FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D5.2005-1
Report on experiments on hierarchical representations of space.

Due date of deliverable: 31/12/2005
Actual submission date: 25/01/2006

Start date of project: January 1st, 2004 **Duration** : 48 months

Contributing partners : UvA, EPFL, KTH, LAAS, UniBi

Revision: final
Dissemination level: PU

Executive Summary

In RA5 (Spatial cognition and multimodal situation awareness), work was carried out in two workpackages: WP5.1: 'Models of space', and WP5.2: 'Models of objects'. In this deliverable we present the work carried out in WP5.1.

A presentation is given on the three approaches studied in the Cogniron project on hierarchical representations of space. We describe a) hierarchal representations by means of clustering techniques on the sensory data, b) by means of object detections and c) work on space representations for interaction with humans.

The report consists of a short introduction and summary of the work done, followed by a complete list of publications from WP5.1 and a reprint of a selected number of papers.

Role of spatial representations in Cogniron

Mobile cognitive personal assistants need to have a representation of space to be useful. Many representations of space as presented in robotic literature are of hierarchical nature: this is needed to keep the computational complexity in bounds. On the other hand, literature on cognitive systems also suggests a hierarchy in spatial relations as used by humans. For a cognitive personal assistant it is crucial that the two representations are consistent.

Relation to the Key Experiments

The work on space modelling will mainly be demonstrated in the Home Tour scenario. In a natural interaction with the user, the robot will be guided around the home, learning places and the layout. The robot has to be sent to locations using natural human concepts.

Table of Content

1 Introduction

Mobile robot localization and navigation requires an internal representation of the environment. Several researchers (e.g. [Kuipers77], [Thrun98]) have proposed a hierarchy of maps to represent large environments at different resolutions, or levels of abstraction, simultaneously.
Typically two levels of abstraction are used: a base-level map and a higher-level topological map. The higher-level, abstract map may be used to represent larger areas of a building, for instance as a graph connecting rooms and corridors, without representing the exact spatial relationship of individual locations within rooms and corridors. Such a high-level map is used to construct abstract plans to navigate from one room to another, without having to worry about exact spatial details within the rooms. The base-level map can be used for precise navigation from one room to the next and to target locations within an individual room, without having to worry about other rooms.
The primarily goal of most work on hierarchical representations is to limit computational complexity. However, for a personal assistant the representation should also involve spatial concepts that are meaningful in an interaction with humans. Literature on representations of spatial representations in humans also describes hierarchical representations [McNamara86]. This is the reason why hierarchical representations are important in the Cogniron project.

2. Research questions from the project plan

In the 2nd implementation plan, we formulated a number of research questions:

- *How can we come to a hierarchical representation of space?*
- *How do we achieve a consistent representation between 'local' representations and the higher level (categorical, semantic) representations?*
- *Are the formed categories meaningful to humans and how can human supervision aid the categorization?*
- *Can we develop methods to represent space by detecting objects?*

These questions stood central in the three work task followed by the members of the consortium:

- WP 5.1.1 Sensor-based methods for topological map building.
- WP 5.1.2 Integration of local representations and categorical representations
- WP 5.1.3 Space descriptions using object detection.

In this report we give an overview of the work done on hierarchical models of space in Cogniron. The general picture is sketched, followed by a list of references and the reprints of a selection of publications.

3. Overview

In the Cogniron project five groups are involved in space representations for a personal mobile assistant. The work carried out concentrated on the use of low level sensory data (appearance, visual feature or range data) for a hierarchical representation, the use of objects for space recognition and the use of human interaction in the process of finding concepts. A schematic representation of the work is given in figure 1.

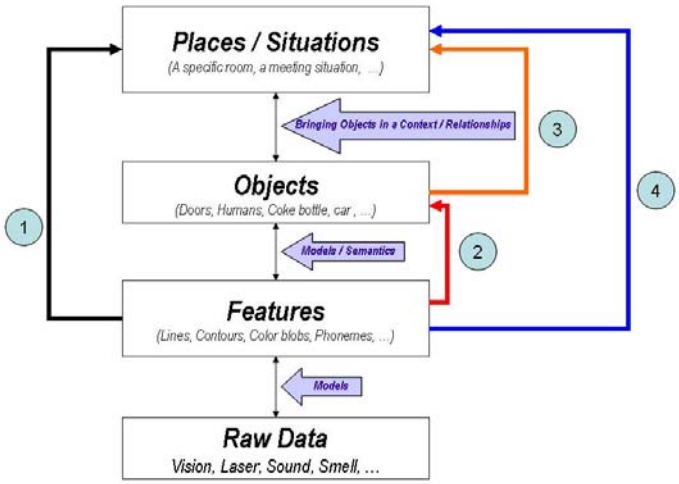


Figure 1. Schematic overview of activities in WP5.1. Overview of the research on space learning. (1)Appearance based topological model (UVA / EPFL) (2) Object detection (LAAS, EPFL) (3) Space models based on detected objects (EPFL) (4) Interaction with humans (KTH, UvA, UniBi)

In the process of building a hierarchical representation, clustering is a way to achieve conceptual representations (or 'classes') without human supervision. We will first describe work on clustering based on sensory data (involving work done in WP 5.1.1 and WP 5.1.2) in section 3.1. Then we present work on clustering based on detected objects (involving work done in WP 5.1.1 and WP 5.1.2) in section 3.2. Integration of local representations and categorical representations (the theme of WP 5.1.2) is natural if the categorical representations are obtained from data-driven clustering methods. However, if concepts are formed from human interactions, this integration is not trivial. We will describe interactions with humans in section 3.3.

3.1 From sensoric data to clusters

In the first year of the project we worked on building categorical and hierarchical representation of space by grouping sensory data. Two approaches developed in the first year are further refined in the second year:

Appearance (vision) based

The appearance based approach is based on grouping images. Omnidirectional cameras are used in order to have a wide view of the scene. Two omnidirectional camera systems are built, calibrated and tested; one is at UvA [Zivkovic05b] and the other one at EPFL. Work was also done on extending the existing techniques for omnidirectional camera calibration [Scaramuzza05].

Early work in the project has explored use of

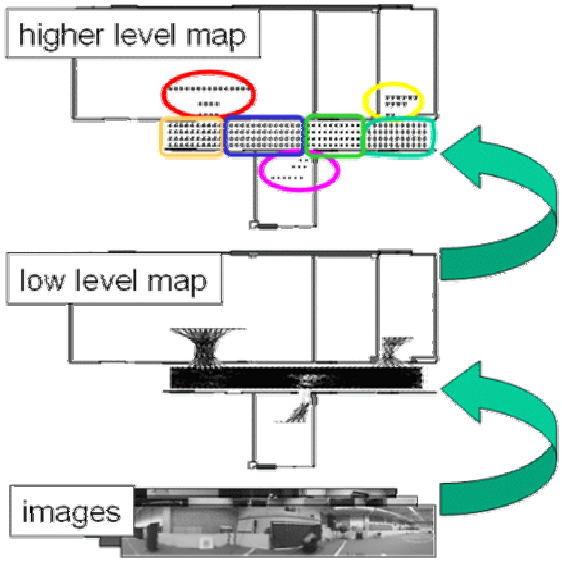


Figure 2. Hierarchical approach of UvA

visual 'fingerprint' similarities (EPFL) to group the images. In the second phase we focused on the similarity measure on the basis of local image features [Zivkovic05c] and epipolar constraints between the images (UvA). The set of images is regarded as a set of nodes of a graph and the similarity of the images define the edges of the graph, see figure. The size of the graph becomes large for large environments. Controlling the size of the graph was discussed in [Booij05]. This low level map (graph) contains, in a natural way, information about how the space in an indoor environment is separated by walls and other barriers. Images from a convex space, for example a room, will have many connections between them, and just a few connections to images from another convex space, for example a corridor, that is connected with the room via a narrow passage, for example a door. A 'graph cut' algorithm [Zivkovic05] was used to group the images from different spaces separated by narrow passages. The grouped images form the higher level representation of the space, see figure 2. On the basis of this research we answered the research question from 5.1.1. that local image properties (SIFT features, colored blobs) lead to a topological correct map of nodes indicating 'convex ' spaces.
The low level graph can be used also for the appearance based navigation and path planning since the epipolar geometry also enables estimating the relative positions between pairs of images. Within 5.1.2. we implemented an appearance based navigation module which is currently being tested in a joint effort of UvA and UniBi.
We showed that the graph method can be used for planning, but that a much more efficient planning can be obtained using the hierarchical representation as we demonstrated in [Bakker05].

Geometry based
At the end of year 1, users at KTH could do an online semantic labeling of a geometric map that is continuously build and updated by a SLAM method that is developed at KTH [Folkesson05]. In the second phase experiments were performed based on grouping laser range data. Categorization of spaces based on analysis of laser data using statistical moments were carried out at KTH.

3.2 From objects to clusters

Modeling the physical world in terms of the objects and the way they relate to each other is one highly intuitive method of interpreting space. Both spatial and semantic inferences are demonstrated in this work, from the model thus created. In this context, object modeling and recognition capabilities together with methods to detect spatial and semantic relationships between objects are required. A first project on this [Tapus05] used a probabilistic feature based simple object recognition system (color, texture). The current approach uses SIFT features for recognizing objects [Lowe04]. A hierarchy

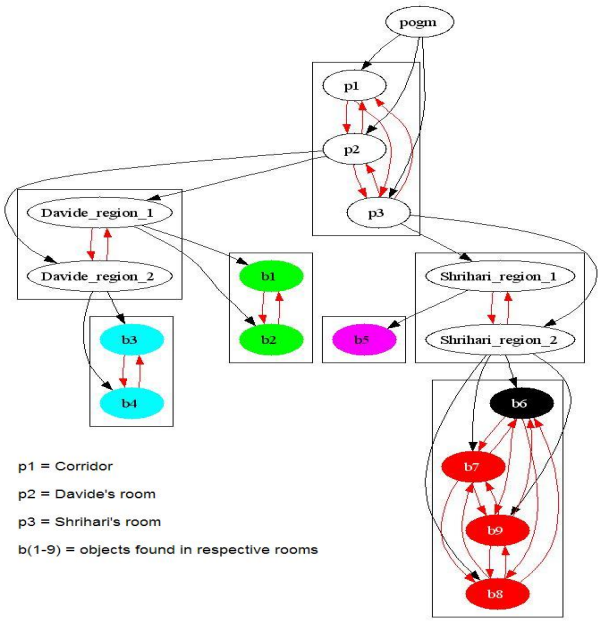


Figure 3 Graphical model of relations between objects and spaces (EPFL)

of graphs is used in which at the low level the nodes indicate objects, at a higher level the nodes represent regions (parts of space defined by collections of objects) and at the highest level the nodes indicate locations (rooms, corridor) (figure 3)

Also at LAAS an object detection system is used for space characterization [Cottret05]. The system consists of an omnidirectional vision system in collaboration with a pan-tilt-zoom camera. When moving in an indoor environment, the robot executes concurrently a pre-attentive process and an attentive one; the former extracts coarsely localized regions of interest (ROI) from the omnidirectional images. The latter analyzes more accurately each ROI by focusing the PTZ camera, grabbing a succession of viewpoints and generating an appearance based model of the region. Based on discriminant and invariant patches, this model provides a set of visual features which will be used to categorize region classes and to recognize region instances in next images.

3.3. Interaction with humans

The hierarchical representations described above are a result from a data driven approach, either directly form sensoric data or first detecting objects. A number of different theories on how spatial relations are acquired and represented by humans have been proposed throughout the years. According to McNamara [McNamara86] those theories can be grouped on the difference in dimensions of a) format (analog vs. propositional), b) functionality (spatial configuration vs. semantic or logical knowledge), c) structure (flat vs. strongly hierarchical), and d) contents (encoded information vs. procedural knowledge to compute information). McNamara used this categorization to design a psychological study on spatial representations, and came to the conclusion that a partially hierarchical model supported his findings most appropriately.
At KTH a a graph based model of the environment (see figure 4) is presented to incorporate the information that is given interactively by humans [Topp05]. The interaction is done on-line, since the assumption is that a “guided tour” is an appropriate way to give the user the possibility to personalize the robot’s general environment representation.

In the interactive map building it is important that the robot is able to follow the user and to distinguish the persons interacting with the robot. A number of papers were published [Klaassen05, Topp05, Zajdel05] on following and identification of users. The work on people following and recognition has a strong relation with RA2 and RA3. Joint research on social distances during following is planned in RA3.

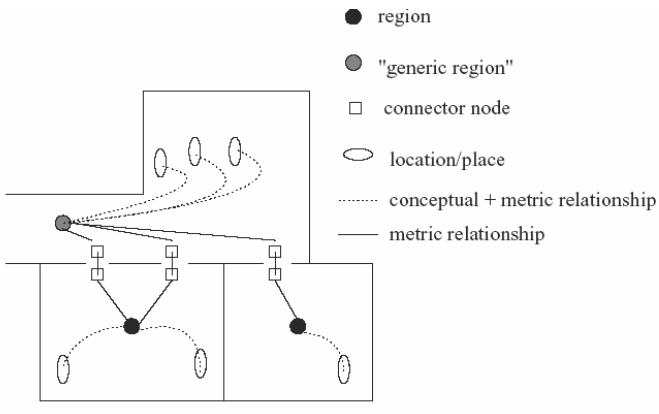


Figure 4. Relations between regions and locations as described in Topp05.

4 Future work

In the coming period we will continue working on the hierarchical space representations consisting of categorical representations at the top and sensoric representations at the bottom. Categorical identification of places and regions allows structuring of models into a hierarchy, and the integration of such models with local geometry or observations is considered central to design of a comprehensive model for spatial reasoning. The low level (sensoric) representation may either be image based, range sensor based but we need to further investigate different intermediate representations, such as objects or regions, which may be formed. Another question we need to address is how we achieve a consistent representation between the hierarchical representations derived from data and the representations derived from interactions with the humans. The role of the human is to steer the process of conceptualization. We need to investigate the formed categories meaningful to humans and how can human supervision aid the categorization.

5 References

5.1 Applicable documents

Bakker, B., Zivkovic, Z, and Kröse, B. , Hierarchical Dynamic Programming for Robot Path Planning, IEEE/RSJ International Conference on Intelligent Robots and Systems, Canada, 2005 (**in appendix**)

Booij O., Zivkovic Z., Krose B. , Pruning the image set for appearance based robot localization, Conference of the Advanced School for Computing and Imaging, the Netherlands, June 2005

Cottret, M, Devy, M : Exploring indoor environments using Attentive and Multi-Resolution Vision, Submitted to COGIS'O6, Symposium on Cognitive systems with Interactive Sensors,, Paris, Mars 2006 (**in appendix**)

M.Cottret, M.Devy. Active learning of local structures from Attentive and Multi-Resolution Vision. 9th Int.Symp. on Intelligent Autonomous Systems (IAS'2006), Tokyo (Japan), March 2006.

J. Folkesson, P. Jensfelt, and H.I. Christensen. Vision SLAM in the Measurement Subspace. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), April 2005.

Gradje Klaassen, Wojtek Zajdel, Ben Krose, Speech-based localization of multiple persons for an interface robot . IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) 2005.

Scaramuzza, D., Martinelli, A. and Siegwart, R., A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In the Proceedings of IEEE International Conference on Computer Vision Systems, ICVS 2006.

Scaramuzza, D., Martinelli, A. and Siegwart, R. , Practical and Accurate Omnidirectional Camera Calibration for 3D Reconstruction, Submitted to ICRA 2006.

Tapus, Adriana, Vasudevan, Shrihari and Siegwart, Roland, Towards a Multilevel Cognitive Probabilistic Representation of Space, In the proceedings of the International Conference on Human Vision and Electronic Imaging X, part of the IS&T/SPIE Symposium on Electronic Imaging 2005, 16-20 January 2005, CA, USA (HVEI 2005) (**in appendix**) .

Elin A. Topp and Henrik I. Christensen, Tracking for Following and Passing Persons, in Proc. of the IEEE International Conference on Intelligent Robots and Systems, August 2005, Edmonton, AB, Canada

E.A.Topp, H.Huettenrauch, H.I.Christensen, and K.Severinson-Eklundh, Acquiring a Shared Environment Representation, Human Robot Interaction '06 - HRI2006, Salt Lake City, Utah, USA. (**in appendix**)

W.Zajdel, Z.Zivkovic and B.Kröse, Keeping track of humans: have I seen this person before?, ICRA 2005, Barcelona, Spain, 2005

Zivkovic, Z, Bakker, B., and Kröse, B. , Hierarchical Map Building Using Visual Landmarks and Geometric Constraints , IEEE/RSJ International Conference on Intelligent Robots and Systems, Canada, 2005 (**in appendix**)

Z.Zivkovic, O.Booij, How did we built our hyperbolic mirror omnidirectional camera - practical issues and basic geometry, UvA technical report IAS-UVA-05-04, 2005.

Z. Zivkovic and B.J.A. Kröse. On matching interest regions using local descriptors - can an information theoretic approach help?. In Proc. British Machine Vision Conference, pages 50-58, 2005.

5.2 Other references

B. J. Kuipers. Representing knowledge of large-scale space. Technical Report TR-418 (revised version of Doctoral thesis), MIT Artificial Intelligence Laboratory, July 1977.

S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence, 99(1):21.71, 1998.

T.P. McNamara. Mental Representations of Spatial relations. Cognitive Psychology, 18:87–121, 1986.

D. G. Lowe. Distinctive image features from scale-invariant keypoints, Accepted for publication in the International Journal of Computer Vision, 2004

Appendix

In this appendix a selection of the Cogniron RA5.1 papers is reprinted. These papers are:

Hierarchical Dynamic Programming for Robot Path Planning,
Bakker, B., Zivkovic, Z, and Kröse, B. , University of Amsterdam
IEEE/RSJ International Conference on Intelligent Robots and Systems, Canada, 2005
Summary: This paper addresses the question how robot planning (e.g. for navigation) can be done with hierarchical maps. The paper shows that the developed method, based on Markov Decision Processes (MDPs) and dynamic programming (DP) is more efficient than standard DP for non-hierarchial MDP's because it reduces the state space for all levels in its hierarchy and it allows reuse of previously computed partial policies.

Exploring indoor environments using Attentive and Multi-Resolution Vision
Cottret, M, Devy, M., LAAS
Submitted to COGIS'O6, Symposium on Cognitive systems with Interactive Sensors,, Paris, 2006
Summary. This paper presents developments on detection and characterization functions, integrated on a companion robot equipped with an omnidirectional color camera and an PTZ camera with pan, tilt and zoom controls. When moving in a indoor environment, our robot executes concurrently a pre-attentive process and an attentive one; the former extracts coarsely localized regions of interest (ROI) from the omnidirectional images and the latter analyzes more accurately each ROI by focusing the PTZ camera.

Towards a Multilevel Cognitive Probabilistic Representation of Space
Tapus, Adriana, Vasudevan, Shrihari and Siegwart, Roland, EPFL
Proceedings of the International Conference on Human Vision and Electronic Imaging X, part of the IS&T/SPIE Symposium on Electronic Imaging 2005, 16-20 January 2005
Summary. This paper addresses the problem of perception and representation of space for a mobile agent. A probabilistic hierarchical framework is suggested as a solution to this problem. The world is viewed from a topological optic, in terms of objects and relationships between them. The hierarchical representation that is proposed permits an efficient and reliable modeling of the information that the mobile agent would perceive from its environment.

Acquiring a Shared Environment Representation
E.A.Topp, H.Huettenrauch, H.I.Christensen, and K.Severinson-Eklundh, KTH
Human Robot Interaction '06 - HRI2006, Salt Lake City, Utah, USA.
Summary Interacting with a domestic service robot implies the existence of a joint environment model for user and robot. This paper presents a pilot study that investigates, how humans present a familiar environment to a mobile robot. Results from this pilot study are used to evaluate a proposed generic environment model for a service robot.

Hierarchical Map Building Using Visual Landmarks and Geometric Constraints
Zivkovic, Z, Bakker, B., and Kröse, B. , University of Amsterdam
IEEE/RSJ International Conference on Intelligent Robots and Systems, Canada, 2005
Summary This paper addresses the problem of automatic construction of a hierarchical map from omnidirectional images. First a low-level map is built that consists of a graph in which relations between images are represented. For this we use a metric based on visual landmarks (SIFT features) and geometrical constraints. Then we use a graph partitioning method to cluster nodes and in this way construct the high-level map. Experiments on real data show that meaningful higher and lower level maps are obtained, which can be used for accurate localization and planning.

