A Sensory Grammar for Inferring Behaviors in Sensor Networks

Dimitrios Lymberopoulos, Abhijit S. Ogale, Andreas Savvides, Yiannis Aloimonos
Dept. of Electrical Engineering, Yale University, New Haven, CT, USA
Dept. of Computer Science, University of Maryland, College Park, MD, USA
{dimitrios.lymberopoulos,andreas.savvides}@yale.edu,{ogale,yiannis}@cs.umd.edu

ABSTRACT

The ability of a sensor network to parse out observable activities into a set of distinguishable actions is a powerful feature that can potentially enable many applications of sensor networks to everyday life situations. In this paper we introduce a framework that uses a hierarchy of Probabilistic Context Free Grammars (PCFGs) to perform such parsing. The power of the framework comes from the hierarchical organization of grammars that allows the use of simple local sensor measurements for reasoning about more macroscopic behaviors. Our presentation describes how to use a set of phonemes to construct grammars and how to achieve distributed operation using a messaging model. The proposed framework is flexible. It can be mapped to a network hierarchy or can be applied sequentially and across the network to infer behaviors as they unfold in space and time. We demonstrate this functionality by inferring simple motion patterns using a sequence of simple direction vectors obtained from our camera sensor network testbed.

Categories and Subject Descriptors: C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems

General Terms: Algorithms, Human Factors

Keywords: Human Activity, Behavior Identification, PCFG, Sensor Grammars, Sensor Networks.

1. INTRODUCTION

Sensor networks are emerging as a very promising technology for the future, that is already beginning to impact many aspects of science, engineering and society. As sensor networks come closer to everyday life applications, the need for understanding the behaviors of the phenomena they observe in physical space is becoming more apparent. In most applications, intelligent sensors are expected to acquire context awareness, observe their environments, understand behaviors and seamlessly react to provide a set of services to their users. For instance, sensor networks deployed to provide safety in the workplace will be expected to sense the development of a hazardous situation, generate warnings and take steps to avoid it. In construction sites for example, such a sensor network would prevent workers from falling into empty elevator shafts or walking under heavy loads while they are moved by a crane. In security systems, sensor networks are expected to provide proactive intelligence. They should autonomously recognize suspicious activity and trigger actions and alarms without requiring humans in the loop. Furthermore, with the capability of understanding human motion behaviors in space, one could also envision the use of sensor networks in entertainment applications.

For a sensor network to be powerful, it needs to be able to develop a model of the activity that is taking place in the area that the network senses. This of course involves people and their actions and it has been subject of research in Surveillance and Computer Vision applications [13, 4, 1, 14]. Among these approaches a recent development [9] proposes a new framework that has a number of appealing properties. The framework suggests that the problem of human action understanding is similar in spirit to the speech understanding problem, where phonemes are replaced by body parts and their movements indexed on the human silhouette that is extracted from the video, and morphemes (or words) are the appropriate grouping of these silhouettes into actions (verbs), which are then grouped together by syntax. Techniques such as Hidden Markov Models [12] or Probabilistic Context Free Grammars [15] are successfully used to parse the videos and recognize human action.

The approach requires the application of various filters in order to discover primitives and it may need more computational power than the one available to the sensor networks that we are studying. We can however capitalize on the grammatical structure of human behavior as it is exemplified in the work described before, and propose a new generic framework for analyzing large scale human behavior using sensor networks with minimal computational capacity and communication capabilities. We believe that before a sensor network discovers that two persons exchanged their briefcases in a corridor, it should be able to idenitfy simpler things like where people are going. Suppose, for example, that we wish to monitor (and reason about) the movement of people inside a building. Then we can write grammars specifying the behaviors we wish to capture and thus turn the interpretation problem of the sensor network to a sensory parsing problem. Depending on the behaviors that need to be understood, the grammars may be context free or context sensitive, thus creating an interesting complexity hierarchy

IPSN'06, April 19–21, 2006, Nashville, Tennessee, USA.

in the space of sensor networks. By posing the problem in this way we avoid ad hoc development where one tailors the algorithms to the particular application. Instead, we would like to pose a large class of questions surrounding the interpretation of the signals from sensor networks as parsing problems, where the behavior of interest can be specified by a grammar (or any other equivalent or similar mechanism, such probabilistic grammars, Petri nets, etc.). The parsing problems become especially interesting because we enforce constraints about communication and computation. Our nodes (sensors) will be able to transmit few bytes to other nodes (neighbors) and will have restricted computation capabilities.

In this paper we present such an approach. We develop a framework of hierarchical sensory grammars to *parse out observable activities into a set of distinguishable actions*. The power of such framework comes from the hierarchical organization of reasoning. This allows the use of simple localized sensor measurements to reason about more macroscopic behaviors taking place in space and time. At the lowest levels of the grammar hierarchy, a grammar converts sensor measurements into a set of symbols that become the inputs of higher order grammars. Each grammar in the hierarchy produces an output that *interprets* and *summarizes* its inputs and thus effectively reduces the data that need to be propagated to the higher layers. Our approach offers several favorable attributes for sensor network implementation. Computations are lightweight, training is confined to the lowest layer of the hierarchy, and grammar hierarchies map naturally to network hierarchies.

Our framework has three main components: 1) **phonemes**, 2) **hierarchical grammar construction** and 3) **messaging model**. To realize this framework, in section 3 we will describe phonemes, Probabilistic Context Free Languages, grammar hierarchies and messaging models. In section 4 we will provide a concrete example of our framework based on location data that were acquired by our camera sensor network testbed. Although our presentation focuses on the interpretation of motion behaviors in physical space, the same framework could be applied to recognize other more abstract behaviors such as the identification of network faults or to identify patterns in sensor data.

