Handling Uncertainty in 3D Object Recognition Using Bayesian Networks

B. Krebs, M. Burkhardt and B. Korn
email:{B.Krebs, M.Burkhardt, B.Korn}@tu-bs.de

Institute for Robotics and Computer Control, Technical University Braunschweig,
Hamburger Str. 267, D-38114 Braunschweig, F.R.G.

Abstract. In this paper we show how the uncertainty within a 3d recognition process can be modeled using Bayesian nets. Reliable object features in terms of object rims are introduced to allow a robust recognition of industrial free-form objects. Dependencies between observed features and the objects are modeled within the Bayesian net. An algorithm to build the Bayesian net from a set of CAD models is introduced. In the recognition, entering evidence into the Bayesian net reduces the set of possible object hypotheses. Furthermore, the expected change of the joint probability distribution allows an integration of decision reasoning in the Bayesian propagation. The selection of the optimal, next action is incorporated into the Bayesian nets to reduce the uncertainty.

1 Introduction

The task to perform recognition for a 3d scene is a highly complex process which involves various types of sensor processing and interpretation algorithms. A human observer identifies the objects by the knowledge of what he expects to see. If an unknown object occurs the observer may try to find a clue by looking from different directions or may try to detect specific features to solve the ambiguity. Thus, the cognitive process of recognition should be described rather as a task-triangle than a simple bottom-up or top-down process (Figure 1). Based on the

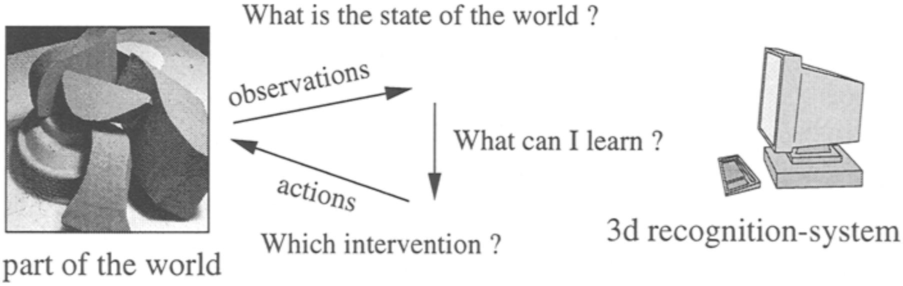

Fig. 1. The task-triangle which has to be modeled in a recognition system.

observations of the world (from sensor data) an interpretation of the scene is established. Since the sensor information may be incomplete or misleading this interpretation has to be considered as uncertain. The observer decides on an action to obtain a better interpretation of the scene. As a result of the observer's intervention the knowledge of the state of the world increases.

All available a priori knowledge has to be integrated to make 3d recognition system reliable. Not only reliable object features (what I expect to see) but also the dependencies between the observations of the object feature (what are the expected relations of the observations) have to be specified. Hence, the domain of observations with their uncertainties has to be modeled and classical probability calculus and decision theory has to be used to guarantee a consistent representation. Now, the recognition task becomes a task to minimize the uncertainty in the scene interpretation.

In section 2 we define reliable free-form object features. This features are treated as elemental observations which are modeled with their dependencies to the objects in a Bayesian net (section 3). Since information from a single view may be insufficient to get reliable results an *active recognition system* has to implement the task-triangle in Figure 1. Dependent on the current evidence appropriate actions have to be executed to acquire more information about the scene. Rimey and Brown showed that evidence values from Bayesian nets can also be used for selecting next actions [19]. However, the decision reasoning is still done with explicit *goodness functions*. We show that the whole decision process can be encoded in a Bayesian net incorporating cost-benefit analysis to select the optimal action of a set of admissible actions.

Early approaches to integrate e.g. viewpoint selection into Bayesian nets don't use the general approach for decision reasoning proposed in this paper which allows a more flexible and more powerful design of various decision schemes [5, 6]. Thus, Bayesian nets and decision theoretic techniques provide a sound formal basis for representation and control in a selective perception system.

2 Rim Curves for Object Identification

The definition of reliable features for free-form objects is a major research field in CAD based vision (CBV). Features based on differential properties are very vulnerable with respect to noise. Some authors proposed to model small, local surface portions, e.g. in terms of "splashes" [21]. Other approaches represented surfaces with a tessellated graph (e.g. [7,10]) or with point samples (e.g. [2,4]). Nevertheless, all these algorithm involve complex computations because surface identification is a two dimensional search problem (3 DOF).

