Human Activity Detection from RGBD Images

Jaeyong Sung and **Colin Ponce** and **Bart Selman** and **Ashutosh Saxena**
Department of Computer Science
Cornell University, Ithaca, NY 14850
js946@cornell.edu, {cponce,selman,asaxena}@cs.cornell.edu

Abstract

Being able to detect and recognize human activities is important for making personal assistant robots useful in performing assistive tasks. The challenge is to develop a system that is low-cost, reliable in unstructured home settings, and also straightforward to use. In this paper, we use a RGBD sensor (Microsoft Kinect) as the input sensor, and present learning algorithms to infer the activities. Our algorithm is based on a hierarchical maximum entropy Markov model (MEMM). It considers a person's activity as composed of a set of sub-activities, and infers the two-layered graph structure using a dynamic programming approach. We test our algorithm on detecting and recognizing twelve different activities performed by four people in different environments, such as a kitchen, a living room, an office, etc., and achieve an average performance of 84.3% when the person was seen before in the training set (and 64.2% when the person was not seen before).

Introduction

Being able to infer the activity that a person is performing is important for making personal assistant robots useful. For example, a key concern in elderly care is making sure that the person drinks enough water throughout the day to avoid dehydration. In such a situation, if a robot could keep track of how much a person has been drinking, it can remind the person to drink water. In this work, we are interested in reliably detecting daily activities that a person performs in their house, such as cooking, drinking water, brushing teeth, opening pill containers, and in their office, such as talking on phone, working on a computer, and so on.

Most previous work on activity classification has focused on using 2D video (e.g., Ning et al., 2009; Gupta et al., 2009) or RFID sensors placed on humans and objects (e.g., Wu et al., 2007). The use of 2D videos leads to relatively low accuracy (e.g., 78.5% in Liu et al., 2008) even when there is no clutter. The use of RFID tags is generally too intrusive because it requires a placement of RFID tags on the people.

In this work, we perform activity recognition using an inexpensive RGBD sensor (Microsoft Kinect). We show that we can achieve quite reliable performance in detection and recognition of common activities performed in typical cluttered human environments. Giving robots the ability to interpret human activity is an important step towards enabling humans to interact with robots in a natural way, as well as enabling a robot to be a more useful personal assistant.

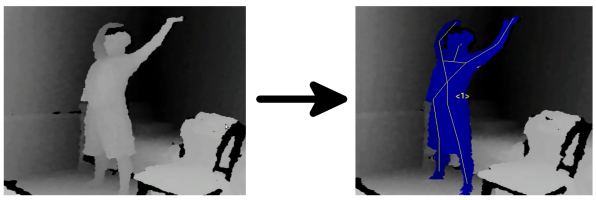


Figure 1: The RGBD data from the Kinect sensor is used to generate an articulated skeleton model of the person.

Our method is based on machine learning techniques, in which we first extract meaningful features based on the estimated human skeleton from the Kinect. Human activities, such as drinking water and brushing teeth, generally consist of a sequence of well defined sub-activities. For example, to drink water one pours the water, lifts the glass, tilts it, and then places it down. We capture such sequences of activities using a probabilistic graphical model—specifically a two-layered maximum-entropy Markov model (MEMM). The top layer represents activities, and the mid-layer represents sub-activities connected to the corresponding activities in the top-layer. One challenge that arises is to infer the structure of the graphical model itself; that is, to which top-layer activity do we assign a mid-layer node? We present an efficient method to do so based on dynamic programming.

We evaluated our method on twelve different activities (see Figure 3) performed by four different people in five different environments: kitchen, office, bathroom, living room and bedroom. Our results show an average overall performance of 84.3%in detecting the correct activity when the person was seen before in the training set (and 64.2% when the person was not seen before). We made the dataset and code available at: `http://pr.cs.cornell.edu/humanactivities`

Related Work

Activity Recognition from a 2D Video: Human activity recognition has been previously studied by a number of different authors. One common approach is to use space-time features to model points of interest in video (Laptev, 2005; Dollar et al., 2005). Several authors have supple-

mented these techniques by adding more information to these features (Jhuang et al., 2007; Wong, Kim, and Cipolla, 2007; Wu et al., 2007; Liu, Ali, and Shah, 2008). Other, less common approaches for activity recognition include filtering techniques (Rodriguez, Ahmed, and Shah, 2008), and sampling of video patches (Boiman and Irani, 2005).

Hierarchical techniques for activity recognition have been used as well, but these typically focus on neurologically-inspired visual cortex-type models (Giese and Poggio, 2003; Serre, Wolf, and Poggio, 2005; Mutch and Lowe, 2006; Ranzato et al., 2007). Often these authors adhere faithfully to the models of the visual cortex, using motion-direction sensitive "cells" such as Gabor filters in the first layer (Jhuang et al., 2007; Ning et al., 2009).

Another class of techniques used for activity recognition is that of the hidden Markov model (HMM). Early work by Brand, Oliver, and Pentland (1997) utilized coupled HMMs to recognize two-handed activities. Weinland, Boyer, and Ronfard (2007) utilize an HMM together with a three-dimensional occupancy grid to model three dimensional humans. Martinez-Contreras et al. (2009) utilize motion templates together with HMMs to recognize human activities. Sminchisescu et al. (2005) utilized conditional random fields and maximum-entropy Markov models for activity recognition, arguing that these models overcome some of the limitations presented by hidden Markov models. However, the use of 2D videos leads to relatively low accuracies.

