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

Toward Scalable Activity Recognition for Sensor Networks

Christopher R. Wren and Emmanuel Munguia Tapia*

TR2006-011 March 2006

Abstract

Sensor networks hold the promise of truly intelligent buildings: buildings that adapt to the behavior of their occupants to improve productivity, efficiency, safety, and security. To be practical, such a network must be economical to manufacture, install and maintain. Similarly, the methodology must be efficient and must scale well to very large spaces. Finally, be be widely acceptable, it must be inherently privacy-sensitive. We propose to address these requirements by employing networks of passive infrared (PIR) motion detectors. PIR sensors are inexpensive, reliable, and require very little bandwidth. They also protect privacy since they are neither capable of directly identifying individuals nor of capturing identifiable imagery or audio. However, with an appropriate analysis methodology, we show that they are capable of providing useful contextual information. The methodology we propose supports scalability by adopting a hierarchical framework that splits computation into localized, distributed tasks. To support our methodology we provide theoretical justification for the method that grounds it in the action recognition literature. We also present quantitative results on a dataset that we have recorded from a 400 square meter wing of our laboratory. Specifically, we report quantitative results that show better than 90% recognition performance for low-level activities such as walking, loitering, and turning. We also present experimental results for mid-level activities such as visiting and meeting.

To Appear in LoCA 2006

*Mr. Munguia is currently with the Massachusetts Institute of Technology; 1 Cambridge Center, 4FL; Cambridge, MA, 02142 USA; emunguiamit.edu

December 2005: To Appear in the 2nd International Workshop on Location- and Context-Awareness

January 2006: filed patent (MERL-1753) with USPTO.

February 2006: accepted to the 2nd International Workshop on Location- and Context-Awareness (LoCA), Dublin, May 10, 2006.

1 Introduction

Buildings should be experts in the day to day activities of their inhabitants. This would make buildings safer by providing census data during emergencies. It would enhance security allowing the building to recognize daily patterns and flag unusual activity. It could improve efficiency by predicting demand for heating, lighting, and elevators. It could enrich human effort by providing presence and availability information, or supporting social networking applications. There is a tremendous potential benefit when buildings become experts in themselves, experts in the activities that occur within them.

Sensor networks have been investigated for such tasks as environmental monitoring, and resource tracking[15, 5]. We present a sensor network and inference methodology that enables buildings to sense and interpret the context of the human occupants in a potentially economical, scalable, efficient, and privacy-sensitive manner. Our sensor network is composed of passive infrared motion detectors. These sensors only detect presence and movement of heat sources, so they preserve much of the privacy of the occupants.

The system estimates the physical topology of the network and uses that information to form *context neighborhoods* around each node. Loosely, a context neighborhood is the collection of nodes that have a semantically-grounded link to the central node. That is, nodes form a neighborhood if they are physically near to each other, and the constraints of the space allow people to move freely between their sensor range, so that their sensor readings are related to each other by the dynamics of the space. These neighborhoods are the basis for portable behavior recognition and system scalability and we will define the several specific kinds of neighborhood in this paper.

We choose the smallest neighborhoods to be large enough to accurately detect the atomic components of human behavior in a building. We do not require individuals to be tracked before behavior is recognized, this allows the system to be built with cheaper sensors, and eliminates much of the computational overhead associated with high-fidelity tracking. By accurately detecting low-level movement behaviors locally, we also greatly reduce the amount of data that must be communicated outside the neighborhoods. These features support scalability.

The neighborhoods are also defined small enough to be invariant to the larger context of a building. This means that the detectors should be portable from location to location. This fact reduces the overall cost by eliminating much of the on-site calibration and engineering cost. There is no need to accurately position the sensors, they only need to tile the space in a rough grid.

Scalability and re-usability benefits can be found by building larger neighborhoods as collections of smaller neighborhoods. In this paper we will present this hierarchical neighborhood architecture. The architecture makes sense both from a communication efficiency point of view[19] and from a behavioral context point of view. We will present our taxonomy of building occupant behaviors and discuss how those behaviors map onto our sensor hierarchy.

In Section 5, we support our claims with experimental results form our test

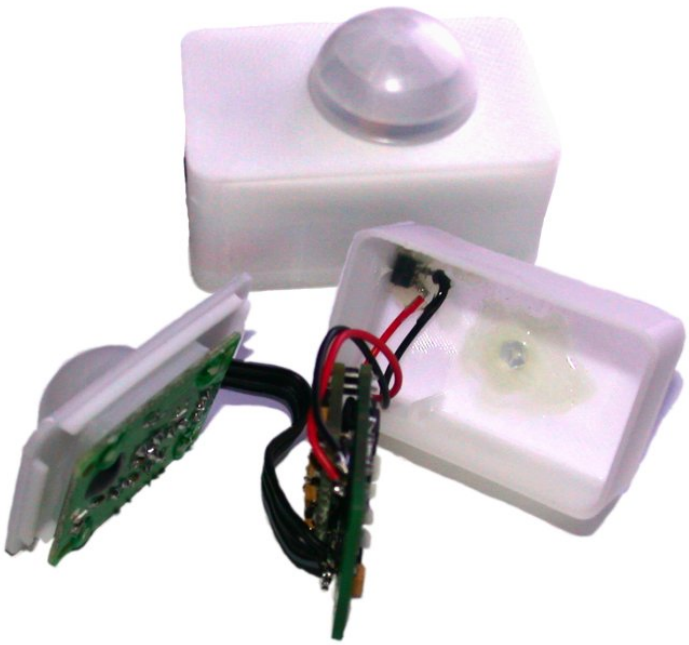

Figure 1: The hardware implementation of the motion detector node.

facility. The current test facility is a 27 node network observing the hallways and walkways of a 400 square meter wing of our building. The map in Figure 2 depicts the test area. It is occupied by 16 administrators and executives and is a central hub of activity for all 90 employees at the site. This facility represents the first phase of a 250-node network that will eventually cover both floors of our 3500 square meter facility. All observations for evaluation include the real, spontaneous, potentially multi-actor behavior of the building occupants: never a scripted or otherwise contrived scenario.