2. PARSING BEHAVIORS IN SENSOR NETWORKS

The need for recognizing behaviors comes at all levels of a sensor network hierarchy during the data collection process. In many applications, it is more practical to filter out redundant information as close to the sensors as possible so as to reduce the computation and communication requirements across the network. At the same time, we would like to interpret the sensed information so that the network can understand what is happening in the physical world and provide a response. Camera sensor networks are a good example for this requirement. Camera sensors can provide qualitatively and quantitatively better information about a scene. Communicating and processing images across the network however is an expensive process requiring significant communication and processing bandwidth. A better approach would be to process the image information locally at the node level. Nodes in the network can then interpret a behavior by exchanging short packets containing symbolic information about their sensors.

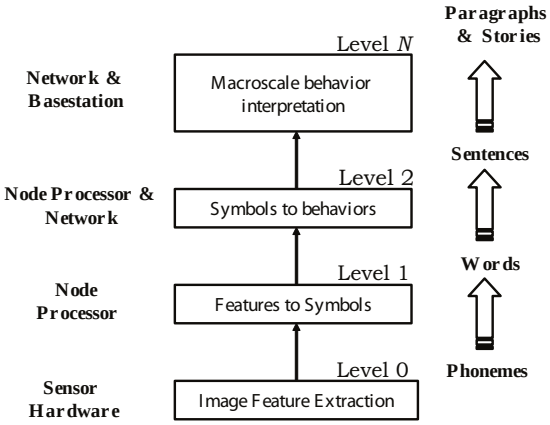

Figure 1: Deployment Scenario.

Our framework creates a modularized stack, as shown in Figure 1, that bears similarities to languages. Intelligent sensors at the physical layer will extract a set of features from the environment that are analogous to phonemes in natural language(Level 0). These phonemes will eventually be interpreted into actions (or verbs), sentences, paragraphs and stories that describe the happenings inside the sensor network(Level 1 to Level N). The proposed framework can be applied sequentially and across the network to interpret behaviors that unfold in space and time.

2.1 Challenges in parsing behaviors

The lack of global information on individual sensor nodes makes the problem of training the sensor network to parse macro-behaviors extremely challenging. In particular, the training of the sensor network should be:

1. **Topology/Location independent:** The detection and recognition of behaviors is independent of the location in the network where the behavior takes place. Therefore, the training data for such a sensor network should not embed any location or topology information.

2. **Scaling independent:** The detection and recognition of a behavior is independent of the scale of the behavior.

For instance, consider a sensor network that recognizes people moving on a circle in a room. The topology/location independence rule implies that the "moving on a circle" behavior is the same independently of the location in the room where it is performed. "Moving on a circle" close to the exit of the room is the same as "moving on a circle" in the middle of the room. The scaling independence rule implies that the "moving on a circle" behavior is independent of the actual size of the circle. The motion pattern of a human moving on a circle of $1m$ radius is the same as the motion pattern of a human moving on a circle of $5m$ radius.

The above two requirements have to be enforced on the training of the sensor network for two main reasons:

(i) **Size of the training data:** Differentiating among the same behaviors that take place at different locations in the network or among behaviors that appear at different scales, would create a huge training data set.

(ii) **Network scalability:** This huge training data set would also depend on the topology and the size of the network affecting the flexibility and scalability of the network. If a number of nodes dies or a number of nodes is added to

the network, the topology and relative locations of the nodes automatically change, dictating the partial re-training of the sensor network.

To avoid running into scalability issues, our framework simplifies the extend of training required by the sensor network by adopting a hierarchy of grammars. Instead of requiring to train the network for all behaviors, we structure our framework so as to simplify and reduce the amount of required training. The grammar at the bottom of the hierarchy operates directly on the sensor measurements and converts them into a more symbolic form, that becomes the input for higher order grammars. *This structure not only reduces the amount of training required but also facilitates the interaction of multiple sensing modalities at the higher levels of the hierarchy.* If one is able to design a grammar (or engineer sensors) in a way that a Level-0 grammar can convert the measurements into a more symbolic form, then one could use the outputs of multiple Level-0 grammars that represent multiple sensing modalities, as inputs to higher level grammars that reason about behaviors.

2.2 Case Study using Tracking Data

To make our presentation more concrete, we present our framework in the context similar to tracking applications which have been well studied in sensor networks. The study case that will be discussed in detail in section 4, uses a trace of location data (a time series of location measurements) extracted form a camera sensor network testbed. As we will explain later on, our testbed is configured to generate a stream of locations when observing a single target moving along the sensor field. The framework we will describe in the next section will use this "tracking" data to parse the motions of the target into lines, left and right turns, U-turns and S-turns. To simplify our discussion, throughout the paper we assume that the system observes a single target, and that the fields of view of the camera sensor nodes are non-overlapping. Before describing this in more detail we first describe the framework design methodology.

3. FRAMEWORK DESIGN METHODOLOGY

To realize our framework in this section we discuss four main topics: 1. *Identifying the phonemes*, 2. *Specifying the grammar*, 3. *Hierarchical Language Construction*, and 4. *Messaging Model*.

3.1 Identifying the Phonemes

Phonemes are the most fundamental component of the framework. The network designer must know the application well enough to specify a set of terminal symbols for the language. An important direction of our framework is that these phonemes should be specified in a way that will allow their use at the node and sensor level. Ideally, the sensor should be intelligent enough to output features that can be used as phonemes. This implies that a Level-0 grammar would be embedded in the sensor node hardware. Alternatively, the sensor node processor will have to interpret raw sensor data as phonemes. By confining as much as possible the production of phonemes to a Level-0 grammar, we essentially confine the training requirements to the sensor node level. Once the phonemes are successfully extracted, the rest of the network will be able to interpret complex behaviors by operating on a vocabulary generated at each level of the hierarchy.

