Hierarchical Recognition of Articulated Objects from Single Perspective Views*

Alexa Hauck
Lehrstuhl für Prozeßrechner
Technische Universität München
hauck@lpr.e-technik.tu-muenchen.de

Stefan Lanser, Christoph Zierl
Forschungsgruppe Bildverstehen (FG BV)
Technische Universität München
{lanser,zierl}@informatik.tu-muenchen.de

Abstract

This paper presents an approach to the recognition of articulated 3D objects in monocular video images. A hierarchical object representation models objects as a composition of rigid components which are explicitly connected by specific kinematic constraints, e.g., rotational and/or translational joints. The recognition task follows this tree-like structure by first estimating the 3D pose of the static component (root) and afterwards determining the relative 3D pose of the remaining components recursively. This method limits the search space for the actual correspondences between image and model features and copes with the problem of self-occlusion. Experiments in the context of autonomous, mobile robots show the practicability of this approach.

1 Introduction

The vision-based recognition of well-structured rigid 3D objects from monocular video images has been widely investigated in the past years leading to a wide range of solutions for specific domains. More recently, research activities have focused on objects with complex surfaces, deformable objects, and the recognition of generic object classes [15]. Without the use of 3D sensor data many of these approaches seem to be limited to even more specific domains, though. This work concentrates on articulated objects, i.e., objects consisting of multiple rigid components connected by joints. The components are approximated by polyhedrals. Despite of these restrictions our approach can handle a large number of man-made objects in real world applications.

In the specific case of the recognition of articulated objects different approaches have been proposed. The naive method is to localize each component separately before determining the inner joint states, e.g. [8]. These approaches neither exploit the kinematic constraints imposed by different joint types nor can they deal with self-occlusion caused by the object components. The more formal solution, the extension of the aspect-graph concept by object configurations [16], leads to an explosion of the number of possible aspects even in simple cases. Global parametric methods like [13, 1] simultaneously estimate the poses of all object components. These approaches suffer from an explosion of the search space for correspondences between image and model features.

Our approach is motivated by the sequential evaluation of joints suggested in [7]: Starting with one component the pose of the connected components is estimated, making use of the kinematic constraints and the already obtained information. In contrast to [7], which uses stereo data, our approach is based on 2D features extracted from a single video image. Furthermore, we introduce a hierarchical model representation, a mechanism to handle self-occlusion, and a more robust method for establishing correspondences incorporating knowledge about the current configuration.

After a brief discussion of different appropriate object models for recognition tasks, Sec. 2 presents a framework for the hierarchical representation of articulated objects. Section 3 introduces the application of this framework to the vision-based recognition of rigid and articulated objects. Additional experiments are shown in Sec. 4, followed by a short conclusion.

*This work was supported by *Deutsche Forschungsgemeinschaft* within the *Sonderforschungsbereich 331, "Informationsverarbeitung in autonomen, mobilen Handhabungssystemen"*, projects L9 and Q5.

2 Hierarchical Object Representation

Most modeling techniques described in literature are specialized on a certain application and therefore are well adapted to specific perception tasks and sensors but cannot be used in a general way. This is especially true for vision–based object recognition systems which heavily depend on the underlying description. In the past years two orthogonal approaches for object representation have evolved (discussed in detail in [6, 14]), both having advantages and disadvantages: *Geometric representations* allow the building of large databases, enable part–based descriptions and therefore can be used for generalized objects and object classes. By predicting views of the assumed object the segmentation process can be assisted in a top–down manner. *Appearance–based representations* on the other hand implicitely take into account surface properties like texture or reflectance, offer easier identification, since the compared data is very similar, but rely heavily on robust segmentation, which is problematic in the case of cluttered scenes or occlusion.

We have developed a hybrid modeling system [4] that combines elements from both approaches by using a geometric description to permit sensor independent abstractions and to enable hierarchical object structures, together with sensor–specific features to integrate information on appearance. In the context of autonomous mobile robots this model serves as the central knowledge base. Sensor data interpretation is facilitated by providing a fast access to the relevant model data in form of predicted sensor views.

