Hauck, A.; Stöffler N. O.: "A Hierarchic World Model Supporting Video-based Localization, Exploration and Object Identification", Proceedings ACCV '95, Vol 3, 176-180, December 1995, Singapore

A Hierarchic World Model Supporting Video-Based Localization, Exploration and Object Identification*

Alexa Hauck, Norbert O. Stöffler

Department of Process Control Computers
Prof. Dr. -Ing. G. Färber
Technical University of Munich, Germany
e-mail: {hauck|stoffler}@lpr.e-technik.tu-muenchen.de

Abstract The design of mobile robots which can cope with unexpected disturbances like obstacles or misplaced objects is an active field of research. Such an autonomous robot assesses the situation by comparing data from one or more sensors with an internal representation of its environment. In this paper we present a hierarchically structured world model that combines a general geometric object representation with sensor- and task-specific features and therefore can be used for various sensors and perception tasks. A symbolic layer enables communication with planning instances or other robots.

1 Introduction

This work is part of a research project towards development of autonomous mobile robots (AMR) which can fulfil service and transport tasks in structured environments like office buildings and industrial plants. For planning its actions such an autonomous robot needs an internal representation of its environment (*environmental* or *world model*). To keep this representation up to date it constantly has to compare the model with data acquired by one or more sensors. A model can also be considered as a set of hypotheses regarding the positions and states of the elements of the world, including the AMR itself. These hypotheses can be tested and modified by comparison with sensor data.

Raw sensor data is preprocessed to extract sensor-specific features. In case of a grey-scale video sensor these features are typically edges. To confirm or reject hypotheses corresponding features are predicted from the model and compared with extracted ones.

Most modelling techniques described in literature are specialized on a certain application; thus such models are well adapted to specific perception tasks and sensors [2] or environments [4] but cannot be used in a general way. Information is often stored merely on feature level; sometimes several models are held in parallel and used independently for different tasks. In contrast to this our approach aims at designing a generally applicable world model by combining information needed for various sensors and robot tasks into one consistent description on different levels of abstraction.

In this paper we emphasize the interactions with video sensors. In chapter 2 we describe the internals of the model itself and in chapter 3 the possible accesses by several perception tasks, concentrating on one experimental example in chapter 4.

2 Structure of the model

To permit sensor independent abstractions the model structure is based on three dimensional solid modelling techniques.

Elements of the world influence sensor images in two ways. They can be the source of sensor-specific features and they can hide other elements. The latter aspect is modelled by a polyhedral boundary representation, called the obstacle description.

If possible, sensor-specific features are calculated from the obstacles like points of normal incidence for a radar beam. Because of their dependency of various factors like colours and illumination, video-specific edges are modelled separately by line-segments which are tested for their visibility and projected on the image-plane during the prediction. These line-segments are based on the same set of vertices as the obstacles but do not necessarily coincide with boundary edges. This dualism allows the representation of the obstacles by exclusively convex polygons which facilitates the visibility calculation.

To initiate a prediction the obstacles and features inside the vision pyramid are determined. The obstacles are rendered into a z-buffer [5]. Against the resulting depth-map the corresponding features are then tested for their visibility. To access the vertices inside the vision pyramid appropriate index structures are nescessary. In first experiments demonstrating localization in a static environment, a two-dimensional spatial-tree has proven to be an efficient index structure for otherwise unrelated world elements [7, 8].

To allow for more complex perception tasks and non-static environments, we are now examining a hierarchic

*The work presented in this paper is supported by the *Deutsche Forschungsgemeinschaft* as part of an interdisciplinary research project on "Information Processing in Autonomous Mobile Robots" (SFB 331).

structure with additional symbolic information (see figure 1).

Elements of the world are aggregated to form *named objects*. Because it is neither possible nor necessary to describe the complete environment of the robot in terms of distinguishable, named objects, a pseudo object called *background* is introduced. It encompasses all world elements without special object assignment.

