UNIVERSITY of CALIFORNIA Riverside
BOURNS COLLEGE OF Engineering

Distributed Camera Networks

Integrated sensing and analysis for wide-area scene understanding

Amit K. Roy-Chowdhury
University of California, Riverside

Acknowledgments

- Graduate Students/Post-docs: Dr. B. Song, C. Ding, T. A. Kamal.
- Collaborators: Prof. J. A. Farrell
- Funding Agencies: NSF IHCS Program, ARO, ONR.

- An overview of our work can be found in the May 2011 issue of Signal Processing Magazine (this has reference to other papers with more details).

What are camera networks?

- Small and large networks of video cameras are being installed in many applications: security and surveillance, environmental monitoring, disaster response.

- Almost complete manual analysis
  - Tedious
  - Overwhelming

- We need automated processing to assist the user – very challenging problem.

Challenges in Camera Networks

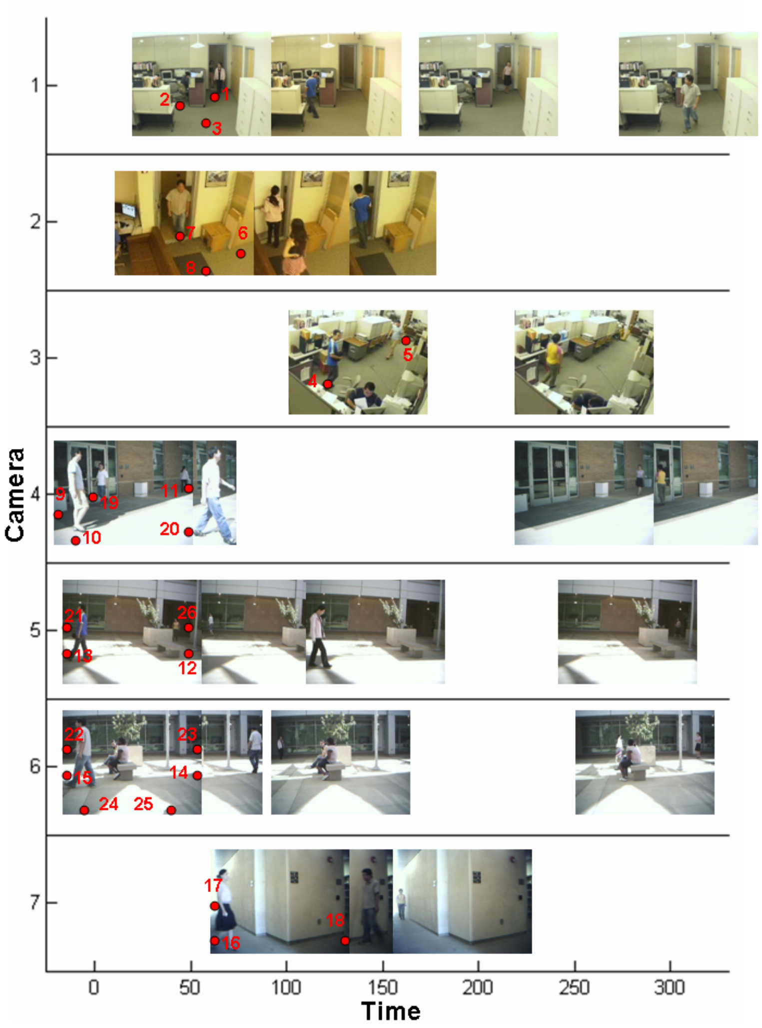
- Traditional computer vision challenges in tracking and recognition – robustness to pose, illumination, occlusion, clutter… *(in fact, these issues are more significant in large network).* **However, we will ignore them today!**

