Proc 5th Europ Conf on Mobile Robots ECMR 2011, 7-9 Sep 2011, Örebro, Sweden pp. 7-12

On Process Recognition by Logical Inference

Arne Kreutzmann Immo Colonius Lutz Frommberger Frank Dylla Christian Freksa Diedrich Wolter
SFB/TR 8 Spatial Cognition, University of Bremen, Germany

***Abstract*— The ability to recognize and to understand processes allows a robot operating in a dynamic environment to rationally respond to dynamic changes. In this paper we demonstrate how a mobile robot can recognize storage processes in a warehouse environment, solely using perception data and an abstract specification of the processes. We specify processes symbolically in linear temporal logic (LTL) and pose process recognition as a model verification problem. The key feature of our logic based approach is its ability to infer missing pieces of information by logic-based reasoning. The evaluation demonstrates that this approach is able to reconstruct histories of good movements in a lab-simulated warehouse.**

***Index Terms*— plan recognition, temporal logic, spatio-temporal reasoning**

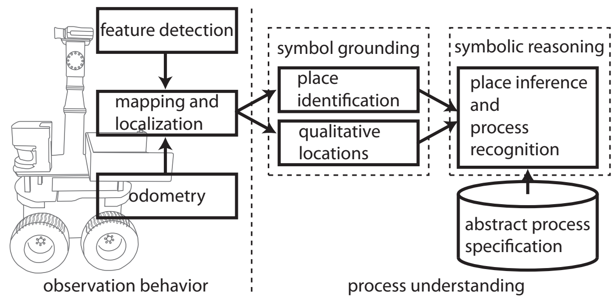


Fig. 1. Conceptual overview of our software architecture

I. Introduction

Mastering dynamic environments is a demanding challenge in autonomous robotics. It involves recognition and understanding processes in the environment [7]. Recent advances in simultaneous localization and mapping (SLAM) [20, 21, 22] build the basis for sophisticated navigation in dynamic environments, but but our aim of *understanding* processes goes beyond navigation.

In this paper we indicate how the problem of recognizing processes can be tackled on a conceptual level in the domain of warehouse logistics. In a warehouse, there is a constant flow of goods which are moved through space, establishing functional zones that are connected with certain types of storage processes (for example, admission of goods into a warehouse makes use of a buffer zone to temporarily store goods for quality assurance). Knowing about the in-warehouse processes and their whereabouts enables warehouse optimization. Hildebrandt et. al. argue for using autonomous robots as a minimally invasive means to observe in-warehouse processes [10]. However, the sensory system of the robot provides uncertain and incomplete knowledge about the environment and the observed spatio-temporal patterns. Thus the challenge is to interpret the observations sensibly.

Many approaches to process recognition rely on statistical data to train probabilistic classifiers such as Markov networks [6, 13], Bayesian networks [23], or supervised learning [5]. Approaches based on statistical data perform very well in terms of recognition rate, but, aside from the need for training, they do not support flexible queries about processes and they have they have to be re-trained if new elements or processes are introduced in the domain. Symbolic approaches have none of these downsides, but require a model of the observable processes, which is given in our environment. Additionally, a well constructed model allows for efficient use of heuristics to speed up query processes[8]. Usually, symbolic approaches are used to tackle plan recognition, which is closely related to process recognition—see [2, 3] for an overview.

In the following we present a logic-based approach that allows us to recognize activities purely from qualitative process descriptions without prior training. By integrating and abstracting sensory information we are able to answer queries about observed spatio-temporal activities (such as “How often have goods been relocated within the storage zone?”) as well as about regions in space (e.g., “Which areas in the warehouse have been used as a buffer zone?”). Answering such queries is an important step towards logistic optimization. The contribution of this paper is to demonstrate how processes and their whereabouts can be inferred in a previously unknown environment.

Referring to the decomposition of process detection by Yang [23], we propose a multi-step approach to get from low-level sensory observations to high-level symbolic representations (see Fig. 1). In our scenario, a robot performs a surveillance task in the warehouse. Object recognition is outside the scope of this paper, but in many logistics scenarios goods can easily be identified by unique labels attached to them (such as barcodes or RFID tags). Thus, we assume that the robot is able to uniquely identify goods in the warehouse. The integration of position estimates for the goods in itself presents a feature-based SLAM problem. Uncertain and incomplete position estimates of entities gathered by a probabilistic mapping procedure must be transferred into a symbolic representation in a symbol grounding process to allow for high-level descriptions of the system dynamics. What has been an uncertain *position estimate* in the mapping process must become a stable qualitative notion of *location*. Based on correspondence of features and locations in time, we are able to specify processes of interest in an abstract formal language and, in a third step, tackle the process recognition problem by model verification.