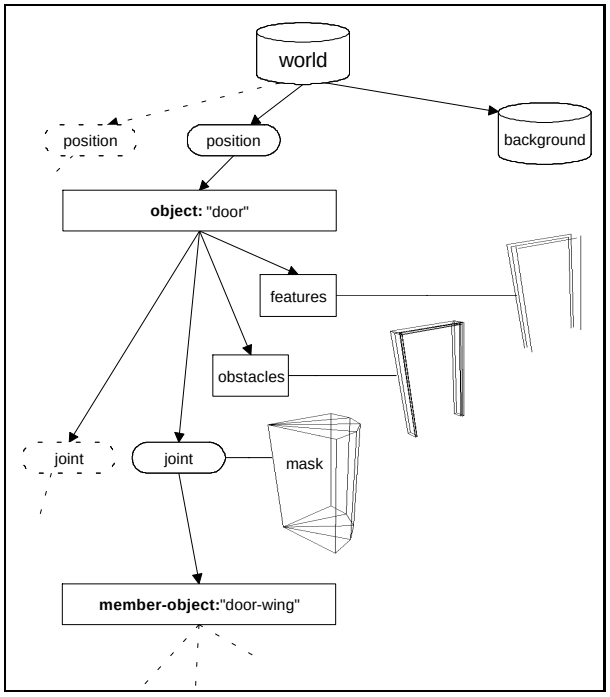

Figure 1: Model structure

The description of a named object is built up recursively. An object can contain other objects which are termed *member-objects*. The distinction between object and member-objects is made by kinematic degrees of freedom. That means object and member-object are always connected by a rotatory or translatoric joint, represented by a modified Denavit-Hartenberg formalism. The possible positions of a joint are normalized to the unit interval allowing a unified treatment of joint-states; additionally there exists a state called *unknown* and a list of *preferred states*.

Each branch in the object-tree carries its own obstacle and feature description. The representation of features is also extended by the possibility of defining aggregations of simple features and attributes to form complex ones. Those complex features may be task-specific to alleviate special matching problems. An example for the use of such aggregated features is given in chapter 4.

To deal with unknown states during a prediction as explained in chapter 3.1, the space potentially being occupied by a moving member-object is stored as an additional obstacle, called *mask*.

Geometrically identical objects form an *object class*. The invariant parts of an object description are stored only once for each class; the objects (i.e. the instances of a class) differ in their individual positions and joint states.

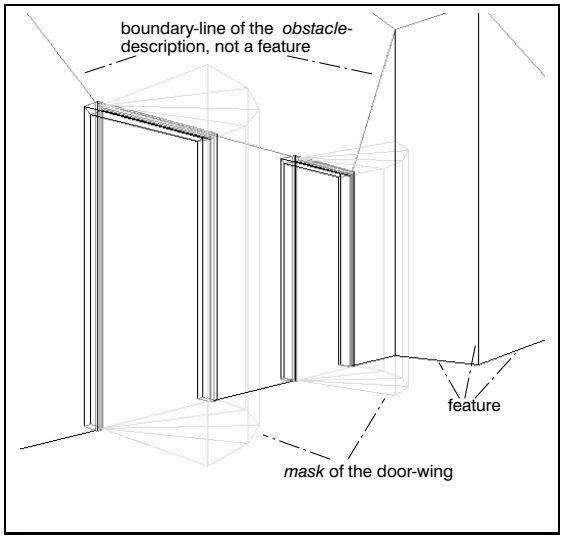

Figure 2: Model of a corridor

Figure 2 shows an exemplaric model of a corridor, consisting of two doors of the same class and some walls which are part of the background. For this depiction no hidden lines were removed. black lines are video-specific features, dark-grey lines stand for obstacle boundaries which do not coincide with features and the light-grey lines represent the masks of the door-wings. The door-wings themselves are left out for simplification reasons.

3 Application framework and model access

A possible application framework for the world model is shown in figure 3.

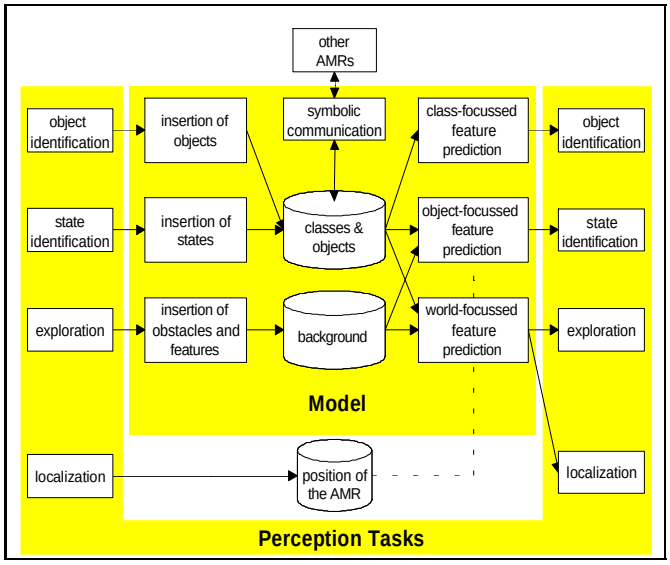

Figure 3: Application framework

Several perception tasks[1] are engaged in keeping the internal representation of the robot's environment up to date by testing stored information against sensor data. They interact with the model on different levels of abstraction according to the parts of the model they regard as hypothetic. Each perception task is implemented by a separate client module which extracts its own relevant features from the sensor data and simultanously requests predictions from the model server. Then it compares the two sets, interpretes the difference and updates the model accordingly.