Hierarchical Dynamic Programming for Robot Path Planning*

Bram Bakker and Zoran Zivkovic and Ben Kröse
Intelligent Autonomous Systems group, Informatics Institute
University of Amsterdam
Kruislaan 403, 1098 SJ, Amsterdam, The Netherlands
{*bram, zivkovic, krose*}*@science.uva.nl*

***Abstract*— This paper addresses the question how robot planning (e.g. for navigation) can be done with hierarchical maps. We present an algorithm for hierarchical path planning for stochastic tasks, based on Markov Decision Processes (MDPs) and dynamic programming, that is more efficient than standard dynamic programming for "flat" MDPs, because it reduces the state space for all levels in its hierarchy and it allows reuse of previously computed partial policies. This computational advantage comes at the cost of some extra memory and overhead to represent and coordinate the hierarchical system, and in some cases somewhat longer paths to target locations. We demonstrate the method on artificially generated MDP data, and on real robot data from our vision-controlled robot navigating in an office environment.**

***Index Terms*— Planning, Dynamic Programming, Hierarchical methods, Multiresolution methods, Mobile Robots**

I. Introduction

Many researchers (e.g. [9], [14], [13]) have proposed a hierarchy of maps to represent mobile robots' environments. The central idea in all those proposals is that it makes sense to represent large environments at different resolutions, or levels of abstraction, simultaneously.

Low-level, local maps may be used to represent, for example, individual rooms in a large building in great spatial detail, without having to represent the spatial relationships to locations in other rooms. Such a low-level map may be used for navigation to precise target locations within an individual room, without having to worry about other rooms.

Higher-level, abstract maps may be used to represent larger areas of a building, for instance as a graph connecting rooms and corridors, without representing the exact spatial relationship of individual locations within rooms and corridors. Such a high-level map may be used to construct abstract plans to navigate from one room to another, without having to worry about exact spatial details, and it may be used for fruitful communication with humans, because the elements in the high-level map (e.g., the nodes in the graph) can be made to correspond to concepts that make sense to humans (rooms, corridors). A hierarchy of maps may thus facilitate communication, map building, and planning based on the maps.

This paper addresses the question how planning (e.g. for navigation) can be done with hierarchical maps. We present an efficient algorithm for hierarchical path planning for stochastic tasks. The maps are formalized as Markov Decision Processes (MDPs), and planning tasks as dynamic programming problems (e.g. [2], [3]). We describe how a hierarchy of MDPs can be constructed and solved using a hierarchical variation of value iteration. This approach is related to other hierarchical approaches based on MDPs [12], [4], [5], [1].

We show that path planning done in this way can be much more efficient than when one uses one monolithic, "flat" MDP, especially when the state space becomes large and when paths must be planned to many possible target locations. This computational advantage, which is particularly important in the context of real-time robot planning, comes at the cost of some extra memory and overhead to represent and coordinate the hierarchical system, and in some cases somewhat longer paths to target locations. Since our hierarchical method is based on standard MDPs, it can be used for planning tasks other than path planning as well.

The next section briefly reviews MDPs and the corresponding, standard dynamic programming solution method called value iteration. Section III describes our hierarchical MDP and dynamic programming method, and compares it to related work. Section IV describes experiments based on artificial data, illustrating the method and comparing it empirically to standard flat dynamic programming. Section V describes similar experiments, but now on data from our real mobile robot [15]. Section VI presents conclusions and possible future work.

II. MDPs and dynamic programming

For path planning in possibly stochastic domains, robot maps are commonly formalized as Markov Decision Processes (MDPs), such that the planning task becomes a dynamic programming problem [2], [3]. This formalization is appropriate for such robot planning tasks because first of all it takes into account uncertainty in the execution of actions (i.e. stochastic state transitions). Secondly, policies are computed for the entire state space, which is necessary when action outcomes are uncertain. Thirdly, the resulting policies are optimal in the sense that they lead to lowest expected cost (e.g. distance travelled). Finally, it allows for straightforward inclusion of cost factors other than distance travelled, such as energy consumption and obstacle avoidance (or other factors if the task goes beyond navigation).

*This work is supported by the EU FP6-IST2020 "Cogniron" project.

A. MDPs for path planning

An MDP $\mathcal{M}$ is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$. $\mathcal{S}$ is a finite set of states s, some of which may be terminal states. $\mathcal{A}$ is a finite set of actions a, whose availability may depend on the state. $\mathcal{R}$ is the reward function that defines the immediate reward r. The Markov property for state and reward representations requires that the state and reward at time $t+1$ depend only on the state and action at time t, such that the following holds:

$$p\{s_{t+1} = s, r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, \ldots, s_0, a_0\} = p\{s_{t+1} = s, r_{t+1} = r \mid s_t, a_t\}. \quad (1)$$

Given that the Markov property holds, we can define $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, the state transition function that describes the probability $p(s'|s, a)$ that the system will move from state s to s' after performing the action $a \in \mathcal{A}$. Successors are defined as those states s' for which $p(s'|s, a) \neq 0$. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ defines the immediate real-valued reward $r(s, a, s')$ when action a is taken in state s and the transition to s' is made. $r(s, a, s')$ may be a negative cost function, e.g. based on distance travelled between s and s'.

For path planning, states corresponding to target locations become target locations. Furthermore, we assume that the set of actions in a state s simply corresponds to the set of successors s'. That is, at the current state the robot can choose from the successor states where to go next; but it only arrives at the successor state with probability $p(s'|s, a)$.

A policy is defined as a mapping $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The objective is to find an optimal policy π^* that maximizes the expected, possibly discounted, future cumulative reward, or expected return. In the finite horizon case, i.e. when each episode ends at some time T (e.g. because a target is reached) and without discounting of future rewards, this corresponds to:

$$V^*(s) = \max_\pi V^\pi(s) = \max_\pi E\left[\sum_{t=0}^{T} r(s_t, \pi(s_t), s_{t+1}) | \pi, s_0 = s\right] \quad (2)$$

for each state s. The expectation operator $E[\cdot]$ averages over reward and stochastic transitions.

B. Value iteration

MDPs with known state transition functions and reward functions can be solved optimally using dynamic programming methods. Dynamic programming iteratively computes the value function $V(s)$, which represents the estimate of the expected return attainable from each state. It is guaranteed to converge to the optimal value function $V^*(s)$, which represents the maximum attainable expected return (eq. 2). One well-known method, value iteration, repeatedly sweeps through the state set of the MDP and, in the undiscounted, finite horizon case, updates each state's value according to

$$V(s) \leftarrow \max_a \sum_{s'} p(s'|s, a)[r(s, a, s') + V(s')] \quad (3)$$

until the largest change in value of any of the states, Δ, is smaller than a small constant threshold. After convergence, the optimal policy is followed by simply taking the greedy action in each state:

$$\pi^*(s) = \arg\max_a \sum_{s'} p(s'|s, a)[r(s, a, s') + V^*(s')]. \quad (4)$$

III. Hierarchical MDPs and dynamic programming

A. Hierarchical MDPs

We define two types of MDPs making up the complete hierarchical system. They are both derived from a given flat MDP. The first type, $\mathcal{M}^n$, a tuple $\langle \mathcal{S}^n, \mathcal{A}^n, \mathcal{T}^n, \mathcal{R}^n \rangle$, represents the given MDP at a particular level of abstraction. n indexes the level in the hierarchy, N is the number of levels in the hierarchy (decided by the designer). The given flat MDP is $\mathcal{M}^0$. $\mathcal{M}^n$ for $n \geq 1$ is constructed from $\mathcal{M}^{n-1}$ by clustering the states in $\mathcal{S}^{n-1}$. A possible clustering method is briefly described below and in more detail in [15]. Each cluster of states from $\mathcal{S}^{n-1}$ becomes a single state in $\mathcal{S}^n$.

The state transition function $\mathcal{T}^n$, which defines $p(s^n_m|s^n_k, a^n)$, is constructed by determining, for all states $s^{n-1}_i \in \mathcal{S}^{n-1}$ that belong to one cluster and correspond to state s^n_k, the cluster labels of successors s^{n-1}_j that belong to other clusters and correspond to states s^n_m. The probability $p(s^n_m|s^n_k, a^n)$ is estimated by averaging over the corresponding probabilities $p(s^{n-1}_j|s^{n-1}_i, a^{n-1})$. Similarly, the reward function $\mathcal{R}^n$, which defines $r(s^n_k, a^n, s^n_m)$, is constructed by determining, for each state transition from s^n_k to s^n_m, the corresponding $r^{n-1}(s^{n-1}_i, a^{n-1}, s^{n-1}_j)$, and averaging over them. As before, the action set $\mathcal{A}^n$ is defined as the set of successors s^n_m for each state s^n_k.

The second type of MDPs making up the complete hierarchical system is defined only for $n \geq 1$ and is denoted by $\mathcal{M}^{n-1}_{s^n_k, s^n_m}$. $\mathcal{M}^{n-1}_{s^n_k, s^n_m}$ is an MDP that represents the lower level $(n-1)$ task of navigating from higher level (n) state s^n_k to state s^n_m. It is essentially a subset of $\mathcal{M}^{n-1}$, whose states are only those states $s^{n-1}_i \in \mathcal{S}^{n-1}$ that correspond to state s^n_k, combined with those states $s^{n-1}_j \in \mathcal{S}^{n-1}$ that are successors of states s^{n-1}_i and that correspond to state s^n_m. The states s^{n-1}_j are terminal states. $\mathcal{A}^{n-1}_{s^n_k, s^n_m}$, $\mathcal{T}^{n-1}_{s^n_k, s^n_m}$, and $\mathcal{R}^{n-1}_{s^n_k, s^n_m}$ follow directly from $\mathcal{M}^{n-1}$. $\mathcal{M}^{n-1}_{s^n_k, s^{n-1}_j}$ is a special case intended for navigating to a specific low-level state s^{n-1}_j *within* a high-level state s^n_k. $\mathcal{S}^{n-1}_{s^n_k, s^{n-1}_j}$ contains only those states $s^{n-1}_i \in \mathcal{S}^{n-1}$ that correspond to state s^n_k, and s^{n-1}_j is the only terminal state.

Figure 1 depicts schematically how a 2-level hierarchy of MDPs is derived from a simple flat MDP using our method.

B. Hierarchical value iteration

With q states and a maximum of m admissible actions for any state, standard value iteration (for flat MDPs) requires for each sweep through the state space at most $O(mq)$ operations in the deterministic case and $O(mq^2)$ operations in the stochastic case. Because of this, it can become very slow with large numbers of states. Furthermore, there is no obvious way to reuse value functions obtained with previously

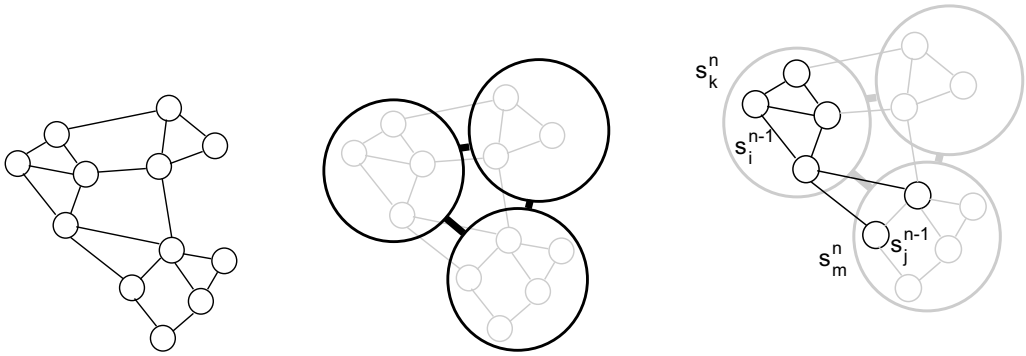


Fig. 1. Illustration of a 2-level hierarchy of MDPs. Left: The flat, low-level MDP $\mathcal{M}^0$. Middle: $\mathcal{M}^1$ is constructed by clustering the states of $\mathcal{M}^0$ and averaging over state transition probabilities and rewards. Right: $\mathcal{M}^{n-1}_{s^n_k, s^n_m}$ is an MDP which represents the low-level task of navigating from high-level state s^n_k to state s^n_m.

selected target states for new target states. Our hierarchical approach can have benefits on both of these issues. Thus, we use value iteration, but adapt it to our hierarchical framework.

The hierarchy of MDPs defined in the previous section allows us to efficiently compute a value function/policy for the entire state space to every possible target state of the original, flat MDP $\mathcal{M}^0$. For a specific low-level target state, the higher level target states in $\mathcal{M}^n$ are determined in which this low-level target state lies. At their own levels, they become terminal states. Next, the path planning task is performed from the highest level down, using value iteration at each level. State transitions from s^n_k to s^n_m dictated by a value function at level n are modeled by the appropriate $\mathcal{M}^{n-1}_{s^n_k, s^n_m}$ and subsequently planned, again using value iteration. In this way, the complete planning task to a low-level target state is planned recursively.

In contrast to standard, flat value iteration, when paths must be planned to multiple target states, the algorithm can in many cases reuse value functions computed for earlier target states. This is because high-level value functions remain the same as before when high-level target states do not change for this new low-level target state, or because certain high-level state transitions may remain the same as before so lower-level value functions realizing those state transitions can be reused. A second advantage over standard flat value iteration is that the state spaces at all levels are reduced, leading to fewer operations per sweep through the state set and faster convergence.

Algorithm 1 provides pseudocode for the complete hierarchical planning method, assuming that a flat MDP $\mathcal{M}^0$ is given and MDPs $\mathcal{M}^n$ up to $n = N - 1$, the top level, have already been constructed using a clustering algorithm. MDPs $\mathcal{M}^{n-1}_{s^n_k, s^n_m}$ have not yet been constructed; they are constructed only when needed. The specific algorithm described in pseudocode assumes that the lowest level ($n = 0$) target state is given by the user. The algorithm can also be used if the user provides higher-level target states, as in a command like "go to the kitchen". In that case no target states are identified at levels below the level of the assigned high-level target state,

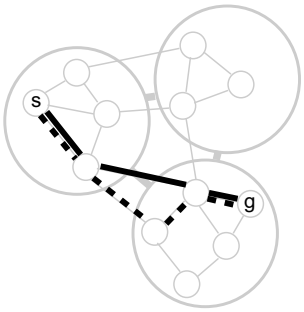


Fig. 2. Illustration of a case when our hierarchical planning method may lead to longer paths from a start state s to a target state g. The solid line is the optimal path. The dashed line is the path computed by the hierarchical method, which optimally goes to the bottom high-level state, and within the bottom high-level state optimally goes to the target state.

and the algorithm ends when paths are learned that reach the high-level target state.

The algorithm's increased efficiency comes at the cost of some extra memory and overhead to represent and coordinate the hierarchical system. Furthermore, in some cases the system can converge to slightly longer paths to target locations than standard flat value iteration. This situation can arise because low-level value functions which are optimal with respect to reaching the next high-level state from the current high-level state are not always optimal with respect to reaching the final target state (see figure 2 for an illustration). It is also possible that high-level reward functions are not completely accurate because they average over low-level rewards and ignore rewards within high-level states. The latter problem could be remedied by updating expected rewards for (s^n_k, s^n_m) state transitions in $\mathcal{M}^n$ with the values computed by their corresponding $\mathcal{M}^{n-1}_{s^n_k, s^n_m}$.

Importantly, however, given that the Markov property holds for the $\mathcal{M}^n$ at all levels in the hierarchy ($0 \leq n < N$), i.e.

$$p\{s^n_{t+1} = s^n, r^n_{t+1} = r^n \mid s^n_t, a^n_t, r^n_t, s^n_{t-1}, \ldots, s^n_0, a^n_0\} = p\{s^n_{t+1} = s^n, r^n_{t+1} = r^n \mid s^n_t, a^n_t\}, \quad (5)$$

$s_g^0 \leftarrow$ new target state for $\mathcal{M}^0$
for all $0 < n < N$ **do**
  $s_g^n \leftarrow$ determine target state for $\mathcal{M}^n$
end for
$V^{N-1} \leftarrow$ Solve($\mathcal{M}^{N-1}$, $N-1$).

function Solve($\mathcal{M}$, n)
if value function V for this $\mathcal{M}$ does not exist **then**
  while $\delta > \Delta$ (a tiny threshold) **do**
    for all $s \in \mathcal{S}$ **do**
      $V_{new} \leftarrow \max_a \sum_{s'} p(s'|s,a)[r(s,a,s') + V(s')\ ]$
      if $|V_{new} - V(s)| > \delta$ **then**
        $\delta \leftarrow |V_{new} - V(s)|$
      end if
      $V(s) \leftarrow V_{new}$
    end for
  end while
end if
if $n > 0$ **then**
  for all $s \in \mathcal{S}$ **do**
    if $s = s_g^n$ **then**
      if $V^{n-1}_{s_g^n, s_g^{n-1}}$ does not exist **then**
        Construct $\mathcal{M}^{n-1}_{s_g^n, s_g^{n-1}}$ from $\mathcal{M}^{n-1}$
        $V^{n-1}_{s_g^n, s_g^{n-1}} \leftarrow$ Solve($\mathcal{M}^{n-1}_{s_g^n, s_g^{n-1}}$, $n-1$)
      else
        $s^* \leftarrow \arg\max_a \sum_{s'} p(s'|s,a)[r(s,a,s') + V(s')]$
        if V^{n-1}_{s,s^*} does not exist **then**
          Construct $\mathcal{M}^{n-1}_{s,s^*}$ from $\mathcal{M}^{n-1}$
          $V^{n-1}_{s,s^*} \leftarrow$ Solve($\mathcal{M}^{n-1}_{s,s^*}$, $n-1$)
        end if
      end if
    end if
  end for
end if
Return V

Algorithm 1: Pseudocode of our hierarchical value iteration algorithm.

this hierarchical dynamic programming method is guaranteed to converge to value functions and policies which correspond to valid paths to the target location from any state from which the target location can be reached. Moreover, it converges to policies which are *optimal given the hierarchy* (cf. [5]), i.e. it will do the best it can possibly do given the pre-defined clustering of states of $\mathcal{M}^0$. Enforcing the Markov property at all levels means, essentially, that the clustering algorithm may not give the same label to different, unconnected sets of states at any level $0 < n < N$. We discuss below (and in [15]) a clustering algorithm that enforces this criterion.

C. Related work

Other studies that use some form of hierarchical planning similarly emphasize the benefit of reducing state spaces through hierarchy. These include logic-based planners [6] which use very different representations and solution methods but similarly compute abstract plans before computing detailed plans. Furthermore, within robotics there is work on multiresolution motion planning [8] which is similar in spirit to our hierarchical planning approach but uses A* planning techniques, does not take into account uncertainty in action outcomes, plans open loop policies from specific starting states rather than closed loop policies for all states, and does not exploit our particularly effective reuse of previously computed policies.

There is some work on other variations of hierarchical dynamic programming [11], [12]. Similar to our work, these studies use a form of state abstraction to compute coarse policies before working out the details in fine-grained policies. However, the coarse policies are not valid for the whole state space and they are used only as initial approximations, and they are subsequently refined incrementally until the lowest-level policy is found; thus losing some of the advantages of high-level policies, such as reuse of previously computed policies.

Within reinforcement learning, which can be viewed as approximate, sampling-based dynamic programming techniques, there is a growing literature on exploiting hierarchy [4], [5], [1]. Some of those studies [4], [5], [1] similarly allow reuse of previously computed value functions. However, all of these methods are model-free learning methods, which means they they require many (often millions of) interactions with the environment before they learn a task, and they cannot exploit the more powerful dynamic programming methods afforded by models. Furthermore, most of those methods do not exploit state abstraction which is very beneficial for hierarchical methods (but see [4], [1]), or they do not have the same performance guarantees as our method (but see [5]).

IV. Experiment based on artificial data

A. The low-level map and clustering method

We first demonstrate our hierarchical method on artificial data, allowing us to investigate it under strictly controlled conditions. The data set, i.e. the lowest level MDP $\mathcal{M}^0$, is generated as follows. First, 4000 points are randomly generated and placed in a simulated 2D 10 m by 10 m world. All points becomes states. Next, Euclidean distances are computed between all states. States are connected by possible transitions (edges) if and only if the Euclidean distance $d(s,s')$ between them is less than 1 m, and the reward (cost) of the transition is $r(s,s') = -d(s,s')$. An action a' takes the system from a state s to the desired successor s' with a randomly generated probability $0.5 < p(s'|s,a') < 1$; the remaining probability mass is evenly distributed over all other successors.

We construct a 2-level hierarchy. To cluster the states of $\mathcal{M}^0$ into higher level states for $\mathcal{M}^1$, we use the normalized graph cut algorithm from graph theory [7], explained in more detail for this particular type of application in [15]. Essentially, the algorithm cuts edges so as to arrive at unconnected subsets of the overall graph (MDP in our case), each of

which becomes a cluster. It minimizes the number of edges it has to cut to yield the desired number of clusters, and simultaneously maximizes the number of edges within each cluster. The normalized graph cut algorithm is appropriate for our problem for multiple reasons. First of all, it will often provide cluster boundaries at intuitive places, e.g. at narrow transitions (doors) between larger spaces (rooms). Secondly, if it fulfils its objective of cutting only one edge to distinguish between two clusters, our hierarchical method will lead to optimal paths between all states in the two clusters (because both hierarchical methods and flat methods need to pass through this one transition). Thirdly, by its nature it enforces the Markov property for the higher level states. We use the normalized graph cut algorithm based on distances between states, with 20 clusters, and subsequently construct $\mathcal{M}^1$ as described in section III-A.

