Context-Based Bayesian Intent Recognition

Richard Kelley, Alireza Tavakkoli, Christopher King, Amol Ambardekar, Monica Nicolescu, Mircea Nicolescu

***Abstract*—One of the foundations of social interaction among humans is the ability to correctly identify interactions and infer the intentions of others. To build robots that reliably function in the human social world, we must develop models that robots can use to mimic the intent recognition skills found in humans. We propose a framework that uses contextual information in the form of object affordances and object state to improve the performance of an underlying intent recognition system. This system represents objects and their affordances using a directed graph that is automatically extracted from a large corpus of natural language text. We validate our approach on a physical robot that classifies intentions in a number of scenarios.**

I. Introduction

A precursor to social interaction is social understanding. Every day, humans observe each other and on the basis of their observations "read people's minds," correctly inferring the goals and intentions of others. Moreover, this ability is regarded not as remarkable, but as entirely ordinary and effortless. If we hope to build robots that are similarly capable of successfully interacting with people in a social setting, we must endow our robots with an ability to understand humans' intentions. In this paper, we propose a system aimed at developing those abilities in a way that exploits both an understanding of actions and the context within which those actions occur.

Our approach is ultimately based on psychological and neuroscientific evidence for a theory of mind [1], which suggests that the ease with which humans recognize the intentions of others is the result of an innate mechanism for representing, interpreting, and predicting other's actions. The mechanism relies on taking the perspective of others [2], which allows humans to correctly infer intentions.

Although this process is innate to humans, it does not take place in a vacuum. Intuitively, it would seem that our understanding of others' intentions depend heavily on the contexts in which we find ourselves and those we observe. This intuition is supported by neuroscientific results [3] which suggest that the context of an activity plays an important and sometimes decisive role in correctly inferring underlying intentions.

Our approach to developing this ability in robots consists of two stages: *activity modeling* followed by *intent recognition*. During activity modeling, our robot performs the activities it will later be expected to understand, using data it collects to train parameters of hidden Markov models (HMMs) representing the activities. Each HMM represents a single "basic activity." The hidden states of those HMMs correspond to small-scale goals or subparts of the activities. Most importantly, the visible states of a model represent the way in which parameters relevant to the activity change over time. For example, a visible state *distance-to-goal* may correspond to the way in which an observed agent's distance to some goal of the activity is changing, growing larger, smaller, or staying the same.

During intent recognition, the robot observes other agents interacting and performing various activities. The robot takes the perspective of the agents it is tracking and from their perspective calculates the changes in all parameters of interest. It uses the results of its calculations as inputs to its previously trained HMMs, inferring intentions using those models in conjunction with its prior knowledge of likely intention given the robot's (previously determined) spatio-temporal context.

The rest of our paper is structured as follows. Section II summarizes related work in activity modeling, intent recognition, and word sense disambiguation. Section III introduces lexical digraphs and their use in intent recognition. Section IV examines the intent recognition problem in detail, and demonstrates the necessity of contextual awareness in any general-purpose intent recognition system. Section V describes the novel computer vision methods that support our intent recognition system. Section VI provides the details of our recognition system, Section VII describes experiments in which we validated our approach on a physical robot, Section VIII gives a discussion of our approach and outlines possibilities for future work, and Section IX summarizes our paper.

II. Related Work

Whenever one wants to perform statistical classification in a system that is evolving over time, hidden Markov models may be appropriate [4]. Such models have been very successfully used in problems involving speech recognition [5]. There is also some evidence that hidden Markov models may be just as useful in modeling activities and intentions. For example, HMMs have been used by robots to perform a number of manipulation tasks [6][7][8]. These approaches all have the crucial problem that they only allow the robot to detect that a goal has been achieved *after* the activity has been performed; to the extent that intent recognition is about prediction, these systems do not use HMMs in a way that facilitates the recognition of intentions. Moreover, there are reasons to believe (see Sec. IV) that without considering the disambiguation component of intent recognition, there will be unavoidable limitations on a system, regardless of whether it uses HMMs or any other classification approach.

The problem of recognizing intentions is important in situations where a robot must learn from or collaborate with a human. Previous work has shown that forms of simulation or perspective taking can help robots work with people on joint tasks [10]. More generally, much of the work in learning by

Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557 USA. e-mail: {rkelley, tavakkol, cjking, ambardek, monica, mircea}@cse.unr.edu

demonstration has either an implicit or an explicit component dealing with interpreting ambiguous motions or instructions. The work we present here differs from that body of research in that we mostly focus on recognition in which the human is not actively trying to help the robot learn – ultimately intent recognition and learning by demonstation differ in this respect. However, we expect that the conclusions we have reached about the importance of linguistic information in disambiguation could be useful for the learning by demonstration community.

The use of HMMs in intent recognition (emphasizing the prediction element of the intent recognition problem) was first suggested in [9]. That paper also elaborates on the connection between the HMM approach and theory of mind. However, the system proposed there has shortcomings that the present work seeks to overcome. Specifically, that paper shows that in the absence of addition contextual information, a system that uses HMMs alone will have difficulty predicting intentions when two or more of the activities the system has been trained to recognize appear very similar. The model of perspective-taking that uses HMMs to encode low-level actions alone is insufficiently powerful to make predictions in a wide range of everyday situations.

The problem of intent recognition is also of great interest to researchers in neuroscience. Recent research in that field informs us that the mirror neuron system may play a role in intent recognition, and that contextual information is employed by the brain when ascribing intentions to others [3].

III. Lexical Digraphs

As mentioned above, our system relies on contextual information to perform intent recognition. Given that context is sometimes the decisive factor enabling human intent recognition [3], it makes sense to create robot architectures that use contextual information to improve performance. While there are many sources of contextual information that may be useful to infer intentions, we chose to focus primarily on the information provided by object affordances, which indicate the actions that one can perform with an object. The problem, once this choice is made, is one of training and representation: given that we wish the system to infer intentions from contextual information provided by knowledge of object affordances, how do we learn and represent those affordances? We would like, for each object our system may encounter, to build a representation that contains the likelihood of all actions that can be performed on that object.