Interferences between the quasi-parallel model accesses of different tasks are avoided by private communication channels, called *accessors*. These accessors contain the current set of parameters, like assumed camera pose, states and two virtual pointers termed *focus* and *zoom*. The focus points on the task relevant part of the model; after "focussing" an object the states of a private copy of this object are accessible. This allows testing hypothetic states without changing the world-model. The zoom influences the result of the feature prediction in a way comparable to a camera zoom: After pointing it on a node of the object-tree only features of the downward parts are predicted. Note that still all obstacles are used for the visibility calculation. Both, focus and zoom, can be moved step by step up and down the object tree. To allow successive tests of similiar hypotheses, the parameters are handed to the model incrementally: E.g. a client can first set the camera pose, focus on the object of interest and then alternatingly set the state of a joint and get a prediction without having to specify pose and object again each time. The use of those virtual pointers also encapsulates the internal representation of the object-tree, allowing further optimizations without changes in the model access interface.

3.1 Localization

A common parameter which is assumed to be known by several perception tasks is the position of the robot itself, i.e. the camera pose.

It is updated by a task called *localization* [9]. For the prediction all world elements belonging to objects or the background are taken into account at their current states. Thus the accessor of this task is "focussed on the world".

Due to the spatio-temporal restriction of the robot-position a tracking approach can be applied. The last calculated or dynamically extrapolated position is used for the next prediction. If the cycle-time is short enough in comparision with the speed of the robot, features can still be matched and the position hypothesis improved. Feature extraction only needs to take place inside of search windows around the predicted features.

[1] An experimental version of this framework has been realized as part of the interdisciplinary research project SFB 331 ("Information Processing in Autonomous Mobile Robots") with different groups working on the individual tasks.

Since the quality of the pose estimation depends on an accurate match only features with guaranteed visibility may be predicted. Member-objects with unknown joint state are replaced by the appropriate *mask*. The projection of this mask into the z-buffer literally "masks out" features which may be hidden by the moveable member-object.

Figure 4 illustrates the effect. While the position of the left door-wing is known (maybe from a concurrent state identification task) the right one is replaced by its mask. The obstacles are shaded according to the contents of the z-buffer; light shades represent small z-values. Actually predicted features are displayed as black lines, hidden ones as light grey lines.

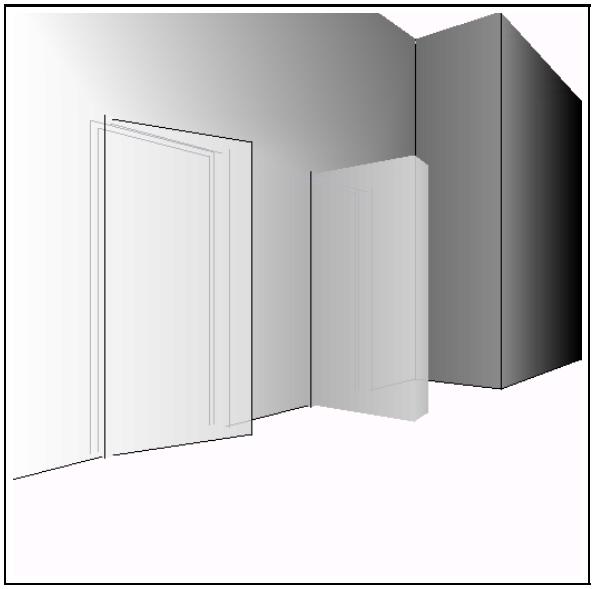

Figure 4: Contents of the z-buffer

3.2 Exploration

Severe mismatch of the features predicted for the current robot-position (according to chapter 3.1) and the extracted ones indicate either a change of the features' position or the presence of new objects. Therefore a second task called *exploration* is charged with evaluating these mismatches and eventually updating feature positions respectively inserting new elements on various levels into the model [3]. For this purpose it tracks and consolidates new features, tries to reconstruct the obstacle description and initiates object identifications. If none of the known object classes can be matched, the new features and obstacles are inserted into the model as part of the background. This at least prevents collisions and further mismatches.