Nevertheless, usual industrial objects do not consist of an overall smooth surface but contain rims, edges, and holes. Using these *object rims* the matching problem is reduced to a one dimensional search (1 DOF). A rim is defined as a 3d space curve separating two adjacent surfaces at a discontinuity. The object rims are represented by 3d B-spline curves. A 3d curve can be identified by curvature

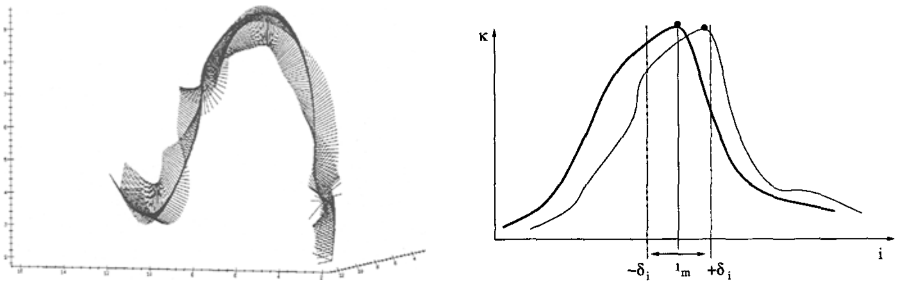

Fig. 2. Matching of subcurves around a curvature extremum.

and torsion values which are invariant with respect to 3d transformations. Rim curves describing surface discontinuities can be efficiently computed from CAD data ([8]) and range images as well ([14]) allowing an robust CAD based vision system. For a 3d space curve at each curve point p_i a Frenet-Frame F_i is defined by

$$F_i = \left[\boldsymbol{t}_i = \frac{\dot{p}_i}{||\dot{p}_i||} \;,\; \boldsymbol{b}_i = \frac{\dot{p}_i \times \ddot{p}_i}{||\dot{p}_i \times \ddot{p}_i||} \;,\; \boldsymbol{m}_i = \boldsymbol{t}_i \times \boldsymbol{b}_i \right] \tag{1}$$

which allows the computation of the curvature $\kappa = \frac{d\alpha}{ds}$ and torsion $\tau = \frac{d\beta}{ds}$:

$$d\alpha_i = arccos \left(\frac{< \boldsymbol{t}_i, \boldsymbol{t}_{i+1} >}{|\boldsymbol{t}_i||\boldsymbol{t}_{i+1}|} \right) \quad , \quad d\beta_i = arccos \left(\frac{< \boldsymbol{b}_i, \boldsymbol{b}_{i+1} >}{|\boldsymbol{b}_i||\boldsymbol{b}_{i+1}|} \right) . \tag{2}$$

Thus, an equidistanly sampled 3d B-spline is uniquely represented by two feature sequence:

$$f_{s_\kappa} = [d\alpha_0, \ldots, d\alpha_{n_s}] \quad , \quad f_{s_\tau} = [d\beta_0, \ldots, d\beta_{n_s}]. \tag{3}$$

The *similarity* between two feature sequences f_s and f_c is defined by

$$s(i,j) = \begin{cases} 1 & (f_s[i] - f_c[i+j])^2 \leq \varepsilon_{max} \\ 0 & else \end{cases} \tag{4}$$

$$s_f(i) = \sum_j s(i,j). \tag{5}$$

Since curvature and torsion are significant for a 3d curve an extracted curve can be identified by the *combined similarity*:

$$s(i) = \frac{\gamma_1 s^\kappa(i) + \gamma_2 s^\tau(i)}{\gamma_1 + \gamma_2} \quad \gamma_1, \gamma_2 \in [0,1] \tag{6}$$

whereas γ_1, γ_2 determine whether curvature or torsion is more discriminant. In our case curvature is more important because the rim curve lay within the object surfaces, i.e. $\gamma_1 = 0.8, \gamma_2 = 0.2$. The best match is described by the *maximum of similarity*

$$i_{max} = \max_{i \in [0, n_c - n_s]} s(i) \tag{7}$$

with the *matching evidence*

$$e_m = \frac{s(i_{max})}{n_c} \in [0,1] \tag{8}$$

the rim curves are divided into ***subcurves*** to guarantee locality (*Local Feature Focus*) and to handle different curve length. Subcurves are selected around a curvature extremum. This allows a simple matching by only computing the similarity around the maxima at the index i_m (Figure 2), i.e. the index j for computing the subcurve's similarity is only taken from the interval $j \in [i_m - \delta_i, i_m + \delta_i]$ within the subcurve in equation (4) and (5).

