ROBUST VIDEO-BASED OBJECT RECOGNITION USING CAD MODELS

STEFAN LANSER, OLAF MUNKELT, and CHRISTOPH ZIERL

Technische Universität München, Institut für Informatik – Lehrstuhl Prof. Dr. B. Radig, Orleansstr. 34, D-81667 München, Germany

Abstract. *In this paper we present a robust approach to the recognition of à priori known* 3-D *objects in single* 2-D *video images. In this context recognition includes the identification of objects as well as the determination of their attitude relative to the camera position. The way of modeling objects (appropriately transformed CAD models) increases the flexibility of the approach. The robustness of our system results from the concept of two independent modules for the generation and the refinement of object interpretations. In contrast to our previous research we deliver the assumption of a complete separation of foreground and background. The current implementation supports tasks of a mobile system like grasping workpieces or identifying obstacles.*

Key Words. 3-D object recognition; CAD-based vision; pose estimation; localization.

1 INTRODUCTION

The problem of recognizing objects is studied for a long time in the field of computer vision. Systems have been built e.g. for recognizing targets in aerial images [Bro83], for realizing bin-picking tasks [Ike87], for landmark recognition by an autonomous mobile system [FHR+90], or for sensing the environment under varied conditions like different sensors [SIK92]. These approaches can be classified by the dimension of the used model and image features (2-D/2-D, 3-D/3-D, 3-D/2-D), by the feature type (edge, plane, specularity, combination of primitve features, etc.), and by the basic methods of the recognition process itself (interpretation tree, aspect graph, etc.).

Our approach to object recognition uses either edges or regions of a single 2-D video image and establishes correspondences between these image features and model features following the combination of a generalization of the aspect idea and a modification of Lowe's approach [Low91]. It is composed of the following steps: First image features are extracted. Second correspondences between these features and model features are established. From these correspondences we derive object hypotheses also including a rough estimation for the six degrees of freedom of the attitude of the object. These hypotheses are subsequently verified and refined by a module for pose estimation.

Equation (1) defines the result of this recognition process as an interpretation $\mathcal{I}$ [Gri89, FJ91]:

$$\begin{aligned} \mathcal{I} \;=\; & \langle\ object, \\ & \{(I_{j_1}, M_{i_1}), \ldots, (I_{j_k}, M_{i_k})\}, \\ & (R, t)\ \rangle \end{aligned} \tag{1}$$

with *object* the object hypothesis, (I_{j_l}, M_{i_l}) the correspondence between image feature I_{j_l} and model feature M_{i_l} (assuming n the number of image features, m the number of model features, and $k \leq min(n, m)$), and (R, t) the estimation for the attitude of the object represented by a rotational matrix R and a translational vector t. Two interpretations according to Equation (1) are built during the recognition process. Correspondences are established using aspect trees [Mun94] resulting in the interpretation $\mathcal{I}_1$. This interpretation is the basis for the verification and refinement stage building an improved interpretation $\mathcal{I}_2$. $\mathcal{I}_1$ contains either edges or regions as features, whereas the pose estimation module uses edges exclusively.

Figure 1 gives an overview of the recognition process. In Section 2 the model generation is described. The building of the interpretation $\mathcal{I}_1$ is presented in Section 3. Section 4 shows the refinement of $\mathcal{I}_1$ resulting in $\mathcal{I}_2$. The power and the broad usage of the approach is demonstrated in Section 5. Section 6 summarizes the results.