Although there are many possible approaches to constructing such a representation, we chose to use a representation that is based heavily on a graph-theoretic approach to natural language – in particular, English. Specifically, we construct a graph in which the vertices are words and a labeled, weighted edge exists between two vertices if and only if the words corresponding to the vertices exist in some kind of grammatical relationship. The label indicates the nature of the relationship, and the edge weight is proportional to the frequency with which the pair of words exists in that particular relationship. For example, we may have vertices *drink* and *water*, along with the edge $((drink, water), direct\_object, 4)$, indicating that the word "water" appears as a direct object of the verb "drink" four times in the experience of the system. From this graph, we compute probabilities that provide the necessary context to interpret an activity. It seems likely that spoken and written natural language is not enough (on its own) to create reasonable priors for activity and intent recognition. However, we suggest that for a wide range of problems, natural language can provide information that improves prediction over systems that don't use contextual information at all.

A. Using Language for Context

The use of a linguistic approach is well-motivated by human experience. Natural language is a highly effective vehicle for expressing facts about the world, including object affordances. Moreover, it is often the case that such affordances can be easily inferred directly from grammatical relationships, as in the example above.

From a computational perspective, we would prefer models that are time and space efficient, both to build and to use. If the graph we construct to represent our affordances is sufficiently sparse, then it should be space efficient. As we discuss below, the graph we use has a number of edges that is linear in the number of vertices, which is in turn linear in the number of sentences that the system "reads." We thus attain space efficiency. Moreover, we can efficiently access the neighbors of any vertex using standard graph algorithms.

In practical terms, the wide availability of texts that discuss or describe human activities and object affordances means that an approach to modeling affordances based on language can scale well beyond a system that uses another means for acquiring affordance models. The act of "reading" about the world can, with the right model, replace direct experience for the robot in many situations.

Note that the above discussion makes an important assumption that, although convenient, may not be accurate in all situations. Namely, we assume that for any given action-object pair, the likelihood of the edge representing that pair in the graph is at least approximately equal to the likelihood that the action takes place in the world. Or in other words, we assume that linguistic frequency well approximates action frequency. Such an assumption is intuitively reasonable. We are more likely to read a book than we are to throw a book; as it happens, this fact is represented in our graph. We are currently exploring the extent to which this assumption is valid and may be safely relied upon; at this point, though, it appears that the assumption is valid for a wide enough range of situations to allow for practical use in the field.

B. Dependency Parsing and Graph Representation

To obtain our pairwise relations between words, we use the Stanford labeled dependency parser. The parser takes as input a sentence and produces the set of all pairs of words that are grammatically related in the sentence, along with a label for each pair, as in the "water" example above.

Using the parser, we construct a graph $G = (V, E)$, where E is the set of all labeled pairs of words returned by the

parser for all sentences, and each edge is given an integer weight equal to the number of times the edge appears in the text parsed by the system. V then consists of the words that appear in the corpus processed by the system.

C. Graph Construction and Complexity

1) Graph Construction: Given a labeled dependency parser and a set of documents, graph construction is straightforward. Briefly, the steps are

1) Tokenize each document into sentences.
2) For each sentence, build the dependency parse of the sentence.
3) Add each edge of the resulting parse to the graph.

Each of these steps may be performed automatically with reasonably good results, using well-known language processing algorithms. The end result is a graph as described above, which the system stores for later use.

One of the greatest strengths of the dependency-grammar approach is its space efficiency: the output of the parser is either a *tree* on the words of the input sentence, or a graph made of a tree plus a (small) constant number of additional edges. This means that the number of edges in our graph is a linear function of the number of nodes in the graph, which (assuming a bounded number of words per sentence in our corpus) is linear in the number of sentences the system processes. In our experience, the digraphs our system has produced have had statistics confirming this analysis, as can be seen by considering the graph used in our recognition experiments. For our corpus, we used two sources: first, the simplified-English Wikipedia, which contains many of the same articles as the standard Wikipedia, except with a smaller vocabulary and simpler grammatical structure, and second, a collection of childrens' stories about the objects in which we were interested. In Figure 1, we show the number of edges in the Wikipedia graph as a function of the number of vertices at various points during the growth of the graph. The scales on both axes are identical, and the graph shows that the number of edges for this graph does depend linearly on the number of vertices.

The final Wikipedia graph we used in our experiments consists of 244,267 vertices and 2,074,578 edges. The childrens' story graph is much smaller, being built from just a few hundred sentences: it consists of 1754 vertices and 3873 edges. This graph was built to fill in gaps in the information contained in the Wikipedia graph. The stories were selected from what could be called "childrens' nonfiction:" the books all contained descriptions and pictures of the world, and were chosen to cover the kinds of situations we trained our system to work in. The graphs were merged to create the final graph we used by taking the union of the vertex and edge sets of the graphs, adding the edge weights of any edges that appeared in both graphs.

2) Induced Subgraphs and Lexical "Noise": In some instances, our corpus may contain strings of characters that do not correspond to words in English. This is especially a problem if the system automatically crawls a resource such as the world wide web to find its sentences. We use the term *lexical noise* to refer to tokens that have vertices in our graph but are not in fact words in English. The extent to which such noise is a problem depends in large part on how carefully the documents are acquired, cleaned up, and tokenized into sentences before being given to the parser. Given the highly variable quality of many sources (such as blogs and other webpages) and the imperfect state of the art in sentence tokenization, it is necessary that we have a technique for removing lexical noise. Our current approach to such a problem is to work with induced subgraphs.

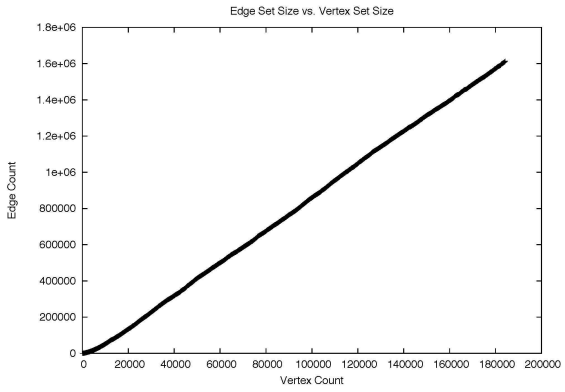

Fig. 1. The number of edges in the Wikipedia graph as a function of the number of vertices during the process of graph growth.