Subcurves have to be distinct to allow a reliable subcurve identification. Therefore, in an off-line preprocessing the set of all subcurves $\acute{S}$ is computed from a given set of CAD objects ([8]). All subcurves which are similar, i.e. $e_m > \varepsilon_1$, are represented by the subcurve with the highest maximum to form the set of ***significant subcurves*** $S = \{S_0, \ldots, S_n\}$. If an extracted subcurve S_c is identified by the matching with $e_c = \max_{S_i \in S} e(S_c, S_i) \geq \varepsilon_m$ then the transformation T_f to map the model subcurve S_i onto the extracted subcurve S_c is computed by a least square minimization [11].

3 Bayesian Nets for 3d Recognition

The uncertainty in the recognition process has two major sources. On the one hand erroneous sensor data leads to an erroneous feature extraction or to noisy feature values. Reasoning with Bayes rules for low level feature computation from sensor data has been successfully applied (e.g. [1,9]).

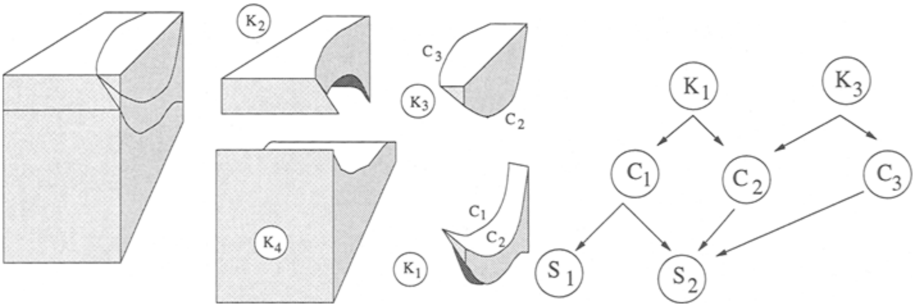

Fig. 3. Dependencies between subcurves, curves and objects define a Bayesian net.

On the other hand uncertainty may result from a misclassification between an object and extracted features or from an inseparability between objects which have similar features. To cope with these problems the statistical behavior of the features has to be modeled within the recognition process. Using *Bayesian nets* not only the statistical behavior of the feature values but also the dependencies between features and objects can be used.

A Bayesian net is a acyclic directed graph (DAG) where the nodes represent a random variable with a fixed set of states and the directed links represent the dependencies between the nodes. For the 3d object recognition example each node represents a hidden cause or an observed feature with the states present (=y) or not present(=n).

In most object recognition systems using Bayesian nets the nodes represent subparts of an object [3,16,19]. However, observable features need not to coincide with object subparts, i.e. features from the same subpart are not statistical independent as it is assumed by constructing objects from subparts. Furthermore, general free-form objects can not be constructed from a simple set of primitives.

Thus, constructing a Bayesian net the dependencies between observations and not between geometrical primitives have to be modeled. Furthermore, the leaf nodes in a Bayesian net have to be independent. The first step is to define reliable object features which can easily be extracted from sensor data and CAD data. In the previous section subcurves of object rims are defined to provide simple and robust features to identify industrial free-form objects. The match of a subcurve with the matching evidence e_m (8) represents an observation and the evidence is entered into the corresponding leaf node of the Bayesian net.

Nevertheless, a major problem using Bayesian nets for recognition is the encoding of the object's locations. Sakar and Boyer proposed the Perceptual Inference Network (PIN) which is a special Bayesian net propagating not only belief values but also position information [20]. But their methods involve highly complex propagation algorithms. Hence, the PIN is reduced to a tree-like structure and is used for reasoning in 2d gray level images only.

We propose to model only relational properties of objects and features in a Bayesian net. The nodes represent subsets of object's hypotheses and the links point from more discriminative to more general subsets.

The Bayesian net can be constructed by "simulating" the possible observations in the CAD data. In a first step for each object in the database a root node, a node with no parents, is introduced in the Bayesian net. In a second step rim curves are extracted from the CAD models [8]. The curve matching from the previous section allows the computation of the similarity between these rim curves. A high matching evidence

$$e_m > \varepsilon_e \tag{9}$$

indicates a dependency between the two curves. The threshold ε_e indicates the ability to separate different types of curves depending on the uncertainty of the sensor data. The value is determined by experiments with real data and in our case is set to $\varepsilon_e = 0.90$. All similar curves are mapped onto a single node. Links between the root nodes and the curve nodes are established if a high matching evidence is found by equation (9).