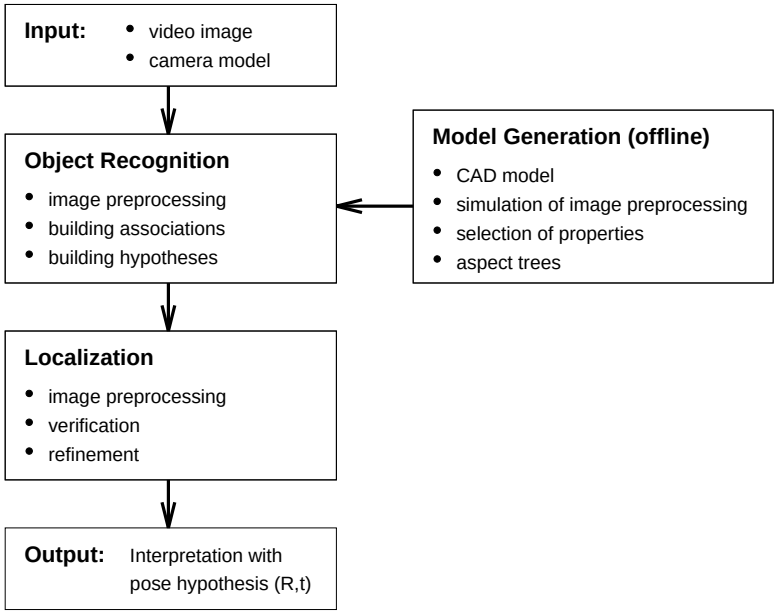

Figure 1 *The modules being part of the recognition system.*

2 MODEL GENERATION

The model generation module computes a set of normalized 2-D views of an object. These normalized views contain the model features used for establishing the correspondences $(I_{j_l}, M_{i_l}) \in \mathcal{I}_1$. The input for this module are CAD models. CAD models are frequently available for objects in manufacturing environments like machines, loadstands, or workpieces. We implemented a postprocessor for the CAD System EUCLID generating a B-Rep following the STEP specification. The used B-Rep contains only planar faces. Objects consist of polygones; spherical faces are approximated by planar faces.

2.1 Simulation of the Image Preprocessing

A triangulated Gaussian Sphere is used to generate 320 2-D views of an 3-D object. The center of gravity of the object is posed in the origin of the sphere triangulated with a given radius [FJ91]. The 2-D views are obtained using an object space approach [SSS74] and the camera model proposed by [LT88]. The view generation preserves the whole 3-D information subsequently needed for the verification and refinement of hypotheses.

Models in CAD systems are highly detailed, especially if they have been built to manufacture the investigated objects. Thus, the 2-D views contain details which certainly cannot be detected during the image preprocessing (due to unsufficient resolution etc.). Therefore, we apply the destructive morphological operation *opening*(.) to reduce the complexity of the model. Furthermore, model derived thresholds for the minimum size of model features are used to eliminate features too small to be detected.

This simulation of the image preprocessing guarantees the comparability of model and image features. In contrast to [IR91, RWVt94] we are not heading for realistic synthetic images. We apply image preprocessing operations both on our models and to the video image in order to reduce the impact of artefacts during the preprocessing. The 2-D views modified by the simulation of the image preprocessing are called *normalized* views. Figure 2 shows some normalized views of a sample object.

3-D model features can be mapped to separated 2-D features due to self-occlusion in a specific view. Naturally, a low level image preprocessing cannot collate the corresponding separated image features. Therefore, these 2-D model features are considered to be independent and the original 3-D model features are split into several appropriate new 3-D model features. As a consequence the number of model features increases while dealing with non-convex objects [Mun94].

2.2 Selection of Properties

Each model feature will be characterized by a set of shape properties. This section describes how to select an "optimal" subset out of a given set of 22 shape properties. An optimal subset has to contain non-redundant information and the values of the included properties should discriminate between different views. We construct a matrix where the rows are formed by the feature vector

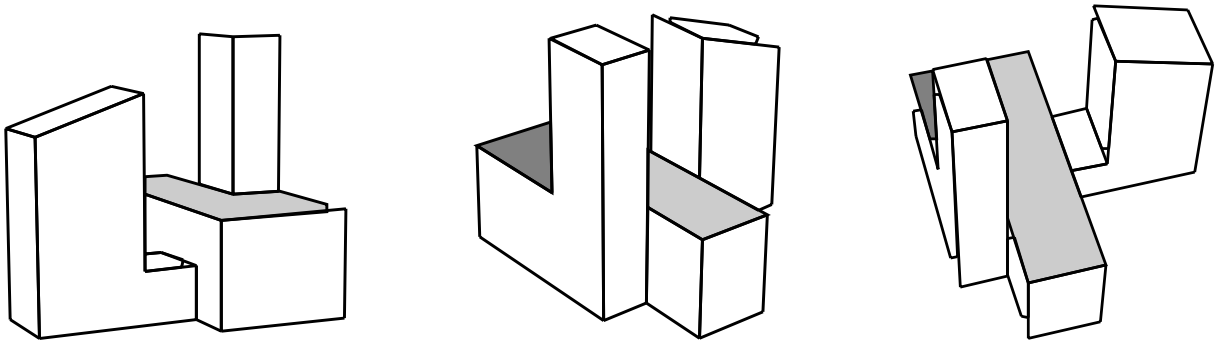

Figure 2 *Three normalized* 2-D *views of the object* BAUHAUS*: The simulation of the image preprocessing smoothes corners and suppresses too small model features. Some model features are split in several new features due to self-occlusion.*

belonging to a specific view. The columns correspond to the normalized properties. Obviously, if two columns are linear dependent they contain redundant information. We compute a symmetric matrix where each element contains a measure for the linear dependence for each two columns. All combinations of subsets of properties in this matrix are evaluated and the combination with the highest rating is the optimal subset of shape properties [Mun93].

