SCTAG: A Mildly Context-Sensitive Formalism For Modeling Complex Intentions In Spatially Structured Environments

Peter Kiefer
Laboratory for Semantic Information Technologies
Otto-Friedrich-University Bamberg
96049 Bamberg, Germany

Abstract

The way we represent intentions, behaviors, and the spatial context, is crucial for any approach to mobile intention recognition. Formal grammars are cognitively comprehensible and make expressiveness properties explicit. By adding spatial domain knowledge to a grammar we can reduce parsing ambiguities. We argue that there are a number of mobile intention recognition problems which require the expressiveness of a mildly context-sensitive grammar and discuss Spatially Constrained Tree-Adjoining Grammars.

Introduction

Intention recognition is the problem of inferring a person's intentions to act from observations of that person's behavior[1]. While early research has considered problems like natural language story understanding (Charniak and Goldman 1993), the design of intelligent mobile assistance systems has received much attention recently, e.g. (Liao et al. 2007). We refer to the problem of inferring a mobile person's intentions as *mobile intention recognition*.

Any system that recognizes human intentions needs to have a model of the behaviors and intentions a person can have. Our interest is the representational formalism with which we build our model. In contrast to the probabilistic network based formalisms chosen by (Liao et al. 2007) and other authors, this paper discusses approaches based on formal grammars. By using formal grammars, the trade-off between expressiveness and complexity becomes explicit. For instance, we need at least the expressiveness of a context-free grammar (CFG) if we try to capture intention structures of the form $a^n b^n$. Plans of this form can easily be imagined: $pickItem^n payItem^n$ (paying as many items in a supermarket as picked before), or $enter^n leave^n$ (entering and leaving nested polygons in a structured spatial environment, (Schlieder 2005)). For intention structures with cross-dependencies we need at least a mildly context-sensitive grammar (MCSG) (Geib and Steedman 2007).

Besides the formal expressiveness, the cognitive interpretability of our models becomes crucial if a non-computer scientist should be able to understand, modify, or create them. Formal grammars are generally easy to understand, while at the same time flexible to model domain knowledge across different domains. Domain knowledge includes intentions and behaviors, as well as contextual factors. In mobile intention recognition, the most important type of context is space. The aim of our research is to combine spatial knowledge with formal grammars. One approach in this direction are Spatially Grounded Intentional Systems (SGIS) that enhance CFGs with spatial knowledge (Schlieder 2005).

This paper discusses Spatially Constrained Tree-Adjoining Grammars (SCTAG), a spatial MCSG that allows for modeling of complex relations between intentions and space. We argue that mobile intention recognition problems often bear a complexity that affords for mild context-sensitivity. While we have introduced SCTAG in (Kiefer 2008) from a cognitive point of view, we discuss them here with the emphasis on expressive capabilities. Our use case is an intelligent user interface for the location-based game CityPoker. However, the general idea of SCTAG is likely to be of interest also for other domains where we need to model complex human intentions in space.

The paper is structured as follows: section 2 exemplifies and formally defines the mobile intention recognition problem. In section 3 we explain how space as special form of context can be integrated in CFGs, before we proceed to the SCTAG formalism in section 4. We discuss the possibilities and limitations of SCTAG in the context of related work in section 5, and conclude with an outlook.

The Mobile Intention Recognition Problem

Use Case: Mobile Assistance For CityPoker

In the location-based game CityPoker two players try to optimize their poker hands by finding and changing hidden playing cards in an urban environment. CityPoker is member of a larger class of location-based games, called *Geogames*, which blends strategic reasoning with sportive activity (Schlieder, Kiefer, and Matyas 2006). The game is supported by a mobile assistance system (J2ME on smartphones) with GPS localization.

CityPoker imposes the following partonomial structure on the city: the game board contains five rectangular cache-regions ($\text{CACHE-REGION}_1, \ldots, \text{CACHE-REGION}_5$),

[1] The intention recognition problem is closely related to the plan recognition problem (Schmidt, Sridharan, and Goodson 1978).