Subsequently, the curves are divided into subcurves to allow object identification even if only a part of a curve is visible. Only ***significant subcurves*** are mapped onto leaf nodes in the Bayesian net to guarantee the independence of the leaf nodes. Similar subcurves given by equation (9) are represented only by a single subcurve, i.e. are represented by the subcurve with the highest curvature maximum.

If a significant subcurve is similar to a curve by equation (9), i.e. if the significant subcurve $S_i \in C_n$ is similar to the subcurve $S_j \in C_m$, then there is a link between (S_i, C_n) and (S_i, C_m). The nodes representing subcurves are the expected observations. All the other nodes represent the hidden causes containing a set of likely object hypotheses. Therefore, the proposed Bayesian net contains three node layers describing subcurves, curves and objects (Figure 3). The propagation of detected features can be performed without considering the positions of the subcurves.

To complete the network specification we have to specify the prior and the conditional probabilities at the nodes. If prior knowledge of the probabilities of occurrence for the objects in the scenes is available then the priors must represent the expected probability of occurrence. In the absence of any prior knowledge we assume equal prior probabilities:

$$P(O_i = n) = P(O_i = y) = \frac{1}{2} \tag{10}$$

The conditional probability at a node represents the conditional belief that a child node is caused by the parent node, i.e. a set of hypotheses sharing a curve causes the observation of a specific subcurve. The matching evidence (8) determines the ***likelihood***

$$L(C|S) := e_m \tag{11}$$

that the observation of a specific subcurve S was caused by a curve C and the likelihood $L(O|C)$ of the curve C to have been caused by the object O.

The conditional probability table at each node with parents describes the strength of the links to the parents. If evidence is entered into a node N the evidence must be distributed to the parents $pa(N) = P = \{N_0, \ldots, N_l\}$. Hence, the correct conditional probabilities $P(N|P^*)$ for all states in a node N dependent on all combinations of the states in the parent nodes have to be specified. A likelihood value represents only the strength to a single parent. The whole conditional probability table is computed by the "***noisy or***" operation [12, 18].

Each event $C_i = \textit{present} = y$ causes $S = \textit{present} = y$ unless an inhibitor prevents it, and the probability for that is

$$q_i = 1 - L(C_i|S). \tag{12}$$

That is, $P(S = n|C_i = y) = q_i$ assuming all inhibitors to be independent. The conditional probabilities are defined by:

$$P(S = n|C_0, \ldots, C_n) = \prod_{j \in Y} q_j \tag{13}$$

where Y is the set of indices for variables in the state *present* $\Rightarrow$ y [12]. The conditional probabilities $P(C|O^*)$ are computed likewise.

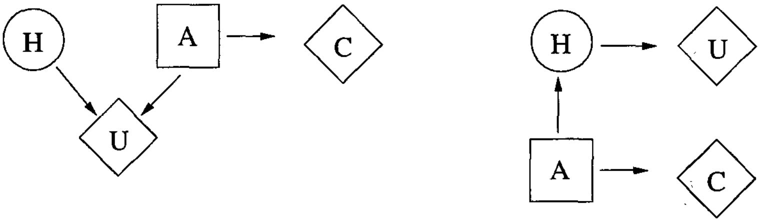

Fig. 4. There are two different types of actions: an action with has no impact on the probability distribution (left) and an action changing the probability of H (right).

4 Decision Reasoning in Uncertainty

To take decisions under uncertainty the global goal is to reduce the uncertainty within the recognition process. The recognition has to decide in a quest for more information which action to take. Let the knowledge be represented in one variable called a hypothesis H with a probability distribution $P(H)$, i.e. given a hypothesis with n_h exclusive states with a probability value each. The driving force for the evaluation of the information value is the variable H, i.e. the decision process is called *hypotheses driven* [12].

Given a set of admissible actions $A = \{a_0, \ldots, a_n\}$ and a *utility table* $U(a, h)$ which describes the utility of each action to reduce the uncertainty if the state $h \in H$ is true. The *value of information* $V(a)$ for each action is the expected

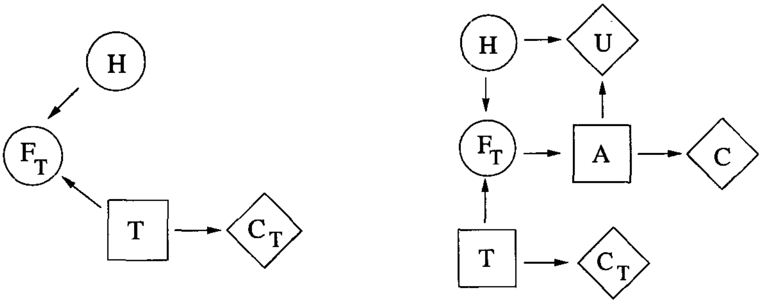

Fig. 5. A test action is an intervening action connected to a node which carries the test outcome as states (left Figure). A test can be easily combined with any arbitrary action (right Figure).

