Collaborative Networks of Image Sensors

Abbas El Gamal, EE
Leonidas Guibas, CS
Balaji Prabhakar, EE&CS

Ali Ozer Ercan, EE
Jaewon Shin, EE
Danny Yang, CS
Wireless Sensor Networks

- Distributed systems consisting of small, untethered, low-power nodes capable of sensing, processing, and wireless communication
- Applications in scientific, industrial, commercial, and military settings
Wireless Sensor Networks, II

Advantages:
- sensors can be close to signal sources, yielding high SNR
- monitoring of phenomena widely distributed across space and time
- scalable, robust and self-repairing systems

Challenges:
- sensor tasking and control
- in-network, distributed processing
- management, service establishment, software layers
- power awareness
Networking Sensor Networks

- Network support for a small number of collaborative tasks.
- Data-centric, (as opposed to a node-centric) view of the world.
- Monitoring processes may migrate from node to node, as the phenomena of interest move or evolve.
- Communication flow and structure is dictated by the geography of signal landscapes and the overall network task.
Image Sensor Networks

CMOS technology enables the production of small, low-cost and low-power integrated image sensors.

Cameras (still or video) and other image sensors are becoming cheaper, smaller, and nearly ubiquitous.

However, truly distributed networked systems of image sensors are still not here.
Current Multi-Imager Networks

- Data is transported over a wired network to a central location
- Human operators look at the data

This approach cannot scale:
- vast amounts of data to move
- wiring is expensive
- automatic ways to filter the data are needed
Distributed Imager Challenges

- Imagers are **high data rate** sensors; therefore data must be compressed and summarized
  - compression must take into account shared data
  - goal of compression need not be reconstruction
- Vision algorithms can be **expensive** to run on low-power devices
- Visibility is **non-local** and **discontinuous** (occlusions, etc)
- Issues of privacy, etc.
The Vision: Collaborative, Task-Driven Image Sensing

- Large numbers of simple, inexpensive cameras collaborate over a wireless network to accomplish a task.
- Data is compressed locally and aggregated within the network.
- Cameras are only tasked as the situation demands.
- The system can be expanded incrementally to large numbers of nodes.

The goal is to estimate certain high-level, global attributes of the environment.
The Project

- Use a camera network to obtain information about space occupancy by people.
- Useful for aggregate tracking, counting, etc.
- Crowd density implies multiple occlusions – no one camera by itself can do this.
- No image reconstruction -- just high-level distributed spatial reasoning.
The Current Lab

- Web cameras:
  - 16 firewire webcams with 49 degree FOV
  - Placed around a 22 x 19 foot room

- Linux computers
  - Each PC is connected to 2 webcams
  - A separate process is running for each webcam to simulate an individual camera node
  - All processes can communicate with each other over the network
System Architecture

Autonomous Background Subtraction, Data Compression

Collaborative Visual Hull Estimation, Camera Tasking

Problem Solution
Local Processing

- Perform background subtraction
- Collapse to a single scan-line

640x480 RGB image $\rightarrow$ 640 bit scan-line (which can be further compressed)
Occupancy Representation: The 2-D Visual Hull
A Visual Hull Example

Top view of room with 5 people

Scanlines from 16 cameras
The Visual Hull Overestimates Occupancy

Visual hull regions surround each object.

Visual hull regions may also be empty; we call these the **phantom regions**
Pruning the Visual Hull

- Using more cameras reduces the overestimation – but it can never be fully eliminated.
- Motion can allow the pruning of phantom regions.
An Application: Counting People

- Given occupancy, bound the number of objects in each polygon of the visual hull.
- The bounds over time can be used to constrain the count, using a tree data structure.
A Counting Example
Tasking Subsets of Cameras

To scale the system to large numbers of cameras, we must understand how to select appropriate subsets of the sensors to activate.

Using a small subset can save communication bandwidth and energy. Often it can be done without significant estimation degradation.

![Diagram of optimal selection of 2 out of 3 cameras]
Number of Cameras vs. Visual Hull Quality

Let’s us decide how many cameras we need, to obtain a visual hull of a certain quality.
Camera Tasking Strategies

We need strategies for selecting a good subset of the cameras to task
- Uniform
- Greedy
- Incremental greedy
- (Brute force enumeration – optimal)

We also need to consider maintenance of this camera subset as objects move
- Do not want to recompute the entire subset at every step
Simulation Results, 5 Objects
Simulation Results, 5 Objects, II

Mobile Objects
Real Results, 3 People
Application: Tracking

6 cameras with no tasking
MSE: 0.370

Tasking 5 cameras with inc. greedy
MSE: 0.018
Future Work

- **Low-level processing**
  - Color histograms
  - Optical flow / Motion vectors

- **Theory**
  - Camera selection/placement

- **More interesting networking and distributed algorithms**
  - Autonomous camera clusters, detection of abnormal situations
  - Distributed attention, change of focus mechanisms, in-network reasoning
  - Push application-layer semantics into the networking layer
Large-Scale Deployment

- What can we do with many hundreds to thousands of inexpensive, perhaps low-resolution camera nodes?
- How would we increase the capability of the network by incorporating active sensing (pan/tilt, simple controllable lights, structured light mini-projectors, etc).
- What is the right balance between inference by the system and inference by a human observing data obtained by the system?