Integrating Aspects of Active Vision into a Knowledge-Based System

U. Ahlrichs, D. Paulus, H. Niemann
Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen-Nürnberg, Martensstraße 3,
91058 Erlangen, Germany
ahlrichs@informatik.uni-erlangen.de

This paper is published at icpr'2000
September 2000, Barcelona, Spain

Contents

U. Ahlrichs, D. Paulus, H. Niemann: *Integrating Aspects of Active Vision into a Knowledge-Based System*,

Integrating Aspects of Active Vision into a Knowledge-Based System

U. Ahlrichs, D. Paulus, H. Niemann
Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen-Nürnberg, Martensstraße 3, 91058 Erlangen, Germany
ahlrichs@informatik.uni-erlangen.de

Abstract

While the strategy of active vision is well established in early vision, it is not widespread in high-level vision. In this paper we suggest an approach for integrating aspects of active vision into a knowledge-based system. One aspect is the selection of optimal *camera actions which are chosen to make the recognition process more reliable and efficient. We integrated such camera actions into our knowledge base. In addition, we describe the extensions of the control algorithm which is needed to use the information represented in the knowledge base, closing the loop between acting and sensing. Experiments show the efficiency and flexibility of the system. As an example, the task of locating objects in an office room is evaluated.*

1 Introduction

Active perception [1, 2] which has become more and more popular during the last years, deals with modeling and control strategies for perception [2]. In contrast to the Marr paradigm, a camera controls the image acquisition process as an *active* observer to get *optimal* images concerning subsequent image processing steps. This includes, for example, the adjustment of zoom if the image contains objects which cannot reliably be recognized in wide-angle images. In addition, modeling of sensors and the environment including the involved objects is essential. We use semantic networks for knowledge representation. In order to integrate the ideas of active perception, not only the information about objects is required in the knowledge base, but also the knowledge about the adjustment of camera parameters.

In order to use a-priori knowledge represented in the knowledge base during the data-interpretation process, control strategies are needed, which include the control of the interaction between the individual modules like image acquisition and object recognition. Furthermore, a feedback between modules has to be performed by the control algorithm. Strategies for decision making are also needed to guide the data interpretation; we use utility-based judgments for decision making where the functions for the computation of the judgments are integrated into the knowledge base. The control algorithm which is based on an A*-search uses these judgments to select an appropriate camera action depending on the state of the data interpretation process.

In classical image analysis, of course, many systems like SIGMA [6] are known which use information represented in a knowledge base. None of these systems include an active camera control component. Related work to our system can be found, for example, in [5, 8]. A review concerning selective perception can be found in [3].

The knowledge base for the application domain is introduced next (section 2). Afterwards, we outline the control algorithm (section 3). Finally, we demonstrate the feasibility and efficiency of our approach by experiments with a system for exploration of office scenes. (section 4).

2 Knowledge Base

The application domain chosen here is the exploration of arbitrary office scenes. Since the main contribution of the paper is the conceptional work regarding the integration of *camera actions*, i.e. the adjustment of camera parameters, into a semantic network and regarding the extensions of the control algorithm, the object-recognition task of the system is simplified in this context: At the moment only red objects are considered, i.e. the task of the system is to find three predefined red objects, a punch, a gluestick, and an adhesive tape dispenser which need not be visible in an image taken with the initial camera set-up. The 2-d object models used at the moment can easily be substituted by more sophisticated ones in a later stage. Additionally, the knowledge base can be easily extended due to the modularity of the concept-centered representation.

[1] This work was partially funded by the German Research Foundation (DFG) under grant number Ni 191/12-1.