Activity Recognition for Smart Houses: The goal of building activity classification systems can be approached from different directions. Our personal robotics approach could be contrasted with approaches based on so-called smart homes, which are homes wired with sensor networks that monitor the user (Chan et al., 2008). One important drawback of the smart home is cost. Simply installing the sensor network is often cost prohibitive. Another issue concerns privacy. Having cameras or other sensors throughout the house is often found to be too intrusive. A robotic solution functions much more like having an actual person in the house, thus avoiding the cost and intrusiveness of smart homes. In particular, the subject can choose not to let the robot in certain rooms or areas by simply closing a door. In earlier work on smart home environments, it was shown that one can get quite reliable in-home activity recognition when sufficiently accurate sensor data is available. However, this may require subjects to wear RFID sensors in an environment labeled with RFID tags, as was shown in (Philipose et al., 2004; Patterson et al., 2005), or highly engineered and task specific visual recognition procedures (Mihailidis et al., 2003). Also, GPS traces of a person can be utilized for activity recognition through a model based on hierarchical conditional random fields (CRF) (Liao, Fox, and Kautz, 2007). However, their model is only capable of off-line classification using several tens of thousands of data points. Our model, in contrast, is capable of on-line classification, which makes it more applicable to assistive robots.

Robotics Assistance and Recognition: Various robotic systems have used activity recognition before. Theodoridis et al. (2008) used activity recognition in robotic systems to discern the difference between normal and aggressive activity in humans. Li et al. (2007) discuss the importance of nonverbal communication between human and robot and develop a method to recognize simple activities that are nondeterministic in nature, while other activity recognition work has focused on developing robots that can utilize activity recognition to learn to imitate human activities (Demiris and Meltzoff, 2008; Lopes, Melo, and Montesano, 2007). However, we are more interested here in assistive robots. Assistive robots are robots that assist humans in some task. Several types of assistive robots exist, including socially assistive robots that interact with another person in a non-contact manner, and physically assistive robots, which can physically help people (Feil-Seifer and Matarié, 2005; Tapus, Ţăpuş, and Matarié, 2008; Nguyen et al., 2008; Li et al., 2011; Saxena, Driemeyer, and Ng, 2008; Jiang et al., 2011; Saxena et al., 2001).

Hierarchical Dynamic Bayesian Networks: Other authors have worked on hierarchical dynamic Bayesian networks. Hierarchical Hidden Markov Models (HHMMs) were first proposed by Liao, Fox, and Kautz (2007). Bui, Phung, and Venkatesh (2004) then extended this to a general structure in which each child in the hierarchy can have multiple parents instead of just one. Truyen et al. (2008) then developed a hierarchical conditional random field that could be used in partially observable settings. None of these models fit our problem, however, the Hierarchical Hidden Markov Model is the closest model to ours, but does not capture the idea that a single state may connect to different parents only for specified periods of time, as occurs in our model.

Our Approach

We use a supervised learning approach in which we collected ground-truth labeled data for training our model. Our input is RGBD images from a Kinect sensor, from which we extract certain features that are fed as input to our learning algorithm. We will train a two-layered Markov model which will capture different properties of human activities, including their hierarchical nature, the transitions between sub-activities over time, and the correspondence between sub-activities and human skeletal features.

Features

We can recognize a person's activity by looking at his current pose and movement over time, as captured by a set of features. The input sensor for our robot is a RGBD camera Kinect that gives us an RGB image as well as depths for each point. In order to compute the human pose features, we describe a person by a rigid skeleton that can move at fifteen joints (see Fig. 1). We extract this skeleton using a tracking system provided by PrimeSense (2011). The skeleton is described by the length of the links and the joint angles. Specifically, we have the three-dimensional Euclidean coordinates of each joint with respect to the sensor, and the orientation matrix of each joint also with respect to the sensor. We compute features from this information as follows.

Body pose features. The joint orientation is obtained with respect to the sensor. However, a person's pose is indepen-

dent of the angle from which the robot observes the person, and so our feature representation should also be invariant with respect to observation angle. Therefore, we transform each joint's rotation matrix so that the rotation is given with respect to the person's torso. For 10 joints, we convert each rotation matrix to half-space quaternions in order to more compactly represent the joint's orientation. (A more compact representation would be to use Euler angles, but they suffer from representation problem called gimbal lock (Saxena, Driemeyer, and Ng, 2009).) Along with these joint orientation, we would like to know whether person is standing or sitting and whether person is leaning towards any direction. Such information is observed from position of each foot in respect to the torso ($3*2$) and comparing the position of head against the heap joints. We have $10*4+3*2+1 = 47$ features for the body pose.

Hand Position. Hands play an especially important role in carrying out many activities, and so information about what hands are doing can be quite powerful. In particular, we want to capture information such as "the left hand is near the stomach" or "the right hand is near the right ear". To do this, we compute the position of the hands with respect to the torso, and with the respect to the head in the local coordinate frame. Though we capture the motion information as described next, in order to emphasize hand movement, we also observe hand position over last 60 frames and record the highest and lowest vertical hand position. We have $(6 + 2) = 16$ features for this.

Motion Information. Motion information is also important for classifying a person's activities. We select nine frames spread out over the last three seconds and compare current joint information to the joint information for those frames (for 11 joints, not counting the 4 joints at the ends of the hands and feet). For each frame, we compute the rotation that has occurred for each joint between the past frame and the current frame (and is added in the feature in the half-space quaternion representation). We choose nine frames over last three seconds, spaced as follows: $\{-5, -9, -14, -20, -27, -35, -44, -54, -65\}$, where the numbers refer to the frames chosen. We thus have $9*11*4 = 396$ features that represents motion information.

This gives us a total feature vector of size $47+16+396 = 459$, which is the input to our learning algorithm.