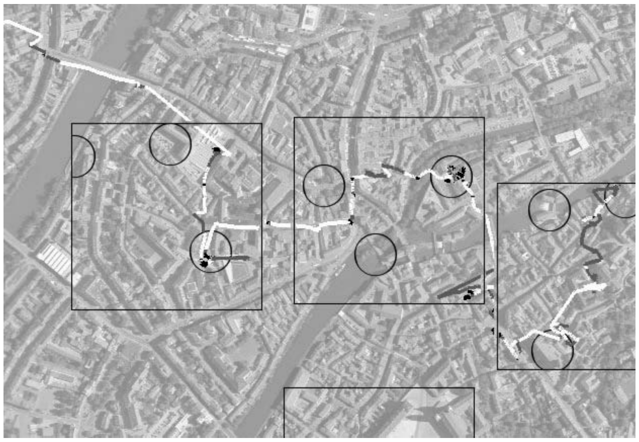

Figure 1: Spatial structure (regions and caches) and motion track of one player in CityPoker

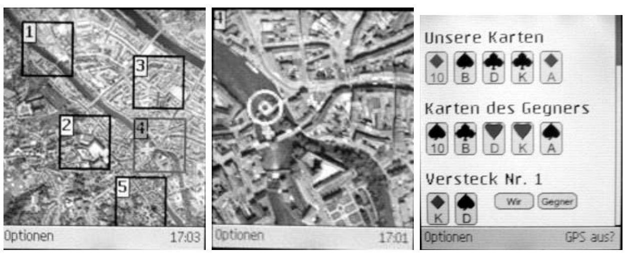

Figure 2: Three examples for information services in CityPoker: overview map, detail map, and game status screen.

each of which contains three circular caches (CACHE$_{1,1}, \ldots,$ CACHE$_{5,3}$). Figure 1 displays only those cache-regions the player entered during the game.[2]

When a player enters a cache-region, the mobile device poses a multiple-choice quiz with three answers. Each answer will lead the player to one of the three potential caches. Arriving at the cache, the device reveals a perceptual hint (e. g. 'the cards are hidden under a green wooden object'). This hint helps the player to locate the exact place of the cards in the circular cache. The wrong answer will lead the player to the wrong cache. After some searching time she will correct her answer and head for another cache. All game actions happen concurrently (i.e. no turn-taking) which puts players under time pressure. After a certain time limit the game ends, and the team with the best poker hand wins.

Players in CityPoker move by bike at high speed which severely restricts the possibilities of interacting with the device. Our goal is to automatically select and present the information service that fits the user's needs most, depending on her behavior. The information services available are maps of different zoom level (*citymap*, *regionmap*, *cachemap*), *perceptual hint*, *game status*, and *quiz* (Fig. 2 displays three of them).

[2]The track was recorded during a CityPoker game on Oct. 30, 2006, played by a team of 4 girls of age between 10 and 14.

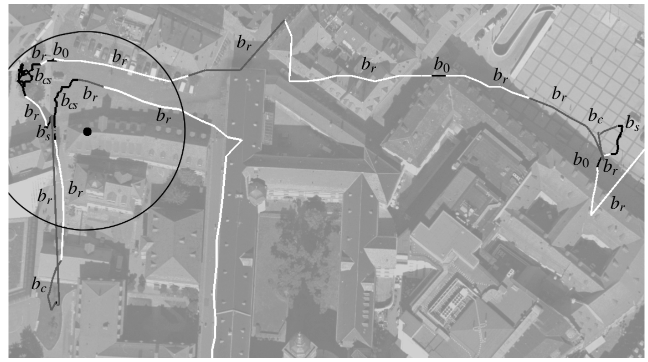

Figure 3: Segmented motion track with classified behavior sequence from a CityPoker game. (The player enters from the right.)

