Detecting Patterns for Assisted Living Using Sensor Networks: A Case Study

Dimitrios Lymberopoulos, Thiago Teixeira and Andreas Savvides
Electrical Engineering, Yale University
New Haven, CT, USA
{dimitrios.lymberopoulos,thiago.teixeira,andreas.savvides}@yale.edu

Abstract

In this paper we demonstrate the application of a probabilistic grammar-based formulation to detect complex activities from simple sensor measurements. In particular, we present a grammar hierarchy for identifying "cooking activity" from low-level location measurements in an assisted living application. Using real data from a pilot network deployment, we show that our system can recognize complex behaviors in a manner that is invariant across multiple different instances of the same activity. Our experiments also demonstrate that substantial data interpretation can take place at the node level, allowing the network to operate on compact symbolic representations.

1 Introduction

The growing numbers of aging baby boomers and the increasing healthcare cost obviates the need for automated services that will increase the independence and autonomy of elders living at home. Wireless sensor networks offer a promising technology for realizing such services. On one hand, small wearable devices can collect biometric information, provide feedback and automatically update medical records. On the other hand, other devices deployed inside the living environment, can monitor behaviors to prevent unsafe situations, post reminders, automate tasks and even initiate conversation.

Our work focuses on the latter aiming to create models and frameworks that would render wireless sensors capable of understanding behaviors and other patterns and react to them to provide services. In this paper, we demonstrate this possibility through a case study that focuses on the recognition of a complex pattern by putting the sensory grammars framework we proposed in [6] to work. Using data from an ongoing pilot deployment in a house [9] we demonstrate how the framework can be used to detect a complex cooking pattern from a series of time-stamped location measurements. Our experiments shows that a proper sensory grammar definition can recognize multiple instances of cooking performed by different people, using a single sensor node. The scope of our presentation is focused on providing an insight into the development of grammars for detecting complex patterns. Our up to date sensor network deployment is presented in [9] and the power aspects of our sensor nodes are explored in [4].

The success of these experiments demonstrate two very important implications for sensor networks. First, the sensory grammars framework provides a powerful tool for recognizing complex patterns from simple, low-level sensor measurements. Second, the process results in significant data reduction that can lead to long-lived, battery-operated deployments. As our experimental results demonstrate, a large number of measurements obtained from a camera sensor node are reduced to a one bit output at the sensor node level: cooking or no cooking. This capability allows our sensor network to operate on very low-bandwidth symbolic information, avoiding expensive raw data exchanges.

The rest of the paper is organized as follows. Section 2 describes in detail how cooking activity can be identified from a sequence of simple localized measurements over time. In Section 3 we evaluate the proposed grammar hierarchy on a dataset acquired from a pilot network deployment in an actual house. Section 4 provides an overview of the related work and Section 5 concludes the paper.

2 Recognizing Cooking Activity Using Sensory Grammars

The main goal is to be able to robustly recognize if a person is cooking or not by coarsely monitoring the person's activity inside a kitchen. However, recognizing individual instances of the "cooking activity" is not enough:

1. "cooking activity" should be differentiated from any other type of similar activities that might take place in the kitchen. For instance, our system should not identify the process of cleaning up the kitchen after dinner as a "cooking activity" even though the two activities

(a) (b) (c)

Figure 1. a) Kitchen layout, b) Ceiling camera view of the kitchen, c) iMote2 node with camera module.

are very similar.

2. "Cooking activity" recognition should be person as well as dish invariant. In other words, we should correctly classify the monitored activity as "cooking activity" independently of the person that is performing it and independently of the dish that is prepared.

2.1 Initial Grammar Formulation

Our description of the "act of cooking" (we will refer to this as "cooking activity" from now on) is based on the kitchen floor plan shown in Figure 1(a). To simplify our discussion we first abstract out the sensing modality by assuming that there is a sensor that can reliably detect if a person is in areas D, R, P, S, and ST in Figure 1(a). These areas denote where the subject will be located when using the dining table, refrigerator, pantry, sink, and stove respectively. The symbol E is also used to denote the exit area of the kitchen. The whole kitchen was monitored by $iMote2$ sensor nodes from Intel equipped with a camera module we designed for this application (Figure 1(c)). The module uses an OV7649 camera module from Omnivision coupled to a 162 degree lens. This camera node acquires images at 8 frames per second, downsamples it to a 128×128 resolution and uses an image processing algorithm to extract the location of a person inside the kitchen. All the processing is done on the PXA271 processor on the node, and the node transmits a binary decision if cooking is detected.