The evaluation is modified by applying measures of variation. Properties with slowly changing values in neighbouring views are to be preferred. This is again extended by possible artefacts of the image preprocessing: Small deformations of image features should not change the values of properties significantly. This is in accordance to our definition of an *aspect*, see Subsection 2.3. We use two measures of variation: The first evaluates the variation of the values of properties for each view and its neighbouring views on the triangulatd Gaussian sphere. The second measure of variation is obtained by mapping the values of properties on the views and calculating the distances between the corresponding points of view on the sphere.

2.3 Aspect Trees

Each model feature M_i is now characterized by a set of approx. five shape properties. The values of the properties are grouped into *aspect trees*. This is basically motivated by the aspect idea first proposed by [KvD79], see also [WF90, GCS91, PD86]. There an aspect is defined as a set of topological equivalent views. This definition causes some constraints of what kind of objects are to be recognized in real world images [EBD+93]. Our definition of an aspect is based on the values of properties of model features in order to overcome these drawbacks.

Let $\mathcal{V}_{M_i}$ be the set of all views from which an arbitrary 3-D model feature M_i is visible and p_{ij} a property from the optimal set of properties for M_i. Let m_{i_k} be the 2-D feature corresponding to M_i visible in the view v_k. A set $A \subseteq \mathcal{V}_{M_i}$ is called an *aspect* if and only if the values of the property p_{ij} for m_{i_k} in all views $v_k \in A$ are within a given interval. The generation of aspect trees by hierarchical grouping of these intervals is straightforward.

Aspect trees have been found to be a suitable data structure for establishing initial correspondences between model and image features [Mun94].

3 ESTABLISHING CORRESPONDENCES

In our system an object is represented by a set of normalized views and the generated aspect trees. Of course the model generation described in the previous section is done offline. Based on this representation correspondences between model and image features are established resulting in a first interpretation $\mathcal{I}_1$.

3.1 Image Preprocessing

In order to get regions as image features the video image is segmented applying an edge based color segmentation algorithm using the CIE-$L^*a^*b^*$ color space. This color space is equidistant with respect to the human perception. The segmentation is done in five steps: First the input RGB image is smoothed using a Gaussian filter and transformed into the CIE-$L^*a^*b^*$ color space. A generalized Sobel filter computing the *color gradient* is applied to the transformed image. Color edges are extracted by thresholding the gradient image followed by a contour closure algorithm [WZN93]. The complement of this edge image is separated into connected regions, the raw segments. Finally, small segments are merged with adjacent segments using a region-adjacency-graph if appropriate similarity constraints are fulfilled.

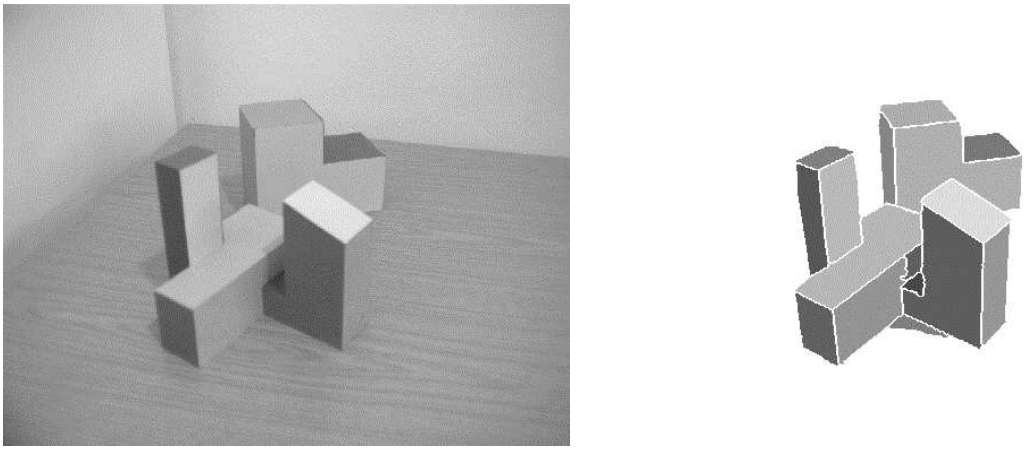

Figure 3 *A video image of the object* BAUHAUS *in front of another object and the result of the color segmentation. Regions touching the image border were removed.*

Edges as image features are extracted applying the modified Deriche filter [LE92].