2.1 Low-level Representation

Objects influence sensor images in two ways. They can be the source of sensor–specific *features* and they can hide other elements. For a correct sensor view prediction, full geometric and physical models of object and sensor are necessary. In the specific case of a video camera such models are difficult to obtain because various factors as illumination or surface properties have to be taken into account.

As a compromise, an object *obj* can be represented by the tuple $\langle B_{obj}, \mathcal{M}_{obj} \rangle$, where B denotes a polyhedral boundary representation and $\mathcal{M} = \{M_\nu\}$ a set of video-features (up to now 3D line segments). The features are derived in a two-step process: First, 3D line segments which are based on the same set of vertices as the boundaries are generated. Then the model data is compared with a set of images, using the methods described in Sec. 3 to establish correspondences. Only those features that can actually be detected by the sensor are kept in the model, along with a measure of their detectability. The boundary representation can either be derived from a CAD model or be reconstructed by a sensor-based exploration process.

2.2 Aggregated Features

It has been shown that the recognition of objects can be facilitated by establishing correspondences not only between single model and image features, but between groupings of them [12]. This approach reduces the number of correspondence hypotheses to be tested, while at the same time the probability of mismatches decreases with the increasing complexity of the grouping. Therefore, in our appraoch the feature concept has been extended by allowing the definition of *aggregated features*. Aggregated features have the form $\langle \mathcal{M}, \mathcal{C}, R, \mathcal{A} \rangle$, with $\mathcal{M} = \{M_1 \dots M_m\}$ being a set of features, $\mathcal{C} = \{C_1(\mathcal{M}) \dots C_n(\mathcal{M})\}$ a set of constraints that have to be satisfied by the features, and $\mathcal{A}$ a set of freely definable attributes. R denotes a rule with which the visibility of the aggregated feature in a sensor view prediction is determined depending on the visibility of its feature components.

This generic mechanism can be used to create complex feature clusters, but also serves as a powerful tool to define feature templates which express relations between features, as for example the topological relations used by the object recognition process described in Sec. 3.1 [11].

2.3 Object Structure

To enable the modeling of articulated objects, the representation described above has been extended to a tree-like structure which reflects the hierarchic composition of an articulated object. Every node in this tree represents a rigid *component* of the object; therefore, the root represents the static component. An object component c_ν is defined recursively as the tuple

$$\langle {}^{c_\mu}\mathbf{T}_{c_\nu}, j_\nu, B_\nu, \mathcal{M}_\nu, \mathcal{C}_\nu \rangle \qquad (1)$$

with ${}^{c_\mu}\mathbf{T}_{c_\nu}$ being a homogeneous transformation matrix relating the local coordinate system (*frame*) to the one of its parent component c_μ, j_ν describing the joint that links c_ν and c_μ, and $\mathcal{C}_\nu = \{c_{\nu_1} \dots c_{\nu_n}\}$ being the set of (sub-)components that are each linked to c_ν by a separate joint. In the case of the root component c_0 (= *obj*), the parent w is the world reference frame. Therefore, the transform ${}^{w}\mathbf{T}_{obj}$ describes the pose of the object in the world; the joint j_0 can be used to model displacements of objects.

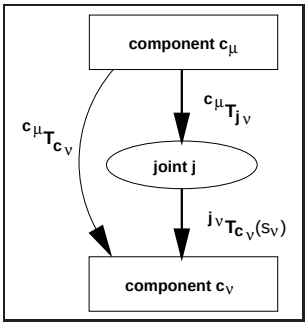

Figure 1. Static and variable part of a joint.

