Understanding Activities and Intentions for Human-Robot Interaction

Richard Kelley, Alireza Tavakkoli, Christopher King,
Monica Nicolescu and Mircea Nicolescu
University of Nevada, Reno
United States of America

1. Introduction

As robots move from the factory and into the daily lives of men, women, and children around the world, it is becoming increasingly clear that the skills they will require are vastly different from the majority of skills with which they were programmed in the 20th century. In fact, it would appear that many of these skills will center on the challenge of interacting with humans, rather than with machine parts or other robots. To this end, modern-day roboticists are actively studying the problem of human-robot interaction – how best to create robots that can interact with humans, usually in a social setting. Among the many problems of human robot interaction, one of the most interesting is the problem of *intent recognition*: the problem of predicting the intentions of a person, usually just by observing that person. If we understand intentions to be non-observable goal-directed mental activities, then we may (quite understandably) view the intent recognition problem for robots as one of *reading peoples' minds*.

As grandiose as this claim may sound, we believe that this understanding of intent recognition is quite reasonable; it is this interpretation that we seek to justify in the following pages.

Every day, humans observe one another and on the basis of their observations "read people's minds," correctly inferring the 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 review the intent recognition problem, and provide as an example a system we have been developing to recognize human intentions. Our approach is ultimately based on psychological and neuroscientific evidence for a theory of mind (Premack & Woodruff, 1978), 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 (Gopnick & Moore, 1994), 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

Source: Human-Robot Interaction, Book edited by: Daisuke Chugo,
ISBN 978-953-307-051-3, pp. 288, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

neuroscientific results (Iacobini et al., 2005), which suggest that the context of an activity plays an important and sometimes decisive role in correctly inferring underlying intentions.

Before considering this process in detail, we first look at some of the related work on the problem of intent recognition. After that, we reconsider the problem of intent recognition, looking at it from a new perspective that will shed light on how the process is accomplished. After looking at this re-framing of the problem, we consider some more general questions related to intent recognition, before moving on to describe a specific example system. We describe the architecture of our system, as well as experimental results we have obtained during validation of our system. We move on to describe some of the challenges facing future intent recognition systems, including planning based on recognized intentions, complexity of recognition, and the incorporation of novel sources of information for intent recognition systems. We then conclude with a summary of the central issues in the field of intent recognition.

2. Related work

Whenever one wants to perform statistical classification in a system that is evolving over time, hidden Markov models may be appropriate (Duda et al., 2000). Such models have been very successfully used in problems involving speech recognition (Rabiner, 1989). Recently, there has been some indication that hidden Markov models may be just as useful in modelling activities and intentions. For example, HMMs have been used by robots to perform a number of manipulation tasks (Pook and Ballard, 93), (Hovland et al., 96), (Ogawara et al., 2002). 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. 3) 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 use of HMMs in intent recognition (emphasizing the prediction element of the intent recognition problem) was first suggested in (Tavakkoli et al., 2007). 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.

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 (Iacobini et al., 2005).

3. Reconsidering the intent recognition problem

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. To understand such pairs of activities, our system must be able to recognize intentions even when making intent-based predictions is not necessary.

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.

4. Vision-based capabilities

We provide a set of vision-based perceptual capabilities for our robotic system that facilitate the modelling and recognition of actions carried out by other agents. As the appearance of these agents is generally not known a priori, the only visual cue that can be used for detecting and tracking them is image motion. Although it is possible to perform segmentation from an image sequence that contains global motion, such approaches -- typically based on optical flow estimation (Efros et al., 2003) -- are not very robust and are time consuming. Therefore, our approach uses more efficient and reliable techniques from real-time surveillance, based on background modelling and segmentation:

- During the *activity modelling* stage, the robot is moving while performing various activities. The appearance models of other mobile agents, necessary for tracking, are built in a separate, prior process where the static robot observes each agent that will be used for action learning. The robot uses an enhanced mean-shift tracking method to track the foreground object.
- During the *intent recognition* stage, the static robot observes the actions carried out by other agents. This allows the use of a foreground-background segmentation technique to build appearance models on-line, and to improve the speed and robustness of the tracker. The robot is stationary for efficiency reasons. If the robot moves during intent recognition we can use the approach from the modelling stage.