B. Results

We use the hierarchical value iteration method described in Algorithm 1 to compute policies to 100 randomly selected states. The results are compared to flat value iteration (section II-B), performed on the flat, low-level MDP $\mathcal{M}^0$.

Table I summarizes the results. Most important is the total number of value updates until convergence for all 100 target locations planned for, for both the flat and the hierarchical method (*#ValUpd*). Even with just one target location that must be planned for (*#ValUpd1*), the number of value updates for the hierarchical method is significantly lower than for the flat method: $494,652$ vs. $3,351,160$. This is due to the hierarchical method having lower numbers of states at both the higher and the lower level, leading to fewer states to update and faster convergence.

The difference in required value updates, and the difference in runtime (*RunTime*) quickly becomes very large as more target locations must be planned for, thanks in part to the hierarchical method's frequent reuse of previously learned value functions (*#Reuse*). These numbers suggest that when the MDPs become much larger, the hierarchical method would remain feasible when the flat method would no longer be feasible. However, the average maximum likelihood path length (*AvPath*) for the hierarchical method is 5.71, which is higher than the flat method's (optimal) value of 5.27.

V. Experiment based on real robot data

A. The low-level map and clustering method

Next, we turn to real robot data [15]. The data are obtained using our mobile robot equipped with an omnidirectional camera navigating in an office environment. The use of vision was dictated by the requirement to operate in relatively realistic, natural environments. The data consist of 234 panoramic images taken at regularly spaced intervals in an office environment (see figure 3). The observed environment essentially consists of 3 rooms, connected through a single corridor. This database constitutes an appearance-based map, and each image becomes a state in $\mathcal{M}^0$.

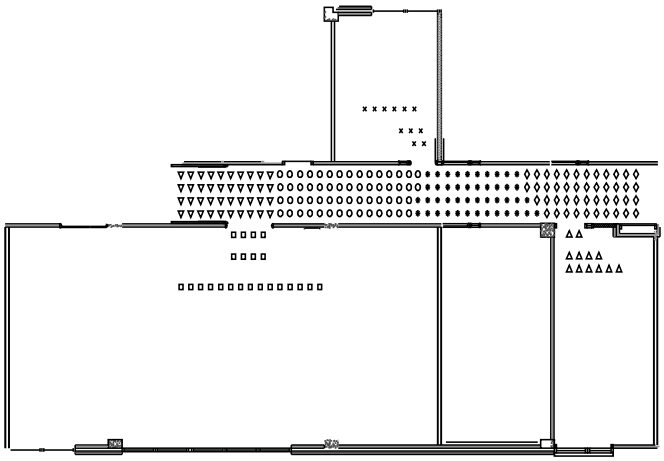


Fig. 3. Bird's eye view of the office environment, with the locations where images were captured. Different symbols indicate different clusters, and thus different high-level states.

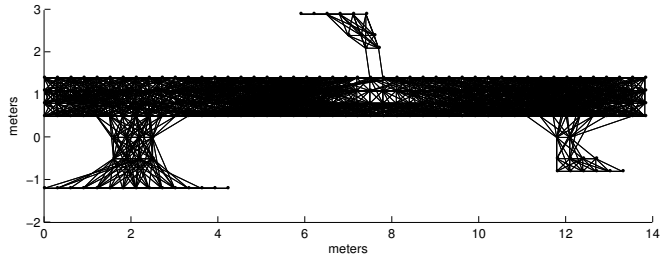


Fig. 4. The MDP extracted from the images taken in the office environment.

Given the database of images, from each image a set of distinctive local image features is extracted, using the SIFT method [10]. SIFT features in different images are subsequently matched, and matches are possibly rejected due to geometric constraints (see [15] for details). If a sufficient number of matches remain, the images are sufficiently similar and are assumed to come from similar locations. In this case, it is possible to navigate from one state to the other, using homing based on matched features. Therefore, an edge (possible transition in the MDP) is added between the two states. In this experiment, we do not compute more accurate estimates of Euclidean distance or transition probability between two states, so we simply assume $r(s, s') = -1$ and $p(s'|s, a') = 1$ if an edge exists and 0 otherwise. Figure 4 shows the resulting MDP $\mathcal{M}^0$ within the office environment.

Again, $\mathcal{M}^0$ is clustered using the normalized graph cut algorithm, this time using 7 clusters. Figure 3 shows the clustering obtained in this case. Note how separate rooms become separate clusters. The corridor is also divided into a number of clusters.

B. Results

We do a similar experiment as in the previous section. We use our hierarchical value iteration method to compute policies to all possible states, comparing the results to flat

TABLE I

PLANNING RESULTS FOR BOTH DATA SETS. *FDP* IS THE FLAT DYNAMIC PROGRAMMING METHOD, *HDP* IS THE HIERARCHICAL DYNAMIC PROGRAMMING METHOD. *#ValUpd* IS THE TOTAL NUMBER OF VALUE UPDATES REQUIRED TO COMPLETE THE TASK. *#Sweeps* IS THE TOTAL NUMBER OF SWEEPS THROUGH THE DATA SET. *RunTime* IS THE TOTAL RUN TIME. *#Reuse* IS THE NUMBER OF TIMES A VALUE FUNCTION IS REUSED. *#ValUpd1* IS THE NUMBER OF VALUE UPDATES FOR THE FIRST TARGET LOCATION. *AvPath* IS THE AVERAGE MAXIMUM LIKELIHOOD PATH LENGTH TO THE TARGET.

Data	Method	#ValUpd	#Sweeps	RunTime	#Reuse	#ValUpd1	AvPath
Artificial	FDP	482,115,441	120,559	4,684 sec	0	3,351,160	5.26
Artificial	HDP	4,207,787	20,454	50 sec	1,886	494,652	5.61
Real Robot	FDP	445,496	1,912	2 sec	0	1,631	3.48
Real Robot	HDP	36,328	895	1 sec	1,622	807	4.13

value iteration.

Table I summarizes the results. As in the previous section, it is apparent that the hierarchical method has clear computational advantages over the flat method. For the first target location that must be planned for, the number of value updates for the hierarchical method is 807 as opposed to the flat method's $1,631$. To plan for all target locations, the number of value updates is $36,328$ for the hierarchical method vs. $445,496$ for the flat method. This experiment shows the viability of our hierarchical method on data from a real robot and its possible advantage over the standard, flat method, which could be especially relevant in real-time robot applications.

VI. CONCLUSIONS

The results of this paper show the feasibility and promise of the hierarchical mapping and planning approach for robot navigation. Robot environments were formalized as a hierarchy of MDPs and subsequently solved using a variation of dynamic programming. Our hierarchical dynamic programming approach leads to significant savings in terms of the number of value updates required for convergence, compared to standard, flat dynamic programming, especially when the state space becomes large and when navigation policies must be computed to many target locations. This advantage is due to reduced state spaces and efficient reuse of previously computed value functions. This may come at the cost of somewhat longer paths to target locations, but given that the Markov property holds for the low-level and high-level MDPs, value functions and policies are computed which always correspond to possible paths to the goal and which are optimal given the hierarchy.

Future work includes testing the algorithms in a real robot experiment (as opposed to the simulation experiment based on real robot data used now), investigating hierarchies with levels $N > 2$ for very large, realistic tasks. Furthermore, the hierarchical method could be adapted for tasks other than path planning, including infinite horizon discounted reward problems. It would also be interesting to investigate ways to diminish the hierarchical method's disadvantages, notably by adapting the clustering method to reduce the disadvantage of longer path lengths, and by making estimates of higher-level state transition probabilities and rewards more accurate. Finally, these methods or similar methods may be combined with methods for dealing with partial observability (POMDPs) and multiple agents.

REFERENCES

[1] B. Bakker and J. Schmidhuber. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Kröse, editors, *Proceedings of the 8-th Conference on Intelligent Autonomous Systems, IAS-8, Amsterdam, The Netherlands*, pages 438–445, 2004.

[2] D. P. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, Belmont, MA, 1995.

[3] Joachim M. Buhmann, Wolfram Burgard, Armin B. Cremers, Dieter Fox, Thomas Hofmann, Frank E. Schneider, Jiannis Strikos, and Sebastian Thrun. The mobile robot RHINO. *AI Magazine*, 16(2):31–38, 1995.

[4] P. Dayan and G. E. Hinton. Feudal reinforcement learning. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5: Proceedings of the 1992 Conference*, San Mateo, Ca., 1993. Morgan Kaufmann Publishers.

[5] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[6] C. Galindo, J.-A. Fernandez-Madrigal, and J. Gonzalez. Improving efficiency in mobile robot task planning through world abstraction. *IEEE Transaction on Robotics*, 20 (4):677–690, 2004.

[7] J.Shi and J.Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Anlysis and Machine Intelligence*, 22(8):888–904, 2000.

[8] S. Kambhampati and L. S. Davis. Multiresolution path planning for mobile robots. *IEEE Journal of Robotics and Automation*, 2 (3):135–145, 1986.

[9] B. J. Kuipers. Representing knowledge of large-scale space. Technical Report TR-418 (revised version of Doctoral thesis May 1977, MIT Mathematical Department), MIT Artificial Intelligence Laboratory, July 1977.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 (2):91–110, 2004.

[11] J. K. Peterson. Path planning in analog valued obstacle arrays using hierarchical dynamic programming and neural networks. In *Proceedings of the Artificial Neural Networks in Engineering (ANNIE91) Conference*, pages 789–794, 1991.

[12] C. Raphael. Coarse-to-fine dynamic programming. *IEEE Transactions on Pattern Anlysis and Machine Intelligence*, 23(12):1379–1390, 2001.

[13] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[14] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Simultaneous localization and map building: A global topological model with local metric maps. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2001*, 2001.

[15] Z. Zivkovic, B. Bakker, and B. Kröse. Hierarchical map building using visual landmarks and geometric constraints. Technical Report submitted to IROS2005, http://www.science.uva.nl/~bram/, Informatics Institute, University of Amsterdam, 2005.

Book Title
Book Editors
IOS Press, 2003

Active learning of local structures from Attentive and Multi-Resolution Vision [1]

Maxime Cottret [a] and Michel Devy [a,2]
[a] *LAAS-CNRS, Toulouse, France*

Abstract. In order to execute tasks, a robot needs a complex visual system to cope with the detection, characterization and recognition of places and objects. This paper presents on-the-way developments on detection and characterization functions, integrated on a companion robot equipped with an omnidirectional color camera and an PTZ camera with pan, tilt and zoom controls. When moving in a indoor environment, our robot executes concurrently a pre-attentive process and an attentive one; the former extracts coarsely localized regions of interest (ROI) from the omnidirectional images. The latter analyzes more accurately each ROI by focusing the PTZ camera, grabbing a succession of viewpoints and generating an appearance based model of the region. Based on discriminant and invariant patches, this model provides a set of visual features which will be used to categorize region classes and to recognize region instances in next images.

Keywords. local structures, attention, active vision, multi-resolution.

1. Introduction

In ten or twenty years, everyone will have a companion robot at home to perform some simple domotic tasks. The cognitive functions of such a robot are currently studied by several robotics team, in the framework of the COGNIRON european project. Several testbeds are integrated , with many sensors used to learn knowledges, to control navigation and manipulation tasks and to interact with the user.

The embedded system of a companion robot, integrates functional and decisional functions, especially to learn knowledges on the environment, autonomously or interactively with a user. In order to generate motions, several spatial models must be learnt; at the more abstract level, the environment is described by places a posteriori labelled by the user according to some semantical informations (labels like kitchen, room, living-room...) and connected in a topological graph. In order to control the execution of these motions, the robot will require other knowledges to locate itself, i.e. discriminant landmarks linked to places where they have been detected or characteristic images for every area. Classical free space representations like occupancy grids could be generated in cluttered areas.

[1]This work is funded by the FP6 european project COGNIRON: http://www.cogniron.org
[2]Correspondence to: Michel Devy, LAAS-CNRS, 7 avenue du Colonel Roche, 31077 Toulouse Cedex 04, FRANCE. Tel.:+33 5 61 33 63 31 ; Fax:+33 5 61 33 64 55 ; E-mail: michel.devy@laas.fr

A companion robot has to recognize, locate and grasp objects. First, a user could ask the robot to bring him an object (a pen, a glass, the TV control box…) or to execute autonomously a more complex task that has been learnt a priori like *Water plants*. Moreover, the companion robot could execute by itself some simpler tasks, planned from analysis of the user behaviour to recognize its intention, e.g. *Bring an object the user is looking for*.

Many learning functions required on a companion robot, have been studied in the robotics community, taking advantage of the Bayesian framework to deal with uncertainties and noisy measurements, e.g. the automatic construction of stochastic maps [13] to represent landmarks detected from visual or laser data [5,15], the autonomous learning of topological models from panoramic images, including the place categorization [17,14], the automatic construction of a geometrical model for a 3D object to be grasped by the robot … The robot could learn these models interactively with the user, e.g. it could construct navigation maps following an operator who guides the robot during the home tour and who gives a semantical label to every place [16], ot it could construct the model of an object that the operator moves so that every aspect will be viewed [4].

This paper gives preliminary results on a more autonomous strategy applied by our robot to learn local knowledges, from two different visual modalities: a color omnidirectional camera and an active camera with pan, tilt and zoom parameters that can be controlled by the system (PTZ camera). When learning the environment map from panoramic images, eventually following an operator, our robot will also detect and track regions of interest (namely, ROIs): a ROI is a *salient* image region, i.e. a region that can be easily distinguished in a local neighborhood. These ROIs correspond to local structures of the environment, discriminant and invariant enough to be tracked in an image sequence and to be exploited in further motions, either planare objects like quadrangles [5], or isolated 3D objects laid on the ground (chairs...) or on tables (glasses, plates...). From high resolution images acquired by the PTZ camera, the robot will build an invariant representation for such local structures, fusing data acquired from several view points and asking the operator to label this entity for further interactions.

So, two processes will be executed asynchronously to extract local knowledges; the panoramic images will be analyzed in a pre-attentive process, while high resolution images will be acquired and analyzed by the attentive process. The section 2 gives a short state of the art and details our general approach. In the section 3, the pre-attentive process is described: it provides a list of salient regions detected on low resolution images. The section 4 is devoted to the attentive process, for the categorization of local structures extracted from high resolution images acquired on every ROI. The section 5 gives preliminary results on the integration of such visual processes on our experimental robot. Finally, the section 6 will summarize our contribution and our current works.

2. General approach

In recent years, numerous authors in the Vision community, have provided contributions on cognitive vision, visual learning or object categorization. Some authors [1] propose neuronal approaches to analyze the whole image, but many others exploit only invariant features or patches. Numerous regions descriptors have been proposed [9]:

Scale Saliency Patches [7] are mostly used in object recognition and categorization [3,2], while SIFT features [8] have become very popular in robotics [17]. J.Ponce et C.Schmid [11,12,10] build representations from patches inspired from the classical Harris points [9]... In these contributions, an object class is characterized by a set of invariant patches extracted from numerous images acquired from different viewpoints. The described entities correspond generally to a single aspect of a 3D object (the back side of a car, a right side of an airplane) like in [3]. Typically these authors do not address the localization problem, which is mandatory for a robot which will interact with the described object (landmark-based navigation, object grasping, visual servoing). Moreover, images are simple, with isolated objects during the learning step; only one passive visual modality is exploited to acquire images; only view-based representations are built, assuming that all patches extracted from every image belong to the same object.

Here we take advantage of active and multi-focal vision to improve the learning of local structures extracted and tracked from sequences of complex images. At this step, the learning task is autonomous; interactions with the operator will be considered in a next step, in order to add semantics to the model. The robot moves along a known trajectory, and during this motion, executes concurrently two asynchronous visual processes.

The pre-attentive process detects salient regions from low resolution images. It exploits quick operators [6] to create from every low resolution image, a *saliency map*, segmented by a clustering method, in order to generate Regions of Interest, memorized in a ROI list. The attentive process will learn only from these regions. It is executed asynchronously, taking as input, the ROI list; it controls the PTZ camera to acquire high resolution images focused successively on the salient regions. For every one, it generates an appearance based representation, more precisely a set of discriminant scale-invariant *patches* [7]. Then, this representation will be used to categorize local structures in object classes, and to recognize object instances in next images [3].

The pre-attentive process provides image-based information only for short-term memorization; on the contrary the attentive process generates knowledge, for long-term memorization. Such a structure could correspond to an isolated object; but generally, it will be difficult to obtain a right object segmentation from an image; a user interaction will be required later for the final interpretation.

3. Pre-attentive vision

This process takes place before any decision. The robot need to detect highly salient regions in order to choose where to focus. Numerous definitions of visual saliency exist in the literature, like *Jagersand*, *Itti* or *Kadir*'s ones. As our robot will interact with humans, it has to focus on the same discriminant features than humans. Therefore, our pre-attentive vision system is based on the saliency detector developed by L. Itti [6]: according neurobiologic studies on primates, particular locations in the scene only are treated in the visual cortex, in a bottom-up process and without a priori knowledge. These studies also showed that three types of visual components take most part in the pre-attentive vision : color, intensity and orientations. Thus, after extraction of these components into three maps, they are put in competition and only highly informative locations remain leading to a saliency map of the visual field.

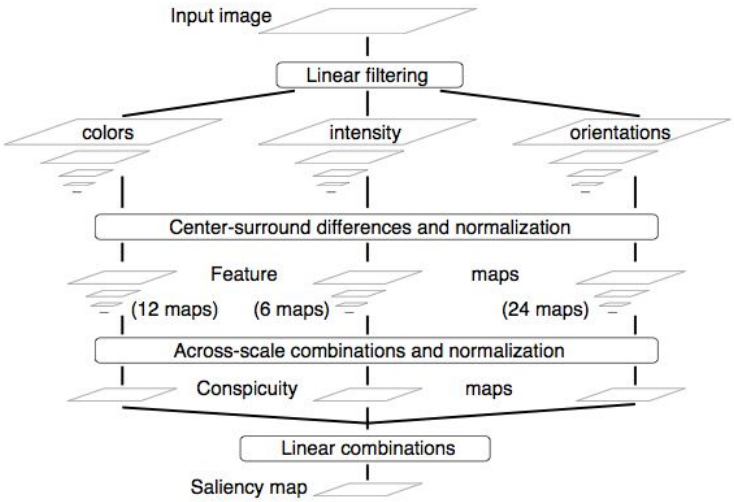


Figure 1. Saliency map construction

Saliency map construction

The figure 3 summarizes the pipeline of operations from the input image to the saliency map. Three filters are applied to extract intensity, color and orientation discontinuities at several levels of the input image resolutions. Three feature maps (one for each filter) are computed at several image resolutions, using center-surround comparisons, in order to model the sensitivity difference in the human visual system between the center and the edge of the retina.

$$I(c,s) = |I(c) \ominus I(s)|$$
$$RG(c,s) = |(R(c) - G(c)) \ominus (G(s) - R(s))| \; BY(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|$$
$$O(c,s,\theta) = |O(c,\theta) \ominus O(s,theta)|$$

with $c \in \{2,3,4\}$ and $s = c + \delta$, $\delta \in \{3,4\}$ So, the first step consists in computing gaussian pyramids, with scale $\sigma \in [0..8]$, to detect discontinuities on the intensity $I(\sigma)$, the color $RG(\sigma)$ for $G - R$, $BY(\sigma)$ for $B - Y$, and the orientation $O(\sigma,\theta)$, using Gabor filters with angles $\theta = 0°, 45°, 90°$ and $135°$. Feature maps are normalized and integrated to form a single conspicuity map for each visual attributes I, $G - R$, $B - Y$ and O. A non-linear normalization is applied to each conspicuity map to amplify peaks of contrasts relative to noise in the background. In the final stage, conspicuity maps are combined at a given scale (here $\sigma = 3$) to produce a single saliency map S:

$$\overline{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(I(c,s))$$
$$\overline{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} |\mathcal{N}(RG(c,s)) + \mathcal{N}(BY(c,s))|$$
$$\overline{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} (\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(O(c,s,\theta)))$$
$$S = \frac{\mathcal{N}(\overline{I}) + \mathcal{N}(\overline{C}) + \mathcal{N}(\overline{O})}{3}$$

The figure 2-b presents the saliency map and intermediate intensity, color and orientation conspicuity maps, computed on a low resolution image acquired in a kitchen: let us note that the more salient local structure is the coffee machine.