To specify a sensory grammar that recognizes cooking, we must first decompose the cooking activity into a sequence of basic actions. On the one hand, these actions should not be too abstract or too general because the difficulty of robustly detecting these actions increases significantly. On the other hand, these actions should be general enough to capture multiple instances of the activity. According to these considerations, we decompose the food preparation process into 4 main components, each of which requires a set of smaller actions:

1. **Get ingredients** from the refrigerator and/or the pantry.
2. **Prepare the dish** by spending time at the sink.
3. **Cook the food** by spending time at the stove.
4. **Serve dish** at the dining table.

Using this decomposition of the food preparation process, one could describe cooking as the ordered sequence of actions 1, 2, 3 and 4. However, this simple description of the cooking process is not adequate to capture all the different instances of a real cooking activity. Humans tend to forget and/or repeat actions without any obvious reason. For instance, people often do not get all the ingredients at once. Usually, they get a portion of them, they prepare it, then they get more ingredients and so on. Also, even in a specific activity, such as cooking, people tend to multi-task. For instance, while the food is on the stove, appetizers can be prepared at the sink or the initial preparation of the table might take place (put the dishes at the table, get sodas and drinks from the refrigerator, etc.). It becomes apparent from these observations that there is a huge number of different sequences of actions that describe a realistic cooking activity. A robust grammar definition therefore, should be able to recognize as many of these instances as possible and at the same time differentiate them from other similar activities that might take place in the monitored area.

2.2 Detailed Grammar Specification

Figure 2 shows the structure of a 2-Level grammar hierarchy for recognizing cooking activity based on the formulation presented in the previous section. At the lowest level, a sensor correlates a subject's location with areas and provides a string of symbols, where each symbol corresponds to an area in the kitchen (e.g. R, S, etc.). This string of symbols is then fed as input to the first level grammar which translates it and summarizes it to a new string of higher level semantics related to the detection of the cooking activity (e.g $AccessFood$, $CookFood$, etc.). The second-level grammar uses the high-level semantics identified at the

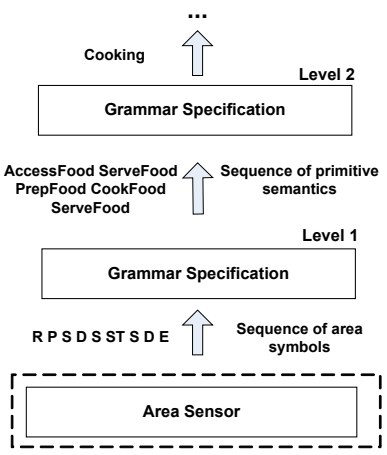


Figure 2. 2-Level grammar hierarchy for the detection of cooking activity.

immediate previous level to describe and identify a typical cooking activity. In the same way the output of the second-level grammar can be fed to any other higher level grammar for the detection of even higher level semantics.

The detailed implementation of the proposed grammar hierarchy is shown in Table 1. The grammar at Level 1 identifies the four cooking activity components ($FoodAction$) by assuming that the underlying sensing modality will provide a sequence of activity regions; the *phonemes* of this language. Lines 1 and 2 specify the non-terminal and terminal symbols of this language. The terminal symbols are fed as input to the grammar and represent the different activity regions. Therefore, an input to the Level 1 grammar consists of a string of the predefined activity regions R, P, S, ST, and D. The non-terminal symbols include the four cooking components and a set of standard symbols including the $Start$ and M symbols[1]. The non-terminal symbols in a grammar represent the semantics to which the input of the grammar is mapped. In this case, the output of the first-level grammar is any equence of the following semantics: $AccesFood$, $PrepFood$, $CookFood$, $ServeFood$.