Fig. 1 shows the block diagram of the proposed object tracking frameworks.

4.1 Intent recognition visual tracking module

We propose an efficient Spatio-Spectral Tracking module (SST) to detect objects of interest and track them in the video sequence. The major assumption is that the observer robot is static. However, we do not make any further restrictions on the background composition, thus allowing for local changes in the background such as fluctuating lights, water fountains, waving tree branches, etc.

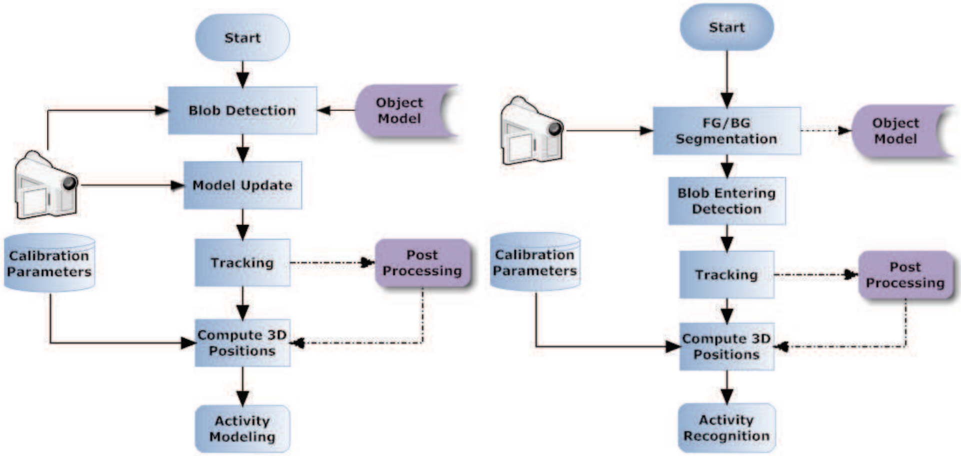

Fig. 1. The two object tracking frameworks for (a) *activity modelling* using a modified mean-shift tracker and (b) *intent recognition* using a spatio-spectral tracker.

The proposed system models the background pixel changes using an Incremental Support Vector Data Description module. The background model is then used to detect foreground regions in new frames. The foreground regions are processed further by employing a connected component processing in conjunction with a blob detection module to find objects of interest. These objects are tracked by their corresponding statistical models that are built from the objects' spectral (color) information. A laser-based range finder is used to extract the objects' trajectories and relative angles from their 2-D tracking trajectories and their depth in the scene. However, the spatio-spectral coherency of tracked objects may be violated in cases when two or more objects occlude each other.

A collision resolution mechanism is devised to address the issue of occlusion of objects of interest. This mechanism uses the spatial object properties such as their size, the relative location of their center of mass, and their relative orientations to predict the occlusion (collision).

4.2 Incremental support vector data description

Background modelling is one of the most effective and widely used techniques to detect moving objects in videos with a quasi-stationary background. In these scenarios, despite the presence of a static camera, the background is not completely stationary due to inherent changes, such as water fountains, waving flags, etc. Statistical modelling approaches estimate the probability density function of the background pixel values. If the data is not drawn from a mixture of normal distributions the parametric density estimation techniques may not be useful. As an alternative, non-parametric density estimation approaches can be used to estimate the probability of a given sample belonging to the same distribution function as the data set (Tavakkoli et al., 2006). However, the memory requirements of the non-parametric approach and its computational costs are high since they require the evaluation of a kernel function for all data samples.

Support Vector Data Description (SVDD) is a technique that uses support vectors in order to model a data set (Tax & Duin, 2004). The SVDD represents one class of known data samples in such a way that for a given test sample it can be recognized as known, or rejected as novel. Training of SVDDs is a quadratic programming optimization problem. This optimization converges by optimizing only on two data points with a specific condition (Platt, 1998) which requires at least one of the data points to violate the KKT conditions – the conditions by which the classification requirements are satisfied (Osuna et al., 1997). Our experimental results show that our SVDD training achieves higher speed and require less memory than the online and the canonical training (Tax & Duin, 2004).

