SCOPES: Smart Camera Object Position Estimation System

EWSN 2009

Ankur Kamthe, Lun Jiang, Matthew Dudys and Alberto Cerpa

School of Engineering
University of California Merced
Merced, CA, 95343, USA

Feb 13, 2009

Introduction

Motivation and Goals

- Inexpensive sensing with untethered wireless sensor network nodes has made data acquisition easier

Motivation and Goals

- Inexpensive sensing with untethered wireless sensor network nodes has made data acquisition easier
- Caveats: Network lifetime, Latency and Detection performance

Motivation and Goals

- Inexpensive sensing with untethered wireless sensor network nodes has made data acquisition easier
- Caveats: Network lifetime, Latency and Detection performance
  - Energy is the critical bottleneck (every bit transmitted moves the node one step closer to death)

Motivation and Goals

- Inexpensive sensing with untethered wireless sensor network nodes has made data acquisition easier
- Caveats: Network lifetime, Latency and Detection performance
  - Energy is the critical bottleneck (every bit transmitted moves the node one step closer to death)
  - Data processing lags due to slower software

Motivation and Goals

- Inexpensive sensing with untethered wireless sensor network nodes has made data acquisition easier
- Caveats: Network lifetime, Latency and Detection performance
    - Energy is the critical bottleneck (every bit transmitted moves the node one step closer to death)
    - Data processing lags due to slower software
    - Detection and tracking applications need data within a certain time deadline

Motivation and Goals

- Inexpensive sensing with untethered wireless sensor network nodes has made data acquisition easier
- Caveats: Network lifetime, Latency and Detection performance
  - Energy is the critical bottleneck (every bit transmitted moves the node one step closer to death)
  - Data processing lags due to slower software
  - Detection and tracking applications need data within a certain time deadline
- **Goals: design a distributed camera sensor network for occupancy estimation**

Current Technology for Occupancy Estimation

- Motion detectors and Temperature and CO_2 sensors

Current Technology for Occupancy Estimation

- Motion detectors and Temperature and CO_2 sensors
- Drawbacks:

Current Technology for Occupancy Estimation

- Motion detectors and Temperature and CO_2 sensors
- Drawbacks:
  - motion detectors cannot distinguish the number of occupants of a room

Current Technology for Occupancy Estimation

- Motion detectors and Temperature and CO_2 sensors
- Drawbacks:
  - motion detectors cannot distinguish the number of occupants of a room
  - temperature and CO_2 sensors are more suitable for general trends at long time scales (on the order of hours)

Current Technology for Occupancy Estimation

- Motion detectors and Temperature and CO2 sensors
- Drawbacks:
  - motion detectors cannot distinguish the number of occupants of a room
  - temperature and CO2 sensors are more suitable for general trends at long time scales (on the order of hours)
  - Need to fully instrument the building to use these sensors

Current Technology for Occupancy Estimation

- Motion detectors and Temperature and CO2 sensors
- Drawbacks:
  - motion detectors cannot distinguish the number of occupants of a room
  - temperature and CO2 sensors are more suitable for general trends at long time scales (on the order of hours)
  - Need to fully instrument the building to use these sensors
- A faster method (on the order of tens of seconds) is needed to respond quickly to rapid changes in occupancy

Problem Statement

- Given wireless camera sensors, deploy a system that can minimize cost and intrusion with no infrastructure needed (power and communication independent)

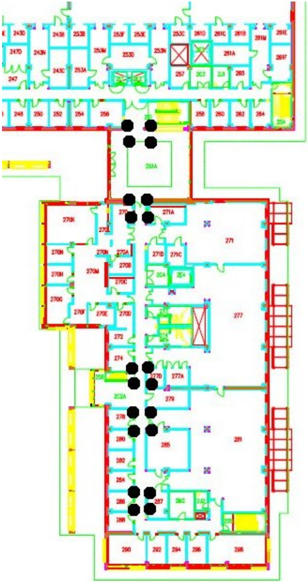
Problem Statement

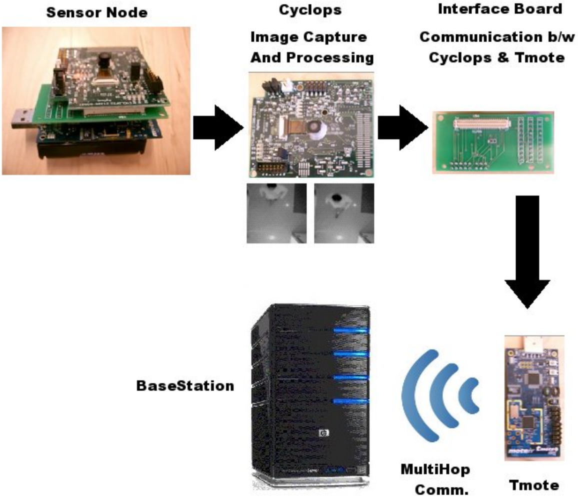
- Given wireless camera sensors, deploy a system that can minimize cost and intrusion with no infrastructure needed (power and communication independent)