The rest of the lines in Table 1 describe the production rules of the first-level grammar. Lines 3 and 4 describe how to recursively generate an arbitrary sequence of $FoodAction$ semantics. Line 5 describes the $FoodAction$ semantic as any of the $AccessFood$, $PrepFood$, $CookFood$ or $ServeFood$ semantics. Each one of these semantics is defined as a sequence of terminal symbols in Lines 6-9. Line 6 defines the $AccessFood$ semantic as any trip between the refrigerator R and the pantry P, that ends at the sink S or the stove ST. Lines 7 and 8 define the $PrepFood$ and $CookFood$ semantics as being at the sink S and the stove ST respectively. Line 9 describes $ServeFood$ as any sequence of trips between any of the possible areas R, P, S, and ST and the dinning table D. Note that the number of appearances of each of the terminal symbols or their order of appearance is not explicitly defined in Lines 6 and 9. However, the recursive nature of the production rules allows the unified description of numerous different expressions for the $AccessFood$ and $ServeFood$ semantics. *This shows the great generative power of grammars where very simple rules similar to the one in the human language can be used to describe numerous instances of the same complex activity*.

The grammar at Level 2 takes as input the activity components identified at Level 1 to describe a typical cooking activity. As it can be seen by Line 2, the vocabulary of the second level grammar is composed by the output semantics of the first level grammar. The output of this level is a sequence of $Cooking$ semantics. Lines 3 and 4 use recursion to allow multiple appearances of the cooking activity. The $Cooking$ semantic is described in Line 5 as any sequence of the $CookFood$, $Process$ and $ServeFood$ semantics that starts with the $Process$ or $Prepare$ semantics, ends with the $Process$ semantic and contains at least one $CookFood$ semantic. Line 6 describes the $Prepare$ semantic as any sequence of the terminal symbols excluding the $CookFood$ symbol. Line 7 defines the $Process$ semantic as any sequence of the $Prepare$ and $ServeFood$ semantics that contains at least once the $ServeFood$ semantic. *Note that because of the recursive nature of their definition, each of the production rules in Lines 5 and 6 can correspond to a huge number of different instances of the cooking activity. However, this large number of different instances are described in 7 lines of production rules for the second level grammar and 9 lines of production rules for the first level grammar*.

Since our grammar is probabilistic, each production rule is associated with a probability denoted as a superscript inside parentheses at the end of each production rule. Note that the sum of the production probabilities for each non-terminal sums up to one. In the grammars shown in Table 1, we assume that there is a uniform probability distribution for the production rules. However, in some particular scenarios these probabilities could be learned from ground truth data. This could be done by applying this grammar on real data and keeping track of how often each production rule is used. The more often a production rule is used the higher its probability.

The grammar parser makes use of these probabilities, to calculate the most probable string of non-terminal symbols for a given input string of terminal symbols. Level 1 of the grammar translates a sequence of object areas (such as R, D etc.) into a new sequence of basic cooking components ($FoodAction$) in a probabilistic way. The probabilistic na-

[1]The $Start$ symbol is a standard symbol used in grammar descriptions to represent the starting point of the grammar. We use the M symbol for recursion.

Table 1. Cooking Grammar Hierarchy

Level 1 Grammar
Input: A sequence of any of the terminal symbols:
$\{R, P, S, ST, D\}$
Output: A sequence of any of the following non-terminal symbols: $\{AccessFood, PrepFood, CookFood, ServeFood\}$

1. V_N	$=$	$\{Start, M, Action, FoodAction, CookFood, ServeFood, AccessFood, PrepFood,$
2. V_T	$=$	$\{R, P, S, ST, D\}$
3. $Start$	$\rightarrow$	$M^{(1.0)}$
4. M	$\rightarrow$	$M\ FoodAction^{(0.5)} \mid FoodAction^{(0.5)}$
5. $FoodAction$	$\rightarrow$	$AccessFood^{(0.25)} \mid PrepFood^{(0.25)} \mid CookFood^{(0.25)} \mid ServeFood^{(0.25)}$
6. $AccessFood$	$\rightarrow$	$R\,AccessFood^{(0.16)} \mid P\,AccessFood^{(0.16)} \mid R\,S^{(0.16)} \mid P\,S^{(0.16)} \mid R\,ST^{(0.16)} \mid P\,ST^{(0.16)}$
7. $PrepFood$	$\rightarrow$	$S\,PrepFood^{(0.5)} \mid S^{(0.5)}$
8. $CookFood$	$\rightarrow$	$ST\,CookFood^{(0.5)} \mid ST^{(0.5)}$
9. $ServeFood$	$\rightarrow$	$ServeFood\,S\,D^{(0.1)} \mid ServeFood\,R\,D^{(0.1)} \mid ServeFood\,ST\,D^{(0.1)} \mid ServeFood\,P\,D^{(0.1)} \mid$ $ServeFood\,D^{(0.1)} \mid S\,D^{(0.1)} \mid R\,D^{(0.1)} \mid ST\,D^{(0.1)} \mid P\,D^{(0.1)} \mid D^{(0.1)}$