In human language recognition, speech processing operates on the language phonemes. In computer vision, these terminal symbols can be the keyframes extracted from a sequence of images observing and action. These keyframes are the minima and maxima points in motion behaviors that are sufficient to describe a behavior [9]. Handwriting recognition approaches use direction, angle or velocity information [2]. In our case study, the phonemes are straight lines inferred from a set of direction vectors extracted from a camera sensor.

3.2 Specifying the PCFG

Before delving into the details of PCFGs we have to first define context-free grammars [15]. A context-free grammar G is an ordered quadruple $\langle V_N, V_T, Start, Pr\rangle$ where:

- V_N is an alphabet of non-terminal symbols.
- V_T is an alphabet of terminal symbols.
- $V_N \cap V_T = \emptyset$. $V = V_N \cup V_T$ is called the vocabulary.
- $Start \in V_N$ is the start symbol.
- Pr is a finite nonempty subset of $V_N \times V^*$ called the production rules.

The set of all strings that are composed of non-terminal, terminal or both non-terminal and terminal symbols are represented by V_N^*,V_T^*, and V^* respectively. Let capital letters: $A, B, C, \ldots$ represent the non-terminal symbols and small letters: $a, b, c, \ldots$ represent the terminal symbols. The production rules of a context-free grammar are then written as: $A \rightarrow a$, where the lefthand side can be any non-terminal symbol while the righthand side can be any combination of terminal and non-terminal symbols.

Starting from the start symbol $Start$ and by successively applying the same or different production rules, differrent strings can be generated. In general, we say that string α derives string β ($\alpha \Rightarrow \beta$) if there is a sequence: $\alpha = \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n = \beta, n \geq 0$, of strings in V^* such that: $\alpha_0 \Rightarrow \alpha_1, \alpha_1 \Rightarrow \alpha_2, \ldots, \alpha_{n-1} \Rightarrow \alpha_n$. The language $L(G)$ generated by a context-free grammar G is the set: $L(G) = \{x | Start \Rightarrow x, x \in V_T^*\}$. In other words, $L(G)$ is the set of all terminal strings derivable from the start symbol $Start$.

Having defined in detail the context-free grammars (CFG), we can describe a probabilistic context-free grammar PG as a CFG paired with a set of probabilities $P = \{p_{ij}\}$[15]. This set of probabilities must satisfy the following rules:

1. For each production $P_{ij} \in Pr$ there is one and only one probability $p_{ij} \in P$.
2. $0 < p_{ij} \leq 1, \forall i, j$
3. For every i with $1 \leq i \leq |V_N|$: $\sum_{1 \leq j \leq n_i} p_{ij} = 1$, where n_i is the number of productions with the ith non-terminal on the lefthand side.

This definition assigns a constant probability to each production rule in grammar G. These production probabilities can be used to generate probabilities for sentences. The basic assumption is that the choice of production rules used in deriving a sentence is "context-free" in the sense that each rule is chosen independently of all the others in the

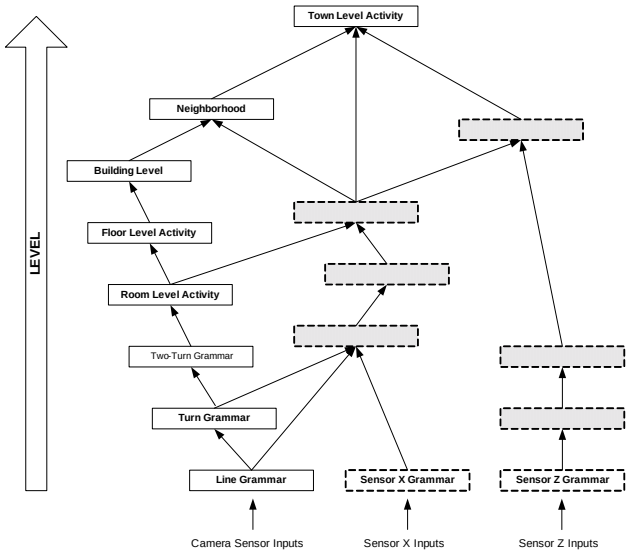

Figure 2: Example grammar hierarchy.

derivation. This allows us to compute the probability of a sentence as the product of the production probabilities that were used to generate this sentence. If the same sentence can be derived in more than one ways then its probability is the sum of the probabilities of all possible derivations.

3.3 Hierarchical Grammar Construction

Perhaps the most versatile feature of the framework is the ability to create a hierarchical grammar. At Level-0 the symbols in the vocabulary of the PCFG are obtained from the sensors, and the probabilities p_{ij} for each symbol need to be obtained by off-line training. When a measurement is obtained, the Level-0 grammar uses the measurement to update the probability for each production (behavior). These outputs become the vocabulary for grammars in the subsequent levels of the grammar hierarchy. In this way each level of the hierarchy computes a set of probabilities for each behavior it describes. The lower levels of the hierarchy infer simple behaviors that help higher levels infer macro-behaviors.

Consider for example the grammar hierarchy in Fig. 2. The first three levels inside the grey box describe the hierarchy that we will define in detail in section 4. Level-0 processes inputs from cameras to infer probabilities for four types of straight lines, pointing north, south, east and west. The outputs of Level-0 become the terminal symbols for Level-1, that classifies the line behavior to left, right and 180-degree turns. A Level-2 grammar then operates on the outputs of Level-1 to infer another set of probabilities for S-turns and U-turns. A higher level grammar can detect zig-zag and spirals, that can provide inputs to yet another grammar to infer what a person does at the room level (i.e walking in circles, dancing etc). Similarly, one could expand the hierarchy both horizontally, by including more sensing modalities, or vertically to infer activities in a room, floor, building, neighborhood, town.