The extracted image features are the input for the subsequently described computation of an interpretation. A complete separation between foreground and background is not necessary. Image features belonging either to the background or to other objects will be rejected by subsequent modules described later in this section. A preselection of object features based on color information or distance data [MLR+93] is implemented in our system to reduce the complexity of the recognition process. Figure 3 shows an image of a sample object and the result of the color segmentation.

3.2 Building Associations

To establish correspondences between model and image features numerous associations are built. An association is defined as a quadruple (I_j, M_i, v, c_a) where I_j is an image feature, M_i is a model feature, $v \in \mathcal{V}_{M_i}$ is a `2-D` view, and c_a is the confidence of the correspondence between M_i and I_j. The confidence c_a is obtained by traversing the aspect trees belonging to M_i. For each possible combination of a model feature M_i and an image feature I_j the selected shape properties of M_i are computed for I_j and fed into the aspect trees. The traversion of the tree stops in a node n_e if the value of the property is not contained in the intervals attached to the succeeding nodes. Every node in an aspect tree is labeled with a set of views. In each of these views the value of the considered property lies within the interval attached to the node. The confidence of an association (I_j, M_i, v, c_a) depends on the size of the interval labeled to n_e and the number of views corresponding to n_e: the smaller the interval or the number of views the higher the confidence.

In our previous research [Mun94, MZ94, LMZ94] we took into account the relative size and position of the model contour and the contour of the union of all image features assuming a complete separation of foreground and background. This is not assumed any longer and thus, not all image features must belong to the object and the contour cannot be used any longer. Instead we exploit the relative position of model features to each other to measure the plausibility of each association by comparing the topology of model and image features. We thereby reduce the confidence of implausible associations.

3.3 Building Hypotheses

In order to select the "correct" view among the 320 views the associations are used to build hypotheses. For each `2-D` view v_i all associations containing this view are considered. From this set of associations the subset of associations $\mathcal{A}_i$ with the highest rating forming a *topological consistent labeling* of image features is selected. Topological consistency implies both 1:1 correspondences and similar geometric transformations between model and image features.

A hypothesis is defined as a quadruple $(object, \mathcal{A}_i, v_i, c_i)$. The confidence c_i is the average of the confidences of the associations in $\mathcal{A}_i$ with a penalty term for non-mapped features. The use of à priori knowledge suppresses impossible views.

The time complexity for building associations and hypotheses is approx. $O(n \cdot m)$ where n is the number of image features and m is the number of considered model features.

The building of hypotheses results in a list of hypotheses sorted by their confidence values. Note that each hypothesis contains the correspondences between model and image features being part of the interpretation $\mathcal{I}_1$. Figure 4 shows the two high-

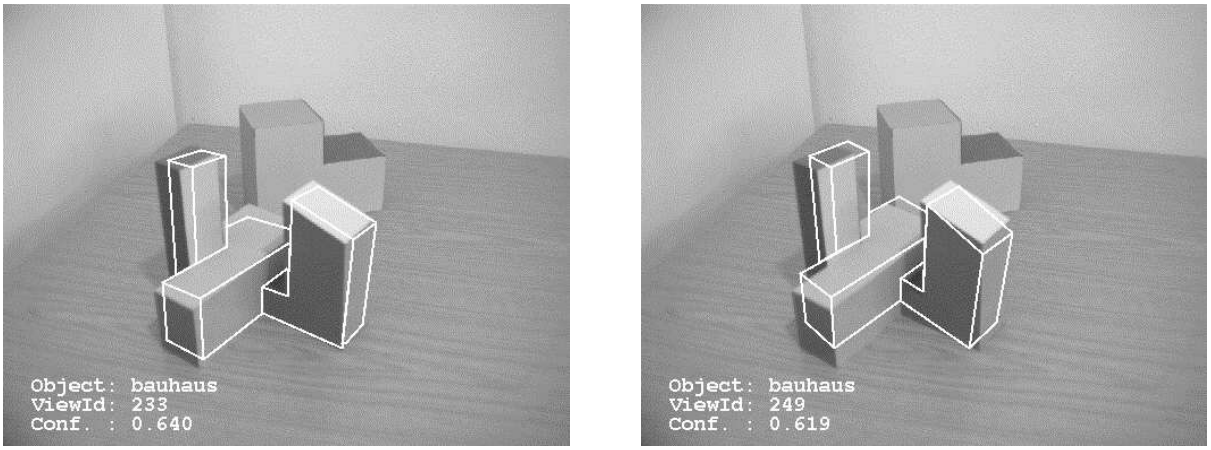

Figure 4 *The two highest rated hypotheses for the object* BAUHAUS *in Fig. 3.*

est rated hypotheses which have been built using the image features from Fig. 3.