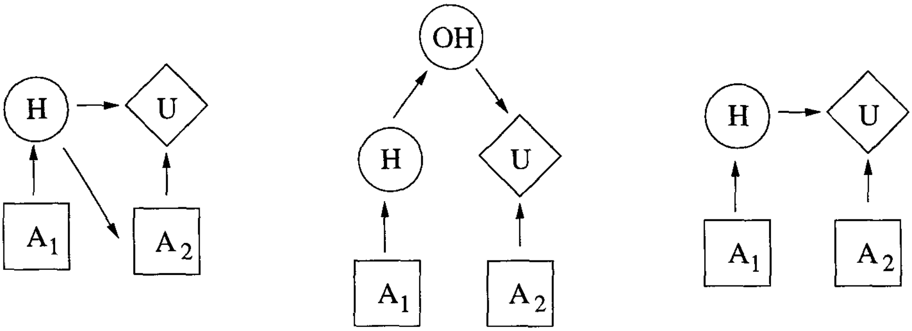

Fig. 6. Decision reasoning with known results (left), uncertain results (middle) and unknown results (right) from action $a_1 \in A_1$.

utility $EU(a)$ of the action:

$$V(a) = EU(a) = \sum_{h \in H} U(a,h)P(h). \tag{14}$$

The *value of information* V for the whole decision process is a function of the distribution of H defined by the *expected utility* EU of performing the *optimal action* $opt(A) \in A$:

$$V(P(H)) = EU(P(H)) = \max_{a \in A} \sum_{h \in H} U(a,h)P(h). \tag{15}$$

The optimal action $opt(A)$ is the argument which maximizes the value of information V:

$$opt(A) = \arg\max V(a). \tag{16}$$

Introducing costs for each action the *expected gain* $g(a)$ of an action is defined by

$$g(a) = \max(V(a) - c(a), 0) \in [0,1] \tag{17}$$

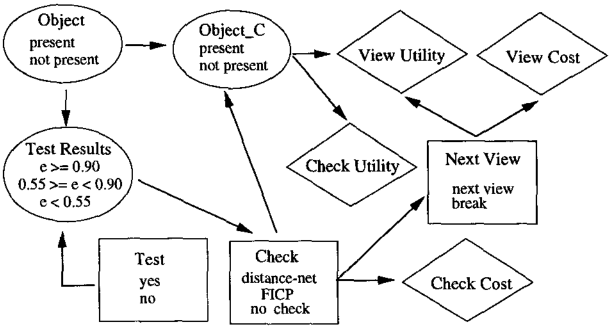

Fig. 7. An example task net to select the optimal action for 3d object recognition.

and the optimal action $opt_g(A)$ maximizes now the expected gain with

$$opt_g(A) = \arg\max g(a). \tag{18}$$

This decision problem can be mapped conveniently on a Bayesian net with a *influence graph* [12]: A two state utility node U is connected with the hypothesis. The decision table is mapped on the conditional probability table of U. The action's costs are mapped likewise onto a utility node C. The set of actions is mapped on a n state decision node A and the cost node C is connected to it. Now, we have only to distinguish between *intervening* and *non-intervening* actions (Figure 4). The execution of a non-intervening action doesn't change the probability distribution in the hypothesis H whereas the execution of a intervening action does, i.e. the decision node A is connected to the utility node U or the hypotheses node H is connected to the decision node A. The *expected utility* of a non-intervening actions is

$$EU(a|e) = \sum_{h \in H} U(a,h)P(h|e). \tag{19}$$

The optimal action for a non-intervening actions can be computed by an Bayesian net (see left Figure 4) with:

$$opt(A) = \arg\max(P(U = y|a,e) + C(a)). \tag{20}$$

Several sets of non-intervening actions can be computed independently and the optimal action is the Cartesian product of the separated optimal actions [12].

The *expected utility* of a intervening actions is

$$EU(a|e) = \sum_{h \in H} U(h)P(h|a,e). \tag{21}$$

The optimal action for an intervening actions can be computed by an Bayesian net (see right Figure 4) with:

$$opt(A) = \arg\max \left[\frac{P(a|U = y, e)}{P(a|e)} + C(a) \right]. \tag{22}$$

Several sets of intervening actions must be solved by simulating each action [12].