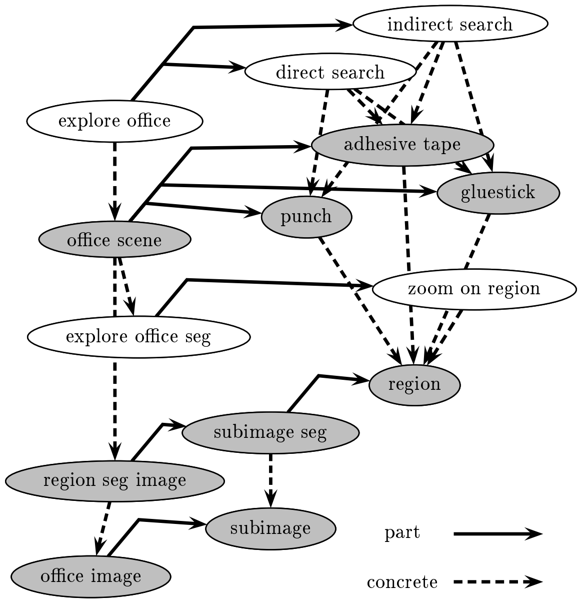

Figure 1. Semantic network for our domain.

2.1 Declarative Knowledge

The structure of the knowledge base for our application domain is shown in Figure 1. We use our semantic network formalism ERNEST for knowledge representation [7]. The knowledge base which was specified manually unifies the representation of objects and their relations and the representation of camera actions on different levels of abstraction. The knowledge base consists of so-called *concepts* which are depicted as ovals in Figure 1. The gray ovals contain, for example, the *objects* of the application domain, e.g. the concepts “punch”, “gluestick” or “adhesive_tape”. These three concepts are *parts of* the concept “office_scene” and they are connected with the concept “color_region” by a *concrete* link, which was introduced in [7] for relating concepts of different conceptual systems to each other.[1]

Concepts for *camera actions* are also integrated into the knowledge base. On the highest level of abstraction one can find camera actions which are equivalent to search procedures and which are used to find objects in a scene. The first example is the concept “direct_search”. Each computation of this concept calculates a new pan angle for the camera in such a way, that overview images (images captured with a small focal length) are obtained. The second example is the concept “indirect_search”. This concept represents an indirect search [10], i.e. the search for an object using an intermediate object, e.g. in order to find a punch we first find a table and then search for the punch on it.

On the intermediate level of abstraction in Figure 1, the camera action “zoom_on_region” can be found. The effect of this action is the fovealization of regions which are hypotheses for the objects. If an object in a hypothesis is too small to be reliably recognized, i.e. the height and width of the dedicated object cannot be determined reliably (cf. section 2.2), we use the fovealization to get more detailed information about it. We refer to images captured after fovealization as *close-up views*.

The *computation* of a camera action concept leads to the selection of new camera parameters or the *performance* of a camera action. So, only one camera action concept can be computed at once. In order to represent competing camera actions, i.e. actions which cannot be performed at the same time, we make use of *modalities* [7]. Modalities have been introduced to represent different concurrent realizations of a concept, such as a chair with or without arm rests or with varying number of legs. 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 same holds for the concept “office_scene” which contains two modalities, one for “explore_office_seg” and one for “region_seg_image”.

During analysis these ambiguities arising from modalities are resolved and so-called *instances* are computed for each concept. The *instantiation* of a concept includes the computation of its components, i.e. the *attributes* and the *relations*, as well as of its judgment. The judgments indicate the match between image data and a-priori knowledge. Additionally, they specify the *utility* of camera actions (section 2.2). Based on these judgments the camera action which is optimal with respect to the criterion defined by the judgment functions can be selected by the control algorithm.

2.2 Procedural Knowledge

The functions for the computation of the attributes and relations of a concept and the judgment of the corresponding instance build up the *procedural* knowledge of the network which includes the functions for attribute calculation and the judgment functions.

The task of our system - considered from the image processing view - splits into several subtasks. First hypotheses for the object location have to be determined. This is done by histogram-backprojection [9] where histograms of the interesting objects are learned before analysis. Using the resulting hypotheses, subimages can be built on which a color-region segmentation is performed. The subimages are represented by the concept “subimage”, whereas the segmented color regions are an attribute of “subimage_seg”. Usually, the objects in the overview images are too small to be reliably verified. In this case the color region segmen-