The formal language we choose to formalize processes

This paper presents work done in the project R3-[Q-Shape] of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition. Funding by the German Research Foundation (DFG) is gratefully acknowledged.

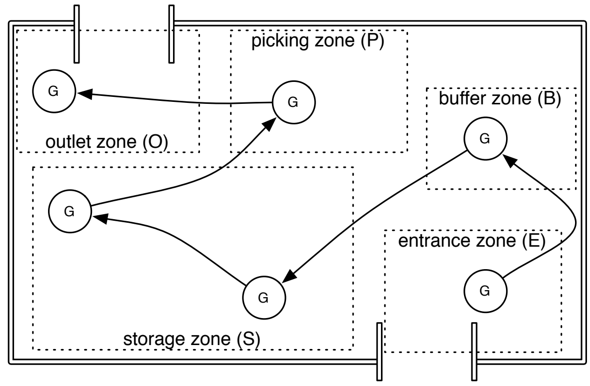


Fig. 2. A warehouse and its functional zones.

and to state queries is *linear temporal logic* (LTL) [17, see Sect. III-A]. LTL was proposed earlier as a tool for mobile robotics [1], especially for robot motion planning from high-level specifications [11, 18]. Recently, this approach has also been applied to real robotic systems [12]. In the domain of smart environments, an approach to process detection by LTL model verification has been presented in [14]. LTL not only allows for queries about processes, but also about spatial relations of regions. This approach covers a wide range of reasoning tasks adequately. In particular, it allows us to query the occurrence of processes operating on spatial regions and the concrete whereabouts of those regions at the same time in one and the same reasoning process.

II. The Warehouse Scenario

We address the problem of understanding so-called *chaotic* or *random-storage warehouses*, characterized by a lacking predefined spatial structure, that is, there is no fixed assignment of storage locations to specific goods. Thus, storage processes are solely in the responsibility of the warehouse operators and basically not predictable: goods of the same type may be distributed over various locations and no data base keeps track of these locations. This makes it a hard problem for people aiming at understanding the internal storage processes.

On a conceptual level, storage processes are defined by a unique pattern [19]: On their way into and out of the warehouse, goods are (temporarily) placed into *functional zones* which serve specific purposes (see Fig. 2). All goods arrive in the *entrance zone* (E). From there, they are picked up and temporarily moved to a *buffer zone* (B) before they are finally stored in the *storage zone* (S). Within the *storage zone* redistribution of goods can occur arbitrarily often. When taking out goods, they are first moved to the *picking zone* (P) from where they are taken to the *outlet zone* (O), before being placed on a truck.

A mobile robot observing such a warehouse is not able to directly perceive these zones, as they are not marked. For all zones we know that they exist (that is, that such regions are used within the storage operations), but not their concrete spatial extents or their number of occurrences, as they appear as a result of dynamic in-warehouse storage processes. The robot can detect and identify goods, and estimate their position. So when observing this kind of environment, we face the challenge that for detecting concrete storage processes we rely on the existence of certain zones, but we do not know their whereabouts.

III. In-Warehouse Process Detection with Linear Temporal Logic

To interpret raw sensory data such that we achieve a symbolic representation of the processes of interest, we first introduce linear temporal logic and the axiomatization of our domain. All queries are stated as LTL formulas and can be answered by model verification. Following this, we describe the symbolic grounding. Then, we specify the in-warehouse processes in linear temporal logic and demonstrate the inference process by an example.