4.3 Blob detection and object localization

In the blob detection module, the system uses a spatial connected component processing to label foreground regions from the previous stage. However, to label objects of interest a blob refinement framework is used to compensate for inaccuracies in physical appearance of the detected blobs due to unintended region split and merge, inaccurate foreground detection, and small foreground regions. A list of objects of interest corresponding to each detected blob is created and maintained to further process and track each object individually. This raw list of blobs corresponding to objects of interest is called the spatial connected component list.

Spatial properties about each blob such as its center and size are kept in the spatial connected component list. The list does not incorporate individual objects' appearances and thus is not solely useful for tracking purposes. The process of tracking individual objects based on their appearance in conjunction with their corresponding spatial features is carried out in the spatio-spectral tracking mechanism.

4.4 Spatio-spectral tracking mechanism

A system that can track moving objects (i.e. humans) requires a model for individual objects. These appearance models are employed to search for correspondences among the pool of objects detected in new frames. Once the target for each individual has been found in the new frame they are assigned a unique ID. In the update stage the new location, geometric and photometric information for each visible individual are updated. This helps recognize the objects and recover their new location in future frames.

Our proposed appearance modelling module represents an object with two sets of histograms, for the lower and upper half of the body. In the spatio-spectral tracking module a list of known objects of interest is maintained. This list represents each individual object and its corresponding spatial and color information along with its unique ID. During the tracking process the system uses the raw spatial connected component list as the list of observed objects and uses a statistical correspondence matching to maintain the ordered objects list and track each object individually. The tracking module is composed of three components:

- **Appearance modelling.** For each object in the raw connected component list a model is generated which contains the object center of mass, its height and width, the upper and lower section foreground masks, and the multivariate Gaussian distribution models of its upper and lower section pixels.
- **Correspondence matching.** The pixels in the upper and lower sections of each object in the raw list are used against each model in the ordered list of tracked objects. The winner model's ID then is used to represent the object.
- **Model update.** Once the tracking is performed the models will be updated. Any unseen object in the raw list is then assigned a new ID and their models are updated accordingly.

4.5 Collision resolution

In order for the system to be robust to collisions -- when individuals get too close so that one occludes the other-- the models for the occluded individual may not reliable for tracking purposes. Our method uses the distance of detected objects and uses that as a means of detecting a collision. After a collision is detected we match each of the individual models with their corresponding representatives. The one with the smallest matching score is considered to be occluded. The occluded object's model will not be updated but its new position is predicted by a Kalman filter. The position of the occluding agent is updated and tracked by a well-known mean-shift algorithm. After the collision is over the spatio-spectral tracker resumes its normal process for these objects.

5. Recognition system

5.1 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 represents a single well-defined activity. The hidden states of 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.

We train our HMMs by having our robot perform the activity that it later will recognize. As it performs the activity, it records the changes in the parameters of interest for the activity, and uses those to generate sequences of observable states representing the activity. These are then used with the Baum-Welch algorithm (Rabiner, 1989) to train the models, whose topologies have been determined by a human operator in advance.

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.

5.2 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. Moreover, the context can be either *static* or *dynamic.*

A static context consists of a name for the context and a probability distribution over all possible intentions. This is the simplest approach to context-based intent recognition in our framework, and is useful for modelling context that depends on unchanging location of an observer robot (as we would see in the case of a guard or service robot that only works in a single room or building), or on time or the date.

A dynamic context consists of features that are inferred by the observer. This could include objects that are being manipulated by the observed agents, visually detected features of the agents, or aspects of the environment that vary in hard-to-predict ways. In general, a dynamic context consists of a name and a probability distribution over *feature values* given the context. While being obviously more general than static context, a dynamic-context approach depends on good algorithms outside of the intent recognition domain, and can be (very) computationally expensive. However, the flexibility of the approach may justify the cost in a large number of potential applications.

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 the s that maximizes $p(s \mid v, c)$, where the probability structure is determined by the activity model w.

We can further simplify matters by noting that the denominator is independent of our choice of s. Moreover, because the context is simply a distribution over intention names, the observable symbols are independent of the current context. Based on these observations, we can say that $p(s|v,c)$ is approximately equal to $p(v|s)p(s|c)$.

