Knowledge Based Image and Speech Analysis for Service Robots

U. Ahlrichs, J. Fischer, J. Denzler, Chr. Drexler, H. Niemann, E. Nöth, D. Paulus
Chair for Pattern Recognition (Computer Science 5)
University Erlangen–Nuremberg
Martensstr. 3, 91058 Erlangen, Germany
ahlrichs@informatik.uni-erlangen.de

Abstract

Active visual based scene exploration as well as speech understanding and dialogue are important skills of a service robot which is employed in natural environments and has to interact with humans. In this paper we suggest a knowledge based approach for both scene exploration and spoken dialogue using semantic networks. For scene exploration the knowledge base contains information about camera movements and objects. In the dialogue system the knowledge base contains information about the individual dialogue steps as well as about syntax and semantics of utterances. In order to make use of the knowledge, an iterative control algorithm which has real–time and any–time capabilities is applied. In addition, we propose appearance based object models which can substitute the object models represented in the knowledge base for scene exploration. We show the applicability of the approach for exploration of office scenes and for spoken dialogues in the experiments. The integration of the multi–sensory input can easily be done, since the knowledge about both application domains is represented using the same network formalism.

1: Introduction

While most robots used in industry are highly specialized for a certain task, service robots aimed at the use in populated environments must be equipped with human–like skills to be able to cope with many different kinds of disturbances. They have to operate in changing environments with unexpected moving obstacles and they have to identify and manipulate objects at unknown positions which makes an exploration necessary.

Current, state of the art projects try to develop robots which meet these requirements. The explorative capabilities of MORTIMER, developed at the University of Karlsruhe and employed as a footboy in a hotel, is still solely based on sonar sensors and a laser scanner and navigates with a built–in map [15]. The goal of another project PRIAMOS is, used to investigate problems like active perception, exploration, and machine learning based on the fusion of various sensor modalities. MINERVA's [5] localization tasks as a museum's guide are solved by the CONDENSATION algorithm using visual data and it's counterpart RHINO [4], besides doing also tour guides, is capable of finding and fetching previously learned objects which have been presented in front of its cameras.

This work was partially supported by the "Deutsche Forschungsgemeinschaft" under grant NI 191/12–1, and the "Bayerische Forschungsstiftung" under grant DIROKOL. Only the authors are responsible for the content.

One can see from current research efforts that learning and vision play key roles in developing robots which are intended to aid and work together with humans. This goal is also pursued by the recently started project DIROKOL[1] which is partially treated by our group. The project aims at the development of a service robot to aid people in private or public health care environments. Knowledge of the expected environment and the objects to be manipulated is vitally important in order to perform fetch–and–carry services, to clean, and to help disabled people. It should integrate seamlessly in, for example, hospital environments without the need for installing artificial navigation aids or electric door openers.

Not only the execution of tasks is important for service robots but also the way of how to command them. The demand for userfriendlyness leads naturally to a dialogue controlled speech interface, which enables everyone to communicate easily with the robot. This is not only needed for convenience but also for lowering the inhibition threshold for using the robots which still might be a problem for wide–spread usage. For strongly handicapped persons, unable to use keyboards or touchscreens, speech understanding and dialogue is one of the main preliminaries. Almost all autonomous systems mentioned above at best provide rudimentary speech processing capabilities and do not go beyond spoken command recognition.

The complexity inherent in the problems leads to knowledge based approaches as they make modeling of complex coherencies possible. In DIROKOL this is intended to be used at several places. Well understood is the use of semantic networks for speech analysis and natural language dialogue systems. As a new approach, semantic network based scene exploration is used to achieve an active control strategy for investigating unknown environments. To be robust in scattered environments, common to service robot scenarios, it is important to use all the information available about the object. Therefore object models are stored, and — dependent on the requirements — appropriate information is used to achieve a reliable recognition.

Image and speech understanding need not to be looked at separately. A fusion of the multi–sensory input should in principle improve the overall performance of both problems: speech might be used together with gesture recognition to find objects, and analyzing gestures might help in understanding speech. Performing a dialogue, i.e. acquiring missing information, will also improve the acceptance of autonomous systems.

For 15 years our group has worked on semantic networks for knowledge based pattern analysis [39]. Independently, problems in the area of image processing [10, 36, 41, 46] as well as speech understanding [24] have been successfully solved. The systems had to deal with static [42] as well as dynamic environments [10, 40, 37]. Despite the fact, that no integration of speech and image analysis at the knowledge based level has been done yet, the same underlying concepts have been applied in both areas, namely the semantic network formalism ERNEST and a control algorithm based on the A^* graph search [33]. This paper summarizes current research at our lab and contributes to the question, how knowledge based image and speech processing can be integrated into service robots. The first step will be to install active scene exploration and a dialogue system on a robot for man-machine-communication. A future step — and the even more interesting one — will be the integration of image and speech at the semantic network level For these open problems in general we do not claim to have solutions or systems running currently, but we argue that with such a common framework which we use for knowledge based image and speech understanding.

The paper is structured as follows. In the next section we motivate our work by describing a typical scenario for service robots in hospitals. In Section 3 the semantic network formalism ERNEST is shortly presented. To exploit the knowledge represented in a semantic network a control

[1] The acronym stands for DIenstleistungsROboter in KOstengünstiger Leichtbauweise.

algorithm, e.g. the parallel iterative control, is needed which is introduced in Section 4. Two main topics of our research in the area of knowledge based processing follow: managing a dialogue in the system EVAR, to be able to answer inquiries about the German train timetable (Section 5), and knowledge based active vision for exploring static scenes (Section 6). In both sections, a literature review of related work and extensions of the control algorithm are presented as well as experimental results showing the quality of the knowledge based approach. It has been shown, that processing in semantic networks can be speed up, when using so called *holistic instantiation* [43]. Because of that, in Section 7 we investigate different approaches for 2–D and 3–D object recognition. The results are compared on a data set of the project DIROKOL, i.e. objects which can typically be found in hospitals. The paper concludes with a summary in Section 8.

2: Description of Scenario

Conventional autonomous robots can operate and perform their tasks in many cases without visual and audio capabilities. They can navigate using their dedicated sensors and built–in plans. In contrast, service robots which operate in environments where people are present need capabilities to communicate with trained and untrained persons. This is essential for safety reasons as well as for increasing the acceptance of such technical products by the users. Two major modes of human communication are speech and visual interpretation of gestures, mimics, and possibly lip movements.

The following example scenario is intended to show the necessary abilities of a service robot in hospitals. The robot is called and moves into the room. It asks for a task: "*Hello, what can I do for you?*". The patient lying in his bed wants to get the red book of Russel, Norvig: Artificial Intelligence. He thinks, that the book is located in the bookcase: *"I like to get the book about Artificial Intelligence.*" The robot analyzes the spoken sentence and realizes, that some important information is missing, to initialize the task of fetching the book. It asks "*Where can I find the book?*". "*It's the red one in the bookcase over there*" pointing into the direction of his bookcase, since a second one is located in the room, which is used by the other patient. The system analyzes speech and gesture and finds, that now enough information is available to initialize the fetch–service. It looks toward the bookcase and first actively scans for red objects as hypotheses for the Russel book. Then, from each of the red objects a close–up view is taken by zooming the camera to the corresponding position. The titles are read and finally the book is found and brought to the patient.

This short scenario shows the problems, which a service robot is confronted with when communicating with humans. The human utterances provide incomplete information (where is the book), references occur to already spoken concepts (it — the book — is over there). This is typical for information retrieval dialogue systems, such as e.g., train timetable inquiries. Additionally, sometimes utterances are combined with gestures which need to be analyzed and fused with the corresponding part of the spoken utterance. Another problem occurs, when performing the requested task. In computer vision in many cases the field of view of the camera does not completely cover all relevant parts of the scene (the bookcase). Sometimes in the initially taken image, the objects which are looked for are too small to be analyzed and recognized reliably (the titles of the books in the bookcase). Thus, camera actions must be performed, i.e. changes of the position of the camera as well as adjustments of the camera parameters (focal length, aperture). Both problems also occur in the automatic exploration of an office scene, where different objects in the offices should be found efficiently.

At present, we have no system which is able to perform a task similar to the example given

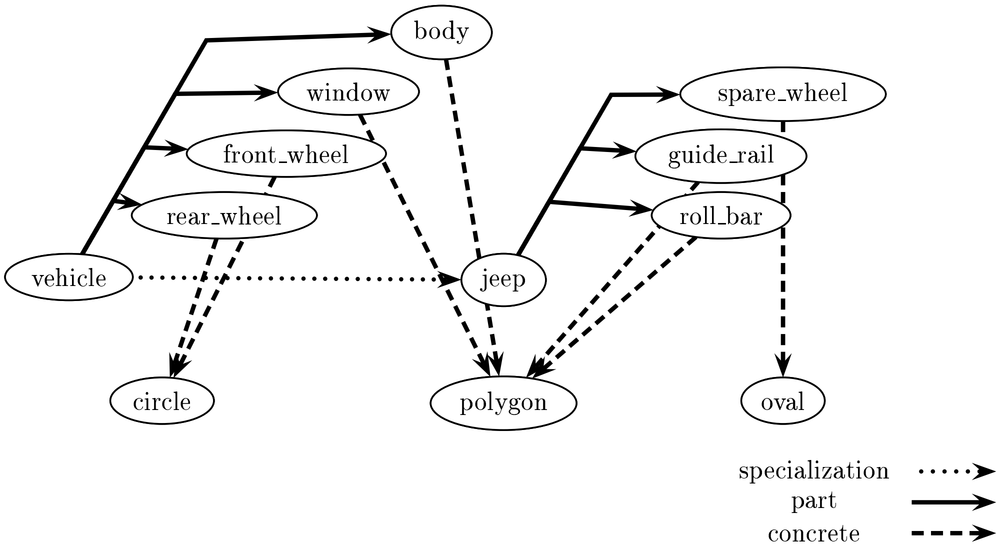


Figure 1. Example of a semantic network

above. But in the following we will present two systems from very similar problem domains. The information retrieval dialogue system (Section 5) in the area of train timetable inquiries corresponds to the information retrieval task, which the robot must perform. The system for active exploration of office scenes (Section 6) corresponds to the task, which must be performed to find requested objects for fetch and carry services. Both applications use the knowledge based approach of semantic networks, which will be shortly introduced in the next section.

3: Semantic Networks

As indicated in Section 1, semantic networks are used for pattern analysis purposes in the following. In this section we introduce this representation mechanism by some examples.

In knowledge based pattern analysis, a pattern $\boldsymbol{f}$ originating from some sensor has to be interpreted. In general, we search for a description of that pattern. This description $\mathcal{B}$ of a pattern $\boldsymbol{f}$ is computed using internal knowledge and an initial description $\mathcal{A}$ of $\boldsymbol{f}$ which can be computed without an explicit model [32]. Due to errors in the initial description $\mathcal{A}$ (arising from noise and processing errors) and ambiguities in the knowledge base, judgments **G** are defined for each computed quantity. The judgments are computed by functions which measure the degree of confidence of a quantity and its expected contribution to the success of the analysis of the whole pattern. They belong to the *procedural* knowledge of the knowledge base.