This example demonstrates the promise of hierarchical grammars. Each level of the hierarchy *interprets* the output of the previous level and *summarizes* the information into a higher order behavior. Through such hierarchy one can correlate measurements from a local scale to infer a behavior that takes place at a more macroscopic level. When mapped onto a network hierarchy, a grammar hierarchy honors a very desirable attribute for communication in sensor networks. Each grammar level interprets information and produces outputs in more compact form, that reduces the amount of data that needs to be propagated across the network hierarchy. This is explained in more detail with the messaging model.

3.4 Messaging Model

The messaging model specifies how information flows across the grammar hierarchy, and between nodes in the network. To explain the messaging model let us consider two neighboring nodes A and B with adjacent, non-overlapping sensing regions observing a target moving across the sensing region of A and into the sensing region of B. Each node runs an instance of the three-level grammar hierarchy of our example application. Every time there is a new observation, the node that made the observation updates the probabilities of detection for all the behaviors at all the levels of its grammar hierarchy.

When the target moves out of the sensing range of node A into the sensing range of node B, node A has to transmit its update probabilities to node B to continue the computation. In particular, given a PCFG G and a string $w_1 w_2 \dots w_n$ we want to find the most probable parse tree in the grammar for the given string:

$$argmax_{tree} P(tree|w_1 \dots w_n, G) \quad (1)$$

Note that the maximization problem in the last equation is a global maximization problem since information about the whole string is required. However, in the case of a distributed sensor network, nodes make local observations and they are not aware of the observations made at other nodes. This means that each node can observe only a substring of $w_1 w_2 \dots w_n$ at different places in space and time. Consequently, such a global maximization would be feasible only if all nodes were communicating their local observations to a central node, where the actual maximization problem could be solved.

Fortunately, it turns out that the global maximization problem in equation 1 can be decomposed to a set of local maximization problems. This can de done using a dynamic programming algorithm called the Viterbi search path algorithm [3]. Given the PCFG G and the string $w_1 w_2 \dots w_n$ we want to find the most probable derivation tree of the given string. Let $V[X, i, j]$ be the maximum probability of any single derivation of the string $w_i \dots w_{j-1}$ from the non terminal symbol X. Then, in normal Chomsky form $\forall j > i + 1$:

$$V[X, i, i+1] = P(X \rightarrow w_i) \quad (2)$$

$$V[X, i, j] = max_{X \rightarrow YZ}^{i<k<j} P(X \rightarrow YZ) V[Y, i, k] V[Z, k, j] \quad (3)$$

The last set of equations shows that the initial global maximization problem can be decomposed to a sequence of local maximization problems. This means that the sensor node that makes the k^{th} observation (node B in our example) needs to run a local maximization problem based on: 1) its local observation, 2) all the possible production rules of the grammar based on its local observation, and 3) the result of the local maximization on the sensor node that made the $k-1$ observation (node A in our example).

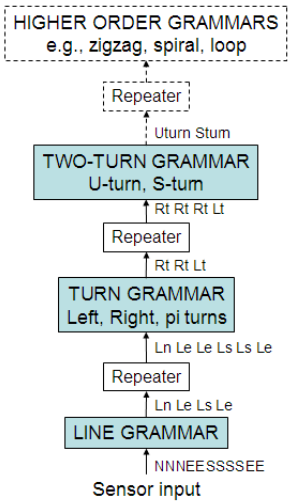

Figure 3: Hierarchical behavior interpretation using a sensory grammar.

Note that the only non-local information needed by the node that makes the k^{th} observation is the result of the maximization on the sensor node that made the $k-1$ observation. In addition, all the local maximizations performed for the observations $1, 2, \ldots, k-2$ are not needed at step k because all this information is already embedded in the maximization performed at the $k-1$ step. Note that the actual amount of data that needs to be transmitted to the node that computes step k is very small. For instance, the three level grammar used in our case study in section 4, computes three probabilities at each level. Assuming that each probability is stored in a 16-bit variable, then each transition of a target from the sensing region of one node to the sensing region of another node will result in the transmission of a packet carrying 18 bytes of state data. Here we assume that the packet transmission is triggered by an external handoff process that recognizes the transition of the target from one region to the next.

4. CASE STUDY: INTERPRETING BEHAVIORS FROM TRACKING DATA

This section describes an instance of the framework that uses a series of locations in time obtained from a camera network. Our discussion begins with the use of a binary 1-bit sensor that can sense presence or absence of a person in its sensing range. This assumption will be relaxed later on when we describe our testbed setup. Our goal is to recognize the motion patterns of a person walking along a corridor inside a building. More specifically, we are interested in recognizing left-turns, right-turns, U-turns and S-turns.

4.1 Phoneme and PCFG Specification

Assume that we have a grid of sensor nodes in the corridor of a building indicating the presence or absence of a person. The person is only allowed to move in four directions along this grid: North (N), South (S), East (E), and West (W). These four symbols are the terminal symbols or phonemes of the grammar. The sensor grid returns a string of these phonemes as a person moves through the grid, e.g., NNNEESSSSEE.

As a first step, we can define a grammar to detect straight line movements of arbitrary length along each of the four directions. The four possible straight lines have symbols Ln, Ls, Le, Lw, which correspond to lines of arbitrary length along the North, South, East and West directions respectively. A simple grammar for detecting these lines is shown below. Note that each expansion rule has an associated probability denoted by the superscript. These probabilities are currently distributed uniformly, but can be learned in a real system.

$$\begin{aligned} V_N &= \{Start, M, L, Ln, Ls, Le, Lw\} \\ V_T &= \{N, S, E, W\} \\ Start &\rightarrow M^{(1.0)} \\ M &\rightarrow M\,L^{(0.5)} | L^{(0.5)} \\ L &\rightarrow Ln^{(0.25)} | Ls^{(0.25)} | Le^{(0.25)} | Lw^{(0.25)} \\ Ln &\rightarrow Ln\,N^{(0.5)} | N^{(0.5)} \\ Ls &\rightarrow Ls\,S^{(0.5)} | S^{(0.5)} \\ Le &\rightarrow Le\,E^{(0.5)} | E^{(0.5)} \\ Lw &\rightarrow Lw\,W^{(0.5)} | W^{(0.5)} \end{aligned}$$