Suppose that we have a lexical digraph $G = (V, E)$, and a set of words $S \subseteq V$. We assume that we do *not* care about the "words" in $V - S$ (in fact, they may not even be words in our target language). Then instead of working with the graph G, we use the graph $G' = (S, E')$, where

$$E' = \{(x, y) \mid (x, y) \in E \wedge x, y \in S\}.$$

In addition to solving the problem of lexical noise, this approach has the benefit that it is easy to limit the system's knowledge to a particular domain if appropriate. For instance, we might make S a set of words about cars if we know we will be using the system in a context where cars are the only objects of interest. In this manner, we can carefully control the linguistic knowledge of our system and remove a source of error that is hard to avoid in a fully automated knowledge acquisition process.

IV. Reconsidering the Intent Recognition Problem

A. Disambiguation

Although some researchers consider the problems of activity recognition and intent recognition to be essentially the same, a much more common claim is that intent recognition differs from activity recognition in that intent recognition has a predictive component: by determining an agent's intentions, we are in effect making a judgment about what we believe are the likely actions of the agent in the immediate or near future. Emphasizing the predictive component of intent recognition is

important, but may not reveal all of the significant facets of the problem.

In contrast with the more traditional view of intent recognition, we contend that *disambiguation* is an essential task that any completely functional intent recognition system must be capable of performing. In emphasizing the disambiguation component of an intent recognition system, we recognize that there are some pairs of actions that may appear identical in all respects except for their underlying intentions.

For an example of intent recognition as disambiguation, consider an agent playing chess. When the agent reaches for a chess piece, we can observe that activity and ascribe to the agent any number of possible intentions. Before the game, an agent reaching for a chess piece may putting the piece into its initial position; during the game, the agent may be making a move using that piece; and after the game, the agent may be cleaning up and putting the piece away. In each of these cases, it is entirely possible (if not likely) that the activity of reaching for the piece will appear identical to the other cases. It is only the intentional component of each action that distinguishes it from the others. Moreover, this component is determined by the context of agent's activity: before, during, or after the game. Notice that we need to infer the agent's intention in this example even when we are not interested in making any predictions. Disambiguation in such circumstances is essential to even a basic understanding of the agent's actions.

B. Local and Global Intentions

In our work, we distinguish between two kinds of intentions, which we call local and global intentions. "Local" intentions exist on smaller time scales and may correspond to the individual parts of a complex activity. For example, if two agents are performing a "meeting" activity, they may approach one another, stop for some length of time, and then part ways. Each of these three components would correspond to a different local intention. In our approach, these local intentions are modeled using the hidden states of our HMMs, although of course there will be other ways to achieve the same result. As this modeling choice implies, though, local intentions are closely tied with particular activities, and it may not even be sensible to discuss these sorts of intentions outside of a given activity or set of activities.

In contrast, global intentions exist on larger time scales and correspond to complex activities in a particular context. In our chess example, "setting up the board," "making a move," and "cleaning up" would all correspond to to possible global intentions of the system.

This distinction between local and global intentions may be most useful during the activity modeling stage, if the activities being considered are sufficiently simple that they lack the internal structure that would lead to several local intentions, it may be that HMMs are not necessary for the system, so that a simpler purely Bayesian approach could be used instead. In this way, the distinction between local and global intentions can be used to develop a sense of the complexity of the activities being modeled in a given application.

V. Vision-Based Capabilities

In support of our intent recognition system, we require a number of visual capabilities for our robot. Among these, our system must be able to segment and track the motion of both humans and inanimate objects. Because we are interested in objects and their affordances, our system must also be able to visually identify objects and, for objects whose state can change over time, object states. Moreover, tracking should be done in three-dimensional space. To support this last requirement, we use a stereo-vision camera.

To perform segmentation and object recognition, we use a variant of maximally stable extremal regions for color images [19]. In our variant, we identify "strong" and "weak" edges in the image (based on our thresholding), and constrain the region-merging of color-based MSER so that region growth is inhibited across weak edges and prevented entirely across strong edges. This approach allows for increased stability, for multiple regions of different homogeneity to coexist near one another, and for more coherent segmentation of textured regions.

Having segmented a frame into regions, we perform object recognition using a mixture of Gaussians, computing probabilities at the *region* level rather than the pixel level. Because objects tend to consist of a smaller number of regions than pixels, this can lead to a substantial speedup.

Once we have segmented a frame and identified the regions of interest in that frame, we perform tracking via incremental support vector data descriptions and connected component analysis. We refer the interested reader to other, vision-specific work [9].

VI. Recognition System

As the visual tracking system performs its analysis of the scene, it passes along its results to the recognition system. That system then uses hidden Markov models and contextual information to calculate the intentions of each agent the robot observes.

A. Low-Level Recognition via Hidden Markov Models

As mentioned above, our system uses HMMs to model activities that consist of a number of parts that have intentional significance. Recall that a hidden Markov model consists of a set of hidden states, a set of visible states, a probability distribution that describes the probability of transitioning from one hidden state to another, and a probability distribution that describes the probability of observing a particular visible state given that the model is in a particular hidden state. To apply HMMs, one must give an interpretation to both the hidden states and the visible states of the model, as well as an interpretation for the model as a whole. In our case, each model λ represents a single well-defined activity. The hidden states of λ represent the intentions underlying the parts of the activity, and the visible symbols represent changes in measurable parameters that are relevant to the activity. Notice in particular that our visible states correspond to dynamic properties of the activity, so that our system can perform recognition as the observed agents are interacting.