2 Related Work

Wilson and Atkeson [17] also utilize a network of motion detectors. Their system is targeted at the home, where they assume that only a few individuals will be present. This allows them to pursue a classic track-then-interpret methodology. More people means more ambiguity, and more ambiguity means exponentially more hypotheses that must be considered during tracking. Therefore, this approach is only applicable to low-census buildings, such as homes. Wilson and Atkeson also assume strategic placement of sensors. That level of specialization

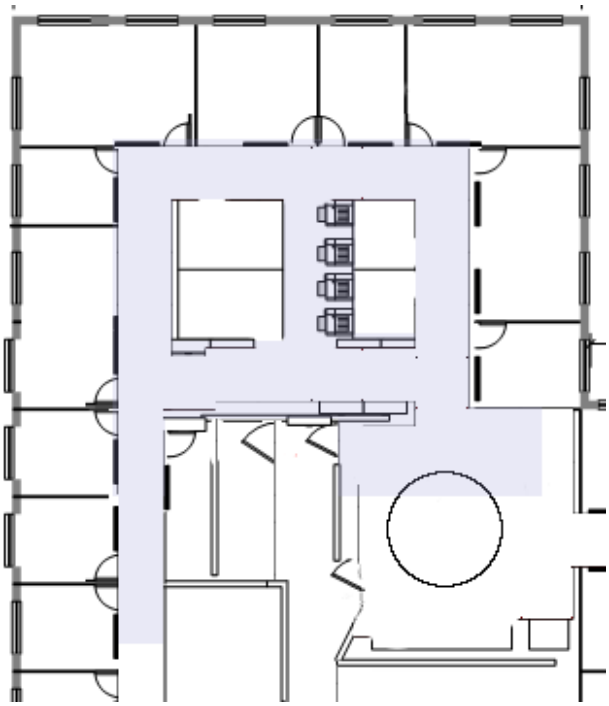

Figure 2: The floor plan of the wing where experiment data was collected. In the very center is a collection of copiers and printers. Surrounding those are a set of cubicles. On the outside are offices. The areas observed by sensors (shaded) are hallways.

is not economical in large buildings, or where usage patterns change regularly. We assume that our network will be built into the lights, outlets, and vents, and that it will likely be installed by professional electricians and ventilation engineers, rather than behavioral psychologists or eldercare specialists.

There is a significant body of literature surrounding the interpretation of human behavior in video[14, 9, 11, 2, 12]. A common thread in all of this work is that tracking is the very first stage of processing. That limits the work to sensor modalities that can provide highly accurate tracking information in the absence of any high-level inference. In particular, the ambiguities inherent in using a motion detector network can be expected to introduce enough noise in the tracking results to render most of these approaches unusable.

There are a few works that have attempted to step outside this framework[16, 6]. These systems learn task-specific state models that allow the behaviors to be recognized directly from the sensor data, without tracking. Our work follows this philosophy, and adapts it to the domain of sensor networks.

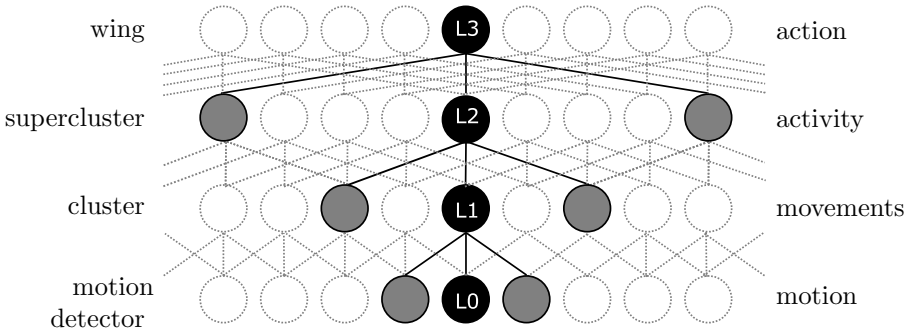

Figure 3: The Spatial relationship of the neighborhood hierarchy, from sensors (L0), to clusters (L1), superclusters (L2), and finally wings (L3).

3 Hierarchies of Neighborhoods

Bobick[1] presents a framework for thinking about the role of time and context in the interpretation of human behavior. He breaks behavior down into a tripartite hierarchy consisting of *movements*, *activities*, and *actions*. The most basic behaviors are called *movements* and have no link to the situational context and no temporal structure. Short sequences of movements may be combined with some temporal structure to form *activities*. And finally, activities may be interpreted within the larger context of the participants and the environment to recognize *actions*.

