MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Self-configuring, Lightweight Sensor Networks for Ubiquitous Computing

Christopher R. Wren and Srinivasa G. Rao

TR2003-24 October 2003

Abstract

We show that it is possible to extract geometric descriptions of the spaces observed by sensor networks, even if the network consists of sensors that are of very limited ability: such as motion detectors. By using statistical techniques and relying only on the unconstrained patterns generated by the occupants of the building we show how to recover information about sensor geometry. This is important to the ubiquitous computing community since ubiquitous sensors and the context that the provide will only become a reality if the sensors are cheap, low-power, and self-configuring.

Abridged version in Adjuct Proc. of the 5^{th} *International Conf. on Ubiquitous Computing*

Patent filed October 10, 2003.
Abridged version in UbiComp2003 Adjunct Proceedings October 13, 2003.

1 Introduction

The occupants of a building generate patterns as they move from place to place, stand at a corner talking, or loiter by the coffee machine. A cheap network of sensors can sense these patterns and provide useful information to all of the context sensitive systems in a building, but what makes such a network cheap? As the sensing and computational elements become cheaper to manufacture, the cost of such a network is quickly becoming dominated by installation, configuration and maintenance costs.

This paper explores some of the possibilities that exist for such networks to auto-calibrate, given only the unconstrained movements of those being observed. Furthermore, we strive to adopt an approach that will limit computational overhead. That means that the algorithms should not require recognition, tracking, or any but the absolute simplest of perceptual mechanisms. In fact, we will assume for the rest of this paper that our sensors are simple motion detectors. We also assume that the system will consist solely of sensors embedded in the environment, and not any component that navigates or is carried through the environment.

2 Related Work

Many ubiquitous context projects start from the assumption that the human inhabiting the space will be an active participant in calibrating the system[5], or that the system will accomplish calibration by utilizing an active element that can explore the environment[3]. For many applications, the level of detail desired about the building geometry does not warrant this level of labor cost or system complexity.

Lee, Romano, and Stein present a method for calibrating a overlapping security cameras using observed motion in the scene[4]. However their approach requires high-resolution sensors and far-field viewing, so that the moving objects will be small relative to the size of the desired calibration error. In our situation, we wish to employ only very0low-resolution sensors and in the office environment, the moving elements of the scene are large relative to the sensor field of view. Their approach also requires a brute-force search of all possible feature correspondence hypotheses to find correct matches, while our algorithm does avoids the requirement for knowledge of absolute correspondence by relying on statistical inference.

Caspi and Irani present a compelling algorithm to calibrate non-overlapping video streams[1]. the spirit of this work is very similar in that it attempts to recover geometric information about possibly non-overlapping sensors by exploiting temporal information. However, the Caspi approach requires significant common motion between the sensors that is usually only observed by pairs of cameras attached to rigid frames. In the office setting the sensors are assumed to be attached to the stable infrastructure, and will not observe these kinds of motions.

3 Our Sensor Network

We have covered $175m^2$ of office space with 17 ceiling-mounted sensors and collected motion event data. The sensors report motion events in their active area at 7.5Hz. They adapt to novel, but perfectly stationary objects, and other changes in the environment, on a 20 second time-scale.

The area covered consists of the high-traffic core of our building: the elevator lobby, reception lobby, restroom entrances, and connecting hallways.

In fact, for this experimental setup, the sensors are cheap, IEEE-1394, board cameras. They are mounted in the ceiling, pointed straight down at the floor with 75 degree angle lenses. The imagery from the cameras is processed by an adaptive background subtraction algorithm[6] built on top of the Open Computer Vision Library[2]. Obviously this is not the cheapest way to implement motion detectors, but it does provide the maximum flexibility for experimental design.

Motion events are reported as a threshold on the ratio of the foreground area to background area. The cameras capture 160x120 pixel images at 7.5Hz. The cameras view an area of approximately $12m^2$, so each pixel observes roughly $6cm^2$, or 1 square inch, on the floor. On the most heavily loaded machine, with 7 cameras, the perceptual process consumes 25% of a 1GHz Pentium III Processor.