For representation of the *declarative* task–specific knowledge we use the semantic network formalism of ERNEST [33, 38] which provides the following types of network *nodes*:

- **Concepts** representing abstract terms, events, tasks, actions, etc.;
- **Instances** representing concrete realizations of concepts in the sensor data;
- **Modified Concepts** representing restrictions arising from intermediate results (i.e., the interpretation of part of the sensor data leads to restrictions on the interpretation of the remaining sensor data);

The task–specific knowledge is represented by a *model* $\mathcal{M} = \langle C \rangle$, which is a network of concepts linked to each other by the various types of links.

The main components of a concept C are its *parts* P, its *concretes* K, its *attributes* A and its *structural relations* S. Figure 1 shows an excerpt of a semantic network representing knowledge about vehicles. One can see that, for example, a front wheel is part of a vehicle, and a circle is a concrete of a wheel, since it is not a physical part of it but represents it on a lower level of abstraction.

Since there may be many different possibilities for the realization of a concept, *modalities* are introduced[2] with the implication that each individual modality $H_l^{(k)}$ may define the concept C_k. Furthermore, parts of a concept may be defined as being *obligatory* or *optional*. In Figure 1, modalities are suggested by two different types of a jeep:

- Modality 1:
  obligatory parts: *rear_wheel, front_wheel, body, window, spare_wheel*
  optional parts: *guide_rail*
- Modality 2:
  obligatory parts: *rear_wheel, front_wheel, body, spare_wheel*
  optional parts: *guide_rail, window, roll_bar*

For the definition of properties or features, a concept C has a set of *attributes* A. There may also be *structural relations* S between the attributes of a concept. Each attribute references two functions, one for the computation of its value and one for the calculation of its judgment. For relations, a function is referenced which judges the degree of fulfillment of each relation.

The occurrence of a specific pattern in the sensor data is represented by an *instance* $I(C)$ of the corresponding concept C. For the computation of $I(C)$, all attributes, relations and the judgment of C have to be computed. Before instantiation, instances for all the obligatory parts and concretes of the concepts have to exist.

The goal of pattern analysis, i.e. scene exploration as well as speech dialogue is represented by one or more concepts, the *goal concepts* C_{g_i}. Subsequently, an interpretation $\mathcal{B}$ of $\boldsymbol{f}$ is represented by an instance $I(C_{g_i})$ of a goal concept. Now, in order to find the 'best' $\mathcal{B}$, the computation of an *optimal instance* $I^*(C_{g_i})$ is required. Thus, the interpretation problem is viewed as an *optimization problem* and is solved as such. The semantic network formalism of ERNEST provides an A^*–based control algorithm for solving this problem. In our approach, a *parallel iterative control* is used [9, 11], which provides the pattern analysis system with *any–time* and *real–time* capabilities.

It is thus natural to request the computation of an *optimal instance* $I^*(C_{g_i})$ of a goal concept and define knowledge based processing as the optimization problem

$$\begin{aligned} I^*(C_{g_i}) &= \underset{\{I(C_{g_i})\}}{\operatorname{argmax}}\{\mathbf{G}(I(C_{g_i})|\mathcal{M},\mathcal{A})\} \ , \\ \mathcal{B}(\boldsymbol{f}) &= I^*(C_{g_i}) \end{aligned} \tag{1}$$

with $\mathcal{A}$ being the initial description in case of image analysis or data driven hypotheses for speech understanding.

4: Parallel Iterative Control

The task of the control algorithm is to find an "efficient" solution to the optimization problem stated in (1). Recall that an instance $I(C)$ may be computed for each modality of a concept C

[2] Another possibility for the representation of different realizations of a concept is to define separate concepts for each realization; this, however, prevents a compact knowledge representation.

and for different subsets of the initial segments, which can be, for example, color regions or word hypotheses. Furthermore, if an instance of a concept is to be computed, instances of at least all its obligatory parts and concretes must be available, i.e. computed beforehand. Now, let us define a *primitive attribute* A_i as being an attribute of a concept on the lowest level of abstraction, i.e. the level which is nearest to the initial segmentation. This attribute represents the initial segment for which an instance was computed. For example, in speech understanding a concept on the lowest level of abstraction may model a word hypothesis; it may have an attribute *hypothesis* which represents the start frame and end frame of the word hypothesis, the acoustic quality, the word number in the lexicon, etc. Considering these facts, one can state that the computation of an instance of a goal concept $I(C_{g_i})$ depends only on

- the assignment $(A_i, {O_j}^{(i)})$, $i = 1, \ldots, \mu$, of segmentation results O_j to the primitive attributes A_i of all primitive concepts which have to be computed for the computation of $I(C_{g_i})$, and
- the choice $(C_k, {H_l}^{(k)})$, $k = 1, \ldots, \lambda$, of a modality H_l for each ambiguous concept C_k that enables multiple definitions of an object and for which an instance has to be available for the computation of $I(C_{g_i})$.

This means that for each segment assignment and modality choice exactly one instance of the goal concept, i.e. one interpretation, and its corresponding score is clearly defined and can be computed.

The control algorithm we use [9, 32] is based on this statement. It treats the search for $I^*(C_{g_i})$ as a *combinatorial* optimization problem (see below) and solves it by means of *iterative* optimization methods, e.g. simulated annealing [20], stochastic relaxation [13], and genetic algorithms [14]. By using iterative methods the *any-time* capability is provided, since after each iteration step a (sub-)optimal solution is always available and can be improved if necessary by performing more iterations. Another advantage is that the algorithm allows an easy exploitation of *parallelism*. For example, competing instances for different combinations of segment assignments to primitive attributes and modality choices to ambiguous concepts can be computed in parallel on a local network of workstations by using PVM (*Parallel Virtual Machine*, cf. [12]).

For reasons of efficiency (cf. e.g. [32]), the concept-centered semantic network is first compiled into a fine-grained task-graph, the so-called *attribute network*. This network represents the dependencies of all attributes, relations, and judgments of those concepts to be considered for the computation of goal instances $I(C_{g_i})$. Figure 2 shows how the parallel iterative control principally works. The attribute network is automatically generated in two steps from the semantic network:

- **Expansion**
  Since each concept is stored exactly once in the knowledge base, and it may be necessary to create several instances for a concept during analysis (consider, for example, the concept “SY_Png” in Figure 3, which represents a prepositional group, e.g. “from Hamburg” or “to Munich”: an instance of “SY_Png” is necessary to compute an instance of “S_Source”, another instance of “SY_Png” is necessary to compute an instance of “S_Goal”), the semantic network is first expanded top-down such that all concepts which are necessary for the instantiation of the goal concepts C_{g_i} exist.

- **Refinement**
  The expanded network is then refined by the determination of dependencies between subconceptual entities (attributes, relations, judgments, etc.) of all concepts in the expanded network. For each sub-entity, a node v_k is created. Dependencies are represented by means of directed links $e_{lk} = (v_l, v_k)$ and express the fact that the computation of the sub-entity

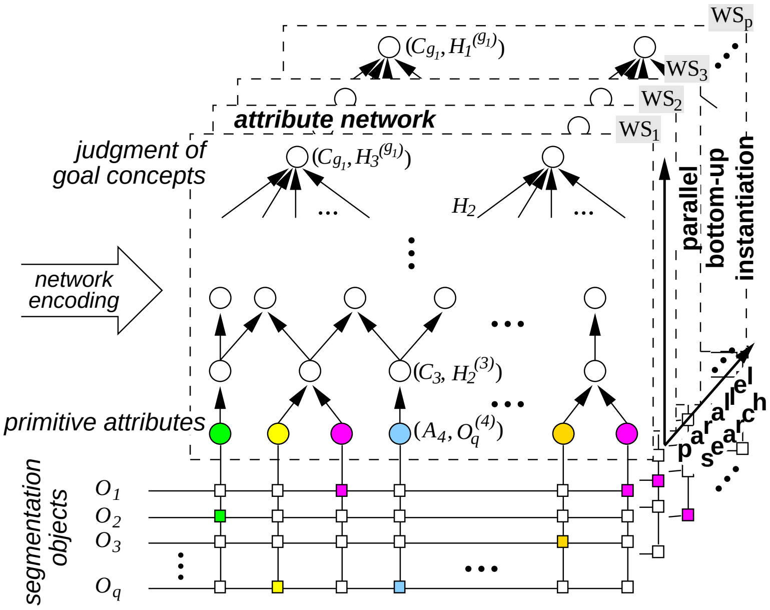


Figure 2. Scheme of the parallel iterative control algorithm.

represented in v_l must have been completed before the computation of the sub-entity in v_k may start.

Both steps are executed off-line, before the analysis, and depend only on the syntax of the semantic network representation. Nodes without predecessors represent primitive attributes which provide the interface to the initial segmentation, and nodes without successors represent the judgments (i.e. confidence measures) of goal concepts (cf. Figure 2). Now, for the computation of instances of the goal concepts, all nodes of the attribute network are processed in a single bottom-up step. Therefore the flow of information is fixed from the primitive attribute nodes to the judgment nodes of the goal concepts. This bottom-up instantiation corresponds to a single iteration of the iterative optimization.

Parallelism can be exploited on the network and on the control level. Each node of the attribute network on the same layer, for example, may be mapped to a multiprocessor system for parallel processing. In addition, several competing instances of the goal concept may be computed in parallel (see above).

Assuming the availability of the judgment $\mathbf{G}$ (cf. Section 3) for the scoring of instances, the instantiation of goal concepts results in a judgment vector

$$\boldsymbol{g} \;=\; (\mathbf{G}(I(C_{g_1})), \ldots, \mathbf{G}(I(C_{g_i})), \ldots, \mathbf{G}(I(C_{g_K}))). \qquad (2)$$

As mentioned above, the search space which the control algorithm has to deal with is determined by the competing segmentation results which can be assigned to the primitive attributes, and by the different modalities of the concepts in the knowledge base. Therefore we define the *state of analysis* of our combinatorial optimization as a vector $\boldsymbol{r}$ which contains the assignment of segmentation results to primitive attributes and the choice of a modality for each concept (if a concept is not ambiguous, it has only one modality):

$$\boldsymbol{r} \;=\; \left[(A_i, O_j^{(i)}); (C_k, H_l^{(k)})\right]^T \quad \text{with } i = 1, \ldots, M \;;\; k = 1, \ldots, N \,. \qquad (3)$$

where M denotes the number of primitive attributes, j the index of the segmentation results, N the number of concepts in the semantic network, and l the index of the modality for each corresponding concept. Now, the result of instantiation is rewritten as a function

$$\boldsymbol{g}(\boldsymbol{r}) \quad = \quad (\mathbf{G}(I(C_{g_1})), \dots, \mathbf{G}(I(C_{g_i})), \dots, \mathbf{G}(I(C_{g_\kappa}))|\boldsymbol{r}), \tag{4}$$

of the state of analysis vector $\boldsymbol{r}$, and a function $\phi(\boldsymbol{r})$ is introduced. This function has to be minimized or maximized (depending on whether it measures the costs or the quality, respectively) by means of the iterative optimization methods already mentioned, e.g. stochastic relaxation. Let us assume that we have only one goal concept C_g and consequently, $\boldsymbol{g}(\boldsymbol{r}) = (\mathbf{G}(I(C_g)) \mid \mathbf{r})$. Now one can choose, for example, $\phi(\boldsymbol{r}) = \boldsymbol{g}(\boldsymbol{r})$. Since $\mathbf{G}$ defines a measure of quality (or confidence) for the computed instance, $\phi(\boldsymbol{r})$ has to be maximized, implying the maximization of $\mathbf{G}$ (and that is exactly what we are looking for: an instance for the goal concept with maximal score).