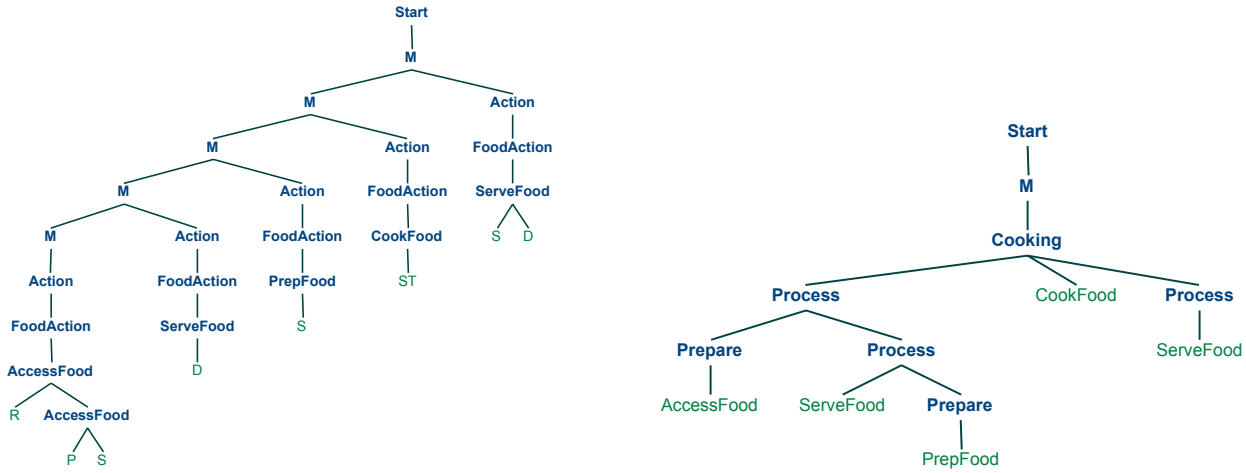
Level 2 Grammar
Input: A sequence of any of the terminal symbols:
$\{AccessFood, PrepFood, CookFood, ServeFood\}$
Output: A sequence of any of the following non-terminal symbols: $\{Cooking\}$

1. V_N	$=$	$\{Start, M, Cooking, Process, Prepare\}$
2. V_T	$=$	$\{AccessFood, PrepFood, CookFood, ServeFood\}$
3. $Start$	$\rightarrow$	$M^{(1.0)}$
4. M	$\rightarrow$	$M\ Cooking^{(0.5)} \mid Cooking^{(0.5)}$
5. $Cooking$	$\rightarrow$	$Process\,Cooking^{(0.2)} \mid CookFood\,Cooking^{(0.2)} \mid Prepare\,Cooking^{(0.2)} \mid$ $Process\,CookFood\,Process^{(0.2)} \mid Prepare\,CookFood\,Process^{(0.2)}$
6. $Prepare$	$\rightarrow$	$AccessFood\,Prepare^{(0.25)} \mid PrepFood\,Prepare^{(0.25)} \mid AccessFood^{(0.25)} \mid PrepFood^{(0.25)}$
7. $Process$	$\rightarrow$	$ServeFood\,Process^{(0.25)} \mid Prepare\,Process^{(0.25)} \mid ServeFood\,Prepare^{(0.25)} \mid ServeFood^{(0.25)}$

ture of this translation implies that the same input sequence might correspond to different sequences of the basic cooking components according to the grammar definition. For each of these possible different output sequences a probability is computed based on the individual probabilities of the production rules used to derive each output sequence. The output sequence with the highest probability is chosen as the final output. This output is then fed into a Level 2 grammar which in a similar way translates a sequence of basic cooking actions to a sequence of cooking actions. For instance, Figure 3 shows the most probable parse trees for both levels and for a given input sequence of object areas. As it can be easily verified, each edge in the tree corresponds to a production rule of the corresponding grammar. The probability assigned to the parse tree is computed by multiplying the probabilities at each branch from the root to the leaves and then summing the probabilities of all the branches in the tree. For instance, in Figure 3(a) there are 5 branches with probabilities (p_1 corresponds to the leftmost branch and p_5 to the rightmost branch):