4 The Experimental Setup

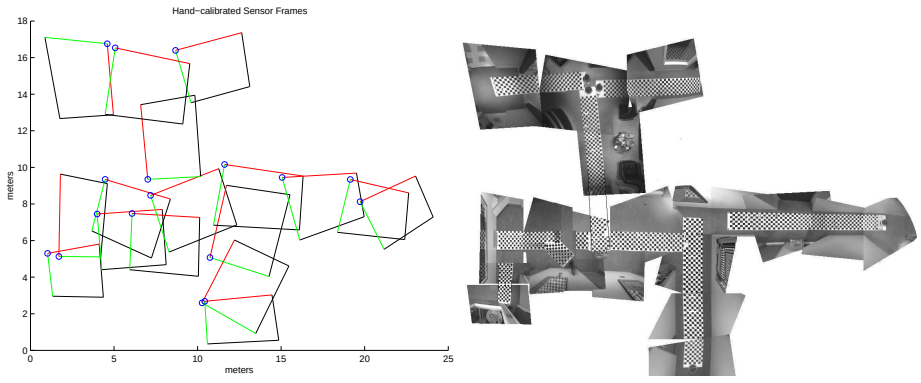

Figure 1: The Hand-calibrated Geometry: **Left:** The relative layout of the camera frames in the global coordinate system of the floor. **Right:** The calibration widgets.

Since the sensors are cameras, it was possible to use well-known techniques to recover the geometry of the cameras relative to the space observed. This provides us with ground-truth about the positions and viewing areas of

the sensors that we can use to validate our experimental results. Figure 1-left illustrated the measured geometry of the sensors.

Huge calibration grids printed on a poster printer were used in the estimation process. These calibration grids can be seen in Figure 1-right. Despite their large size, it was still necessary to bridge some camera frames through multiple intermediate frames to reach the global coordinate frame. This allows numerical error to accumulate and that leads to errors in the geometry of the ground-truth. However, the ground-truth does a good job of capturing the relative structure of the sensor network, and this is what we will be trying to recover .

5 Method

Since we treat the cameras simply as motion detectors, the underlying representation of the data will be the *event list*: $E_{j,t}$. An entry in the event list is active, $E_{j,t} = 1$, if there was a motion event at time t in sensor j. These events indicate merely the presence of some kind of motion anywhere in the field of view, but no indication of the number of people, the direction of motion, or any other such secondary information.

Our low-cost perceptual engine will be co-occurrence statistics: $C_{i,j,\delta}$. The co-occurrence is the count of events that co-occur at a given temporal offset:

$$C_{i,j,\delta} = \sum_{t=0}^{\inf} E_{i,t} E_{j,t+\delta}$$

where $\delta \geq 0$, and $E_{i,t}$ is a boolean value. For a given temporal offset, it is useful to manipulate the $i \times j$ co-occurrences between all sensors, for a given time offset, as a matrix. For a given pair of sensors, it is also useful to consider the family of co-occurrences parameterized by the temporal offset. Taken together, the $C_{i,j,\delta}$, for all possible δ are equivalent to the cross-correlation of the event lists for sensors i and j. However, the entire cross-correlation is not useful, and is very memory-intensive to compute, so we will only ever consider relatively small values δ: in particular, values that represent timescales that are relevant to human behavior.

We will be comparing this structure to a similar data derived from the ground-truth calibration data. Figure 2-left shows a binary map showing which cameras the ground truth indicates as overlapping. Figure 2-right shows the minimum distance between view areas for each camera. Overlapping cameras have a distance of zero. For cameras that do not overlap, the value is the minimum distance between the camera view polygons.

6 Discussion

We can demonstrate two things from this data: co-occurrence matrices that reveal the structure of the sensor overlap and structure in peak offsets in the

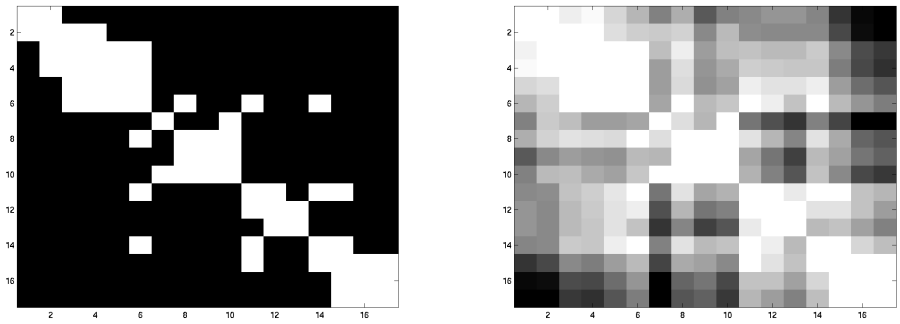

Figure 2: The Hand Calibration Data. **Left:** The overlap map for the sensors. The white blocks indicate two cameras that overlap. **Right:** The relative distance between the cameras. Overlapping cameras are white, darker blocks indicate more separation.