Learning Model

Human activity is complex and dynamic, and therefore our learning algorithm should model different nuances in human activities, such as the following.

First, an activity comprises a series of sub-activities. For example, the activity "brushing teeth" consists of sub-activities such as "squeezing toothpaste," "bringing toothbrush up to face," "brushing," and so forth. Therefore for each activity (represented by $z \in Z$), we will model sub-activities (represented by $y \in Y$). We will train a hierarchical Markov model where the sub-activities y are represented by a layer of hidden variables (see Figure 2).

For each activity, different subjects perform the sub-activities for different periods of time. It is not clear how

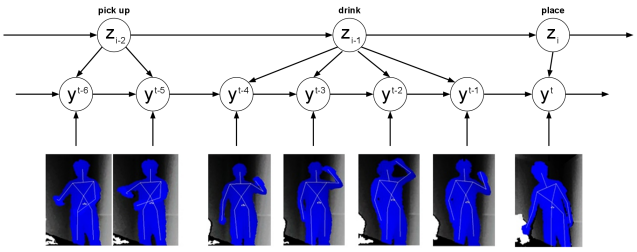


Figure 2: Our two-layered maximum-entropy Markov model.

to associate the sub-activities to the activities. This implies that the graph structure of model cannot be fixed in advance. We therefore determine the connectivity between the z and the y layers in the model during inference.

Model. Our model is based on a maximum-entropy Markov model (MEMM) (Mccallum, Freitag, and Pereira, 2000). However, in order to incorporate the hierarchical nature of activities, we use a two-layered hierarchical structure, as shown in Figure 2.

In our model, let x^t denote the features extracted from the articulated skeleton model at time frame t. Every frame is connected to high-level activities through the mid-level sub-activities. Since high-level activities do not change every frame, we do not index them by time. Rather, we simply write z_i to denote the i^{th} high-level activity. Activity i occurs from time $t_{i-1}+1$ to time t_i. Then $\{y^{t_{i-1}+1}, ..., y^{t_i}\}$ is the set of sub-activities connected to activity z_i.

MEMM with Hierarchical Structure

As shown in Figure 2, each node z_i in the top layer is connected to several consecutive nodes in the middle layer $\{y^{t_{i-1}+1}, ..., y^{t_i}\}$, capturing the intuition that a single activity consists of a number of consecutive sub-activities.

For the sub-activity at each frame y^t, we do not know a priori to which activity z_i it should connect at the top layer. Therefore, our algorithm must decide when to connect a middle-layer node y^t to top-layer node z_i and when to connect it to next top-layer node z_{i+1}. We show below how selection of graph structure can be done through dynamic programming. Given the graph structure, our goal is to infer the z_i that best explains the data. We do this by modeling the joint distribution $P(z_i, y^{t_{i-1}+1} \cdots y^{t_i} | O_i, z_{i-1})$, and for each z_i, we find the set of y^t's that maximize the joint probability. Finally, we choose the z_i that has the highest joint probability distribution.

We use a Gaussian mixture model to cluster the original training data into separate clusters, and consider each cluster as a sub-activity. In particular, we constrain the model to create five clusters for each activity. We then combine all the clusters for a certain location's activities into a single set of location specific clusters. In addition, we also generate a few clusters from the negative examples, so that our algorithm becomes robust to not detecting random activities. Specifically, for classifier for each location, we create a single cluster from each of the activities that do not occur in that location.

Our model consists of the following three terms:

- $P(y^t|x^t)$**:** This term models the dependence of the sub-activity label y^t on the features x^t. We model this using the Gaussian mixture model we have built. The parameters of the model is estimated from the labeled training data using maximum-likelihood.
- $P(y^{t_i-m}|y^{t_i-m-1}, z_i)$ (where $m \in \{0, ..., (t_i - t_{i-1} - 1)\}$). A sequence of sub-activities describe the activities. For example, we can say the sequence "squeezing toothpaste," "bringing toothbrush up to face," "actual brushing," and "putting toothbrush down" describes the activity "brushing teeth." If we only observe "bringing toothbrush up to face" and "putting toothbrush down," we would not refer to it as "brushing teeth." Unless the activity goes through a specific set of sub-activities in nearly the same sequence, it should probably not be classified as the activity. For all the activities except *neutral* , the table is built from observing the transition of posterior probability for soft cluster of Gaussian mixture model at each frame.
  However, it is not so straightforward to build $P(y^{t_i-m}|y^{t_i-m-1}, z_i)$ when z_i is *neutral*. When a sub-activity sequence such as "bringing toothbrush to face" and "putting toothbrush down" occurs, it does not correspond to any known activity and so is likely to be *neutral*. It is not possible to collect data of all sub-activity sequences that do not occur in our list of activities, so we rely on the sequences observed from non-*neutral* activities. If N denotes *neutral* activity, then
$$P(y^{t_i-m}|y^{t_i-m-1}, z_i = N) \propto 1 - \sum_{z_i \neq N} P(y^{t_i-m}|y^{t_i-m-1}, z_i)$$
- $P(z_i|z_{i-1})$. The activities evolve over time. For example, one activity may be more likely to follow another, and there are brief moments of *neutral* activity between two non-*neutral* activities. Thus, we can make a better estimate of the activity at the current time if we also use the estimate of the activity at previous time-step. Unlike other terms, due to difficulty of obtaining rich data set for maximum likelihood estimation, $P(z_i|z_{i-1})$ is set manually to capture these intuitions.