We can use this grammar to parse a string such as the following: "NNNEESSSSEE", and create a more compact representation from it such as "Ln Le Ls Le", which says that the motion which took place comprised just three straight line motions. Before we go further, let us define a device or process which we call a *Repeater*, which takes any string of symbols, and returns a string with all symbols duplicated except the first and the last symbol. For example, if we pass the string "Ln Le Ls Le" to the repeater, it returns "Ln Le Le Ls Ls Le". Note the repetition of the middle "Le" and "Ls". Having this process is necessary since some symbols are shared by subtrees in higher grammars.

At the next level, we can define a grammar of *single turns*, which takes as input the output of the grammar of straight lines after passing through a Repeater. The output of the grammar of straight lines was "Ln Le Ls", which the Repeater changed to "Ln Le Le Ls Ls Le". This string can be parsed by the *grammar of single turns* shown below.

$$\begin{aligned} V_N &= \{Start, M, T, Lt, Rt, \Pi t\} \\ V_T &= \{Ln, Ls, Le, Lw\} \\ Start &\rightarrow M^{(1.0)} \\ M &\rightarrow M\,T^{(0.5)} | T^{(0.5)} \\ T &\rightarrow Lt^{(0.33)} | Rt^{(0.33)} | \Pi t^{(0.33)} \\ Lt &\rightarrow Ln\,Lw^{(0.25)} | Lw\,Ls^{(0.25)} | Ls\,Le^{(0.25)} | Le\,Ln^{(0.25)} \\ Rt &\rightarrow Ln\,Le^{(0.25)} | Le\,Ls^{(0.25)} | Ls\,Lw^{(0.25)} | Lw\,Ln^{(0.25)} \\ \Pi t &\rightarrow Ln\,Ls^{(0.25)} | Ls\,Ln^{(0.25)} | Le\,Lw^{(0.25)} | Lw\,Le^{(0.25)} \end{aligned}$$

This grammar defines three simple turns, the left turn (Lt), the right turn (Rt), and the in-place 180^0 turn (Πt). Given the string "Ln Le Le Ls Ls Le", this grammar can reduce it to a sequence of three simple turns "Rt Rt Lt".

We can now pass this output through another Repeater to get "Rt Rt Rt Lt". This string is now ready to be parsed by an even more complex grammar of *two turns*. This grammar consists of a *Uturn* which involves two consecutive turns in the same direction, or an *Sturn* which involves consecutive turns in opposite directions.

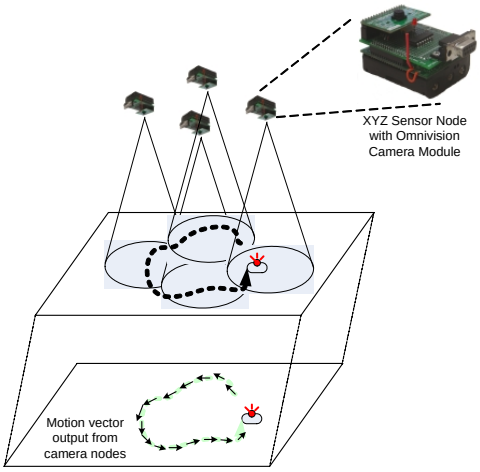

Figure 4: Experimental Setup.

$$
\begin{aligned}
V_N &= \{Start, M, T, U-turn, S-turn\} \\
V_T &= \{Rt, Lt\} \\
Start &\rightarrow M^{(1.0)} \\
M &\rightarrow M\,T^{(0.5)} | T^{(0.5)} \\
T &\rightarrow U-turn^{(0.5)} | S-turn^{(0.5)} \\
U-turn &\rightarrow Rt\,Rt^{(0.5)} | Lt\,Lt^{(0.5)} \\
S-turn &\rightarrow Rt\,Lt^{(0.5)} | Lt\,Rt^{(0.5)}
\end{aligned}
$$

The given string "Rt Rt Rt Lt" can be now condensed to "Uturn Sturn".

In the same manner, we can continue to define more complex behaviors such as a zigzag motion (which consists of alternate U and S turns), or a spiral motion (which is a sequence of U turns). If we have to identify closed curves such as a square shape, we can do so by augmenting or lexicalizing the context-free grammar with step counts, since closure requires us to keep track of the number of transitions of the person in each direction in the grid. Another option is to use a context-sensitive grammar. Figure 3 shows how such a hierarchy of increasingly complex behaviors may be created by starting out with simple behaviors such as straight lines and turns.

4.2 Message Passing

The messages exchanged fall into two different categories: the *internal* and the *external* messages. Internal message passing describes the communication between the different layers of the grammar hierarchy that co-exist on the same node. The output of each level becomes the input for the level right above it. This communication is triggered by the local observations and does *not* involve any radio communication. Conversely, whenever the target exits the field of view of node A and enters the field of view of node B, a message containinng the output of each level running on node A is sent from node A to node B. In our implementation this will result in a packet containing 18 bytes of data.

5. EVALUATION

To validate the proposed framework we used a fraction of our camera-based sensor network testbed. Our test configuration is comprised of a 4×4 grid of XYZ sensor nodes [8] equipped with a low-power camera module from Omnivision (OV7649). The cameras are attached to the ceiling facing into the room as shown in Fig. 4. These nodes can localize in their image plane a mobile node carrying a bright red LED. The mobile node is used to emulate the motion of an individual walking along a corridor. Each node is programmed to capture images at 4 frames per second, and identify the image coordinates of the red LED in the scene with the same frequency. The camera network is pre-calibrated and uses the LED image coordinates at consecutive frames to compute motion vectors for the LED. 3-D Node localization and camera calibration can also be performed online using the implementation we previously described in [7].