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 $p(s|c) > 0$, we compute the probability $p(v|s)p(s|c)$ and choose as our intention that s whose probability is greatest. Because we assume a static context, the probability $p(s|c)$ is available by assumption, and if the HMM w represents the activity model associated with intention s, then we assume that $p(v|s) = p(v|w)$. In our case this assumption is justified since our intention models contain only a name and an activity model, so that our assumption only amounts to assuming that observation sequences are independent of intention names.

5.3 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.

6. Experimental validation

6.1 Setup

To validate our approach, we performed a set of experiments using a Pioneer 3DX mobile robot, with an on-board computer, a laser rangefinder, and a Sony PTZ camera. We trained our robot to understand three basic activities: *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. 3, the activity of "moving one's hand toward a chess piece" could be interpreted as "making a move" during a game but 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 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 visual data was given to the system several times, each with different a context, to determine whether the system could use the context alone to disambiguate agents' intentions. We considered three pairs of scenarios, which provided the context we gave to our system: 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 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.

6.2 Results

In all of the scenarios considered, our robot was able to effectively observe the agents within its field of view and correctly infer the intentions of the agents that it observed.

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. We consider the correct duration measure in more detail for each of the cases in which we were interested.

6.3 One activity, many intentions

Table 1 indicates 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.

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 a Drink)	91.1
Vending (Repair)	91.4

Table 1. Quantitative Evaluation.

6.4 Similar-looking activities

As we can see from Table 2, 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.

Meet (No Context) – Agent 1	65.8
Meet (No Context) – Agent 2	74.2
Meet (Context) - Agent 1	97.8
Meet (Context) – Agent 2	100.0

Table 2. Quantitative Evaluation.

6.5 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 (Figure 2). 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 intention-based control using context and hidden Markov models is a feasible approach.

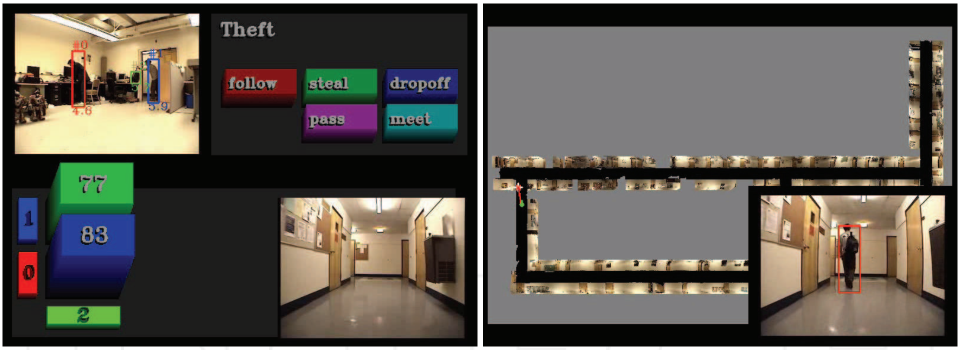

Fig. 2. An observer robot catches a human stealing a bag (left). The top left view shows the robot equipped with our system. The bottom right is the view of a patrol robot. The next frame (right) shows the patrol robot using vision and a map to track the thief.

6.6 Complexity of recognition

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.

7. Discussion

7.1 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.

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.

7.2 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 scenario 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.

8. Extensions

To deal with the problems of flexibility and scalability, we extend the system just described in two directions. First, we introduce a new source for contextual information, the lexical digraph. These data structures provide the system with contextual knowledge from linguistic sources, and have proved thus far to be highly general and flexible.

To deal with the problem of scalability, we introduce the *interaction space*, which abstracts the notion that people who are interacting are "closer" to each other than people who aren't, we are careful about how we talk about "closeness." In what follows, we outline these extensions, discussing how they improve upon the system described thus far.

9. Lexical digraphs

As mentioned above, our system relies on contextual information to perform intent recognition. 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.

There are a number of justifications for and consequences of the decision to take such an approach.

9.1 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 modelling 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.

9.2 Dependency parsing and graph representation

To obtain our pairwise relations between words, we use the Stanford labeled dependency parser (Marneffe et al., 2006). 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.

9.3 Graph construction and complexity

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 3, 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.