- Tracking over wide areas – hold targets as they appear and disappear

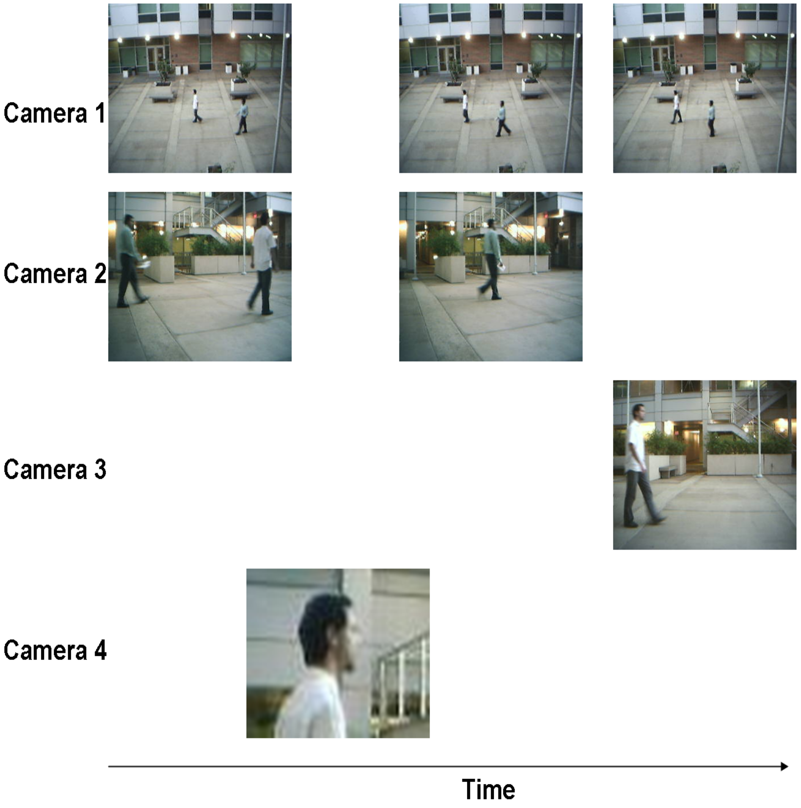
- **Centralized vs. Distributed Processing**

- **Static vs. Active camera networks – to control or not?**

Two Examples



Static Network



Active Network

Distributed vs. Centralized

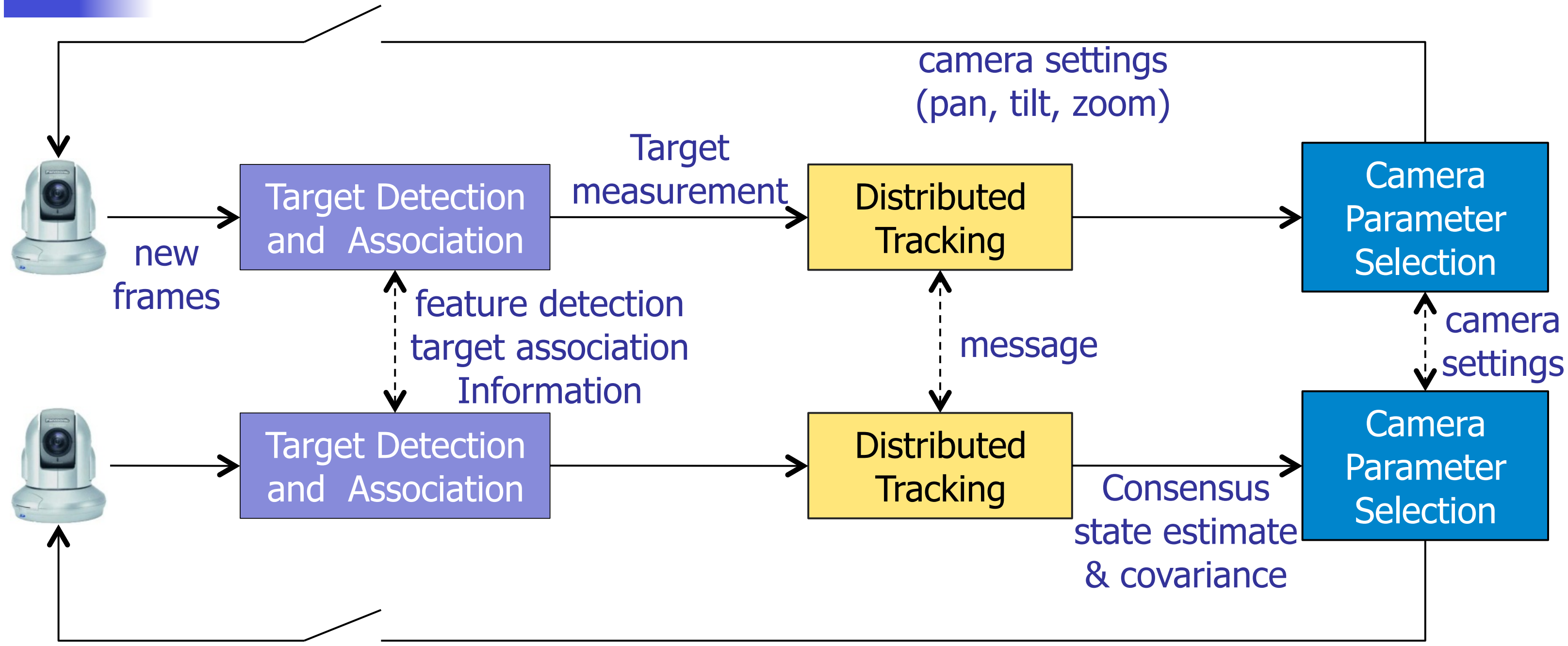
- In many applications, ***distributed*** video analysis is desirable
  - Bandwidth constraints
  - Security issues
  - Difficulties in analyzing a huge amount of data centrally