Inference. Consider the two-layer MEMM depicted in Figure 2. Let a single z_i activity node along with all the y^t sub-activity nodes connected directly to it and the corresponding x^t feature inputs be called a *sub-structure* of the MEMM graph. Given an observation sequence $O_i = x^{t_{i-1}+1}, ..., x^{t_i}$ and a previous activity z_{i-1}, we wish to compute the joint probability $P(z_i, y^{t_{i-1}+1} \cdots y^{t_i}|O_i, z_{i-1})$:

$$\begin{aligned}
&P(z_i, y^{t_{i-1}+1} \cdots y^{t_i}|O_i, z_{i-1}) \\
=&P(z_i|O_i, z_{i-1})P(y^{t_{i-1}+1} \cdots y^{t_i}|z_i, O_i, z_{i-1}) \\
=&P(z_i|z_{i-1}) \cdot \prod_{t=t_{i-1}+2}^{t_i} P(y^t|y^{t-1}, z_i, x^t) \\
&\cdot \sum_{y^{t_{i-1}}} P(y^{t_{i-1}+1}|y^{t_{i-1}}, z_i, x^{t_{i-1}+1})P(y^{t_{i-1}})
\end{aligned}$$

We have all of these terms except $P(y^t|y^{t-1}, z_i, x^t)$ and $P(y^{t_{i-1}+1}|y^{t_{i-1}}, z_i, x^{t_{i-1}+1})$. Both terms can be derived as

$$P(y^t|y^{t-1}, z_i, x^t) = \frac{P(y^{t-1}, z_i, x^t|y^t)P(y^t)}{P(y^{t-1}, z_i, x^t)}$$

We make a naive Bayes conditional independence assumption that y^{t-1} and z_i are independent from x^t given y^t. Using this assumption, we get:

$$P(y^t|y^{t-1}, z_i, x^t) = \frac{P(y^t|y^{t-1}, z_i)P(y^t|x^t)}{P(y^t)}$$

We have fully derived $P(z_i, y^{t_{i-1}+1} \cdots y^{t_i}|O_i, z_{i-1})$.

$$\begin{aligned}
&P(z_i, y^{t_{i-1}+1} \cdots y^{t_i}|O_i, z_{i-1}) \\
&= P(z_i|z_{i-1}) \\
&\quad \cdot \sum_{y^{t_{i-1}}} \frac{P(y^{t_{i-1}+1}|y^{t_{i-1}}, z_i)P(y^{t_{i-1}+1}|x^{t_{i-1}+1})}{P(y^{t_{i-1}+1})} P(y^{t_{i-1}}) \\
&\quad \cdot \prod_{t=t_{i-1}+2}^{t_i} \frac{P(y^t|y^{t-1}, z_i)P(y^t|x^t)}{P(y^t)}
\end{aligned}$$

Note that this formula can be factorized into two terms where one of them only contains two variables.

$$P(z_i, y^{t_{i-1}+1} \cdots y^{t_i}|O_i, z_{i-1}) = \mathcal{A} \cdot \prod_{t=t_{i-1}+2}^{t_i} \mathcal{B}(y^{t-1}, y^t)$$

Because the formula has factored into terms containing only two variables each, this equation can be easily and efficiently optimized. We simply optimize each factor individually, and we obtain:

$$\begin{aligned}
&\max P(z_i, y^{t_{i-1}+1} \cdots y^{t_i}|O_i, z_{i-1}) = \\
&\max_{y^{t_{i-1}+1}} \mathcal{A} \max_{y^{t_{i-1}+2}} \mathcal{B}(y^{t_{i-1}+1}, y^{t_{i-1}+2}) \cdots \max_{y^{t_i}} \mathcal{B}(y^{t_i-1}, y^{t_i})
\end{aligned}$$

Graph Structure Selection

Now that we can compute $P(z_i|O_i, z_{i-1})$, the probability of an activity z_i being associated with the i^{th} substructure and the previous activity, we wish to use that to compute the probability of z_i given all observations up to this point. However, to do this, we must solve the following problem: for each observation y_t, we must decide to which high-level action z_i it is connected (see Figure 2). For example, consider the last y node associated with the "drink" activity in Figure 2. It's not entirely clear if that node really should connect to the "drink" activity, or if it should connect to the following "place" activity. Deciding with which activity to associated each y node is the problem of hierarchical MEMM graph structure selection.

Unfortunately, we cannot simply try all possible graph structures. To see why, suppose we have a graph structure at time $t-1$ with a final high-level node z_i, and then are given a new node y^t. This node has two "choices": it can either connect to z_i, or it can create a new high-level node z_{i+1}

and connect to that one. Because every node y^t has this same choice, if we see a total of n mid-level nodes, then there are 2^n possible graph structures.

We present an efficient method to find the optimal graph structure using dynamic programming. The method works, in brief, as follows. When given a new frame for classification, we try to find the point in time at which the current high-level activity started. So we pick a time t', and say that every frame after t' belongs to the current high-level activity. We have already computed the optimal graph structure for the first t' time frames, so putting these two subgraphs together give us a possible graph structure. We can then use this graph to compute the probability that the current activity is z. By trying all possible times $t' < t$, we can find the one that gives us the highest probability, and we select that as our graph structure at time t.

The Method of Graph Structure Selection. Now we describe the method in detail. Suppose we are at some time t; we wish to select the optimal graph structure given everything we have seen so far. We will define the graph structure inductively based on graph structures that were chosen at previous points in time. Let $G_{t'}$ represent the graph structure that was chosen at some time $t' < t$. Note that, as a base case, G_0 is always the empty graph.