We borrow this framework, and map it onto our sensor network. Bobick defines movements to be behaviors without significant temporal structure, therefore we may recognize them with computationally light-weight models. They are also defined as not relying on the larger context, so we may detect them using only local information. Activities are defined as groups of movements, so they may cover a larger area, but may still be detected locally, without the benefit of the global context. Activities may incorporate some significant temporal structure, so we must be careful to manage the computational resources those models may impose on the sensor network. Finally, actions require global context to recognize, and may have a complex grammar to their structure. Therefore actions may best be recognized centrally, instead of within the sensor network. That is, they are best recognized at the floor, or building level. Thus, we see that this context-based hierarchy maps well onto a spatial and computation hierarchy for the sensor network. This hierarchy is abstractly illustrated in Figure 3. The motion detectors are at the bottom of the hierarchy, providing the observations. Successive levels tap progressively wider areas of context. Black nodes are the cluster leader for that level, drawing information from the black and gray nodes one level down.

The rest of this section further explores this analytical decomposition. The next section, by contrast, will cover the implementation of the system.

3.1 Topology

We make the idea of locality concrete in the form of neighborhoods of sensor nodes. To create the neighborhoods, we need the physical topology of the network. The topology tells us both which nodes are the neighbors of each leader, and also the ordering of the neighbors around each leader. Note that this does *not* require manual calibration of the system. It has been shown that it is possible to recover the geometry of a sensor network from unconstrained motion[18]. We use a similar technique, where the unconstrained motion of building inhabitants is captured for a period of time and analyzed to find statistical evidence for causal links between the nodes. The topology is inferred directly from these links. Nodes that may be physically near each other, but are separated by a wall barring direct pedestrian traffic between them will not be linked in this topology, for example. That is right, since the behavior observed by those two nodes will be independent. Typically robust topologies can be estimated from just one day of data, but it depends on the character of the data captured.

Once the topology is known, then we can construct a neighborhood around each node. Each node is given a look up table that maps the IDs of its neighbor nodes into an ordered list. For convenience of the presentation we will say that each neighbor is given a label such as "Top". "Left", "Right", "Bottom", or "Center". Note that "Top" may be arbitrarily defined to some real-world direction, say West. The exact metric definition of these labels is not important. What is important is the local relationships between the nodes is consistent: for example that the "Top" node is both counter-clockwise from the "Right" node and antipodal to the "Bottom" node. This insensitivity to imprecise or poorly documented installation is an important feature of the system.

For clarity of presentation we also assume that all neighborhoods have exactly five nodes: the center node plus the top, left, right, and bottom nodes (C, T, L, R, B). The neighborhoods will be illustrated as idealized crosses, as in Figure 4. In practice it is straightforward to generalize to neighborhoods with different numbers of adjacent neighbors.

3.2 The Node

The lowest level of our hierarchy is the individual sensor node. The single motion detector is the *Level 0* neighborhood, a degenerate neighborhood with only a single member. Our sensor nodes are wireless motion detectors that detect motion over a small area. In our case, the coverage area of each sensor is about four square meters. The motion detectors are not very capable devices, for example: they cannot differentiate one person from a group of several people, or a person from an animal. However, they are a well-developed technology that is both inexpensive and robust.

Motion detectors generate binary events in response to change in the environment, and this is the basic unit of observation that we assume as input to the higher layers of processing. Any sensing technology can be filtered to generate such a stream of binary events, and so reasonably could be substituted at this

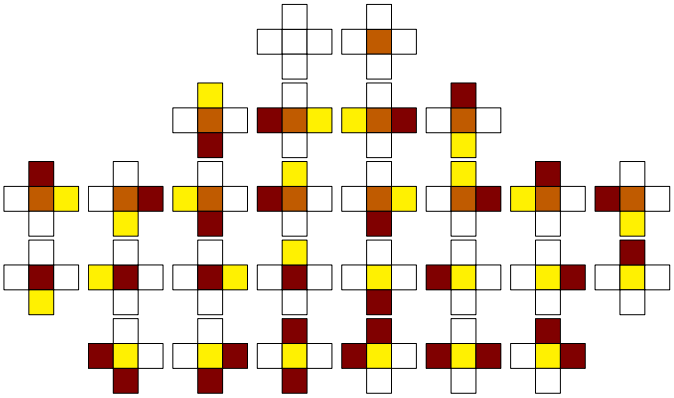

Figure 4: The neighborhood behaviors at Level 1. 1^{st} line: canonical neighborhood layout, and the still behavior. 2^{nd}: passing though movements. 3^{rd}: turning movements. 4^{th}: entering and exiting the space. 5^{th}: some joining movements (splitting not shown).

level. In the rest of the text we will call these detections *motions* to differentiate them from the more interesting *movements* in Bobick's taxonomy.

3.3 The Cluster

The next level of the hierarchy is the sensor *cluster*, or *Level 1* neighborhood. Every sensor defines a cluster: that node, plus all the nodes in the immediate vicinity. The immediate vicinity is defined as the nodes that are one step away in any direction in the network topology. We assume that the space is tiled with sensors in a grid: with little or no overlap between sensor activation fields, but also with little or no gap between activation fields. If each sensor has a radius of two meters, and the space is tiled with sensors, then a typical cluster should consist of less than ten nodes and have a radius of approximately six meters.