In a current iteration step, judgments and the corresponding costs for the goal concepts are computed for a current state of analysis $\boldsymbol{r}$. After this, a new state of analysis has to be created. This is done by first selecting with equal probability a tuple $(A_i, O_j{}^{(i)})$ or $(C_k, H_l{}^{(k)})$ from among $\boldsymbol{r}$, and then exchanging the term $O_j{}^{(i)}$ or $H_l{}^{(k)}$, respectively, by a possible alternative, again with equal probability. If the new state generated this way leads to *lower* cost or a *higher* score (depending on the optimization function $\phi(\boldsymbol{r})$), it is accepted in case that the optimization method used is stochastic relaxation. However, this does not allow to escape from local optima. Thus, one can also employ, for example, simulated annealing for optimization, since it allows the acceptance of a new state with higher cost (or lower quality), in order to escape from local optima. The decision for an optimization method depends mainly on the function to be optimized. If it has only one optimum, optimization with stochastic relaxation is appropriate. Iterations are performed until an optimal solution is found or until no more computing time is available. Please recall that after each iteration step a (sub-)optimal solution is available.

Figure 2 shows, for example, that in the current state of analysis for which the attribute network is computed on the first workstation (WS$_1$), the segmentation object O_q is assigned to the primitive attribute node A_4, modality 2 is assigned to the concept C_3, and modality 3 is assigned to the goal concept C_{g_1}. Furthermore, it is shown that competing instances of goal concepts (recall that the computation of instances of goal concepts depends only on the current state of analysis) are computed on several workstations: the current state of analysis for which the $I(C_{g_i})$ are computed on workstation WS$_\mathrm{p}$ differs from that on WS$_1$ at least by the assignment of a different modality to the goal concept C_{g_1}.

In the following Sections, current applications of the parallel iterative control algorithm for speech analysis (information retrieval for German train timetable) and image analysis (exploration of office scenes) are described. Recall that these are separate applications which were chosen in order to show the feasibility of the approach and which differ from the scenario described in Section 2. In each of these applications only one goal concept is defined, which is “explore_office” (explore office, cf. Section 6) and “D_Info_Dia” (information dialogue, cf. Section 5). The function to be optimized is based on the judgment function of these goal concepts and has to be maximized.

5: Information Dialogue

As a framework for our speech understanding task, the dialogue system EVAR [25] is used. This system was initially developed using the semantic network formalism of ERNEST which provides a bottom-up/top-down A^*-based control algorithm and is able to answer queries about the German train timetable. The knowledge base of EVAR is arranged in 5 levels of abstraction:

- *Word hypotheses:* is the lowest level of abstraction and represents the interface between speech recognition and speech understanding; it requests and verifies word hypotheses from the acoustic-phonetic front-end;
- *Syntax:* represents the level of syntactic constituents (e.g. noun group: "*the next train*", prepositional group: "*on Tuesday*"); it involves the identification of syntactic constituents in the set of word hypotheses;
- *Semantics:* is used to model verb and noun frames with their deep case, verifies the semantic consistence of the syntactic constituents, compounds them to larger ones, and performs task independent interpretation (e.g. goal: "*to Hamburg*");
- *Pragmatics:* interprets the constituents sent by the semantic module in a task-specific context (e.g. place of arrival: "*to Hamburg*");
- *Dialogue:* models possible sequences of dialogue acts, it operates in accordance with the level of identified intention of the spoken utterance (e.g. user's first information request: "*I want to go to Hamburg*", systems demand for further information: "*When do you want to leave?*");

The word hypotheses are generated by the acoustic processing module, which analyzes the speech signal and generates, by means of Hidden Markov Models and a stochastic grammar, word hypotheses. This word hypotheses serve as input for the linguistic analysis. Focus of interest here is the linguistic analysis. In our approach, the former A^*-based control provided by ERNEST has been substituted by the parallel iterative control presented in Section 4.

The ultimate goal of the dialogue system is to answer an information query. Therefore, the system must be able to carry out a dialogue with the user, since the user may not provide the system with all the necessary information for a database access. Thus, dialogue steps follow, alternately, by the user and by the system. If the user utters something, supplying the system with information about what he wants to know, the system has to *interpret* the user's utterance. Otherwise, if the system is supposed to react to the users utterance in some way, e.g. by asking for more information, one can say that it has to perform an *action*. User's dialogue steps (interpretation steps) as well as system's dialogue steps (actions) are both modeled in our knowledge base by means of concepts as defined in Section 3. Since interpretation and action steps do not compete at a time, i.e., either an action is performed by the dialogue system or an interpretation step, the concepts modeling these steps are not competing goal concepts C_{g_i} as described in Section 3 and 4. Rather, an estimation of sub-goal concepts has to be performed to predict the next dialogue step. For this purpose, the parallel control algorithm had to be expanded, since it was primarily designed to handle (competing) interpretation steps. In [8], we demonstrated the success of the approach on the interpretation level. In the following section we describe how the parallel iterative control was expanded in order to handle alternating interpretation and action steps.

5.1: Expanding the Control for the Dialogue Steps

Figure 3 shows an excerpt of the semantic network of EVAR. The overall goal, an information dialogue, is represented by "D_Info_Dia". It may begin with the user's initial utterance, "D_U_Inf_Req" (e.g. "*hello, I want a train to Hamburg*"). If the system needs more information for database access, it asks for it, "D_S_Demand" (e.g. "*when do you want to leave?*"). The user will supply the requested information, "D_U_Suppl" (e.g. "*tomorrow in the morning*"). The system may request for a confirmation, depending on the chosen dialogue strategy, "D_S_Conf_Req" (e.g. "*you want to travel to Hamburg tomorrow in the morning. Where do you want to leave from?*"). This may

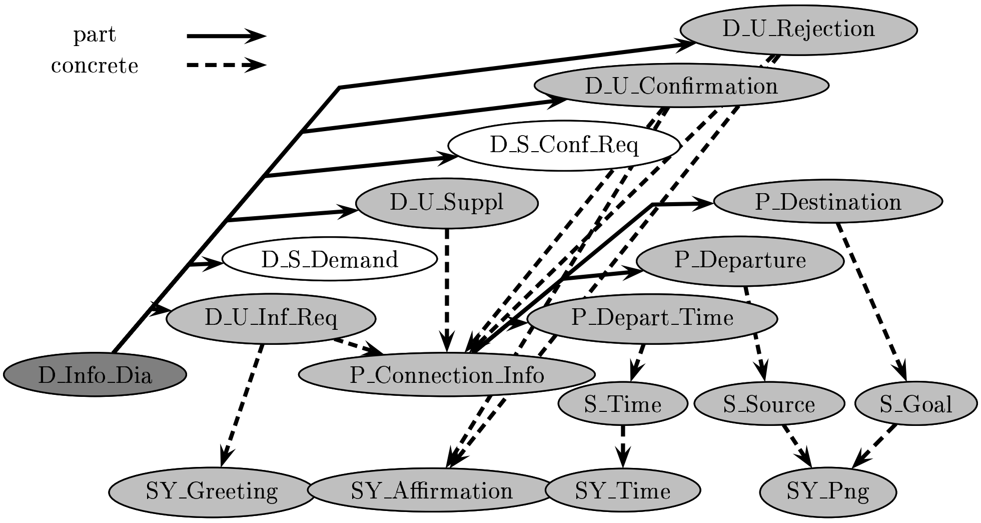


Figure 3. Excerpt from the knowledge base of EVAR showing part of the dialogue model. The white ovals represent the action level (system's dialogue steps), the gray ovals represent the interpretation level, on which linguistic knowledge is represented (user's dialogue steps).

be followed by a confirmation or a rejection by the user, "D_U_Confirmation", "D_U_Rejection". If all necessary information is available, the system accesses the database, retrieves the requested information and provides it to the user, "D_S_Answer", and says goodbye, "D_S_Goodbye" (not shown in the excerpt). It should be noted that these are only some of the 20 dialogue steps represented in the original knowledge base of EVAR.

As it was explained before, the parallel iterative control has to be extended in order to allow a loop of alternating interpretation and action steps. In our speech understanding application, the overall goal concept, which models the hole dialogue is "D_Info_Dia", which we will denominate C_G (note that we use the index G here instead of g, since C_g will be used to denominate sub-goals on interpretation level, as explained below). Let us consider, now, an attribute network computed for C_G. Since C_G models all user and system dialogue steps, an instance for C_G cannot be computed in one step (i.e., a bottom-up processing of all nodes of the attribute network for C_G is not possible). If a bottom-up processing of the network is performed, instances for all dialogue steps (which are parts of C_G) are computed at once, which makes no sense and is not necessary anyway, since user dialogue steps and system dialogue steps do not compete. Thus, the instantiation of user dialogue steps and system dialogue steps have to be handled separately in the attribute network. Therefore, we divide our global optimization task which is to find an optimal instance for C_G, into several local optimization tasks C_{g_i}, C_{a_i}, where C_{g_i} are sub-goal concepts on interpretation level, representing user dialogue steps, and C_{a_i} are sub-goal concepts on action level, representing system dialogue steps. Now, for the computation of an instance of C_G, instances for sub-goals C_{g_i} on the interpretation level (user dialogue steps) have to be considered alternately with instances of sub-goals C_{a_i} on the action level (system dialogue steps). Thus, the additional task of the control is to make a *sub-goal concept estimation.* For this purpose, a *control-shell* was implemented. It computes, in a pre-processing step, sub-networks $dsub_{g_i}$ and $dsub_{a_i}$ out of the attribute network for all C_{g_i} and C_{a_i}. For example, let us assume that the sub-goal C_{g_1} on interpretation level is the user's first utterance modeled by the concept "D_U_Inf_Req". So, $dsub_{g_1}$ will be a sub-network consisting of all nodes of the attribute network which belong to those concepts for which instances have to be computed in order to get an instance for the concept "D_U_Inf_Req". Figure 4 shows this procedure. begins. Thus, for the computation of an instance $I(C_{g_i})$ or $I(C_{a_i})$, only

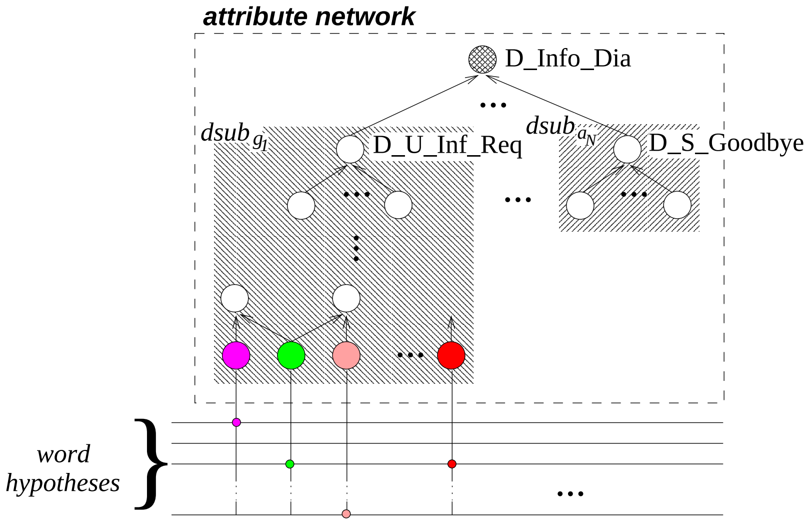


Figure 4. Sub-networks for dialogue steps in the attribute network.

