Extracting Spatial Relations Among Objects for Failure Detection

Mustafa Ersen[1], Sanem Sariel-Talay[1], and Hulya Yalcin[2]

[1] Artificial Intelligence and Robotics Laboratory, Computer Engineering Department
[2] Electronics and Communication Engineering Department
Istanbul Technical University, Turkey
{ersenm,sariel,hulyayalcin}@itu.edu.tr

Abstract. A cognitive robot may face failures during the execution of its actions. These failures are mostly due to the gap between the physical world and the constructed symbolic plans, some internal problems that may occur in its embodiment or unexpected external events. In this paper, we propose a visual scene interpretation system for extracting spatial relations among objects in a scene and using these relations to detect failures during the plan execution. Our system uses LINE-MOD and HS histograms in order to recognize textureless objects with different shapes and colors. Then, it analyzes the scene to specify the world state after each action execution. Our focus in this research is on particularly the following spatial relations: *on*, *on_table*, *clear* and *unstable*. In the experiments, we test the performance of our system on recognizing objects, determining pairwise spatial relations among them, and detecting failures using these relations. Our preliminary results reveal that our system can be successfully used to extract spatial relations in a scene, and to determine failures during plan execution by using this information.

Keywords: cognitive robots, failure detection, spatial reasoning, object recognition, automated planning

1 Introduction

A cognitive robot possesses abilities to construct symbolic plans to solve given problems and to execute these plans in the real world. Automated planners are commonly used for determining a valid sequence of actions for a robot to achieve its goals. These planners use high-level description of the problem and the domain (i.e. initial/goal states and operators corresponding to real-world actions) to construct a plan. After obtaining a valid plan, the robot needs to execute the corresponding real-world actions in order to attain the desired goal. However, it may face several types of failures during the execution of its actions in the real world [1]. These failures may arise due to the gap between the real-world facts and their symbolic representations used during planning, unexpected events that may change the current state of the world or internal problems.

Ensuring robostness is crucial for a cognitive robot in order to accomplish the given goals in the real world. In this work, we investigate how spatial relations among objects are determined using visual data from an RGB-D camera and how this information is used to detect action execution failures in the real world. As a motivating example to illustrate the stated problem, consider the object manipulation task in the blocks world domain. An example plan constructed for a 3-block problem is given in Figure 1. In this toy problem, the aim is stacking three blocks on top of each other where all blocks are initially on the table and without any other objects on top of them (i.e. satisfying *clear* predicate). During the execution of the generated plan, the robot may fail in executing action *stack*. Possible reasons for this failure might be the weight of the object, improper grasp position or a vision problem. To ensure robustness in such cases, the robot needs to continuously monitor the state space for anomalies during action execution.

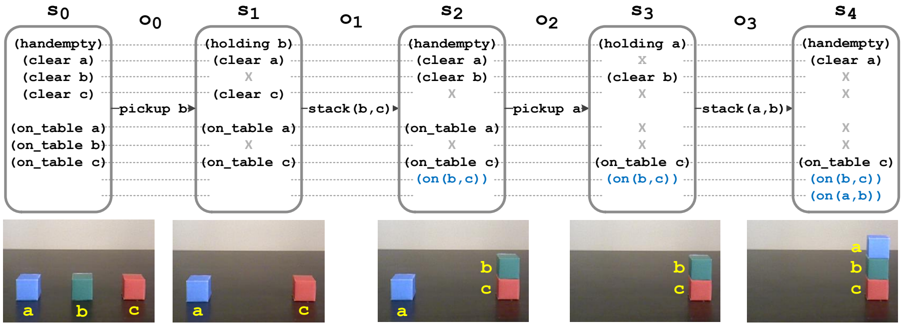

Fig. 1. The execution trace for solving the blocks world problem with a three block case is given. (top) The successor states and the actions taken at each state, (bottom) the visual scene observed at each world state are presented.