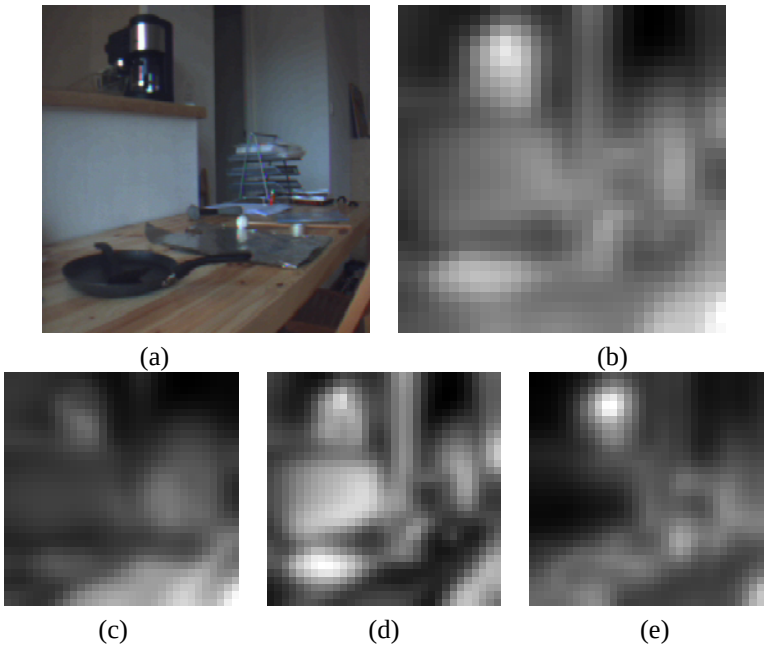


Figure 2. (a) input image, (b) saliency map, (c) intensity conspicuity, (d) color conspicuity, (e) orientation conspicuity (from Gabor filters)

Regions of interest extraction.

Our selection algorithm takes the saliency map as a statistical map for the purpose of Monte-Carlo selection in order to extract N from the most salient points. It uses rejection sampling, i.e. randomly reject points according a randomly biased threshold. The resulting points cloud, distributed over the most salient regions of the map, is then clustered in R^3 (spatial position and saliency) using the K-means method, with K a priori fixed; we are evaluating a Mean Shift algorithm to avoid the classic al problem on the K selection. The regions of interest are stocked as the barycenter - weighted by the saliency - , the bounding box of the points from a resulting class and the mean saliency.

The pre-attentive process.

Whereas the robot goes through its environment, ROIs are detected using the omnidirectional camera and stored with a timestamp and a position tag. The treatment of ROIs through the attentive process is done according the mean saliency weighted by: Ð distance from the current position of the robot : the ROI must be in the field of view of the PTZ camera. Ð distance from the current timestamp : even if most of the environment is almost static, some objects can be moved, involving a change in the interest of the ROI. So, we considere that the saliency of a ROI become insignificant with time. ROIs are erased once treated or if they become too old (the time threshold is manually fixed).

4. Attentive vision

Once the robot "knows" where salient structures rely in its environment, it can address planning or decision making problems of higher abstraction, in order to build a long-term model for these entities. By now, our development are focussed on the learning of appearance-based models; later a geometrical ones will be built from the same image sequences.

Figure 3. Extracted features on focalized images

Our companion robot evolves in a cluttered environment, where plenty of partially occulted objects, hardly separable from the background, can be found. Therefore the robot's model has to be incrementally built and/or updated. It appears that global object description like PCA or moments are not adapted for our purpose (it code the whole shape and appearance of the object in one description). Therefore a local structure is modelled by a constellation of features.

Scale Saliency Patches

T.Kadir and M.Brady [7] introduce a local invariant descriptor based on information and scale-space frameworks. Each point of the input image x, is characterized by a descriptor vector D that take values $(d_1...d_r)^t$. A local probability density function $p_{x,s}$ (pdf) is estimated using histogram on a circular window and for a range of scale s (radius). The entropy $H_{x,s}$ of this pdf is: $H_{x,s} = \Sigma_i p_{x,s}(d_i) log_2(p_{x,s}(d_i))$

The local maximum of entropy over the scale gives the patch characteristic scale $\bar{s}$ .Thus, a local feature is given by a pixel s with its scale $\bar{s}$ and a score S

$$S = H_{x,\bar{s}} \Sigma_i |p_{x,\bar{s}}(d_i) - p_{x,\bar{s}-1}(d_i)|$$

In order to reduce the feature number, extracted features are clustered in R^3 (spatial positions and scale): a cluster of Scaled Salient Patches has a local support (nearby points with similar scores and scales). Each cluster must be sufficiently distant from all other in. Thus, a local structure corresponds to such a cluster, given by the barycenter of a selected local support (mean pixel, mean scale and mean score): the figure 3 presents Scaled Salient Patches extracted from focalized views acquired on the coffee machine seen on the global view in figure 2.

The attentive process.

The attentive process focus on the most salient ROI given by the pre-attentive process. For each ROI, Scaled Salient Patches are extracted and stored as a thumbnails clus-

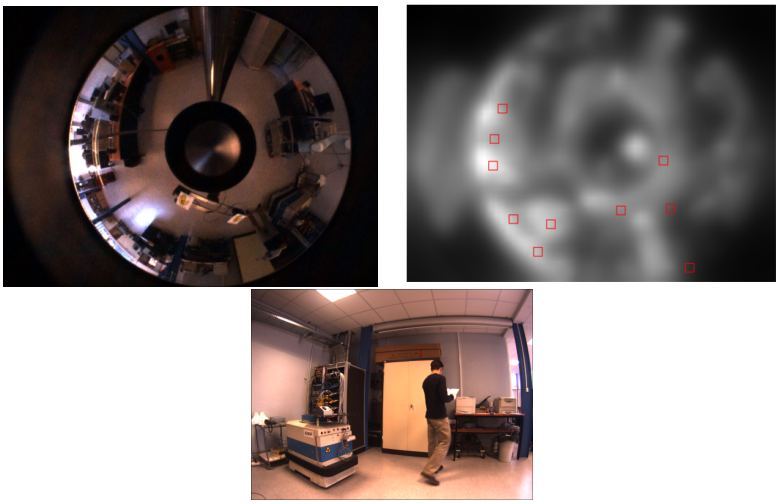


Figure 4. (top left) input from omnidirectional camera; (top right) saliency map (boxes mark centers of ROIs); (bottom) focalized image on the bottom right salient region

ter with a position tag and label them as "location" of the environment. As describe in [10], we use pixel correlation for thumbnails matching and so "location" recognition : there is recognition if most features in the new viewpoint are matched with the location's one. Then, the location's list of thumbnails is updated by merging the new viewpoint's ones and eliminates those that never match. A coarse 3D localization using the position of the different viewpoints, can also be computed because the robot motion is known.

5. Preliminary results

By now, operations involved in the pre-attentive and attentive processes have been evaluated on several images manually acquired at home. Only the pre-attentive process is integrated on a robot. Panoramic images are acquired, moving the robot with the joystick; salient regions must correspond to same areas in the environment; we are studying how to match salient regions extracted from a sequence of panoramic images. The omnidirectional camera is coarsely calibrated, so we can compute a 3D direction from each ROI's center; on figure 4, salient ROIs are extracted from the panoramic image on the top left, and a focalized image is acquired from the PTZ camera pointed toward the more salient region, here around the clear cupboard behind a person.

6. Conclusion

This paper has proposed an original approach to learn local structures that a robot could detect by itself indoor. These models will be used to understand how the environment is structured: they could be combined either with a topological map built by another learning process, to recognize places (such an entity characteristic for such a place), or with a metrical stochastic map to locate the robot. Current works are devoted to two problems:

1) how to take into account contextual knowledge, mainly to make easier the segmentation of the local structures (for example, assuming that these structures are on large uniform planar faces of the environment: ground, walls ...)? and 2) how to categorize and recognize objects, from their patch representations built from several viewpoints, using the Bayesian framework proposed in [2]?

References

[1] N. Dohuu, W. Paquier, and R. Chatila. Combining structural descriptions and image-based representations for image, object and scene recognition. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2005.

[2] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of objects categories. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2004.

[3] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Int. Conf. on Vision and Pattern recognition (CVPR)*, 2003.

[4] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through action: Initial steps towards artificial cognition. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[5] J.B. Hayet, F. Lerasle, and M. Devy. Visual landmarks detection and recognition for mobile robot navigation. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Vol.II, pages 313–318*, 2003.

[6] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence(PAMI), Vol. 20, No. 11*, pages 1254–1259, 1998.

[7] T. Kadir and M. Brady. Scale, saliency and image description. *Int. Journal on Computer Vision (IJCV)*, 2001.

[8] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision, 60, 2*, pages 91–110, 2004.

[9] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2001.

[10] J. Ponce, S. Lazebnik, F. Rothganger, and C.Schmid. Toward true 3d object recognition. In *Proc. AFRIF/AFIA Conf. Reconnaissance de Formes et Intelligence Artificielle (RFIA)*, 2004.

[11] F. Rothganger, S. Lazebnik, C.Schmid, and J. Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *Proc. IEEE Int. Conf. on Vision and Pattern Recognition (CVPR)*, 2003.

[12] C. Schmid. Constructing models for content-based image retrieval. In *Proc. IEEE Int. Conf. on Vision and Pattern Recognition (CVPR), Vol II*, pages 39–45, 2003.

[13] R.C. Smith and P. Cheeseman. On the representation of spatial uncertainty. In *Int.Journal on Robotics Research*, 1987.

[14] A. Tapus, S. Heinzer, and R. Siegwart. Bayesian programming for topological global localization with fingerprints. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[15] S. Thrun and Y. Liu. Multi-robot slam with sparse extended information filers. In *Proc. Int. Symp. of Robotics Research (ISRR)*, 2003.

[16] E.A. Topp and H.I. Christensen. Tracking for following and passing persons. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.

[17] Z. Zivkovic, B. Bakker, and B. Krose. Hierarchical map building using visual landmarks and geometric constraints. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.

Towards a Multilevel Cognitive Probabilistic Representation of Space

Adriana Tapus, Shrihari Vasudevan and Roland Siegwart

Ecole Polytechnique Federale de Lausanne (EPFL)
Autonomous Systems Lab
1015 Lausanne, Switzerland
{Adriana.Tapus, Shrihari.Vasudevan, Roland.Siegwart}@epfl.ch

ABSTRACT

This paper addresses the problem of perception and representation of space for a mobile agent. A probabilistic hierarchical framework is suggested as a solution to this problem. The method proposed is a combination of probabilistic belief with "Object Graph Models" (OGM). The world is viewed from a topological optic, in terms of objects and relationships between them. The hierarchical representation that we propose permits an efficient and reliable modeling of the information that the mobile agent would perceive from its environment. The integration of both navigational and interactional capabilities through efficient representation is also addressed. Experiments on a set of images taken from the real world that validate the approach are reported. This framework draws on the general understanding of human cognition and perception and contributes towards the overall efforts to build cognitive robot companions.

Keywords: cognitive mapping, hierarchical topological representation, probabilistic belief

1. INTRODUCTION

Interpreting and understanding a scene from the environment beyond single object recognition is a hard task. Humans use various sensory cues to extract crucial information from the environment. This is processed in the cortex of the brain in order to obtain a high-level representation of what has been perceived. Intuitively, it appears that humans represent knowledge in a hierarchical fashion. With a view of having robots as companion of humans, we are motivated towards developing a knowledge representation system along the lines of what we know about us. While recent research has shown interesting results, we are still far from having concepts and algorithms that interpret space, coping with the complexity of the environment.

Most of the related research on formalizing levels of abstraction in literature can be found in cognitive science (e.g. hierarchical representation and reasoning with knowledge), mobile robotics and networking. Space representation plays an important role in any cognitive and autonomous intelligent system. The idea of cognitive maps (i.e. the animal internal representation of space) was introduced for the first time by Tolman in [17]. Significant progress has been made since the seminal papers by Kuipers [7, 8] where the cognitive maps are described as a body of knowledge representing large scale space. In his work, a spatial semantic hierarchy is suggested, which represents space at different levels of abstraction and attempts navigation using such a representation (see [9]). An approach similar to the previous one can be found in [10] where a hierarchical multi-resolution space representation is addressed. Voicu uses landmarks and associations between them to construct a cognitive map of a large environment in [19]. The information from this cognitive map is then used for path planning and exploration. The authors of [14] use a hierarchical hidden Markov model (HHMM) to learn the route between two labs. The higher level states are the more abstract/distinct ones like corners and intersections. The lower level states represent intermediate positions. All the above mentioned works seem to capture a hierarchical representation of space with a navigational flavor in them.

The work done by Brezetz et al in [1] bears close resemblance to our work presented here. It assumes the presence of an even ground with objects on it. Range images are obtained and segmentation of these images yields semantic information contained in them. The information is further used to extract spatial relationships between objects. Motivation for this work has also originated from beyond the robotics community. The problem being addressed in

this paper has found great interest in the Computer Science and Geographical Information Systems communities as well. The work by Papadias and Theodoridis in [13] suggests different topological and directional relationships between objects based on the concept of minimum bounding rectangles (MBR's). They make use of "R-Trees" to represent information. A different variation of the "R-Tree" was conceived by the authors of this work as well. However graphs were found to be more general, more suited to our long term goals and more extensible. This explains the reasoning behind our work. Grini et al have done something similar in [4]. They however address the problem of deriving more complex and sound inferences and look into complexity issues regarding it. They also visualize a multi-resolution framework of various spatial relationships. The work in [5] explores similar concepts in a different context – content based image analysis. They use absolute and relative spatial relations to describe inter-relationships between objects. They also compare the spatial-relationship identification between objects, when performed by humans with the software tool they developed for it. This helped to identify what spatial relationships should be present and also points out the limitations that one would face (unable to describe many relationships) when using 2D image processing as compared to 3D image processing – something we also noticed.

The remainder of this paper is organized as follows. We present in Section 2, the way objects from the environment are modeled and the method used for performing the recognition. In Section 3, a framework to represent objects occurring in the environment and inter-relationships between them, as a topological map, is suggested. Experimental results are presented in Section 4. Section 5 concludes the paper with a discussion of the proposed approach and further research directions.

2. OBJECT MODELING AND RECOGNITION

Modeling the physical world in terms of the objects present in it and the way they relate to each other is one highly intuitive method of interpreting space. Both spatial and semantic inferences are demonstrated in this work, from the model thus created. In this context, object modeling and recognition capabilities together with methods to detect spatial and semantic relationships between objects are required. A simple probabilistic feature based object recognition system has been used here. The system has been kept simple in terms of the features used. This work demonstrates various concepts using single colored boxes in a non-cluttered setting. It must be emphasized that object modeling and recognition are not the themes of this paper. However, they are critical to realizing the overall concept presented here.

2.1. Object Modeling

Figure 1 depicts different low-level features that can be extracted from the visual information used to model the objects. Broadly, features may be classified as color, shape, edges, etc. and several methods can be employed to represent them. In our system, we used only color and shape, thereby keeping it in very simple. More details about the extraction of low-level visual features can be found in [16].

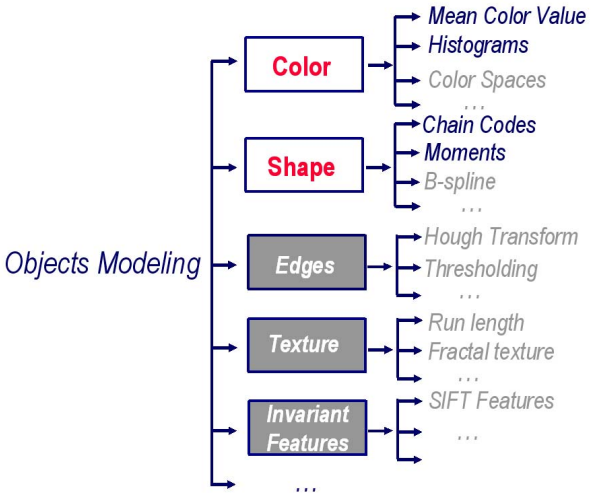


Figure 1: Objects Modeling: low-level features representation

A brief description of the features employed in our approach is given below:

A. Mean Color Value

The Hue Saturation Value (HSV) color space was used. The hue represents the color, the saturation represents the "purity" of the color and the value denotes the total power of the spectrum. All three values are normalized to the (0, 1) range for ease of use. The HSV color space is obtained by a non-linear transformation of the RGB color space. This color space was preferred due to its intuitive appeal over most other standard color spaces. The means of the hue, saturation and value components were computed.

B. RG-Chromaticity histograms

The RG Chromaticity Space is a two dimensional color space with no color intensity information. Each pixel represents the contribution of the red and green components. It is derived from the RGB color space as follows:

$$r = \frac{R}{(R + G + B)} \quad (1)$$

$$g = \frac{G}{(R + G + B)} \quad (2)$$

As each component of the RGB is normalized, the blue contribution can be easily computed, whenever required, as

$$b = 1 - (r + g) \quad (3)$$

The advantage of this space is that changing light intensities in the environment will not cause a change in the basic color of the object. A histogram with 100 bins for each component (i.e. r, g and b) of the image was formed.

C. Freeman's Codes (Chain Codes)

The shape of an object is detected by using the Freeman's code (chain code) [3] method as illustrated in [16]. The chain code models an object shape as a series of directed unit-pixel line segments. The result is a sequence of symbols that represents the inner boundary of the object/region under consideration.

D. Moments

Moment descriptions of regions describe a normalized gray level/binary image as a probability density of a 2 dimensional random variable. The moments are computed as shown in [16]. They are based on the principle of the moments of order (p+q) given by the formula:

$$m_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p \; j^q \; f(i, j) \quad (4)$$

This is translation, rotation and scaling dependant. Translation invariance is obtained by using centralized moments. To achieve scaling invariance, scaled central moments are introduced and rotation invariance is incorporated by choosing the coordinate axes such that the (1+1)th order central moment is zero. Essentially, a group of seven translation, rotation and scaling invariant moments as discussed in [6] were used.

2.2. Object Recognition

As mentioned previously, object recognition is realized as a probabilistic feature matching process. The recognition methodology can be understood as an essentially color-based one. Simple polyhedral and single colored objects in uncluttered environments are used to demonstrate the concept proposed here. Prior to the recognition process, a database of object (the model objects) features is constructed. The color and the shape features are extracted and stored using the representations described above. The recognition process proceeds in the following manner:

- **Color Thresholding**
- **Feature Extraction**

- **Feature Matching** – The extracted features are compared with those of the object models in the database. Dissimilarity measures are computed. These are used to arrive at a belief measure of how closely the region resembles the object with which it is compared, on the basis of the particular feature.
- **Belief Computation** – Probabilities obtained from the feature matching process are combined to produce a single "belief" value that represents the degree of similarity between the region and the object (model) under consideration. Similar measures for other object models are found. The region is then inferred to be an object of the type represented by the object model for which the highest belief measure is obtained.

3. OBJECT GRAPH MODELS: CONSTRUCTION, UPDATING AND INFERENCE

The main significance of this work is to establish a correspondence between animals (humans) and robots. By understanding how animals (humans) navigate and build their own spatial representation, the observed phenomena are applied in robotics. In order to have a robust and reliable framework for navigation (i.e. in order to move within an environment, manipulate objects in it, avoid undesirable mishaps (collisions), etc.) space representation, perception, localization and mapping are all needed. The work of Tolman [17] showed that space is modeled as mental or cognitive maps. Tolman's model advocates that animals (rats) don't learn space as a sequence of movements; instead, the animal's spatial capabilities rest on the construction of maps which represent spatial relationships between various objects encountered in the environment. This has led to the concept of topological representation of space. Later, O'Keefe and Dostrovsky [11] discovered the hippocampal place cells (i.e. cells whose firing pattern is dependent on the location of the animal in the environment), which led to the idea that the hippocampus works as a cognitive map of space [12].

Next, we discuss a framework for describing an approach similar to the phenomena encountered in the hippocampal place cells and perirhinal cortex (i.e. part of the brain where the recognition of objects is performed).

3.1. OGM Construction and Update

Physical space is viewed in terms of objects and relationships between them. Thus, the environment is represented as a graph structure – which is referred to as an Object Graph Model (OGM). Knowledge representation using OGM involves: object recognition (described in section 2.2) and graph representation. After performing the object recognition step, a set of objects and object descriptions are obtained. These are used in order to model/describe a scene from the environment by using a graph representation. The existence of the object has an associated belief, which is computed from the beliefs of its features (i.e. from object recognition step). Object descriptions include color, shape, centroid coordinates, bounding boxes, and so on. Each object is represented as a node in the graph. The object descriptions are used to check for the existence of every possible relationship between each pair of objects (using binary relationships). Identified relationships are represented as directed arrows between two nodes. Each relationship also has an associated belief which depends on how well (unambiguously) the two objects satisfy the criteria that define the existence of a particular relationship apart from the beliefs in the existence of the objects involved. The resulting graph structure, so formed, is what we refer to as the Object Graph Model (OGM).