For every $t' < t$, define a new candidate graph structure $\tilde{G}_t^{t'}$ that consists of $G_{t'}$ (the graph structure that captures the first t' time frames), followed by a single substructure from time $t'+1$ to time t that connects to a single high-level node z_i. Note that this candidate graph structure sets $t_{i-1} = t'$ and $t_i = t$. Given the set of candidate graph structures $\{\tilde{G}_t^{t'} | 1 \leq t' < t\}$, the plan is to find the graph structure and high-level activity $z_i \in Z$ that maximizes the likelihood given the set of observations so far.

Let O be the set of all observations so far. Then $P(z_i|O;\tilde{G}_t^{t'})$ is the probability that the most recent high-level node i is activity $z_i \in Z$, given all observations so far and parameterized by the graph structure $\tilde{G}_t^{t'}$. We initially set $P(z_0|O;G_0)$ to a uniform distribution. Then, through dynamic programming, we have $P(z_{i-1}|O;G_{t'})$ for all $t' < t$ and all $z \in Z$ (details below). Suppose that, at time t, we choose the graph structure $\tilde{G}_t^{t'}$ for a given $t' < t$. Then the probability that the most recent node i is activity z_i is given by

$$\begin{aligned} P(z_i|O;\tilde{G}_t^{t'}) &= \sum_{z_{i-1}} P(z_i, z_{i-1}|O;\tilde{G}_t^{t'}) \\ &= \sum_{z_{i-1}} P(z_{i-1}|O;\tilde{G}_t^{t'}) P(z_i|O, z_{i-1};\tilde{G}_t^{t'}) \\ &= \sum_{z_{i-1}} P(z_{i-1}|O;G_{t'}) P(z_i|O_i, z_{i-1}) \quad (1) \end{aligned}$$

The two factors inside the summation are both terms that we know, the former due to dynamic programming, and the latter as described in the previous section.

Thus, to find the optimal probability of having node i be a specific activity z_i, we simply compute

$$P(z_i|O;G_t) = \max_{t'<t} P(z_i|O;\tilde{G}_t^{t'})$$

We store $P(z_i|O;G_t)\ \forall\, z_i$ for dynamic programming purposes (Equation 1). Then, to make a prediction of an activity at time t, we compute

$$\begin{aligned} \text{activity}_t &= \arg\max_{z_i} P(z_i|O) \\ &= \arg\max_{z_i} \max_{t'<t} P(z_i|O;\tilde{G}_t^{t'}) \end{aligned}$$

Optimality. We show that this algorithm is optimal by induction on the time t. Suppose we know the optimal graph structure for every time $t' < t$. This is certainly true at time $t = 1$, as the optimal graph structure at time $t = 0$ is the empty graph. The optimal graph structure at time t involves a final high-level node z_i that is connected to $1 \leq k \leq t$ mid-level nodes.

Suppose the optimal structure at time t has the high-level node connected to $k = t - t'$ mid-level nodes. Then what graph structure do we use for the first t' nodes? By the induction hypothesis, we already know the optimal graph structure $G_{t'}$ for the first t' nodes. That is, $G_{t'}$ is the graph structure that maximizes the probability $P(z_{i-1}|O)$. Because z_i is conditionally independent of any high-level node before z_{i-1}, the graph structure before z_{i-1} does not affect z_i. Similarly, the graph structure before z_{i-1} obviously does not depend on the graph structure after z_{i-1}. Therefore, the optimal graph structure at time t is $\tilde{G}_t^{t'}$, the concatenation of $G_{t'}$ to a single substructure of $t - t'$ nodes.

We do not know what the correct time $0 \leq t' < t$ is, but because we try all, we are guaranteed to find the optimal t', and therefore the optimal graph structure.

Experiments

Hardware. We used the Microsoft Kinect sensor, which consists of an RGB camera and an infrared structured light source for inferring depths. It outputs an RGB image together with aligned depths at each pixel at a frame rate of 30Hz. It produces a 640x480 depth image with a range of 1.2m to 3.5m. The sensor is small enough that it can be mounted on inexpensive mobile ground robots such as in (Li et al., 2011).

Data. We considered five different environments: office, kitchen, bedroom, bathroom, and living room. Three to four common activities were identified for each location, giving a total of twelve unique activities (see Table 1). Data was collected from four different people: two males and two females. None of the subjects were associated with this project (and hence were not knowledgeable of our models and algorithm). We collected about 45 seconds of data for each activity and each person. When collecting, the subject was given basic instructions on how to carry out the activity, such as "stand here and chop this onion," but was not given any instruction on how the algorithm would interpret his or her movements. All of the data was collected in a regular household setting with no occlusion of body from the view of sensor (see Figure 3).

Our goal is to perform human activity *detection*, i.e., our algorithm must be able to distinguish the desired activities from other random activities that people perform. To that

Figure 3: Samples from our dataset. Row-wise, from left: brushing teeth, cooking (stirring), writing on whiteboard, working on computer, talking on phone, wearing contact lenses, relaxing on a chair, opening a pill container, drinking water, cooking (chopping), talking on a chair, and rinsing mouth with water.

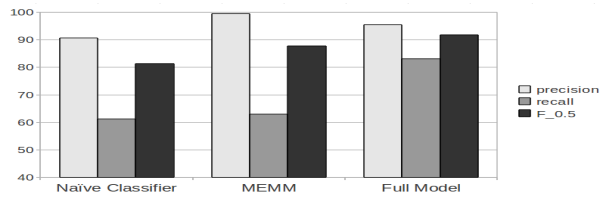


Figure 4: Comparison of our model against two baselines on a preliminary smaller dataset on the "person seen before" setting.

end, we collected *random* activities by asking the subject to act in a manner unlike any of the previously performed activities. The *random* activity ranges from a person standing still to a person walking around and stretching his or her body. Note that *random* data was only used for testing.