Throughout the paper, we first give some background information on the areas of automated planning, object recognition and scene interpretation. Then, we describe the details of our system for determining spatial relations among the objects in order to detect failures. We then give empirical results of our approach followed by the conclusions.

2 Background

In this section, we formulate the planning problem and give a brief review of the approaches used for recognizing objects and determining spatial relations in the scene. Then, in the following section, we present our solution to the stated problem.

2.1 Automated Planning

Cognitive robots may use automated onboard action planning for online generation of action sequences to accomplish given tasks against exogenous events.

A planning task Π can be described on a state space S containing a finite and discrete set of states including an initial state s_0 and a goal state s_G, and a state transition function $s_{t+1} = f(o_t, s_t)$ where $o_t \in O(s_t)$ is an operator applied in state s_t. State transition function is realized through planning operators $o \in O$ that are defined as symbolically abstracted representations of real world actions $a \in A$. A planning operator can be formalized as a tuple $o = \{pre(o), add(o), del(o), cost(o)\}$ where $pre(o)$ defines the preconditions, $add(o)$ and $del(o)$ define the effects of the operator and $cost(o)$ represents the cost of the corresponding action. $o_t \in O(s_t)$ is defined as the set of applicable operators in a state $s_t \in S$ determined by checking preconditions of the operators to satisfy $pre(o_t) \subseteq s_t$. By applying o_t at state s_t, a new state $s_{t+1} = add(o_t) \cup (s_t \setminus del(o_t))$ is observed. Planning task is achieved by a planner to reach s_G from s_0 by selecting a sequence of operators from $O(s_t)$ at successive states s_t and executing the corresponding actions $a_t \in A$ in the given order. After searching the whole space of operators, the planner constructs a valid plan $P = o_{0:G}$ by considering an optimization criteria (e.g., makespan) and the duration/cost of each operator. Having generated a valid plan P, the robot can execute each corresponding action $o_t \rightarrow a_t \in A$ in sequence in the physical world. If all goes well with execution, the robot successfully attains s_G. However, due to non-deterministic actions and different sources of uncertainty in physical environments, several failures may be encountered [1]. Our primary focus is action execution failures. To detect a failure, the robot should monitor its execution, recognize the objects it interacts with (if any) and interpret the scene continuously.

2.2 Object Recognition

There are various approaches for recognition of objects in a scene using different types of visual clues. These approaches can be categorized as 2D object recognition approaches based on local invariant feature descriptors and 3D object recognition approaches based on surface normals computed from the depth map. In the case of 2D color data, local feature descriptors are used to determine patterns in the image which differ from the other pixels in their neighborhood. These distinguishing parts of the image (i.e. keypoints) are generally chosen by considering sharp changes in color intensity and texture. To store the keypoints, descriptors are computed around them which are suitable for measuring their similarity. The idea of using local invariant descriptors became popular when Scale-Invariant Feature Transform (SIFT) [2] was proposed in 1999. SIFT is a keystone in the area, and it is used as the base of the state of the art techniques. It is known to be invariant against geometric transformations such as scale, rotation, translation and affine transformation to a sufficient extent for a lot of applications. It is also claimed to perform well against noises and changes in the illumination. However, SIFT-based approaches are known to have deficiencies in recognizing textureless objects. Information on the 3D shapes of the objects and their colors can be used in order to deal with this problem. By the development of RGB-D sensors, it is possible to get depth information as well as color and texture information for this purpose. To utilize the depth values captured using these types of sensors, different 3D descriptors have been proposed [3]. These descriptors can be divided in two categories: local descriptors and global descriptors. Local descriptors are used to describe the local geometric properties of distinguishing points (i.e., keypoints) whereas global descriptors capture depth-based features globally for a presegmented object without storing local information for extracted descriptors. Among these, LINE-MOD [4] is unique as it is a linearized multi-modal template matching approach based on weak orientational features which can be used to recognize objects very fast making this approach the most suitable one for real-time robotic applications.