As an example, consider the activity of meeting another person. To a first approximation, the act of meeting someone consists of approaching the person up to a point, interacting with the stationary person in some way (talking, exchanging something, etc.), and then parting. In our framework, we would model meeting using a single HMM. The hidden states would correspond to approach, halt, and part, since these correspond with the short-term intermediate goals of the meeting activity. When observing two people meeting, the two parameters of interest that we can use to characterize the activity are the distance and the angle between the two agents we're observing; in a meeting activity, we would expect that both the distance and the angle between two agents should decrease as the agents approach and face one another. With this in mind, we make the visible states represent changes in the distance and angle between two agents. Since each of these parameters is a real number, it can either be positive, negative, or (approximately) zero. There are then nine possibilities for a pair representing "change in distance" and "change in angle," and each of these nine possibilities represents a single visible state that our system can observe.

a) Training: We train our system in two ways. In situations where the robot can perform the activity, we have the robot perform that activity. With a Pioneer robot, this approach makes sense for activities like "follow an agent," or "pass by that person." As the robot performs the activity, it records features related to its motion (speed, direct, changes in its position relative to other agents, etc.). These are then converted to discrete symbols as described in the previous section. The symbols are then used to train HMMs representing each activity.

In situations where the robot cannot perform the activity (in our case, this included reaching for most objects), the system observes a human performing the task. The same features of the motion are recorded as in the previous training method, and are used to train an HMM.

In both cases, the topologies the HMMs and the interpretations of the hidden and visible states are determined by hand. The number of training examples generate with either method was limited due to the fact that a human had to perform the actions. In all cases below, we found that with just one or two dozen performances of the activities the system was able to train reasonably effective HMMs.

b) Recognition: During recognition, the stationary robot observes a number of individuals interacting with one another and with stationary objects. It tracks those individuals using the visual capabilities described above, and takes the perspective of the agents it is observing. Based on its perspective-taking and its prior understanding of the activities it has been trained to understand, the robot infers the intention of each agent in the scene. It does this using maximum likelihood estimation, calculating the most probable intention given the observation sequence that it has recorded up to the current time for each pair of interacting agents.

For more details on this use of HMMs for perspective taking and intent recognition, see our previous work [9].

B. Context Modeling

To use contextual information to perform intent recognition, we must decide how we want to model the relationship between intentions and contexts. This requires that we describe what intentions and contexts *are*, and that we specify how they are *related*. There are at least two plausible ways to deal with the latter consideration: we could choose to make intentions "aware" of contexts, or we might make contexts "aware" of intentions. In the first possibility, each intention knows all of the contexts in which it can occur. This would imply that we know in advance all contexts that are possible in our environment. Such an assumption may or may not be appropriate, given a particular application. On the other hand, we might make contexts aware of intentions. This would require that each context know, either deterministically or probabilistically, what intentions are possible in it. The corresponding assumption is that we know in advance all of the possible (or at least likely) intentions of the agents we may observe. Either of these approaches is possible, and may be appropriate for a particular application. In the present work, we adopt the latter approach by making each context aware of its possible intentions. This awareness is achieved by specifying the content of *intention models* and *context models*.

An intention model consists of two parts: first, an activity model, which is given by a particular HMM, and secondly a name. This is the minimal amount of information necessary to allow a robot to perform disambiguation. If necessary or desirable, intentions could be augmented with additional information that a robot could use to support interaction. As an example we might augment an intention model to specify an action to take in response to detecting a particular sequence of hidden states from the activity model.

A context model, at a minimum, must consist of a name or other identifier to distinguish it from other possible contexts in the system, as well as some method for discriminating between intentions. This method might take the form of a set of deterministic rules, or it might be a discrete probability distribution defined over the intentions about which the context is aware. In general, a context model can contain as many or as few features as are necessary to distinguish the intentions of interest. For our work, we focused on two kinds of information: the location of the event being observed, and the identities of any objects being interacted with by an agent. Context of the first kind was useful for basic experiments testing the performance of our system against a system that uses no contextual information, but did not use lexical digraphs at all; contexts and possible intentions we determined entirely by hand. Our other source of context, object identities, relied entirely on lexical digraphs as a way to represent object affordances. One of the major sources of information when inferring intent is contained by object affordances. Affordances indicate the types of actions that can be performed with a particular object and through their relative probabilities constrain the possible intentions a person can have when interacting with an object. For example, one can *drink from*, *break*, *empty*, or *wash* a glass, all with different probabilities. At the same time, the state of the object can

further constrain the potential intentions: it is more likely that one would drink from a full glass, while for an empty, diry glass, the most probable intention would be to wash it. We use the system described in section III to extract information about object affordances. The goal was to build a representation that contains, for each object, the likelihood of all actions that can be performed on that object. The system produces a weighted graph linking words that are connected in a dependency parse of a sentence in the corpus. The weights count the number of times each relationship appears, and must be converted to probabilities. To obtain the probability of each action given an object O, we look at all verbs V in relation to O and compute the probability of the verb V given the object O:

$$p(V|O) = \frac{w(O,V)}{\sum_{V' \in N(O)} w(O,V')},$$

where $N(O)$ consists of all verbs in the digraph that receive an arc from the O node and $w(O,V)$ is the weight of the arc from O to V, which we use as an approximation to the probability $p(O,V)$.

For objects that have different states (full vs. empty, open vs. closed, etc.), we infer the biased probabilities as follows:

- Merge the state vertex v_s and the object vertex v_o to obtain a new vertex v.
- update each edge weight $w(v, v_{neighbor})$ as follows:
  - 0 if $v_{neighbor}$ was not adjacent to both v_o and v_s.
  - $\min w(v_o, v_{neighbor}), w(v_s, v_{neighbor})$ otherwise.
- Normalize probabilities as in the stateless case.
- Return the probability distribution.

In this way, we can extract probabilities of actions for objects that are stateless as well as objects containing state.

c) Inference Algorithm: Suppose that we have an activity model (*i.e.* an HMM) denoted by w. Let s denote an intention, let c denote a context, and let v denote a sequence of visible states from the activity model w. If we are given a context and a sequence of observation, we would like to find the intention that is maximally likely. Mathematically, we would like to find

$$\arg\max_s p(s \mid v, c),$$

where the probability structure is determined by the activity model w.