From raw-data to primitive behaviors

As in any intention recognition scenario we need to cross the 'semantic gap between the output of the low-level processes and the high-level inference engines' (Kautz 1991, p.65). That is, the gap between raw GPS data and intentions. A traditional location-based service simply assumes the intention 'get information about some nearby object', given a certain GPS position. This is not appropriate for frequently occurring use cases, as exemplified by the *room-crossing problem* (Schlieder 2005) which consists in deciding whether a mobile user has the intention to cross or visit a certain spatial region (like a museum room). We introduce several layers to cross the gap, and process a stream of (lat/lon)-pairs as follows:

1. *Preprocessing* The quality of the raw GPS data is improved.
2. *Segmentation* The motion track is segmented at the border of regions, and when the spatio-temporal properties of the last n points have changed significantly.
3. *Feature Extraction* Each segment is analyzed and annotated with certain features, like speed and curvature.
4. *Classification* Using these features, each motion segment is classified to one *behavior*. We can use any mapping function from feature vector to behaviors, for instance realized as a decision tree.

The details of the processing hierarchy can be found in (Kiefer and Stein 2008). As output we get a stream of *behaviors*, each annotated with the region of occurrence. The set of possible behaviors depends on the use case. For CityPoker we distinguish the following spatio-temporal behaviors: riding (b_r), curving (b_c), slow curving (b_{cs}), sauntering (b_s), standing (b_0), and add the non-spatio-temporal behavior change card (b_{card}). Refer to Fig. 3 for an example with real data.

Mobile intention recognition

We should say something about the difference between *plans* and *intentions* although an elaborate discussion of this issue is beyond the scope of this paper. In line with common BDI agent literature, we see intentions as 'states of mind' which are directed 'towards some future state of affairs' ((Wooldridge 2000, p.23)). We see 'plans as *recipes* for achieving intentions.' (Wooldridge 2000, p.28).

Mobile intention recognition problems differ in two important ways from the non-mobile problem: First, all observations are ordered temporally (which is not necessarily assumed in general plan recognition research). Second, and most importantly, all observations are annotated with the region of occurrence, i.e. our input is not a behavior sequence $BSeq = b_1, ..., b_n$, but a sequence of spatial behaviors: $BSeq_S$ = (b_1, r_1), ..., (b_n, r_n). The regions r_i are not coordinates but qualitative representations of geographic features, typically polygons like $cache_{5,2}$. Note that $BSeq_S$ has the *spatial continuity* property which means that for every (b_i, r_i) and (b_{i+1}, r_{i+1}) in $BSeq_S$ the two regions r_i and r_{i+1} must either be identical, neighbors (relation *touches* holds), or in a parent-child relation. In other words: an agent cannot beam herself.

The mentioned spatial relations (*identical*, *touches*, *childOf*) are only some that may hold between geographic features (Egenhofer and Franzosa 1991). Other include *overlap* or directional relations like *northOf*. In our mobile intention recognition problems we have a *geo model* that stores these relations between regions. In CityPoker, for instance, we have a geo model of type *partonomy* which means our regions are structured hierarchically with *childOf* relations. Properties of these relations, like transitivity, inverse relation, and others, are not interesting in the following. We assume spatial relations to be precomputed and stored in a look-up table.

Formally, we end up with the following definition of the mobile intention recognition problem: Let I denote a set of intentions, and B a set of behaviors, both sets being finite. Let $G = (R, RT, SR)$ denote a geo model with a set of regions R, a set of spatial relation types RT, and a set of spatial relations $SR \subseteq RT \times R \times R$. $BSeq_S$ = (b_1, r_1), ..., (b_n, r_n) is a spatial behavior sequence for which the spatial continuity property holds (which we do not define formally here). The mobile intention recognition problem consists in finding an intention sequence $ISeq = i_1, ..., i_n$ that explains $BSeq_S$.

'Spatialized' grammars

Consider modeling mobile intention recognition problems with a CFG. We would use spatialized intentions as nonterminals, and spatialized behaviors as terminals. Our rules would be of the following form (with $s_j \in I \cup B$): $(i, r) \rightarrow (s_1, r_1)...(s_k, r_k)$. By running an incremental parsing algorithm on $BSeq_S$ we could determine the *current intention* as the parent of the last terminal. However, writing arbitrary rules for pairs of (behavior,region) is not sensible: the model designer could disobey the spatial continuity property. In addition, there are typically rules that apply to several regions, or rules that are inherited from parent regions which would mean an overflow of rules.