Now we have to consider a sequence of actions, e.g. taking a test before executing an action. The outcome of the test has a direct influence of the merit of an action. Furthermore, a test action is an intervening action and may influence the probability of the hypothesis H, i.e. the evidence whether an object is "present" or not. If a test T with cost $c(T)$ yields the outcome t then the value of the new information is

$$V(P(H|t)) = \max_{a \in A} \sum_{h \in H} U(a,h)P(h|t). \tag{23}$$

Since the outcome of T is not known only the *expected information value* $EV(T)$ can be considered:

$$EV(T) = \sum_{t \in T} V(P(H|t))P(t). \tag{24}$$

The expected gain $g(T) \in [0,1]$ performing the test T is

$$g(T) = \max(EV(T) - V(P(H)) - c(T), 0). \tag{25}$$

A test is easily incorporated into a Bayesian net (see left Figure 5). A chance node F_T with a state for each test result is connected to a decision node T containing the test actions. After performing the test evidence for the outcome t is entered into the net. Deciding which action to execute by first taking a test can be easily combined in one Bayesian net and no further computation is necessary (see right Figure 5). Taking a test is only profitable if the decision to select an action is changed through the outcome of the test. Hence, not only the impact of taking test T onto the selection of an action from A but also the decision whether to take the test can be computed by equation (22).

The procedure to select an action after taking a test uses a single look-ahead and is called the *myopic* approach (e.g. [12,19]). Nevertheless, the results from an action my either be uncertain or unknown. Therefore, the look-ahead has to be performed over a sequence of actions to allow non-myopic action selection; e.g. the utility of two subsequent actions may be greater than the utility of each single action. Deciding on a sequence of actions $(a_1 \in A_1, a_2 \in A_2)$ three different cases can be distinguished: the results from action a_1 are known, myopic strategy (Figure 6 left), the results are known but are uncertain (Figure 6 middle) or are unknown, non-myopic strategy (Figure 6 right). This can be implemented in a Bayesian net as well by changing the order of the summation/maximization. The utility for the myopic case $MEU_1(A_2|a_1)$, the uncertain case $MEU_2(A_2|a_1)$ and the non-myopic case $MEU_3(A_2|a_1)$ computes to:

$$MEU_1(A_2|a_1) = \sum_h \max_{a_2} U(a_2, h)P(h|a_1) \tag{26}$$

$$MEU_2(A_2|a_1) = \sum_o \max_{a_2} (\sum_h U(a_2, h)P(o, h|a_1)) \tag{27}$$

$$MEU_3(A_2|a_1) = \max_{a_2} \sum_h U(h, a_1)P(h|a_1) \tag{28}$$

whereby the summation performs a marginalization over all unknows. Since a non-myopic decision reasoning is more uncertain the following equation holds:

$$MEU_3 \leq MEU_2 \leq MEU_1 \tag{29}$$

This theoretical background allows to build highly complex decision schemes. Furthermore, the selection and execution of actions can be easily incorporated within the evidence propagation scheme allowing an interaction of scene interpretation and e.g. sensor actions in the recognition.

5 Recognition Results

The range images are acquired with a sensor based on the coded light approach (CLA) which is a well-known active triangulation method. The Bayesian net was

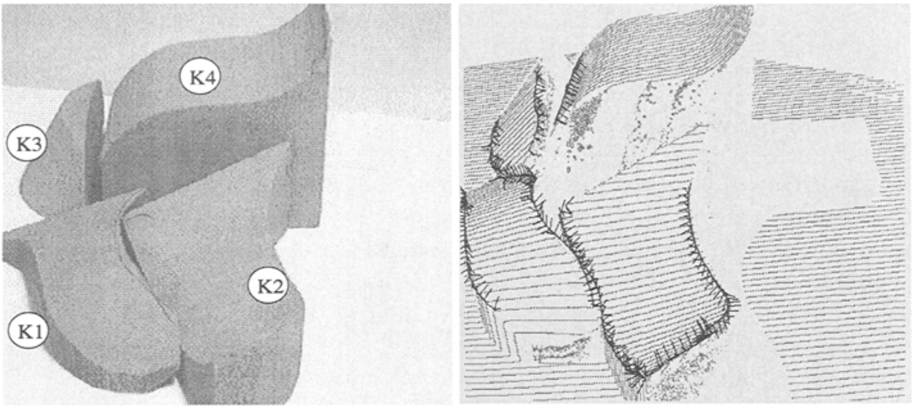

Fig. 8. An example view of a scene with four objects (left) and the extracted 3d spline curves in the corresponding range image (right).

implemented using the HUGIN software system [17]. The proposed algorithms are implemented on a SUN Sparc V. The modelbase of significant subcurves is either extracted from CAD models or learned in a previous range data extraction for each single object.

