Recognizing Activities from Context and Arm Pose using Finite State Machines

Thiago Teixeira, Deokwoo Jung, Gershon Dublon and Andreas Savvides
Yale University, New Haven, CT 06511 — e-mail: firstname.lastname@yale.edu

***Abstract*—We present an activity-recognition system for assisted living applications and smart homes. While existing systems tend to rely on expensive computation of comparatively large-dimension data sets, ours leverages information from a small number of fundamentally different sensor measurements that provide *context* information pertaining the person's location, and *action* information by observing the motion of the body and arms. Camera nodes are placed on the ceiling to track people in the environment, and place them in the context of a building map where areas and objects of interest are pre-marked. Additionally, a single inertial sensor node is placed on the subject's arm to infer *arm pose*, *heading* and *motion frequency* using an accelerometer, gyroscope and magnetometer. These four measurements are parsed using a lightweight hierarchy of finite state machines, yielding recognition rates with high precision and recall values (0.92 and 0.93, respectively).**

I. Introduction

Recognizing human activities around the house in a robust and practical way would be an asset for several home monitoring applications. The main difficulty in recognizing activities is that no matter the types or number of sensors used, the measurement space is invariably overwhelmed by the immensity of the state space of human activities. Starting from a simplistic definition, activities can be described as sequences of "actions" in time. These "actions" are atoms of motion which include the person's pose and small time-scale motion primitives. Considering, however, that the human body has at least 244 degrees of freedom [27], it is clear that acquiring and processing this information to produce high-fidelity poses is intractable, due not only to computational constraints of a real-time embedded system but also to the sheer number of sensors required to measure it. But that is only part of the problem. It is also important to consider contextual information such as setting (where does the action take place?), date/time, cultural factors, and so on. What is more, the same underlying action may be performed in many different manners. Hence, taken in such a broad sense, activity recognition is a substantial problem.

In this paper, we describe a prototype sensor network that uses a person's location along with minimal pose and motion information to detect activities in a house, provided the locations of objects and furniture are known. By using an object-oriented hierarchy of Finite State Machines (FSMs) augmented with time constraints we are able to detect common household activities such as '*cooking*', '*eating*', '*brushing teeth*', and '*fetching a glass of water*' in uncontrolled environments and with little computational cost. Where existing systems tend to rely on expensive computation of comparatively large-dimension data sets, ours leverages information from a small number of fundamentally different sensor measurements. From our experience with real-world camera sensor network deployments [2], we have found that the largest obstacle to activity recognition is the widespread presence of noise, both due to measurement errors (leading to false-positives) and to the interspersing of activities when multitasking or changing plans. For example, when a person starts cooking, then leaves to answer the phone, goes back to cooking, goes to watch television, then finishes cooking and eats at the table. This effect was visible in our previous work [10][11], in which sequences of location symbols were interpreted using probabilistic context-free grammars (PCFGs) to infer higher-level behaviors. Although attractive due to their expressive power and strong theoretical grounding, PCFGs have limited noise tolerance without extensive training.

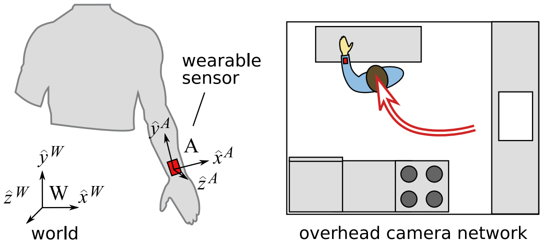

Fig. 1. Overview of our system and the associated coordinate systems.

The first contribution of this paper is a system that uses the person's location along with a minimal number of pose and motion measurements to detect human activities in uncontrolled, real-world scenarios. For this, context information from a pre-annotated map is used to fill in the blanks from the low-accuracy sensing. Observations of pose and motion are abstracted into four components: body speed, arm tilt, arm heading, and arm motion frequency. These are inferred with an infrastructure of non-overlapping camera nodes along with a single wrist-mounted inertial sensor node per person. The second contribution of this paper is the development of an object-oriented hierarchy of finite state machines that we extend with time constraints in order to parse the noisy stream of input measurements, with good demonstrated detection rates. In this work we consider solely the single-person case, given that multiple-person data can be separated using our ongoing research in person-identification [20][21].

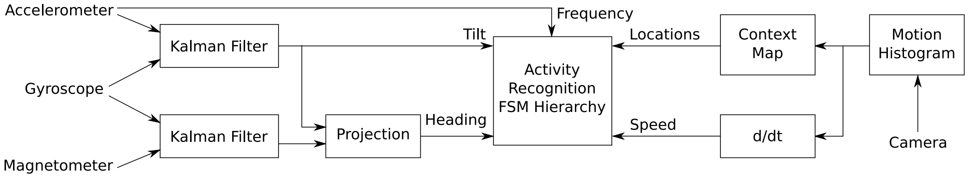

Fig. 2. Block diagram of the entire system. Measurements from the wearable sensor node are displayed on the left, while the centroids detected by the camera sensor nodes are on the right.

II. Background and Related Work

Traditionally, activity recognition has been almost exclusively in the domain of computer vision. In that field, the most common strategy is to detect poses as precisely as possible for each video frame. This is sometimes done in a top-down manner, with template-based approaches that employ holistic image features [13], [7] or build spatiotemporal shapes from silhouettes [4], [24]. Other times, a bottom-up approach is used, detecting body parts to fit into a prior model of the human body [14], [18]. Regardless of the approach, camera-only solutions to activity recognition must take into account occlusions and the view-variance of poses. Even when multiple cameras or 3D constraints are used to mitigate these factors, fundamental lower-level problems make most approaches infeasible for uncontrolled real-world scenarios. For example, the background-subtraction algorithms that are often required in preprocessing stages usually fail or adapt too slowly when a person moves an object in the scene (such as a chair). For these reasons, in this work we avoid relying on high-level inferences from cameras, using them only to detect the subject's position and speed.