- The cameras, acting as autonomous agents
  - Analyze the raw data locally
  - Exchange only distilled information
  - Reach a shared, global analysis of the scene

Integrated Sensing and Analysis

- Most existing camera networks: fixed cameras covering a large area
  - Targets are often not covered at the desired resolutions/ viewpoints
  - Make the analysis of the video difficult
- Integrate the analysis and sensing tasks more closely
- Control the parameters of a pan-tilt-zoom (PTZ) camera network to fulfill the analysis requirements

An Integrated Sensing and Analysis Framework



camera settings (pan, tilt, zoom)

new frames

Target Detection and Association

Target measurement

Distributed Tracking

Camera Parameter Selection

feature detection target association Information

message

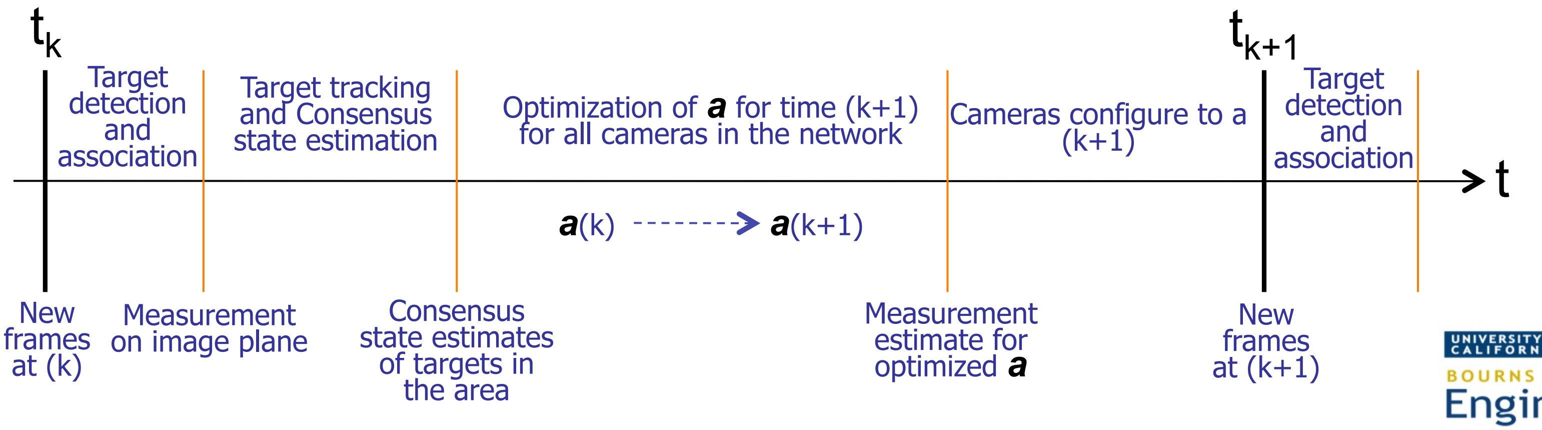
camera settings

Target Detection and Association

Distributed Tracking

Consensus state estimate & covariance

Camera Parameter Selection



t_k

Target detection and association

Target tracking and Consensus state estimation

Optimization of ***a*** for time (k+1) for all cameras in the network

Cameras configure to a (k+1)

t_{k+1}

Target detection and association

t

a(k) - - - - - - -> ***a***(k+1)

New frames at (k)