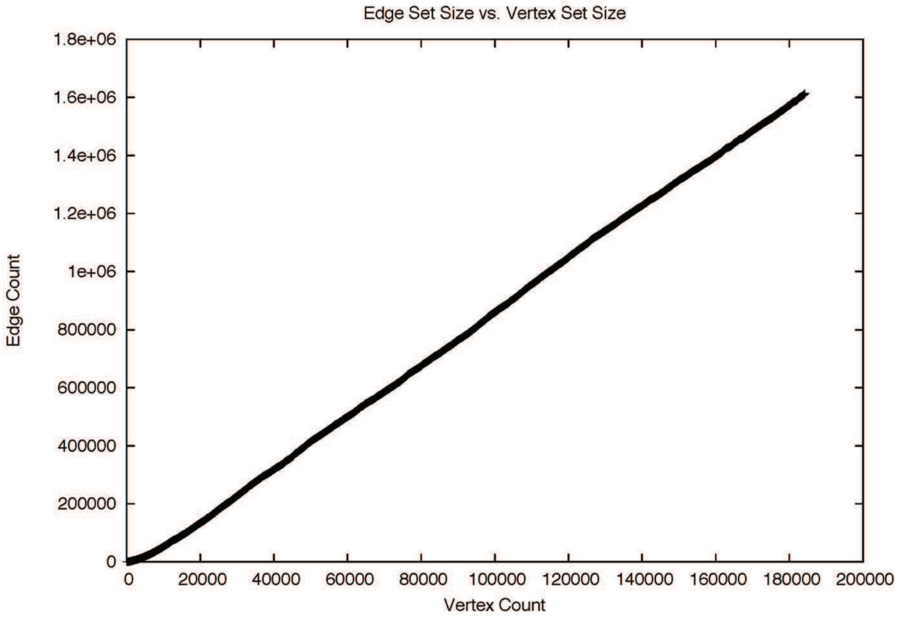

Fig. 3. The number of edges in the Wikipedia graph as a function of the number of vertices during the process of graph growth.

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 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.

9.4 Experimental validation and results

To test the lexical-digraph-based system, we had the robot observe an individual as he performed a number of activities involving various objects. These included books, glasses of soda, computers, bags of candy, and a fire extinguisher.

To test the lexically informed system, 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 catches on fire, and the robot observes the human using a fire extinguisher to put the fire out.

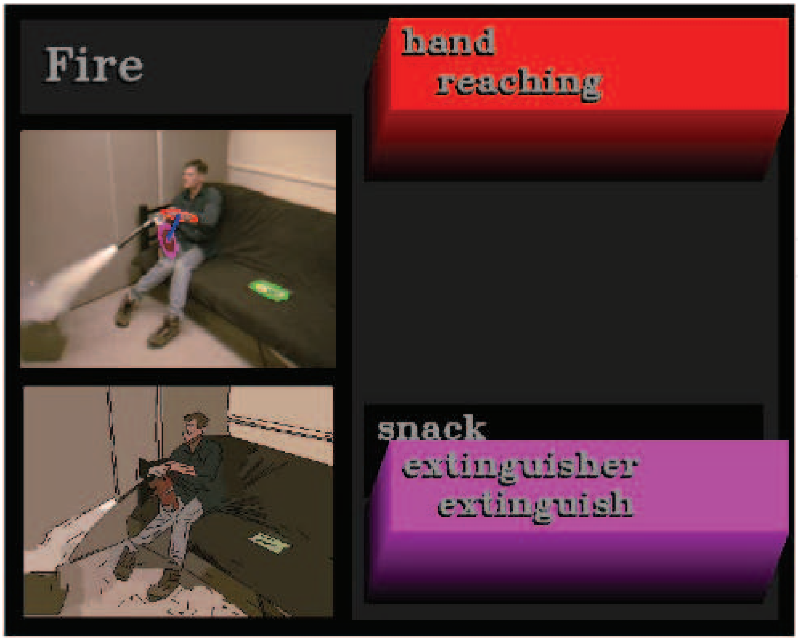

Fig. 4. The robot observer watches as a human uses a fire extinguisher to put out a trashcan fire.