In recent years, however, the research effort in human activity recognition has spread to a diverse number of fields. Approaches generally break down into the extraction of meaningful statistics from sensors, and the inference of high-level activity information from the statistics. Various sensing modalities and inference algorithms have been applied. Sensor types range from location information from WLAN RF-signals [26], to motion detectors, break-beam sensors, pressure mats, and contact switches [23]. These usually extract only a rudimentary amount of activity information due to the limited size of the measurement space. Higher-level activity recognition schemes have also been proposed, employing multiple sensors such as accelerometers, gyroscopes, and magnetometers [12], [25], [3]. In those papers, various inference techniques are adopted such as linear classifiers [12] and support vector machines [25]. A review of these methods can be found in [17]. However, systems that rely on multiple wearable sensors can be cumbersome for the user, which is why we limit ourselves to a single wearable node per person, placed on the arm. In the future, our sensor node can be placed on a bracelet or wrist-watch form factor.

Recently, RFID-based activity recognition has also been proposed using Hidden Markov Models (HMMs) [15] [16] [5] [26]. It is not hard to see why the HMM formulation is such a popular choice in activity recognition: if activities are represented as sequences of states, then the activity recognition problem can be described as that of finding the most probable state given a sequence of observations. Yet the Markov assumption (that only the last N states are required to predict the next state) does not apply to human activities when multitasking is a factor. That is, if a person interrupts an activity to do a number of other activities, and then returns to finish the interrupted activity, then the number of states becomes unpredictable. What is more, although HMMs can attain good noise-resilience through a learning procedure, the learned model parameters are likely to be specific to that person and/or house. For example, if the measurements are composed solely of the person's location in the house, the likelihood of observing a '*next to laundry machine*' location while the person cooking is approximately zero except for homes where the laundry machine is found in the kitchen. Finally, in order to introduce time and context to HMM or PCFG-based approaches, the state space must be augmented to encompass the Cartesian product of all three states spaces (the original state space, plus time and context state spaces), which greatly increases the complexity of a solution that was already computationally-heavy to begin with. In this paper, we opt for a finite state machine approach similar to [19], [1]. State machines are lightweight, human-readable and easy to parse. However, as further detailed in Section VI-A, they quickly fail in the presence of noise, leading to false-positive detections. In this paper we describe a simple extension to FSMs that utilizes intuitive time constraints to mitigate this problem, with good results. As we will show, adding time constraints to FSMs is natural and straight-forward, and leads to noise-resilient activity parsing.

With few exceptions, the common theme with the existing approaches is that activity recognition is attempted solely based on pose or motion information — but not context. What sets our approach apart is that we bypass the admittedly challenging problem of extracting detailed pose information from a scene. Instead we extract only lightweight pose and motion properties and make use of context to counter-balance the lack of sensing detail.

III. System Overview

Our system consists of ceiling-mounted cameras nodes employed alongside a single wearable sensor-node. The cameras detect and track the person, acquiring information about speed and position relative to a map. Meanwhile, the wrist-mounted inertial sensor node provides the angle of the arm with respect to gravity (tilt) and to the local magnetic field (heading). The overall setting is illustrated in Figure 1. Although the inertial node is placed only on the person's most dexterous arm, more information can be gathered with the addition of a similar

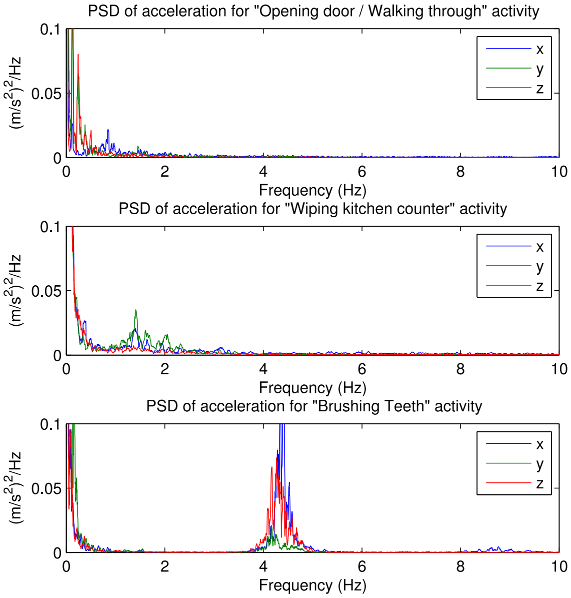

Fig. 3. The power spectral density of the wrist acceleration can be used to assist the differentiation between activities.

sensor node on the other arm. We have found, however, that a single arm contains enough information to detect a set of activities including '*eating*', '*cooking*', '*cleaning*' and more, as described in our evaluation (Section VII).

In this work, we define *activities* as sequences of actions, and *actions* as pose transitions of short duration taking place within some context. For this, at each time instant we combine the inertial and camera data to generate an observation vector whose components contain both pose/motion information and location context. The observation vector is, then, used as input for a hierarchy of state machines representing higher level actions and activities.