Joints. A joint j_ν can exhibit 6 degrees of freedom (DOF) (3 translational and 3 rotational). Its current state is denoted as $\mathbf{s}_\nu = \langle\, t_x, t_y, t_z, \phi, \theta, \psi \,\rangle$, using an Euler angle representation. The set $\mathcal{J}_{obj}$ of the joint states $\mathbf{s}_\nu$ of all components of an object *obj* is called its *joint configuration*. The transformation matrix ${}^{c_\mu}\mathbf{T}_{c_\nu}$ can be divided into a static part and a variable part:

$$ {}^{c_\mu}\mathbf{T}_{c_\nu} = {}^{c_\mu}\mathbf{T}_{j_\nu} \cdot {}^{j_\nu}\mathbf{T}_{c_\nu}(\mathbf{s}_\nu) $$

with ${}^{c_\mu}\mathbf{T}_{j_\nu}$ describing the (static) transform between the parent frame and the *joint frame*, while ${}^{j_\nu}\mathbf{T}_{c_\nu}(\mathbf{s}_\nu)$ denotes the (variable) transform between the joint frame and the component frame, depending on the current joint state $\mathbf{s}_\nu$, see Fig. 1. Thus, the joint frame coincides with the component frame for the state $\langle\, 0, 0, 0, 0, 0, 0 \,\rangle$.

In the special case that all joints exhibit at most one rotational or translational degree of freedom, the object description can be constructed using a modified Denavit–Hartenberg formalism, which was originally developed to model manipulator kinematics.

Masks. A joint state can be explicitely declared as *unknown*. Unknown joint states have to be specially treated during a sensor view prediction since the position of both the features and the boundary of the components following the joint can vary over a large range. In order to avoid mismatches a conservative approach was taken: if a joint state is unknown, the features of all the components following the joint are ignored. The boundary description is extended by the so-called *masks*, which model the space potentially being occupied by the moving component and its subcomponents. For a sensor view prediction, the masks are treated as additional obstacles; if a feature is hidden by a mask, it will be predicted, but with an additional attribute marking it as *possibly-occluded*.

Object classes. Geometrically identical objects form an *object class*. The instances of a class differ in their individual position ${}^{w}\mathbf{T}_{obj_i}$ and their current joint

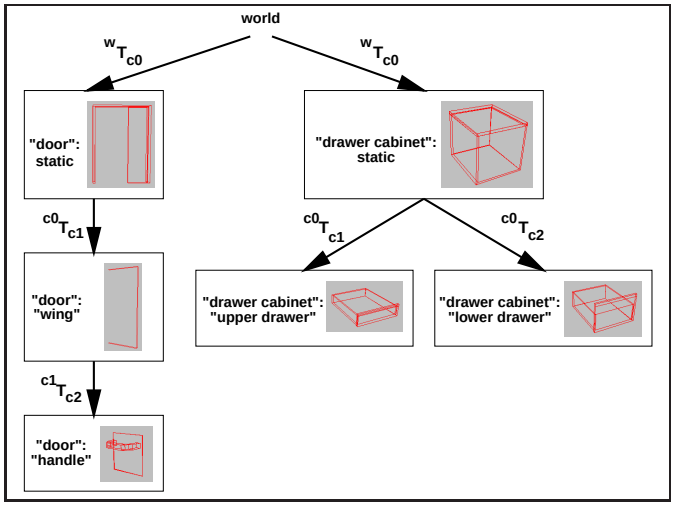

Figure 2. Hierarchical object model of doors and drawer cabinets.

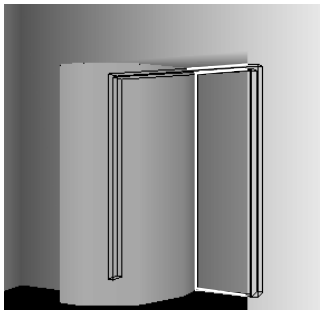

Figure 3. Sensor view prediction of a door with a mask due to the unknown joint state.

configuration $\mathcal{J}_i$. The invariant parts of an object description (i.e., the static transforms ${}^{c_\mu}\mathbf{T}_{j_\nu}$, boundaries, and features) are stored only once for each class.

Figure 2 illustrates the modeling concept at the example of two object classes, a door and a movable drawer cabinet containing two drawers.