3.3 Object Identification

In the experimental framework, several algorithms for object identification have been examined. All have the need for feature predictions for the assumed object in

common [3, 6]. The accessor is "focussed on object classes" and poses are supplied in class-relative coordinate systems. If an object finally is identified, a new instance is created and inserted into the model.

3.4 State Identification

The task with the most model interactions is the identification of yet unknown object states. It includes focussing on the regarded object, recursive testing of hypothetic states and movements in the object-tree. An experimental example is given in chapter 4.

3.5 Communication

In addition to these sensor-specific access channels, information can be retrieved and manipulated on a symbolic level. Independently operating robots can communicate about the environment by exchanging object names and attributes, i.e. states and positions, via a symbolic communication medium [10].

3.6 A-priori knowledge

In the described framework the description of mission relevant object classes has to be known in advance. In the experimental scenarios, e.g. the mentioned transport and service tasks, most parts of the background and the positions of some relevant objects are also assumed to be known in advance to enable the execution of useful missions. The distinction between objects and background in the a-priori knowledge is made by mission relevance.

Dynamic changes in the environment, like opened or closed doors, moved chairs, changed illuminations etc. are explored "on the fly".

4 Application example

How to access the model and make use of the information stored in it shall be demonstrated with the help of an exemplary robot task: A mobile robot wants to pass through a door and has to determine the opening angle of the door-wing and the position of the door-handle with the aid of a single grey-scale CCD-camera. The door is modelled as already shown in figure 1. The identification of the state of an object is not feasible with typical localization methods since the combined object consisting of door-frame and wing has too many different aspects which would have to be treated separately. The hierarchic object structure on the other hand allows breaking down this complex perception task into simpler ones: First the the door-frame is localized (and with it the axis of rotation of the wing) and then the state of the wing is determined.

At the beginning the state of the wing is *unknown*, the door-handle is in its preferred state. To identify a rotary joint state three aggregated types of features have been found to be appropriate: *Radial edges* (edges whose starting point coincides with the axis of rotation), *parallel-rotary edges* (edges that are parallel to the axis) and *rotary corners*, which connect a *radial* and a *parallel-rotary* edge. The door-wing possesses two *radial* and one *parallel* edge and two *rotary corners* (see figure 5).

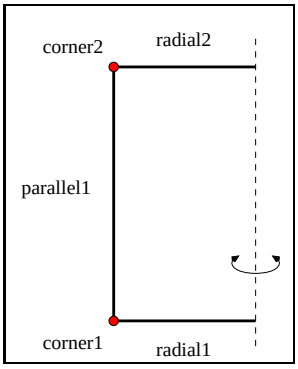

Figure 5: Features of the door-wing

In a first step the robot focusses and zooms on the door. Figure 6 shows the corresponding object-tree and the result of a feature access. Note that the wing's features are not predicted as its state is still *unknown*. By evaluating the difference between predicted and extracted features the door's position is corrected.

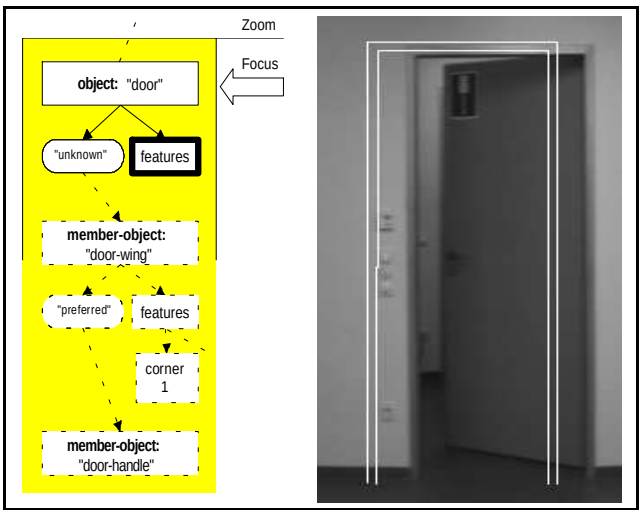

Figure 6: Localization of the door-frame

To determine the opening angle of the wing it is sufficient to locate one of the corners in the image. This can be accomplished e.g. by requesting predictions of the chosen corner for several states of the wing, comparing them with the extracted features and finding the best correspondence. For that purpose the accessor is zoomed e.g. on *corner1* of the wing and predictions for several states are requested. Figure 7 shows the corresponding object-tree and the result of the feature accesses.