the corresponding sub-network $dsub_{g_i}$ or $dsub_{a_i}$ have to be processed bottom-up. Furthermore, a task-dependent function *next_dsub* is provided by this shell which estimates the sub-goal (or set of competing sub-goals) to be considered next. This estimation is done at the moment by means of some decision rules which were extracted from the former dialogue model implemented with the A^*-based control; it will be substituted by statistical methods in the future. Competing sub-goals at a time are, for example, "D_U_Confirmation" or "D_U_Rejection": a user may confirm (e.g. "*yes, tomorrow at about eight o'clock*") or reject (e.g. "*no, I want to leave today in the evening*") a system's confirmation request (e.g. "*You want to go from Munich to Hamburg tomorrow. When do you want to leave?*"). Thus, after the concept "D_S_Conf_Req" was instantiated, *next_dsub* estimates "D_U_Confirmation" and "D_U_Rejection" as being the set of competing sub-goal concepts to be considered next. The information extracted in each instantiation and the progression of the dialogue is stored in an *information memory*. In Section 5.2, results are cited regarding the user's initial utterance to show the efficiency of the control algorithm in this application domain. At present, the dialogue steps "D_U_Inf_Req", "D_U_Suppl", "D_S_Demand", "D_S_Precision" (the system demands for precision, e.g. "*You want to travel from Munich to Hamburg tomorrow. When do you want to leave tomorrow?*"), "D_S_Answer", and "D_S_Goodbye" have been fully implemented for the new control.

5.2: Experimental Results for Information Dialogue

The goal of the experiments was to evaluate the performance of the system with respect to processing speed and the percentage of correctly analyzed pragmatic intentions, for example:

$$\underbrace{\textit{We}}_{\textsc{Traveller}}\ \textit{want to go}\ \underbrace{\textit{to Hamburg}}_{\textsc{Destination}}\ \underbrace{\textit{today.}}_{\textsc{Dep\_Time}}$$

These are pragmatic information units (e.g. DESTINATION) the system needs to "know" in order to react to the user's request; in the knowledge base of EVAR they are represented by concepts on the pragmatic level, such as “P_Depart_Time”, which models a time of departure in the context of a train timetable information request. The only dialogue step considered was the user’s first request, “D_U_Inf_Req”. Further experiments will be carried out to evaluate the additional dialogue steps. The context for our experiments was as follows:

- goal concept C_G of the attribute network was “D_Info_Dia”;
- the attribute network itself consisted of about *10 500 nodes* and was generated for the dialogue steps “D_U_Inf_Req”, “D_S_Demand”, “D_U_Suppl”, “D_S_Answer”, and “D_S_Goodbye”;
- the dialogues consisted of the steps “D_U_Inf_Req”, “D_S_Answer”, and “D_S_Goodbye”;
- the optimization method used was *stochastic relaxation* (cf. Section 4);
- the function to be optimized was based on the judgment function of the current sub-goal concept and had to be maximized;
- as input for the linguistic analysis we used the *transliterated* utterances (simulating a 100% word-accuracy).

Stochastic relaxation was used since in [9] this method turned out to provide the best results apart from genetic algorithms which lead to even better results but required more overhead, being, thus, less efficient. Parallelization on the control level was simulated on a single processor (parallelization with PVM is being implemented at the moment). The attribute network was processed sequentially.

Two test corpora were used: a corpus of 146 *thought up* and *read* (i.e., not spontaneous) user’s first utterances, 8.3 words (2.7 seconds) per utterance and a total of 447 pragmatic intentions (part of the ASL-Sued corpus[3]) and a corpus of *spontaneous speech* consisting of 327 user’s first utterances collected over the public telephone network, 8.7 words (3.0 seconds) per utterance and a total of 1 023 pragmatic intentions (part of the EVAR-Spontan corpus [6]). Table 1 shows the number of correctly analyzed pragmatic intentions (in %) for EVAR-Spontan and ASL-Sued.

These results were obtained after a careful choice of an initial state of analysis vector based on the incoming word-chain and some heuristic rules. This initialization is presently being replaced by a statistical approach which was implemented for a restricted attribute network and showed to be successful (cf. [18]). This is the reason why quite good solutions have already been found in the first iteration step. One can see that after $N_\mathrm{n} = 5$ iterations using $N_\mathrm{p} = 5$ processors (simulated sequentially on a single processor) 97% and 90% of all pragmatic intentions of the ASL-Sued and EVAR-Spontan corpora were found. The majority of the pragmatic intentions not identified were time specifications which are syntactically more complex than the expressions of other intentions. Furthermore (not shown in Table 1), after $N_\mathrm{n} = 5$ and using $N_\mathrm{p} = 5$ processors, 100% and 94% of all destinations were correctly recognized for ASL-Sued and EVAR-Spontan, respectively. This means that in almost all cases the system is able to maintain a dialogue with the user by confirming the destination location and asking for information it has not yet acquired, e.g. the departure time.

The mean processing time for a single iteration is 0.2 seconds (on a 9000/735 HP-Workstation). Considering that a user’s first utterance consisted on average of 9 words, which means approximately 3 seconds, a *real-time* factor of ≈ 0.3 for five iterations ($N_\mathrm{n} = 5$) on a single processor ($N_\mathrm{p} = 1$) can be computed. Please note that this time evaluation does not include the processing time for computing the word chain out of the acoustic signal. Since parallelization on several workstations was simulated on a single processors for the experiments performed here, using $N_\mathrm{p} = 5$ processors will require 5 times more processing time at the moment. We intend to keep the com-

[3] The ASL-Sued corpus was developed at the University of Regensburg and consists of first user requests of train timetable information collected through Wizard-of-Oz experiments.

	ASL-Sued				
N_n	$N_\mathrm{p}=1$	$N_\mathrm{p}=2$	$N_\mathrm{p}=3$	$N_\mathrm{p}=4$	$N_\mathrm{p}=5$
1	86.2	89.9	92.9	95.0	95.3
5	89.0	92.9	95.7	97.0	97.6
10	91.8	95.0	97.4	98.5	99.1
25	95.9	97.6	98.7	99.6	99.8
50	98.7	99.6	100	100	100

	EVAR-Spontan				
N_n	$N_\mathrm{p}=1$	$N_\mathrm{p}=2$	$N_\mathrm{p}=3$	$N_\mathrm{p}=4$	$N_\mathrm{p}=5$
1	74.0	77.3	78.1	88.3	88.5
5	78.6	88.1	88.6	89.3	90.2
10	83.1	86.4	88.0	90.5	91.1
25	88.2	91.5	91.8	91.9	92.5
50	90.6	92.3	92.6	92.7	92.8
100	92.0	92.7	93.0	93.0	93.0

Table 1. Percentage of correctly analyzed pragmatic intentions for N_n iterations and N_p processors on ASL-Sued and EVAR-Spontan.

munication between processors low when performing parallel processing on several processors by means of PVM, such that the decrease of the real-time factor when performing five iterations on five processors (compared to that for a single processor as computed above) will not be significant. As mentioned before, these results were obtained for the user's initial utterance. Nevertheless, first experiments for the dialogue steps "D_U_Inf_Req", "D_S_Answer", and "D_S_Goodbye" were carried out. The additional processing time, which comprises the prediction of the new dialogue step to be performed, and the instantiation of "D_S_Answer" (including access to the database) and "D_S_Goodbye" is about 0.3 seconds. A speed-up by a factor of approximately 10 could be achieved in first experiments, comparing the system with the new control to the former system with A^*-based control. This has to be confirmed by further experiments.

6: Active Scene Exploration

The goal of our image analysis system is the exploration of arbitrary scenes. As we have already pointed out in Section 1 this is one important skill a service robot has to possess. In order to fulfill an exploration task a closed–loop of acting and sensing is essential. If the robot, for example, wants to localize an object which is not visible for the current camera settings, the pan and tilt angle have to be changed and the analysis has to be started again with a new, more suitable image. Further examples are the modification of focus, if the image is blurred, or the modification of zoom, if an object is too small to be reliably recognized. This adaptation of camera parameters is suggested by the strategy of *active vision* [1]. One goal of this strategy is to change the camera parameters such that they are suitable for later processing steps. The criterion of suitability is task-dependent and has to be defined adequately. In addition to knowledge about changing the camera settings, information about the objects which have to be localized, and about relations between objects has to be provided for the system. We suggest an integrated approach which uses a *unique* representation of objects and camera actions within a knowledge base. The calculation of new camera settings is reduced to selecting an optimal camera action.

In classical image analysis, of course, many knowledge based approaches like VISIONS [16], SPAM [27] or SIGMA [26] are known. But all these systems lack a representation of camera actions. Related work on the selection of actions using Bayesian networks can be found in [35, 23, 21]. In contrast, we use a semantic network for knowledge representation, because we found this kind of representation particularly suitable for the description of objects. This representation is less obvious using Bayesian networks [21].

The integrated knowledge representation of camera actions and objects allows the use of *one* control algorithm. Here we use the parallel iterative control algorithm (Section 4) which is extended as shown in Section 6.3. In order to interpret the represented information, a graph search algorithm can be used for analysis, as well.

In the following we present a system for the exploration of arbitrary office scenes where the scene contains no moving objects. The goal is to find preselected objects in the scene. Because the main contribution of the approach is the conceptional work concerning the integration of camera actions into the semantic network, the object recognition task is simplified by using only red objects. The task of the system is restricted to find three red objects, a punch, a gluestick, and an adhesive tape in an office. The objects need not be visible in the initial set-up and their locations are not fixed inside the scene. Currently no pose estimation is done. The system can easily be extended to solve more ambitious tasks, for example, by integration of the holistic object recognition described in Section 7.

6.1: Declarative Knowledge

The knowledge about the objects and about the necessary camera actions is represented in the semantic network which is depicted in Figure 5. As we have motivated in the introduction, the knowledge base unifies the representation of objects and their relations and the representation of camera actions. The gray ovals represent information which can be found in almost any conventional semantic network. This set of concepts contains, for example, the objects of the application domain, e.g. the concepts “punch”, “gluestick” or “adhesive_tape”, and their concrete representation which is the concept “color_region”. The concepts representing the objects are parts of the concept “office_scene”. The concept “color_region” is a context-dependent part of the concept “subimage_seg”[4]. This means that an instance of the concept “color_region” can only be computed, if an instance of the concept “subimage_seg” exists which establishes a context. The same holds for the concepts “subimage” and “office_image”. The concept “office_image” represents an image which visualizes the whole scene, whereas the concept “subimage” represents only parts of this office image.

