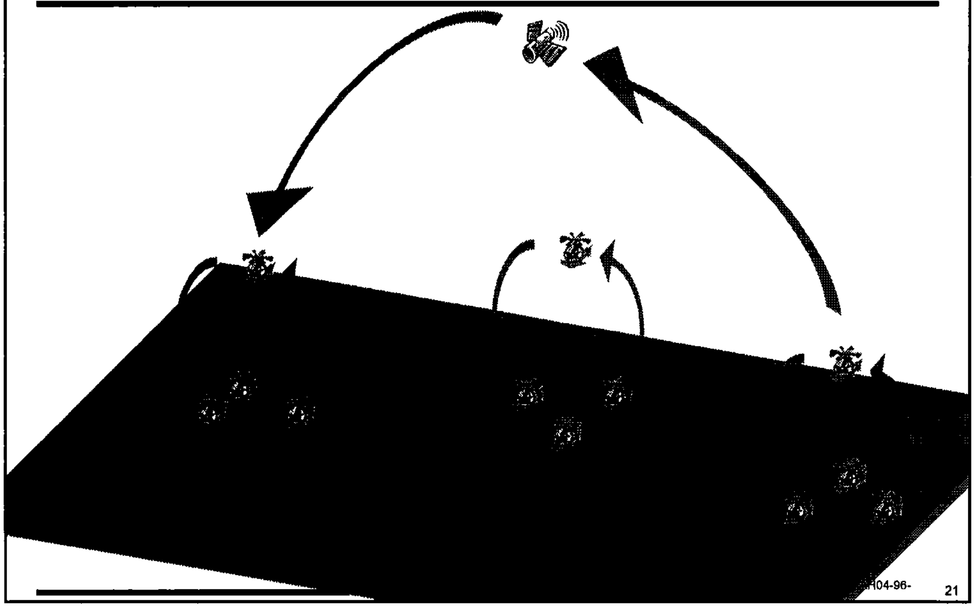
relying on

individual **UAVs** capable of low level navigation & evader detection

that rely on vision and actuation systems

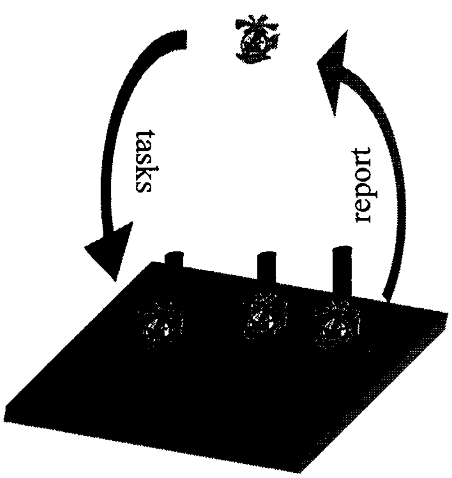
ARO DAAH04-96-1-0341 20

Hierarchical architecture



04-96- 21

Hierarchical architecture



A **high-level planner** capable of winning the game with certain probability…

relying on

individual **UAVs** capable of low level navigation & evader detection

that rely on vision and actuation systems

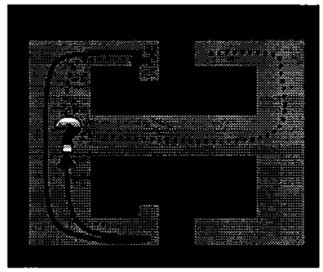
ARO DAAH04-96-1-0341 22

Strategies for pursuit-evasion games

LaValle, Latombe, Guibas, *et al.* considered a similar problem but assume the map of the region is **known**, the pursuers have **perfect sensors**, and **worst case** trajectories for the evaders:

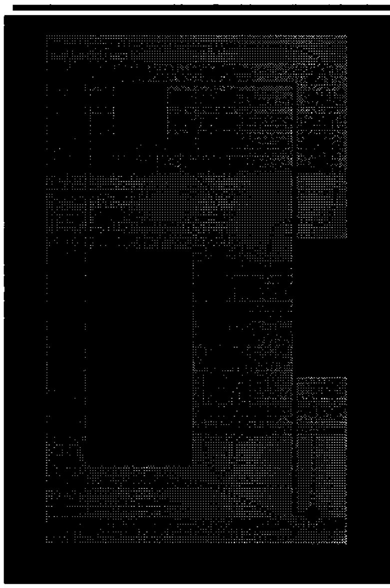
How many *UAVs are needed to win the game in finite time?*

1 agent is sufficient



2 agents are needed
(no matter what strategy a single pursuer chooses, there is a trajectory for the enemy that avoids detection)

Exploring a region to build a map



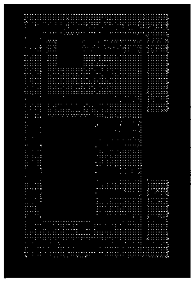
Deng, Papadimitriou, *et al.*, study the problem of building a map (seeing all points in the region) traversing the smallest possible distance.