SGIS (Schlieder 2005), a CFG enhanced with spatial knowledge, overcomes these shortcomings. An SGIS rule is formalized for behaviors and annotated with the regions in which it applies: $i \rightarrow s_1...s_k | \{r_1, ..., r_m\}$. A rule is applicable if all symbols are located in any of the annotated regions,

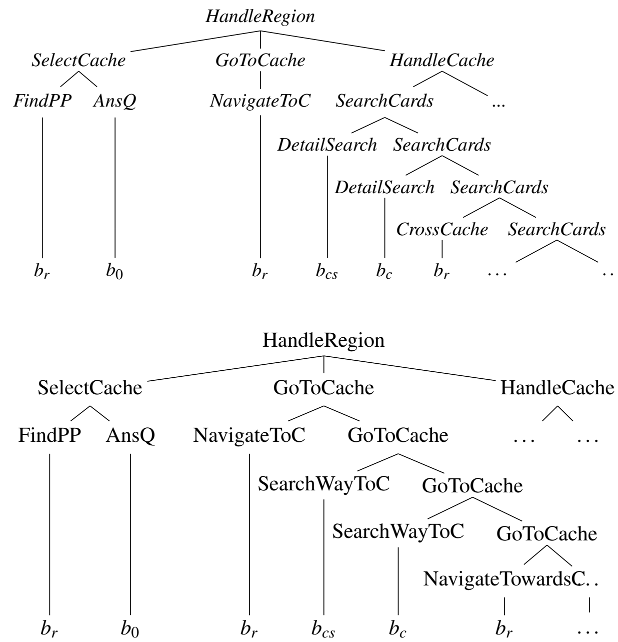

Figure 4: Parsing ambiguity if we had no spatial knowledge (see track from Fig. 3). Through spatial disambiguation in SGIS we can decide that the bottom parse tree is correct.

or in any of the (transitive) child regions. This is not only a notational convenience, but also resembles typical human activities in space: something the agent can strive for in the parent region (ChangeCard in the gameboard region) is also a possible intention in any of the child regions. Even if the agent crosses the child region accidentally the higher-level intention is applicable. The added spatial knowledge makes SGIS more efficient than a CFG without any spatial information.

Consider the behavior sequence from Fig. 3 part of which we try to parse in Fig. 4. With a certain set of CFG rules, there are at least two interpretations: either the user's current intention at the sixth behavior is *CrossCache* (the user has reached the cache and is looking for the cards) or *NavigateTowardsCache* (the user is still searching her way to the cache). Without spatial knowledge, both interpretations are possible. In an SGIS, the rule $HandleCache \rightarrow SearchCards\ '...'$ is annotated with all caches $\text{CACHE}_{1,1}, \ldots, \text{CACHE}_{5,3}$ and thus not applicable at the sixth behavior where the user is still outside of the cache. Thus, spatial knowledge reduces ambiguity that would occur in non-spatial intention recognition.

Spatially Constrained Tree-Adjoining Grammars

SCTAG port the idea of spatial disambiguation to Tree-Adjoining Grammars, a mildly context-sensitive formalism from NLP. Compared to SGIS, the SCTAG formalism supports cross-dependencies and other spatial relations than *childOf*.

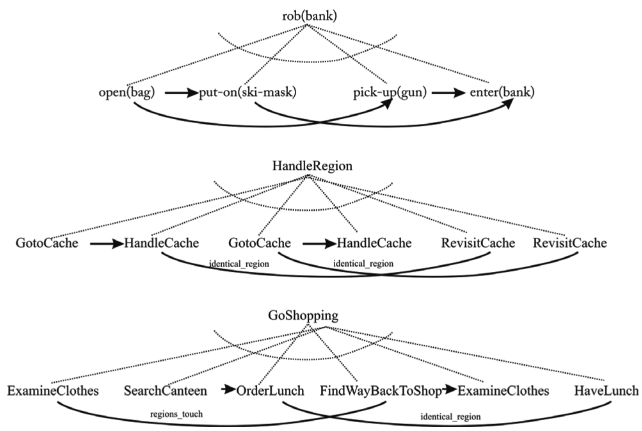

Figure 5: Crossing dependencies in a non-mobile plan recognition scenario (from (Geib and Steedman 2007), top), and in two mobile intention recognition scenarios (CityPoker and shopping center).