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 still make the interesting observation 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. However, the results we have thus far obtained are encouraging.

10. Identifying interactions

The first step in the recognition process is deciding what to recognize. In general, a scene may consist of many agents, interacting with each other and with objects in the environment. If the scene is sufficiently complex, approaches that don't first narrow down the likely interactions before using time-intensive classifiers are likely to suffer, both in terms of performance and accuracy. To avoid this problem, we introduce the *interaction space* abstraction: for each identified object or agent in the scene, we represent the agent or object as a point in a space with a weak notion of distance defined on it. In this space, the points ideally (and in our particular models) have a relatively simple internal structure to permit efficient access and computation. We then calculate the distance between all pairs of points in this space, and identify as interacting all those pairs of entities for which the distance is less than some threshold. The goal in designing an interaction space model is that the distance function should be chosen so that the probability of interaction is decreasing in distance. We should not expect, in general, that the distance function will be a metric in the sense of analysis. In particular, there is no reason to expect that the triangle inequality will hold for all useful functions. Also, it is unlikely that the function will satisfy a symmetry condition: Alice may intend to interact with Bob (perhaps by secretly following him everywhere) even if Bob knows nothing about Alice's stalking habits. At a minimum, we only require nonnegativity and the trivial condition that the distance between any entity and itself is always zero. Such functions are sometimes known as premetrics.

For our current system, we considered four factors that we identified as particularly relevant to identifying interaction: distance in physical space, the angle of an entity from the center of an agent's field of view, velocity, and acceleration. Other factors that may be important that we chose not to model include sensed communication between two agents (this would be strongly indicative of interaction between two agents), time spent in and out of an agent's field of view, and others. We classify agents as interacting whenever a weighted sum of these distances is less than a human-set threshold.

10.1 Experimental validation and results

To test the interaction space model, we wished to use a large number of interacting agents behaving in a predictable fashion, and compare the results of an intent recognition system that used interaction spaces against the results of a system that did not. Given these requirements, we decided that the best approach was to simulate a large number of agents interacting in pre-programmed ways. This satisfied our requirements and gave us a well-defined ground truth to compare against.

The scenario we used for these experiments was very simple. The scenario consisted of *2n* simulated agents. These agents were randomly paired with one another, and tasked with approaching each other or engaging in a wander/follow activity. We looked at collections of eight and thirty-two agents. We then executed the simulation, recording the performance of the two test recognition systems. The reasoning behind such a simple scenario is that if a substantial difference in performance exists between the systems in this case, then regardless of the absolute performance of the systems for more complex scenarios, it is likely that the interaction-space method will outperform the baseline system.

The results of the simulation experiments show that as the number of entities to be classified increases, the system that uses interaction spaces outperforms a system that does not. As we can see in Table 3, for a relatively small number of agents, the two systems have somewhat comparable performance in terms of correct duration. However, when we increase the number of agents to be classified, we see that the interaction-space approach *substantially* outperforms the baseline approach.

	8 Agents	32 Agents
System with Interaction Spaces	96%	94%
Baseline System	79%	6%

Table 3. Simulation results – correct duration.

11. Future work in intent recognition

There is substantial room for future work in intent recognition. Generally speaking, the task moving forward will be to increase the flexibility and generality of intent recognition systems. There are a number of ways in which this can be done. First, further work should address the problem of a non-stationary robot. One might have noticed that our work assumes a robot that is not moving. While this is largely for reasons of simplicity, further work is necessary to ensure that an intent recognition system works fluidly in a highly dynamic environment.

More importantly, further work should be done on context awareness for robots to understand people. We contend that a linguistically based system, perhaps evolved from the one described here, could provide the basis for a system that can understand behavior and intentions in a wide variety of situations.

Lastly, beyond extending robots' *understanding* of activities and intentions, further work is necessary to extend robots' ability to *act* on their understanding. A more general framework for intention-based control would, when combined with a system for recognition in dynamic environments, allow robots to work in human environments as genuine partners, rather than mere tools.