IV. Measurement Space

For each time step, we extract an observation vector of the following form:

$$\boldsymbol{z} = [\underbrace{L, V}_{body}, \underbrace{P, D, F}_{arm}] \tag{1}$$

where L and V are the location and speed of the person, measured from the cameras, and P, D and F are the arm pose, arm direction and arm motion frequency. All of these are described nominally rather than numerically. That is:

$$\begin{aligned} V &\in \mathcal{V} = \{\text{`}moving\text{'}, \text{`}stopped\text{'}\} \\ P &\in \mathcal{P} = \{\text{`}up\text{'}, \text{`}high\text{'}, \text{`}middle\text{'}, \text{`}low\text{'}, \text{`}down\text{'}\} \\ D &\in \mathcal{D} = \{\text{`}North\text{'}, \text{`}East\text{'}, \text{`}South\text{'}, \text{`}West\text{'}\} \\ F &\in \mathcal{F} = \{\text{`}high\text{-}freq\text{'}, \text{`}low\text{-}freq\text{'}, \text{`}stopped\text{'}\} \end{aligned}$$

Furthermore, the location L is given not in terms of points in a coordinate system, but as high-level descriptions: '*at the stove*', '*at the kitchen sink*', and so on. Thus the components of the observation vector $\boldsymbol{z}$ carry information regarding context (the high-level location L), pose (P and D) and motion (V and F), albeit with low resolution.

The rationale for such a observation vector is that a large set of actions can be performed with similar poses or motions, but they differ in their *focus*. With eyes located on their front side, people usually turn their bodies to face the object of their attention, and due to their limited range of actuation, they place themselves close to it (within arms length). Similarly, since a large set of human actions involves handling objects or manipulating devices, the location of the hands contains invaluable information about the object onto which an action is focused. In our system, the focus inferred from hand locations is fused with prior knowledge of the location of furniture and household utilities, which we refer to as context.

However, while it is relatively simple to obtain L (we simply overlay the person's coordinates onto a pre-annotated building map), robustly detecting the location of the hands is a challenging task. Instead, we obtain only an indirect measure of the focus of activity through the vertical and horizontal angles of the forearm. These angles are referred to as tilt θ and heading ϕ, and are acquired using the accelerometer and magnetometer on the person's wrist, as described in Sections V-A and V-B. Thus, two activities that can occur in the same area, such as accessing the medicine cabinet and using the counter below it, can be differentiated given the person's arm tilt. The heading, meanwhile, can be used to filter meaningful area visits. For example, although a person may pass in front of the fridge quite often, a meaningful visit would require the person to extend their arm toward it.

Other than measurements of "focus of activity" (location and arm pose), heuristic measures of "level of activity" are used in the absence of more precise low-level motion information: the body speed V and the arm motion frequency F. As seen in Figure 3, the frequency components of different actions can vary drastically. In some cases, such as '*brushing teeth*' versus '*shaving*', the arm frequency can be the main disambiguation factor.

Note that the measurements in $\boldsymbol{z}$ come from sensors with quite different sampling frequencies. Rather than interpolating the data into a common frequency, we update each component of the observation vector asynchronously as soon as new data is available. When this happens, an event notification is generated by the system, and propagated through the activity recognition FSM hierarchy. Meanwhile the components with no new data retain their previous values. This event-driven scheme lowers the data rate to a bare minimum.

V. Detecting Arm Pose

For completeness, in this section we briefly describe the process of calculating the arm tilt θ and heading ϕ, from which the nominal values of pose P and direction D are extracted by quantizing.

A. Tilt Detection

The typical MEMS accelerometer measures acceleration through the displacement of a proof mass with respect to frame that houses it. When a force is exerted on the housing, the proof mass is displaced in the opposite direction. Due to

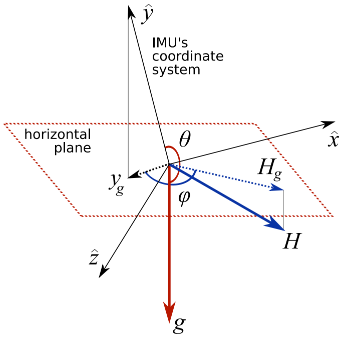

Fig. 4. The orientation of the accelerometer with respect to the world coordinate system is given by two angles: the tilt, which is the angle θ between the y axis and the gravity vector $\boldsymbol{g}$; and the heading, which is the angle ϕ between the projection of the y axis onto the horizontal plane and that of the magnetic force vector $\boldsymbol{H}$.

this, the output of the accelerometer contains an additional acceleration component as an effect of gravity on the proof mass — or, more precisely, from the normal force that is a reaction to gravity. This additional component is present any time the accelerometer is not in free fall. The following equation describes the output of the accelerometer mathematically:

$$\boldsymbol{a}_{meas} = \boldsymbol{g} + \boldsymbol{a}_{motion} + \boldsymbol{e}_{meas} \tag{2}$$

where $\boldsymbol{a}_{meas}$ is the measured acceleration, $\boldsymbol{g}$ is the component due to gravity, $\boldsymbol{a}_{motion}$ is the acceleration of the accelerometer in space, and $\boldsymbol{e}_{meas}$ is a combination of accelerometer bias, quantization errors, and zero-mean Gaussian noise. The $\boldsymbol{g}$ vector, if correctly extracted from $\boldsymbol{a}_{meas}$, can be used to calculate the tilt θ of the accelerometer using basic geometry (Figure 4):

$$\theta = \arccos\left(\hat{\boldsymbol{y}} \cdot \boldsymbol{g} \,/\, |\boldsymbol{g}|\right) \tag{3}$$

where the accelerometer is oriented so that unit the vector $\hat{\boldsymbol{y}}$ is a unit vector that points toward the arm's positive y direction (Figure 1).