Measurement on image plane

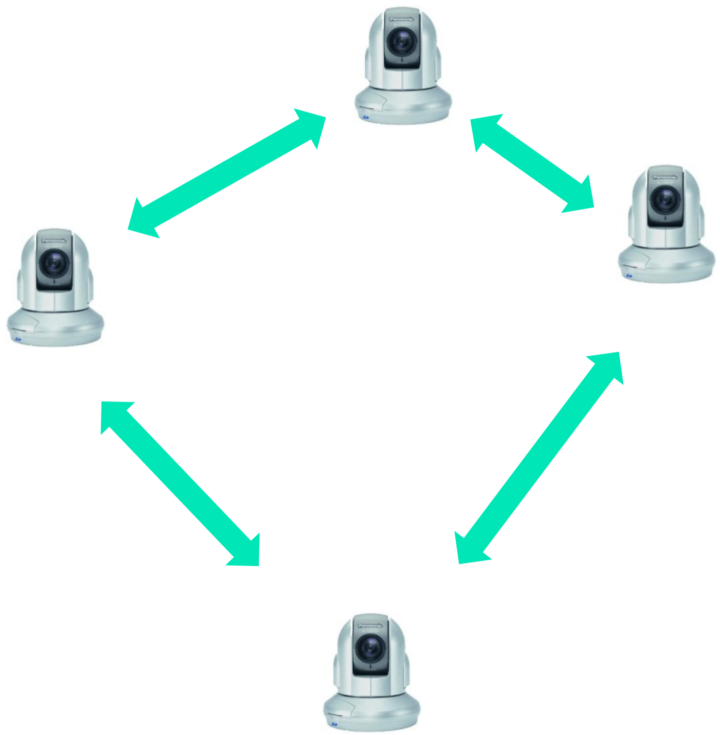
Consensus state estimates of targets in the area

Measurement estimate for optimized ***a***

New frames at (k+1)

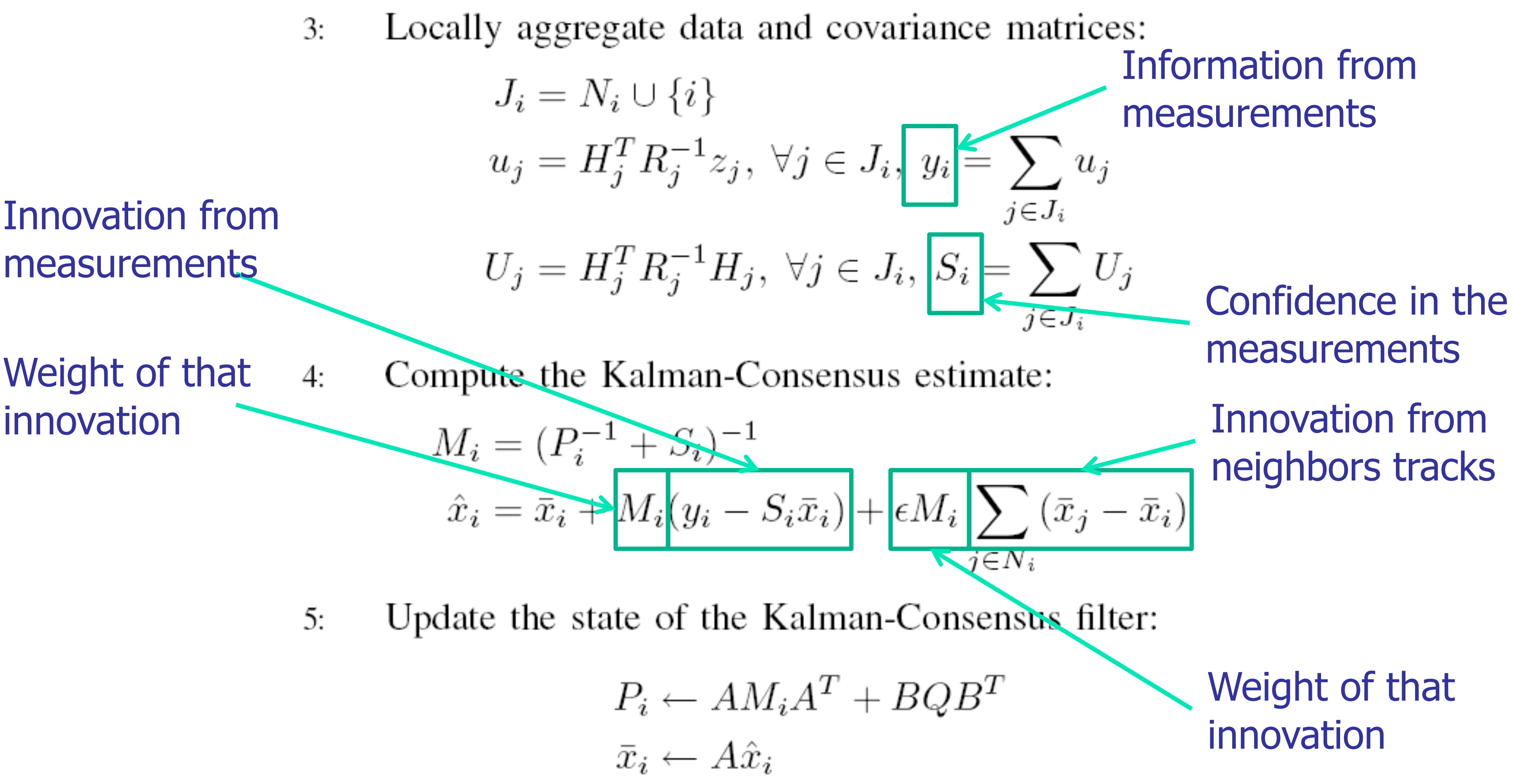
DISTRIBUTED TRACKING

Distributed Tracking through Consensus



1. Each camera estimates states of targets based on local measurement
2. Send message to neighboring cameras
3. Receive message from neighbors
4. Fuse information to compute consensus state estimate