The clusters are where real movement recognition occurs in our system. We define a set of possible movements that occupants of a building might exhibit in the small area: passing though, standing, turning, entering, leaving, joining, and splitting. Some example movements are illustrated in Figure 4. In the illustrations time moves from dark to light, so the leftmost figure in the second row represents walking though, from bottom to top. We believe that these behaviors are so basic, and so local, that we should be able to define them, train detectors for them, and then use those detectors in novel environments. That is, so long as the sensors are installed in a similar configuration. the detectors for these movements should be invariant to the context that the cluster is immersed in, and thus can be built before installation, and reused across buildings.

The cluster leader collects the motion activations from Level 0, that is, from its neighbor nodes. The stream of motion activations are segmented into spans of contiguous time that contain motion. Within the spans, the leader computes a

number of simple features, such as: the total number of activations, the sequence of neighbor activations, and which neighbors were activated first or last. These features, which will be discussed in more detail in Section 4, are fast to compute, and are designed to make the detectors invariant to orientation and velocity. Since movements do not have complex temporal structure, the detectors take the form of naïve Bayesian classifiers. The detectors are thus computationally efficient. This is important since they are consuming motion events that are possibly being generated several times a second.

Note that, if there are 100 sensors, then there will also be 100 clusters. Each node leads one cluster, even while it participates in the many clusters around it. All behaviors are defined as happening *at* the lead sensor in a cluster. It is therefore necessary to have clusters at each node, to detect the movement behaviors that happen under that node.

3.4 The Superclusters

The next level of the hierarchy, the *Level 2* neighborhood, is the *supercluster*. Superclusters are clusters of clusters. They consist of a lead cluster and all the clusters in the immediate vicinity. If sensors are a couple of meters across, and clusters are about six meters across, then superclusters are 10-15 meters across, depending on how *immediate vicinity* is defined.

The supercluster leader receives movement detections from the constituent clusters and uses this information to perform activity recognition. That is a super cluster might infer that a meeting has occurred when its sees a sequence of "enter enter enter", that is, several people entering a room in secession. At ten meters, the superclusters cover a span of hallway, or an intersection and it's local context, or other reusable elements of building structure. While they are large enough to begin to incorporate elements of building context, we assert that they still have sufficient locality to represent reusable components of behavior.

The Level 2 models must incorporate both spatial and temporal context to recognize activities in their field of view. The models take the form of dynamic belief networks. The results we present below include three activities: visits, chatting, and meeting. *Visiting* is an activity where a person approaches a locale, dwells in that locale for a short time, and then leaves. Examples include visiting fixed resources such as a printer or coffee pot, but also short visits to an individual's office. *Chatting* is an activity that involves two people joining in a hallway, presumably to have a short conversation. *Meeting* is the activity where several people converge on a location over a period of minutes, presumably to participate in a scheduled meeting.

While we claim that these models are reusable across buildings, they obviously are not as universal as the movement models. These models are appropriate to a corporate setting, and are likely portable to other collaborative environments. However, there are probably a large number of activities that could be observed at the supercluster level. Some of these activities will have more or less meaning depending on the context. Each class of application domain (factory, retail, office, home) would need a library of activities appropriate

to that context.

3.5 The Multi-Actor Problem

A major issue when observing multiple people is the data association problem: what observations belong to which person? Most systems approach this problem by assuming that individuals are accurately tracked within the space before any interpretation is attempted. In that case, all data is associated to a track first, and the track becomes the representation used by the recognition engine.

This approach assumes that the sensors used in the system will have sufficient fidelity and coverage to make tracking possible. That implies either ubiquitous camera coverage, or the presence of tracking and identification tags attached to individual users. In situations where this assumption is valid, the prior literature is already rich with solutions. However, we claim that these assumptions are not currently valid in most buildings. Further, we claim that economic, ethical, and privacy concerns surrounding ubiquitous cameras and microphones are likely to keep many, if not most spaces from implementing such systems.

Rather than trying to distinguish individuals at the very first stage of processing, we chose instead to first draw a distinction between independent individuals and co-acting individuals, Instead of assuming that we can track individual people, we assume that people within a certain distance of each other are not independent, that they are, together, engaged in some recognizable movement. Specifically, that distance in the radius of a Level 1 neighborhood. If two people meet in a particular neighborhood, then that is recognized as a single movement: joining.

At Level 2, we must begin to resolve the multi-actor problem. The radius of a Level 2 neighborhood could be ten meters, so it is unreasonable to assert that the movements of people 5-10 meters apart are significantly correlated. Such weakly correlated actors would cause an explosion in the variability of behavior, and therefore an explosion in the number and complexity of movement models that we would need to consider. Our solution at Level 2 is to recognize all possible interpretations of the observed activity. This allows us to capture recognizable activities that might occur in the presence of distracting motions due to other actors. The ambiguity generated by these non-exclusive detections is passed up to the next level, to be resolved using external context.

3.6 Architectural Spaces

We find that above Level 2, we begin to naturally refer to the neighborhoods with architectural terms: a lab, a wing, a floor, a building, a campus. We believe that behaviors at the floor- or wing-level naturally include the notion of individuals and places: person A left her office, visited the coffee machine, and returned. We posit therefore, that the next level of processing will necessarily include some form of stochastic parsing or chaining. This process will have much in common with tracking, except that it will be based not on the similarity of signal characteristics, but instead on the consistency of interpretations along

the chain. Because this form of processing is very different from what we've described so far, and because it is well covered in the existing literature, for example see the work of Ivanov[7], we will not discuss it further in this work.