A. Linear Temporal Logic (LTL)

LTL [17] is a modal logic that extends propositional logic by a sequential notion of time. A formula ϕ in LTL is defined over a finite set of propositions with a set of the usual logic operators ($\wedge$, $\vee$,$\neg$, $\rightarrow$). The temporal component is established by an accessibility relation R that connects worlds (or states) and a set of modal operators, of which we use the following:

- $\circ\phi$ – next. A formula ϕ holds in in the following world
- $\Box\phi$ – always. A formula ϕ holds now and in all future worlds
- $\Diamond\phi$ – eventually. ϕ will hold in some world in the future ($\Diamond\phi \leftrightarrow \neg\Box\neg\phi$)

B. Axiomatizing the Warehouse Scenario

1) Propositions: We define the propositions that model the desired processes in our logic with the help of the following atomic observables:

- a set $\mathcal{G} = \{G_1, \dots, G_n\}$ of uniquely identifiable goods
- a set $\mathcal{L} = \{L_1, \dots, L_m\}$ of locations in space at which goods have been perceived by the robot
- a set $\mathcal{Z} = \{E, B, S, P, O\}$ of functional zones as described in Sect. II.

The following atoms need to be defined over $\mathcal{G}$, such that we obtain a finite set of atoms, $\mathcal{L}$, and $\mathcal{Z}$:

- $\text{at}(G, L)$ – holds iff a good G is known to be at location L
- $\text{in}(L, Z)$ – holds iff a location L lies within zone Z
- $\text{close}(L_1, L_2)$ – holds iff two locations L_1, L_2 are close to one another

2) Axioms: Based on constraints of space and general knowledge about our domain, we axiomatize our domain. One constraint is that we disregard continuous motion and therefore only deal with snapshots of the world. This means that all observed goods are temporarily fixed at their positions.

- A good G can only be at one location at a time. We introduce the following axioms for all $G \in \mathcal{G}$ and $L_i, L_j \in \mathcal{L}$, $i \neq j$:

$$A1_{G,L_i,L_j} = \Box\neg\big(\text{at}(G, L_i) \wedge \text{at}(G, L_j)\big) \qquad (1)$$

- Any object is always located within a zone $Z \in \mathcal{Z}$. We have for all $G \in \mathcal{G}$ and $L \in \mathcal{L}$:

$$A2_{G,L} = \Box\big(\text{at}(G,L) \rightarrow \bigvee_{Z\in\mathcal{Z}} \text{in}(L,Z)\big) \tag{2}$$

- Locations in different zones are not close to each other, that is, zones are at least some minimum distance apart. We have for all $Z_k, Z_l \in \mathcal{Z}$ ($k \neq l$) and $L_i, L_j \in \mathcal{L}$ ($i \neq j$):

$$\begin{aligned} A3_{L_i,L_j,Z_k,Z_l} =& \Box\big(\text{in}(L_i,Z_k) \wedge \text{in}(L_j,Z_l) \\ & \rightarrow \neg\text{close}(L_i,L_j)\big) \end{aligned} \tag{3}$$

- Zones are static. We have for all $Z_k, Z_l \in \mathcal{Z}$ ($k \neq l$) and $L \in \mathcal{L}$:

$$A4_{L,Z_k,Z_l} = \text{in}(L,Z_k) \rightarrow \neg\Diamond\text{in}(L,Z_l) \tag{4}$$

A set $\mathcal{A}$ subsumes all axioms (1) – (4).

C. Grounding Symbols

So far, we have formal descriptions of the high-level in-warehouse observables on one hand, and sensory perceptions from the robot, on the other hand. These need to be connected to each other in order to perform reasoning on real world data. That is, we need to transform the sensory information to our logical propositions at(G, L), close(L_1, L_2), and in(L, Z).

Mapping a perceived good to a symbol G is trivial in this task due to the unique identifiers. However, for the good's location we will only have an uncertain *position estimate* $(x, y) \in \mathbb{R}^2$ for the entity observed from the mapping process. These estimates are subject to noise and thus will vary over time although the observed object remains static. A *location* is a qualitative abstraction from positional measurements that abstracts from uncertainty emerging from sensory perceptions and the mapping process. Therefore, we need to transform position estimates to a discrete and finite set of symbols, i.e., to subsume similar or comparable positions. This transformation is a function $f : \mathbb{R}^2 \rightarrow \mathcal{L}$, that is, every position estimate is mapped to a single location (see Axiom (1)). To this end, a clustering method can be applied to map estimates to a set of prototypical positions—the locations (see Section IV-B). We ground close(L_1, L_2) by applying a metric and checking whether the distance between L_1 and L_2 is below a certain threshold.