The OGM's previously formed must be updated with change in scene description. Three kinds of update operations are requisite:

A. New OGM

Each new scene observed is compared with all previously modeled scenes which have been modeled using the OGM. There is a data association problem involved here, in identifying two corresponding scenes. Two metrics are used to approach this, (1) the number of objects the two scenes share in common and (2) the number of these objects obeying the same relationships (on the vertical plane only, as horizontal plane relationships may change). Each of these metrics produces a belief measure, which are in turn combined. The resulting belief is a measure of how similar the current scene is, with respect to a previously modeled scene. Depending on this measure and a threshold that is preset, either a new OGM is constructed for the current scene or the previously modeled scene is updated as explained below.

B. Belief Update

When objects are repeatedly viewed across scenes (i.e. different views which correspond to the same modeled scene), the belief in their existence should increase. On the other hand, the belief in the existence of objects, not observed across scenes, should decrease. This is similar to the approach adopted in [18]. The same trend should also be applicable for relationships between objects. A data association problem involved in identifying

"corresponding" objects between a previously modeled scene and the scene under consideration is present here. This is overcome by taking a simple experimental scenario wherein two objects do not look exactly alike (e.g. they differ in either shape / color).

C. Add/Delete objects

When a new scene corresponds to a previously modeled scene (i.e. thus already existing as an OGM), a provision is required by which the new objects in the current scene are incorporated in the OGM while un-observed objects have their beliefs reduced. If their belief goes below a certain threshold – they are removed from the OGM as the belief in their existence (i.e. in that scene) has become negligibly small.

3.2. Example – OGM Representation of a scene

Let us illustrate the construction of an OGM with an example (see Figure 2). Figure 2 a) illustrates a possible scene in the environment and it is composed of seven objects. These are: one table, three mugs represented with the orange color and by the symbol *mi* (i =1,2,3) and three green boxes of different sizes, denoted by *bj* (j =1,2,3). The relative positions (e.g. above/below etc.) between the objects has been used to represent the relationships between them. In the topological framework, the objects are the nodes and the relationships between the objects are the edges. Thus the OGM of the current scene can be constructed as shown in Figure 2 b).

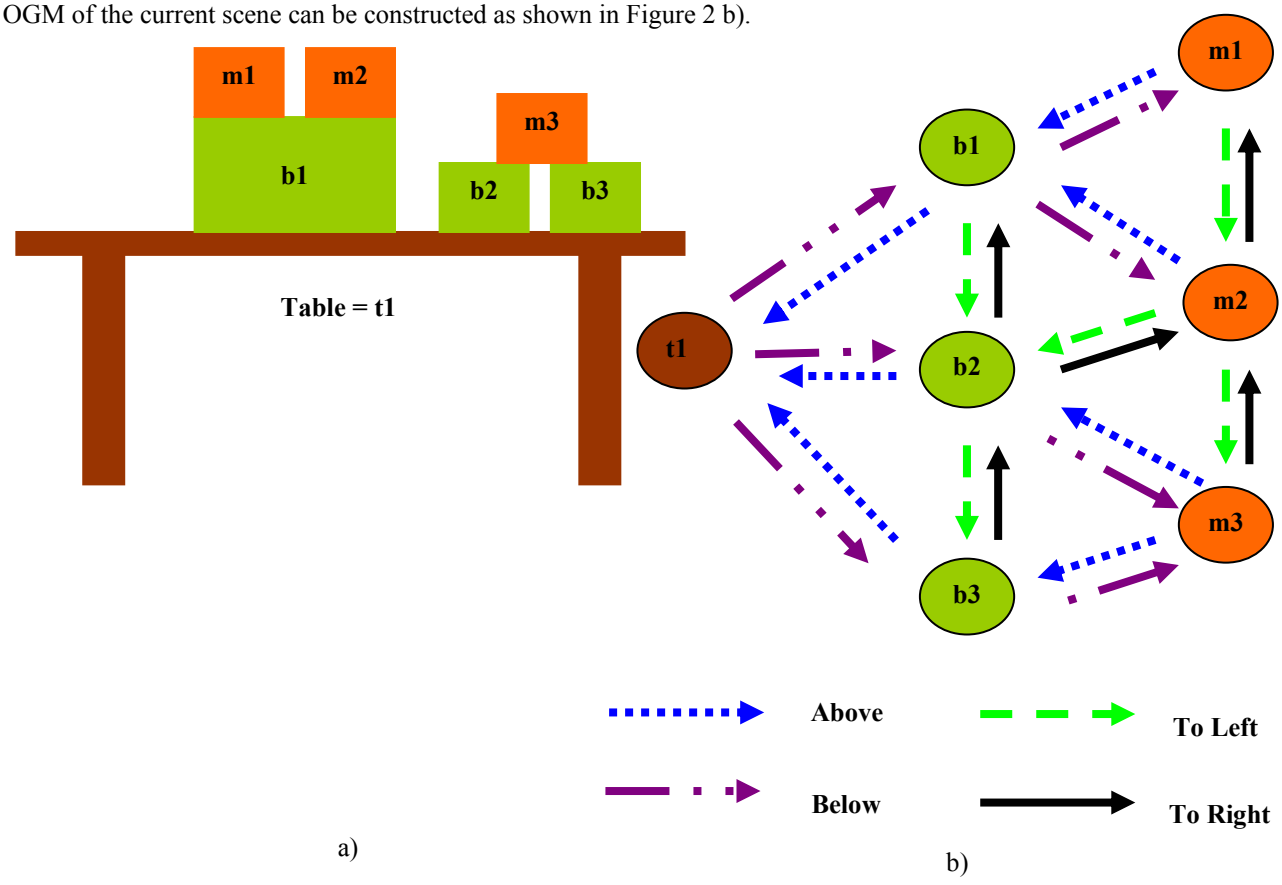


Figure 2: Scenario Representation. a) Table (t1) with Boxes (bi) and Mugs (mi) on it; b) Object Graph Model (OGM) for the example shown in Figure a).

3.3. Spatial Relations

Spatial relations are required in order to make meaningful and useful inferences about the objects in the scene. This work uses both directional and topological relationships between objects. A more comprehensive repertoire of relationships is possible if the image processing and object recognition are done in 3D. Directional relationships include those like the North (N), South (S), East (E) and West (W). Egenhofer, in [2], points out the existence of as many as nine meaningful topological relationships between objects. The use of 2D image processing limited us to use

only those like overlap and disjoint. It must be noted that horizontal plane relations are relative to the line of sight and hence cannot be relied on, to hold good, for more than the scene in which they are observed. Vertical plane relations are more persistent in that the robot (which models the environment) is assumed not move in the vertical plane. Relationships between objects also have associated beliefs. These beliefs are computed by taking into account the probability of the relationship holding between the two objects given their features and the probabilities of the existence of each of the objects. It may be computed using the following expression:

$$P(R_{A,B}) = P(R|A,B)\ P(A)\ P(B) \qquad (5)$$

where R is a binary relationship and A and B are the objects under consideration. The belief value is intuitive in that the belief in the presence of a relationship between two objects also takes into account the "clarity of the evidence" of the relationship existing between two objects.

3.4. Inference

The inference methods used here are based on binary relationships between objects. When objects are recognized, their bounding boxes are used to compute the existence and belief of these relationships. The following types of inference have been conceived in the system: purely spatial, purely semantic, spatio-semantic and multi OGM. Purely spatial inference tests for specific/all possible spatial relationships between objects (e.g. find all objects related to object *A* or how are objects *A* and *B* related). Semantic inference is realized by making queries on properties of the objects like shape and color (e.g. find all objects of a given color / shape). A hybrid inference, the spatio-semantic one, is obtained by combining the previous two types queries (e.g. find all existing relationships between all *red* colored objects in a scene). Finally, multi OGM inference makes queries across scenes (i.e. each scene being modeled by a separate OGM) (e.g. find all *red* colored objects between two different scenes). In our system, inference is realized by looking up an already computed exhaustive set of relationships for a particular one – this is in effect a graph search. More complex inference may be obtained by "chaining" inference results to obtain higher level complex results of greater use in practical situations.

4. EXPERIMENTAL RESULTS

For the experiments, a digital firewire color camera SONY DFW-VL500 has been used. The camera has a resolution of 640x480 pixels at 25Hz. The camera is mounted horizontally on the BIBA robot, a fully autonomous mobile robot.

4.1. Object Recognition

The test setup was the following: The training set included a single image of each of the possible objects used for the experiments. The robot extracted the four features from a test set of 15 images. The images have been taken from different angles and the distance from the robot to the objects was almost the same throughout the experiment. This simplification could be omitted by using scale invariant features. The recognition output was classified into the following categories: object identified correctly, data association problem between objects of the same color, false positives, merging of objects and completely missing the presence of an object. As pointed out in section 2, we extracted only low-level visual features to represent an object just for simplicity of the model, so that the OGM and related concepts presented in this paper can be quickly demonstrated.
Before giving a detailed description of the results and their implications, we will use an example to explain the different situations that can be obtained (see Figure 3):

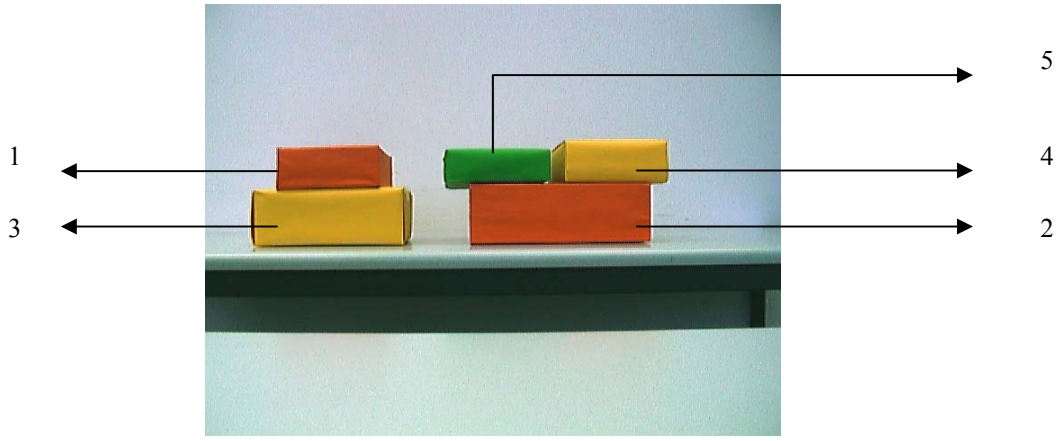


Figure 3: Test image composed of 5 different objects (annotated from 1 to 5). The different boxes can be associated to real world objects. For instance, the green box could correspond to a can of coke, the yellow box to a bottle of orange juice, etc.

The image depicted in Figure 3 contains five objects annotated from 1 to 5. As illustrated in Table I, all 5 objects were recognized as expected. One object was identified by its color correctly, but it was confused with another object of the same color (i.e. the color association was perfect but the correct object association was inaccurate in this case). Thus the recognition rate was 90%. There were no false positives (i.e. existence of a object when it isn't actually the case) observed, no merging problems between objects and no objects were completely missed.

TABLE I. OBJECTS CLASSIFICATION FOR THE EXAMPLE SHOWN IN FIGURE 3

Results	ID Objects				
Expected recognition output	1	2	3	4	5
Observed recognition output	2	2	3	4	5

Many statistics have been collected for the test set of 15 images. Table II shows how many cases of false positives, merged objects and missed objects were encountered. A good result that must be underlined is that no false positives were detected. Another outcome that was checked for was the case of merged objects. From the entire experiment, only three occurrences of this were found. This was due to an inaccuracy in the color thresholds. With regards to completely missing objects, our system did well in that only one object in one case was overlooked. This was caused by an illumination change, too pronounced for our system to handle. Table II thus draws out a lot of positive aspects about our simple object recognition mechanism.

TABLE II. MEASURE ON FALSE POSITIVE, MERGED OBJECTS AND MISSED OBJECTS CASES FOR THE SET OF 15 IMAGES.

Cases of False Positives	Cases of Merged objects	Cases of overlooking of objects
None	3	1

The table below (Table III) shows the results obtained from the object recognition module for 15 different images. The image annotation (i.e. the id corresponding to each object type) is exactly the same as shown in Figure 3. For the first few images, two objects of type 1 were used. The sequences (both observed and expected ones) are formed by listing the objects in the order of its color – first, the orange ones, followed by the yellow and finally the green ones. The table points out several interesting facts. Firstly, most of the objects were recognized in each image and most of the identified objects were in turn identified correctly. A correct identification corresponds to correct object association. There are however cases wherein object color is identified correctly but it is confused with other objects of the same color. In Table III, all cases wherein the number of observed objects is less than the number of expected objects, one of the three errors detailed in Table II occurred.

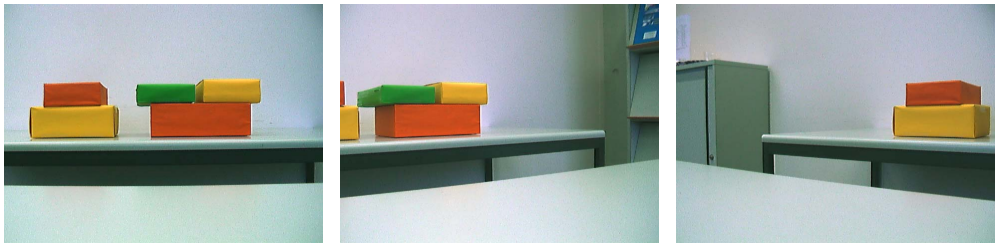
TABLE III. STATISTICS PERFORMED ON THE SET OF 15 IMAGES.

Image ID	Output Observed	Expected Output	Number of Objects expected	Number of objects observed	Number of correctly identified objects	Number of objects : color identified correctly but object confused with another of same color
0	122335	112345	6	6	4	2
1	135	245	3	3	1	2
2	1435	12345	5	4	1	3
3	12345	112345	6	5	4	0
4	1235	11235	5	4	3	0
5	222345	112345	6	6	4	2
6	1345	112345	6	4	3	0
7	212335	211435	6	6	4	2
8	14	24	2	2	1	1
9	24	24	2	2	2	0
10	22345	12345	5	5	4	1
11	1435	2345	4	4	1	3
12	23	13	2	2	1	1
13	235	235	3	3	3	0
14	2235	1235	4	4	3	1

The positive feature about the method is that in most cases, the number of objects correctly identified exceeds the number of objects which were identified correctly in terms of color.

4.2. Belief Update

The belief update method has also been tested. Its performance however depends heavily on the object recognition module. Figure 4 shows three images (i.e. image id = 10, 11, 12) that represent three different scenes in the environment. These images have used to demonstrate the updating concept. In order to correctly demonstrate the belief update procedure, the object association errors in the output of the recognition module have been corrected (i.e. as shown in Table III, in all three cases, there are objects whose color has been correctly identified, but they have been confused with other objects of the same color. These errors have been corrected to the expected outputs in order to facilitate the demonstration of the concept). Next, we will show the overall mechanism used for the belief update process.



Scene 1 Scene 2 Scene 3

Figure 4: Three scenes from the environment used to show the updating process. The ids of the objects are the same as the ones depicted in Figure 3.

In the beginning, scene 1 is processed by the object recognition module. All objects and their interrelationships are identified. This information is then used to construct the OGM for scene 1. Identified objects, their properties, interrelationships between them, and the beliefs in each of these were stored in the OGM. Figure 5 shows the partial OGM formed with respect to object 2.

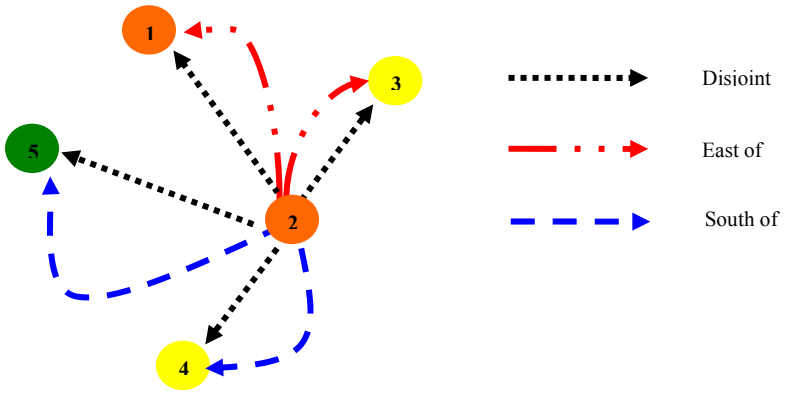


Figure 5: Partial OGM for the object 2 in scene 1

The other possible inter-relationships between objects are not shown in order to preserve clarity in the figure. After processing scene 2, its OGM is formed in a similar fashion. The two OGM's are compared to determine if they represent the same scene. The resulting comparison factor was smaller than the preset threshold and thus the OGM of scene 2 is now used to update the OGM of scene 1 in order to form a single, consistent and up-to-date representation. All objects and relationships that are also observed in scene 2 (in addition to being observed in scene 1) have their beliefs in the OGM representation of scene 1 updated with the current belief. Conversely, un-observed objects and relationships have their beliefs reduced in the OGM of scene 1. For e.g., the belief in object 2 and its relationship with object 4 are both updated with their current beliefs in scene 2. On the other hand, the belief in object 2's relationship with object 1 is reduced as object 1 is not anymore observed in scene 2. Another important OGM update operation is the removal of objects and relationships when their beliefs fall below a certain threshold. This is intuitive in that the belief is a measure of the existence of an entity. In the example considered here, object 3's relationships with object 4 are removed when scene 2 updates the OGM for scene 1, as the belief in their existence becomes negligibly small. Scene 3 is processed in a similar way. It is compared with the previously modeled scene (scene 1 updated with scene 2) and deemed to match it. Thus its OGM is used to update the previously accumulated representation. As object 2 is not observed in scene 3, the belief in object 2 and that of all of its relationships is reduced after scene 3 updates the accumulated representation. Thus the belief update procedure proceeds exactly as per expectations.

This example clearly points out that the belief update mechanism is very easy to apply, very intuitive, permits modeling the dynamics of the environment and yet produces a single consistent and up-to-date representation of the environment as perceived by the robot.

5. CONCLUSIONS AND FUTURE WORK

This paper has presented a method for topological space representation using a combination of probabilistic belief with object graph models (OGM). This kind of representation permits efficient modeling of the information that a robot would perceive from its environment. The larger goal of the work presented here is to address the integration of navigational and interactional capabilities of robots through an efficient unified multi-level representation of space. This work puts forward a general, graph-based, probabilistic representation of the environment and is meant to contribute towards the larger goal. This directly relates to the research area of Human Perception and Cognition. The concept was validated through experiments that showed very encouraging results. Some future works will concentrate on developing a method for color constancy, based on a PID controller that will stabilize the mean luminance, and on the improvement of the object recognition and modeling mechanism. Use of invariant features like the SIFT and color invariant features seems to be a promising option towards robust object recognition. 3D object modeling will also be required in order to have effective models for interaction. In addition, efforts will also be directed towards extending this work into a multi-level cognitive probabilistic representation with multi-modal data suited for both navigation and interaction. Incorporating more geometric, semantic and functional information about objects into the OGM's will be

required for higher level inference, better navigational and interactional capabilities. While adding multimodal data, sound uncertainty models for each of the inputs will also be required in order to effectively handle the uncertainty in the environment. In summary, to have real cognitive agents using this very promising concept, the implementation must be scaled up in all facets, it has to be subjected to an exhaustive set of test scenarios and complexity issues must be dealt with to ensure scalability.

ACKNOWLEDGMENTS

The authors would like to thank the COGNIRON FP6-IST-002020 and BIBA IST-2001-32115 EU projects, which are funding this research. We would also like to express our gratitude to Bjoern Jensen for offering several valuable and timely suggestions during the implementation phase of this work.