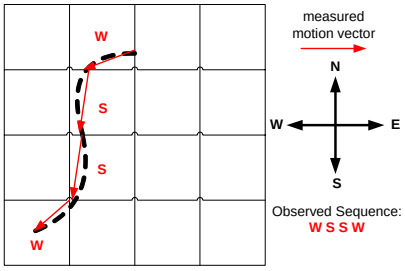

Figure 5: An $S-turn$ pattern inside a 4x4 grid. The recorded motion vectors for each square grid and the final observation sequence are also shown.

The data recorded using our testbed was used to simulate the performance of the proposed hierarchical grammar. We used the PCFG implementation in the Natural Language Toolkit (NLTK, *http://nltk.sourceforge.net*), which incoorporates a Viterbi-style parser for PCFGs. NLTK's PCFG parser is a bottom-up parser that uses dynamic programming to find the single most probable parse for a sequence of observed events (motion vectors in our case). It parses its input by iteratively filling in a most probable constituents table. This table records the most probable tree structure for each span (number of observed events combined together) and node value (terminal or non-terminal symbol). In particular, it has an entry for every start index, end index, and node value, recording the most probable subtree that spans from the start index to the end index, and has the given node value. Once the table has been completely filled in, the parser simply returns the entry for the most probable constituent that spans the entire input, and whose node value is the start symbol.

The output of our simulations are the most probable parse trees generated from the Viterbi parser. In other words, for every sequence of motion vector observations that is given as input to our simulation tool, we report the most probable parse tree for every level of the proposed hierarchical grammar. Each parse tree represents the behavior that was detected at each level based on the input data.

5.1 Experimental Results

In our first experiment each camera-enabled sensor node records the point of entrance and point of exit of the moving target on the image plane, and computes a motion/direction vector using the localization algorithm from [7]. This direction vector is then mapped into one of the four possible phonemes N, S, E, and W. In particular, the phoneme that minimizes the directional difference from the measured di-

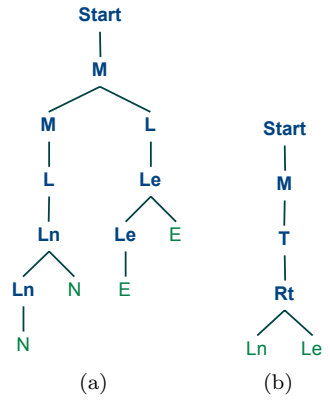

Figure 6: Most probable parse trees for (a) Level 0 and (b) Level 1. Level 2 does not produce any output. The output of the grammar is the correct motion behavior: "Rt".

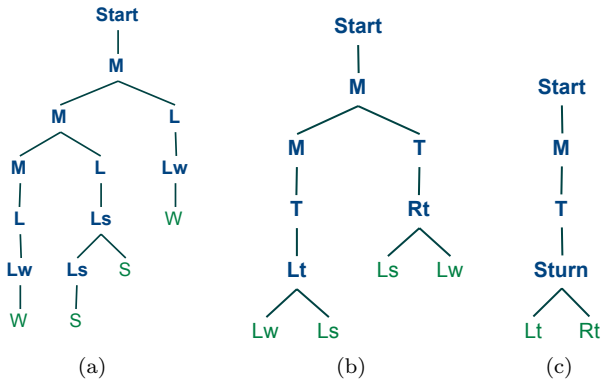

Figure 7: Most probable parse trees for (a) Level 0 (b) Level 1 (c) Level 2. The output of the grammar is the correct motion behavior: "S-turn".

rection vector is chosen. This setup corresponds to a 4x4 grid of camera nodes where each node records only one motion vector. To evaluate the accuracy of the proposed hierarchical grammar, a moving person performed several right turns, S-turns and U-turns covering a large fraction of the monitored area every time. The typical most probable parse trees for each one of these motion behaviors can be seen in Figures 6,7, and 10. In all cases the hierarchical grammar identifies the correct motion behavior.

To better understand how the proposed framework identifies behaviors at different levels let us consider the case of the $S-turn$ pattern in Figure 5. The motion pattern takes place in 4 different squares(camera-enabled sensor nodes) inside the 4×4 grid. Each sensor node observes a phoneme (N, S, E, or W) as it was described earlier. In that way, the $S-turn$ can now be expressed as an ordered sequence of phonemes: "W S S W". This ordered sequence of phonemes is given as input to the first level grammar(Figure 7(a)). The output of this level is a new ordered sequence: "LwLsLw". The output of the first level is fed to a repeater and it becomes: "LwLsLsLw". The second level of the hierarchical grammar(Figure 7(b)) transforms this sequence to: "LtRt". The new sequence is fed to the third level grammar after passing through a repeater. By definition, the repeater will not change the sequence "LtRt". The third level grammar

Figure 8: Most probable parse tree (4×4 grid) for the Level 0. No result is produced for Levels 1 and 2. The output of the grammar is: "Ls".

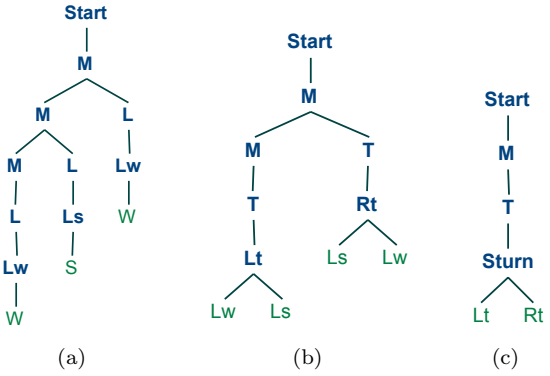

Figure 9: Most probable parse trees (8×8 grid) for (a) Level 0 (b) Level 1 (c) Level 2. The output of the grammar is the correct motion behavior: "S-turn".