2.3 Determining Spatial Relations

Detecting and representing structures with spatial relations in a scene is known as the scene interpretation problem. While this is a trivial task for humans, interpreting spatial relations by processing visual information from artificial vision systems is not a totally solved problem for autonomous agents [5]. In the recent years, some approaches have been proposed to solve this problem [6–9]. Some of these works use 2D visual information for extracting qualitative spatial representations in a scene [6, 7]. In these works, some topological and orientational relations among objects are determined in the scene. In another work, proximity-based high-level relations (e.g., relative object positions to find objects that are generally placed together) are determined by comparing Euclidean distance between pairs of recognized objects in the scene [8]. This system relies on 3D data obtained using an RGB-D sensor and an ARToolkit marker acting as a reference coordinate system. Sjöö et al. have proposed a method for determining topological spatial relations *on* and *in* among the objects, and this information is used to guide the visual search of a robot for the objects in the scene [9]. Object recognition approach used in their work is based on matching SIFT [2] keypoints on a monocular image of the environment.

Our proposed work differs from the previous studies in two ways. First, determining spatial relations is done for a higher level task of detecting failures after action executions. Second, the object recognition system used in this work is more generic as it can deal with textureless objects that do not have any distinguishing texture information.

3 Scene Interpretation for Monitoring Action Executions

We propose a failure detection system based on visual information. The system involves three main procedures, namely, object recognition, scene interpretation and failure detection. In the system, first, each object of interest is modelled by creating multi-modal LINE-MOD [4] templates from different viewpoints. A template involves the surface normals within an object and the color gradients around its borders. As well as the multi-modal templates of LINE-MOD, a color histogram is generated to model each template in Hue-Saturation-Value

(HSV) color space. In this histogram, V(value) is omitted as it is strongly dependent on illumination conditions, and normalized values are taken for H(hue) and S(saturation). By using these histograms, object recognition process is improved as color values inside the templates are also considered. Then, LINE-MOD templates for all the objects of interest are searched in the scene using a sliding window approach to find matches. The threshold is specified as 80% by taking into account the noisy data captured using an RGB-D sensor. These matches are then verified comparing HS histograms of corresponding templates with match regions based on normalized correlation, and false positives are eliminated. The threshold value is taken as 0.5 by considering the changes in the illumination.

After the objects are recognized and located in the scene, qualitative spatial relations are determined for failure detection. In the blocks world domain, these relations are *on*, *on_table*, *clear* and *unstable*. Initially all the recognized objects are assumed to be *on_table* and *clear*. Then the *on* relation is determined between each pair of objects as follows,

$$\forall obj_i, obj_j, (EC(obj_i, obj_j) \vee PO(obj_i, obj_j)) \wedge N(obj_i, obj_j) \Rightarrow on(obj_i, obj_j)$$

where *EC(externally connected)* and *PO(partially overlapping)* are topological predicates of RCC8 [10] and *N(north)* is a directional predicate of cardinal direction calculus [11]. After determining the *on* relation, *clear* and *on_table* relations are updated for the objects involving in this relation as follows,

$$\forall obj_i, obj_j, on(obj_i, obj_j) \Rightarrow \neg on\_table(obj_i) \wedge \neg clear(obj_j)$$

To eliminate false positives in the extraction of *on_table* relations due to recognition failures, the area under the object is checked in order to see if it is planar or not. If the area is not a horizontal plane, than it is assumed that the corresponding object is not on the table. Finally, *unstable* relation is determined by checking the horizontal projections of the aligned object templates. If the horizontal projection of the upper object in the *on* relation has an unsupported part (i.e., out of the area covered by the object below) of more than 1/4 ratio to its length, this *on* relation is assumed to be *unstable*.

Action execution failures are detected by checking the state of the world with respect to spatial relations after each action execution. We consider three states after executing an action [12], namely *success*, *fail-safe* and *fail-unsafe* of which we repeat definitions here for convenience.