2.4 Prediction of Sensor Views

A sensor view prediction consists of projecting the model features into the image plane and testing their visibility against the boundary descriptions. For this, a modified z–buffer algorithm is used, which originally was developed for computer graphics applications [3]. The result is a set of 2D features $\{m_i\}$.

Figure 3 shows a sensor view prediction of a door superimposed on the underlying z–buffer: White lines denote video–specific features that are visible, black lines (possibly) hidden ones. The joint state of the door–wing is unknown, so the corresponding mask is predicted, hiding features of the door–frame.

3 Hierarchical Object Recognition

This section presents our approach to vision-based 3D object recognition using single images captured by a monocular CCD sensor [10]. The basic principle is to establish correspondences between detected image features and appropriate model features provided by the framework described in the previous section.

The recognition of solid objects, i.e., objects without joints or with known joint configuration, can be formalized by an interpretation [2]

$$\langle\, obj, \{(I_{j_1}, M_{i_1}) \dots (I_{j_k}, M_{i_k})\}, {}^{cam}\mathbf{T}_{obj} \,\rangle \qquad (2)$$

with obj the object hypothesis, (I_{j_l}, M_{i_l}) the correspondence between image feature I_{j_l} and model feature M_{i_l} (subsequently called an *association*), and ${}^{cam}\mathbf{T}_{obj}$ the transform describing the estimated 3D pose of the object relative to the camera.

Dealing with articulated objects with (partially) unknown joint configuration, this definition of an interpretation has to be extended to

$$\langle\, obj, \{(c_\nu, \{(I_{j_1}, M^\nu_{i_1}) \dots (I_{j_k}, M^\nu_{i_k})\}, {}^{c_\mu}\mathbf{T}_{c_\nu})\} \,\rangle \qquad (3)$$

with obj the object hypothesis consisting of different object components $\{c_\nu\}$. The set $\{(c_\nu, \{(I_{j_1}, M^\nu_{i_1}), \dots, (I_{j_k}, M^\nu_{i_k})\}, {}^{c_\mu}\mathbf{T}_{c_\nu})\}$ of triples describes the joint state of each object component c_ν relative to its parent c_μ and the underlying correspondences. Note, that ${}^{cam}\mathbf{T}_{c_0}$ corresponds to ${}^{cam}\mathbf{T}_{obj}$ in Eq. (2).

3.1 Objects with Known Joint Configuration

In case there is hardly any knowledge about the current pose of a rigid object, the recognition process is divided into two phases. First, rough pose hypotheses are generated by combining consistent sets of associations. These pose hypotheses serve as a starting point for the subsequent pose verification and refinement.

Characteristic views. The recognition of rigid objects is based on a set of characteristic 2D views (*multiview representation*), called CVs. The determination of the viewpoints defining the CVs is based on the triangulated Gaussian sphere which guarantees an approximately homogeneous distribution of viewpoints around the object [10]. Alternatively, an aspect-based approach to select object-specific viewpoints could be employed [17]. The CVs are predictions of model features (sensor view) provided by the model.

Knowledge about the current scene can be used to further restrict the number of CVs to be considered: For example, if the tilt of the CCD camera relative to the object is approximately known, the viewpoints must lie within a "torus" around the object, see Fig. 4.

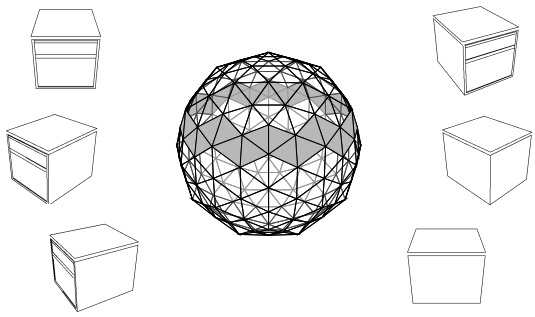

Figure 4. Characteristic views of a rigid 3D object, given an approximately known camera tilt.

Building rough hypotheses. During the recognition process, possible associations between model and image features are generated for each CV. The confidence value for each association is obtained by a geometrical comparison of the features incorporating a measure for the local topological consistency [11]. The underlying topological relations are provided by the geometric model using the mechanism of feature aggregation, see Sec. 2.2.