4 Implementation

This section will cover the implementation of the sensor network: both hardware implementation and analytic techniques.

4.1 The Node

The Level 0 detector is implemented in hardware, using passive infra-red (PIR) motion detectors. This is the same sensing technology used in most motion-activated lights and appliances on the market today. The sensors are inexpensive, approximately $30 per node in quantities of 500. They also require little power: they are able to run on a single nine volt battery for several months. Finally, what little they actually do, they do very reliably. We have used the widely available KC7783R sensor package from Comedia Ltd. The nodes are approximately 2cm by 3cm by 5cm. A node is pictured in Figure 1.

As it comes from the factory, the KC7783R is only able to generate events once every few seconds. We modified the boards to reduce the recovery time so that events may be generated at about 1Hz. When an individual is within view of the sensor, the moving heat source changes the thermal signature measured by the device, and a rising voltage edge is generated. The sensor is noisy and sometimes generates both false positive and false negative signals. However it is insensitive to changes in visible lighting, and therefore has a distinct advantage over cameras.

The output of the node, at the Level 0, is simply a stream of binary events. When the motion is detected, a sensor-specific ID is broadcast over a wireless network. In our research prototype system, the packet is associated to a global time stamp and copied to a conventional LAN for central storage and analysis. However, we anticipate that in a production system, the nodes would communicate only locally, passing information directly between immediate neighbors to be analyzed locally.

4.2 The Cluster

The goal of a cluster is to process the binary motion activation events from its participant sensor nodes at level 0 and classify them into one of the 17 movements. The 17 movements to recognize are: entering, leaving, turning-top-right, turning-top-left, turning-bottom-right, turning-bottom-left, turning-right-bottom, turning-right-up, turning-left-bottom, turning-left-up, walking-up, walking-down, walking-right, walking-left, still, join, split. Note that the goal is not only to recognize if a person is "turning" but which direction (right

vs. left and top vs bottom) the person is turning to with respect to an arbitrary reference point shared by all nodes. Furthermore, note that detecting movements at any point in the network only requires information from the local neighborhood or cluster (5 sensors in our case) of motion detectors.

Movement detection is accomplished in three, computationally light-weight steps: segmentation of motion events, feature extraction, and detection. The continuous stream of binary motion events is segmented using what we call idle segmentation. In idle segmentation, the leader node of the cluster starts collecting data as soon it receives a motion event from any of its neighbors and stops storing events after an idle time window of 3 seconds, containing no activations. The idle window corresponds to the average time it takes a person to walk away from a neighborhood at normal walking speed. Note that a conventional running window of fixed length could have been used to perform the segmentation of the motion events, however, idle segmentation was preferred for the lower number of false positives generated and less detections required by the system.

The features we extract are simple, yet powerful, so that they can be computed using the limited computational resources available at the sensor nodes. The first step in the feature computation is to use a look-up-table to convert the local motion event labels into the more portable top, bottom, left, right, and center labels that describe the local topological relationship between the nodes, as discussed in section 3.1. The first type of feature that is computed is temporal precedence. These features indicate the gross temporal relationship between the sensor activations. The mean value of all the timestamps associated with the motion events received from each sensor (T, B, L, R, C) is used to compute this feature. The total number of precedence features is $5 \times 5 = 25$. Another feature is the total number of motion events that comprise the segment. We also compute binary features that indicate if the center node or one of the neighbors was the first or the last sensor to be activated. Finally there are binary feature that indicate if a particular node was activated at all. For example, during an idealized example of the turning-bottom-left movement, The nodes B, C, L would be activated once each. The feature vector for that activity would be 30 elements with the following non-false values: B, C, L, $B \prec C$, $B \prec L$, $C \prec L$, $neighborsFirst$, $neighborsLast$, and $total = 3$. The notation B means "the Bottom sensor was activated." The notation $B \prec C$ means "the Bottom sensor was activated before the Center sensor."

Note that the feature vector is not a temporal sequence, it is just single vector that summarizes the entire observation sequence. In general, the features are designed to be invariant to the overall execution speed of the movement.

Once the features are extracted for a segment, detection is accomplished by using a naïve Bayesian classifier. The classifier takes the vector of 30 features and computes the likelihood for each of the 17 movements. Previous experimental testing has demonstrated that naïve Bayes networks are surprisingly good classifiers on some problem domains, despite their strict independence assumptions between attributes and the class and their computational simplicity. In fact, simple naïve networks have proven comparable to much more com-

plex algorithms, such as the C4 decision tree algorithm [10, 8, 3]. The naïve Bayesian classifier was trained on 3 weeks of hand-labeled data were the number of training examples for each movement varies from 4–28. Examples of the 17 movement categories were hand labeled by watching 7.5Hz video from 20 ceiling mounted cameras. The examples were drawn from real data collected continuously over three weeks from the administrative wing. The confusion matrix and classification results are presented below,in Section 5.

4.3 Superclusters

At Level 2, the leader of a super cluster recognizes activities by segmenting and classifying the movement detection results from its neighbor leaders at Level 1. The activities that we recognize are chatting, meeting, and visiting. The recognition of these activities requires access to a broader spatial context as well as more detailed temporal models. The segmentation of level 1 events is performed using idle segmentation with an idle window of 10 seconds. It is important to notice that different idle window lengths could be used for different activities, however, good results were obtained using the 10s window.