3.4 Computing a First Pose Hypothesis

A view in a hypothesis determines one translational and two rotational degrees of freedom of the object pose. Furthermore, while building the hypotheses a scale factor and a translational vector is computed in the image plane. Using this weak perspective `2-D` model lines are computed. A nonlinear optimization (see Section 4) is used to compute a first rough estimation of the six degrees of freedom of the pose hypothesis (R_E, t_E) from the fixed correspondences between `2-D` and `3-D` model lines. The interpretation $\mathcal{I}_1$ is completed by this last step.

4 VERIFICATION AND REFINEMENT

For the verification and refinement of the interpretation $\mathcal{I}_1$ an approach is applied which can be used for the video-based localization of a mobile system, too [RLMR93]. The verification module checks both the object hypothesis *object* and the pose hypothesis (R_E, t_E) in $\mathcal{I}_1$. Interpretations that cannot be verified are rejected. Otherwise the refinement module computes a modified interpretation $\mathcal{I}_2$ containing new correspondences $\{(I_{j_1}, M_{i_1}), \ldots, (I_{j_k}, M_{i_k})\}$ and an improved pose hypothesis (R, t).

4.1 Preprocessing

For the verification and refinement of interpretations correspondences between projected `3-D` model lines and image edges are used. As a first step search spaces in the image are computed based on a given uncertainty of the input pose hypothesis (R_E, t_E). The search spaces are the convex hull of the projections of the `3-D` model lines corresponding to all possible positions of the object. Additionally, the orientations of possibly corresponding image edges are constrained. Edges are detected within the computed search spaces based on the modified Deriche filter [LE92].

The raw edges are split into lines by a split-and-merge algorithm and subsequently replaced by regression lines. This results in a set of possibly corresponding image lines $\mathcal{K}_i$ for each `3-D` model line M_i.

4.2 Verification

The verification of interpretations is done by checking for each model line M_i whether a cluster of possibly corresponding image lines exists in $\mathcal{K}_i$ overlapping the `2-D` projection: A model line is *verified* if and only if the length of the projection of the cluster of image lines on the `2-D` model line exceeds a threshold. In our experiments this threshold is set to 60 % of the length of the `2-D` model line. Note that this kind of verification is robust against some wrong correspondences. A pose hypothesis is verified if the percentage of verified model lines exceeds another threshold (70 % in our experiments). Only verified model lines are considered for the subsequent refinement of the pose hypothesis.

4.3 Refinement

Successfully verified interpretations are refined in the last step of the recognition process. Modified correspondences $\{(I_{j_1}, M_{i_1}), \ldots, (I_{j_k}, M_{i_k})\}$ are established and the input pose hypothesis (R_E, t_E) is corrected to (R, t). For this correction we use a pose estimation algorithm based on the minimization of the distances between projected model lines and corresponding image lines [Low91].