The associations are grouped to geometrically and topologically consistent hypotheses. This list of hypotheses is sorted by a confidence value, which incorporates the confidence values of the included associations, the percentage of mapped model features, and the global topological support [11]. Once the 2D correspondences are established, the CV v consisting of the 2D features $\{m^v_i\}$ underlying the selected hypothesis is scaled (by factors s_1, s_2) and shifted (by a vector $(t_x\, t_y)^T$) in the image plane to match the image features. Aligning this modified CV with the original 3D model $\{M_i\}$ leads to a rough pose estimate ${}^{cam}\tilde{\mathbf{T}}_{obj}$ by minimizing

$$\sum_{i=1}^{n} \left[\mathrm{proj}({}^{cam}\tilde{\mathbf{T}}_{obj} \cdot M_i) - \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} \cdot m^v_i - \begin{pmatrix} t_x \\ t_y \end{pmatrix} \right]^2 \qquad (4)$$

$$\mathrm{proj}\left((x\, y\, z\, 1)^T \right) := \begin{pmatrix} f\frac{x}{z} \\ f\frac{y}{z} \end{pmatrix} \quad \text{with } f \text{ the focal length}$$

In Eq. (4) the modified CV is used instead of the original image features to reduce the impact of mismatches.

Final pose estimation. The exact 3D pose ${}^{cam}\mathbf{T}_{obj}$ is determined by traversing a dynamically re-arranged interpretion tree. This process combines the pose estimation and the search for final correspondences. The

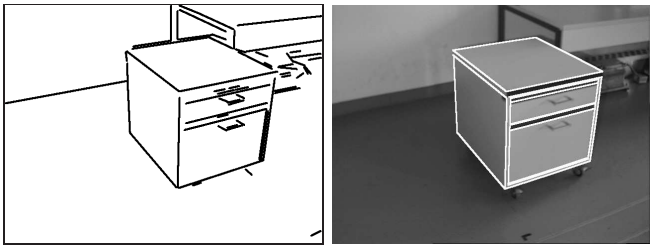

Figure 5. Extracted image line segments and computed 3D pose estimation of a rigid object.

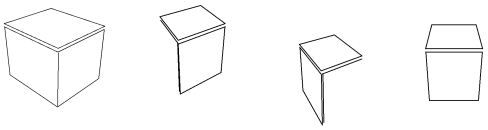

Figure 6. Characteristic views of the the static component of a drawer cabinet with an unknown joint configuration of the two drawers.

traversion is controlled by topological constraints and the *viewpoint consistency* constraint. On each level of the interpretation tree the remaining uncertainty of the current pose estimate restricts the search space for the still unmapped model features [9].

Similar to [13], the pose estimation itself is based on a weighted least-squares technique minimizing

$$\sum_{k=1}^{m} e\left(M_{i_k}, I_{j_k}, {}^{cam}\mathbf{T}_{obj}\right)^2 \qquad (5)$$

using an appropriate error function e, e.g., the distance between the projected endpoints of model lines and the corresponding image lines.

Figure 5 shows the extracted image line segments and the result of the recognition of a drawer cabinet with a priori known joint configuration.

3.2 Objects with Unknown Joint Configuration

For the recognition of articulated objects a recursive process is proposed, which is guided by the hierarchical structure of the object model. First the static object component c_0 is recognized by applying the method described in Sec. 3.1. Potential self-occlusion by yet unrecognized object components is automatically taken into account by the *mask* technique during the computation of the CVs as described in Sec. 2.3. If enough model features are predicted, which are not occluded by masks, *possibly-occluded* features are excluded from the interpretation. Figure 6 shows some exemplary CVs

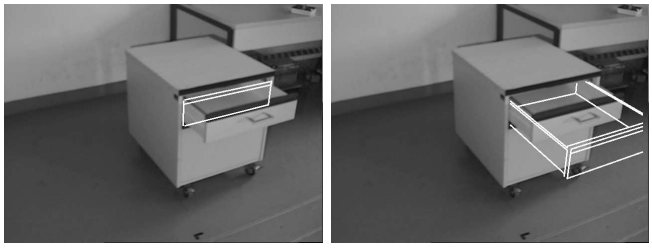

Figure 7. Two of the characteristic views of the upper drawer predicted using the estimated pose of the static object component.

of the static component of the movable drawer cabinet. Here, model features which are marked as *possibly-occluded* are suppressed. Figure 8 (a) shows the result of the recognition of the static component.