In addition to the representation of scene concepts, concepts for camera actions are integrated into the knowledge base. On the highest level of abstraction one can find camera actions which are equivalent to search procedures, in order to find objects in a scene. The first example is the concept “direct_search”. Each instantiation of this concept computes a new pan angle and a new zoom of the camera in such a way, that overview images, which are images captured with a small focal length, are obtained. If we look at them altogether as one image, we get a scan of the whole scene. The second example is the concept “indirect_search”. By instantiation of this concept the search for an object using an intermediate object is performed [47]. Usually large objects like tables or book shelves are used as intermediate objects. These objects have in common that they

[4] In the following the “_seg” part of the concept names stands for segmentation

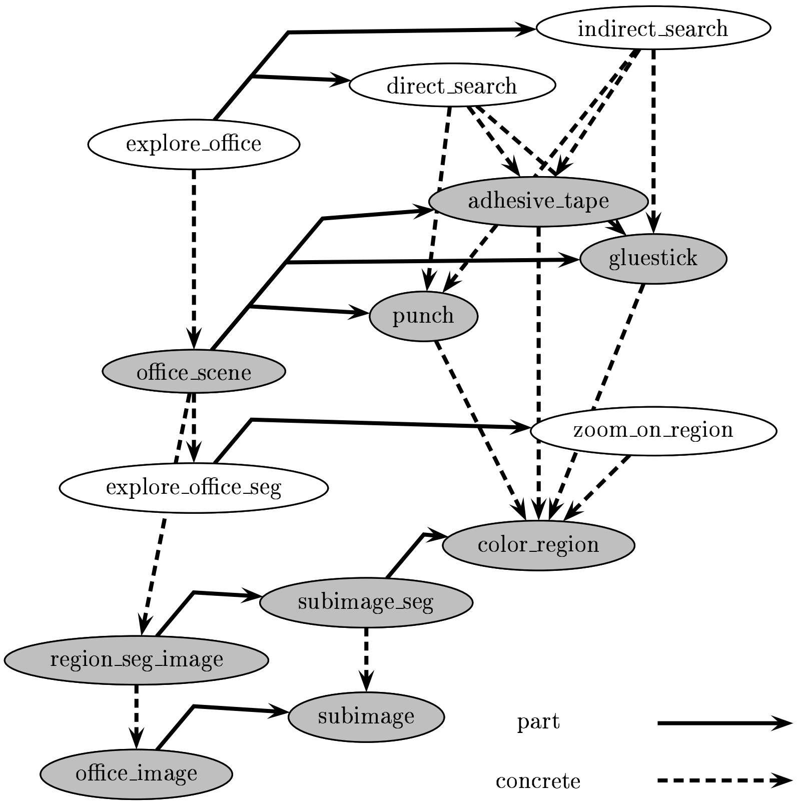


Figure 5. Semantic network which combines the representation of camera actions (white ovals) with the representation of the scene (gray ovals). In addition, the network contains generalizations for some concepts which are left out for the sake of clarity.

are relatively large and therefore can be found in an image captured with a small focal length. This is advantageous, because less images are necessary to scan a whole scene. In addition, the focal length can be increased, if the intermediate object has been found, and a search for the smaller objects can be performed with a high resolution.

On the intermediate level of abstraction the camera action "zoom_on_region" can be found. The effect of this action is a fovealization of a region, that is the pan angle and the zoom of the camera are adjusted. The fovealization is based on the observation that the regions which are found in the overview images and which correspond to the hypotheses for instances of the objects of interest, are too small for a good verification. The effect of this camera action is that close-up views of the hypotheses are taken when the analysis starts again after performing the camera action. This corresponds to an instantiation of the concept "subimage".

In order to represent competing camera actions, i.e. actions which cannot be performed at the same time, we make use of modalities (cf. Section 3). For example, the concept "explore_office" has as parts the concepts "direct_search", and "indirect_search", each of them is represented in one modality of "explore_office". The concept "office_scene" is another example for a concept which has two modalities. One modality contains "explore_office_seg" and "region_seg_image" and the other one contains only "region_seg_image". If the latter modality is chosen during analysis no camera action is performed. In Section 4 we explain how we deal with the ambiguities arising from the modalities.

6.2: Procedural Knowledge

In addition to the declarative knowledge, each concept contains procedural knowledge which consists of the functions for value and judgment computation of attributes, relations and instances.

In the concepts of our scene representation, for example, functions for attribute value computation are defined which compute the color of a region (in the concept "color_region") or the height and the width of an object (in the concepts "punch", "gluestick" and "adhesive_tape"). In addition, each of these concepts contains attributes for the focal length and for the distance of the represented object referring to the close-up views of the objects.

A management of uncertainty is provided by the control based on the judgment functions (cf. Section 4). The judgments are determined using an utility-based decision calculus [19]. The utility of an action can be defined, for example, as the gain in information the system achieves by performing the action. The utility of each action is stored in an utility table, which depends not only on the actions but also on the state of the system, because dependent on the state only some actions are useful. In our approach the state of the system corresponds to the hypothesis if the objects have been found up to a certain point in the scene exploration. Because this hypothesis is not reliable, we have to define a measure of certainty whether the objects have been found or not. In the utility-based decision calculus this measure of certainty is provided by probabilities, for example, the probability whether we have found an adhesive tape. Therefore we use probabilities as judgments of the objects' instances. In order to judge these instances the attribute judgments of the corresponding concepts are needed. These judgments correspond to probabilities, as well, where we assume, that the attributes are statistically independent. The probabilities are calculated using a priori trained normal distributions for the individual attributes, the height, the width, and the color of the objects. During training we calculate the mean and variance of these attributes for each object using 40 images. Using the judgment of the instances a utility for each camera action can be determined. The utilities of the camera actions "indirect_search" and "direct_search", for example, depend on whether an instance of the intermediate objects has already been detected with a high certainty, i.e., a high probability. In this case the indirect search gets an higher utility than the direct search because it is less time consuming in this situation, where the gain in information stays the same.

6.3: Expanding the Control to Actions

The goal in our application is to instantiate the concept "explore_office". Therefore we *alternately* have to interpret the image data and perform camera actions. This alternating computation cannot directly be expressed by neither the syntax of the semantic network nor by the attribute network. If we compute the whole attribute network for our goal in one bottom–up step as described in Section 4, we select the camera actions "direct_search" or "indirect_search" possibly without an optimal interpretation of the concepts "punch", "gluestick" and "adhesive_tape". This makes no sense, because we first need to find an optimal interpretation for the view under the actual camera setting, before deciding which camera action has to be performed next. In addition, if we instantiate the concept "zoom_on_region" and then go on with the bottom–up computation of the attribute network we get an instance of "explore_office" based on image data which was taken with the old zoom setting.

In order to solve these problems, the control for the bottom–up instantiation of the attribute network (cf. Figure 2) is extended by a goal concept estimation as it was already successfully demonstrated for the speech dialogue system (cf. Section 5.1). The instantiation is divided into several data driven instantiations of sub-networks of the attribute network. The division is initiated by specifying sub-goals prior to the computation of the attribute network. This induces an order of the sub-goals in the attribute network. From the sub-goals, the sub-networks can be automatically derived by the network encoder, which is used to transform the semantic network into the attribute network (cf. Section 4). Initial sub-goals are chosen from the lowest level in the network. Analysis starts by the instantiation of the initial sub-goal. This means that the bottom-up computation of the corresponding sub-network is iterated until an optimum is found. Afterwards, the control chooses the next sub-goal to be instantiated, and so on. This process continues until an optimal instance of the goal concept is found.

To give an example: If the user chooses, for example, the concepts “region_seg_image” and “office_scene” as sub-goals, the control starts finding the best instance of “region_seg_image”. Based on the segmentation results provided by “region_seg_image” the control searches for the best instance of “office_scene”. This is done until the goal concept, that is “explore_office”, is reached or a camera action is performed. In the latter case the analysis starts again with the sub-goal “region_seg_image”. If the judgment of the instance of the goal concept is below an application dependent threshold, the control starts in both cases again with the sub-goal “region_seg_image”.

6.4: Experimental Results for scene exploration

So far experiments have been performed for the part of the knowledge base which represents the knowledge about the scene (“office_scene”) and the high-level camera actions (“direct_search” and “indirect_search”). The part of the knowledge base (Figure 5) which contains the concepts “office_image”, “region_seg_image”, “subimage_seg”, “subimage”, “zoom_on_region”, and “explore_office_seg” is provided in one module. In this module hypotheses for the red objects are computed by a histogram backprojection [44] which is applied to an overview image taken with the minimal focal length of the camera. In order to verify these hypotheses, they are fovealized by moving the camera and varying the camera’s focal length. This is exactly the task of the lower part of the knowledge base shown in Figure 5. The primitive concept in the experimental set-up is the concept “color_region”. Thus, the input of the primitive attribute nodes are color regions which are calculated on the basis of the images with the fovealized objects. In Figure 6 results are presented for this module. One can see two overview images, where for each image the potential position for the sought objects have been determined. Each hypothesis is visualized in a close-up view captured automatically during analysis.

In 20 experiments, seven red objects are used, where three of them are modeled in the knowledge base. The positions of the objects differ in each experiment. The three modeled objects which are interesting for the verification step were found in 46 cases of 60 possible ones by the data driven hypotheses generation module using histogram backprojection. On average six close-up views were generated, that is, six object hypotheses were found in each overview image. In the close-up views between 54 and 152 color regions were segmented. In order to reduce the search space for the iterative control algorithm, restrictions concerning the color of the objects are exploited, i.e., only the red color regions were used as input of the primitive attributes. The restrictions which belong to the concepts “punch”, “gluestick” and “adhesive_tape” were propagated once from the higher concepts to the primitive concepts at the beginning of analysis. In Table 2 the results are shown for the 20 experiments. The recognition rates give the ratio between the number of correctly recognized objects and the total number of tested object hypotheses, using N_p processors and performing N_n iterations.

The parallelization on control level ($N_\mathrm{p} = 1, \dots, 4$) was simulated on a monoprocessor system,

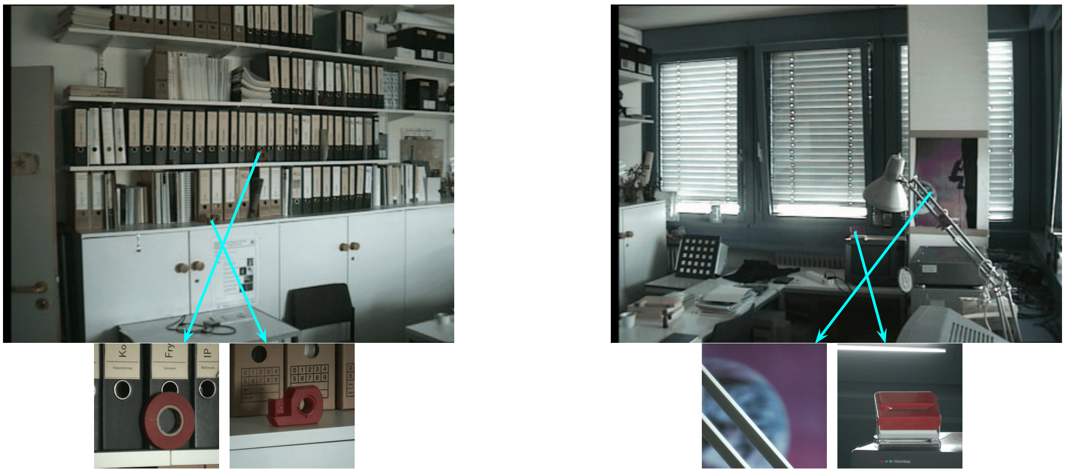


Figure 6. Results for the data driven determination of hypotheses for object positions in an overview image

	Office			
N_n	$N_\mathrm{p}=1$	$N_\mathrm{p}=2$	$N_\mathrm{p}=3$	$N_\mathrm{p}=4$
25	19.5	34.7	41.3	50.0
50	36.9	43.4	52.1	56.5
100	56.5	63.3	60.8	69.6
150	65.2	69.6	71.7	71.7
200	71.7	71.7	71.7	71.7

Table 2. Percentage of correct recognized objects for N_n iterations and N_p processors

and the computation of the nodes of the attribute network was performed sequentially. One can see that the recognition rate increases with the number of iterations and the number of processors up to a maximum of 71.7%. This is due to the fact that the features which are used to recognize the objects are currently view-point dependent. That is the reason why the punch and the adhesive tape are frequently mixed up.