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

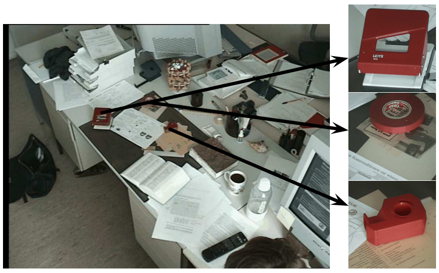

Figure 2. Typical office scene and close-up views for hypotheses.

tation is performed using close-up views which are captured after a camera move such that the optical axis points to the center of the hypothesized area resulting from backprojection. In addition to the region representation, the concept “color_region” contains attributes for the region’s height and width as well as for the region’s color. The objects are recognized by their height and width. Judgments are required to guide the instantiation of concepts and to select the sequence of camera actions.

A management of uncertainty is provided by the control algorithm based on the judgment functions. Probabilities are used to rate the instances of the scene concepts “punch”, “gluestick” and “adhesive_tape”. The judgment of the instance $I(C_k)$ related to the concept C_k subsume the judgments of the concepts’ components $comp^k$. Therefore, the judgment of an instance is defined as $p(I(C_k)|comp^k) = \alpha p(I(C_k)) \prod_{l=1}^{n} p(comp_l^k|I(C_k))$. The constant α denotes the normalization factor. We assume that the individual distributions are pairwise independent and $p(I(C_k))$ is uniformly distributed. In order to rate the individual attributes, parameters of a normal distribution for each attribute are estimated using 40 images for each object. During interpretation values for the attributes are calculated and judged according to the corresponding distribution.

Camera actions are performed in order to provide more information about the scene and reduce the uncertainty of intermediate results. The control algorithm has to decide whether new information is needed and which camera action yields the information with lowest cost. Therefore, *utilities* are used to judge the camera actions [4]. The utility measure relies on the intermediate results of the interpretation, i.e. the evidence if all searched objects have been found. The judgment of an instance which corresponds to an object reflects this information. For each instance we have a hypothesis with states *object found* and *object not found*. Depending on these states the optimal camera action is chosen. The utilities are calculated using a utility table which contains the utility of an action a provided that the hypothesis is in state h, where a belongs to the set of executable actions and h is a state of the random variable H. In general, just the distribution of H is known. Therefore, we can only compute the mean utility $EU(a|e) = \sum_{h \in H} U(a,h)p(h|e)$. The variable e denotes the evidence which arises from the intermediate results of analysis. The control algorithm chooses the action which maximizes the mean utility.

To give an example: Recall the two camera actions “indirect_search” and “direct_search”, which form an action set. We define the vector $\boldsymbol{v} = (I(L), I(A), I(K)) \in \{0,1\}^3$ as hypothesis H. Hence, the states of H are all configurations of this vector which describe if instances of the punch $I(L)$, the adhesive tape $I(A)$ and the gluestick $I(K)$ are available. Within this vector, 1 denotes “object found”, and 0 denotes “object not found”. For example, if $\boldsymbol{v} = (1,1,1)$ the punch, the adhesive tape and the gluestick have been found. At the moment we use 0 and 1 as utilities. For example, if we have found a well-rated instance of the intermediate object, the indirect search is more useful than the direct search. For the action “zoom_on_region” we use a hand-crafted utility function based on the region’s size and the current zoom setting.

3 Using Knowledge - the Control Algorithm

During analysis, the observed image data and the knowledge stored in the semantic network are matched with each other. The task of the control algorithm is to find the best rated instance of the *goal concept* and the *optimal* sequence of camera actions with respect to the criterion defined by the judgment functions. The goal concept corresponds to the goal of the interpretation process which is “explore_office” in Figure 1. Matching is done by expanding the network and instantiating its concepts. During the expansion and instantiation, so-called *search tree nodes* are built which contain the intermediate results of analysis. Competing segmentation results or competing instances, which are modality-dependent, are assigned to competing search tree nodes. For example, one search tree node contains the instance of “direct_search” and another the instance of “indirect_search” depending on the modality of the instance of “expl_office”.

These search tree nodes form the search space of the A*-search algorithm, where the judgment of each node corresponds to the judgment of the goal concept’s instance. Therefore, the judgments of the camera actions which influence the judgment of this instance as explained in section 2.2 are the basis for the control algorithm’s decision which action should be performed.