For testing, we experimented with two settings. In the "new person" setting, we employed leave-one-out cross-validation to test each person's data; i.e. the model was trained on three of the four people from whom data was collected, and tested on the fourth. In the other "person seen before" setting of the experiment, we wanted the model to have previously seen the person carry out the same activity. To achieve this setting, we halved the testing subject's data and included in the training data set. So, even though the model had seen the person do the activity at least once, they had not seen the testing data itself.

Finally, to train the model on both left-handed and right-handed people without needing to film them all, we simply mirrored the training data across the virtual plane down the middle of the screen.

We have made the data available at: `http://pr.cs.cornell.edu/humanactivities`

Models

We first compared the following algorithms against our full model (two-layered MEMM) on a preliminary dataset composed of two subjects in a setting where the person was seen before in the training. (See Figure 4.)

- *Baseline: Naive Classifier.* As the baseline model, we used a multi-class support vector machine (SVM) as a way to map features to corresponding activities. Here we rely only on the SVM classification of features to predict high-level activities directly, The SVM was trained separately for each location.
- *One-level MEMM.* This is a one-level MEMM model which builds upon the naive classifier. As explained before, $P(y^t|x^t)$ is computed by fitting a sigmoid function to the output of the SVM. Transition probabilities between activities, $P(y^t|y^{t-1})$, use the same table we have built for full model, which in that model is called $P(z_i|z_{i-1})$. Using $P(y^t|x^t)$ and $P(y^t|y^{t-1})$, we can compute the probability that the person is engaged in activity j at time t as follows.

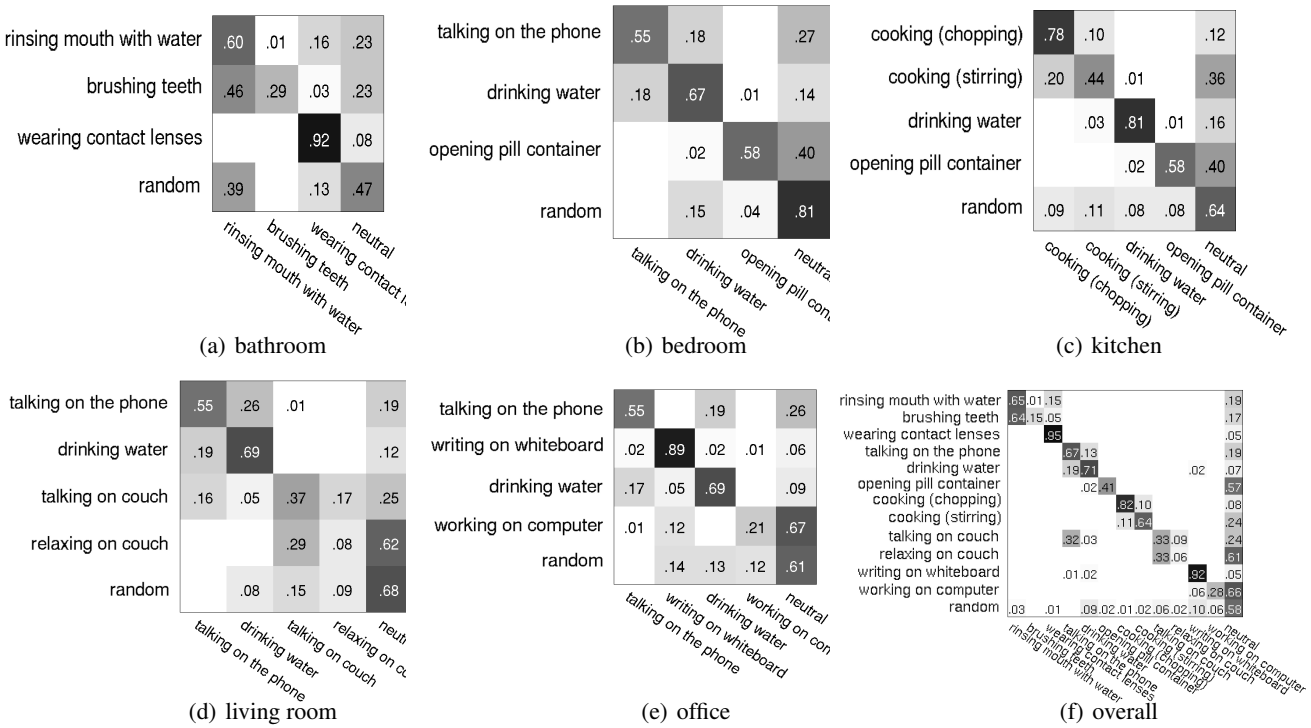
Table 1: Learning algorithm results. The table shows precision, recall and $F_{0.5}$ score for two training scenarios: when the person is previously seen by the algorithm during training, and when the person is not previously seen. Note that the test dataset consists of all kinds of random movements (in addition to the activities considered), ranging from a person standing still to walking around while waving his hands.