To find the tilt using equation (3) one must have a good estimate of the gravity vector $\boldsymbol{g}$. It is clear from equation 2 that when the person's arm undergoing any motion the unknown component $\boldsymbol{a}_{motion}$ becomes an obstacle toward the computation of $\boldsymbol{g}$ from the accelerometer reading $\boldsymbol{a}$. However, making use of the fact that $\boldsymbol{g}$ has fixed magnitude (except in a free-fall scenario, which is out of the scope of this paper) and that its orientation is a function of the rotation of the accelerometer, equation (4) can be derived by incorporating the rotation from gyroscope measurements:

$$\dot{\boldsymbol{g}} = \begin{bmatrix} 0 & -\omega_z & +\omega_y \\ +\omega_z & 0 & -\omega_x \\ -\omega_y & +\omega_x & 0 \end{bmatrix} \boldsymbol{g} \tag{4}$$

where ω_x, ω_y, ω_z are the angular velocities around the x, y, and z axes of the target, measured by a gyroscope. The fusion of accelerometer and gyroscope measurements to estimate orientation is desirable because the errors in these two sensors are independent. We use a Kalman filter based on equation (4) to estimate $\boldsymbol{g}$, as is common practice in the literature [28][6].

Although greatly minimized through the Kalman filter, the estimated $\boldsymbol{g}$ still contains a strong residual component that is correlated with $\boldsymbol{a}_{motion}$. This is the largest source of error in the tilt computation, causing the estimate to deviate upon large external accelerations. This results in the extraction of false-positive arm poses P (which are quantized from θ) whenever the person moves their arm abruptly. In Section VI-A we describe a method of mitigating the effect of such false positives.

B. Heading Detection

In order to obtain the direction measurement D, we first compute the heading angle ϕ of the inertial node using a 3-axis measurement $\boldsymbol{H}$ of the local magnetic field supplied by an integrated magnetometer. Since the absolute heading with respect to true North is not of interest in our application, in this section the word "North" is used to indicate the local direction of the magnetic field. The heading computation in this paper assumes the local magnetic field is approximately constant over time and space. From our experience, this assumption should hold in typical assisted-living scenarios.

Except when measured directly on the equator, $\boldsymbol{H}$ tends to tilt vertically, pointing down into the ground in the northern hemisphere and up in the southern hemisphere. As such, to find the local North a tilt-compensation method must be used. We accomplish this by projecting $\boldsymbol{H}$ onto the plane tangent to the Earth's surface at the IMU's location, which we call the ground plane. The calculation is done in the following manner:

$$\boldsymbol{H}_g = \boldsymbol{H} - (\boldsymbol{H} \cdot \hat{\boldsymbol{g}})\hat{\boldsymbol{g}} \tag{5}$$

where $\hat{\boldsymbol{g}} = \boldsymbol{g}/|\boldsymbol{g}|$. The heading angle is, then, the angle between $\boldsymbol{H}_g$ and the projection of the $\hat{\boldsymbol{y}}$ unit vector onto the ground plane, given that the y axis is parallel to the arm (Figure 1). We project $\hat{\boldsymbol{y}}$ onto the horizontal plane in the same manner as (5) to obtain $\boldsymbol{y}_g$. From here, the angle can be found in the customary manner using the dot product:

$$\phi' = \arccos \boldsymbol{H}_g \cdot \boldsymbol{y}_g \tag{6}$$

and the clockwise heading angle ϕ is given by verifying whether $\boldsymbol{y}_g \times \boldsymbol{H}_g$ points in against the direction of $\boldsymbol{g}$, in which case ϕ is $2\pi - \phi'$:

$$\phi = \begin{cases} 2\pi - \phi' & \text{if } \hat{\boldsymbol{g}} \cdot (\boldsymbol{y}_g \times \boldsymbol{H}_g) < 0 \\ \phi' & \text{otherwise} \end{cases} \tag{7}$$

Note that when $\boldsymbol{H} \parallel \boldsymbol{g}$, the projection onto the ground plane yields $\boldsymbol{H}_g = \boldsymbol{0}$, and the heading is undefined. This is not a problem since in these cases the heading angle loses any physical meaning, as the arm must be pointing straight up or down. We have found that the projection-based method presented here produces better results than the rotation-based method found the in the literature [6].

From the equations above, it is clear that the heading estimate contains two sources of noise: one from magnetometer measurements, and the other from the estimation of the gravity vector $\boldsymbol{g}$. We have found that the latter is the most severe, causing the heading estimation to degrade in periods of strong

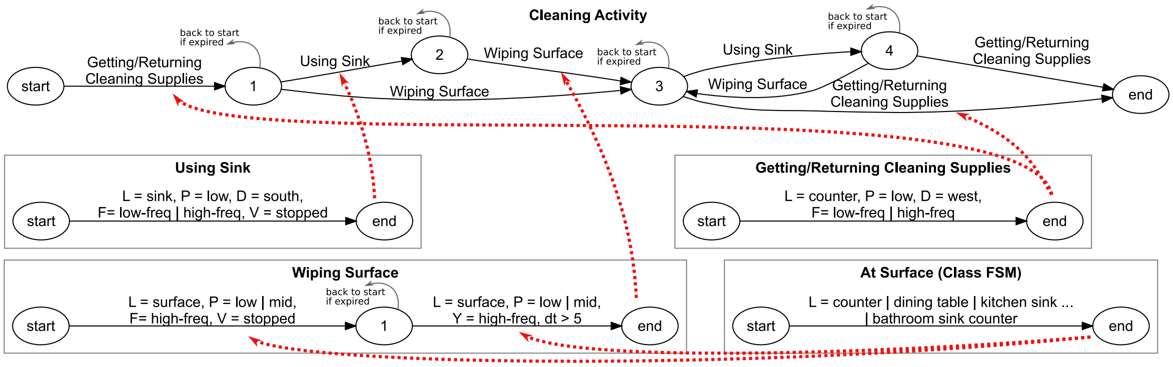
Cleaning Activity
start
Getting/Returning Cleaning Supplies
back to start if expired
1
Using Sink
2
Wiping Surface
3
Using Sink
4
Wiping Surface
Getting/Returning Cleaning Supplies
Getting/Returning Cleaning Supplies
end
Using Sink
start
L = sink, P = low, D = south,
F= low-freq | high-freq, V = stopped
end
Getting/Returning Cleaning Supplies
start
L = counter, P = low, D = west,
F= low-freq | high-freq
end
Wiping Surface
start
L = surface, P = low | mid,
F= high-freq, V = stopped
1
L = surface, P = low | mid,
Y = high-freq, dt > 5
end
At Surface (Class FSM)
start
L = counter | dining table | kitchen sink ...
| bathroom sink counter
end