To ground in(L, Z), we need to identify the functional zones in the warehouse. These zones are constituted by sets of locations. For zones Z whose extents are known a-priori by introducing the respective in-atoms the corresponding locations $\mathcal{L}_Z \subseteq \mathcal{L}$ can be assigned directly. All remaining locations $L_i \in \mathcal{L}\backslash\mathcal{L}_\mathcal{Z}$ are known to be not a part of Z, i.e., $\neg$in(L_i, Z), but (according to (2)) must be part of one of the other zones: in(L_i, Z') with $Z' \in \mathcal{Z}\backslash Z$.

In addition to the axioms $\mathcal{A}$, the propositions close and in are persistent over all worlds. The set

$$\mathcal{B} = \mathcal{A} \cup \bigcup_{L_i,L_j\in\mathcal{L}} \text{close}(L_i,L_j) \cup \bigcup_{L\in\mathcal{L},Z\in\mathcal{Z}} \text{in}(L,Z) \tag{5}$$

is called *background knowledge*. The only proposition that changes over different worlds is at(G, L). We traverse through the time steps t and map all goods G_i with their position estimates (x_i, y_i) to corresponding observations obs(t, G_i, L_j) that assign that G_i has been observed at L_j at time step t. This yields a series of sets of observations $\mathcal{O}_t = \bigcup_{G_i\in\mathcal{G}} \text{obs}(t, G, L_j)$ over time. A new world is established as soon as our observations change, that is, $\mathcal{O}_{t+1} \neq \mathcal{O}_t$. Then, from obs$(t, G_i, L_j)$ follows at(G_i, L_j), and the new world consist of $\mathcal{B} \cup \bigcup_i \text{at}(G_i, L_j)$.

D. In-Warehouse Processes

We now formalize the in-warehouse processes Admission, Take-out, and Redistribution:

- *Admission* – a good G is delivered to the warehouse's entrance zone E and moved to the storage zone S via the buffer zone B. For all $G \in \mathcal{G}$ and $L_i, L_j, L_k \in \mathcal{L}$:

$$\begin{aligned} &\text{Admission}_{G,L_i,L_j,L_k} = \text{at}(G,L_i) \wedge \text{in}(L_i,E) \rightarrow \\ &\Diamond\Big(\text{at}(G,L_j) \wedge \text{in}(L_j,B) \rightarrow \Diamond\big(\text{at}(G,L_k) \wedge \text{in}(L_k,S)\big)\Big) \end{aligned} \tag{6}$$

- *Take-out* – a good G is moved from the storage zone S to the outlet zone O via a picking zone P. For all $G \in \mathcal{G}$ and $L_i, L_j, L_k \in \mathcal{L}$:

$$\begin{aligned} &\text{Takeout}_{G,L_i,L_j,L_k} = \text{at}(G,L_i) \wedge \text{in}(L_i,S) \rightarrow \\ &\Diamond\Big(\text{at}(G,L_j) \wedge \text{in}(L_j,P) \rightarrow \Diamond\big(\text{at}(G,L_k) \wedge \text{in}(L_k,O)\big)\Big) \end{aligned} \tag{7}$$

- *Redistribution* – a good G is moved within the storage zone S. For all $G \in \mathcal{G}$ and $L_i, L_j \in \mathcal{L}$, $i \neq j$:

$$\begin{aligned} &\text{Redistribution}_{G,L_i,L_j} = \text{at}(G,L_i) \wedge \text{in}(L_i,S) \rightarrow \\ &\Diamond\big(\text{at}(G,L_j) \wedge \text{in}(L_j,S)\big) \end{aligned} \tag{8}$$

Process detection can be posed as a model checking problem: An in-warehouse process is detected when we can find a model (based on the sensory observations from the robot) that satisfies the corresponding formula. The *history* of a good is the chain of processes that the good is part of and can also be stated as a formula. A history for a good would be *admission*, zero or more *redistributions* and its *takeout*.

E. Example

A good G entered the warehouse and was stored in the entrance zone E at position L_1 at time t_0. At t_1, it was moved to a location L_2 and at t_2 it was moved to L_3. All these locations are not close to one another. Let us assume that we observe the following from this process: We perceived G to be at L_1 at t_0, at L_2 at t_1 and at L_3 at t_4. See Fig. 3 for a depiction and the logical interpretations—to ease understanding the worlds are labeled just like the time points.