Kalman-Consensus Filter



3: Locally aggregate data and covariance matrices:

$$J_i = N_i \cup \{i\}$$

$$u_j = H_j^T R_j^{-1} z_j, \ \forall j \in J_i, \ \boxed{y_i} = \sum_{j \in J_i} u_j$$

$$U_j = H_j^T R_j^{-1} H_j, \ \forall j \in J_i, \ \boxed{S_i} = \sum_{j \in J_i} U_j$$

Information from measurements

Confidence in the measurements

Innovation from measurements

Weight of that innovation

4: Compute the Kalman-Consensus estimate:

$$M_i = (P_i^{-1} + S_i)^{-1}$$

$$\hat{x}_i = \bar{x}_i + \boxed{\boxed{M_i}\boxed{(y_i - S_i \bar{x}_i)}} + \boxed{\boxed{\epsilon M_i}\boxed{\sum_{j \in N_i} (\bar{x}_j - \bar{x}_i)}}$$

Innovation from neighbors tracks

Weight of that innovation

5: Update the state of the Kalman-Consensus filter:

$$P_i \leftarrow A M_i A^T + B Q B^T$$

$$\bar{x}_i \leftarrow A \hat{x}_i$$

Ongoing Work

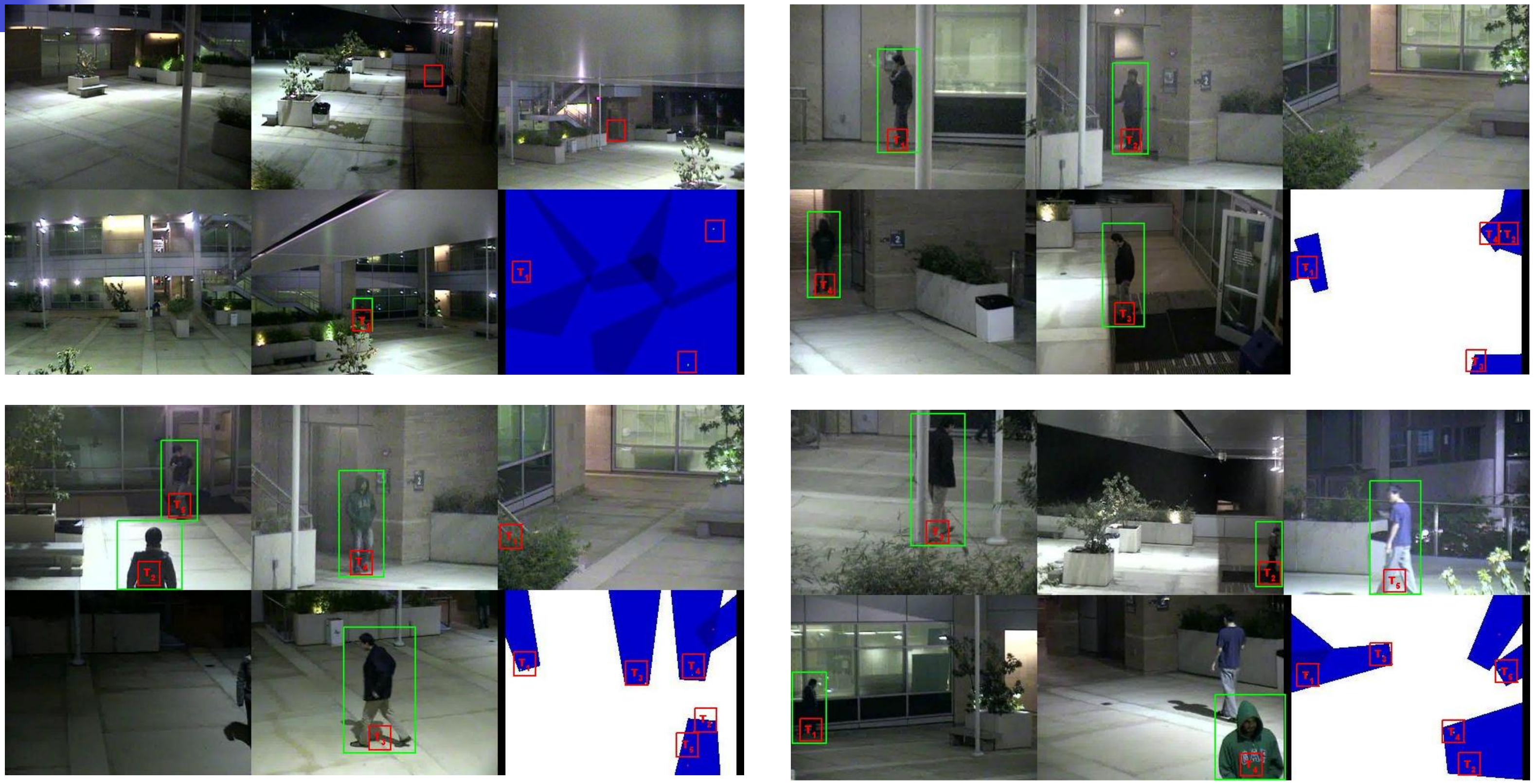
- Wide area camera networks have sparse connectivity, which causes a lag in the estimation.
- Can lead to completely incorrect tracks.
- Ongoing work: modifications to the KCF framework for video networks (can discuss offline).