co-occurrence matrices that reflect the relative distances between sensors.

The $C_{i,j,0}$ co-occurrence matrix shows us the sensors that exhibit synchronized events. Since sensors always instantaneously co-occur with themselves, we see the highest values on the diagonal. However, off-diagonal elements with high values indicate sensors that overlap: they are often seeing the same event. Given that there are an unrestricted number of people moving around the space, we expect noise from coincidental events, but Figure 3-right shows that this noise is low compared to the signal. For this sensor network, we get 97% of the 136 non-trivial overlap decisions correct. Furthermore, all the false-negatives (3 of the 4 total errors) are actually mistakes in the ground-truth: two situations where un-modeled walls block views from sensors that would otherwise overlap, and one case where the geometry predicts a tenuous overlap that is obscured by un-modeled radial distortion in the lens of the sensor. Leaving out these errors gives us a 99% accuracy.

It is possible to see in Figure 4 that $C_{i,j,0}$ is not a good estimate of the inter-sensor distance in general. This is because it has no data about sensors that do not overlap. Unfortunately simply increasing δ does not help. Examples of these matrices can be found in Figure 5. We'll discuss what these structures represent below.

Figure 6 shows us a possible direction to recover inter-sensor distance for non-overlapping sensors. The plots depict, from top to bottom the co-occurrence between one particular camera and a set of cameras in order of increasing distance from the first camera. We can see that there is a strong peak that occurs at the minimum time it takes a significant fraction of the population to transit from the first camera to the second.

The windowed cross-correlation represented by $C_{i,j,\delta}$ over all δ and a given pair of sensors provides a way to estimate the average trip time between the

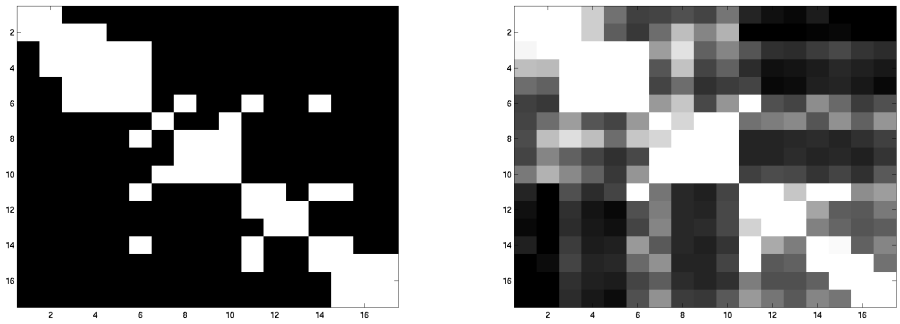

Figure 3: The ground truth overlap (left) compared to the statistical transition probability matrix (right).

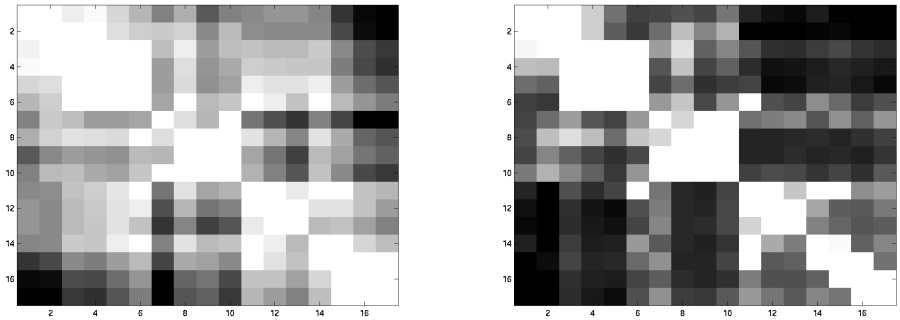

Figure 4: The ground truth distance map (left) compared to the statistical transition probability matrix (right).

two sensors. The time offset corresponding to the first major peak for a set of cameras provides an estimate of the average trip time between the sensors. We can use these pairwise constraints to form an estimate of the relative geometry of the whole network.