Fig. 5. Hierarchy of finite state machines that pertain to Cleaning Activity detection. As shown in equation (8), each non-terminal state also has an implied expiry transition back to the start state.

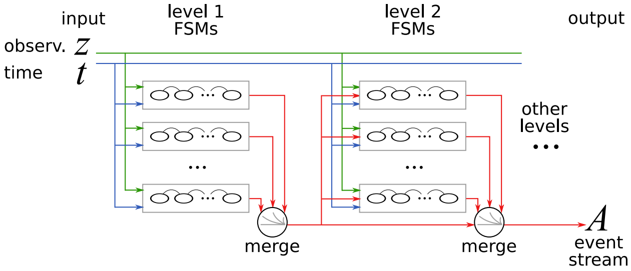

Fig. 6. We detect activities using a hierarchy of finite state machines. The input of each stage is the measurement vector $\boldsymbol{z}$, the current time t, and the outputs of the previous stages, the event stream A.

acceleration. However, since many arm actions culminate in a period of low-magnitude motion, arm heading still proves to contain useful information. Again, the false-positive directions D detected by quantizing ϕ are handled through timing constraints as described in Section VI-A.

As with the tilt-detection case, we use a Kalman filter to smoothen the input signal before calculating the heading. In reality, a single filter is used to estimate both the tilt and the heading. For space considerations, we leave the Kalman filter derivation out of this paper, as it can be readily found in the literature [28].

VI. Activity Recognition

We detect actions and activities (sequences of actions) using a set of object-oriented finite state machines that run in parallel with one another. The lowest-level inputs to this hierarchy are the observation vectors described in the previous section. Whenever an FSM runs to completion (reaching the state S_{end}), an output symbol is produced and inserted into an output stream A, indicating that the corresponding activity has been detected, and the state machine is reset back to the start state S_{start}. The output detections can, then, can be used by higher-level FSMs to detect more complex activities. This can be seen in Figure 5, where the transitions of the '*Cleaning Activity*' FSM depends solely on the outputs of low-level state machines. Each FSM starts at the state S_{start} and moves from state to state according to the transition function $\alpha(S_i, \delta t_i, \boldsymbol{z}, A)$ where the parameters are: the current state S_i, the time spent in the current state so far δt_i, the observation vector z, and the set A containing the activity outputs of the lower layers of the hierarchy for the current time step. This is shown in Figure 6. The hierarchy of FSMs runs in an event-driven manner that propagates through the different levels: whenever the input observation vector $\boldsymbol{z}$ changes, the 1^{st}-level FSMs are parsed, their outputs (if any) are forwarded to the 2^{nd}-level FSMS, which are then parsed. The outputs of the 1^{st} and 2^{nd} levels are then input into the 3^{rd} level, and so on.

A. Relevance period

In our experience, the stream of observation vectors of a person in the course of an activity contains large amounts of false-positive observations. As a consequence, it is common for false-positives to randomly trigger transitions that incorrectly alter the states of one or more FSM. Given enough time, there is a high probability that some FSM completely unrelated to the person's activity will run to completion, generating a false detection.

We guard against this by introducing the concept of *relevance period*. The relevance period (T_i) of a state S_i is a user-defined time period inside of which any action related to that state must take place. Otherwise, after T_i seconds have gone by all future actions are deemed to be unrelated to the current state and the state machine is reset to S_{start}. This decreases the probability that noisy observations trigger unrelated transitions and lead to false positive detections. In Figure 5, we show the relevance period transitions as small gray arrows in each non-terminal state. In our formulation, a parameter T_i is required for *every* state S_i, except S_{start} and S_{end} (terminal states). The terminal states S_{start} and S_{end} are defined as having a δt equal to ∞ and 0, respectively. Thus, for each transition function α declared by the state machine developer, the system automatically replaces it with a transition function α' that creates expiry transitions conditioned upon δt_i:

$$\alpha'(S_i, \delta t_i, \boldsymbol{z}, A) = \begin{cases} S_{start} & \text{if } \delta t_i > T_i \\ \alpha(S_i, \delta t_i, \boldsymbol{z}, A) & \text{otherwise} \end{cases} \quad (8)$$

B. Hierarchies and Object-Orientation

Figure 7 shows a state machine that is able to reliably detect '*cooking*' activity. This FSM contains a single non-terminal state whose transitions serve solely to renew the state's expiration timer δt_i. The simplicity of the '*cooking*' FSM comes from its dependence on lower-level FSMs to detect '*fetching food*', '*heating up*', and '*handling object over*

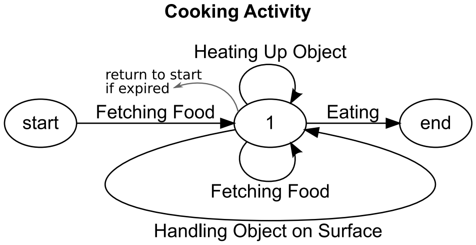

Fig. 7. A simple state machine to detect cooking, used mainly with the time constraints to assure the actions take place within sensible time intervals of one another. In our experiments, every instance of cooking activity was correctly detected.

surface'. The formulation as a hierarchy promotes a divide-and-conquer approach, and allows for the organization of map locations into classes.