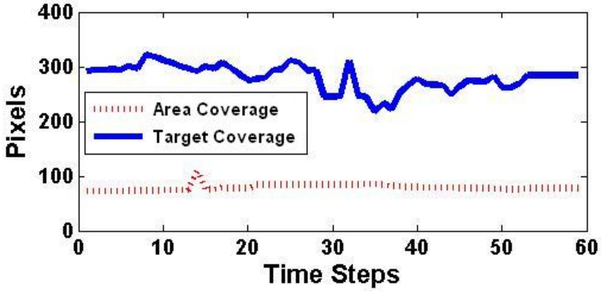
CAMERA NETWORK CONTROL

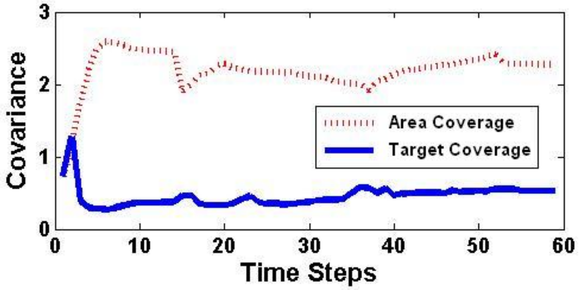
Game Theoretic Framework for Camera Network Reconfiguration

- Design each camera to be a rational decision maker
- Formulate the problem as a multiplayer game, where each camera is a player and interested in optimizing its own utility
- By designing the camera utility functions to be aligned with the global utility function, the game is a potential game with the global utility function being the potential function
- The agreeable settings of cameras (Nash equilibria), should lead to high, ideally maximal, global utility.