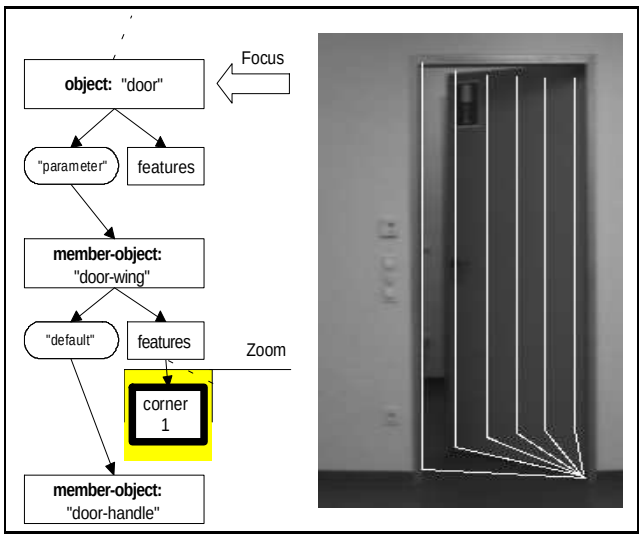

Figure 7: Determining the opening angle

When the corner is found, the state of the door-wing can be computed by intersecting the path of the cornerpoint, in this case a circle, with the corresponding projection ray through the image corner. Figure 8 shows the result of a feature access after setting the state of the door-wing to the computed one; now the door-handle is automatically predicted, too. By unfocussing the accessor the identified states are made public.

Figure 8: Result of the state identification

5 Conclusion

We have presented a hierarchic world model with geometric and symbolic layers which can be used to facilitate video image interpretation by predicting video-specific features. By "focussing" on the object of interest and "zooming" on parts of it, task-relevant information can be accessed easily.

Future work will include the analysis of further aggregated feature types and of task-specific visibility. The object-oriented structure will be expanded to derivation concepts [1] to allow the fusion of similiar classes to abstract ones. To achieve real time behaviour of the model accesses further index structures and stategies for incremental visibility calculations according to incremental changes of the parameters will be evaluated.

References

[1] G. Booch. *Object-oriented design with applications.* Benjamin/Cunning Publishing Company, 1991.

[2] M. Buchberger, K. Jörg, and E. von Puttkamer. Laserradar and sonar based world modelling and motion control for fast obstacle avoidance of the autonomous robot mobot-iv. In *Proc. IEEE Int. Conf. Robotics and Automation, Atlanta*, 1993.

[3] D. Burschka and C. Eberst. Exploration of unknown or partially known environments. In *2. Asian Conference on Computer Vision, Singapore, 5. – 8. Dec.*, 1995.

[4] E. Dickmanns. Active vision through prediction-error minimization. In *Active Perception and Robot Vision*, volume 83 of *NATO ASI Series F*, pages 71–90. Springer Verlag, 1992.

[5] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics – Principles and Practice.* Addison Wesley, Reading, Massachusetts, 1990.

[6] S. Lanser, O. Munkelt, and C. Zierl. Robust video-based object recognition using cad models. In U. Rembold, R. Dillmann, L. Hertzberger, and T. Kanade, editors, *Proc Conf. Intelligent Autonomous Systems*, pages 529–536. IOS Press, 1995.

[7] G. Magin, A. Ruß, D. Burschka, and G. Färber. A dynamic 3d environmental model with real-time access functions for use in autonomous mobile robots. *Robotics and Autonomous Systems*, 14:119–131, 1995.

[8] A. Ruß. *Sensornahe Umgebungsmodellierung für autonome mobile Roboter.* PhD thesis, Technische Universität München, 1994.

[9] A. Ruß, S. Lanser, O. Munkelt, and M. Rozmann. Kontinuierliche Lokalisation mit Video- und Radarsensorik unter Nutzung eines geometrisch-topologischen Umgebungsmodells. In G. Schmidt, editor, *9. Fachgespräch "Autonome Mobile Systeme", München*, pages 313–327, 1993.

[10] J. Schweiger, A. Koller, and K. Ghandri. A distributed real-time knowledge base for teams of autonomous systems in manufacturing environments. In *Proc. of the Seventh Int. Conf. on Industrial and Engineering Appl. of Artificial Intelligence and Expert Systems*, May–June 1994.