The increase with the number of iterations shows particularly well the any-time capability of the algorithm. The results demonstrate furthermore that 150 iterations are sufficient to achieve an optimal result for a specific camera setting. The stochastic relaxation was used as optimization method.

The processing cycle for one camera setting for interpretation (i.e., from the data driven hypotheses generation up to the computation of an optimal instance of “explore_office”) lasts around five minutes. It turned out that the time requirement splits into the processing time for histogram backprojection (80 sec), for the color region segmentation (60 sec), for the verification of the objects (2.25 sec) and for moving the camera axes, waiting until the goal position is reached. Therefore, the time requirement for the object verification amounts to less than 1 % of the total processing time.

7: Appearance Based Object Recognition

This section proposes another approach for the object recognition part used in Section 6. The knowledge base in Figure 5 shows that the system needs a lot of knowledge just to instantiate an object. For example, if the punch is the object of interest, we have to instantiate all concepts the punch depends upon, for example "color_region", "subimage_seg" etc. In addition, semantic networks are a means for geometric modeling of objects, i.e. objects are modeled by their constituents. Therefore, first the constituents have to be instantiated before the object of interest can be hypothesized. This can be difficult if, for example, parts are occluded. Hence, for some tasks of a service robot it can be more efficient and reliable to use a *holistic* object detector [17], which replaces the object's subnet in Figure 5. If the robot should be able to answer a question like *Is a punch on the table?* an object detector is sufficient and the system can use the appearance based approach introduced in this section. However, if the robot should be able to carry an object, it needs information about parts of the object like the grip of a punch and a geometric approach is more suitable. In the following we will describe the appearance based approach for object recognition in detail.

Recently, appearance based object recognition systems have regained attention because of their ability to deal with complex shaped objects with arbitrary texture under varying lighting conditions. While segmentation based approaches [7, 45, 31] suffer from difficult model generation and unreliable detection of geometric features [30, 31], these methods are solely based on intensity images without the need for segmentation, neither for model generation nor during the classification stage. In contrast to segmentation which tries to extract only the important information needed for the recognition task, appearance based approaches retain as much information as possible by operating directly on local features of the image. The input is either the unprocessed image vector or the result of a local feature extraction process.

As an example for the latter, [34] uses local Fourier and wavelet transformations for preprocessing to receive local features and approximates statistical density function, serving as object models, during model generation. In contrast to this, in [29] unprocessed intensity images undergo a principle component analysis to form an object specific eigenspace in which the object itself is represented as a parametric subspace.

This work extends the latter approach with respect to robust object recognition in the presence of noise and occlusion based on [2, 3, 22]. The new idea of incorporating additional knowledge about the objects, like color or texture, is also explained.

7.1: Eigenspace Approaches for Appearance Based Object Recognition

Object recognition is performed by assigning the image of an object, represented by an intensity matrix $\boldsymbol{I} = [f_{k,l}] (1 \leq k \leq N, 1 \leq l \leq M)$, to a class number $\kappa, 1 \leq \kappa \leq K$ when K object classes Ω_κ exist. In the case of eigenspace approaches, the correct mapping between an image $\boldsymbol{I}$ and a class number κ is learned by means of a principle component analysis of different object views. During a training step, a mapping $\boldsymbol{\Phi}_\kappa$ from P training images ${}^{i}\boldsymbol{I}_\kappa, 1 \leq i \leq P$ which are known to belong to the class Ω_κ, onto low–dimensional feature vectors ${}^{i}\boldsymbol{c}_\kappa$ is learned. The vectors ${}^{i}\boldsymbol{c}_\kappa$ form the object model $\mathcal{M}_\kappa$ which can be further improved by approximating with parametric surfaces. For pose estimation the ground truth pose parameters ${}^{i}\boldsymbol{r}_\kappa$ of the training images are stored together with the feature vectors.

The recognition task is performed by mapping the image $\boldsymbol{I}'$ onto each $\boldsymbol{c}'_\kappa, 1 \leq \kappa \leq K$ and by choosing that Ω_κ for which the distance $d(\boldsymbol{c}'_\kappa, \mathcal{M}_\kappa)$ is minimal. The pose is calculated according to associated parameter vectors of the nearest model vectors.

For the training, i.e. the calculation of $\boldsymbol{\Phi}_\kappa$, a training set consisting of the P images is used for constructing image vectors ${}^i\boldsymbol{f}_\kappa \in \boldsymbol{R}^m$ with $m = N \cdot M$ by concatenating the rows of the training image ${}^i\boldsymbol{I}_\kappa$ and interpreting the result as column vector. Using the *Karhunen–Loeve–Transform*, a set of the first n eigenvectors $\boldsymbol{\varphi}_\nu, 1 \leq \nu \leq n$ according to the largest eigenvalues of

$$\boldsymbol{Q}_\kappa = \boldsymbol{F}_\kappa \boldsymbol{F}_\kappa{}^T \tag{5}$$

with

$$\begin{aligned} \boldsymbol{F}_\kappa &= ({}^1\boldsymbol{f}_\kappa - \boldsymbol{f}_{\kappa,\mu}, \dots, {}^P\boldsymbol{f}_\kappa - \boldsymbol{f}_{\kappa,\mu}) \qquad \text{and} \\ \boldsymbol{f}_{\kappa,\mu} &= \frac{1}{P}\sum_{i=1}^{P} {}^i\boldsymbol{f}_\kappa \end{aligned}$$

from which $\boldsymbol{\Phi}_\kappa = (\boldsymbol{\varphi}_{\kappa 1}, \dots, \boldsymbol{\varphi}_{\kappa n})^T$ is composed.

As $\boldsymbol{Q}_\kappa \in \boldsymbol{R}^{m \times m}$ is usually very large, the eigenvector calculation is done by regarding the *implicit matrix*

$$\widehat{\boldsymbol{Q}}_\kappa = \boldsymbol{F}_\kappa{}^T \boldsymbol{F}_\kappa, \qquad \widehat{\boldsymbol{Q}}_\kappa \in \boldsymbol{R}^{P \times P}. \tag{6}$$

It is shown in [28] that there exists a simple relationship between the eigenvalues and eigenvectors of $\boldsymbol{Q}_\kappa$ and $\widehat{\boldsymbol{Q}}_\kappa$. Assuming that $P \ll m$, the eigenvectors of $\boldsymbol{Q}_\kappa$ can be efficiently computed.

Using the matrix $\boldsymbol{\Phi}_\kappa$, the image vectors ${}^i\boldsymbol{f}_\kappa$ are projected into the eigenspace formed by the row vectors of $\boldsymbol{\Phi}_\kappa$ via

$${}^i\boldsymbol{c}_\kappa = \boldsymbol{\Phi}_\kappa {}^i\boldsymbol{f}_\kappa. \tag{7}$$

From the resulting points ${}^i\boldsymbol{c}_\kappa$ in eigenspace the object model is generated. In [29], for example, parametric curves for interpolating the sparse data are used for this. Figure 7 shows an example of a manifold projected onto the first three eigenvectors. Besides manifolds, other object models like Gaussian densities are possible.

Classification of an image vector $\boldsymbol{f}'$ is then performed according the mapping

$$\zeta : \begin{cases} \boldsymbol{f}' \to 1, \dots, \kappa \\ \zeta(\boldsymbol{f}') = \arg\min_\kappa d(\mathcal{M}_\kappa, \boldsymbol{\Phi}_\kappa, \boldsymbol{f}') \end{cases} \tag{8}$$

with $d(\mathcal{M}_\kappa, \boldsymbol{\Phi}_\kappa, \boldsymbol{f}')$ as the distance of $\boldsymbol{c}'_\kappa = \boldsymbol{\Phi}_\kappa \boldsymbol{f}'$ and the Model $\mathcal{M}_\kappa$.

A rejection class Ω_0 can be introduced by defining a upper bound θ for the accepted distance $d_\kappa = d(c'_\kappa, \mathcal{M}_\kappa, \boldsymbol{f})$. If $d_\kappa > \theta$ holds, then the image vector $\boldsymbol{f}'$ is assigned to Ω_0.

7.2: Robust Classification in the Presence of Clutter and Occlusion

The problem of calculating the feature vector $\boldsymbol{c}'_\kappa$ for an image vector $\boldsymbol{f}'$ via $\boldsymbol{c}'_\kappa = \boldsymbol{\Phi}_\kappa \boldsymbol{f}'$ is, that elements $f'_\iota, 1 \leq \iota \leq m$ belonging to occluded or noisy image parts lead to arbitrary errors [22]. The idea is to reformulate the projection problem so that no longer all elements f'_ι are used but only a subset.