$$
\begin{aligned}
p_1 &= (0.5)^4 \times 0.5 \times 0.5 \times 0.25 \times (0.166)^2 = 0.0001075 \\
p_2 &= (0.5)^4 \times 0.5 \times 0.25 \times 0.1 = 0.000781 \\
p_3 &= (0.5)^3 \times 0.5 \times 0.25 \times 0.5 = 0.0078125 \\
p_4 &= (0.5)^2 \times 0.5 \times 0.25 \times 0.5 = 0.015625 \\
p_5 &= (0.5) \times 0.5 \times 0.25 \times 0.1 = 0.00625
\end{aligned}
$$

The probability of the tree is equal to $\sum_{i=1}^{5} p_i = 0.0305$. In exactly the same way, a probability for the tree shown in Figure 3(b) can be computed using again the probabilities assigned to the production rules shown in Table 1.



(a) Level-1 parse tree

(b) Level-2 parse tree

Figure 3. Example parse trees for the 2-Level cooking grammar hierarchy.

3 Evaluation

Our scheme was evaluated on data acquired in a series of experiments in the kitchen deployment described earlier. In every experiment the person in the kitchen was preparing either breakfast or dinner. The data collection started when the person was entering the kitchen or while the person was already in the kitchen. It was stopped when the person started eating breakfast or dinner at the dining table. The person in each experiment was not aware of what he would have to cook until a couple of minutes before the recording of the data. This prevented the person from using pre-meditated moves. The person cooking was also unaware of the actual grammar hierarchy definition. In total, 10 cooking traces were collected lasting from approximately 10 minutes (breakfast) to 50 minutes (dinner) each.

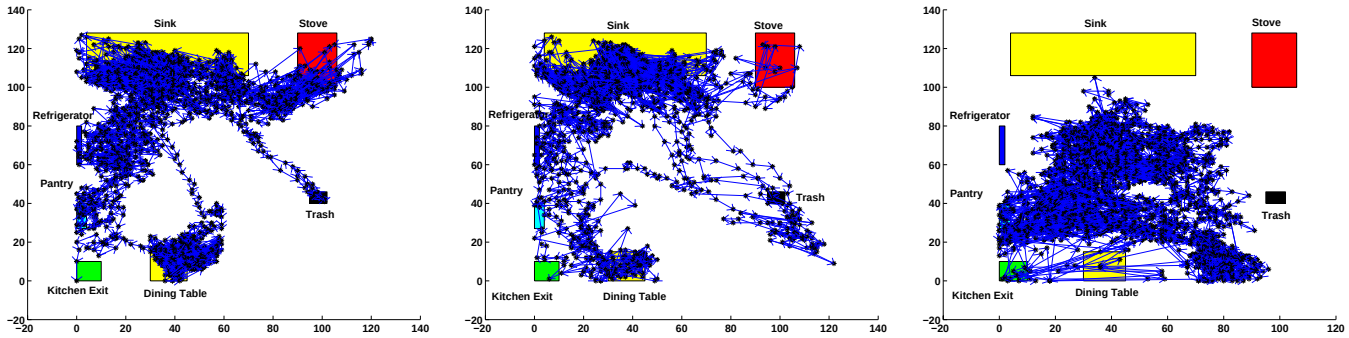
In order to challenge the capabilities of the proposed scheme, we also recorded a set of activities other than cooking in the same kitchen area. In total, 5 different traces were recorded on different days. These activities included cleaning the kitchen after having dinner, cleaning the floor of the kitchen and sorting the groceries after returning from the super-market. Especially when cleaning up the kitchen after having dinner, the areas visited are almost the same as when cooking. This can be seen in Figure 4(a) and Figure 4(b). The recorded traces of image locations are very similar. However, the grammar hierarchy should only recognize the cooking activity trace.

For each recorded activity trace the ground truth area information activity was also recorded. This was done manually by a person that examined a recorded video for each recorded trace. The ground truth area information was used to investigate the false negatives and false positives of the area sensor.