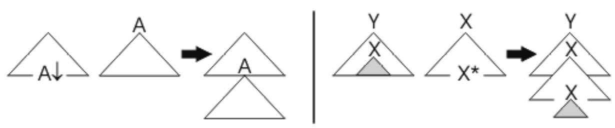

Figure 6: Substitution (left) and adjoining (right) on a TAG (taken from (Joshi and Schabes 1997, Fig. 2.2))

Crossing Dependencies

The need for modeling non-mobile crossing dependencies in general plan recognition has been discussed in (Geib and Steedman 2007), see Fig. 5, top. In this example, entering the bank depends on two preconditions (putting on a ski mask and picking up the gun) while each of these two again depends on opening the bag. In which way we ever try to order the two middle actions, we will not resolve the cross-dependency. These kind of dependencies, caused by pre- and post-conditions, may certainly also occur in a mobile setting.

However, we are more interested in the specific features of space: here, we often find dependencies on the regions of the input. For instance, the intention of visiting a region the second time will occur in regions with a certain spatial relation to the original intention. In our example CityPoker, we can say that *RevisitCache* must occur in the *identical* region as *HandleCache*. In other settings we might want to express *FindWayBackToX* as an intention occurring in the region that *touches* the original region. In general, we want to formulate constraints with arbitrary spatial relations from our relation types RT. These dependencies may also cross, as displayed in Fig. 5.

SCTAG

MCSG are a class of formal grammars with common properties (Vijay-Shanker and Weir 1994). Their expressiveness falls between CFGs and context-sensitive grammars, and they support certain kinds of dependencies, including

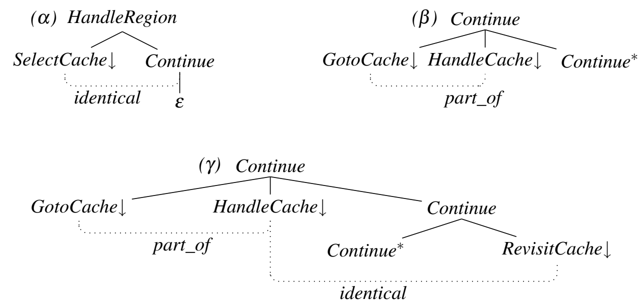

Figure 7: Initial tree (α) and auxiliary trees (β and γ) in an SCTAG for CityPoker.

crossed and nested dependenciess. They are polynomially parsable and thus especially attractive for mobile intention recognition. *Tree-Adjoining Grammars* (TAG) are a MCSG with an especially comprehensible way of modeling dependencies. The fundamental difference to CFGs is that TAGs operate on trees, and not on strings. A good introduction to TAG is given by Joshi and Schabes in (Joshi and Schabes 1997). A *TAG = (NT, T, IT, AT, S)* consists of non-terminals, terminals, initial trees, auxiliary trees, and a starting symbol.

The two operations *substitution* and *adjoining* define which manipulations we can perform on initial, auxiliary, and derived trees to get the corresponding tree language (see Fig. 6). Substitution replaces a substitution node (marked with an arrow) with an other tree headed by the same non-terminal. It is the adjoining operation that makes TAGs unique: we can adjoin an auxiliary tree labeled with X into an interior node of another tree with the same label. This operation works as follows: (1) we remove the part of the tree which is headed by the interior node, (2) replace it by the auxiliary tree, and (3) attach the partial tree which was removed in step 1 at the foot node. This leads to a 'wrapping' of parts of the old tree around the new one. Using this operation we can create cross-dependencies.