Therefore the pseudo–inverse matrix of $\boldsymbol{\Phi}_\kappa$

$$\boldsymbol{\Phi}_\kappa^+ = \boldsymbol{\Phi}_\kappa^T \left(\boldsymbol{\Phi}_\kappa \boldsymbol{\Phi}_\kappa^T\right)^{-1} \tag{9}$$

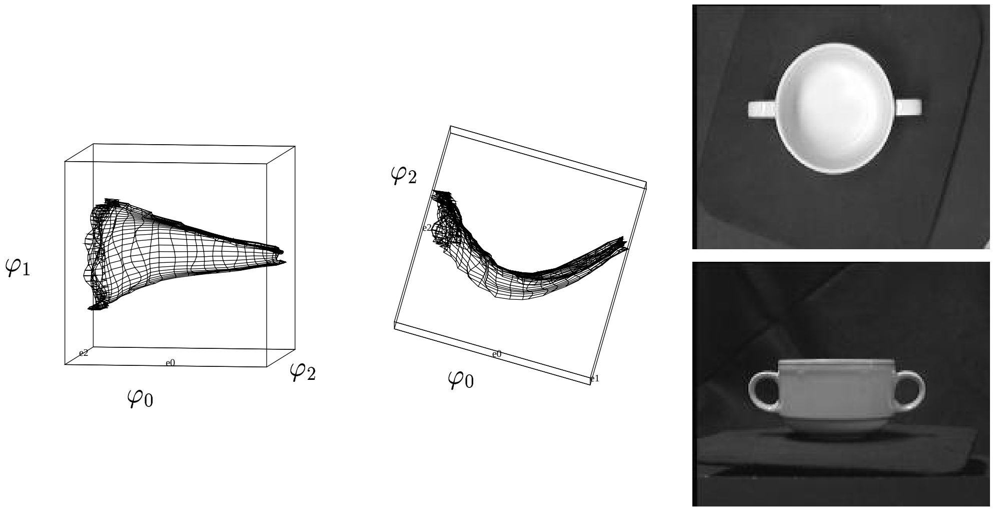


Figure 7. Example of a manifold model with two degrees of freedom generated from views of a cup (two example views on the right side).

is introduced to get

$$\boldsymbol{f}' \quad = \quad \boldsymbol{\Phi}_\kappa^+ \boldsymbol{c}'_\kappa \tag{10}$$

resulting in an equation system of m equations for the n unknowns $c'_{\kappa,1}, \ldots, c'_{\kappa,n}$ of $\boldsymbol{c}'_\kappa$

$$\begin{aligned} f'_1 &= \varphi^+_{\kappa,11} c'_{\kappa,1} + \ldots + \varphi^+_{\kappa,1n} c'_{\kappa,n} \\ &\vdots \\ f'_m &= \varphi^+_{\kappa,m1} c'_{\kappa,1} + \ldots + \varphi^+_{\kappa,mn} c'_{\kappa,n} \end{aligned} \tag{11}$$

with $\boldsymbol{f}' = (f'_1, \ldots, f'_m)^T$ and $\boldsymbol{\Phi}^+_\kappa = [\varphi^+_{\kappa,\sigma\tau}](1 \leq \sigma \leq m, 1 \leq \tau \leq n)$.

Based on the observation that in the absence of interferences it would be sufficient to choose $r_{min} = n$ independent equations out of the m from this equation system to compute a solution for the n components of the feature vector $\boldsymbol{c}'_\kappa$, an approximation $\boldsymbol{c}^*_\kappa$ can be calculated by choosing a set $\mathcal{S} = \{s_1, \ldots, s_r\}$ with $n \leq r \ll m$ and solving

$$\begin{aligned} f'_{s_1} &= \varphi^+_{\kappa,s_1 1} c^*_{\kappa,1} + \ldots + \varphi^+_{\kappa,s_1 n} c^*_{\kappa,n} \\ &\vdots \\ f'_{s_r} &= \varphi^+_{\kappa,s_r 1} c^*_{\kappa,1} + \ldots + \varphi^+_{\kappa,s_r n} c^*_{\kappa,n} \end{aligned} \tag{12}$$

in the least square sense for $\boldsymbol{c}^*_\kappa$ using singular value decomposition (SVD).

The set of chosen equations for $\boldsymbol{f}'_{s_\iota}, s_\iota \in \mathcal{S}$ can be partitioned into $\mathcal{S}_o$, for which $f'_{s_\iota}, s_\iota \in \mathcal{S}_o$ are undisturbed object pixels, and $\mathcal{S}_b$, which represents background pixels and outliers. The approximation for $\boldsymbol{c}^*_\kappa$ according to (12) can only be adequate if $|\mathcal{S}_o| > |\mathcal{S}_b|$ holds. To achieve this [2] suggests to generate a number H of hypotheses ${}^t\mathcal{S}, 1 \leq t \leq H$ for each class Ω_κ by generating the elements ${}^ts_\iota$ on a random basis and to compute

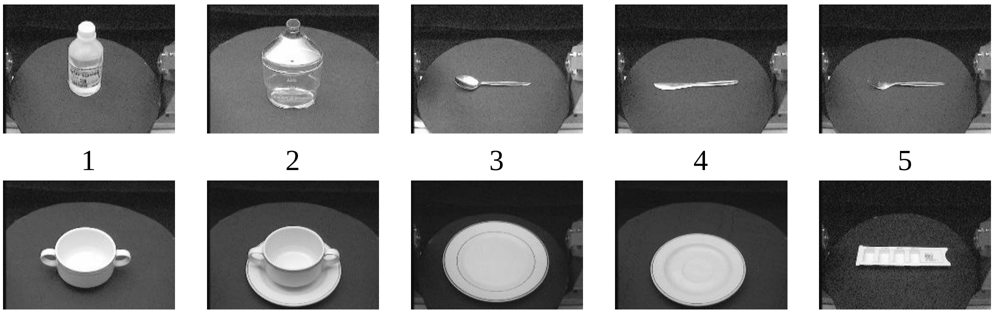
$${}^t\boldsymbol{f}' \quad = \quad {}^t\boldsymbol{\Phi}_\kappa {}^t\boldsymbol{c}^*_\kappa \tag{13}$$

for each hypothesis. An iterative update and selection scheme based on the *minimum description length* leads to the final set $\mathcal{S}_f$ for calculating $\boldsymbol{c}^*_{\kappa f}$. The classification is then performed as described in Section 7.1.

While this selection scheme works fine for compact objects, e.g. those for which the ratio of object to background pixels within the bounding box is considerably high, it fails for oblong objects as the probability of getting a sufficient amount of good object points for the generation of hypotheses is low. By incorporating additional knowledge about object properties the initial selection scheme can be improved if only pixels are regarded as possibly good candidates if object specific conditions like local texture features or color, are fulfilled. Up to know, only the average object intensity is used for restricting the point selection.

7.3: Recognition Results

The methods have been tested on typical objects from hospital environments (DIROKOL sample set, Figure 8 and Table 3). For all experiments the object scale was constant and no 2D–localization within the image plan has been performed. Only pose estimation for the rotational degrees of freedom has been calculated. The varying number of training and test images is due to the different symmetries of the objects, e.g the plates are completely symmetric resulting in only one degree of freedom.



1 2 3 4 5

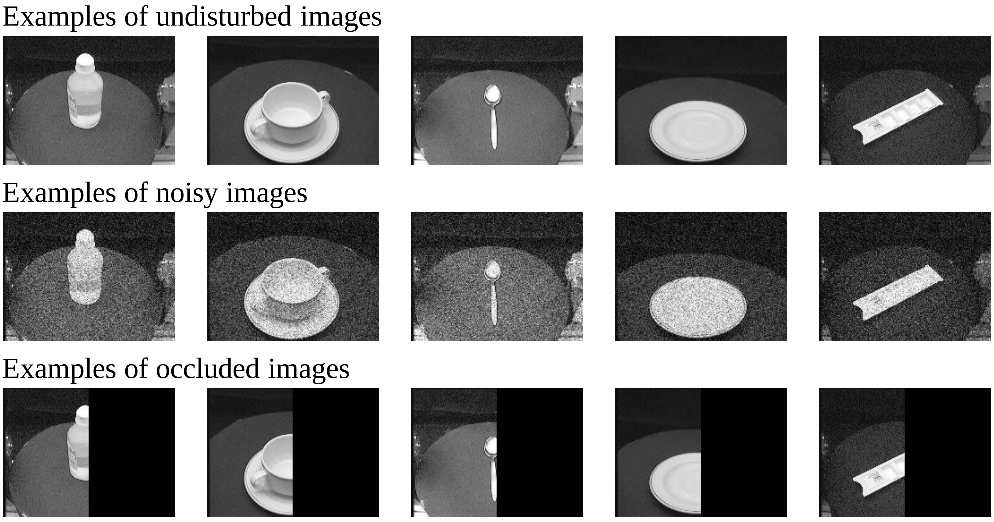
6 7 8 9 10

Figure 8. Objects from the DIROKOL sample set.

Nr. of training images	40–480	
Nr. of test images	2–24	(disjoint to training set)
Nr. of classes	10	
Nr. of eigenvectors used	20	
image sizes	192×144	(grayscale)

Table 3. DIROKOL sample set data

To verify the approach, experiments have been made with fully visible objects in front of a homogenous background at different views. Then the original test images are superimposed by Gaussian noise and lastly, for testing recognition rates in the presence of occlusion, the right half of the test images has been masked out. Figure 9 shows examples of the used test images and Table 4 lists the achieved recognition rates.



Examples of undisturbed images

Examples of noisy images

Examples of occluded images

Figure 9. Examples of images used for testing.

The main problem is the mixing up between objects 3,4 and 5 (numbers according to Figure 8). The second line in Table 4 gives therefore recognition results when object 3,4 and 5 (*spoon*, *knife* and *fork*) are treated as one class (*cutlery*). Table 5 shows the confusion matrix for the occluded test images with the joint class *cutlery*. One can see that the transparent object 2 is difficult to classify but also objects 10 and 6 have high confusion rates.

	undisturbed	Gaussian noise added	50% occlusion
objects 3,4 and 5 belong to different classes	92.5	70.42	58.62
objects 3,4 and 5 are treated as one class	98.44	82.81	70.67

Table 4. Averaged classification results (in %).

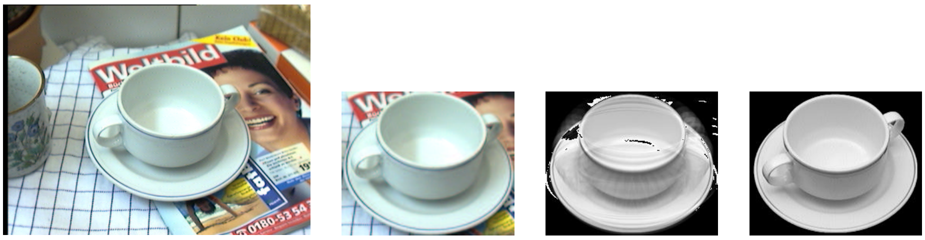
The classification time depends on the number of hypotheses generated per class, the number of points initially used for each hypothesis, and the number of object classes. For measuring computation times, 4 hypotheses have been generated per image, 200 initial points were selected per hypothesis, and 10 object classes.

On a personal computer with a PentiumII processor running at 333 MHz, about 2 seconds are needed for the classification of one test image. Most of the time is spent on calculating the SVD for approximating $\boldsymbol{c}^*_\kappa$ (about 55%). The computation time can be further reduced by doing the calculation in single precision, up to now double precision is used.

As a preliminary result, Figure 10 shows an example of the successful recognition of object 7 with heterogenous background. The bounding box for the subimage has been selected manually, then $\boldsymbol{c}^*_{\kappa f}$ has been calculated for each Ω_κ. The two images on the right side of Figure 10 show the reconstructed images, after the successful assignment to class Ω_7, for $\boldsymbol{c}^*_{7f}$ and for the model point with closest distance to $\boldsymbol{c}^*_{7f}$.

	1	2	3,4,5	6	7	8	9	10
1	24	0	0	0	0	0	0	0
2	1	1	15	2	1	0	4	0
3,4,5	0	1	69	0	0	0	2	0
6	0	2	2	6	0	0	2	0
7	0	0	0	0	12	0	0	0
8	0	0	0	0	0	2	0	0
9	0	0	0	0	0	0	2	0
10	7	0	12	0	0	0	3	4

Table 5. Absolute confusion matrix for occluded test images and joint class *cutlery* **(left column: number of actual class, top row: number of assigned class).**



Scene

extracted subimage

reconstructed projection

reconstruction of nearest model point

Figure 10. Object recognition with heterogenous background.