Since similar actions can belong to different contexts (a person can perform the '*open*' action on a door as well as a drawer), the predefined areas can be abstracted into classes (e.g. '*openable*') to produce more general activity definitions. The class '*openable*' may encompass all doors, drawers, closets, fridge, and so on. Instead of defining an '*open*' activity for each of those areas individually, a single activity model is utilized which applies to everything that is '*openable*'. Moreover, higher specificity can be achieved using subclasses: the class '*container*' may be used for objects such as closets and fridge, while only the latter may belong in the subclass '*food container*'. This way, the '*fetching food*' activity used by the '*cooking*' FSM (Figure 7) can be specified as a trip to a '*food container*'. This reduces the complexity of the recognition layer while also making the activity models more generic. If the same system is used on a different home, one with an extra pantry, for example, the '*cooking*' activity should continue to be recognized, so long as the extra pantry is defined as belonging to the '*food container*' class. These classes of FSMs are themselves implemented using state machines, and can be used to group locations and/or lower-level FSMs.

Classes are implemented by creating a trivial state machine containing only the initial and final states, S_{start} and S_{end}. Then a transition to the end state S_{end} is conditioned upon the presence of any element from some set of locations or activities. Hence, the transition function α from state S_{start} takes the following form:

$$\alpha(S_{start}, \delta t_{start}, \boldsymbol{z}, A) = \begin{cases} S_{end} & \text{if } L \in C \\ S_{end} & \text{if } \exists\, a \in A \text{ s.t. } a \in C \\ S_{start} & otherwise \end{cases} \tag{9}$$

where $C = \{c_1, c_2, ...\}$ where c_i is either a location (L) or a lower-level activity (some $a \in A$).

An example of a class-definition FSM can be seen in the bottom-right of Figure 5. That FSM defines the locations '*kitchen counter*', '*dining table*', '*kitchen sink*' and '*bathroom counter*' as belonging to the class '*surface*'.

VII. Evaluation

We recorded 40 traces where a person performed one of eight activities in their home: cooking, eating, brushing teeth, using toilet (up), using toilet (down), cleaning, fetching water, taking out trash. The traces were acquired over the course of several days. Since the experiments were not scripted, there were large variations between instances of the same activity. This is especially true for the cooking traces, where meals varied from quick sandwiches to long roasts spanning around 2 hours. Additionally, some traces captured more than one activity, such as '*using toilet*' followed by '*brushing teeth*', and no manual or automatic segmentation of any sort was performed.

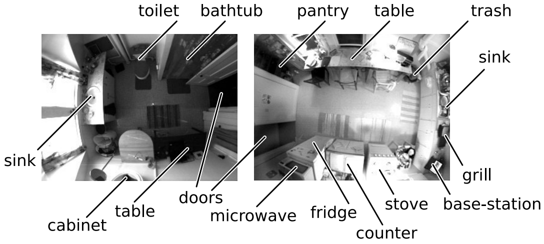

Fig. 8. Images taken with deployed camera sensor nodes with some of the labeled areas shown. Left: bathroom. Right: kitchen.

For these experiments, a small camera network was installed, with one camera node in the kitchen, one in the bathroom, connected wirelessly to a base node attached to a laptop (Figure 8). The camera nodes consist of Intel iMote2 nodes equipped with custom camera sensor boards. They were attached to the ceiling, facing down, so that the entire room could be seen in the camera's FOV (using a $162\deg$ wide-angle lens). The iMote2's PXA271 processor was set to operate at $208MHz$, allowing it to perform real-time, online image processing and human detection on the nodes at frame rate of 14Hz. The data was transmitted over the wireless channel and recorded at the laptop.

The camera nodes were programmed with a detection algorithm described in our previous work [22]. The algorithm detected people using two key properties: motion and size. First, motion is segmented by differencing consecutive frames. Since the ceiling height is known, the approximate area occupied by a person in the image plane can be estimated. We construct a histogram that divides the image into overlapping person-sized blocks. The value of each histogram bin is set to the number of foreground pixels that fall within its corresponding block. The coordinates of the modes of the histogram represent the detected position of each person. This system is designed from the ground up to operate in uncontrolled indoor scenarios. It is robust against the most common types of false detections, such as ghost-detections that can appear when objects are moved. The trade-off is that the detected coordinates have a resolution of approximately $15cm$. Since our activity recognition system is only interested in contextual information (as opposed to precise localization), we find that this level of accuracy is adequate.