Table 2 shows the recognition results of the proposed grammar hierarchy for all the recorded activities and for both the ground truth data and the actual data provided by the area sensor. In both cases, all the cooking activities are correctly classified. What is even more interesting is the fact that the proposed scheme can differentiate between very similar activities such as cooking and cleaning. This demonstrates that the grammar definition is general enough to capture various instances of cooking activity, but at the same time it is specific enough to robustly differentiate cooking from other similar activities. This is due to the fact that the grammar hierarchy definition imposes specific restrictions into the sequence of measured locations over time. For instance, when people are cleaning, they either do not visit the stove area or some other areas (i.e. cleaning the floor or sorting the groceries) or they do not move to the dining table after cleaning everything. This type of restrictions in the description of the cooking activity allow the system to differentiate between cooking and cleaning.

However, as for example shown in Table 2, the proposed system fails to correctly classify the cleaning activity shown in Figure 4(b) when the ground truth data is used. This is due to the successful calibration of the area sensor. The table area was defined by using real image locations acquired when a person was sitting at the table. This data gave us a very precise definition of the table area. While cleaning the table (i.e. picking up the plates etc.), people do not sit at the dining table and therefore the area sensor would rarely detect the dining table area in such a case. However, this table area information is recorded in the ground truth data resulting into an incorrect classification result.

The experimental data provides insight on how to better calibrate the area sensor. Table 3 shows the number of area symbols generated by the area sensor versus the



(a) Cooking dinner **(b)** Cleaning up after dinner **(c)** Cleaning the floor

Figure 4. Area definitions and example data sets.

Table 2. Recognition performance results

Kitchen Activity	Number of Traces	Correctly Classified (Ground Truth)	Correctly Classified (Filtered)
Cooking	10	10	10
Cleaning	5	4	5
Other	1	1	1

Table 3. The effect of imperfect sensing

Kitchen Activity	Number of Areas (Ground Truth)	Number of Areas (After Filtering)
Dinner	116	109
Breakfast	15	19
Cleaning 1	12	9

ground truth number of area symbols for three of the collected traces. It is clear that the area sensor gives both false positives and false negatives. The false positives are caused by the fact that the area of the kitchen used in our experiments was small. As a result, the areas of the refrigerator and the pantry (Figure 1) are very close and when a person tries to use the pantry it is possible that the refrigerator area will also be recognized. The false negatives are mainly caused by small movements of the person in the kitchen that cannot be robustly captured at the 128×128 resolution. For instance, in many cases the person was able to reach the sink by simply stretching but without moving out of the stove area. In this case, the sink would appear in the ground truth data but not in the output of the area sensor.

In order to reduce the size of the input to the grammar hierarchy (and thus its execution time) a 3-stage sensor calibration mechanism was implemented. The first stage converts the time series of image locations to a time series of visited areas. The $Undefined$ area symbol can be produced when the monitored person is moving in any place in the kitchen that is not one of the predefined areas. The second filtering stage, removes all the $Undefined$ area symbols because they are not used by our grammar and they increase the size of the input to the grammar hierarchy. After removing the $Undefined$ area symbols, consecutive appearances of the same area symbol might appear. The third filtering stage merges all these consecutive appearances to a single area symbol.

The overall average information reduction from this 3-stage calibration mechanism is approximately 99%. This is the percentage of reduction in the number of symbols that are given as input to the grammar hierarchy with respect to the number of image locations initially recorded. The average percentages of information reduction for each one of the three stages of filtering are: 85% (from translating the raw image locations to areas) , 50% (from simply removing all the $Undefined$ areas) and 90% (from merging consecutive area symbols) respectively. Due to the sensor calibration the number of symbols eventually fed as input to the grammar hierarchy (approximately 10 to 100) are orders of magnitude less than the initial number of image centroids recorded (2583 to 6648), as shown in Table 4. These numbers demonstrate the feasibility of such a system running in real time on a sensor network. An input of 10 to 20 symbols is relatively small and can be parsed in a very short period of time even on an a sensor node as will be made clear in the next section. In addition, the fact that activities lasting as much as 50 minutes can be reduced down to a sequence of only 100 symbols shows that modeling human activity as a sequence of actions could meet the real time requirements and limitations of sensor networks. To test the feasibility of running parsing on the sensor node we have implemented cooking recognition on the iMote2 processor. The typical execution times as a function of the input symbols for both levels vary from a couple of hundred of microseconds (10 input symbols) up to a few milliseconds (100 input symbols).