These observations constitute a model that satisfies (6), such that the observed process is an admission, starting in world t_1 and ending in world t_4, and also deduces that location L_2 is in the buffer zone and L_3 is in of the storage zone. Note that deduced start and end times differ from the real admission times: While the admission took place from t_0 to t_3, we detect it from observations t_1 to t_4; this is due to incomplete observation of the world.

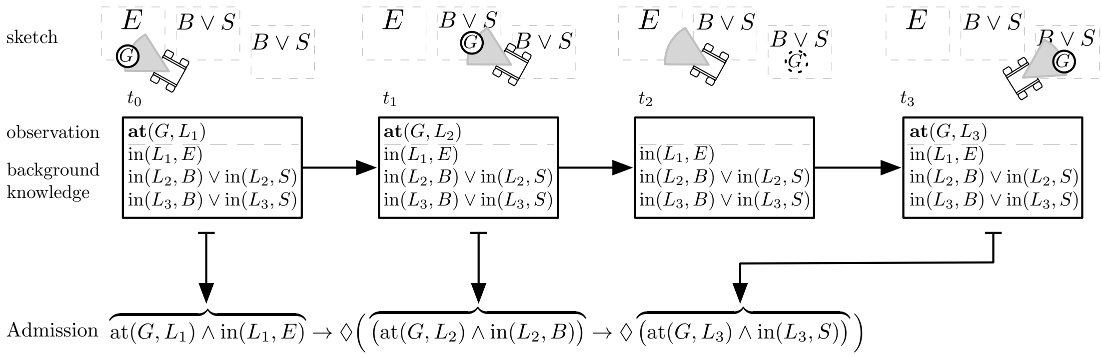


Fig. 3. Example: Model verification for an admission process of good G (only the relevant assertions for each world $t_{0\ldots3}$ are shown). $\mathrm{in}(L_1, E)$ is background knowledge, also it is known that locations L_2 and L_3 are either part of the buffer zone (B) or the storage zone (S) but not close to one another so that they cannot belong to the same zone. From this admission refined knowledge about the buffer and storage zone can be inferred: $\mathrm{in}(L_2, B) \wedge \mathrm{in}(L_3, S)$.

IV. Implementation

A. Mapping of Positions of Goods

We use visual tags to represent our observable features. To ease the evaluation, some tags are known to be static throughout the experiments. This allows the map constructed by the robot to be easily aligned with the ground-truth for evaluation. The positions of the tags relative to the camera are estimated using the tag detection routine provided by the ARToolkit software library[1], for which we determined a measurement model. Positions of detected tags with a sufficient quality as well as odometry of the robot are fed into the TreeMap SLAM algorithm [9]. In contrast to [22] we deal with dynamic objects by using only one map layer in which we handle the movement of a good by simply comparing its current position measurement with its expected position. If the positions are too far apart (in our experiments >1 meter), the good is treated as having been moved and is added as a new feature into the SLAM algorithm. This results in a sequence of maps that contain position estimates and a mapping of goods to positions at each time step.

B. From Positions to Locations

Measured positions are clustered after each step and the generated cluster centroids are used as qualitative locations. Therefore, the mapping of positions to clusters needs to stay fixed even when new centroids are generated by added data. We implemented two clustering methods for later comparison: The first clustering method assigns position estimates to predefined locations (shown in Fig. 4(a)). We used this method for evaluation purposes. The second clustering method computes locations automatically by employing a straightforward greedy algorithm: Positions are clustered together if their surrounding circle is below a certain size; otherwise a new cluster is created (shown for one test run in Fig. 4(b)). Each observation of a good is now attributed by a location and a time step ($\mathrm{obs}(t, G, L)$), which is the starting point for the symbol grounding as described in Section III-C.

[1] `http://artoolkit.sourceforge.net/`

C. From LTL–Worlds to Histories

As described at the end of Section III-D histories of goods are also LTL formulas and as such can also be used during model verification. It is straightforward to implement the rules as Prolog clauses and let Prolog try to prove them. The connection of the world is realized by an ordered list, i.e., two worlds W_i and W_j are connected if W_j directly follows W_i in the list.