Because the input events at this level are discrete movement labels generated relatively infrequently (once every several seconds), we can afford to recognize them with discrete output Hidden Markov Models (HMMs)[4]. HMMs are parametric models that contain a set of states and a model of how the process transitions through those states. Each state is associated with a distinct conditional probability distribution over the space of all possible observations. In our case, the observations are the discrete movement detections from level 1. We compute the optimal number of hidden states using a cross-validation procedure over the training data and the Baum-Welch algorithm assuming a uniform prior state distribution. Since our observation variable is discrete, the observation likelihood function is represented as a discrete set of probabilities:

$$b_i(\mathbf{f}_i) = Pr\left[\mathbf{f}_i\right]$$

where $\mathbf{f}_i$ is the vector of features at index i. The transitions are assumed to be first-order Markov, shaped by a transition probability matrix $\mathbf{A}$.

$$P(\mathbf{f}_i|\mathbf{F},\lambda) = \sum_{q=1}^{N} b_q(\mathbf{f}_i)\left[\sum_{p=1}^{N} P(\mathbf{F}|Q=p,\lambda)\mathbf{a}_{pq}\right] \tag{1}$$

where a_{pq} is the element of $\mathbf{A}$ that specifies the probability of transitioning from state p to state q, Q is the current state, $\mathbf{F}$ is the collection of prior feature observations, and $b_q(\mathbf{f})$ is the probability of making the observation $\mathbf{f}$ while in state q. This model incorporates information about the temporal structure of a process in the transition matrix. It offers invariance to warping of the temporal signal. The observation process also allows it to tolerates noise in the signal.

We recognize the activities by creating one HMM for each activity to classify and computing the likelihood over the segmented data sequence using the

forward-backward algorithm[13]. The final classification result is given by the activity label associated with the HMM that obtains the highest likelihood over the segment.

The training data for each activity model is usually obtained by observing and hand labeling video sequences of the different activities. However, given the simplicity and the ease of interpreting the features used at this level (simple movement events), it is possible to directly write down a set of hypothetical training examples using common sense. This is important because it means that new activities can be hand-defined on-the-fly by the end user of the system just by having a common sense understanding of the temporal relationships among the movement events. In our case, we created 20 training examples composed of five unique examples for each activity. For example, the 'meeting' activity was defined by these sequences of movements: "entering entering", "entering entering entering", "leaving leaving", and "leaving leaving leaving", among others. This allows us to identify meetings as events were at least two people consecutively enter or leave an office or space.

5 Experimental Results

The map in Figure 2 depicts the test area. Executives occupy the offices around the outside edge of the space. Support staff occupy the cubicles in the center. At the very middle there are printers and copiers. The open hallways to the lower left and right provide access to the rest of the lab. The 16 occupants of the area form a tightly collaborative group, so there are many person-to-person behaviors that occur completely within this relatively small space. That is one reason the area was chosen for this pilot study. The space is also visited often by the 70 employees when they seek the services of the area occupants. During the course of the evaluation the occupants were notified of data gathering, but were never instructed to behave in a particular way, nor were there any artificial constraints placed on the number of people who could be moving at any given time. The data contains observations of the honest, natural behavior of the occupants of this busy space.

The Level 1 detectors were trained from a pool of hand-labeled examples in the ground-truth video sequences. We expect these models to be portable to any Level 1 neighborhood, so the examples were collected from different points in the space. The models were trained and tested in a leave-one-out cross-validation framework on the segmented data. Therefore, models were always trained and tested on data from different parts of the experimental area. The leave-one-out methodology was chosen to make the most efficient use of the limited quantity of hand-labeled data, which was very time consuming to generate. The confusion matrix is shown in Table 1. Performance over the 221 segmented test examples was 91%, with half the errors coming in the split and join movements. These movements show the most variability, and it is possible that they should be considered activities to be recognized at a higher level of processing. The rows of the table indicate the result of classifying all the examples of a known type,

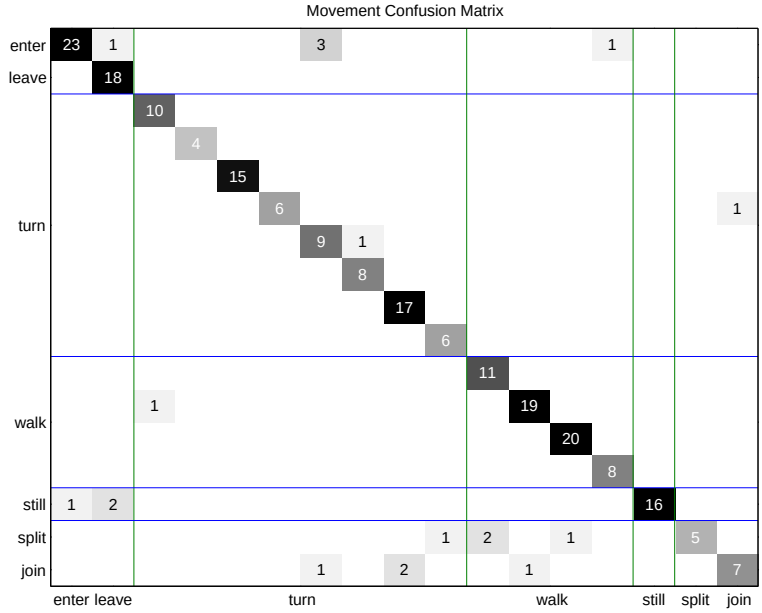

Table 1: Confusion matrix for Movement detection experiments

for example there are 28 "enter" events labeled in the test set. Numbers along the diagonal are correct: the known label on the row matches the classifier output on the column. Off-diagonal elements are errors: one enter event was incorrectly classified as a "leave" event, and three were incorrectly classified into one of the many "turn" classes.