		Person seen before			New Person		
Location	Activity	Prec	Rec	$F_{0.5}$	Prec	Rec	$F_{0.5}$
bathroom	rinsing mouth	69.8	59.5	67.4	41.3	60.1	44.0
	brushing teeth	96.8	74.2	91.2	97.1	28.6	65.6
	wearing contact lens	80.3	91.2	82.3	74.1	91.6	77.0
	Average	82.3	75.0	80.3	70.8	60.1	62.2
bedroom	talking on the phone	88.2	80.2	86.5	74.7	54.6	69.6
	drinking water	88.5	78.2	86.2	65.8	67.3	66.1
	opening pill container	91.2	81.8	89.2	92.1	58.5	82.6
	Average	89.3	80.1	87.3	77.5	60.1	72.8
kitchen	cooking (chopping)	80.2	88.1	81.6	73.4	78.3	74.4
	cooking (stirring)	88.1	46.8	74.8	65.5	43.9	59.7
	drinking water	93.2	82.8	90.9	87.9	80.8	86.4
	opening pill container	86.6	82.2	85.7	86.4	58.0	78.7
	Average	87.0	75.0	83.3	78.3	65.2	74.8
living room	talking on the phone	75.7	82.1	76.9	61.2	54.9	59.8
	drinking water	84.5	80.3	83.6	64.1	68.7	64.9
	talking on couch	91.7	74.0	87.5	45.1	37.4	43.3
	relaxing on couch	85.7	84.6	85.4	24.4	8.3	17.5
	Average	84.4	80.3	83.4	48.7	42.3	46.4
office	talking on the phone	87.3	81.3	86.0	74.3	55.0	69.4
	writing on whiteboard	91.6	84.9	90.2	74.8	89.4	77.3
	drinking water	84.6	78.5	83.3	67.3	69.1	67.7
	working on computer	93.7	76.7	89.7	61.5	21.1	44.5
	Average	89.3	80.3	87.3	69.5	58.6	64.7
Overall Average		**86.5**	**78.1**	**84.3**	**69.0**	**57.3**	**64.2**

Our tests on the preliminary dataset are shown in Figure 4, which shows that our two-layered model outperforms the baseline models.

Results and Discussion

Table 1 shows the results of our full two-layered model for the "person seen before" and "new person" settings as described earlier. The precision, recall, and $F_{0.5}$ measures are



(a) bathroom (b) bedroom (c) kitchen

(d) living room (e) office (f) overall

Figure 5: Leave-one-out cross-validation confusion matrix for each location with the full model in "new person" settings. For definition of *random*, see text. The *neutral* activity denotes that the algorithm estimates that the person is either not doing anything or the person is engaged in some other activity that we have not defined. The last matrix (bottom-right) shows the results aggregated over all the locations.

used as metrics for evaluation. Since it is better for robot to not detect any activity rather than classifying an activity incorrectly (or start classifying a random activity as an expected activity), we used the measure $F_{0.5}$, which puts more emphasis on precision. Our model was able to classify with a $F_{0.5}$ measure of 84.31% and 64.17% in "person seen before" and "new person" settings, respectively.

Figure 5 shows the confusion matrices between the activities. When it did not classify correctly, it usually chose the *neutral* activity, which is typically not as bad as choosing a wrong "active" activity. As Figure 5-f shows, the most misclassified activity is "relaxing on couch" which was misclassified 33% of the time as "talking on couch"—these two activities are actually very similar, except that the person is more likely to be leaning backwards on the couch when relaxing. The other most misclassified activity is "brushing teeth" which was misclassified 64% of the time as "rinsing mouth with water". Since the skeleton tracking system was developed for the purpose of interaction in a gameplay setting, it is not yet designed to be robust against very subtle movements, causing activities such as "brushing teeth" to miss detailed hand motion information.

The model typically performed better in the "person seen before" setting. This is because every human has their own habits and slightly different way of doing each activity—recall that the activities we consider are daily activities which have a lot of variations and our instructions to the subjects were left to decide by themselves how to do the activity during our data collection. When our model had seen the person do same activity before, it becomes an easier classification task, thus giving an average $F_{0.5}$ score of 84.3% (as compared to 64.2% when the person was not seen before).

The real strength of all our model is that it also capable of correctly classifying *random* data as *neutral* most of the time, which means that it is able to distinguish whether the provided set of activities actually occurs or not—thus our algorithm is less likely to mis-fire when person is doing some new activity that the algorithm has not seen before. Also, since we trained on both the regular and mirrored data, the model is capable of performing well regardless of left- or right-handedness of the person.

However, there are some limitations to our method. First, our data only included cases in which the person was not occluded by an object; our method does not model occlusions and may not be robust to such situations. Second, some activities require more contextual information other than simply human pose. For example, knowledge of objects being used could help significantly in making human activity recognition algorithms more powerful in the future.

Conclusion

In this paper, we considered the problem of detecting and recognizing activities that humans perform in unstructured environments such as homes and offices. We used an inexpensive RGBD sensor (Microsoft Kinect) as the input sensor, the low cost of which enables our approach to be useful for applications such as smart homes and personal assistant robots. We presented a two-layered maximum entropy Markov model (MEMM). This MEMM modeled different properties of the human activities, including their hierarchical nature, the transitions between sub-activities over time, and the relation between sub-activities and human skeletal features. During inference, our algorithm exploited the hi-

erarchical nature of human activities to determine the best MEMM graph structure. We tested our algorithm extensively on twelve different activities performed by four different people in five different environments, where the test activities were often interleaved with random activities not belonging to these twelve categories. Our algorithm achieved an average performance of 84.3% when the person was previously seen in the training set, and 64.2% when the person was not seen before in the training set.

References

Boiman, O., and Irani, M. 2005. Detecting irregularities in images and in video. *IJCV* 74(1):17–31.

Brand, M.; Oliver, N.; and Pentland, A. 1997. Coupled hidden makov models for complex action recognition. In *CVPR*.

Bui, H.; Phung, D.; and Venkatesh, S. 2004. Hierarchical hidden markov models with general state hierarchy. In *AAAI*.

Chan, M.; Estève, D.; Escriba, C.; and Campo, E. 2008. A review of smart homes–present state and future challenges. *Computer Methods and Programs in Biomedicine* 91(1):55–81.

Demiris, Y., and Meltzoff, A. 2008. The robot in the crib: a developmental analysis of imitation skills in infants and robots. *Infant and Child Development* 17(1):43–53.