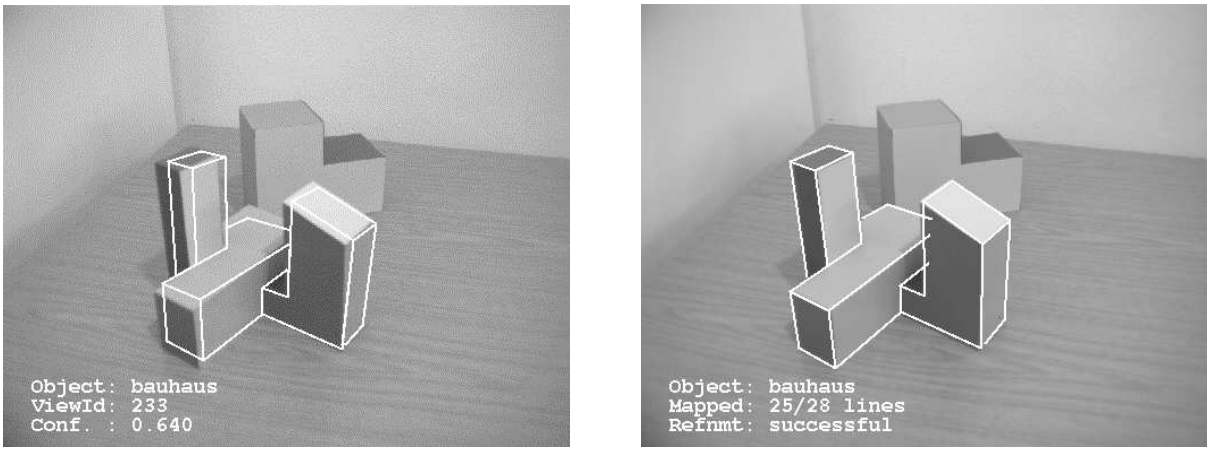

Figure 5 *The result of the refinement of the pose hypothesis for the object* BAUHAUS*: On the left the first interpretation $\mathcal{I}_1$ containing (R_E, t_E) and on the right the improved interpretation $\mathcal{I}_2$ containing (R, t).*

For k correspondences the overall error

$$E_{tot}(R,t) = \|\mathcal{E}\|^2 = \left\| \begin{bmatrix} E_{11} \\ E_{12} \\ \cdots \\ E_{k1} \\ E_{k2} \end{bmatrix} \right\|^2 = \sum_{i=1}^{k} [E_{i1}^2 + E_{i2}^2] \quad (2)$$

is minimized by means of Newton's method. E_{il} is the distance between an end point of the projected model line M_i and the corresponding image line.

A simple combinatorical algorithm establishes correspondences. Model lines are increasingly sorted by the number of possible correspondences, i.e. $|\mathcal{K}_i|$. The first s model lines from this sorted list S_m are used for a first correction of the pose hypothesis (*basic estimation*). The rest of the model lines are used for further refinements. For the s model lines of the basic estimation all combinations of correspondences between model and image lines including the NIL mapping are tested. The value s has been set to 5 in our experiments. The combination with the smallest error E_{tot} is selected. For each NIL mapping a penalty term corresponding to an average distance between an image line and a projected model line is added.

To avoid the combinatorical explosion of tests the sets $\mathcal{K}_i$ are limited using a heuristic preferring long image lines or lines close to the projected model lines. The result of the basic estimation is a first correction of (R_E, t_E).

Subsequent to the basic estimation correspondences for the remaining model lines are established in accordance with the sorted list S_m further improving the pose estimation (R, t). Again all image lines in $\mathcal{K}_i \cup \{NIL\}$ are tested. This procedure is linear in $E(|\mathcal{K}_i|)$ in contrast to the basic estimation. Figure 5 shows the result of the pose refinement.

5 EXPERIMENTS

The approach described in this paper has been tested using several objects within manufacturing environments. Examples for a recognition are presented in Fig. 6 for three of those objects. Lines have been used as model features for generating the interpretation $\mathcal{I}_1$.

Note the good correspondence between the projected model lines and the video image after the refinement step. For the experiments shown in Fig. 5 and 6 the run time as well as the actual number of model and image features are summarized in Table 1. Naturally, the run time depends both on the complexity of the objects (i.e. the number of the model features) and the complexity of the scene (i.e. the number of the image features).

6 SUMMARY

The described approach enables an autonomous mobile system to identify objects like obstacles or workpieces and to compute their exact pose relative to its own position. The separation of the object recognition module from the verification and refinement module increases the robustness of the system. The flexibility of our approach has been demonstrated in different experiments. Further investigations will concentrate on handling occluded objects and increasing the robustness of establishing correspondences by means of topological constraints. Additionally, a quantitative error analysis of the estimated pose has to be done.

7 ACKNOWLEDGEMENT

This piece of work was supported by *Deutsche Forschungsgemeinschaft* within the *Sonderforschungsbereich 331, "Informationsverarbeitung in autonomen, mobilen Handhabungssystemen", Teilprojekt L5.*

8 REFERENCES

[Bro83] R. A. Brooks. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5(2):140–150, March 1983.

[EBD+93] D. Eggert, K.W. Bowyer, C.R. Dyer, H.I. Christensen, and D.B. Goldgof. The Scale Space Aspect Graph. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1114–1129, 1993.

[FHR+90] C. Fennema, A. Hanson, E. Riseman, J. R. Beveridge, and R. Kumar. Model-Directed Mobile Robot Navigation. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(6):1352–1369, November 1990.

[FJ91] P. J. Flynn and A. K. Jain. CAD-Based Computer Vision: From CAD Models to Relational Graphs. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2):114–132, 1991.