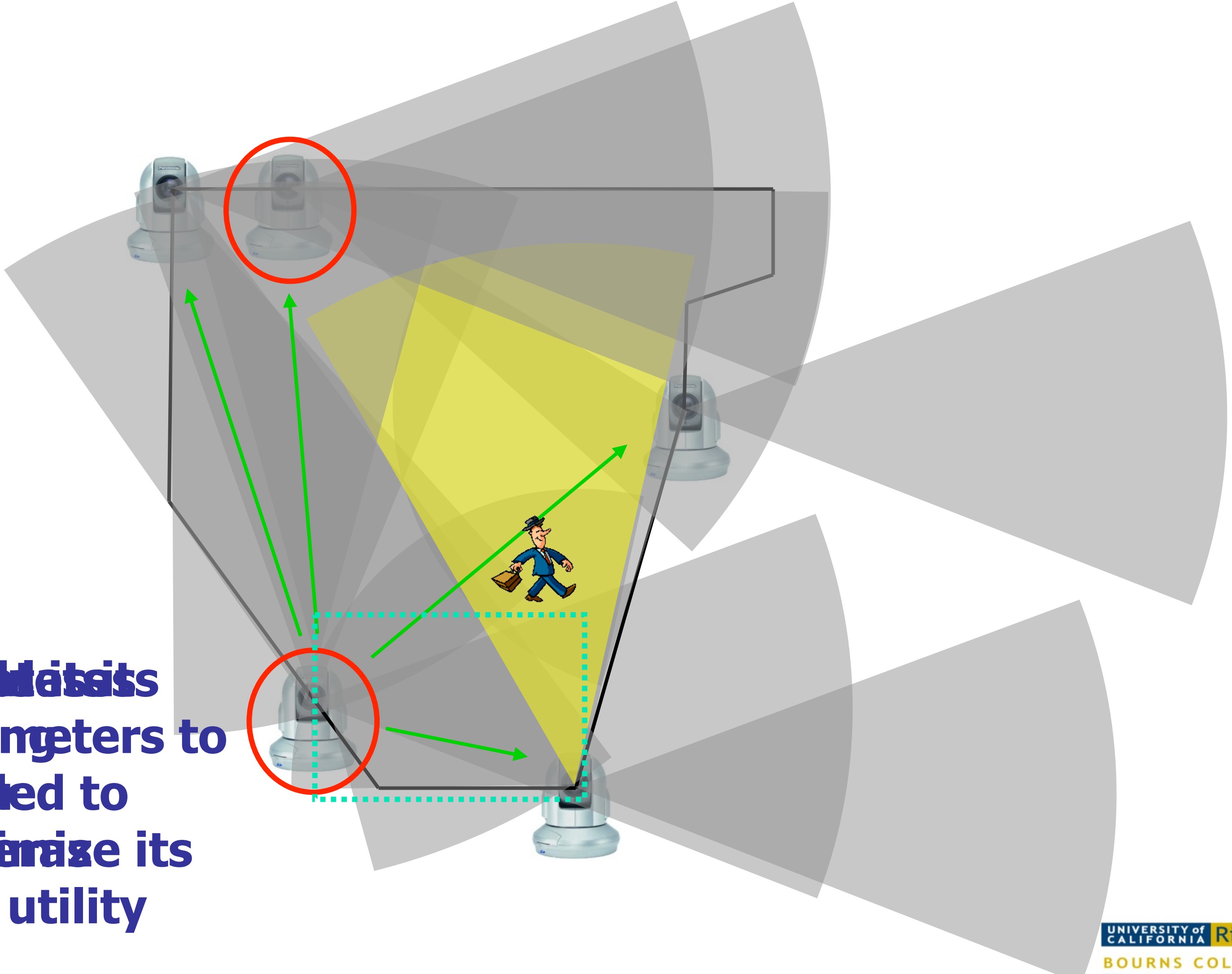
Example - Static vs. Active cameras







Proposed Algorithm



Sends its parameters to other cameras

Updates its setting needed to maximize its own utility

Video
Computing
Group

University of California Riverside
Bourns College of Engineering

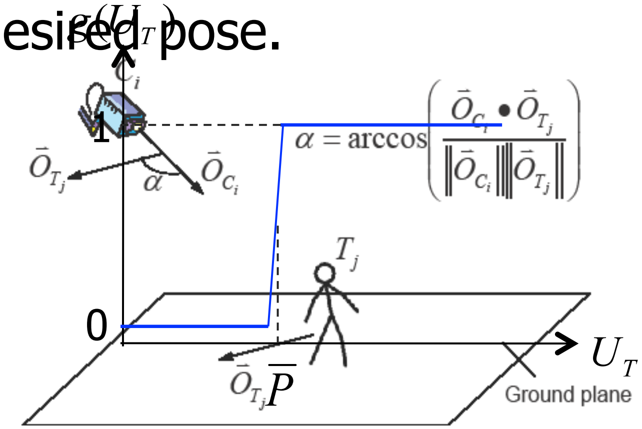
Global Utility

$U_G(a) =$ Tracking Utility + Imaging Utility

$U_G(a) =$ Tracking Utility Weight × Imaging Utility

- Find the set of parameters that result in the most improvement towards our tracking specification.
  - If the estimates of all targets are expected to exceed the tracking accuracy specified
  - Minimize the covariance ellipse of the target.

- Priority is given to obtain high resolution shots
  - Maximize the resolution of the worst tracked target.
  - Obtaining an image from the desired pose.



Camera Utility

Camera Utility should be aligned with global utility

$$U_{C_i}(a_i, a_{-i}) - U_{C_i}(b_i, a_{-i}) > 0 \Rightarrow U_g(a_i, a_{-i}) - U_g(b_i, a_{-i}) > 0$$