REFERENCES

1. Brezetz, S. B., Chatila R. Devy M., (1994), Natural scene understanding for mobile robot navigation, In the Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, USA
2. Egenhofer, M.,J., Reasoning about binary topological relations. In Gunther, O. and Schek, H.J. (eds.), Advances in Spatial Databases, SSD'91 Proceedings, Springer Verlag 143-160, 1991.
3. Freeman H., (1961), On the encoding of arbitrary geometric configuration, IRE Transactions on Electronic Computers, EC-10(2):260 – 268
4. Grigni, M., Papadias, D., Papadimitriou, C. Topological Inference. Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI), Montreal, Canada, AAAI Press, 1995.
5. Hollink L., Nguyen G., Schreiber G., Wielemaker J., Wielinga B. and Worring M.., Adding Spatial Semantics to Image Annotations. 4th International Workshop on Knowledge Markup and Semantic Annotation at ISWC'04
6. Hu M.K.,(1962), Visual Pattern Recongnition by moment invariants, IRE Transactions on Information Theory, 8(2):179-187
7. Kuipers, B. J. (1978), Modeling Spatial Knowledge, Cognitive Science, 2: 129-153, 1978.
8. Kuipers, B. J. (1983), The Cognitive Map: Could it have been any other way?, In Spatial Orientation: Theory, Research and Application. Picks H.L. and Acredolo L.P. (eds.), New York. Plenum Press.
9. Kuipers, B. J. (1996), A Hierarchy of qualitative representations for space, In Proceedings of the 10th International Workshop on Qualitative Reasoning (QR-96), Fallen Leaf Lake, California, USA.
10. Martinelli, A, Tapus A, Arras, K.O., and Siegwart. R. (2003), Multi-resolution SLAM for Real World Navigation, In Proceedings of the 11th International Symposium on Research Robotics, Siena, Italy.
11. O'Keefe, J., and Dostrovsky, J., The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. Brain Res. 34, 171-175, 1971.
12. O'Keefe, J., and Nadel, L., The hippocampus as a cognitive map, Clarendon, Oxford, 1978.
13. Papadias, D., Theodoridis, Y. Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures. International Journal of Geographic Information Science, vol l.11(2), pp. 111-138, 1997
14. Rohanimanesh K., Theocharous G. and Mahadevan S. (2000), Hierarchical Map learning for Robot Navigation, In AIPS Workshop on Decision-Theoretic Planning, Breckenridge, Colorado.
15. Shih H.C. and Huang C.L. (2003), A Semantic Network modeling for understanding baseball video, In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Hong Kong, China.
16. Sonka M., Hlavac V. & Boyle R., (1998), Image Processing, Analysis and Machine Vision, Brooks/Cole Publishing Company
17. Tolman, E. C. (1948), Cognitive maps in rats and men, Psychological Review, 55:189-208.
18. Tomatis, N., I. Nourbakhsh, and R. Siegwart (2003). Hybrid simultaneous localization and map building: a natural integrationof topological and metric. Robotics and Autonomous Systems, 44:3-14.
19. Voicu H. (2003), Building and Using a Hierarchical Representation of Space, In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Portland, Oregon, USA.

Acquiring a Shared Environment Representation

E.A. Topp*, H. Hüttenrauch†, H.I. Christensen* and K. Severinson-Eklundh†
* Centre for Autonomous Systems † Interaction and Presentation Laboratory
Royal Institute of Technology
10044 Stockholm, Sweden
{topp,hehu,hic,kse}@nada.kth.se

ABSTRACT
Interacting with a domestic service robot implies the existence of a joint environment model for user and robot. To enable robot navigation within such a setting requires further to embed a user's mental environment model in the corresponding robotic model. Robots typically use metric models for navigation, therefore those metric models need to be integrated with models provided by users. This paper presents a pilot study that investigates, how humans present a familiar environment to a mobile robot. Results from this pilot study are used to evaluate a proposed generic environment model for a service robot.

Keywords
User study; Environment representation; Human Augmented Mapping; Personalisation

1. INTRODUCTION
Service robots are – often mobile – platforms that provide assistance to humans. Thus, a basic competence for such a mobile robotic system is the ability to move from one location to another to provide its services, which requires navigation and localisation functionalities. Also, the robot has to share the environment with its potential users, which means that it has to move around and reason about its whereabouts in a way that is comprehensible. Mobile robots can navigate on the basis of metric, often feature based maps, and they can build those maps autonomously while exploring an environment for the first time. Methods in robotics research are dealing with this issue of Simultaneous Localization and Mapping (SLAM) [3; 4; 6, among others]. Humans have a topological, (partially) hierarchical, view on their environment [12, as an example]. To enable a service robot to perform tasks for users in arbitrary environments (well known to the user, initially unknown to the robot), a spatial representation that is understandable for both, the robot and the user, is needed. Assuming an indoor environment such as a home or office building, we mean by a commonly understandable map representation, that the robot's notion of the environment appears to be the same as the one the user might refer to. In other words, we need to build a "shared mental model". Such a model is likely to depend on a very personal view a potential user has on the environment. Thus, a service robot, provided with some general world knowledge, can be – and ought to be – personalised. We assume a scenario of a "guided tour" to be an appropriate way to "teach" the robot its environment. The user can guide the robot around and name important regions and specific locations. At the same time, the robot can build a (metric) map of the environment. This map is augmented by the user's information which allows to integrate the robot's metric, feature-based map with the topological map representation of the user. Figure 1 gives an idea on how a scene of such a guided tour could look like.

Figure 1: Illustration of a user showing the kitchen to her robot

An open issue is the question, what strategies to present an environment would be used by different users, and how the given information can be incorporated into an environment model, to actually satisfy the requirements for a shared representation. In earlier work we already introduced the concept of *Human Augmented Mapping* (HAM,[16]), which allows us to subsume different aspects of Human-Robot Interaction (HRI) and robotic mapping. In a previous study [7] aspects of interaction as well as posture and positioning of subjects in relation to a robot were studied. In this case the scenario was also a "guiding the robot around" scenario, but the environment was limited to one room and the robot used in the study was controlled remotely.

With the present paper we describe a user study, in which

Human Robot Interaction '06 Salt Lake City, Utah, USA

subjects guide around an autonomous mobile robot in a complete floor of an office building that they are familiar with. The study investigates how different users present a well known environment to a robot. We suggest a generic robotic environment model and demonstrate with results from an initial pilot study, how this model can be personalised to different individual representations of a given environment.

1.1 Outline of the paper

The rest of this paper is organised as follows. We give an overview of related work and refer to hierarchical environment representations motivated from results in Cognitive Science and Psychology in section 2 to propose a general robotic environment representation in section 3. Section 4 explains the design of our study, in section 5 we present the results from initial experiments, and in section 6 we draw conclusions on this pilot study and its results.

2. REPRESENTATION OF SPACE

In "The intelligent use of space" [9] Kirsh stated, that in order to understand complex (human) models of an environment, we have to observe the interaction of the (human) agent with this environment. Based on those observations, corresponding *robotic* models can be obtained. Transferring this to the interaction of two agents in and about a certain environment, observations from human-human interaction could be the base for a general robotic environment model. Such a model can be used to incorporate information from the interaction with a user, to personalise the robotic system. Personalisation along the taxonomy of Blom [2] means in this context to *accommodate work goals* (to "customise" the robotic system for certain tasks) and to *accommodate individual differences* (of different users in the explicitly stated representation). We propose to observe a human and a robot interacting in an environment (instead of two interacting humans), to learn, what robotic model can be used to build a "shared mental model" that both the user and the robot can refer to later.

In a study that uses a miniature robot on a table top street "map", Kyriakou *et al.* investigate, how computer vision can be used to follow verbal guiding directions [11], by having subjects guide the robot with commands like "follow this road to the station, then turn left". This is another form of "guiding a robot" without actually being part of a collaboratively operating duo in the same work space. One condition for such a setup is the availability of a map that contains all items a potential user considers important at the respective location. Since this is what we wanted to learn about (what do users present in a given environment and how) we do not consider such a setup an option for our study.

Kuipers *et al.* presented a mapping approach that represents the environment as a combination of global topological and local metric maps [10]. The main aspect of this work however is the handling of large scale maps, that can be achieved by representing the environment as local metric maps that are linked in a global, topological (and as such hierarchical) representation. Also in other approaches the segmentation of metric maps and/or organisation of them into hierarchies has been studied as part of SLAM, but primarily as a way to limit computational complexity [3; 14; 15, among others].

Approaches to interactive robotic mapping have been reported by Diosi *et al.* [4] as well as by Althaus and Christensen [1]. Diosi *et al.* obtain a purely metric spatial representation of an office environment by guiding a robot around and defining labelled regions. Althaus and Christensen model the environment rather as a topological graph, but do not consider different levels of granularity in their representation. We believe that not only rooms (or regions) are needed, but also a lower level of complexity has to be integrated in a topological model. This allows to integrate places into the specified regions.

2.1 Motivation for an environment model

A number of different theories on how spatial relations are acquired and represented have been proposed throughout the years. According to McNamara [12] those theories can be grouped different the dimensions of a) format (analog vs. propositional), b) functionality (spatial configuration vs. semantic or logical knowledge), c) structure (flat vs. strongly hierarchical), and d) contents (encoded information vs. procedural knowledge to compute information).

McNamara used this categorisation to design a psychological study on spatial representations that concentrated only on the two latter characteristics (structure and contents). Subjects were given recall and distance estimation tasks on items that were spread out in physically separated regions on a "map". The results indicated, that distance between two items matters as well as co-coexistence in one region. In other words, if two items were close to each other, but in different regions, it was still possible for the subjects to recall and estimate their spatial relation. If the distance was large, this recall and estimation worked better within the same region. Thus McNamara came to the conclusion, that a *partially hierarchical model* supported his findings most appropriately.

Following these findings, we assume, that users would not necessarily follow a hierarchical order when explaining the environment to the robot, e.g. explain a certain place first and then give information about the room or present certain places only, that are located in different rooms. Transferring this to our guided tour implies, that the assumed robotic environment model has to be able to handle spatial information given in arbitrary order. Thus we propose a hierarchical structure, that incorporates the required flexibility with generic entries on each level, in which places can be represented. We express this assumption as well in a number of working hypotheses for the pilot study in section 4.3.

To incorporate other dimensions, particularly the *functionality*, the hierarchy needs to be extended. Galindo *et al.* [5] propose *Multi-Hierarchical Representations* to incorporate semantic information into their environment model used for mobile robotics. Two hierarchies, one conceptual, the other spatial, are linked with anchoring to enable reasoning. Their spatial hierarchy is build from local map representations obtained from sensory data, that are interpreted as open spaces (rooms, corridors) connected in a topological structure. The conceptual hierarchy incorporates concepts such as workspace, room, object and instances of those categories. A semantic model is given a priori, that links objects concep-

tually to rooms. For example it is assumed that an object "bathtub" is to be found in a room called "bathroom". By observing objects, the conceptual hierarchy is used to assign a specific concept ("bathroom") to a local map representation in the spatial hierarchy. As stated above we assume a hierarchical representation of the environment, but do not incorporate the semantics so far.

Along with the functionality one issue is in fact the personalisation [2] of a particular environment representation. From intuition one would expect, that individuals have different preferences and ideas how to interpret and use their surroundings. We consider the fact that different users might give different information to the very same robot as an issue of future work. Our environment model though is flexible enough to model those individual differences within the same framework.

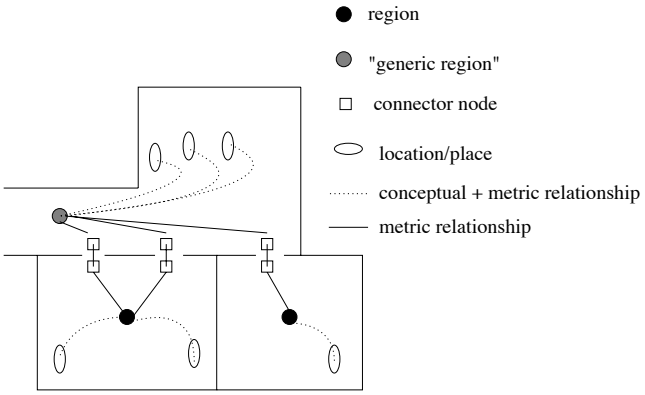


Figure 2: Our graph structure visualised in 2D

3. HUMAN AUGMENTED MAPPING

With our concept of *Human Augmented Mapping* we can establish the link between a robotic map that enables the robot to navigate and the environment representation of a user (also referred to as "cognitive map"). We use a graph based model of the environment, described in section 3.1 to incorporate the information that is given interactively. Our assumption is that a "guided tour" is an appropriate way to give the user the possibility to personalise the robot's general environment representation. An "off-line" personalisation that could be achieved by pointing out items, places and regions on a metric map representation that was *autonomously* created by the robot does not seem useful, since the user would have to remember *exact* spatial relationships between items. When the robot and its user share the same workspace in an interactive setting, it is presumably easier to determine for the user, if the robot "understood" a piece of information correctly. From the robotic point of view we see the advantage of an interactively controlled mapping process in the disambiguation possibilities that arise from the interaction. We further do not assume a full initial environment model, that allows the robotic system to instantiate content entities by autonomous exploration, but consider a more general, structural model that can be filled with personalised information and that can be revised if necessary.

3.1 A hierarchical graph structure

We model the environment by using a hierarchy of graphs. The main concepts we incorporate so far are *locations* (or places) and *regions*, as depicted in figure 2.

We define *locations* as
Specific positions/areas that can represent the position of large objects that are considered static.
Such locations can for example be a closet, a refrigerator or a sofa.
A *region* is then
Any portion of space that is large enough to allow for different locations in it, or at least large enough to navigate in it.
Typically this would be rooms or corridors or parts of those.

A natural extension to a higher level would be *floor* or *building*, but this was not considered for this work. On a lower level smaller objects that can (hypothetically) be manipulated and change their position frequently could be integrated, such as milk bottles in the fridge or brooms in a closet.

Regions are represented by local (metric) maps that can be used for navigation. The local maps are linked metrically by pairs of internal *connector nodes* that have an absolute position with respect to the local map they are in. Since those maps are built at the same time as the graph structure is filled with the information from the user, an initial internal hypothesis needs to be introduced. To maintain the hierarchical structure but allow for partially hierarchical representations as well, we assume a "generic region" in which we start the mapping process. With the "generic region" we can guarantee, that all mapped areas are represented as a region on the respective level of the hierarchy. As soon as a region is assigned a name by the user, it is stored together with the corresponding local representation, that might already contain information on specific locations. When a specified region is left, and the adjacent area was not explored before, this "new" region becomes the "generic region". Note that the "generic region" can consist of several, topologically delimited regions (in the sense of an autonomous mapping approach). Only specified regions are entities in the hierarchy that form a new branch from the respective level downward. This makes it possible to define a specific location in a region, that is not (yet) relevant to name (e.g. a corridor).

4. THE PILOT STUDY

We conducted a pilot study to test our proposed robotic environment model against the information on a specific environment given explicitly by a human user to a mobile robot. Additionally the pilot study serves as a proof of concept for a more comprehensive user study currently prepared. The pilot study comprised experiments with five subjects of about 45 minutes duration. Within this time period the subjects spend about 20 minutes interacting with the robot, the rest of the time was used for instructions before and short interviews after the sessions. All of them received a cinema ticket voucher as compensation for their participation.

4.1 Scenario

The scenario of the pilot study was a "guided tour" through a portion of an office building. Figure 3 shows the floor plan

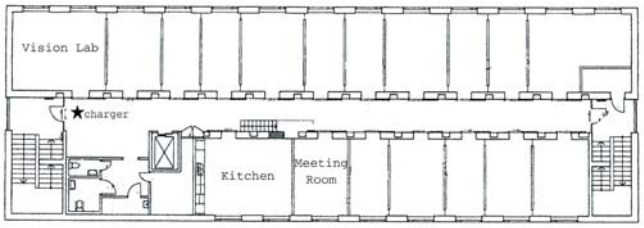


Figure 3: The floor plan of our office environment on which the experiments took place. The star marks the starting point, where subjects encountered the robot

with offices (not marked), the kitchen, the meeting room and the computer vision laboratory of our office building. Subjects were instructed to show the robot around in the environment so that it later could perform service tasks and in order to do this needed to have "seen" the respective locations (a more detailed description of the instructions and the technical realisation is given in sections 4.2.2 and 4.2.3).

4.2 Method

In the following section we explain our selection of subjects, the instructions given to them, and the methods used for data collection.

4.2.1 Subjects

As important precondition to our pilot study we assumed subjects to know the environment they would guide the robot around in. This precondition is important and based on the idea that potential users will "add" service robots to their (to them already well known) homes and offices. Subjects were therefore recruited from the laboratory environment the experiments took place in. We make this a requirement also for our future study as it will ensure that subjects are not primed or biased by information that would have to be given to participants unfamiliar with the given environment and spatial settings. To require familiarity with the robot's operation area is a design choice that differs from other human-robot interaction studies, where subjects often are invited into an unfamiliar or "simulated" environment. The choice comes at a price however: in our own office environment some subjects of the pilot study were expected to be familiar with the internals of robotic systems. As a consequence we plan on (also) using a different environment for our future user study, to make sure that the familiarity with our robotic system is counterbalanced by subjects without experience in robotics research.

To assure at least some variety in familiarity with robotic systems we selected our five subjects actively among the members of the Computer Vision and Active Perception Laboratory[1] that hosts a part of the Centre for Autonomous Systems[2] on our campus. The group of pilot subjects included one secretary (familiar with robots from films, presentations and frequent encounters in the office environment, but not familiar with the internals), three computer vision researchers, one of them somewhat familiar with the internals of robotic systems, and one robotics researcher from the field of robotic mapping. All of them had been working in this particular office environment for about two years.

[1] http://www.nada.kth.se/cvap
[2] http://www.nada.kth.se/cas

4.2.2 Instructions

Our subjects were given an instruction sheet that explained the task and the functionalities and abilities of the robot. The task was to use a number of commands (`follow me, go to <target>, stop, turn left, turn right`) and explanations (`this is <item>`) to make the robot follow and to point out everything that the subject considered important for the robot to know on the floor the experiments took place on. The time frame given to the subjects for the completion of their task was about 20 minutes (15 minutes for the guided tour and five minutes to test the robots "memory"). In the instruction none of the words *region, location, position* or *place* was named. We referred to *"everything, that you think the robot needs to know", "whatever you pointed out before", etc.*, so that subjects were completely free to decide, what they would present to the system and how they would name it. Neither did we give any example (e.g. "You can name for example the coffee maker"), so that we would not include any items that a particular subject would not have considered important in the first place. Nevertheless subjects were informed, that we were not interested in small objects, since the robot had no object recognition abilities, it just would need to know "where" to go. The instruction sheet included a drawing that showed, how the field of view of the robot looked like, and that it used a laser range finder to detect the subject for following and in order to "look around". This information was important, since the laser range finder only offers a forward field of view of 180°, with a range of 8 meters (for the detection of users we reduced it further to 3 meters), and the subjects could thus understand, how the robot perceived its environment. Particularly they were instructed to move a few steps in front of the robot so that it would detect and classify them as user. To make the robot start to move when following they should gain a distance to it of at least one meter, to give it the space to actually move.

The subjects were also informed that the robot was moving autonomously, when it was following a subject according to the task scenario, but that commands were interpreted by an experiment leader and fed manually into the system. Since we did not incorporate any object recognition, we stated that a service task (`go to <target>`) would be successfully completed, when the robot could find its way to the location where the task would have been performed. Also for the actual presentation of an item, the robot was assumed to "see", when it was "facing" the item. The instruction sheet was very open about the robot's abilities: we clearly stated which of the functionalities of the robot were in fact simulated or remotely controlled (see 4.2.3 for details) by an experiment leader that followed the (subject and robot) pair. We also explained, what the subjects should not try to do, as for example to send the robot around alone to explore the environment on its own, use the elevator or try to send it somewhere unknown. Subjects were offered to ask for help before and during the actual experiment, and knew that they could abort the experiment at any time.

4.2.3 Technical realisation

The study was performed with a commercially available Performance PeopleBot by ActivMedia[3]. In a previous study this robot was used in a Wizard-of-Oz-setting [7], where the robot's functionalities were remotely controlled or simulated by two experiment leaders. For the technical realisation of our pilot study scenario we used a laser range data based tracking and following system [16], which has been extended to incorporate a metric laser range data feature based SLAM method [6] and an input option to label regions or locations with name tags. Basic platform control and access to the sensors and text-to-speech system (Festival[4]) are provided by the Player/Stage[5] software library.

The system represents labelled regions and locations in a simple graph structure that distinguishes between specifically labelled positions ("defined place") and internal navigation nodes. The internal nodes are used to build a navigation graph on which the system can perform a graph search to plan a path[6] to a previously named position. Note that this system does not implement the hierarchical model we proposed above, but enables a user to act and interact with the robot according to our scenario to test the validity of our proposed model.

The verbal interaction of the user with the robot was still controlled by the experiment leader, i.e. utterances from the subject were interpreted by the experiment leader and labels of locations or regions and commands were fed into the system via a graphical user interface (GUI) running on a laptop. This allowed us to avoid problems due to miscommunication (as studied by Green *et al.* [8]), which otherwise could have interfered with the actual task. For verbal feedback though we used the text-to-speech system with precoded utterances, so that the robot could refer to its own state and the task given to it (e.g. "I will follow you", "Stopped following", "I think I have lost you", "Stored <item>").