The decision reasoning is done with a task net depicted in Figure 7. The task net has a sequence of three different action types. First a test is performed which checks whether any reliable evidence for each interpretation (object node in the Bayesian net) has arrived. The thresholds are taken from the matching evidence threshold ε_e in equation (9). If the evidence is greater than ε_e the object is identified and *single object hypotheses* are created which carry a position information. Each single hypothesis is validated by a Fuzzy ICP algorithm [15]. If the evidence lies between 0.55 and ε_e a different Bayesian net encoding distance information is used [13]. After performing the validation actions the system checks whether new information by new range images is necessary.

An recognition example for CAD models is depicted in Figure 8. The propagation results of a single view and after a view sequence are listed in table 1. The propagation results after a single view yield that the objects "K2", "K1" and "K3" have a high evidences because many subcurves are correctly matched. The

Table 1. Results for single and final match of CAD models

object	single e_m	$\#S_i$	matched
P(K1=y)	0.7634	16	5
P(K2=y)	0.8673	24	6
P(K3=y)	0.6735	17	4
P(K4=y)	0.4534	8	1

object	final e_m	$\#S_i$	views
K1	0.76	5 of 16	1
K2	0.86	6 of 24	1
K3	0.88	4 of 17	2
K4	0.76	2 of 8	3

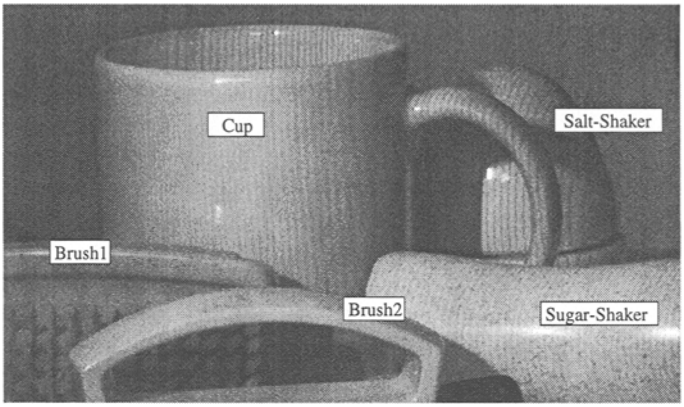

Fig. 9. An example view of every day objects.

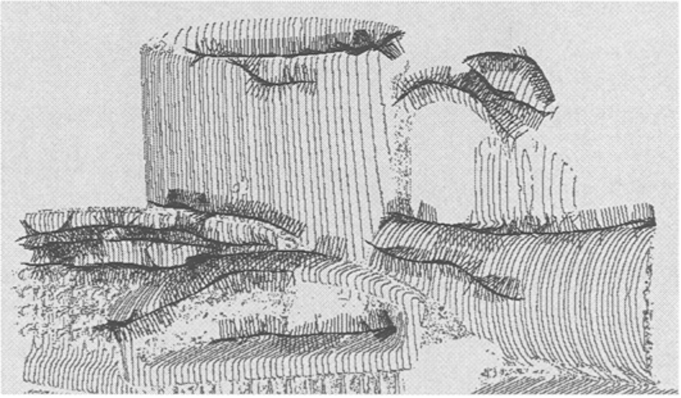

Fig. 10. The extracted 3d spline curves in the corresponding range image.

evidence for object "K4" allows no statement about the finding of the object in the scene because a evidence value about 0.5 is not significant. The sensor data is highly corrupted, thus no correct subcurves are found. After evaluating more views the object "K4" is correctly identified by using the task net from Figure 7 as shown in the table 1.

In Figure 9 a sample of every-day objects is analyzed. The model curves are learned in a previous off-line step for each object. The achieved matching evidence is shown in table 2. Since the edges of the objects for the learned example ar not as crisp as in the CAD example matching evidence is lower. Furthermore, the reflections on the surface disturb the curve extraction. Thus, more views are necessary to identify the objects.

The results show that common objects can be recognized with a database either

Table 2. Matching results for learned subcurves

object	single e_m	$\#S_i$	matched
P(Cup=y)	0.7834	5	2
P(Brushl=y)	0.8073	8	3
P(Brush2=y)	0.54	4	1
P(Sugar-Shaker=y)	0.6	6	1
P(Salt-Shaker=y)	0.57	3	1

object	final e_m	$\#S_i$	views
Cup	0.78	2 of 5	1
Brushl	0.80	6 of 8	1
Brush2	0.88	4 of 4	2
Sugar-Shaker	0.77	2 of 6	5
Salt-Shaker	0.76	2 of 3	7

from CAD descriptions or learned from previous views.