For each subsequent c_ν the corresponding joint state ${}^{j_\nu}\mathbf{T}_{c_\nu}(\mathbf{s}_\nu)$ has to be determined following the object hierarchy. This again can be done by applying the method described in Sec. 3.1. Note, that ${}^{cam}\mathbf{T}_{obj}$ in Eq. (5) now has to be replaced by

$$\begin{aligned} {}^{cam}\mathbf{T}_{c_\nu} &= {}^{cam}\mathbf{T}_{obj} \cdot \ldots \cdot {}^{c_\mu}\mathbf{T}_{c_\nu} \\ &= {}^{cam}\mathbf{T}_{c_\mu} \cdot {}^{c_\mu}\mathbf{T}_{j_\nu} \cdot {}^{j_\nu}\mathbf{T}_{c_\nu}(\mathbf{s}_\nu) \\ &= {}^{cam}\mathbf{T}_{j_\nu} \cdot {}^{j_\nu}\mathbf{T}_{c_\nu}(\mathbf{s}_\nu) \end{aligned}$$

with ${}^{cam}\mathbf{T}_{j_\nu}$ depending only on the static transform of j and the already estimated pose of the higher-level components $c_0 \ldots c_\mu$.

CVs of c_ν are predicted by sampling the according subspace of the joint configuration space, see Fig. 7. Depending on the sampling distance, an uncertainty is associated with each viewpoint, which is propagated to specific search spaces for matching candidates among the image features [9]. When dealing with lines as features a appropriate search space is the region of the input image in which a corresponding image line must lie with a given probability. Consider the projection $m^l_{pix_i}$ of a 3D model point M^l_i in a (sub-)pixel of the video image as a random variable with mean $\overline{m^l_{pix_i}}$ and covariance matrix $\Sigma_{m^l_{pix_i}}$ derived from the uncertainty of the camera pose. Then the desired search space S_{M_i} for a model line $M_i = \langle M^1_i, M^2_i \rangle$ can be computed as

$$\begin{aligned} S_{M_i} &= convexhull\left(S_{M^1_i} \cup S_{M^2_i}\right) \\ S_{M^l_i} &= \left\{m^l_{pix_i} \mid mdist\left(m^l_{pix_i}\right) \leq d\right\} \\ mdist\left(x\right) &= \sqrt{\left(x - \overline{x}\right)^T \Sigma_x^{-1} \left(x - \overline{x}\right)} \end{aligned}$$

where the Mahalanobis distance d controls the desired probability, see [9] for details. Fig. 10 (d) shows an example for such a search space. In a similar way a

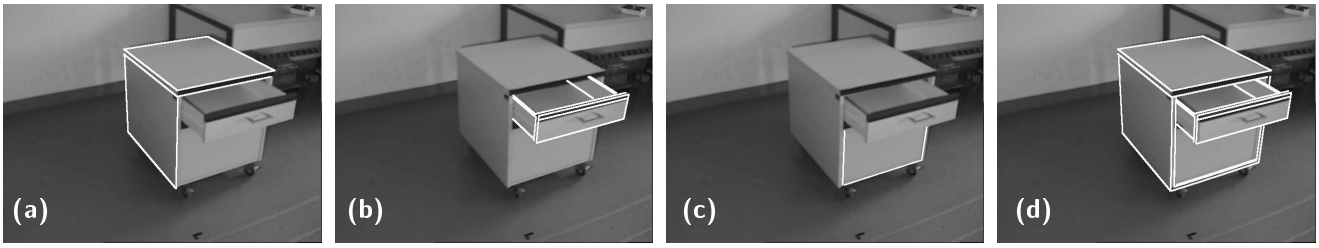

Figure 8. The recursive recognition of a drawer cabinet (with unknown joint configuration).