As the experiment took place on an entire floor of the building, one experiment leader (the robot's supervisor) had to follow the subjects to observe the experiment including all utterances and in general to assure the subject's safety at any time. The implemented system allowed switching from autonomous following based on the mentioned tracking approach to full remote control immediately by invoking a soft joystick implementation. Thus tracking failures and other inconvenient situations could be solved by the experiment leader as in a Wizard-of-Oz setting without having consequences for the mapping process and the labelling.

We provided the robot with two different behavioural strategies for the labelling of either a location or a region. If a location (including a "link" to a region, e.g. a doorway) was presented, the robot did not move and stated immediately, that it stored the given information. If on the other hand a region was presented, the robot stated, that it needed to have a look around and performed a 360° turn before confirming the information. The decision, which behaviour to choose, was made by the experiment leader according to our generic environment model and the respective definitions of regions and locations.

[3]http://www.activmedia.com
[4]http://www.cstr.ed.ac.uk/projects/festival/
[5]http://playerstage.sourceforge.net
[6]implementation part of the CURE library (©2005 Patric Jensfelt and John Folkesson, Centre for Autonomous Systems, Royal Institute of Technology, Stockholm, Sweden)

4.2.4 Observation methods and data collection

By storing the data provided by the sensory systems we used for the technical realisation we could get a full "real time" (graphical) representation of each of the experiments. Additionally we recorded the experiments with two digital video cameras each. One video was recorded from the point of view of the robot, by mounting the video camera on its upper platform. The other camera recorded an external point of view by accompanying the user and the robot. After their experiments our pilot subjects were asked to answer a number of questions on the experiment in a short interview. This interview was roughly scripted with a list of questions on the motivation of the subject for naming or not naming certain locations or regions and for the way to handle the tour scenario. We were particularly interested in whether the subjects had perceived the behaviour of the robot differing depending on what was pointed out (a location of a region) and what they thought about it.

4.3 Hypotheses

We wanted to study, how different individuals present a known environment to a mobile robot and relate the resulting information to an environment model we consider appropriate in the context of Human Augmented Mapping. We assumed that humans do not necessarily follow a hierarchical structure, when they present a known environment to a robot (see section 2). Thus, we started out with a number of working hypotheses about the way subjects would present the regions and locations they considered relevant, as well as about the entities that would be named: "users do not name all regions in the environment" (`WH1`), "users point out locations in regions they did not name before" (`WH2`), and "users point out regions without entering them" (`WH3`). We use these hypotheses to show, in how far the outcome of the pilot study can be related to our environment model. We did not formulate a specific hypothesis for the dependency "familiarity with robotic systems vs. way of explaining the environment to a robot" to explore this issue. Nevertheless we expected robotic researchers familiar particularly with map representations to be more explicit than subjects not familiar with robotic environment representations. Further we assumed, along the argumentation of Sidner *et al.* [13], that the difference in the robot's behaviour would allow the subjects to "understand" the robot's internal processes, when storing either a region or a location.

5. RESULTS FROM THE EXPERIMENTS

In this section we present the results from our pilot study. We are aware that the data set is small and consequently not entirely representative. However, it is possible to analyse the outcome of the experiments in terms of *occurrence* of different phenomena. Additionally, our observations and the subjective answers we obtained in the short interviews allow us to investigate how subjects reasoned about their strategy to show regions and locations and to improve the system for further studies. In general we can state, that the pilot study verified the validity of our approach to get information on different ways to handle an interactive process to

build a map representation. Furthermore we believe that the soundness of our environment model can be demonstrated by its ability to handle the different situations we observed. In table 1 we summarise the quantifiable results to give an overview over our observations and statements from the interview.

5.1 Observations

All subjects but one used the full time frame to present the environment to the robot. The "tour" started for each experiment at one end of the corridor (see Figure 3), where the robot awaited its user. An initial location (the "charger") was generated automatically directly after the system was initialised to enable the robot to go back to this starting point. As a consequence we do not count this location as relevant to our results. All subjects took the robot into the kitchen, probably because this is a central room in our office environment, both from a topological, a functional, and a social point of view. However, the observed diversity in strategies to introduce the kitchen to the robot was quite large, ranging from the pure introduction of *the kitchen* over some combination of *specific locations in the kitchen and the kitchen itself* to *specific locations only*. Already from our small sample of data we can thus conclude that the variety of explicitly stated information that a robotic system in an interactive mapping process would have to cope with is large and needs to be handled by the robot's environment representation. More specifically, these differences in naming observed for the kitchen and its locations correspond to our expectations expressed in hypotheses `WH1` and `WH2`.

We also noted that none of the subjects named the corridor or hallway itself as a region, but all of them pointed out specific locations in it, which gives us further evidence for our hypotheses. One frequently presented location in the corridor was the "elevator" (or "lift") (named by four of the five subjects), which was however only shown by positioning the robot in front of it and pointing to the *doors*. Also rooms were indicated only by pointing to the respective door, confirming our expectation expressed in hypothesis `WH3`.

Most of the subjects stated in the interview that they had pointed out those locations or rooms, that they personally considered important, and left out others on purpose. In other cases the time constraints kept the subjects from presenting more to the robot. We see this as a sign for a strategy to personalise the robot's environment representation to personal needs and preferences.

We asked all subjects that had presented a mixture of rooms and locations (four out of five), if they had perceived the difference in reaction of the robot (turning by 360° for a region vs. not turning for a location). Three out of those four answered, that they had observed the difference in behaviour. All three stated that this behaviour seemed *appropriate* and/or made the robot *look smart*, since it obviously wanted "to understand its surroundings". One subject did not notice the difference in behaviour, possibly because only two rooms were presented, and the subject stated to have been busy figuring out, "why the robot sometimes needed a long time to understand me, and sometimes not". Note that this was stated despite the fact, that written information had been given to all subjects, that all dialogue would be simulated by the experiment leader.

Table 1: Quantifiable results from the pilot study

Observation Subject	VR	VR	VR	SE	RR
Interaction time	22 min	19 min	11 min	25 min	24 min
# regions	4	2	–	2	2
# locations^{I}	4	4	5	4^{II}	8^{III}
# regions w o loc.	3	2	–	1	1
# loc. w o region	3	4	5	2	3^{IV}
# regions w o entering	1	2	1	1	–
Behaviour noticed	Yes	Yes	–	No	Yes
– appropriate	Yes	Yes	–	–	Yes
– appears smart	Yes	No	–	–	Yes

VR: Vision researcher, SE: Secretary,
RR: Robotics researcher

I: including regions that were only pointed to
II: including one small object (salt)
III: including one person and two doorways to respective rooms
IV: excluding doorways

Despite some technical problems (see section 5.4 for details) and the above mentioned timing problem all subjects expressed their satisfaction with the flow of interaction and communication as well as the robot's performance.

5.2 Particular situations

Even with the limited number of subjects we were able to observe some interesting strategies for the presentation of the environment. We relate the observations to statements from the short interviews where possible. We attempted to order them with increasing relevance to the environment model.

Pointing out persons In two cases subjects tried to point out a person. In one case the person was sitting at her desk and the robot was made to store the respective location by the experiment leader. In the other case the subject reacted spontaneously to someone walking out of the elevator right in front of the robot. Here the robot was not made to store the information from the introduction. Nevertheless these situations show, that the system would have had to handle introductions of persons as well, since the introductory phrase "this is $< name >$" was exactly the same as for the kitchen[7].

Possessive pronouns and relations One of the subjects presented "my office" to the robot. In such a case a dialogue system would actually have to analyse "my" and relate it to the subject's name, but this was beyond the scope of our pilot study. In the experiment, the robot was thus constantly referring to "my office".

Extreme personal point of view In one experiment session we observed that two rooms were pointed out, but no locations in them. In the corridor, none of the service points (pigeon holes, printer, etc.) was named, but the two exits

[7] In this particular experiment the subject left out all articles when presenting items

to either side of the building and the elevator, as well as the experiment leader's office (only the door was pointed to). When asked why no other locations in e.g. the kitchen were named, the subject stated that the exits were considerably important, as well as the kitchen as a room, in case that guests had to be met and/or served. The coffee machine and the refrigerator were not important since the subject does not drink coffee at all nor uses the refrigerator. The observation holds both evidence for our hypothesis `H2` and an extreme personal point of view on the environment.

Explaining no rooms at all One of our subjects concentrated only on locations (e.g. pigeon holes, coffee machine, refrigerator) and did not name any room (or other region). On the question, why not for example the kitchen as a whole was named, the answer was, that the robot should rather know about the whereabouts of the places, where it should *do* something. Just sending it to *the kitchen* would by no means help to get a coffee, the subject stated. We see this as a strong evidence for hypothesis `H2`.

Explaining doorways We expected our subject with robotic research and mapping experience to be more precise and explicit than other subjects. This expectation could be confirmed by the fact, that the doors to showed rooms (two in this case) were pointed out explicitly *when the robot was standing exactly in the door opening*. We could also observe, that both named rooms were actually entered. Since only two rooms were presented during this experiment, we can of course not generalise, but we consider at least our expectations for the robotics researcher's strategy confirmed.

5.3 Relation to our environment model

Our observations show, that even with a small, rather homogeneous group of subjects different ways to show and explain the environment are to be expected and dealt with, depending on the individual view and use of particular items and rooms. We see these differences as a proof of concept for our proposed environment model (as introduced in section 3.1) which we consider usable for a robotic map representation.

A general assumption is that a given robotic system has the ability to perceive regions that are delimited from other regions autonomously. This could for example be achieved by door detection or a method like the watershed algorithm [4, as an example of use]. We also assume, that we have a general knowledge model that distinguishes between regions and locations and a dialogue model that uses this knowledge base. From the experiments we got some evidence already on the strategy of the users to point out a region by only showing the respective door. In all observed cases, subjects positioned the robot with the help of "turn commands" so that it was facing the particular "link" (doorway or elevator doors), before naming the region. If these subjects on the other hand presented the region they were currently in they just stated that this was "the $< name >$" without positioning the robot with "turn commands". The detection of such differences in the user's behaviour could give a signal on the actual intention of the user. We hope to find further evidence for such a differing behaviour in our future study.

Departing from our observations we can postulate some key situations, that need to be handled by our robotic environment model and suggest possible solutions.

Presenting persons Given an appropriate dialogue model, it would be possible to ask, if actually the region/room the person is in should be named accordingly (e.g. "Elin's office", in case "Elin" was introduced to the robot).

Locations in an unnamed region If a location is named before the region it is in, or the region is not named at all, this location would end up in the branch of the "generic region" in our hierarchy. If later the information about the region is given, the region needs to be delimited and separated from the generic region. All locations within the observed delimiters (e.g. walls, doorways) are now associated to this new branch in the hierarchy.

Links to regions/rooms With the "connector nodes" of our representation links to rooms (pointed out doorways) can be handled. In the current region (which might be the "generic region") a connector node with a virtual directed edge to the named region is created. Thus the system knows, that it can find the way *to* a certain region, without knowing anything about its appearance.

Pointing out doorways explicitly The environment model could cope with explicitly pointed out doorways by generating a location with the respective name. There are several possibilities to represent it in the hierarchy though. One option is to decide which region it belongs to, based on the name of the respective region (e.g. as observed "this is the door to the kitchen"). The second option is to keep the location in both regions, with a relative position to the respective local map that relates to the same absolute position (if possible). A third option would be to generate an entry of the generic region, that would allow to state that the robot is "in between two regions". However, since we could observe the respective strategy only with the robotics researcher, we assume this to be rarely observable with a differently structured sample.

Summarising we believe that our model holds at least for the variety of strategies to present a known environment to a robot observed in our pilot study.

5.4 Technical issues

During the pilot experiments we observed several issues of the technical realisation that had consequences for the actual interaction between subjects and the robot.

Despite the instruction to give the robot space when it was about to follow, subjects waited standing still for the robot to move. The robot's verbal indication to follow ("I will follow you") was obviously not enough, to indicate that it would actually follow. From carefully studying the recorded interaction on video we concluded that the robot actually needs to indicate with a body (movement) gesture like turning toward the user that it has seen the user and is ready to follow. A similar problem occurred, when subjects made the robot face something to "look at it" and wanted to continue the tour afterwards. We plan to make the robot turn back toward the user to indicate, that it is ready to continue after storing a presented item.

6. CONCLUSION AND FUTURE WORK

In this paper we presented two important aspects of our concept of Human Augmented Mapping, namely the environment representation of the robot and the interactive context that allows to build a shared mental model of an environment. We explained our approach to a robotic map representation, and showed, to what extend this representation holds in different situations within an interactive mapping process. A pilot study was conducted to investigate strategies of users to present a for them well known environment to a robot.

Despite the small number of subjects in the study we were able to observe a rather large variety of strategies to present a known environment to the robot in a "guided tour". Parts of this diversity might be due to differing knowledge in robotics or the individual interest in the robot that our subjects had. However, we can state that all the different situations or strategies, characterised in a number of hypotheses we formulated, occurred at least once. The variety in presentation strategies we observed and the self reflecting comments on them showed us, that there is a need for quite flexible representations, when one robotic system should be used and guided around by different users.

We got mostly positive feedback on the behaviour of the robot, especially on a "region observation strategy" we implemented to enable subjects to understand the internal processes of the robot to some degree. This assures us to keep such a behavioural strategy for further studies, to allow subjects to understand more of the internal procedures of the robot.

The results from the pilot study encourage us to use the proposed setup in a more comprehensive user study and to investigate the applicability of the proposed environment model in a robotic framework in more detail.

7. ACKNOWLEDGMENTS

The work described in this paper was conducted within the EU Integrated Project COGNIRON ('The Cognitive Robot Companion' - www.cogniron.org) and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

We also thank Patric Jensfelt and John Folkesson for a their technical help with the implementation of the system, especially the SLAM method used.

8. REFERENCES

[1] P. Althaus and H.I. Christensen. Automatic map acquisition for navigation in domestic environments. *in Proc. of the International Conference on Robotics and Automation,* September 2003, Taipei, Taiwan.

[2] J. Blom. Personalization: a taxonomy. *CHI'00 extended abstracts on Human factors in computing systems,* 313-314, ACM Press.

[3] M. Bosse, P. Newman, J. Leonard, S. Teller. SLAM in Large-scale Cyclic Environments using the *Atlas* Framework. *International Journal of Robotics Research,* 23(12):1113-1140, December 2004.

[4] A. Diosi, G. Taylor and L. Kleeman. Interactive SLAM using Laser and Advanced Sonar. *in Proc. of the IEEE International Conference on Robotics and Automation,* April 2005, Barcelona, Spain.

[5] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernández-Madrigal, and J. González. Multi-Hierarchical Semantic Maps for Mobile Robotics. *in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* August 2005, Edmonton, AB, Canada.

[6] J. Folkesson, P. Jensfelt, and H.I. Christensen. Vision SLAM in the Measurement Subspace *in Proc. of the IEEE International Conference on Robotics and Automation,* April 2005, Barcelona, Spain.

[7] A. Green, H. Hüttenrauch, K. Severinson-Eklundh. Applying the Wizard-of-Oz Framework to Cooperative Service Discovery and Configuration. *in Proc. of the 13th IEEE International Workshop on Robot and HumanInteractive Communication,* September 2004, Kurashiki, Okayama, Japan.

[8] A. Green, H. Hüttenrauch, K. Severinson-Eklundh, B. Wrede, S. Li. Integrating Miscommunication Analysis in Natural Language Interface Design for a Service Robot. *submitted*

[9] D. Kirsh. The intelligent use of space *Artificial Intelligence,* 73:31–68, 1995.

[10] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy. *in Proc. of the IEEE International Conference on Robotics and Automation,* April 2004, New Orleans, LA, USA.

[11] T. Kyriakou, G. Bugmann, and S. Lauria. Vision-based urban navigation procedures for verbally instructed robots. *Robotics and Autonomous Systems,* 51:69–80, January 2005.

[12] T.P. McNamara. Mental Representations of Spatial Relations. *Cognitive Psychology,* 18:87–121, 1986.

[13] C.L. Sidner, C. Lee, C.D. Kidd, N. Leash, C. Rich, Explorations in engagement for humans and robots. *Artificial Intelligence,* 166(1):140-164, 2005

[14] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid Simultaneous Localization and Map Building: a Natural Integration of Topological and Metric. *Robotics and Autonomous Systems,* 44:3–14, 2003

[15] A. Tapus, G. Ramel, L. Dobler, and R. Siegwart. Topology Learning and Place Recognition using Bayesian Programming for Mobile Robot Navigation. *in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* September 2004, Sendai, Japan.

[16] E.A. Topp and H.I. Christensen. Tracking for following and passing persons. *in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* August 2005, Edmonton, AB, Canada.

Hierarchical Map Building Using Visual Landmarks and Geometric Constraints *

Zoran Zivkovic and Bram Bakker and Ben Kröse
Intelligent Autonomous Systems Group
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{*zivkovic,bram,krose*}*@science.uva.nl*

***Abstract*— This paper addresses the problem of automatic construction of a hierarchical map from images. Our approach departs from a large collection of omnidirectional images taken at many locations in a building. First a low-level map is built that consists of a graph in which relations between images are represented. For this we use a metric based on visual landmarks (SIFT features) and geometrical constraints. Then we use a graph partitioning method to cluster nodes and in this way construct the high-level map. Experiments on real data show that meaningful higher and lower level maps are obtained, which can be used for accurate localization and planning.**

***Index Terms*— mobile robots, vision based navigation, hierarchical map building, topological map**

I. Introduction

Mobile robot localization and navigation requires an internal representation of the environment. Traditionally such a model is represented as a 2D geometric model of the workspace of the robot, indicating admissible and non-admissible areas. Because of recent progress in sensor technology (vision sensors, 2D laser scanners), these models tend to become quite complex (such as 3D planar maps with texture, 3D landmark positions), resulting in a large number of parameters that have to be stored and estimated.

Hierarchal approaches combining higher level conceptual maps (usually topological maps - graph structures with nodes representing places and edges or links representing possible transitions) with lower-level, geometrically accurate, local maps have a number of advantages. One of the problems with complex (3D) maps is that the number of parameters which have to be estimated in a SLAM procedure increases very fast with the spatial extent of the map. The advantage of splitting the representation into smaller parts is that it makes better parameter estimation possible, if the new problem of maintaining consistency between local representations can be solved [12]. A second advantage of a hierarchical representation is that hierarchical path planning methods can be used. We show in [3] that such planning methods have computational advantages over non-hierarchical planning methods. Finally, a third advantage of a hierarchy of maps is that it can facilitate the interaction of the robot with humans, because the elements in the higher-level map (e.g., the nodes in the graph) can be made to correspond to concepts that make sense to humans (rooms, corridors), instead of metric (x,y) coordinates that are not intrinsically meaningful to humans in office and home environments.

The issue addressed in this paper is how to create a higher level conceptual map which can be used in a hierarchical framework. Different approaches have been proposed in earlier work. In human augmented mapping, a human supervisor indicates which places are to serve as nodes in the graph [2]. Alternatively, an existing metric representation can be used to derive a higher level topological map using geometrical methods such as generalized Voronoi graphs [5]. It is also possible to use sensory data directly for the creation of a higher level map. In [16], [10] a set of images of the robot's environment is grouped based on the presence of a number of automatically extracted landmarks.

In this paper we describe an alternative algorithm for generating a higher level topological map directly from images. The algorithm is based on an appearance-based representation, which is a representation where the environment is not modelled geometrically, but as an 'appearance map' that consists of a collection of sensor readings obtained at various poses [11],[14],[8]. In our approach we do not assume the poses to be known but just use an unordered collection of omnidirectional images taken at many places in the building. The algorithm first constructs a graph ('low level' topological map) from all images by using a grouping criterion that takes into account both the presence of the visual landmarks (SIFT features) and the constraints imposed by the geometry of the environment. This initial step is similar to an algorithm from a different field [21] that was used to group images

*The work described in this paper was conducted within the EU FP6-002020 COGNIRON ("The Cognitive Companion") project. We would also like to thank Olaf Booij for useful comments.

from the same scene of a TV movie. We further define a criterion for grouping the images of the environment so that images from a convex area, a room for example, are naturally grouped together. This criterion corresponds to the normalized graph cut criteria from graph theory [9]. The exact solution is computationally expensive and we use a standard approximate solution [9].

In section 2 we give a brief overview of space representations in robotics and relate them to the method presented in this paper. In section 3 we describe how to generate a low-level topological map of the space from the images of the environment. Section 4 gives some details about the underlying computer vision algorithms needed to do that. Section 5 describes our graph-theoretic method for grouping the images and extracting a higher level conceptual map of the space from the low-level topological map defined in sections 3 and 4. Our experimental results are presented and discussed in Section 6.