We then use Prolog to constructively prove the existence of a history for each good by model verification. The history construction includes the deduction of zones as demonstrated in the example shown in Fig. 3. In general, many histories can be verified for the same observations, e.g., moving a good from A to B to C verifies the model $\mathrm{Redistribution}_{G,A,B} \wedge \mathrm{Redistribution}_{G,B,C}$ but also $\mathrm{Redistribution}_{G,A,C}$. In the latter case the observation that the good was at location B is ignored. Therefore, in ambiguous cases we select the maximal model, i.e., the history involving the largest number of observations.

V. Experiments and Evaluation

In our experimental setting we simulated warehouse processes in our lab in order to measure to which extend histories can be identified correctly.[2]

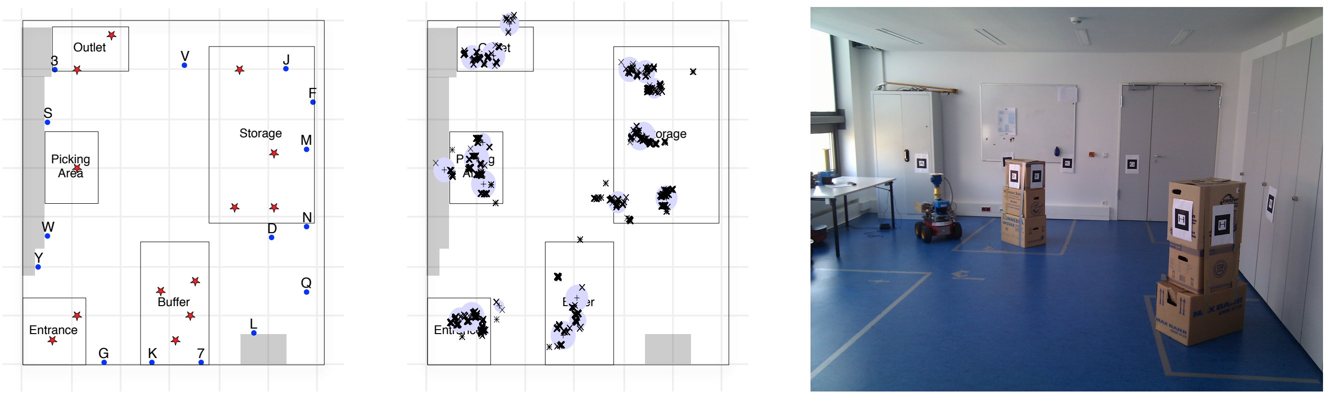
A. Experimental Setup

Our robot platform is an Active Media Pioneer-3 AT controlled by a top-mounted laptop and equipped with a SONY DFW SX900 (approx. 160°FOV) that delivers 7.5 frames per second.

In our lab we simulate a warehouse that consists of five dedicated zones (entrance, buffer, storage, picking, outlet) as seen in Fig. 4(a) and 4(c). 15 tags are distributed within the environment as static landmarks. Goods are represented by boxes with visual tags attached to all sides.

An experiment consists of a series of movements of goods between the zones while our robot is monitoring the environment. For each of the 10 test runs, the robot was placed in the lab and driven around until each landmark had been seen at least twice to ensure a fully mapped environment. Then, we

[2] Video material of a test run is available at `http://www.sfbtr8.uni-bremen.de/project/r3/ECMR/index.html`



(a) Floor plan with zones (b) Automatically generated clusters (c) The lab (picture taken from the outlet zone)

Fig. 4. Experimental Setup: The warehouse (6.12 m $\times$ 7 m) with 5 zones (Entrance, Buffer, Storage, Picking, and Outlet). In (a) the fixed cluster centers are shown as red stars and the landmarks as blue dots with there corresponding letters. In (b) measurements of a single experimental run are shown as black $\times$ and the automatically generated clusters as blue circles.

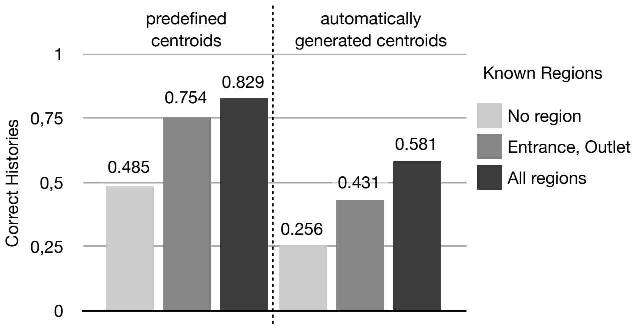


Fig. 5. Evaluation results showing the rate of correct history detections. We evaluated 10 test runs which included a total number of 21 full histories and 18 partial histories.

steered the robot in arbitrary round courses, while we moved boxes through the lab, simulating the previously defined logistic processes (Sect. III-D).