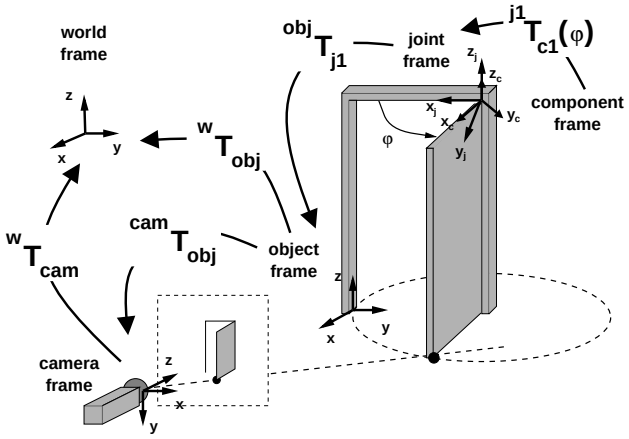

Figure 9. Determining a rotational joint state.

search space in the (θ, ρ) space of line parameters can be computed to further restrict the position and orientation of match candidates among the image lines.

Note, that inverting this mechanism might be used to determine the necessary CVs, given a maximum allowed difference between image and 2D model features. Since self-occlusion cannot be taken into account in this reverse mechanism, we use a fixed number of CVs.

In most cases the joint state of a component c_ν ($\nu \geq 1$) has less degrees of freedom than the object pose ${}^{cam}\mathbf{T}_{c_0}$. Thus, the first recognition phase described in Sec. 3.1 can be omitted. Instead the 3D model features, the fixed transform ${}^{cam}\mathbf{T}_{j_\nu}$ and ${}^{j_\nu}\mathbf{T}_{c_\nu}(\mathbf{s}_\nu)$ underlying the CVs are directly fed to the interpretation tree of the second stage. Dealing with a smaller number of DOF, the viewpoint consistency constraint controlling the tree search during the pose estimation can detect dead ends in the interpretation tree very early, thereby speeding up the process. Note, that Eq. (5) contains less DOF as well, which further facilitates the pose estimation. Figure 8 (b-c) shows the computed joint state of the two drawers of the drawer cabinet.

Joints with one DOF. In the special case of joints with only one (rotational or translational) degree of freedom, specific aggregated model features (*variant corners*) can be used to directly determine the joint state [5] alternatively to the general approach as described above. To determine the joint state it is sufficient to find one variant corner in the image; the state can then easily be calculated by intersecting the path of the variant corner with its projection ray. In case of a rotational joint this path is a circle, see Fig. 9.

Figure 10. (a) Recognition of a door, (b) the computed search space for the two radial model lines of the door-wing corresponding to an uncertainty of $\pm 20°$ in the opening angle.

A simple backtracking mechanism is incorporated into the hierarchical object recognition process. If the localization of a component fails, a new hypothesis for its parent component is tested.

4 Experiments

Our system was tested within an interdisciplinary research project dealing with autonomous mobile robots. The ability to recognize articulated objects increases the flexibility of such a robot, e.g. enables it to open a door or grasp objects out of a drawer cabinet.

Analogous to the determination of the joint configuration of a movable drawer cabinet (Fig. 8 (a-d)) consisting of two translational joints, one example for the determination of a rotational joint state is shown in Fig. 10 (a). First, the static component of a door (the door–frame) is localized taking into account the model features, which are potentially hidden by the door–wing, see Sec. 2.3. Using the estimated pose of the door–frame, characteristic views of the door–wing are predicted. For each model feature a specific search space is computed as described in Sec. 3.2 (see Fig. 10 (b)).

Figure 11. Determining the joint configuration of a whiteboard.