To find the correct s, we start by observing that by Bayes' rule we have

$$\max_s p(s \mid v, c) = \max_s \frac{p(v \mid s, c)p(s \mid c)}{p(v \mid c)}. \quad (1)$$

We can further simplify matters by noting that the denominator is independent of our choice of s. Moreover, we assume without loss of generality that the possible observable symbols are independent of the current context. Based on these observations, we can write

$$\max_s p(s \mid v, c) \approx \max_s p(v \mid s)p(s \mid c). \quad (2)$$

This approximation suggests an algorithm for determining the most likely intention given a series of observations and a context: for each possible intention s for which $p(s \mid c) > 0$, we compute the probability $p(v \mid s)p(s \mid c)$ and choose as our intention that s whose probability is greatest. The probability $p(s \mid c)$ is available, either by assumption or from our linguistic model, and if the HMM w represents the activity model associated with intention s, then we assume that $p(v \mid s) = p(v \mid w)$. This assumption may be made in the case of location-based context for simplicity, or in the case of object affordances because we focus on simple activities such as reaching, where the same HMM w is used for multiple intentions s. Of course a perfectly general system would have to choose an appropriate HMM dynamically given the context; we leave the task of designing such a system as future work for now, and focus on dynamically deciding on the context to use, based on the digraph information.

C. Intention-Based Control

In robotics applications, simply determining an observed agent's intentions may not be enough. Once a robot knows what another's intentions are, the robot should be able to act on its knowledge to achieve a goal. With this in mind, we developed a simple method to allow a robot to dispatch a behavior based on its intent recognition capabilities. The robot first infers the global intentions of all the agents it is tracking, and for the activity corresponding to the inferred global intention determines the most likely local intention. If the robot determines over multiple time steps that a certain local intention has the largest probability, it can dispatch a behavior in response to the situation it believes is taking place.

For example, consider the activity of stealing an object. The local intentions for this activity might include "approaching the object," "picking up the object," and "walking off with the object." If the robot knows that in its current context the local intention "picking up the object" is not acceptable and it infers that an agent is in fact picking up the object, it can execute a behavior, for example stopping the thief or warning another person or robot of the theft.

VII. Experimental Validation

A. Setup

To validate our approach, we performed experiments in two different settings: a *surveillance setting* and a *household setting*. In the surveillance setting, we performed experiments using a Pioneer 2DX mobile robot, with an on-board computer, a laser rangefinder, and a Sony PTZ camera. In the household setting we performed experiments using both a pioneer robot and a humanoid Nao robot.

Fig. 2. HMM structure for the *follow* activity

Surveillance Setting: We trained our pioneer to understand three basic activitires: *following*, in which one agent trails behind another; *meeting*, in which two agents approach one another directly; and *passing*, in which two agents move past each other without otherwise directly interacting.

We placed our trained robot in an indoor environment and had it observe the interactions of multiple human agents with each other, and with multiple static objects. In our experiments, we considered both the case where the robot acts as a passive observer and the case where the robot executes an action on the basis of the intentions it infers in the agents under its watch.

We were particularly interested in the performance of the system in two cases. In the first case, we wanted to determine the performance of the system when a single activity could have different underlying intentions based on the current context (so that, returning to our example in Sec. IV, the activity of "moving one's hand toward a chess piece" could be interpreted as "making a move" during a game buy as "cleaning up" after the game is over). This case deals directly with the problem that in some situations, two apparently identical activities may in fact be very different, although the difference may lie entirely in the contextually determined intentional component of the activity.

In our second case of interest, we sought to determine the performance of the system in disambiguating two activities that were in fact different, but due to environmental conditions appeared superficially very similar. This situation represents one of the larger stumbling blocks of systems that do not incorporate contextual awareness.

In the first set of experiments, the same footage was given to the system several times, each with a different context, to determine whether the system could use context alone to disambiguate agents' intentions. We considered three pairs of scenarios: leaving the building on a normal day/evacuating the building, getting a drink from a vending machine/repairing a vending machine, and going to a movie during the day/going to clean the theater at night. We would expect our intent recognition system to correctly disambiguate between each of these pairs using its knowledge of its current context.

The second set of experiments was performed in a lobby, and had agents meeting each other and passing each other both with and without contextual information about which of these two activities is more likely in the context of the lobby. To the extent that meeting and passing appear to be similar, we would expect that the use of context would help to disambiguate the activities.

Lastly, to test our intention-based control, we set up two scenarios. In the first scenario (the "theft" scenario), a human enters his office carrying a bag. As he enters, he sets his bag down by the entrance. Another human enters the room, takes the bag and leaves. Our robot was set up to observe these actions and send a signal to a "patrol robot" in the hall that a theft had occurred. The patrol robot is then supposed to follow the thief for as long as possible.

In the second scenario, our robot is waiting in the hall, and observes a human leaving the bag in the hallway. The robot is supposed to recognize this as a suspicious activity and follow the human who dropped the bag for as long as possible.

TABLE I
QUANTITATIVE EVALUATION

Scenario (with Context)	Correct Duration [%]
Leave building (Normal)	96.2
Leave building (Evacuation)	96.4
Theater (Cleanup)	87.9
Theater (Movie)	90.9
Vending (Getting Drink)	91.1
Vending (Repair)	91.4
Meet (No context) - Agent 1	65.8
Meet (No context) - Agent 2	72.4
Meet (Context) - Agent 1	97.8
Meet (Context) - Agent 2	100.0

Household Setting: In the household setting, we performed experiments that further tested the system's ability to predict intentions and perform actions based on those predictions. We performed two sets of experiments. In the first set of experiments, we trained the pioneer to recognize a number of household objects and activities and to disambiguate between similar activities based on contextual information. Specifically, we had the system observe three different scenarios: a homework scenario, in which a human was observed reading books and typing on a laptop; a meal scenario, in which a human was observed eating and drinking; and an emergency scenario, in which a human was observed using a fire extinguisher to put out a fire in a trash can.