(Figure 7(c)) translates this sequence to an $S-turn$ which is the initially performed behavior.

5.2 Sampling Implications

In order to verify how the accuracy of the proposed hierarchical grammar is affected by the grid resolution, we applied the same set of turns, S-turns and U-turns on grids of different sizes: 8×8 and 20×20. We simulated those grid sizes using the 4×4 sensor network grid. Each camera-enabled node segmented its image plane to a 2×2 and 5×5 grid. In that case, each camera records a direction vector for every square grid on its image plane. The results of our experiments were not differentiated from the results shown in Figures 6, 7, and 10. The main difference was in the depth of the parse trees. This shows that the proposed hierarchy of grammars scales well with the grid resolution.

The motion patterns that were used in the previous experiments were covering a large fraction of the monitored area. To push the system to its limits, we also created a new data set where a person was performing small scale S-turns that were covering only a small fraction (approximately 15%) of the monitored area. We gathered data for 3 different configurations: 4×4, 8×8, and 20×20 grids. The results for each configuration can be seen in Figures 8, 9, and 11. It is clear that in the case of the 4×4 grid (Figure 8) the output of the grammar is wrong since the performed behavior is not detected. The main reason is the fact that we are recording only one direction vector per square grid. When the scale of the motion pattern is small and the grid resolu-

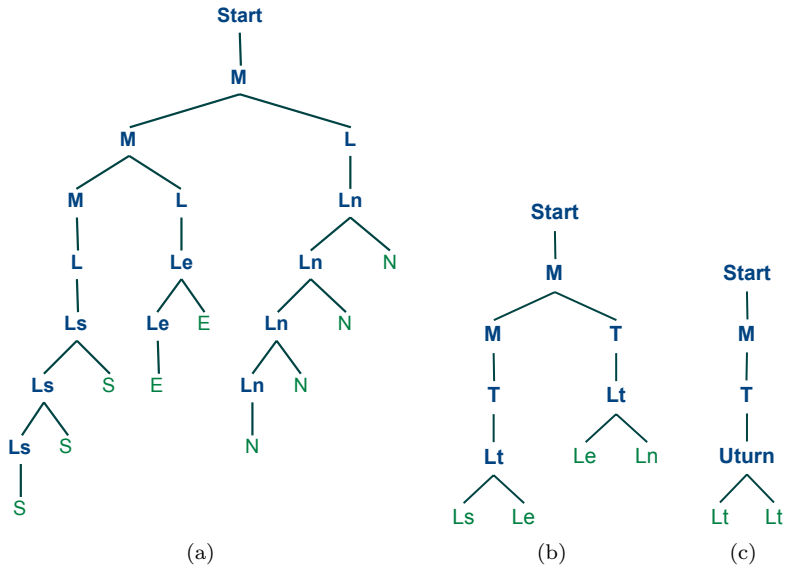

Figure 10: Most probable parse trees for (a) Level 0, (b) Level 1, (c) Level 2. The output of the grammar is the correct motion behavior: "U-turn".

tion is low, using only one direction vector per square grid is not enough because drastic changes in motion behavior might be undetected. For instance, consider the $S-turn$ shown in Figure 5. If a 2×2 grid were used instead of a 4×4 grid, then the ordered sequence of observations would change from: "W S S W" to "S S". While the former sequence of observations provides enough information for the grammar to identify the $S-turn$ the latter does not. This problem could be addressed by segmenting the image plane of a camera to a virtual grid as it was described before. Keeping track of a direction vector per square grid on the image plane allows the camera-enabled sensor node to collect the information required to classify the performed motion pattern. This approach is verified by the results shown in Figure 9. Higher grid resolution allows the collection of more information about the observable motion pattern leading to its correct classification. However, as Figure 11 shows the resolution of the grid cannot be arbitrarily high. In the case of the 20×20 grid the oversampling of the observed motion pattern leads to a wrong classification. These results show that for a given sensor network coverage, there is a minimum and a maximum scale of the behavior that can be identified. Formalizing the correlation between the coverage of the sensor network and the bounds that it sets on the scale of the behavior that can be detected is an interesting research problem that needs to be considered.

6. RELATED WORK

The topic of inferring high-level behaviors from low-level sensors is also studied by Patterson et. al. in [11]. This approach is based on particle filters and it is used to study behaviors of buses on bus routes using GPS sensors. This application bears many similarities to our case study but our approach is significantly different and designed to exploit hierarchical inference with lightweight computation. More examples of inference work applied to assisted living using RFID sensors can be found in [5],[6]. Paskin and Guestrin proposed a very promising inference framework for sensor networks [10]. This work proposes a robust architecture comprised of spanning tree creation, junction tree creation and message passing. Applications of this architecture to inference, regression and optimization has shown that a structured way of reasoning can provide very favorable results. Our framework also tries to provide a structured way of reasoning but focuses more on the hierarchical properties and on specifying a grammatical format to reasoning that can scale vertically to reason about macroscale behaviors.

In our framework similar functionality could be achieved using Hidden Markov Models instead of PCFGs [15]. PCFGs however are more general and more expressive and can be used to describe a large family of HMMs. Using a small set of simple grammar rules we can define families of HMMs, where each family models similar motion behaviors. This representation does not only reduce memory requirements but also makes the reconfiguration of the sensor network easier and more efficient. Instead of changing the definition of a large number of HMMs, in order to detect different type of motion behaviors, we can simply change only a small set of rules on each node. This small set of rules captures exactly the same information as a large number of HMMs.