Dollar, P.; Rabaud, V.; Cottrell, G.; and Belongie, S. 2005. Behavior recognition via sparse spatio-temporal features. In *Int'l Wrksp Visual Surv Perf. Eval. Tracking Surv.*

Feil-Seifer, D., and Matarié, M. J. 2005. Defining socially assistive robots. In *ICORR*.

Giese, M., and Poggio, T. 2003. Neural mechanisms for the recognition of biological movement. *Nature Rev Neurosc.* 4:179–192.

Gupta, A.; Srinivasan, P.; Shi, J.; and Davis, L. S. 2009. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. *CVPR*.

Jhuang, H.; Serre, T.; Wolf, L.; and Poggio, T. 2007. A biologically inspired system for action recognition. In *ICCV*.

Jiang, Y.; Zheng, C.; Lim, M.; and Saxena, A. 2011. Learning to place new objects. In *Cornell University Technical Report*.

Laptev, I. 2005. On space-time interest points. *IJCV* 64(2):107–123.

Li, Z.; Wachsmuth, S.; Fritsch, J.; and Sagerer, G. 2007. *Vision Systems: Segmentation and Pattern Recognition*. InTech. chapter 8, 131–148.

Li, C.; Wong, T.; Xu, N.; and Saxena, A. 2011. Feccm for scene understanding: Helping the robot to learn multiple tasks. In *Video contribution in ICRA*.

Liao, L.; Fox, D.; and Kautz, H. 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. *IJRR* 26(1):119–134.

Liu, J.; Ali, S.; and Shah, M. 2008. Recognizing human actions using multiple features. In *CVPR*.

Lopes, M.; Melo, F. S.; and Montesano, L. 2007. Affordance-based imitation learning in robots. In *IROS*.

Martinez-Contreras, F.; Orrite-Urunuela, C.; Herrero-Jaraba, E.; Ragheb, H.; and Velastin, S. A. 2009. Recognizing human actions using silhouette-based hmm. In *AVSS*, 43–48.

Mccallum, A.; Freitag, D.; and Pereira, F. 2000. Maximum entropy markov models for information extraction and segmentation. In *ICML*.

Mihailidis, A.; Carmichael, B.; Boger, J.; and Fernie, G. 2003. An intelligent environment to support aging-in-place, safety, and independence of older adults with dementia. In *UbiHealth: 2nd Int'l Work. Ubi. Comp. Pervasive Healthcare*.

Mutch, J., and Lowe, D. 2006. Multiclass object recognition using sparse, localized features. In *CVPR*.

Nguyen, H.; Anderson, C.; Trevor, A.; Jain, A.; Xu, Z.; and Kemp, C. C. 2008. El-e: An assistive robots and fetches objects from flat surfaces. In *HRI*.

Ning, H.; Han, T. X.; Walther, D. B.; Liu, M.; and Huang, T. S. 2009. Hierarchical space-time model enabling efficient search for human actions. *IEEE Trans Circuits Sys. Video Tech.* 19(6).

Patterson, D. J.; Fox, D.; Kautz, H.; and Philipose, M. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *Ninth IEEE Int'l Symp on Wearable Computers*.

Philipose, M.; Fishkin, K. P.; Perkowitz, M.; Patterson, D. J.; Fox, D.; Kautz, H.; and Hahnel, D. 2004. Inferring activities from interactions with objects. *Pervasive Computing* 3(4):50–57.

PrimeSense. 2011. Nite middleware. http://www.primesense.com/.

Ranzato, M.; Huang, F. J.; Boureau, Y.-L.; and LeCun, Y. 2007. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*.

Rodriguez, M. D.; Ahmed, J.; and Shah, M. 2008. Action mach: A spatio-temporal maximum average correlaton height filter for action recognition. In *CVPR*.

Saxena, A.; Wong, L.; Quigley, M.; and Ng, A. 2001. A vision-based system for grasping novel objects in cluttered environments. In *ISRR*.

Saxena, A.; Driemeyer, J.; and Ng, A. Y. 2008. Robotic grasping of novel objects using vision. *IJRR* 27(2):157–173.

Saxena, A.; Driemeyer, J.; and Ng, A. 2009. Learning 3-d object orientation from images. In *ICRA*.

Serre, T.; Wolf, L.; and Poggio, T. 2005. Object recognition with features inspired by the visual cortex. In *CVPR*.

Sminchisescu, C.; Kanaujia, A.; Li, Z.; and Metaxas, D. 2005. Conditional models for contextual human motion recognition. In *ICCV*, 1808–1815.

Tapus, A.; Ţăpuş, C.; and Matarié, M. J. 2008. User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy. *Intel. Ser. Robotics* 1(2):169–183.

Theodoridis, T.; Agapitos, A.; Hu, H.; and Lucas, S. M. 2008. Ubiquitous robotics in physical human action recognition: A comparison between dynamic anns and gp. In *ICRA*.

Truyen, T. T.; Phung, D. Q.; Bui, H. H.; and Venkatesh, S. 2008. Hierarchical semi-markov conditional random fields for recursive sequential data. In *NIPS*.

Weinland, D.; Boyer, E.; and Ronfard, R. 2007. Action recognition from arbitrary views using 3d exemplars. In *ICCV*.

Wong, S.-F.; Kim, T.-K.; and Cipolla, R. 2007. Learning motion categories using both semantic and structural information. In *CVPR*.

Wu, J.; Osuntogun, A.; Choudhury, T.; Philipose, M.; and Rehg, J. M. 2007. A scalable approach to activity recognition based on object use. In *ICCV*.