In the second set of experiments, we trained a humanoid robot to observe a human eating or doing homework. The robot was programmed to predict the observed human's intentions and offer assistance at socially appropriate moments. We used these scenarios to evaluate the performance of the lexical digraph approach.

B. Results

In both settings, our robots were able to effectively observe the agents within their fields of view and correctly infer the intentions of the agents that they observed. Videos of system performance for both the pioneer and the humanoid robot can be found at http://www.cse.unr.edu/~rkelley/robot-videos.html.

To provide a quantitative evaluation of intent recognition performance, we use two measures:

- *Accuracy rate* = the ratio of the number of observation sequences, of which the winning intentional state matches the ground truth, to the total number of test sequences.
- *Correct Duration* = C/T, where C is the total time during which the intentional state with the highest probability matches the ground truth and T is the number of observations.

The accuracy rate of our system is 100%: the system ultimately chose the correct intention in all of the scenarios in which it was tested. In practice this means very little. Much more interesting is the correct duration. We consider the correct duration measure in more detail for each of the cases in which we were interested.

1) One Activity, Many Intentions: The first six rows of Table I indicate the system's disambiguation performance. For example, we see that in the case of the scenario *Leave Building*, the intentions *normal* and *evacuation* are correctly inferred 96.2 and 96.4 percent of the time, respectively. We obtain similar results in two other scenarios where the only difference between the two activities in question is the intentional information represented by the robot's current context. We thus see that the system is able to use this contextual information to correctly disambiguate intentions.

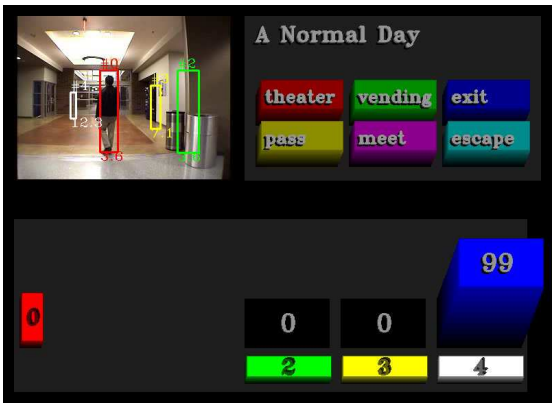

Fig. 3. Using context to infer that an agent is leaving a building, under normal circumstances. The human (with identifier 0 in the image) is moving toward the door (identifier 4), and the system is 99% confident that agent 0's intent is to exit the building. Agent 0 is not currently interacting with objects 2 or 3, so the system does not attempt to classify agent 0's intentions with respect to those objects.

2) Similar-Looking Activities: As we can see from the last four rows of Table I, the system performs substantially better when using context than it does without contextual information. Because *meeting* and *passing* can, depending on the position of the observer, appear very similar, without context it may be hard to decide what two agents are trying to do. With the proper contextual information, though, it becomes much easier to determine the intentions of the agents in the scene.

3) Intention-Based Control: In both the scenarios we developed to test our intention-based control, our robot correctly inferred the ground-truth intention, and correctly responded the inferred intention. In the theft scenario, the robot correctly recognized the theft and reported it to the patrol robot in the hallway, which was able to track the thief. In the bag drop scenario, the robot correctly recognized that dropping a bag off in a hallway is a suspicious activity, and was able to follow the suspicious agent through the hall. Both examples indicate that dispatching actions based on inferred intentions using context and hidden Markov models is a feasible approach.

4) Lexical-Digraph-Based System:

a) Pioneer Robot Experiments: To test the lexically-informed system in the household setting, we considered three different scenarios. In the first, the robot observed a human during a meal, eating and drinking. In the second, the human was doing homework, reading a book and taking notes on a computer. In the last scenario, the robot observed a person sitting on a couch, eating candy. A trashcan in the scene then

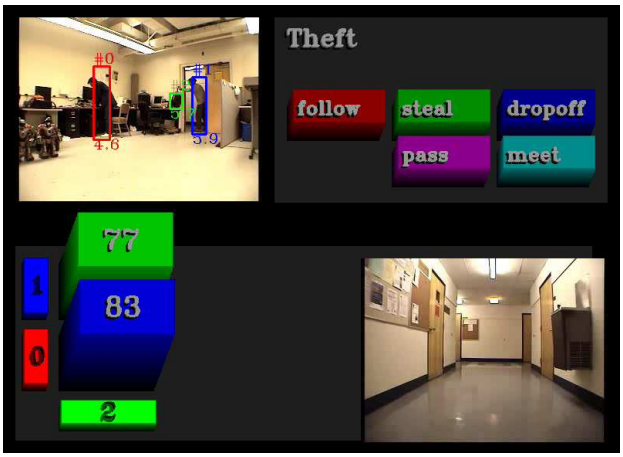

Fig. 4. An observer robot catches an agent stealing a bag. The top left video is the observer's viewpoint, the top left bars represent possible intentions, the bottom right bars are the robot's inferred intentions for each agent (with corresponding probabilities), and the bottom right video is the patrol robot's viewpoint.

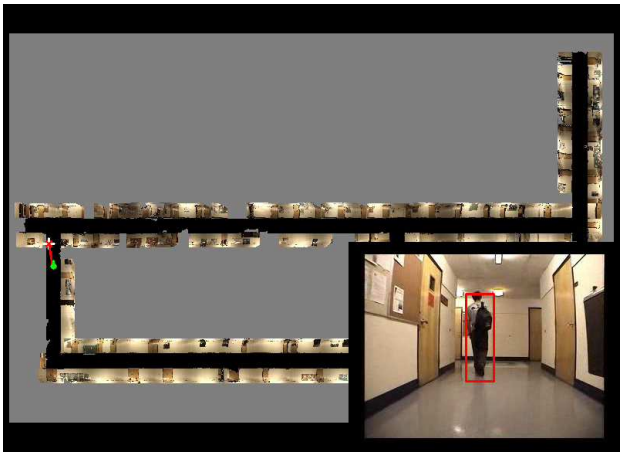

Fig. 5. A patrol robot, notified that a theft has occurred, sees the thief in the hallway and follows him. The video is the patrol robot's viewpoint superimposed on a map of the building.

catches on fire, and the robot observes the human using a fire extinguisher to put the fire out.