Proposed Approach: SCOPES, a distributed camera sensor network for occupancy estimation

SCOPES: Smart Camera Object Position Estimation System

System Overview





Hardware and Software Description

Hardware

- Cyclops:
  - Agilent ADCM-1700 imager module combined with a 4MHz Atmega128L micro-controller with 512kB SRAM memory
  - External SRAM is divided into eight, 64KB memory banks (*nBanks*)
  - Cyclops captures 10 64x64 pixel grayscale images (*nFrames*) per bank (i.e., 80 total)
- Tmote: TI MSP430 micro-controller featuring 10kB of RAM and a Chipcon CC2420 radio

Software:

- Event-based TinyOS operating system

Node Sensing-Processing Cycle

No coordination:

- Overlapping sensing cycle
- Many time intervals with no sensing

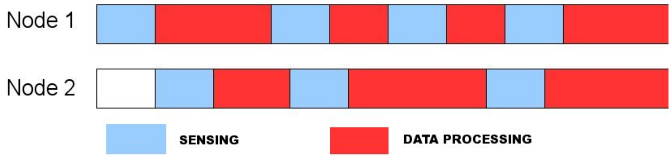


Figure: Illustration of nodes sensing without coordination

Image Processing Algorithms

Object Detection (on Cyclops)

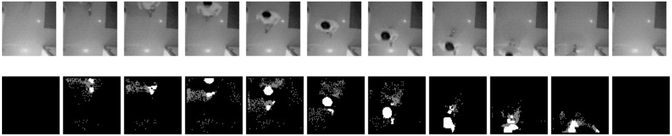


Figure: Top: Images Captured by the Cyclops, Bottom: Output of background subtraction

- background subtraction using fixed thresholds
- assign labels (OBJECT, SHADOW or BG) to each pixel
- Update corresponding background pixel using 3 different Exponentially Weighted Moving Average (EWMA)
- Formula: $bg(i) = \alpha \times bg(i) + (1 - \alpha) \times img(i)$
- $\alpha_{OBJECT} > \alpha_{SHADOW} > \alpha_{BG}$

Image Processing Algorithms

Object Recognition (on Cyclops)

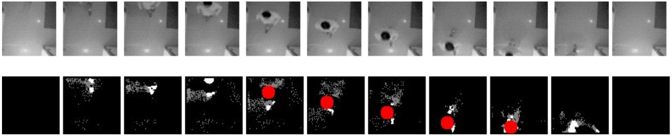


Figure: Top: Images Captured by the Cyclops, Bottom: Output of object recognition

- Raster scanning for connected components
- For each image, counting objects and determining center of mass for each object
- $O(\mathrm{n})$ complexity, where n is number of pixels

Execution Times for Image Processing

Cyclops Action	**nFrames**		
	1	5	10
Image Capture (IC)	0.68	1.43	2.51
IC w/ OD	0.7	1.54	2.74
IC w/ (OD + OR)	1.3	4.71	9.50

Table: Time (in second) for executing various image processing operations *nFrames* images at a time on the Cyclops (Note: averaged over 100 cycles)

Image Processing Algorithms

Direction Inference (on Cyclops):
Objects in successive frames are matched according to their size (in pixels)

Density Estimation (at Base Station):

- From the object data, we infer the direction of motion from the initial position and the displacement vector
- Since, we have prior information about the deployment of the nodes, counting the transitions of objects enables the base station to compute the distribution of people in different sections of the building over time

Performance Evaluation of SCOPES

Objective Functions

Global/Local Density Estimate: How accurately can we estimate the occupancy of each section and of the total area covered?

Memory Usage: How much memory is required to achieve acceptable performance?

Detection Latency: How long does the system take to report data?

Detection Probability: How good is the estimate of the movement of people across different sections in the building?

Experimental Deployment

2 hour experiments for evaluating objective functions. Typically, around 50-80 transitions per section

Ground truth: networked web cameras to record the movement of people

Ground truth data is processed using Haar Cascade classifier in the OpenCV library

Ground truth is manually corrected for errors

Figure: Input Images to OpenCV

Experimental Results I

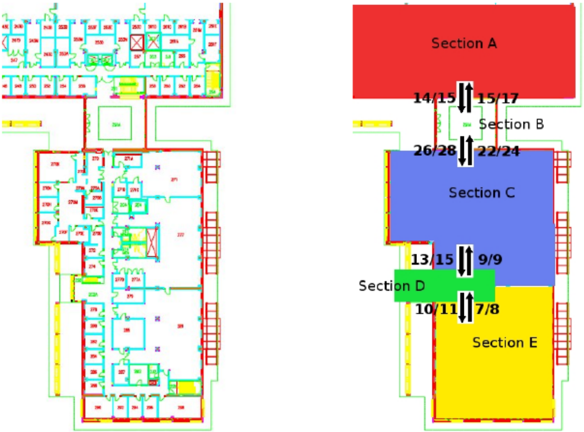


Figure: Occupancy Estimation Result

Experimental Results II

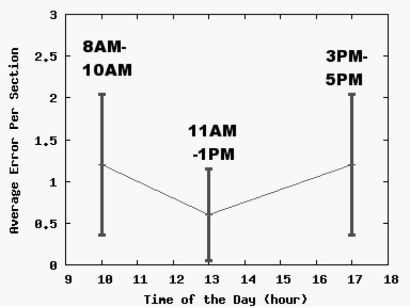


Figure: Density Estimation Error vs Time of the Day

Note:

1. Approx. 50-80 transitions per section per 2 hour interval
2. Density is inferred from transitions counts.

Experimental Results III

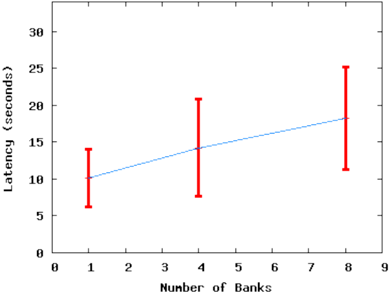
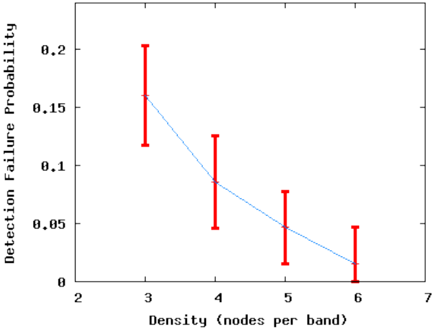
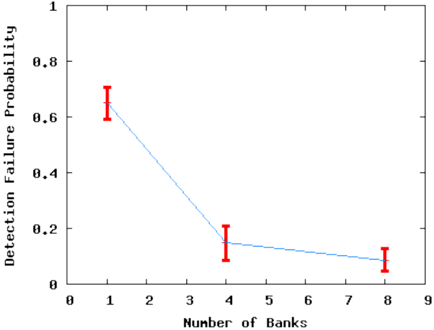


Figure: Detection Latency vs Memory Usage

Experimental Results IV



Detection Failure Probability vs Density of Nodes



Detection Failure Probability vs Memory Usage

Improving SCOPES using Node Coordination

Improving the Node Sensing-Processing Cycle

Node Clustering Algorithm

Cluster nodes based on proximity computed using radio signal strength

Distributed Coordination and Scheduling

Node Coordination scheme minimizes time with no sensing

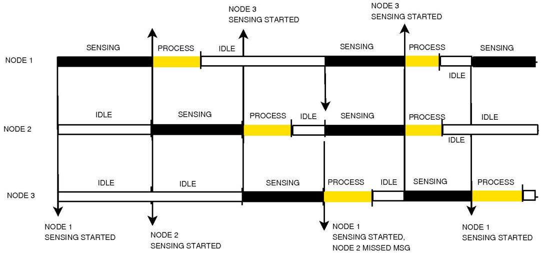


Figure: Illustration of the Group Coordination Algorithm

Comparing performance with and without coordination

Operation Mode	DFP/False Negative %	False Positive %	Number of Objects
Without Coordination	20.0%	21%	85
With Coordination	15.3%	18.5%	103

Table: Comparison of detection performance with and without node coordination

Detection Accuracy

Mis-Detections	Reason	Occurrences	Percentage
False Negatives	Object detected at memory bank border	4 out of 103	3.8%
	Object detected at limit of Sensing Range	8 out of 103	7.7%
	Software Inadequacy	4 out of 103	3.8%
False Positives	Objects in background	15 out of 193	7.7%
	Over-counting due to split-objects	21 out of 193	10.8%

Table: Counting the number of occurrences of false positives and false negatives along with their causes

NOTE:

1. Single, 2 hour expt with 3 nodes working together with coordination
2. Under false negatives, 103 was the total number of people passing beneath the SCOPES nodes
3. Under false positives, 193 is the total number of messages received at the base station

Comparison with other camera sensor networks

Aspects	SensEye	Motion Histogram	SCOPES
Hardware	Cyclops/MICAz/ Webcams/Stargate	Imote2/ EnalabCam	Cyclops/ Tmote
Comp. Speed	6Mhz-400Mhz	104Mhz	4Mhz
Performance	52%-84%	79.8%-89.5%	80%-95%
Object Speed	Moving objects: 100%(0.2m/s)-38%0.6m/s	approx. 1.2m/s	approx. 1.2m/s
Proc. Speed	N/A	8fps	1fps
Expt. Setting	circular objects were projected onto a wall	five people moving in a lab	real-world in ofice building

Table: Comparison with related camera sensor networks

Concluding Remarks

Simplicity aids design.

We showed that deployment of multiple nodes working in coordination with each other eliminates some of the problems associated with network lifetime, data processing latency and quality of detection performance.

SCOPES provides on par or better detection performance than other approaches that have computationally intensive algorithms and more capable hardware, with a slightly higher deployment cost

Future Work

- Upgrade to the faster Imote2 platform with a new camera board IMB400CA
- New image transformation, compression and processing techniques using PCA and SVM instead of summarized data
- People Mobility Models for indoor environments