The duration of a test run was between approx. 11 and 28 minutes in which we moved 3 to 8 goods through the warehouse, resulting in 4 to 19 detectable processes per run including runs with only partial histories. Goods were moved between zones while not covered by sensor surveillance, to comply with axiom 2 in section III-B. The robot was driven by hand in the experiments.

As mentioned in section IV-B, we evaluated our approach with two different clustering methods, each one with 3 different settings of background knowledge. In the first setting all zones are previously known, in the second setting only entrance and outlet are known and in the third one the whereabouts of no zone are known.

B. Evaluation

We evaluate our approach based on correctly identified histories. For each good we query its history, i.e., running the model verification to generate it. A history is correctly identified if temporal order and number of processes match the ground truth.

Fig. 5 shows the result of our evaluation. In the most favorable case of knowing all zones and predefined cluster centroids we achieve an average recognition rate of 83%. The experiments comprise of 21 full histories and 18 partial ones. In partial histories, a good either started in the warehouse or after its admission never left it again. Our current interpretation prefers full histories over partial histories and is biased towards an empty starting warehouse, i.e., if the observations verify both admission and take-out we prefer the admission. Especially in the case of having no prior knowledge we found that partial histories reduced detection rate. In particular, with automatically generated centroids and no prior knowledge about the zones; 37.9% of the full histories were correctly found, but only 13.3% of the partial histories were correctly found.

A significant difference can be observed between the two clustering methods, but both follow the same pattern: additional prior knowledge results in more correctly identified histories. If no zone is known, i.e. all zones needed to be inferred, the results show that the approach is still capable of correctly identifying histories. This clearly demonstrates the utility of inference in process recognition.

VI. Discussion

Our work targets online process detection and online queries while the robot is operating. Thus, we rely on observations of goods as soon as we detect them, even if the position estimate is still uncertain. Over time, stability of positions is achieved by clustering them into locations. Every (new) perception of a good at a different location (immediately) triggers the creation of a new world. Poor position estimates (for example, when few tags are detected due to motion blur while the robot turns) can easily be mapped to locations that incorrectly induce movement of a good or lie outside of the zone the good is in. Such cases result in incorrectly detected histories. The results in Fig. 5 confirm this: When providing stable, pre-defined cluster centers, detection rates are significantly higher, especially when more domain knowledge is included. Thus, excluding estimates with too much uncertainty would improve the detection rate. Using uncertainty estimates for measured positions will also improve the robustness of geometrical shape estimation for the zones. However, the current implementation of the TreeMap SLAM algorithm[3] does not provide uncertainty estimates.

In a real-world environment it is reasonable to assume knowledge about entrance and outlet zones (e.g., by placing tags to mark the end of the warehouse). The observable difference between knowing all zones and knowing only entrance and outlet is relatively small, especially when predefined clusters are used (83% and 75% respectively). These results illustrate the feasibility of our approach.

In this work, we currently restrict ourselves to use inference only on sensory observations. As stated before, the detection of correct histories improves with better clustering (e.g., by using outlier detection).

To query more complex information it would be reasonable to also include knowledge gained within the mapping process. That is, information on goods we have observed before and included into the map, but that we are not able to perceive at the very moment. For these objects, we have a strong *belief* of their existence and position, but this belief can—according to the actual observation—not be validated. A possibility to include reasoning on such beliefs is to use a logic that provides a modal belief operator, such as the logic for BDI agents presented in [16]. Another source of information for more complex queries could be provided by an ontology, as shown in [15].

Our logic foundation also supports multiple instances of the same type of good, e.g. splitting or merging packages for delivery. However, due to limited size in our lab, we did not include this feature in our experiments.

VII. Summary

In this paper we propose an approach to process detection based on a specification of processes as temporal logic formulas. We show in our evaluation that our approach is applicable using real sensory data from a mobile robot. One strength of our approach is that it can fill in missing pieces of information by reasoning about processes and spatial configurations in the same formalism. It is also possible to query about previously unspecified processes as well as about spatial facts, such as functional zones.