A SparkFun 6DoF inertial measurement unit (IMU) was attached to the subject's wrist, transmitting measurements through Bluetooth at a 100Hz sampling rate. While a simple beacon-based time-synchronization scheme was used for the camera nodes, the IMU's measurements were simply time-

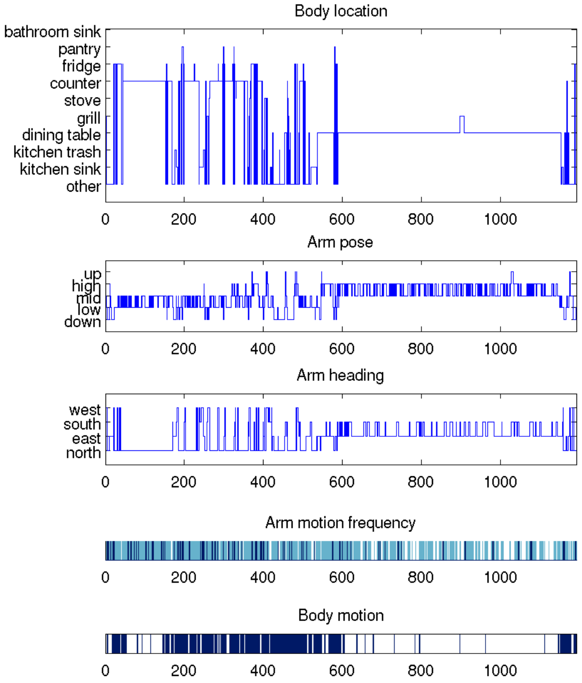

Fig. 9. A collected experimental trace ($\boldsymbol{z}$) for an instance of the '*cooking*' activity. In the arm frequency plot, the color white and two shades of blue are used to portray the ternary values. The time axis is given in units of *seconds*.

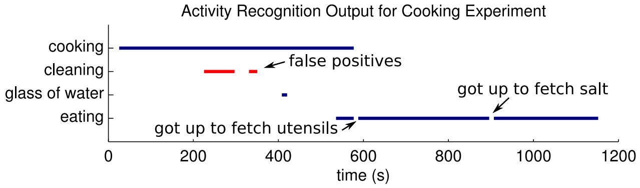

Fig. 10. Output of activity recognition hierarchy when the input is the trace shown in Figure 9. All activities are correctly detected, except for a false-positive cleaning detection. We have found that a higher-level FSM can be used to correct these false-positives.

stamped upon receipt. Since the size of the SparkFun 6DoF proved to be bit cumbersome to wear on the wrist we are currently working on a more portable architecture using TI EZ430-2480 Zigbee-capable wireless nodes.

Clearly, the performance of this system depends heavily on the actual FSMs used for activity detection. Here, we briefly describe some of them to give a general idea regarding their form. The '*brushing teeth*' activity was defined as picking up the toothbrush and toothpaste from the medicine cabinet (arm 'high', in front of cabinet, facing it), using the sink, brushing ('arm up' or 'high' pose, high-frequency motion), using the sink again, and placing the toothbrush back. '*Fetching water*' activity was modeled as picking up a glass ('arm high') from the cupboard, carrying it to the sink ('middle' or 'low' arm poses while moving) and filling it with water ('sink' location, 'south' heading, 'low' hand). The '*eating*' activity was defined as sitting at the dining table while alternating between the 'middle' and 'high' arm poses. '*Cooking*' and '*cleaning*' were defined as shown in Figures 7 and 5.

Activity	Precision	Recall
Cooking	1.00	1.00
Eating	1.00	1.00
Brushing teeth	1.00	1.00
Using toilet (up)	0.71	1.00
Using toilet (down)	1.00	1.00
Cleaning (uncorrected)	0.42*	0.60*
Cleaning	1.00	0.60
Fetching water	0.83	1.00
Taking out trash	0.83	0.83
Total	0.92	0.93
(*excluded from total)		

TABLE I
EXPERIMENTAL RESULTS

We acquired 40 total data traces where the subject performed each activity from several times over a number of days. Figure 9 shows an example of a trace collected by the system, where the person prepared, grilled and ate a sandwich. Sometimes, activities were performed in the midst of others (multi-tasking): for example, in the trace shown in Figure 9, the subject fetched a glass of water while cooking. This was correctly detected by the system. The subject was told to, at times, wander around aimlessly, in order to trigger multiple irrelevant area detections. In our experience, this is the largest source of noise in our long-term deployments, such as [2]. This is where the heading measurements proved most useful, filtering out many of the area detections where the person was facing away. The classification results are shown in Table I, in terms of *precision* and *recall*. *Precision* is a measure of the exactness of classification, obtained as $Precision = TP/(TP + FP)$, where TP and FP are the number of true positives and false positives, respectively. Meanwhile, *recall* measures the completeness of the results, and is calculated by $Recall = TP/(TP + FN)$, where FN is the number of false negatives. Both quantities range from 0 to 1, where 1 is best.

The experimental traces yielded very strong results for '*cooking*', '*eating*', '*brushing teeth*' and '*using toilet (down)*'. The total scores, considering all activities, were found to be 0.81 and 0.93 for precision and recall, respectively. The poor recall score for '*cleaning*' activity is due to the sensors' inability to robustly detect when the subject has picked up or returned the cleaning supplies. In our experiments we defined '*getting/returning cleaning supplies*' as standing by the kitchen counter (under which reside the cleaning supplies) with a '*low*' or '*down*' arm pose. This occurred much too often, especially during cooking. This is portrayed in Figure 9, where '*cleaning*' false positives can be seen. We found that all '*cleaning*' activity false positives were quite obvious when plotted (they all occurred during cooking), and designed a higher-level FSM to automatically detect and correct them. Using this FSM we were able to obtain a much higher precision score, as shown in table I. This raised the total precision to 0.92. Similarly, the poor precision score of '*using toilet (up)*' comes mostly from false positives that occur during actual '*using toilet (down)*' activities, and could also be handled in a higher-level FSM.