A more realistic test of the performance of the movement detectors is run them on a long, unsegmented sequences of motion data and then compute spatial probability models that show where certain kinds of events occur. For this paper we ran the detectors on a 3 week long continuous stream of data, comprised of 3.84 million individual motion sensor activations. For example, Figure 5 depicts the spatial distribution of the walking movement. All of the figures in this section show just the walkways (shaded area) from Figure 2. The rectangles correspond to the coverage area of individual motion detectors. The walking movement is defined as walking though a neighborhood without stopping or turning. The figure shows regions of high probability (dark) along the hallways in the figure. That is, many more walking though detections were recorded along this path than elsewhere in the space. The hallway along the bottom of the map is a very high-traffic route connecting two wings of our building. Note also that at corners the walk probability is very, very low (white). This is due to the fact that it is not possible to walk at the locations: one must either turn or enter an office.

Similarly, Figure 5 shows the spatial distribution of turning movements over the space. Areas where turns are impossible, correctly show a very low probabil-

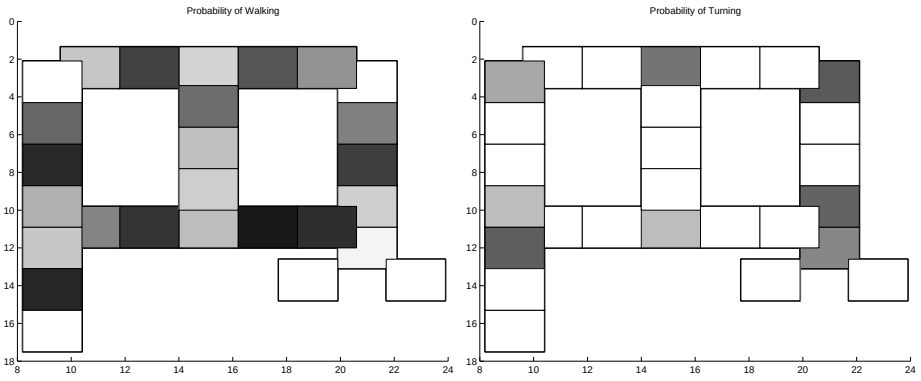

Figure 5: **Left**: The spatial distribution of walking movements in the experimental area. **Right**: The spatial distribution of turning movements in the experimental area.

ity (white) of witnessing a turning movement. Areas like corners and junctions, however, have a high probability (dark) of seeing a turning movement. These two figures, and similar plots for the other movement models, match our intuitions about the space very closely. This gives us confidence that the models are generalizing well across large spans of time. These plots summarize three weeks of movement detection results.

It is very difficult to gather ground truth for 3.84 million sensor activations. Table 1 is intended to provide precise, quantitative detail on the performance of the classifiers: illustrating the nature of mistakes on a small, carefully analyzed section of data. On the other hand, the long sequence data is intended to qualitatively illustrate that the classifiers do work on large streams of data, and do produce sensible summaries of the building activity. These summaries are consistent with the building architecture in that they do not show nonsensical behaviors such as walking into walls They are also consistent with the intuitions of building occupants. For example, correctly highlighting the high-traffic corridor within the space.

The Level 2 detectors provided a similar challenge. While going to meetings may seem more common than we sometimes might like, they are, actually, rare enough that compiling even two examples per week is difficult, and very time consuming. Instead we manually generated models that described what we anticipate scheduled meetings to look like: a few people entering the same room over the course of several minutes. The inputs, the local movement detections, are reliable and abstract enough that this seems to work. The spatial distribution of meetings, shown in Figure 6, matches our intuitions about the way the space is used. Meetings are uncommon at most locations, but occur with higher probability inside the offices of the lab directors. The squares in Figure 6 do not correspond directly to doors. Some observations zones have multiple doors, and some have no doors. The real distribution of doors can be seen in Figure 2.

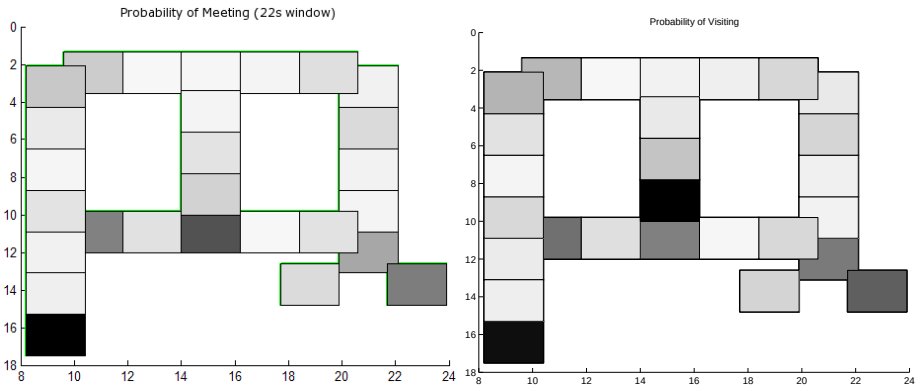

Figure 6: **Left**: The spatial distribution of meeting activity in the experimental area. **Right**: The spatial distribution of visiting activity in the experimental area.