[GCS91] Z. Gigus, J. Canny, and R. Seidel. Efficiently Computing and Representing Aspect Graphs of Polyhedral Objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):542–551, June 1991.

[Gri89] W. Eric L. Grimson. On the Recognition of Curved Objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(6):632–643, June 1989.

[Ike87] K. Ikeuchi. Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin-Picking Tasks. *Int. J. Computer Vision*, 1(2):145–165, 1987.

[IR91] K. Ikeuchi and J. C. Robert. Modeling Sensor Detectability with the VANTAGE Geometric/Sensor Modeler. *Transactions on Robotics and Automation*, 7(6):771–784, December 1991.

[KvD79] J. J. Koenderink and A. J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[LE92] S. Lanser and W. Eckstein. A Modification of Deriche's Approach to Edge Detection. In *11th International Conference on Pattern Recognition*, volume III, pages 633–637. IEEE, 1992.

[LMZ94] S. Lanser, O. Munkelt, and C. Zierl. Robuste videobasierte Identifizierung von Hindernissen und Werkstücken sowie die Bestimmung ihrer räumlichen Lage. In P. Levi, editor, *Autonome Mobile Systeme*, pages 95–106. Springer-Verlag, 1994.

[Low91] D. G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[LT88] R. Lenz and R. Tsai. Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machines Metrology. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(5):713–720, 1988.

[MLR+93] O. Munkelt, P. Levi, B. Radig, M. Rožmann, and J. Detlefsen. Integration eines hochauflösenden Radarsensors in ein videobasiertes Objekterkennungssystem. In G. Schmidt, editor, *Autonome Mobile Systeme*. TU München, 1993.

[Mun93] O. Munkelt. Zur Auswahl von Merkmalen. In S.J. Poeppl, editor, *Mustererkennung*, Informatik aktuell, pages 84–93. Deutsche Arbeitsgemeinschaft für Mustererkennung, Springer-Verlag, 1993.

[Mun94] O. Munkelt. Feature Based Aspect-Trees - Generation and Interpretation. In *Second CAD-Based Vision Workshop*, pages 192–201. IEEE Computer Society Press, 1994.

[MZ94] O. Munkelt and C. Zierl. Fast 3-D Object Recognition using Feature Based Aspect-Trees. In *12th International Conference on Pattern Recognition*, pages 854–857. IEEE Computer Society Press, 1994.

[PD86] W. H. Plantinga and C. R. Dyer. An algorithm for constructing the aspect graph. In *Proc. of the 27th Symp. on Foundations of Comp. Science*, pages 123–131. IEEE, 1986.

[RLMR93] A. Ruß, S. Lanser, O. Munkelt, and M. Rožmann. Kontinuierliche Lokalisation mit Video- und Radarsensorik unter Nutzung eines geometrisch-topologischen Umgebungsmodells. In G. Schmidt, editor, *Autonome Mobile Systeme*, pages 313–327. TU München, 1993.

[RWVt94] M. Robey, G. A. W. West, and S. Venkatesh. An Investigation into the Use of Physical Modelling for the Prediction of Various Feature Types Visible from Different View Points. In *Second CAD-Based Vision Workshop*, pages 282–290. IEEE Computer Society Press, 1994.

[SIK92] K. Sato, K. Ikeuchi, and T. Kanade. Model Based Recognition of Specular Objects Using Sensor Models. *CVGIP: Image Understanding*, 55(2):155–169, March 1992.

[SSS74] I. E. Sutherland, R. F. Sproull, and R. A. Schumaker. A Characterization of Ten Hidden Surface Algorithms. *Computing Surveys*, 6(1):2–55, March 1974.

[WF90] R. Wang and H. Freeman. *Machine Vision for Three Dimensional Scenes*, chapter The Use of Characteristic-Views Classes for 3D Object Recognition, pages 109–162. Academic Press, Inc., 1990.

[WZN93] D. Wetzel, A. Zins, and H. Niemann. Edge and Motion Based Segmentation for Traffic Scene Analysis. *Pattern Recognition and Image Analysis*, 3, 1993.