The definition of Spatially Constrained Tree-Adjoining Grammars ports the idea of spatial constraints to TAGs: $SCTAG = (TAG, G, GC, NLC)$, where

- *TAG = (I, B, IT, AT, S)*, defined over intentions I, and behaviors B.
- $GC \subseteq (IT \cup AT) \times R$ is a set of grounding constraints which restrict the applicability of an elementary tree to a region (as in SGIS).
- *NLC* is a set of spatial non-local constraints. Each constraint has a type from *RT* and is defined for two nodes in one tree from $IT \cup AT$ (these are the complex spatial constraints we denote with dotted lines).

Figure 7 shows how we can solve the 'ReturnToCache' problem with adjoining in an SCTAG: we start with one initial and one auxiliary tree (α and β). A complete grammar would certainly contain more elementary trees, e.g. one for substituting the non-terminal *SelectCache*. We first adjoin β in the *Continue* node of α. We take the resulting tree, perform another adjoining operation with β, and get the tree displayed in Fig. 8.

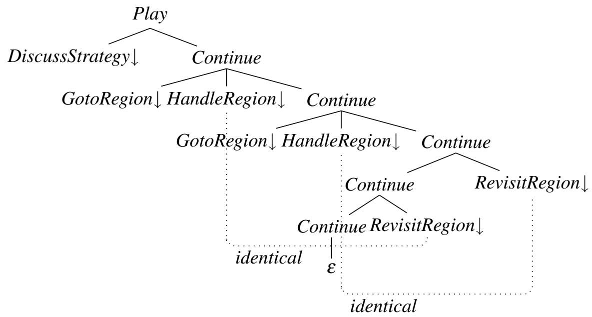

Figure 8: Adjoining in an SCTAG can lead to cross-dependencies of constraints.

Parsing of SCTAG

For parsing a spatially constrained grammar, we modify existing parsing algorithms. An algorithm for parsing TAGs, based on the Cocke-Younger-Kasami algorithm, was proposed in (Vijay-Shanker and Joshi 1985), with a polynomial worst and average case complexity. Unfortunately, this complexity is $O(n^6)$ and thus quite high. Joshi presents a TAG parser that adopts the idea of Earley and improves the average case complexity (Joshi and Schabes 1997).

We build the parser for SCTAG on these Earley-like parsers. Earley parsers work on a chart in which the elementary constructs of the grammar are kept, production rules for CFGs, trees for TAGs. A dot in each of these chart entries marks the position up to which this construct has been recognized. In Joshi's parser the 'Earley dot' traverses trees and not Strings. Earley parsers work in three steps: scan, predict, and complete. The TAG parser has a fourth operation, called 'adjoin', to handle this additional operation.

Our point is that adding spatial constraints to such a parser will not make it slower but faster. The reason is that spatial constraints give us more predictive information. 'Any algorithm should have enough information to know which tokens are to be expected after a given left context' (Joshi and Schabes 1997, p.36). Knowing the spatial context of left-hand terminals we can throw away those hypotheses that are not consistent with the spatial constraints. We add this step after each scan operation.

Discussion and Related Work

We have discussed the expressiveness of SCTAG for mobile intention recognition. We have implicitly assumed that we need at least the expressiveness of a CFG. However, there are also scenarios where formalisms that build on finite state machines (or Markov models in the probabilistic case) are sufficiently expressive for making sense of an agent's behavior, like moving between different places in a city (Ashbrook and Starner 2002), or on a car-park (Dee and Hogg 2004). We have also not considered cognitively motivated agent architectures (Rao and Georgeff 1991) or CSG, for we cannot solve intention recognition efficiently with them.