⟹ The game is a ordinary potential game

⟹ Guarantee the existence of Nash equilibria

$$\boxed{U_{C_i}(a_i, a_{-i}) = U_g(a_i, a_{-i}) - U_g(a_{-i})}$$

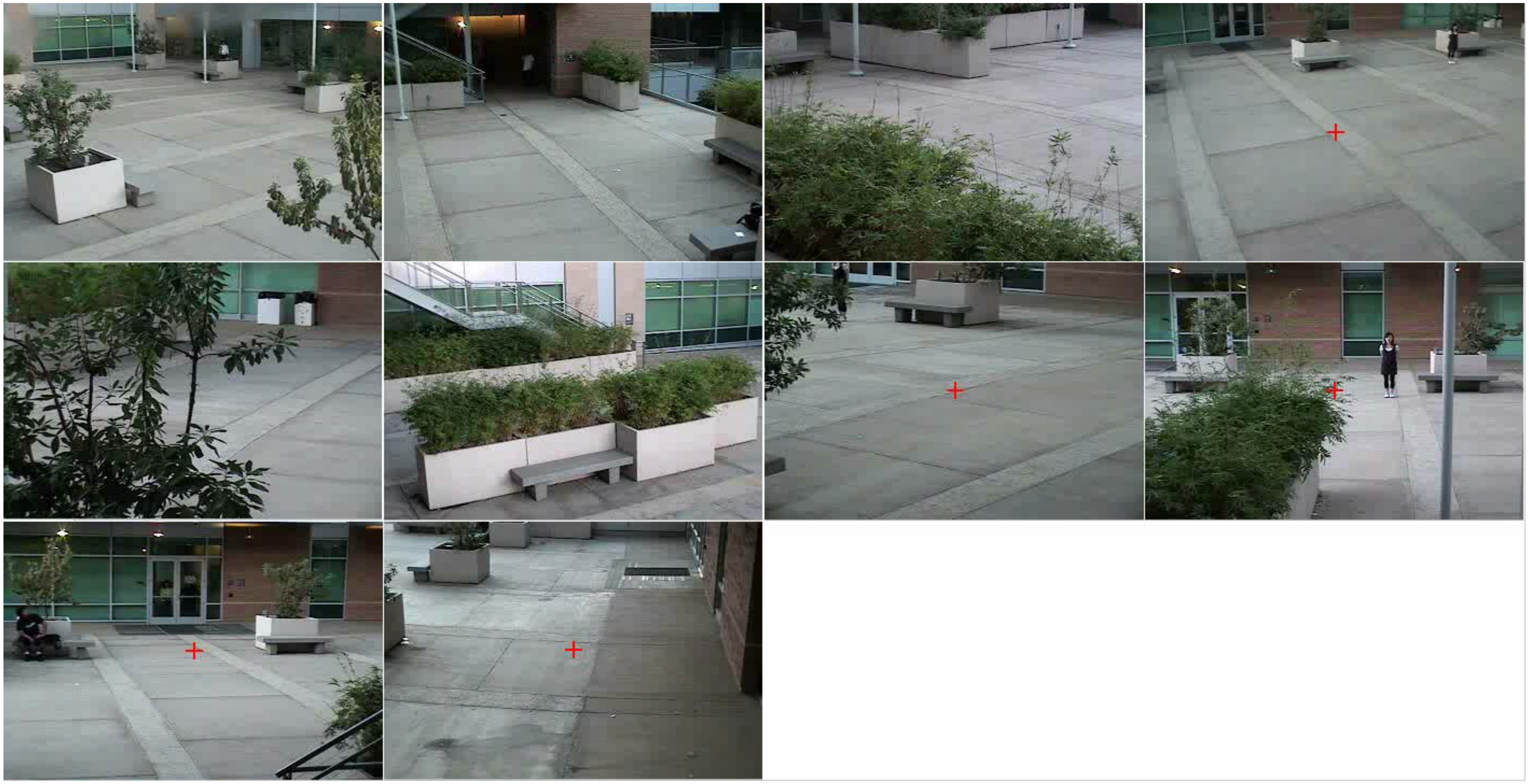
Optimization Goal

A choice of parameter settings $a^* = (a_1^*, \ldots a_{N_c}^*)$ such that no sensor could improve its utility further by deviating from a^*, i.e., by choosing a different set of parameters, the utility functions of all cameras cannot be improved further.

If a_{-i} denotes the collection of targets for all cameras except camera C_i, then a_i^* is a pure Nash equilibrium if

$$U_{C_i}(a_i^*, a_{-i}^*) = \max_{a_i \in A_i} U_{C_i}(a_i, a_{-i}^*), \forall C_i \in C$$

Experimental Results



The video can be downloaded from
http://www.ee.ucr.edu/~amitrc/CameraNetworks.php under Demos.

Future Work

- Design of optimization strategies that enable
    - Robust opportunistic sensing
    - Active control integrated with scene understanding criteria
    - Optimization strategies that allow for semantic descriptions (interface with AI)
- Mobile camera networks
    - Joint optimization of camera parameters and trajectory estimation of mobile platforms.