Figure 6 shows the spatial distribution of the visiting activity. Visiting is an activity where people approach a location, loiter there briefly, and then leave. This activity is common enough that we were able to train the activity models from real data. The result is a very clean probability map. The central spikes correspond to the printer and the copier. The high probability regions in the upper left correspond to the directors' offices, the office of human resources and several key administrators. The high probability node in the lower right is the office of the vice president of business development.

In almost all the plots we see spurious detections on the boundary nodes, at the extreme bottom, left and right, of the map. These boundary nodes represent places where the closed-world assumption is broken. The movement detectors fail because they are blind to motion that happens in what should be part of their local context. This is a strong argument for completely covering spaces with sensors. Ambiguities created by incomplete coverage are very hard to resolve through inference.

6 Applications

These results suggest that a number of context-sensitive applications may soon be not only possible, but practical. An inexpensive sensor network could hence building safety by tuning emergency response to an up-to-the-minute building census. It could enhance security while preserving privacy by providing more complete context information to monitoring systems without the invasiveness or cost of ubiquitous cameras. Current energy saving devices such as motion activated lights tend to be disabled by occupants because they are annoying. By understanding more of the local context, and the habits of the users, it might be possible to build systems that better match the expectations of the people in the building.

7 Summary

We have shown that a network of simple motion detectors can be used to recover useful information about the state of a building in an efficient, scalable, and privacy-friendly manner. It is possible to recognize both simple movements (walking, loitering, entering a room) and more complex activities (visiting and meeting). We see these low- and mid-level behavior detectors as the building blocks for high-level understanding of the context of a building. This recognition is accomplished by adopting a hierarchical framework for interpretation that is carefully tuned to the requirements for recognition of various the components of human activity. The movement detectors are intentionally simple to allow modest computational engines to evaluate them despite relatively high input data rates. The movement detectors locally summarize the data, lowering the data rate and making the more demanding activity recognition models tractable, allowing us to scale up the extent of our network. We have also presented a list of movements that appear to generalize well to novel contexts. We argue that these low-level detectors can provide a powerful tool, enabling the analysis of building activity without the need for significant adaptation to novel contexts. This scalable, reusable, efficient, privacy-friendly framework for behavior understanding in buildings enables an enormous field of applications for the future of responsive buildings.

Acknowledgments

We would like to thank the reviewers and our shepherd Jeffrey Hightower for helping to improve this document with their insightful and helpful comments. The sensor hardware used in this work includes a MITes board designed at the Massachusetts Institute of Technology House_n Group.

References

[1] Aaron F. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions: Biological Sciences*, 352(1358):1257–1265, 1997.

[2] Ross Cutler and Larry Davis. Real-time periodic motion detection, analysis and applications. In *Conference on Computer and Pattern Recognition*, pages 326–331, Fort Collins, USA, 1999. IEEE.

[3] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of a simple bayesian classifier. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*. Morgan Kauffman, 1996.

[4] Ricahrd O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001.

[5] Bryan Horling, Regis Vincent, Roger Mailler, Jiaying Shen, Raphen Becker, Kyle Rawlins, and Victor Lesser. Distributed Sensor Network for Real Time Tracking. *Proceedings of the 5th International Conference on Autonomous Agents*, pages 417–424, June 2001.

[6] Yuri Ivanov, Bruce Blumberg, and Alex Pentland. Em for perceptual coding and reinforcement learning tasks. In *8th International Symposium on Intelligent Robotic Systems*, pages 93–100, Reading, UK, 2000.

[7] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, August 2000.

[8] G.H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, San Mateo, CA, 1995.

[9] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8), 1996.

[10] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, San Jose, CA, 1992. AAAI Press.

[11] David Minnen, Irfan Essa, and Thad Starner. Expectation grammars: Leveraging high-level expectations for activity recognition. In *Workshop on Event Mining, Event Detection, and Recognition in Video, held in Conjunction with Computer Vision and Pattern Recognition*, volume 2, page 626. IEEE, 2003.

[12] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268, 2001.

[13] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–285, 1989.

[14] Chris Stauffer and Eric Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):747–757, 2000.

[15] R. Szewczyk, E. Osterweil, J. Polastre, M Hamilton, A Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, June 2004.

[16] Andrew Wilson and Aaron Bobick. Realtime online adaptive gesture recognition. In *Proceedings of the International Conference on Pattern Recognition*, pages 111–6, Barcelona, Spain, September 2000.

[17] Daniel H. Wilson and Chris Atkeson. Simultaneous tracking & activity recognition (star) using many anonymous, binary sensors. In *The Third International Conference on Pervasive Computing*, pages 62–79, 2005.

[18] Christopher R. Wren and Srinivasa G. Rao. Self-configuring, lightweight sensor networks for ubiquitous computing. In *The Fifth International Conference on Ubiquitous Computing: Adjunct Proceedings*, pages 205–6, October 2003. also MERL Technical Report TR2003-24.

[19] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings 21st International Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, New York, USA, 2002.