VIII. CONCLUSION AND FUTURE WORK

We have described a system that is capable of detecting a set of every-day activities in a house with a minimal number of sensor nodes. Inertial sensors were used to provide limited information regarding the subject's arm pose and arm motion frequency, while overhead cameras tracked the person in the context of a map. We motivated the choice of sensing modalities for this minimal system and demonstrated with 40 experimental traces an overall precision of 0.92 and 0.93 recall. What is more, complex activities such as '*cooking*', '*eating*' and '*brushing teeth*' were correctly classified 100% of the time. Given the promising results from this prototype phase, we are now working on a more tightly integrated system, using a smaller and lower-power wearable node.

In the future, we plan on addressing the weaknesses of our system. Currently, the state machines for each activity must be designed by a field expert — someone with enough insight to be able to dissect the activity of interest into an FSM. There are several approaches to inferring FSMs from its outputs in the literature [8][9], which we are now investigating. It is unclear whether the inferred state machines would be general enough to parse activities from different people, or in different environments. However, after the FSM is inferred, it may be given to a field expert, who uses it to extract the generalized version. Another weakness of the current system is that a number of parameters of this prototype system were chosen in a heuristic manner. Among these are the demarcation of the exact boundaries for each area provided on the map, and the selection of values for the *relevance period* parameter for each FSM state. Extracting these in an automated fashion may be the subject of future research.

Acknowledgments

This work was partially funded by the National Science Foundation under projects CNS 0448082 and ECCS 0622133. Any opinions, findings and conclusions or recommendation expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

References

[1] D. Ayers and M. Shah. Monitoring human behavior from video taken in an office environment. *Image and Vision Computing*, 19(12), Oct. 2001.

[2] A. Bamis, D. Lymberopoulos, T. Teixeira, and A. Savvides. Towards precision monitoring of elders for providing assistive services. In *PETRA '08: Proceedings of the 1st ACM international conference on PErvasive Technologies Related to Assistive Environments*, pages 1–8, New York, NY, USA, 2008. ACM.

[3] N. Bicocchi, M. Mamei, A. Prati, R. Cucchiara, and F. Zambonelli. Pervasive self-learning with multi-modal distributed sensors. In *Self-Adaptive and Self-Organizing Systems Workshops, IEEE International Conference on*, 2008.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2:1395–1402, Oct. 2005.

[5] W. Brunette, J. Lester, A. Rea, and G. Borriello. Some sensor network elements for ubiquitous computing. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005.

[6] W. Dong, K. Y. Lim, Y. K. Goh, K. D. Nguyen, I.-M. Chen, S. H. Yeo, and B.-L. Duh. A low-cost motion tracker and its error analysis. *Robotics and Automation, 2008. IEEE International Conference on*, pages 311–316, May 2008.

[7] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *Computer Vision, 2003. ICCV 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733, Oct. 2003.

[8] K. Koskimies and E. Mkinen. Inferring state machines from trace diagrams. *University of Tampere Department of Computer Science. Technical Report of the*, A-1993-3, July 1993.

[9] D. Lorenzoli, L. Mariani, and M. Pezzè. Inferring state-based behavior models. In *WODA '06: Proceedings of the 2006 international workshop on Dynamic systems analysis*, pages 25–32, New York, NY, USA, 2006. ACM.

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

[11] D. Lymberopoulos, T. Teixeira, and A. Savvides. Detecting patterns for assisted living: A case study. In *Proceedings of SensorComm*, 2007.

[12] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *BSN '06: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 2006.

[13] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(7):1052–1062, July 2006.

[14] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. *Computer Vision and Pattern Recognition. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:326–333, June-July 2004.

[15] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC '05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, 2005.

[16] M. Perkowitz, M. Philipose, K. Fishkin, and D. J. Patterson. Mining models of human activities from the web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, 2004.

[17] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton. Activity identification using body-mounted sensors: a review of classification techniques. *Physiological Measurement*, 30(4):R1–R33, 2009.

[18] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proc. 10th Int'l. Conf. Computer Vision*, volume 1, pages 824–831, 2005.

[19] L. Rodriguez-Benitez, J. Moreno-Garcia, J. Castro-Schez, C. Solana, and L. Jimenez. Action recognition in video sequences using a mealy machine. *World Congress of Science, Engineering and Technology. Proceedings of the*, 31:42–48, July 2008.

[20] T. Teixeira, D. Jung, G. Dublon, and A. Savvides. Identifying people in camera networks
using wearable accelerometers. In *Pervasive Technologies Related to Assistive Environments (PETRA)*, 2009.

[21] T. Teixeira, D. Jung, G. Dublon, and A. Savvides. PEM-ID: Identifying people by gait-matching using cameras and wearable accelerometers. In *ACM/IEEE International Conference on Distributed Smart Cameras*, 2009.

[22] T. Teixeira and A. Savvides. Lightweight people counting and localizing in indoor spaces using camera sensor nodes. In *ACM/IEEE International Conference on Distributed Smart Cameras*, September 2007.

[23] D. Wilson and C. Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In *In The Third International Conference on Pervasive Computing*, 2005.

[24] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:984–989, June 2005.

[25] J. Yin, Q. Yang, and J. Pan. Sensor-based abnormal human-activity detection. *Knowledge and Data Engineering, IEEE Transactions on*, 20(8), Aug. 2008.

[26] J. Yin, Q. Yang, D. Shen, and Z.-N. Li. Activity recognition via user-trace segmentation. *ACM Trans. Sen. Netw.*, 4(4):1–34, 2008.

[27] V. M. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics Publishers, September 1997.

[28] R. Zhu and Z. Zhou. A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. volume 12, pages 295–302, June 2004.