Definition 1 (*success* state) If all the desired effects of the action occurs in the environment, the situation is specified as *success*.

Definition 2 (*fail-safe* state) If the state of an execution is not *success* but the state does not change, the situation is specified as *fail-safe*. For example, the robot fails in picking up an object but the state of the object is not changed.

Definition 3 (*fail-unsafe* state) If the execution of an action fails and there is any damage and/or dangerous situation (e.g., an undesirable state is observed) or the robot cannot judge whether there is any harmful situation, the situation is specified as *fail-unsafe*. For example, the robot fails in picking up an object, and the object is broken into pieces or fallen down the ground out of reach of the robot.

4 Experimental Evaluation

In the experiments, the proposed system is evaluated in real time for different possible situations in the scene using the real-world data captured by an RGB-D sensor. The objects used in these experiments are three paper blocks, two plastic toy grapes and a single toy box (Figure 2). These objects are selected as they have different shape and/or color features.

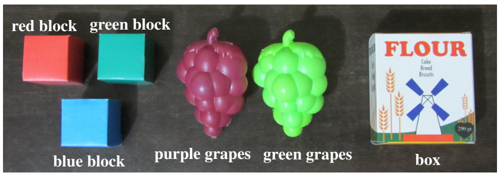

Fig. 2. The objects used in the experiments.

First, the overall recognition performance has been evaluated by comparing the results of LINE-MOD and our approach combining LINE-MOD with HS histograms. The results are illustrated in Table 1 as a confusion matrix for 120 different scenes (20 scenes for each object). As expected, both LINE-MOD and our approach give good recognition rate for different shaped objects. However, LINE-MOD cannot always distinguish similar shaped objects with different colors. Our approach based on checking HS histogram correlations on the results obtained using LINE-MOD leads to much better results in these situations. False negatives in recognition are slightly greater in our approach since some correct results are eliminated by checking color correlation.

Table 1. Confusion matrix for recognition: LINE-MOD / LINE-MOD&HS histograms.

	red block	*green block*	*blue block*	*green grapes*	*purple grapes*	*box*	*not found*
red block	12/19	3/0	4/0	0/0	0/0	0/0	1/1
green block	2/0	13/18	4/0	0/0	0/0	0/0	1/2
blue block	5/0	2/0	12/18	0/0	0/0	0/0	1/2
green grapes	0/0	0/0	0/0	13/20	7/0	0/0	0/0
purple grapes	0/0	0/0	0/0	6/0	13/18	0/0	1/2
box	1/0	2/0	0/0	0/0	0/0	17/20	0/0

Second, the performance of our system for extracting spatial relations: *on*, *on_table*, *clear* and *unstable* has been tested in an experiment involving 100 scenes (50 scenes for the blocks, 50 scenes for the grapes and the box). The results are shown in Figure 3. As given in these results, our system can be used to successfully detect relations for all the objects used in our tabletop scenarios. The highest error is in determining *on* relation by 20% and this is caused by the objects that cannot be recognized. When the object that is located below another object cannot be recognized, this also affects the success of determining *on_table* relation. Similarly, there are some errors in determining *clear* relation for an object when another object that is located on top of it cannot be recognized. Errors in *unstable* relation are due to the failures in recognition or bad alignment of the recognized templates.

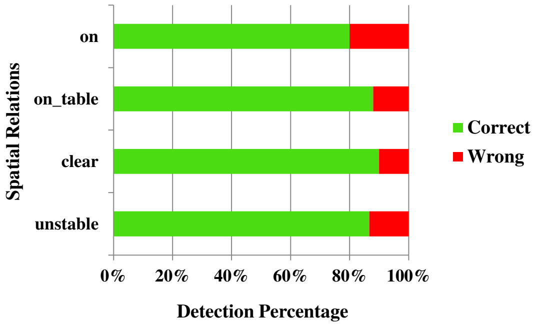

Fig. 3. Performance of the proposed system on determining spatial relations.