In an SCTAG the applicability of a production rule depends on the region the agent is currently in, and the regions she has been to before, which we can see as a state of the environment. A similar idea is followed by Probabilistic State Dependent Grammars (PSDG, (Pynadath and Wellman 2000)): here the probability of a production is dependent on a global state variable. The domain of the state variable is not predefined but may be just anything, depending on the use case. By choosing the set of regions R as domain we can create a dependency of the current region. By choosing the infinite cartesian product of R, we can express a dependency of the region history. This leads to an explosion of the state space. Pynadath and Wellman restrain themselves to examples with finite domains, and say that for more general languages 'although inference [...] is possible, it is impractical in general' (p. 4). Note that it is even not enough to make a production dependent on the region history to reproduce the full expressiveness of SCTAG, because the productions that were chosen along the region history are also relevant. In other words, when trying to express an SCTAG as PSDG, we are aiming at nothing less than turning an MCSG- into a CFG-model, by encoding all the mild context-sensitiveness into one state. While this is interesting from a theoretical perspective[3], practically it will lead to state space explosion, and cognitively little appealing models.

An approach for intention recognition ('policy recognition') based on probabilistic networks is the Abstract Hidden Markov Memory Model (AHM*E*M) (Bui 2003). In an AHM*E*M the choice of the next policy depends on two factors: the internal memory state of the current policy, and a global state variable. The internal memory resembles the structure of grammar rules (a policy may be decomposed into a sequence of other policies). As in PSDG, region-dependencies would need to be captured with the global state variable which again leads to state space explosion.

In contrast to PSDG and AHM*E*M, our 'spatialized' grammars do not consider reasoning under uncertainty. We are aware that, in general, the literature on intention recognition agrees that probabilistic reasoning is necessary for any realistic use case, or in the words of Charniak and Goldman: 'we doubt that any model that does not incorporate some theory of reasoning under uncertainty can be adequate'(Charniak and Goldman 1993). We can find more examples in the related field of activity recognition where probabilistic grammars are used to model expectations on a higher level, see (Minnen, Essa, and Starner 2003). Even with spatial disambiguation there may be two or more possible parse trees for one behavior sequence. Currently, we approach this problem pragmatically: First, we are satisfied with recognizing the correct intention, not necessarily the correct parse tree (intention recognition vs. plan recognition). If two possible parse trees lead to the same intention we are not interested in disambiguating the two trees. Second, if two intentions are mapped to the same information service we also need no disambiguation for our mobile assistance use case. Third, if more than one information service is proposed we simply present one of them and ease the access to the other(s) with a one-click interface.

The idea of integrating a spatial representation into inten-

[3]The automaton related to TAG is the Embedded Pushdown Automaton which works on a stacks of stack (Joshi and Schabes 1997)

tion recognition is not completely new: the simplest spatial model used in many approaches consists of a number of points of interest with circular or polygonal areas around them (Dee and Hogg 2004). Others add a street network to these locations (Liao et al. 2007), use spatial tessellation (Gottfried and Witte 2007), or formalize space with Spatial Conceptual Maps (Samaan and Karmouch 2005). To our knowledge, SCTAG are the first formalism that includes general spatial knowledge in terms of polygons that have spatial relations of various kinds.

The quality of our intention recognition relies on a good preprocessing. Converting a motion track into a qualitative representation has been done by a number of researchers, for instance (Musto et al. 2000). The authors also compare a number of approaches to generalization. For the classification of segments in Fig. 3 we used a simple decision tree. The set of behavior types we are interested in was chosen manually. An automatic detection of motion patterns is the concern of the spatio-temporal data mining community, see e.g. (Laube, van Krefeld, and Imfeld 2004).

Outlook

Our future research will be concerned with a probabilistic variant of SCTAG. Here, one main challenge is certainly the inference mechanism. The other challenge consists in extracting probabilities from behavioral data, especially if only few data exist. We will also consider extending SCTAG with temporal constraints, like 'the duration between these two intentions may not be longer than a certain Δt'.

References

Ashbrook, D., and Starner, T. 2002. Learning significant locations and predicting user movement with GPS. In *Proc. of IEEE 6th Int. Symposium on Wearable Computing*.

Bui, H. H. 2003. A general model for online probabilistic plan recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Charniak, E., and Goldman, R. P. 1993. A bayesian model of plan recognition. *Artificial Intelligence* 64(1):53–79.