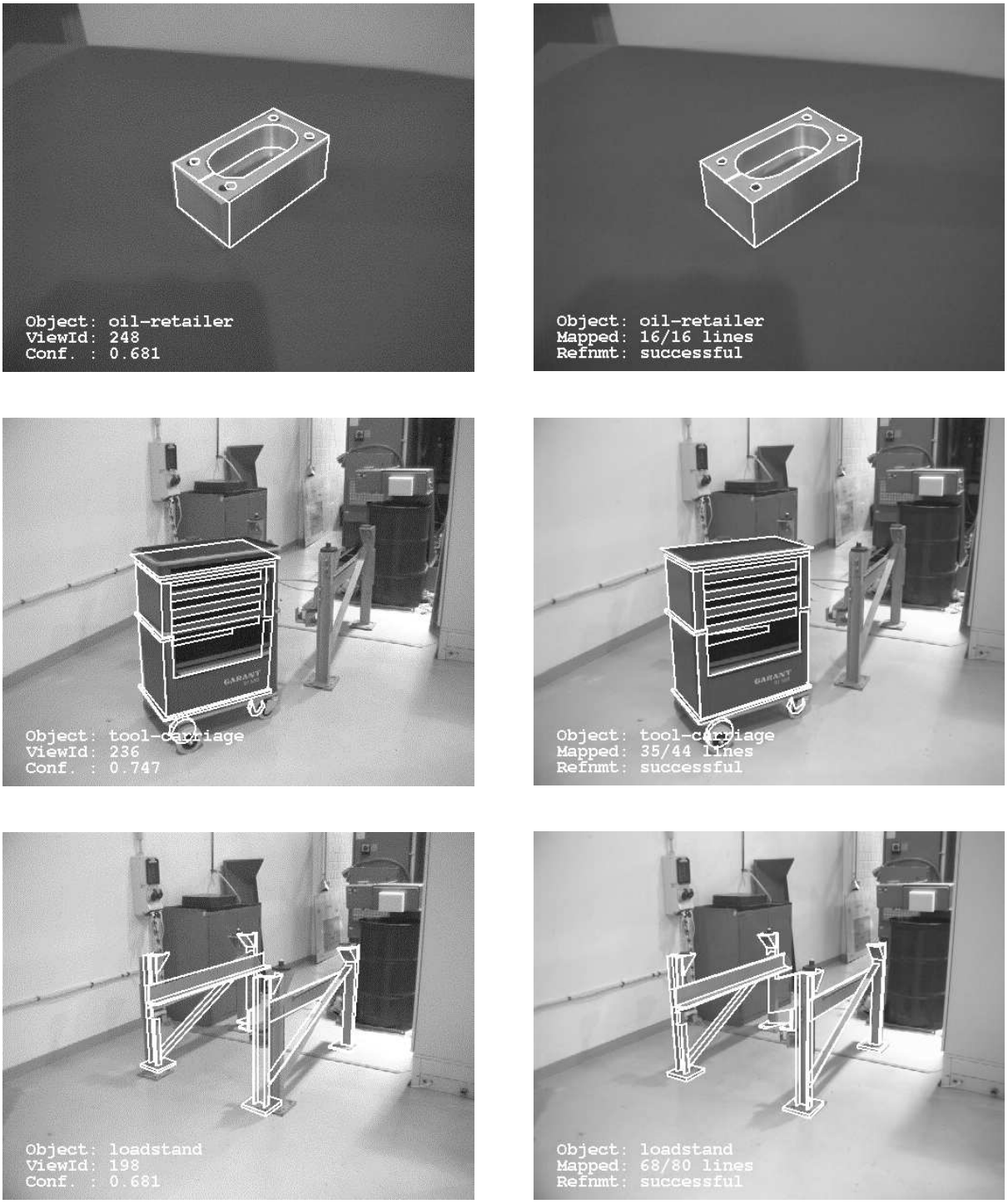

Figure 6 *Experiments using the objects* OIL-RETAILER, TOOL-CARRIAGE, *and* LOADSTAND*: Each row shows the highest rated hypothesis for $\mathcal{I}_1$ and the refined interpretation $\mathcal{I}_2$.*

Object	# Model Features	# Image Features	$\mathcal{I}_1$	$\mathcal{I}_2$
BAUHAUS	27 faces	14 regions	10 + 7 sec.	1 + 1 sec.
OIL-RETAILER	19 lines	16 edges	10 + 2 sec.	1 + 1 sec.
TOOL-CARRIAGE	85 lines	44 edges	16 + 20 sec.	2 + 3 sec.
LOADSTAND	188 lines	41 edges	18 + 35 sec.	2 + 11 sec.

Table 1 *The number of* `3-D` *model features, the actual number of the extracted image features, and the run time on a HP 712/60 for the experiments shown in Fig. 5 and 6. For the generation of $\mathcal{I}_1$ and $\mathcal{I}_2$ values both for the image preprocessing and for the higher level computation are given.*