In the first set of experiments (the homework scenario), the objects their states, and the available activities were:

- Book (open): read, keep, copy, have, put, use, give, write, own, hold, study.
- Book (closed): have, put, use, give, own, open, take.
- Mouse: click, move, use.
- Bottle (full): find, drink, squeeze, shake, have, put, take.
- Laptop (open): boot, configure, break, take, leave.
- Laptop (closed): boot, configure, break, take, leave.

For the eating scenario, the objects, states, and activities were:

- Pitcher: find, drink, shake, have, throw, put, take, pour.
- Glass (full): hold, break, drink.
- Glass (empty): hold, break.
- Plate (full): eat, think-of, sell, give.
- Plate (empty): throw.

And for the fire scenario, the objects and activities were:

- Snack: eat, think-of, sell, give.
- Extinguisher: keep, extinguish, use.

In each scenario, the robot observed a human interacting with these objects by performing some of the above activities.

Defining a ground truth for these scenarios is slightly more difficult than in the previous scenarios, since in these scenarios the observed agent performs multiple activities and the boundaries between activities in sequence are not clearly defined. However, we can report that, except on the boundary between two activities, the correct duration of the system is 100%. Performance on the boundary is more variable, but it isn't clear that this is an avoidable phenomenon. We are currently working on carefully ground-truthed videos to allow us to better compute the accuracy rate and the correct duration for these sorts of scenarios.

b) Humanoid Robot Experiments: To test the system performance on another robot platform, we had our humanoid Nao observe a human doing homework and eating. The objects, states, and activities for these scenarios were the same as in the pioneer experiments listed above, with one additional object in the homework scenario: we trained the system to recognize a blank piece of paper, along with the intention of writing. We did this so that the robot could offer a pen to the human upon recognizing the human's intention to write.

To demonstrate that the robot detects human intentions, the robot takes certain actions or speaks to the human as soon as the intentions is recognized. This is based on a basic dialog system in which, for each intention, the robot has a certain repertoire of actions or utterances it can perform. Our experiments indicate that the robot correctly detects user intentions, before the human's actions are finalized. Moreover, no delays or misidentified intentions occurred, ensuring that the robot's responses to the human were not inappropriate for the human's activities. Tables II and III detail the interactions between the human and the robot in these scenarios.

VIII. Discussion and Future Work

There are a number of strengths and weaknesses of the proposed system that are worth pointing out. Both the strengths and weaknesses point to future work that can be done to improve the system.

A. Strengths

In addition to the improved performance of a context-aware system over a context-agnostic one that we see in the experimental results above, the proposed approach has a few other advantages worth mentioning. First, our approach recognizes the importance of context in recognizing intentions and activities, and can successfully operate in situations that previous intent recognition systems have had trouble with.

In real-world applications, the number of possible intentions that a robot has to be prepared to deal with may be very large. Without effective heuristics, efficiently performing maximum likelihood estimation in such large spaces is likely to be difficult if not impossible. In each of the above scenarios, the number of possible intentions the system had to consider was reduced through the use of contextual information. In general, such information may be used as an effective heuristic for reducing the size of the space the robot has to search to classify agents' intentions. As systems are deployed in increasingly complex situations, it is likely that heuristics of this sort will become important for the proper functioning of social robots.

Most importantly, though, from a design perspective it makes sense to separately perform inference for activities and for contexts. By "factoring" our solution in this way, we increase modularity and create the potential for improving the system by improving its individual parts. For example, it may turn out that another classifier works better than HMMs to model activities. We could then use that superior classifier in place of HMMs, along with an unmodified context module, to obtain a better-performing system.

B. Shortcomings

Our particular implementation has some shortcomings that are worth noting. First, the use of static context is inflexible. In some applications, such as surveillance using a set of stationary cameras, the use of static context may make sense. However, in the case of robots, the use of static context means that it is unlikely that the system will be able to take much advantage of one of the chief benefits of robots, namely their mobility.

Along similar lines, the current design of the intention-based control mechanism is probably not flexible enough to work "in the field." Inherent stochasticity, sensor limitations, and approximation error make it likely that a system that dispatches behaviors based only on a running count of certain HMM states is likely to run into problems with false positives and false negatives. In many situations (such as the theft sccenario describe above), even a relatively small number of such errors may not be acceptable.

In short, then, the system we propose faces a few substantial challenges, all centering on a lack of flexibility or robustness in the face of highly uncertain or unpredictable environments.

C. Future Research

The work presented raises a number of questions and suggests a number of avenues for future research. We are currently exploring extensions to our system that would allow for dynamic context, giving the robot the ability to change the context that it uses to infer intentions, based on either an instruction from a human operator or as a result of its own decision-making process. Along similar lines, we are currently working to give our robots the ability to infer the current context from features of the environment, which would substantially increase the flexibility of the system an allow for greater mobility in the intent-inferring robot.

On a much simpler note, we are exploring the use of multiple contexts during recognition. For example, we may want to be able to separately consider the robot's location and the time of day in determining what an agent's likely intentions are. Or we may want to use a particular context based on the presence or absence of certain objects in the environment (an agent cannot have the goal of throwing a ball if there are no balls in the room). In any case, similar reasoning to

TABLE II
HOMEWORK SCENARIO - THIS TABLE DESCRIBES THE INTERACTIONS THAT TAKE PLACE BETWEEN THE HUMAN AND OUR HUMANOID ROBOT. AT THE END OF THE SCENARIO, THE ROBOT GRABS A PEN AND HANDS IT TO THE HUMAN.

human action	object/context detected	intention	robot action/utterance	human utterance
reach for book	book : closed	take	"Hey, I know that book. It is about robots"	"That is right"
open book and read	book : open	read	"Are you going to read for a long time?"	"A little while"
reach for laptop	laptop : closed	take laptop	"I see you need to start your computer"	"That's right"
open laptop and type	laptop : open	type	"I will get some rest while you type"	"Thank you"
close laptop	laptop : closed	take laptop	"Oh you are done!"	
reach for paper	paper	write	"Do you need a pen for your writing?"	"Sure"