II. Related Work - Higher Level (Topological) Maps From Images

Our method is an appearance map method based on images obtained by the robot using a visual sensor. The images are 2D projections of the 3D space. Standard algorithms for 3D reconstruction from images [7] usually extract the metric information incrementally. Typically the different levels of extracted metric information are:

Step 1: At this level images are grouped based solely on their immediate appearance. Images from the same part of the environment (a room, a corridor) are expected to look similar. Typically, images are grouped together by determining whether they have similar landmarks. This grouping based on immediate image appearance produces a straightforward higher level representation of space. However, in large environments the probability that images from completely different places are grouped together can become high. This is sometimes called 'perceptual aliasing'. To reduce perceptual aliasing, [16] proposes to take into account the horizontal ordering of landmarks in the image, yielding what they call 'fingerprint representations'. Similarly, [10] proposes a global description of images using SIFT features [13] as landmarks and their distribution within the image. This provided a more distinctive representation of the space.

Step 2: At this level the images are grouped based on their immediate appearance but also on the geometry of the space. From two images and a set of matching landmarks one can perform two-view geometric reconstruction of the space (see [7] and section IV). This requires that not only similar landmarks are present, but also that they come from the same real-world 3D positions (up to a scale factor). This requirement is much stricter than those in [16],[10]. Therefore, perceptual aliasing is very rare even in large environments (see [21] and also the remainder of this paper).

Step 3: By matching the landmarks over more than two views it is possible to reconstruct the camera poses for the images, 3D positions of the landmarks, and finally perform dense 3D reconstruction of the space (up to a scale factor) [7]. This dense 3D reconstruction can then be used to obtain a precise 2D geometrical map of the environment. A higher level conceptual map can, in turn, be extracted from this 2D geometrical map using the methods described in [23],[15]. Note that to apply these methods we need to use complex and computationally expensive algorithms to perform the complete 3D reconstruction, and that in general the 3D reconstruction problem cannot be considered completely solved, especially for large environments.

The metric reconstruction in this paper stops at step 2, where geometric constraints are imposed, and for example information about occlusions and visibility is already present. In the remainder of this paper we show (sections 3-5) how to use this information and build a natural higher-level representation of the space.

III. Lower Level Topological Map From Images Using Appearance and Geometrical Constraints

A general definition of a topological map is that it is a graph-like representation of space. A set of n nodes V of the graph represent distinct positions in space, and edges or links of the graph encode how to navigate from one node to the other [6]. The nodes and the edges can be enriched with some local metric information.

In this paper, as is typical in appearance-based approaches, each node represents a location and corresponds to an image taken at that location. As the result from n images we get a graph with n nodes that is described by a symmetric matrix S called the 'similarity matrix'. For each pair of nodes $i,j \epsilon [1, ..., n]$ the value of the element S_{ij} from S defines the similarity of the nodes. In our approach S_{ij} is equal to 1 if and only if it is possible to perform 3D reconstruction of the local space from the two images corresponding to the nodes. Otherwise there is no link between the nodes and $S_{ij} = 0$. Our 3D reconstruction is based on the Scale Invariant Feature Transform (SIFT) features [13] as the automatically detected landmarks in the image. The algorithm we are using for the 3D reconstruction is described in more detail in section IV. An example of such a graph that we obtained from a real data set is given in figure 2b.

This graph contains, in a natural way, information about how the space in an indoor environment is separated by the walls and other barriers. Images from a convex space, for example a room, will have many connections between them, and just a few connections to images from another convex

space, for example a corridor, that is connected with the room via a narrow passage, for example a door. In section 5 we describe how to extract such groups of images automatically from the graph (V, S).

There are various ways to define the similarity metric for S_{ij}. The simple metric we use is directly related to the robot navigation task. For localization and navigation the robot can use the same algorithm as the one used to define the edges of the graph (V, S). An edge in the graph denotes that 3D reconstruction is possible between the images that correspond to the nodes. This also means that if the robot is at one node it can determine the relative location of the other node. Therefore, if there are no obstacles in between, the robot can directly navigate from one node to the other (as, e.g., in [17]). If there are obstacles, one could rely, for example, on an additional reactive algorithm for obstacle avoidance that is using range sensors. Furthermore, additional information can be associated with the edges of the graph. For example, if we reconstruct the metric positions of the nodes (using the images or if we measure them in some other way), we could also associate the Euclidean distance between the nodes with each edge. This could be used for better navigation and path planning using the graph. However, this is beyond the scope of this paper.

IV. Visual Landmarks and Geometric Constraints

Having described the general process of constructing the lower level topological map, we proceed to describe some of the details of the underlying computer vision algorithms. First we extract distinctive points from images. Examples are a corner, T-junction, a white dot on black background etc. Such points are often used in the computer vision community as automatically detected landmarks. Here we use the SIFT feature detector [13]. The SIFT feature detector extracts also the scale of the feature point and describes the local neighborhood of the point by a 128-element rotation and scale invariant vector. This vector descriptor is also robust to some light changes.

A. Matching Landmarks

Visual landmarks are used often in robotics for navigation [22],[19],[18]. It is possible to reconstruct both the camera images and the 3D positions of the landmarks by matching (or tracking) landmarks through images. On-line simultaneous localization and reconstruction of landmark positions was presented in [1], but currently only for small scale environments.

In this paper we consider the general case when we start with a set of unordered images of the environment. This is similar to [20]. In practice we often have some information about ordering of the images (e.g. a movie as in [1]) or some other sensor readings (odometry for example), which should be used in that case.

Most 3D reconstruction algorithms [7] start with finding similar landmarks in pairs of images. When two images are consecutive frames of an image sequence we could track the landmarks from one image to the other [1]. However, it is much more difficult to find matching landmarks in an unordered set of images. Firstly, we need to check all the pairs of images, which is computationally expensive. Secondly, there are no additional constraints as is generally the case in an image sequence.

In this paper we use a heuristic similar to [21]. For each landmark from one image we find the best and the second best matching landmark from the second image. The goodness of the match is defined by the Euclidean distance between the landmark descriptors. If the goodness of the second best match is less then 0.8 of the best one it means that the match is very distinctive. According to the experiments in [13], this typically discards 95% of the false matches and less then 5% of the good ones. This is repeated for each pair of images and it is computationally expensive. Fast approximate methods were discussed in [13]. Since our data sets were not very big we performed the full extensive search.

B. Geometric Constraints

The method described in the previous section finds the possible matches for each pair of images from our data set. Let there be N matching landmark points between the images m and l. The 2D image positions of the points in the m-th image in the homogenous coordinates are denoted as $\{\vec{p}_1^m, ..., \vec{p}_N^m\}$. The corresponding points in the l-th image are $\{\vec{p}_1^l, ..., \vec{p}_N^l\}$. If the i-th point belongs to the static scene, then, for a projective camera, the positions are related by:

$$(\vec{p}_i^m)^T F \vec{p}_i^l = 0 \quad (1)$$

where the matrix F is also known as the 'fundamental matrix'. Estimating F is an initial step for 3D space reconstruction from images.

In case there are initially many false matches [21], they must be removed using a method to detect and remove outliers. Standard robust M-estimators are commonly used, which can deal with a limited number of outliers. If there are more outliers, the robust algorithm called RANSAC is commonly used [7]. It was shown [24] that a combination that performs best in many cases is when RANSAC is used first and then the M-estimator. Here, we use the distinctive matches criterion, described above and in [13], which already discards many false matches. In our experiments we observed that the number of false matches is small and it is possible to use the robust M-estimator directly. We used the Huber

M-estimator and the standard 8-point algorithm [7] for estimating the fundamental matrix F.

Residuals of fitting the model (1) to each pair of images [7] are used to calculate the global standard deviation σ_{global}. This standard deviation is used to decide when the fundamental matrix is properly calculated. The σ_{global} is estimated robustly using the maximum absolute difference estimate. The whole procedure, then, is as follows:

- extract SIFT landmarks from all images
- find distinctive matches between each pair of images
- if there are more than 8 matches:
  - estimate the fundamental matrix using the M estimator (could be RANSAC)
  - discard matches that deviate more than $2.5\sigma_{global}$
  - if there are still more than 8 matches, add an edge in the graph - set the similarity between these images to 1.

V. Constructing Higher Level Topological Map Using Graph Cuts

The central idea behind our method to construct the higher level topological map is to cut the lower level topological map (described above) into a number of separate clusters, each of which becomes a higher level node or higher level state. We will start by introducing some graph-theoretic terms. The *degree* of the i-th node of the graph (V, S) is defined as the sum of all the edges that start from that node: $d_i = \sum_j S_{ij}$. For nodes A (where A is subset of V), *volume* is defined as $vol(A) = \sum_i d_i$. $vol(A)$ describes the 'strength' of the interconnections within the subset A. Graph V can be divided into two subsets A and B by cutting a number of edges. The sum of the values of the edges that are cut is called a graph cut:

$$cut(A, B) = \sum_{i \epsilon A, j \epsilon B} S_{ij} \quad (2)$$

One may cut V into a number of clusters by minimizing (2). This would mean that the graph is cut at the weakly connected places, which usually correspond to doors between the rooms or other narrow passages. However, such segmentation criteria often leads to undesirable results. For example, if there is an isolated image connected to the rest of the graph by only one link, then by cutting only this link we minimize (2). To avoid such artifacts we use a *normalized* version. The normalized graph cut separates the graph V into two subsets A and B by minimizing the following criterion:

$$nCut(A, B) = (\frac{1}{vol(A)} + \frac{1}{vol(B)})cut(A, B). \quad (3)$$

Minimizing this criterion means cutting a minimal number of connections between the two subsets but also choosing subsets with strong interconnections. This criterion naturally groups together images from a convex area, like a room, and makes cuts between areas that are weakly connected. The algorithm is simply applied again to obtain more clusters. Finding the optimal solution is computationally expensive. In this paper we use the fast approximate solution from [9].

The following scenario can give another perspective on the normalized cut criterion. An edge means that the robot might navigate from one node to the other as described in Section III. If we assume that the robot randomly moves from a node to a connected node, it is possible to show [9] that: $nCut(A, B) = P(A \rightarrow B|A) + P(B \rightarrow A|B)$. Here $P(A \rightarrow B|A)$ is the probability of jumping from subset A to subset B if we are already in A and $P(B \rightarrow A|B)$ is the other way around. This means that with this random movement, the segmentation is such that the robot has the lowest probability of moving from one cluster to the other.

In [3] we show how path planning can be done using the resulting hierarchical map (the combination of the lower level and higher level topological map), and we show that planning is actually much more efficient using the hierarchical map, compared to just using the lower level map.

VI. Experiments

The experiments described in this paper were designed to investigate the validity of the method to extract the lower level topological map from the images, and the method to extract the higher level topological map from the lower-level map. The experiments were performed using a robot equipped with an omnidirectional camera with a hyperbolic mirror. Circular images were first transformed to panoramic images. Next, the SIFT features were extracted using the standard method [13].

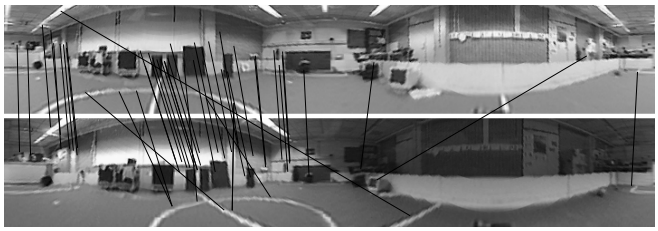
A. Experiment 1: Robustness

Some experimenting was done to test robustness for variability in the images. Figure 1 illustrates the robustness of the method. Despite the different light conditions and occlusions, there were still enough matches to estimate the fundamental matrix. Note that constraint (1) did not remove all false matches. Matches that are false but still close to constraint (1) are not removed.

B. Experiment 2: Perceptual aliasing

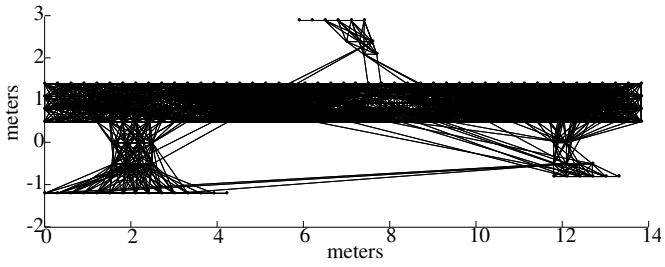
From a data set of 234 images from an office environment we constructed the (lower level) graph using the method described in Section III. The links and the nodes are shown in figure 2. The environment consisted of 3 rooms and a corridor. Two rooms were on one side of the corridor and one on the other side (see also figure 3 which shows the actual layout of the rooms). Figure 2a presents the graph when we match images based only on the presence of the

a) occlusion (the person in the middle of the image)

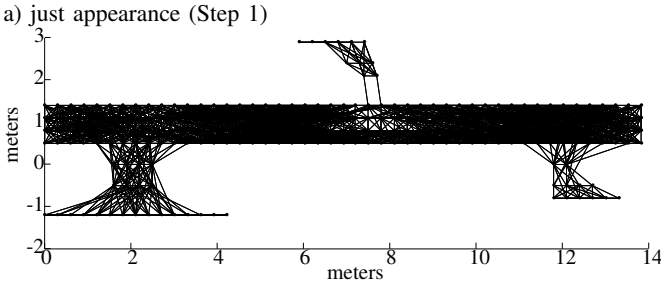


b) different light conditions (two lights turned on or off)

Fig. 1. Pairs of panoramic images taken at approximately 1m distance from each other. The lines indicate the matching landmarks between the two images.



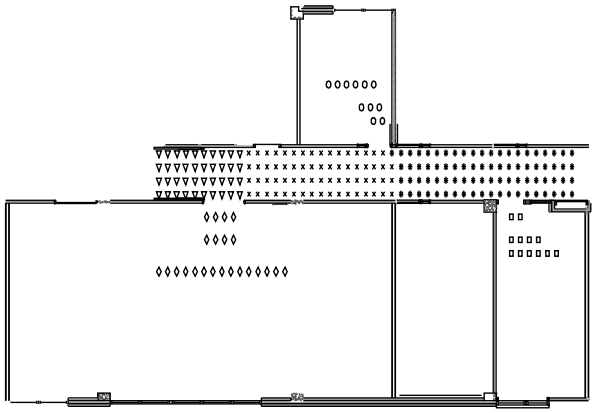
a) just appearance (Step 1)



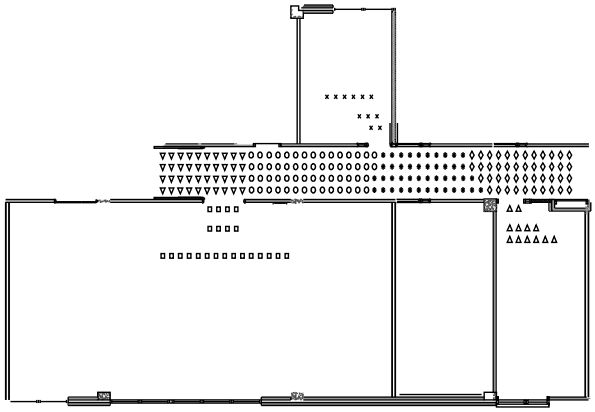
b) with geometric constraints (Step2)

Fig. 2. Reducing perceptual aliasing using geometric constraints. Bird's eye view of the environment with the locations where each of images were taken.

landmarks (see section 2, step 1). The result of taking into account the geometrical constraints is that from the total of 3077 edges, 541 were discarded. For our environment, this removed all perceptual aliasing problems from the graph, as shown in figure 2b.



a) 6 clusters



b) 7 clusters

Fig. 3. Higher level conceptual grouping using minimal normalized cuts. Bird's eye view of the environment with the locations where each of images was taken. Each symbol indicates a specific higher level state. The grouping is obtained directly from the images without using the known ground truth locations.

C. Experiment 3: Building hierarchical map

The normalized graph cut clustering algorithm (section V) was applied to the graph shown in figure 2b. The results, presented in figure 3, show meaningful and natural segmentation of the space. Note that this segmentation was obtained directly from the images in an unsupervised way. One only needs to select the number of clusters, cluster labels need not be assigned to images by a user, and ground truth (actual positions where images were captured) is not used. The results for two different numbers of clusters are shown in the figure.

D. Experiment 4: Higher Level Localization

We applied the algorithm to a data set that contained images that cover a much larger area. Again, a meaningful and natural segmentation of the space is obtained (see figure 4).

For this data set, we perform an experiment to investigate whether the robot can, given a single image it is assumed to observe currently, determine the correct corresponding higher level cluster or higher level state. We select one image from the data set and assume that this is the image observed by the robot. We use the rest of the images as the map. We compute the links of the current image to the images in the map (Section III). The current higher level cluster is then estimated by the robot in a simple way: we look at the cluster labels of the images that have links to the current image and decide the current image's cluster label by a majority vote. This data base has 240 images. For only 5 images the higher level cluster was estimated incorrectly, and they were all at the borders of the clusters.

Note that without any additional information we need to check all the images in order to find the higher level node. It is also possible to speed up this process as discussed in [4].

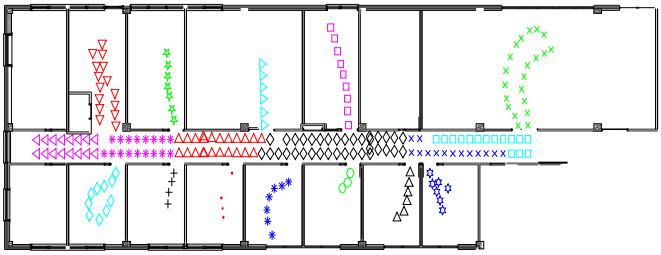


Fig. 4. Higher level conceptual grouping using minimal normalized cuts in a larger environment. Bird's eye view of the environment with the locations where each of the image was taken. Each symbol indicates a specific higher level state. The grouping is obtained directly from the images without using the known ground truth locations.

VII. Conclusions and Further Work

We presented an algorithm for automatically generating hierarchical maps from images. Lower level maps are directly derived from images, higher-level maps are derived from the lower level maps. Experiments on real data show that meaningful hierarchical maps can be obtained. Advantages include its robust handling of the complexities of vision, its appearance-based nature which does not require extensive estimation of metric information, and the possibility for efficient path planning [3]. Our method currently requires one to specify only the number of clusters. In future work we would like to automatically select the number of clusters as well. Online versions of the algorithm (see [4])are also interesting for further research.

References

[1] A.J.Davison and D.W.Murray. Mobile robot localization using active vision. *In Proceedings of the European Conference on Computer Vision*, 1998.

[2] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. I. Christensen. Navigation for human-robot interaction tasks. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1894–1900, April 2004.

[3] B. Bakker, Z. Zivkovic, and B. Kröse. Hierarchical dynamic programming for robot path planning. In *In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[4] O. Booij, Z. Zivkovic, and B. Krose. Pruning the image set for appearance based robot localization. In *In Proceedings of the Annual Conference of the Advanced School for Computing and Imaging*, 2005.

[5] H. Choset and K. Nagatani. Topological simultaneous localisation and mapping: Towards exact localisation without explicit localisation. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, April 2001.

[6] E.Remolina and B.Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152:47–104, 2004.

[7] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision, secon edition*. Cambridge University Press, 2003.

[8] S. D. Jones and J. L. Crowley. Appearance based processes for visual navigation. *IEEE International Conference on Intelligent Robots and Systems, France*, 1997.

[9] J.Shi and J.Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Anlysis and Machine Intelligence*, 22(8):888–904, 2000.

[10] J. Kosecka and F. Li. Vision based markov localization. *In Proceedings of the IEEE Robotics and Automation conference*, 2004.

[11] B.J.A. Krose, N. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 6(19):381–391, 2001.

[12] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *In Proceedings of the International Conference on Robotics and Automation ICRA*, New Orleans, May 2004.

[13] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, 2004.

[14] S. Nayar, S. Nene, and H. Murase. Subspace methods for robot vision. *CUCS-06-95, Technical Report, Department of Computer Science, Columbia University*, 1995.

[15] P.Beeson, N.K.Jong, and B.Kupiers. Towards autonomous place detection using the extended voronoi graph. *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[16] P.Lamon, A.Tapus, E.Glauser, N.Tomatis, and R.Siegwart. Environmental modeling with fingerprint sequences for topological global localization. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, USA*, 2003.

[17] I.Shimshoni R.Basri, E.Rivlin. Visual homing: Surfing on the epipoles. *International Journal of Computer Vision*, 33(2):117–137, 1999.

[18] R.Sim and G.Dudek. Learning and evaluating visual features for pose estimation. *Transactions on Robotics and Automation International Conference Computer Vision*, 1999.

[19] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson. Landmark selection for vision-based navigation. *In Proceedings of the International Conference on Intelligent Robots and Systems, Japan*, 2004.

[20] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets or 'how do i organize my holiday snaps'. *In Proceedings of the European Conference Computer Vision*, 2002.

[21] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92:236–264, 2003.

[22] S. Se, D.G.Lowe, and J.Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 8(21):735–758, 2002.

[23] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[24] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal on Computer Vision*, 24(3):271–300, 1997.