8: Summary

In this paper we have discussed the use of knowledge based approaches for image and speech understanding in the area of service robots. A typical scenario in the area of service robots has been depicted to motivate two of the most important skills, which must be provided by service robots to improve reliability as well as acceptance of such a technical product. First, a robot must be able to perform an information retrieval dialogue, to collect by means of a dialogue all necessary information for understanding and perform a requested task. Second, the robot needs to actively explore the environment, since usually not all important objects or parts of a scene are visible, given a certain configuration of camera parameters. As a consequence, the robot must actively change the camera parameters in an optimal way to collect missing information. Despite the fact, that a couple of autonomous systems exist, which show sophisticated navigation and localization capabilities — mostly based on sonar or laser sensors — to our knowledge no such system exists which comprises the skills described above.

In our work, we have shown two systems which have similiar skills like the skills a service robot has to possess. Both systems use the semantic network system ERNEST, and an iterative control algorithm, which provides real–time and any–time capabilities, and which can be distributed on a workstation cluster to further improve processing speed. First, an information retrievel dialogue system was introduced. In this application domain of train time table queries, the integration of actions (the system's dialogue steps) into the semantic network formalism, together with the modification of the control algorithm has been presented. On two different test corpora up to 97% of all pragmatic intentions have been analyzed correctly, resulting in a real–time factor of less than one on standard workstations. Second, for active scene exploration an office scene application has been presented. The goal is to find efficiently and reliably objects, by actively changing the search strategy (indirect vs. direct search) as well as the camera parameters (small focal length for overview image, large focal length for close–up views). Again, similar to the speech understanding application, actions (search strategy and change of camera parameters) have been integrated in the semantic network formalism. Also, the iterative control algorithm has been used. In the experiments using a database of three objects we could correctly verify object hypotheses in up to 72 % of all cases. It is worth noting, that initially not all objects are located in the field of view of the camera and that for recognizing certain objects a close–up view is necessary. Both problems are solved by the semantic network approach.

To improve object recognition, which is essential for instantiation of concepts in the semantic network in the office scene scenario, first results of an appearance based approach have been shown. On a test set of objects, which are typically found in hospitals, up to 98 % of the objects could be recognized correctly in the case of an homogeneous background and a data base of 8 to 10 object classes.

In conclusion of this paper we like to emphasize, that at present we have no integrated system, which is able to perform tasks, which are described in Section 2. We argue, that the two examples — one from speech understanding the other from image understanding — are highly correlated to the problems, appearing in the area of service robots, that we can show the advantages of a knowledge based approach using semantic networks, where also actions are integrated in the formalism. Further on, an extension of the existing systems to a system necessary for service robots, should be straight forward. Finally, it is worth noting, that also a fusion of both sources of information, speech and image, in the semantic network formalism is natural, and thus allows an integration of speech and image at the knowledge based level.

References

[1] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 2(3):333–356, 1988.

[2] H. Bischof and A. Leonardis. Robust recovery of eigenimages in the presence of outliers and occlusion. *Internationl Journal of Computing and Information Technology*, 4(1):25–38, 1996.

[3] H. Bischof and A. Leonardis. Robust recognition of scaled eigenimages through a hierarchical approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 664–670, June 1998.

[4] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interacitve museum tour–guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*, Madison, Wisconsin, 1998.

[5] F. Daellert, W. Burgard, D. Fox, and S. Thrun. Using the condensation alorithm for robus, vision–based mobile robot localization. Technical report, Computer Science Dept., Carnegie Mellong University, Pittsburgh, 1998.

[6] W. Eckert, E. Nöth, H Niemann, and E.G. Schukat-Talamazzini. Real Users Behave Weird — Experiences made collecting large Human–Machine–Dialog Corpora. In *Proc. of the ESCA Tutorial and Research Workshop on Spoken Dialogue Systems*, pages 193–196, Vigsø, Denmark, June 1995.

[7] O. Faugeras. *Three–Dimensional Computer Vision – A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[8] J. Fischer and H. Niemann. Applying a parallel any–time control algorithm to a real–world speech understanding problem. In *Proceedings of the 1997 Real World Computing Symposium*, pages 382–389, Tokyo, 1997. Real World Computing Partnership.

[9] V. Fischer. *Parallelverarbeitung in einem semantischen Netzwerk für die wissensbasierte Musteranalyse*, volume 95 of *Dissertationen zur Künstlichen Intelligenz*. Infix, Sankt Augustin, 1995.

[10] V. Fischer and H. Niemann. A parallel any–time control algorithm for image understanding. In *Proceedings of the* 13^{th} *International Conference on Pattern Recognition (ICPR)*, pages A:141–145, Vienna, Austria, October 1996. IEEE Computer Society Press.

[11] Fischer, J. and Niemann, H. and Noeth, E. A Real–Time and Any–Time Approach for a Dialog System. In *Proc. International Workshop Speech and Computer (SPECOM'98)*, pages 85–90, St.–Petersburg, 1998.

[12] G. Geist and V. Sunderam. The pvm system: Supercomputing level concurrent computations on a heterogenous network of workstations. In *Proc. of the 6th IEEE Conference on Distributed Memory Computing*, pages 258–261, Portland, Oreg., 1991.

[13] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[14] D. Goldberg. *Genetic Algorithms: Search, Optimization and Machine Learning*. Addison–Wesley Publ. Co., Reading, Mass., 1989.

[15] R. Graf and P Weckesser. Roomservice in a hotel. In *3rd IFAC Symposium on Intelligent Autonomous Vehicles - IAV 98*, pages 641–647, Madrid, ES, 1998.

[16] A. Hanson and E. Riseman. Visions: A computer system for interpreting scenes. In A. Hanson and E. Riseman, editors, *Computer Vision Systems*, pages 303–333. Academic Press, Inc., New York, 1978.

[17] J. Hornegger, E. Nöth, V. Fischer, and H. Niemann. Semantic network meet Bayesian classifiers. In B. Jähne, P. Geißler, H. Haußecker, and F. Hering, editors, *Mustererkennung 1996*, pages 260–267, Berlin, September 1996. Springer.

[18] J. Fischer and J. Haas and E. Nöth and H. Niemann and F. Deinzer. Empowering Knowledge Based Speech Understanding through Statistics. In *ICSLP*, volume 5, pages 2231–2235, Sydney, Australia, 1998.

[19] F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, London, 1996.

[20] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[21] B. Krebs, B. Korn, and F.M. Wahl. A task driven 3d object recognition system using bayesian networks. In *International Conference on Computer Vision*, pages 527–532, Bombay, India, 1998.

[22] A. Leonardis and H. Bischof. Dealing with occlusion in the eigenspace approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–458, 1996.

[23] T. Levitt, T. Binford, G. Ettinger, and P. Gelband. Probability based control for computer vision. In *Proc. of DARPA Image Understanding Workshop*, pages 355–369, 1989.

[24] M. Mast. *Ein Dialogmodul für ein Spracherkennungs- und Dialogsystem*, volume 50 of *Dissertationen zur Künstlichen Intelligenz*. Infix, Sankt Augustin, 1993.

[25] M. Mast, F. Kummert, U. Ehrlich, G. Fink, T. Kuhn, H. Niemann, and G. Sagerer. A speech understanding and dialog system with a homogeneous linguistic knowledge base. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(2):179–194, 1994.

[26] T. Matsuyama and V. Hwang. *SIGMA. A Knowledge-Based Aerial Image Understanding System*, volume 12 of *Advances in Computer Vision and Machine Intelligence*. Plenum Press, New York and London, 1990.

[27] D. McKeown, W. Harvey, and J. McDermott. Rule-based interpretation of aerial imagery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(5):570–585, 1985.

[28] H. Murase and M. Lindenbaum. Spatial temporal adaptive method for partial eigenstructure decomposition of large images. *IEEE Transactions on Image Processing*, 4(5):620–629, May 1995.

[29] H. Murase and S. Nayar. Visual learning and recognition of 3–d object from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[30] H. Niemann. *Klassifikation von Mustern*. Springer, Berlin, 1983.

[31] H. Niemann. *Pattern Analysis and Understanding*, volume 4 of *Series in Information Sciences*. Springer, Berlin Heidelberg, 1990.

[32] H. Niemann, V. Fischer, D. Paulus, and J. Fischer. Knowledge based image understanding by iterative optimization. In G. Görz and St. Hölldobler, editors, *KI–96: Advances in Artificial Intelligence*, volume 1137 (Lecture Notes in Artificial Intelligence), pages 287–301. Springer, Berlin, 1996.

[33] H. Niemann, G. Sagerer, S. Schröder, and F. Kummert. Ernest: A semantic network system for pattern understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9:883–905, 1990.

[34] J. Pösl. *Erscheinungsbasierte statistische Objekterkennung*. Dissertation, Lehrstuhl für Mustererkennung (Informatik 5), Universität Erlangen–Nürnberg, 1998.

[35] R. Rimey. Control of Selective Perception using Bayes Nets and Decision Theory. Technical report, Department of Computer Science, College of Arts and Science, University of Rochester, Rochester, New York, 1993.

[36] G. Sagerer. *Darstellung und Nutzung von Expertenwissen für ein Bildanalysesystem*. Springer, Berlin, 1985.

[37] G. Sagerer. Automatic Interpretation of Medical Image Sequences. *Pattern Recognition Letters*, 8:87–102, 1988.

[38] G. Sagerer. *Automatisches Verstehen gesprochener Sprache*, volume 74 of *Reihe Informatik*. BI Wissenschaftsverlag, Mannheim, 1990.

[39] G. Sagerer and H. Niemann. *Semantic Networks for Understanding Scenes*. Advances in Computer Vision and Machine Intelligence. Plenum Press, New York and London, 1997.

[40] G. Sagerer, R. Prechtel, and H.-J. Blickle. Ein System zur automatischen Analyse von Sequenzszintigrammen des Herzens. *Der Nuklearmediziner*, 3:137–154, 1990.

[41] R. Salzbrunn. Wissensbasierte Erkennung und Lokalisierung von Objekten. Technical report, Dissertation, Technische Fakultät, Universität Erlangen–Nürnberg, Erlangen, 1995.

[42] S. Schröder, H. Niemann, G. Sagerer, H. Brünig, and R. Salzbrunn. A Knowledege Based Vision System for Industrial Applications. In R. Mohr, T. Pavlidis, and A. Sanfeliu, editors, *Structural Pattern Analysis*, volume 19 of *Series in Computer Science*, pages 95–111, Singapore, 1990. World Scientific Publishing.

[43] G. Socher, G.A. Fink, F. Kummert, and G. Sagerer. Talking about 3D Scenes: Integration of Image and Speech Understanding in a Hybrid Distributed System. In *Proc. Int. Conf. on Image Processing*, pages 809–812, Lausanne, 1996.

[44] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.

[45] F. C. D. Tsai. Using line invariants for object recognition by geometric hashing. Technical report, Courant Institute of Mathematical Sciences, New York, February 1993.

[46] A. Winzen. *Automatische Erzeugung dreidimensionaler Modelle für Bildanalysesysteme*, volume 89 of *Dissertationen zur künstlichen Intelligenz*. Infix, St. Augustin, 1994.

[47] L. Wixson. Gaze Selection for Visual Search. Technical report, Department of Computer Science, College of Arts and Science, University of Rochester, Rochester, New York, 1994.