If people only ever transit uninterrupted between the sensors, then we could simply take the maximum of the cross-correlation, as in audio localization. However, we can't discount the possibility that the majority of individuals might stop to perform a task on their way through the space. This could cause a dominant peak to the right of the peak that truly corresponds to the uninterrupted transit time. The only way for a significant distraction to occur left of the true peak would be for a majority of transits to occur *faster* than the average transit time. It's hard to imagine how this would occur unless individuals were routinely running between sensors at a rate significantly faster than the mean walking speed observed elsewhere in the system.

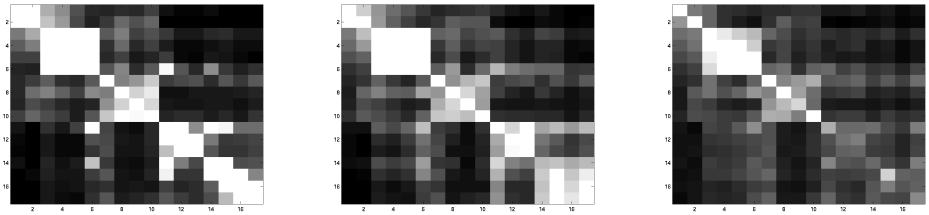

Figure 5: The co-occurrence maps for different intervals of time: **left:** 1s–3.5s, **center:** 3.5s–10s, **right:** 10s–27s

7 Results

These results are shown in Figure 7. On the left is the recovered geometry from the ground-truth distance constraints. On the right is the recovered geometry from the estimated inter-node transit times.

For our dataset, discounting the global scale ambiguity, we obtain an average error of $2.2m$ with only 4 hours of data. If we only consider a sub-set of the sensors that do not overlap, we obtain a slightly higher average error of $2.4m$. Our sensors monitor $3.7m \times 4.9m$ rectangles, so both of the figures represent sub-pixel accuracies.

8 Conclusion

We have shown that it is possible to extract descriptions of the spatial arrangement of a sensor network with very little computation, very poor sensors, and limited constraints on the behavior of the people inhabiting the space. This is important to the ubiquitous computing community since ubiquitous sensors will only become a reality if they are cheap, low-power, and self-configuring.

References

[1] Yaron Caspi and Michal Irani. Alignment of non-overlapping sequences. In *ICCV*. IEEE, 2001.

[2] Intel Corporation. *Open Source Computer Vision Library Reference Manual*, 2001.

[3] Anthony LaMarca1, Waylon Brunette, David Koizumi1, Matthew Lease1, Stefan B. Sigurdsson1, Kevin Sikorski, Dieter Fox, and Gaetano Borriello. Plantcare: An investigation in practical ubiquitous systems. In *Fourth International Conference on Ubiquitous Computing*. Springer, 2002.