12. Conclusion

In this chapter, we proposed an approach to intent recognition that combines visual tracking and recognition 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. Our approach is based on the view that to be effective, an intent recognition system should process information from the system's sensors, as well as relevant social information. To encode that information, we introduced the lexical digraph data structure, and showed how such a structure can be built and used. We demonstrated the effectiveness of separating interaction identification from interaction classification for building scalable systems. We discussed the visual capabilities necessary to implement our framework, and validated our approach in simulation and on a physical robot.

When we view robots as autonomous agents that increasingly must exist in challenging and unpredictable human social environments, it becomes clear that robots must be able to understand and predict human behaviors. While the work discussed here is hardly the final say in the matter of how to endow robots with such capabilities, it reveals many of the challenges and suggests some of the strategies necessary to make socially intelligent machines a reality.

13. References

Duda, R.; Hart, P. & Stork, D. (2000). *Pattern Classification*, Wiley-Interscience

Efros, J.; Berg, A.; Morri, G. & Malik, J. (2003). "Recognizing action at a distance," *Intl. Conference on Computer Vision*.

Gopnick, A. & Moore, A. (1994). "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

Hovland, G.; Sikka, P. & McCarragher, B. (1996). “Skill acquisition from human demonstration using a hidden Markov model,” *Int. Conf. Robotics and Automation* (1996), pp. 2706-2711.

Iacobini, M.; Molnar-Szakacs, I.; Gallese, V.; Buccino, G.; Mazziotta, J. & Rizzolatti, G. (2005). ``Grasping the Intentions of Others with One's Own Mirror Neuron System,'' *PLoS Biol* 3(3):e79

Marneffe, M.; MacCartney, B.; & Manning, C. (2006). “Generating Typed Dependency Parses from Phrase Structure Parses,” *LREC*.

Ogawara, K.; Takamatsu, J.; Kimura, H. & Ikeuchi, K. (2002). “Modeling manipulation interactions by hidden Markov models,” Int. Conf. Intelligent Robots and Systems (2002), pp. 1096-1101.

Osuna, E.; Freund, R.; Girosi, F. (1997) “Improved Training Algorithm for Support Vector Machines,” *Proc. Neural Networks in Signal Processing*

Platt, J. (1998). “Fast Training of Support Vector Machines using Sequential Minimal Optimization,” Advances in Kernel Methods - Support Vector Learning. MIT Press 185--208.

Pook, P. & Ballard, D. “Recognizing teleoperating manipulations,” *Int. Conf. Robotics and Automation* (1993), pp. 578-585.

Premack D. & Woodruff, G. (1978). ``Does the chimpanzee have a theory of mind?'' *Behav. Brain Sci.* 1(4) 515-526

L. R. Rabiner, (1989). “A tutorial on hidden-Markov models and selected applications in speech recognition,” in *Proc. IEEE* 77(2)

Tavakkoli, A., Nicolescu, M., Bebis, G. (2006). “Automatic Statistical Object Detection for Visual Surveillance.” *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation* 144--148

Tavakkoli, A.; Kelley, R.; King, C.; Nicolescu, M.; Nicolescu, M. & Bebis, G. (2007). “A Vision-Based Architecture for Intent Recognition,” *Proc. of the International Symposium on Visual Computing*, pp. 173-182

Tax, D., Duin, R. (2004). “Support Vector Data Description.” *Machine Learning* 54. pp. 45-66.

Human-Robot Interaction
Edited by Daisuke Chugo

ISBN 978-953-307-051-3
Hard cover, 288 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Human-robot interaction (HRI) is the study of interactions between people (users) and robots. HRI is multidisciplinary with contributions from the fields of human-computer interaction, artificial intelligence, robotics, speech recognition, and social sciences (psychology, cognitive science, anthropology, and human factors). There has been a great deal of work done in the area of human-robot interaction to understand how a human interacts with a computer. However, there has been very little work done in understanding how people interact with robots. For robots becoming our friends, these studies will be required more and more.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Richard Kelley, Alireza Tavakkoli, Christopher King, Monica Nicolescu and Mircea Nicolescu (2010). Understanding Activities and Intentions for Human-Robot Interaction, Human-Robot Interaction, Daisuke Chugo (Ed.), ISBN: 978-953-307-051-3, InTech, Available from: http://www.intechopen.com/books/human-robot-interaction/understanding-activities-and-intentions-for-human-robot-interaction

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821