Basing our approach on the well-established linear temporal logic not only works for passive process detection but would also allow us to incorporate so-called search control knowledge and perform high-level planning [4], i.e., doing *active* process detection in the sense of planning where to go for more information. This is the objective of our future research.

References

[1] Marco Antoniotti and Bud Mishra. Discrete event models + temporal logic = supervisory controller: Automatic synthesis of locomotion controllers. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, volume 2, pages 1441–1446, 1995.

[2] D. Avrahami-Zilberbrand, G.A. Kaminka, and H. Zarosim. Fast and complete symbolic plan recognition: Allowing for duration, interleaved execution, and lossy observations. In *Proceedings of the AAAI Workshop on Modeling Others from Observations (MOO)*. Citeseer, 2005.

[3] Dorit Avrahami-Zilberbrand. *Efficient Hybrid Algorithms for Plan Recognition and Detection of Suspicious and Anomalous Behavior*. PhD thesis, Bar Ilan University, 2009.

[4] Fahiem Bacchus and Froduald Kabanza. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 116(1-2):123 – 191, 2000.

[5] Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. *Journal of the ACM (JACM)*, 57(3):1–46, 2010.

[6] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *The International Journal of Robotics Research (IJRR)*, 24(1):39–41, 2005.

[7] Marcello Cirillo, Lars Karlsson, and Alessandro Saffiotti. A human-aware robot task planner. In Alfonso Gerevini, Adele E. Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *Proceedings of the 11th International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI, 2009.

[8] Christophe Dousson and Pierre Le Maigat. Chronicle recognition improvement using temporal focusing and hierarchization. *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI)*, pages 324–329, 2007.

[9] Udo Frese. *An $O(\log n)$ Algorithm for Simulateneous Localization and Mapping of Mobile Robots in Indoor Environments*. PhD thesis, University of Erlangen-Nürnberg, 2004.

[10] Torsten Hildebrandt, Lutz Frommberger, Diedrich Wolter, Christian Zabel, Christian Freksa, and Bernd Scholz-Reiter. Towards optimization of manufacturing systems using autonomous robotic observers. In *Proceedings of the 7th CIRP International Conference on Intelligent Computation in Manufacturing Engineering (ICME)*, June 2010.

[11] Marius Kloetzer and Calin Belta. LTL planning for groups of robots. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 578–583, 2006.

[12] Marius Kloetzer and Calin Belta. Automatic deployment of distributed teams of robots from temporal logic motion specifications. *IEEE Transactions on Robotics*, 26(1):48–61, 2010.

[13] Lin Liao, Donald J. Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5–6):311–331, 2007.

[14] Tommaso Magherini, Guido Parente, Christopher D. Nugent, Mark P. Donnelly, Enrico Vicario, Frederico Cruciani, and Cristiano Paggetti. Temporal logic bounded model-checking for recognition of activities of daily living. In *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)*, Corfu, Greece, November 2010.

[15] Fulvio Mastrogiovanni, Antonio Sgorbissa, and Renato Zaccaria. Context assessment strategies for ubiquitous robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2717–2722, 2009.

[16] John-Jules Meyer and Frank Veltman. Intelligent agents and common sense reasoning. In Patrick Blackburn, Johan Van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, chapter 18, pages 991 – 1029. Elsevier, 2007.

[17] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–57, 1977.

[18] Stephen L. Smith, Jana Tůmová, Calin Belta, and Daniela Rus. Optimal path planning under temporal logic constraints. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3288–3293, Taipeh, Taiwan, October 2010.

[19] Michael Ten Hompel and Thorsten Schmidt. *Management of Warehouse Systems*, chapter 2, pages 13–63. Springer, Berlin Heidelberg, 2010.

[20] Trung-Dung Vu, Julien Burlet, and Olivier Aycard. Grid-based localization and local mapping with moving object detection and tracking. *Information Fusion*, 12:58–69, January 2011.

[21] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26:889–916, September 2007.

[22] Denis F. Wolf and Gaurav S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19:53–65, 2005. 10.1007/s10514-005-0606-4.

[23] Qiang Yang. Activity recognition: Linking low-level sensors to high level intelligence. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 20–25, Pasadena, CA, July 2009.

[3] as provided at `http://openslam.org/treemap.html`