7. CONCLUSIONS AND FUTURE WORK

We have described a novel framework for interpreting behaviors in sensor networks. Complex behaviors can be expressed in a hierarchy of probabilistic context-free grammars whose vocabulary is a set of events that are extractable from the processing of sensor data at different levels of complexity. We presented a concrete example for a "real sensor network" that the community is building and studying, especially one with strong constraints on communication and computation, and demonstrated that the problem of interpreting a set of behaviors amounts to parsing the temporal evolution of the sensor network data (or rather extracted primitive events). Of course, our initial implementation of the proposed framework is simplified by assuming idealized inputs and considering only a single sensing modality but it clearly demonstrates the capabilities of our framework. This initial exposure also revealed that implementing and main-

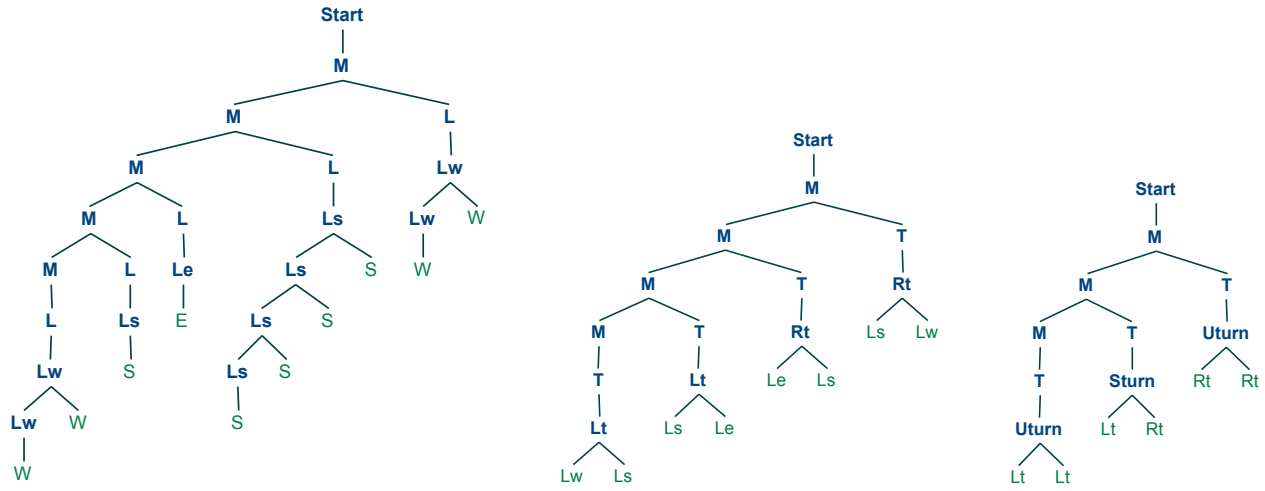

Figure 11: Most probable parse trees (20×20 grid) for (a) Level 0 (b) Level 1 (c) Level 2. The output of the grammar is : "U-turn S-turn U-turn".

taining grammars in a distributed network is an interesting research problem that needs to be explored. The formalism of our framework gives rise to several applications of sensor networks in everyday life situations such as assisted living, security, entertainment and workplace safety. This formalism is very powerful because each level in the hierarchy can be viewed by higher layers as a black box. Each layer infers and summarizes information using lightweight computations and thus effectively reduces the amount of data that needs to be propagated across the network and the network hierarchy. These properties provide a simple, modular, yet very scalable framework that can potentially enable the creation of complex sensor systems.

8. ACKNOWLEDGMENTS

This work was partially funded by the National Science Foundation under award #0448082.

9. REFERENCES

[1] J. Aggarwal and S. Park. Human motion: Modeling and recognition of actions and interactions. *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission*, pages 640–647, 2004.

[2] H. Bunke and T. Caelli. *Hidden Markov Models: Applications in Computer Vision*. World-Scientific, 2001.

[3] G. D. Forney. The viterbi algorithm. In *Proc. IEEE, vol. 61, pp. 268-178*, March 1973.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics Part C*, 34(3):334–352, August 2004.

[5] L. Liao, D. Fox, and H. Kautz. Learning and inferring trasnportation routines. In *Proc. of the Nineteenth National Conference on Artificial Intelligence*, 2004.

[6] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Behavior Recognition in Assisted Cognition. In *Proceedings The AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems*, 2004.

[7] D. Lymberopoulos, A. Barton-Sweeney, and A. Savvides. Sensor localization and camera calibration using low resolution cameras. Technical Report ENALAB 080501, Yale University, 2005.

[8] D. Lymberopoulos and A. Savvides. Xyz: A motion-enabled, power aware sensor node platform for distributed sensor network applications. In *IPSN, SPOTS track*, April 2005.

[9] A. Ogale, A. Karapurkar, and Y. Aloimonos. View-invariant modeling and recognition of human actions using grammars. *Workshop on Dynamical Vision at ICCV'05*, October 2005.

[10] M. A. Paskin, C. E. Guestrin, and J. McFadden. A robust architecture for inference in sensor networks. In *Proc. of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN05)*, April 2005.

[11] D. J. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring High-Level Behavior from Low-Level Sensors. In A. Dey, A. Schmidt, and J. F. McCarthy, editors, *Proceedings of UBICOMP 2003: The Fifth International Conference on Ubiquitous Computing*, volume LNCS 2864, pages 73–89. Springer-Verlag, 2003.

[12] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. IEEE, Vol.77, No. 2, pp. 257-286*, February 1989.

[13] M. Valera and S. Velastin. Intelligent distributed surveillance systems: a review. *IEE Proceedings on Vision, Image and Signal Processing*, 152(2):192–204, April 2005.

[14] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern recognition*, 36:585–601, 2003.

[15] C. S. Wetherell. Probabilistic languages: A review and some open questions. *ACM Comput. Surv.*, 12(4), 1980