The structure of the network determines how to calculate instances and propagate restrictions. It does not impose a sequence of instantiations on the network. Specifically, no performance of a closed loop of camera actions and image processing routines is possible directly. Therefore, we apply so-called *local analysis strategies*. They define the se-

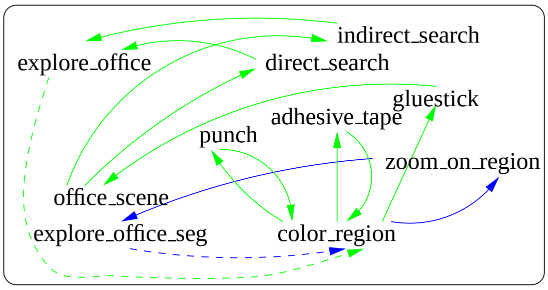

Figure 3. Sequence of concept instantiations

quence of expansions and instantiations within the search tree node and allow for closed loops of acting and sensing. The sequences specify that after the instantiation of a camera action concept a new image has to be taken and analyzed. Additionally, local strategies provide a means to define the direction of analysis, i.e. if the analysis is performed data-driven or model-driven. In Figure 3 the sequence of concept computations is shown for an excerpt of the knowledge base depicted in Figure 1. The control algorithm starts with the concept "color_region" which can be instantiated. Afterwards, depending on the modality of the instance of "office_scene" the objects are instantiated or a new zoom and a new pan value are calculated. During the instantiation of "explore_office_seg", a camera action is performed and therefore, the control algorithm decides to instantiate the concept "color_region" again using the close-up views.

4 Experiments

The knowledge-based system was tested in two different office environments using two similar cameras and actors. The first one, office1, is shown in Figure 2. In order to test the suitability of the whole approach, an active system which corresponds to the knowledge base depicted in Figure 1 was compared to a *passive system*, i.e. a system which does not perform any camera action, but analyses the scene based on the originally provided image data. All the objects which the system had to search for were therefore positioned in the first overview image and neither a direct nor an indirect search was performed. The task of the control algorithm was, besides the search for the best matching regions, to decide if a fovealization would be necessary. 20 experiments were performed in each office with both systems. In each experiment the position of the objects were changed. As the experiments revealed, the active system outperforms the passive system in both office scenes. In office 2 a recognition rate of 80 % was achieved by the active system, in comparison to 66 % using the passive system. The highest recognition rate, 93 %, was achieved in office1 by the active system whereas the passive system achieved 90 %.

In office1 zoom actions were performed if segmentation errors due to reflections occurred. In office2 all objects except the punch were too small to be verified realibly. The judgment function for "zoom_on_region" reflects these observations. However, there are still some problems. If an object hypothesis in office2 which is not fovealized gets a high rating, a zoom action is not performed even if the object is very small. Furthermore, the judgments after performing a camera action are in some cases not optimistic and therefore, the A*-search yields a wrong result. These problems need further research and will be solved in the future.

5 Conclusions

In this article we have proposed an approach for active knowledge-based scene exploration. As the experiments revealed, the approach is suitable to solve the proposed task. In future, we want to integrate more sophisticated object models. Additionally, we will learn local analysis strategies during the exploration of a scene. Instead of the discrete utility values continuous variables will be used.

References

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

[2] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, 1988.

[3] C. Brown. Issues in Selective Perception. In *Proceedings of the International Conference on Pattern Recognition*, volume A, pages 21–30, Los Alamitos, California, 1992. IEEE Computer Society.

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

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

[6] T. Matsuyama and V. Hwang. *SIGMA. A Knowledge-Based Aerial Image Understanding System*, Plenum Press, New York and London, 1990.

[7] 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)*, 12(9):883–905, 1990.

[8] R. Rimey and C. Brown. Task–oriented Vision with Multiple Bayes Nets. In A. Blake and A. Yuille, editors, *Active Vision*, pages 217–236, Cambridge, Massachusetts, 1992.

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

[10] 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.