TABLE III
EATING SCENARIO - WHEN THE HUMAN ACCEPTS THE ROBOT'S OFFER OF A FORK, THE ROBOT HANDS THE FORK TO THE HUMAN. AT THE END OF THE SCENARIO, THE ROBOT WALKS TO THE HUMAN, TAKES THE PLATE FROM HIS HAND, AND THROWS IT AWAY.

human action	object/context detected	intention	robot action/utterance	human utterance
reach for food	paper plate : full	eat	"I see it is time for lunch. Would you like a fork?"	"Sure"
reach for bottle	bottle	pour	"Do you have a glass for your drink?"	"Yes, I have a glass."
reach for class	glass : full	drink	"Be careful - you do not want to spill"	"Yes, thank you."
reach for food on plate	paper plate : full	eat		
reach for empty plate	paper plate : empty	throw away	"Do you want me to throw that away?"	"Sure"

that used in Sec. VI suggests that we can model the situation mathematically using the equation:

$$p(s \mid v, c_1, \ldots, c_n) \approx p(v \mid s) \prod_{i=1}^{n} p(s \mid c_i), \qquad (3)$$

a straightforward (but potentially useful) extension to the present approach.

One of the interesting natural language problems that arose in the course of our work was synonymy. For instance, it's reasonable to think that the neighborhood of the word "laptop" in our lexical graph could be combined with the neighborhood of the word "computer" to produce more robust predictions. We are currently working on methods that exploit the structure of the lexical graph to identify subgraphs containing synonyms or strongly similar words.

Lastly, we recognize that our current approach to intention-based control will probably not remain as successful as the number of activities, intentions, or contexts increases. We are therefore looking into increasing the robustness of control based on inferred intentions. Additionally, we are looking to extend our system to forms of control that range beyond simple action dispatch. Among other possibilities, we are considering how intentional information could be used to bias the outputs of controllers for underactuated systems.

IX. CONCLUSION

In this paper, we proposed an approach to intent recognition that combines theory of mind with contextual awareness in a mobile robot. Understanding intentions in context is an essential human activity, and with high likelihood will be just as essential in any robot that must function in social domains. The approach we propose is based on perspective-taking and experience gained by the robot using its own sensory-motor capabilities. The robot carries out inference using its previous experience and its awareness of its own spatio-temporal context. We described the visual capabilities that support our robot's intent recognition, and validated our approach on a physical robot that was able to correctly determine the intentions of a number of people performing multiple activities in a variety of contexts.

ACKNOWLEDGEMENTS

The work has been supported by the Office of Naval Research under award number N00014-09-1-1121.

REFERENCES

[1] D. Premack and G. Woodruff, "Does the chimpanzee have a theory of mind?" *Behav. Brain Sci.* **1**(4) 515-526 (1978)
[2] A. Gopnick and A. Moore, "Changing your views: How understanding visual perception can lead to a new theory of mind" in *Children's Early Understanding of Mind*, eds. C. Lewis and P. Mitchell, 157-181. Lawrence Erlbaum (1994)
[3] Iacobini, M., Molnar-Szakacs, I., Gallese, V., Buccino, G., Mazziotta, J., Rizzolatti, G. "Grasping the Intentions of Others with One's Own Mirror Neuron System," PLoS Biol 3(3):e79 (2005)
[4] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience (2000)
[5] L. R. Rabiner, A tutorial on hidden-Markov models and selected applications in speech recognition, in Proc. IEEE 77(2) (1989)
[6] P. Pook and D. Ballard, Recognizing teleoperating manipulations, in Int. Conf. Robotics and Automation (1993), pp. 578585.
[7] G. Hovland, P. Sikka and B. McCarragher, Skill acquisition from human demon- stration using a hidden Markov model, Int. Conf. Robotics and Automation (1996), pp. 27062711.
[8] K. Ogawara, J. Takamatsu, H. Kimura and K. Ikeuchi, Modeling manipulation inter- actions by hidden Markov models, Int. Conf. Intelligent Robots and Systems (2002), pp. 10961101.
[9] A. Tavakkoli, R. Kelley, C. King, M. Nicolescu, M. Nicolescu, and G. Bebis, "A Vision-Based Architecture for Intent Recognition," *Proc. of the International Symposium on Visual Computing*, pp. 173-182 (2007)
[10] J. Gray, C. Breazeal, M. Berlin, A. Brooks, J. Lieberman, "Action Parsing and Goal Inference Using Self as Simulator," *IEEE International Workshop on Robot and Human Interactive Communication*, 2005.
[11] Dalal, N., Triggs, B.: Histogram of Oriented Gradients for Human Detection. International Conference on Pattern Recognition. (2005) pp. 886–893

[12] Ramanan, D., Forsyth, D., Zisserman, A.: Tracking People by Learning Their Appearances. IEEE PAMI **29** (2007) pp. 65–81
[13] Efors, J., Berg, A., Morri, G., Malik, J.: Recognizing action at a distance. In: Intl. Conference on Computer Vision. (2003)
[14] Stauffer, C., Grimson, W.: Learning Patterns of Activity using Real-Time Tracking. IEEE Transactions on PAMI **22** (2000) 747–757
[15] Tavakkoli, A., Nicolescu, M., Bebis, G.: Automatic Statistical Object Detection for Visual Surveillance. In proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation (2006) 144–148
[16] Tax, D., Duin, R.: Support Vector Data Description. Machine Learning **54** (2004) 45–66.
[17] J. Platt: Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning. **MIT Press** (1998) 185–208.
[18] Osuna, E., Freund, R., Girosi, F.: Improved Training Algorithm for Support Vector Machines. In Proc. Neural Networks in Signal Processing (1997)
[19] P. Forssen "Maximally Stable Colour Regions for Recognition and Matching," *CVPR* 2007.
[20] Tax, D., Laskov, P.: Online SVM Learning: from Classification and Data Description and Back. Neural Networks and Signal Processing (2003) 499–508.