Figure 11 shows the recognition of a whiteboard, an object with one translational and two rotational joints. Once again, following the hierarchical object structure the recognition process first estimates the pose of the static component. Since all predicted model features of the middle part of the whiteboard are hidden by the masks of the two wings (with still unknown joint state), the correspondence search takes into account the *possibly-occluded* model features as well. After determining the translational joint between the middle part and the static component, the two rotational joints regarding the wings are determined.

5 Conclusion

A systematic framework for the representation of articulated objects has been introduced. The hierarchical model structure decomposes an object into its (rigid) components which are connected by translational and/or rotational joints. The recognition of articulated objects follows this tree-like structure and determines step-by-step the joint configuration taking into account self-occlusion and the kinematic constraints of the components.

Future work will focus on the following topics: The underlying model description has to be extended in order to deal with different primitive features. Object components should include parametric attributes to cope with (generic) object classes. Furthermore, the current backtracking mechanism has to be extended. If the recognition of a single object component c_μ fails, an algorithm for "jumping" directly to the subcomponents of c_μ (*skipping*) has to be developed. Alternatively, the search order of the object hierarchy could be alternated to cope with undetectable components, thus increasing the robustness of the proposed approach.

References

[1] M. Dhome, A. Yassine, and J. M. Lavest. Determination of the Pose of an Articulated Object From a Single Perspective View. In *British Machine Vision Conference*, pages 95–104. BMVA Press, 1993.

[2] 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.

[3] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics – Principles and Practice.* Addison Wesley, Reading, Massachusetts, 1990.

[4] A. Hauck and N. O. Stöffler. A Hierarchical World Model with Sensor- and Task-Specific Features. In *Int. Conf. on Intelligent Robots and Systems*, pages 1614–1621, 1996.

[5] A. Hauck and N. O. Stöffler. Video-Based Determination of the Joint States of Articulated Objects. In *Int. Conf. on Robotics, Vision and Parallel Processing for Industrial Automation, Ipoh, Malaysia*, pages 1018–1023, 1996.

[6] M. Hebert, J. Ponce, T. Boult, and A. Gross, editors. *Int. NFS–ARPA Workshop on Object Representation in Computer Vision, New York City, USA*, LNCS 994 Springer-Verlag, 1994.

[7] Y. Hel-or and M. Werman. Constraint Fusion for Recognition and Localization of Articulated Objects. *Int. J. Computer Vision*, 19(1):5–28, July 1996.

[8] T. Kratchounova, B. Krebs, and B. Korn. Erkennung und Bestimmung der aktuellen Konstellation von Objekten mit Scharniergelenken. In *Mustererkennung 1996*, DAGM, pages 502–509. Informatik aktuell, Springer-Verlag, 1996.

[9] S. Lanser and T. Lengauer. On the Selection of Candidates for Point and Line Correspondences. In *Int. Symp. on Computer Vision*, pages 157–162. IEEE Computer Society Press, 1995.

[10] S. Lanser, O. Munkelt, and C. Zierl. Robust Video-based Object Recognition using CAD Models. In *Intelligent Autonomous Systems IAS-4*, pages 529–536. IOS Press, 1995.

[11] S. Lanser and C. Zierl. On the Use of Topological Constraints within Object Recognition Tasks. In *13th Int. Conf. on Pattern Recognition*, pages 580–584. IEEE Computer Society Press, 1996.

[12] D. G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic, Boston, MA, 1985.

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

[14] J. Ponce, A. Zisserman, and M. Hebert, editors. *Int. Workshop on Object Representation in Computer Vision II, Cambridge, U.K.*, LNCS 1144. Springer-Verlag, 1996.

[15] A. R. Pope. Model-Based Object Recognition. Technical Report TR-94-04, Univ. of British Columbia, 1994.

[16] M. Sallam, J. Stewman, and K. Bowyer. Computing the Visual Potential of an Articulated Assembly of Parts. In *Third Int. Conf. on Computer Vision*, pages 636–643. IEEE Computer Society Press, 1990.

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