4 Related Work

Researchers at Intel Research and MIT have studied human activity recognition in the context of assisted living ap-

Table 4. The effect of raw data filtering.

Kitchen Activity	Number of Centroids	Number of Areas After Filtering		
		Stage 1	Stage 2	Stage 3
Dinner	6648	1456	728	109
Breakfast	2924	446	223	12
Cleaning	2583	421	211	9

plications using RFID tags [1, 8, 7, 5]. This approach requires extensive tagging of objects and people with RFID tags. While our work is absolutely compatible and it could be transparently used with these types of network setups, it makes a significant contribution: it demonstrates that the hierarchical organization of probabilistic grammars provides enough infrerence power for recognizing human activity patterns from low level sensor measurements.

Sensor networks for abnormal activity detection have also been proposed [3, 2] . In this approach, statistical analysis of long-term real data is used to define what a "normal" activity is. Every activity that deviates from the "normal" activity profile is considered to be "abnormal". While this method can be useful, it does not provide enough information about the exact service that has to be triggered by the system. Different types of abnormal activities require different types of services to be triggered. Furthermore, in many cases it is very useful to be aware of the exact activities of a person even though these activities are not considered to be "abnormal". For instance, a sensor network that can understand human behaviors could be used to assist elders living alone.

The approach demonstrated in this paper is also complementary with the Semantic Streams work presented in [10]. Grammar hierarchies, like the one described in this paper, provide a structured bottom-up processing of the sensor data for generating higher level semantics in a way that is similar to streams. These semantics can be easily become the basic processing elemets for answering higher level queries through the top-down user programming interface proposed in [10].

5 Conclusions and Future Work

In this paper we have used the cooking grammar as an example to demonstrate how sensory grammars can be used to detect complex patterns from simple measurements. Our experiences from this experiment indicate that there is a learning curve associated with writing good grammars. Notheless, the sensory grammars framework shifts the effort in programming the sensor network from low-level embedded systems programming to high level grammar scripting, thus allowing domain experts to focus on sophisticated pattern searching using distributed sensor networks. Beyond the assisted living application presented here, our case study illustrates how one could reason with locations and a map, something that could be applied to a much larger scale with sensor networks. The same framework is also directly applicable to other patterns in many domains. As part of our future work, we plan to refine our middleware architecture for sensory grammars and apply it to different domains. Our home deployment of a 6-node sensor network is currently in its second month of deployment and a detailed library of behaviors is currently being developed using the collected dataset. Finally, in addition to procesing spacial information as illustrated in this paper, our middleware has already been expanded to support temporal reasoning. In the near future we plan to conduct analysis on longer term data traces.

Acknowledgments

This work was partially funded by the National Science Foundation under awards #0448082 and #0529186. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. P. et. al. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 03(4):50–57, 2004.

[2] S. R. et. al. Abnormal activity detection in video sequences using learnt probability densities. In *TENCON*, October 2003.

[3] W. P. et. al. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of AAAI*, July 2005.

[4] D. Jung, T. Teixeira, and A. Savvides. Model based design exploration of wireless sensor node lifetimes. In *Proceedings of EWSN*, April 2007.

[5] L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational markov models. In *Nineteenth International Joint Conference on Artificial Intelligence*, 2005.

[6] D. Lymberopoulos, A. Ogale, A. Savvides, and Y. Aloimonos. A sensory grammar for inferring behaviors in sensor networks. In *Proceedings of IPSN*, April 2006.

[7] D. Patterson and M. P. D. Fox, H. Kautz. Fine-grained activity recognition by aggregating abstract object usage. In *IEEE International Symposium on Wearable Computers*, October 2005.

[8] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home setting using simple and ubiquitous sensors. In *PERVASIVE 2004*, 2004.

[9] T. Teixeira, D. Lymberopoulos, and A. Savvides. Experiences from a home sensor network deployment for assisted living. In *to appear in WiDeploy 2007*, April 2007.

[10] K. Whitehouse, J. Liu, and F. Zhao. Semantic streams: A framework for the composable semantic interpretation of sensor data. In *Proceedings of EWSN*, February 2006.