Dee, H., and Hogg, D. 2004. Detecting inexplicable behaviour. In *Proc. of the British Machine Vision Conference*, 477–486. The British Machine Vision Association.

Egenhofer, M. J., and Franzosa, R. D. 1991. Point-set topological relations. *International Journal of Geographical Information Systems* 5(2):161–174.

Geib, C. W., and Steedman, M. 2007. On natural language processing and plan recognition. In *Proc. of the 20th Int. Joint Conference on Artificial Intelligence*, 1612–1617.

Gottfried, B., and Witte, J. 2007. Representing spatial activities by spatially contextualised motion patterns. In Lakemeyer, G., ed., *RoboCup 2007, LNAI 4434*, 329–336. Springer.

Joshi, A. K., and Schabes, Y. 1997. Tree-adjoining grammars. In Rozenberg, G., and Salomaa, A., eds., *Handbook of Formal Languages*, volume 3. Berlin, New York: Springer. 69–124.

Kautz, H. 1991. *A Formal Theory of Plan Recognition and its Implementation*. Morgan Kaufmann Publishers. 69–126.

Kiefer, P., and Stein, K. 2008. A framework for mobile intention recognition in spatially structured environments. In *2nd Workshop on Behavior Monitoring and Interpretation (BMI08)*, 28–41.

Kiefer, P. 2008. Spatially constrained grammars for mobile intention recognition. In Freksa, C.; Newcombe, N. S.; Gärdenfors, P.; and Wölfl, S., eds., *Spatial Cognition VI*, 361–377. Berlin Heidelberg: Springer (LNAI 5248).

Laube, P.; van Krefeld, M.; and Imfeld, S. 2004. Finding remo - detecting relative motion patterns in geospatial lifelines. In *Developments in Spatial Data Handling, 11th Int. Symposium on Spatial Data Handling*, 201–215.

Liao, L.; Patterson, D. J.; Fox, D.; and Kautz, H. 2007. Learning and inferring transportation routines. *Artificial Intelligence* 171(5-6):311–331.

Minnen, D.; Essa, I.; and Starner, T. 2003. Expectation grammars: leveraging high-level expectations for activity recognition. In *Proc. Computer Vision and Pattern Recognition*, volume 2, 626–632.

Musto, A.; Stein, K.; Eisenkolb, A.; Röfer, T.; Brauer, W.; and Schill, K. 2000. From motion observation to qualitative motion representation. In *Spatial Cognition II, LNCS 1849*, 115–126. London, UK: Springer.

Pynadath, D. V., and Wellman, M. P. 2000. Probabilistic state-dependent grammars for plan recognition. In *Proc. of the 16th Conference on Uncertainty in AI*, 507–514.

Rao, A., and Georgeff, M. 1991. Modeling rational agents within a bdi-architecture. In *Proc. of Knowledge Representation and Reasoning (KRR 1991)*, 473–484. Morgan Kaufmann Publishers.

Samaan, N., and Karmouch, A. 2005. A mobility prediction architecture based on contextual knowledge and spatial conceptual maps. *IEEE Transactions on Mobile Computing* 4(6):537–551.

Schlieder, C.; Kiefer, P.; and Matyas, S. 2006. Geogames - designing location-based games from classic board games. *IEEE Intelligent Systems* 21(5):40–46.

Schlieder, C. 2005. Representing the meaning of spatial behavior by spatially grounded intentional systems. In *GeoSpatial Semantics, 1st Int. Conference*, volume 3799 of *LNCS*, 30–44. Springer.

Schmidt, C.; Sridharan, N.; and Goodson, J. 1978. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence* 11(1-2):45–83.

Vijay-Shanker, K., and Joshi, A. K. 1985. Some computational properties of tree adjoining grammars. In *Meeting of the Association for Computational Linguistics*, 82 – 93.

Vijay-Shanker, K., and Weir, D. 1994. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory* 27(6):511–546.

Wooldridge, M. 2000. *Reasoning About Rational Agents*. MIT Press.