In the last set of experiments, the performance of the failure detection process has been tested on 20 different scenes that are set for each execution state: *success*, *fail-safe* and *fail-unsafe*. Similar to the previous set of experiments, it has been observed that, spatial relations are determined correctly in 85% of the scenes where the action is executed successfully. Moreover, the system has been observed to label 90% of the scenes with a *fail-safe* state correctly where the state of the world does not change after executing an action. In these experiments, when the object to be stacked is fallen down the table and this object cannot be recognized in the scene, the state is assumed to be a *fail-unsafe* state. With this assumption, the system has identified all *fail-unsafe* examples in the 3-blocks problem correctly.

5 Conclusion

We have presented an approach for detecting failures to ensure robust task execution in cognitive robotic applications. Our approach is based on using visual information extracted from the scene in order to determine spatial relations among the objects that are involved in manipulation scenarios. First, we have shown how our system can be used to recognize objects with different geometric shapes and colors. Then, we have given the details of the visual scene interpreter for specifying spatial relations among the objects of interest and evaluating these relations for detecting failures during action executions. The preliminary results of the conducted experiments on our system indicate that the system can be used to successfully detect states with failures in an object manipulation scenario. In our future studies, we plan to conduct experiments on larger sets of scenes involving various objects to justify our research. Our ongoing work includes the integration of temporal reasoning into spatial reasoning in order to detect the possible causes of failures from previous states (e.g., an unstable stack of blocks causing a failure when stacking another block on top of them).

6 Acknowledgement

This research is funded by a grant from the Scientific and Technological Research Council of Turkey (TUBITAK), Grant No. 111E-286. We thank Prof. Muhittin Gokmen for his recommendations on vision algorithms.

References

1. Karapinar, S., Altan, D., Sariel-Talay, S.: A robust planning framework for cognitive robots. In: Proc. of the AAAI-12 Workshop on Cognitive Robotics. (2012) 102–108
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the 7th IEEE Intl. Conference on Computer Vision (ICCV'99). (1999) 1150–1157
3. Aldoma, A., Marton, Z.C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S., Vincze, M.: Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation. IEEE Robotics and Automation Magazine **19**(3) (2012) 80–91
4. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P.F., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of textureless objects. IEEE Trans. on Pattern Analysis and Machine Intelligence **34**(5) (2012) 876–888
5. Neumann, B., Möller, R.: On scene interpretation with description logics. Image and Vision Computing (Cognitive Vision Special Issue) **26**(1) (2008) 82–101
6. Falomir, Z., Jiménez-Ruiz, E., Escrig, M.T., Museros, L.: Describing images using qualitative models and description logics. Spatial Cognition and Computation **11**(1) (2011) 45–74
7. Sokeh, H.S., Gould, S., Renz, J.: Efficient extraction and representation of spatial information from video data. In: Proc. of the 23rd Intl. Joint Conference on Artificial Intelligence (IJCAI'13). (2013) 1076–1082
8. Kasper, A., Jäkel, R., Dillmann, R.: Using spatial relations of objects in real world scenes for scene structuring and scene understanding. In: Proc. of the 15th IEEE Intl. Conference on Advanced Robotics (ICAR'11). (2011) 421–426
9. Sjöö, K., Aydemir, A., Jensfelt, P.: Topological spatial relations for active visual search. Robotics and Autonomous Systems **60**(9) (2012) 1093–1107
10. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Proc. of the 3rd Intl. Conference on Principles of Knowledge Representation and Reasoning (KR'92). (1992) 165–176
11. Frank, A.U.: Qualitative spatial reasoning with cardinal directions. In: Proceedings of the 7th Austrian Conference on Artificial Intelligence. (1991) 157–167
12. Karapinar, S., Sariel-Talay, S., Yildiz, P., Ersen, M.: Learning guided planning for robust task execution in cognitive robotics. In: Proc. of the AAAI-13 Workshop on Intelligent Robotic Systems. (2013)