●·······► standard "keep wall to the right" algorithm

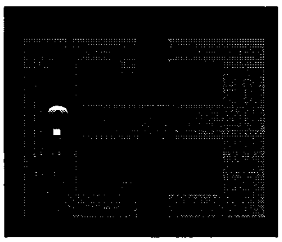
●——► algorithm that takes better advantage of the cameras capabilities

A two step solution...

exploration



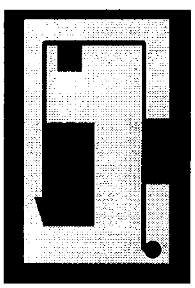
pursuit



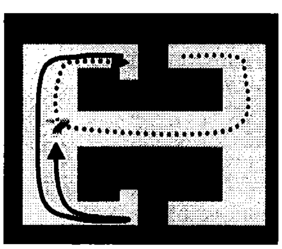
- exploration followed by pursuit is not **efficient**
- sensors are **imprecise**
- **worst case** assumptions the trajectories of the evaders leads to very conservative results

A different approach...

exploration



pursuit



Use a **probabilistic framework** to **combine** exploration and pursuit-evasion games.

Non determinism comes from:

- poorly mapped **terrain**
- noise and uncertainty in the **sensors**
- probabilistic models for the **motion** of the evader and the UAVs

Problem formulation

At each time t:

- the pursuers take ***measurements*** y_t (positions of enemies and obstacles seen)
- ***control actions*** u_t are sent to the pursuers (motion control)

A ***pursuit policy*** is a function g that maps past measurements to the next control action:

$$u_{t+1} = g(y_1, y_2, \dots y_t)$$

Goal: design pursuit policies that "maximize" probability of finding the evader

ARO DAAH04-96-1-0341 27

Optimality criteria

- **Probability of finding** the evader in the next T time steps (to be maximized)

$$J_T(t_0) = \text{Prob(find an evader from } t_0 \text{ to } t_0 + T)$$

- **Expected time** needed to find the evader (to be minimized)

$$J^* = E\{T^*\}$$

$T^* \equiv$ time at which the evader is found

ARO DAAH04-96-1-0341 28

Optimality criteria

- **Probability of finding** the evader in the next *T* time steps (to be maximized)

$$J_T(t_0) = 1 - \prod_{t=t_0}^{t_0+T} (1 - p_g(t) \qquad (\; J_1 = p_g(t_0))$$

- **Expected time** needed to find the evader (to be minimized)

$$J^* = \sum_{t=t_0}^{\infty} t p_g(t) \left(\prod_{\tau=t_0}^{t-1} (1 - p_g(t)) \right)$$

where $p_g(t) \equiv$ Prob(finding evader at time *t*, given that none was found up to *t*)

Sub-optimal policies

greedy policy ≡ pursuit policy that, at each instant, maximizes the probability of finding the evader at the next instant

computationally very attractive because it scales well with the size of the region, number of obstacles, etc.

Sub-optimal policies

greedy policy ≡ pursuit policy that, at each instant, maximizes the probability of finding the evader at the next instant

- **Probability of finding** the evader in finite time is one, i.e.,

  Prob($T^* < \infty$) = 1 $\quad$ T^* ≡ time at which the evader is found

- **Expected time** needed to find an evader is finite, e.g.,

$$J^* = E\{T^*\} \leq \frac{\text{area}}{\text{n. of pursuers}}$$

  when a randomly moving evader is pursued by UAVs that can check any single unit of area at each instant of time

Map building

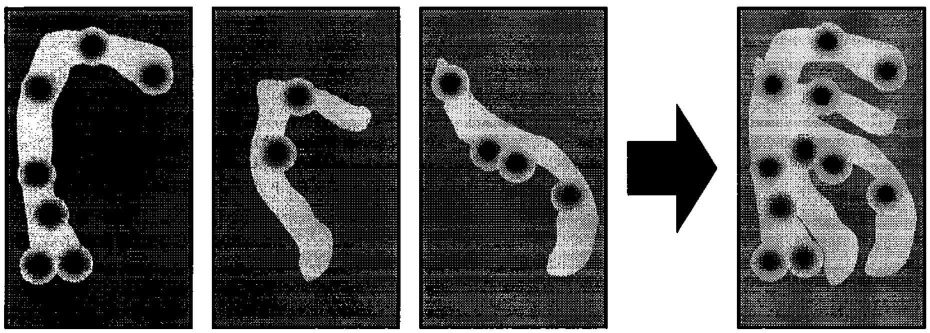
Problem:
For given paths of all the UAVs, how to fuse the data sensed by each of them (in real-time) to build a global map?

Probabilistic map building

Prob(map|new data & old data)
$= k$ Prob(new data|map) Prob(map|old data)

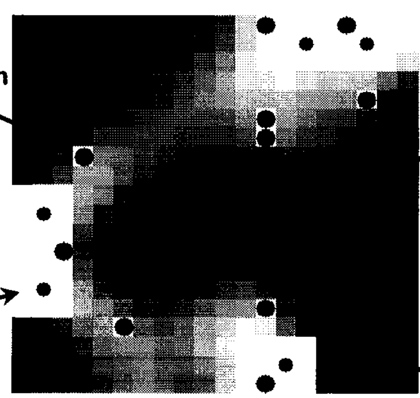
sensor model

Problem:
For given paths of all the UAVs, how to fuse the data sensed by each of them (in real-time) to build a global map?



ARO DAAH04-96-1-0341 33

Simulation (greedy policy)



ARO DAAH04-96-1-0341 34