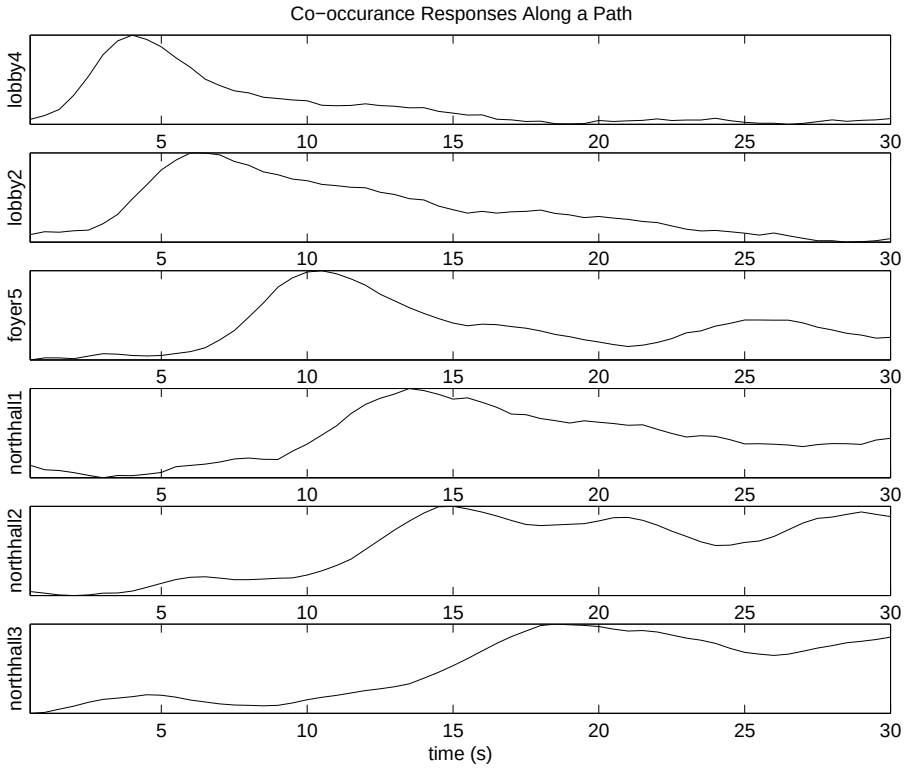

Figure 6: Plots of co-occurrence rate between a fixed camera and the cameras along a contiguous, hand-selected path through the space.

[4] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8), 2000.

[5] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking*, August 2000.

[6] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *ICCV*, pages 255–261. IEEE, 1999.

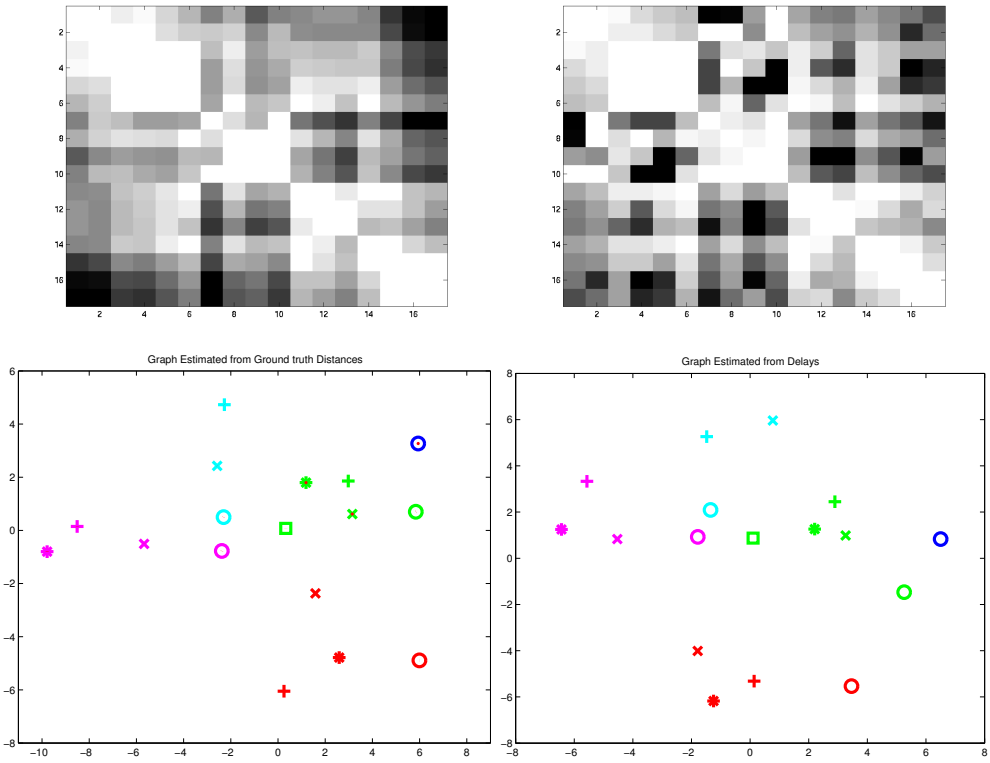

Figure 7: The matrices of relative distances (top) and corresponding two dimensional layout inferred from those matrices (bottom). The ground truth distance map (left) compared to the peak-delay map (right). Distances in meters.