We showed that the separation of position independent from position dependent know-ledge solves a major problem introducing Bayesian nets in recognition systems. Furthermore, selecting appropriate actions depending on the already acquired evidence can be incorporated in the Bayesian nets. The recognition process adapts to the actual information via a task net which minimizes the set of mutual exclusive hypotheses by reducing the uncertainty.

Using influence graphs allows to build highly complex decision schemes even with non-myopic action selection providing a much more flexible decision reasoning within a vision system than any other proposed method (e.g.[5, 6, 19]). Furthermore, the concept of handling uncertainty with Bayesian networks and selecting appropriate actions is quite general and can be easily adapted to other field of recognition and computer vision.

Using features from 3d rim curves provide sufficient information to allow a robust and efficient identification and pose estimation of industrial free-form objects.

References

1. K. L. Boyer, M. j. Mirza, and G. Ganguly. The robust sequential estimator: A general approach and its application to surface organisation in range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):987–1001, 1994.
2. K. Brunnstroem and A. J. Stoddart. Genetic algorithms for free-form surface matching. In *Proc. International Conference on Pattern Recognition, Vienna, Austria*, 1996.
3. D. M. Chelberg. Uncertainty in interpretation of range imagery. In *Proc. International Conference on Computer Vison, Osaka, Japan*, pages 634–657, 1990.
4. C. S. Chua and R. Jarvis. 3d free-form surface registration and object recognition. *Int. J. of Computer Vision*, 17:77–99, 1996.
5. D. Dijan, P. Probet, and P. Rives. Active sensing using bayes nets. In *Proc.International Conference on Advanced Robotics*, pages 895–902, 1995.
6. D. Dijan, P. Rives, and P. Probet. Training bayes nets for model-based recognition. In *Proc.International Conference on Control, Automation, Robotics and Computer Vision, Singapore*, pages 375–379, 1996.
7. C. Dorai and A. K. Jain. Recognition of 3-d free-form objects. In *Proc. International Conference on Pattern Recognition, Vienna, Austria*, 1996.
8. G. E. Farin. *Curves and Surfaces for Computer Aided Geometric Design, a Practical Guide 3rd. ed.* Academic Press, New York, 1993.

9. K. Goldberg and M. Mason. Bayesian grasping. In *Proc. IEEE International Conference on Robotics and Automation, Cincinnati, Ohio*, pages 1264–1269, 1990.
10. K. Higuchi, M. Hebert, and K. Ikeuchi. Bulding 3-d models from unregisterd range images. In *Proc. IEEE International Conference on Robotics and Automation, San Diego, California*, pages 2248–2253, 1994.
11. B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. of America*, 4(4):629–642, 1987.
12. F. V. Jensen. *An Introduction to Bayesian Networks.* UCL Press, 1996.
13. B. Krebs, B. Korn, and M. Burkhardt. A task driven 3d object recognition system using bayesian networks. In *Proc. International Conference on Computer Vison, Bombay, India*, pages 527–532, 1998.
14. B. Krebs, B. Korn, and F.M. Wahl. 3d b-spline curve matching for model based object recognition. In *Proc. International Conference on Image Processing, Santa Barbara, USA*, pages 716–719, 1997.
15. B. Krebs, P. Sieverding, and B. Korn. A fuzzy icp algorithm for 3d free form object recognition. In *Proc. International Conference on Pattern Recognition, Vienna, Austria*, pages 539–543, 1996.
16. W. B. Mann and T. O. Binford. An example of 3d interpretation of images using bayesian networks. In *DARPA Image Understanding Workshop*, pages 793–801, 1992.
17. K. G. Olesen, S. L. Lauritzen, and F. V. Jensen. Hugin: A system creating adaptive causual probabilistic networks. In *Proc. International Conference on Uncertainty in Artificial Intelligence*, pages 223–229, 1992.
18. J. Pearl. *Probababilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1998.
19. R. D. Rimey and C. M. Brown. Control of selective perception using bayes nets and decision theory. *Int. J. of Computer Vision*, 12(2/3):173–208, 1994.
20. S. Sakar and K. L. Boyer. *Computing Perceptual Organization in Computer Vision.* World Scientific, 1994.
21. F. Stein and G.Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.