VIDEO-BASED DETERMINATION OF THE JOINT STATES OF ARTICULATED OBJECTS*

Alexa Hauck, Norbert O. Stöffler

Laboratory for Process Control and Real-Time Systems
Prof. Dr. -Ing. G. Färber
Technische Universität München
80290 München, Germany
fax: +49-89-289-23555   e-mail: {hauck|stoffler}@lpr.e-technik.tu-muenchen.de

Abstract

Pose estimation is a classical problem in computer vision applications like object recognition or robot navigation. Most approaches deal with rigid objects only, though.

We propose an algorithm for the determination of joint states that is based upon a hierarchical object model. This allows to break down the complex registration task into simpler ones, as at every level now only one degree of freedom has to be fixed. Furthermore, the model allows to define task-specific feature types which facilitate the matching process by integrating contextual information.

*The work presented in this paper is supported by the *Deutsche Forschungsgemeinschaft* as part of an interdisciplinary research project on "Information Processing in Autonomous Mobile Robots" (SFB 331).

1 Introduction

Model–based pose estimation, that is recovering the pose of an object by fitting 3D models to 2D image data, is a classical problem in computer vision and fairly well covered in literature. It is important for many applications such as object recognition, tracking or robot navigation. Most approaches deal with rigid objects only, though, not least because articulated objects as e.g. doors or manipulators may present many different views to a camera, thus soon leading to a combinatorical explosion in conventional matching algorithms. Existing methods for the determination of joint states of articulated objects use computationally expensive methods; Lowe [5] e.g. iteratively solves a system of equations with free parameters modelling the degrees of freedom, Hel–Or & Werman [3] model joints as constraints between features and then estimate pose and joint states with the help of a Kalman Filter.

We propose an intuitive algorithm for the determination of joint states that is based upon a hierarchical object model which has been developed for the use on autonomous mobile robots [2]. In this framework, articulated objects are modelled as a tree of sub–objects connected by joints with one degree of freedom. This allows to break down the complex registration task into simpler ones, as at every level now only one degree of freedom has to be fixed. The algorithm is thus easy to implement, needs no sophisticated matching routines and is very robust since model information is used extensively to facilitate sensor data interpretation.

In section 2 we will briefly describe the model structure, followed by the method for joint state determination (section 3) and experimental results (section 4).

2 Model structure

Objects influence sensor images in two ways. They can be the source of sensor–specific features and they can hide other elements. The latter aspect is modelled by a polyhedral boundary representation.

Whenever possible, sensor–specific features are calculated from the boundaries using the corresponding sensor model. In the case of a video sensor such a sensor model is difficult to obtain because of its dependency on various factors like colour and illumination. Therefore in a first step video–specific features, up to now solely edges, are modelled by line–segments which are based on the same set of vertices as the boundaries but do not necessarily coincide with boundary edges. In a second step the model data is compared with a set of images. Only those features that can actually be detected by the sensor are kept in the model, along with an attribute describing their detectability quantitatively in terms of how good they could be fitted to image data. This combination of geometric and sensor–specific information allows a compact representation that is easy to generate and at the same time guarantees that the stored features can actually be detected by the sensor.

The description of an object is built up recursively. An object can contain so–called *member–objects*, which are connected by a joint which exhibits exactly one rotatory or translatorical degree of freedom, following the conventions used in manipulator kinematics. Each object or member–object has its own coordinate system or *frame*, whose relation to that of the parent–object is described by a homogenous transform matrix. The possible positions of a joint are normalized to the unit interval allowing a unified treatment of joint–states; additionally there exists a state called *unknown*. To deal with unknown states during a prediction, the space potentially being occupied by a moving member–object is stored as an additional boundary, called *mask*.

Each branch in the object–tree carries its own obstacle and feature description. The representation of features is extended by the possibility of defining aggregations of simple features and attributes to form complex ones. Those complex features are partly task–specific to alleviate special matching problems, thus integrating context information.

As a part of the interdisciplinary research project SFB 331 ("Information Processing in Autonomous Mobile Robots") an experimental framework implementing the model and various perception tasks dedicated to robot localization, object registration and recognition has already been realized (see [1]).

To provide a task independent interface, the model is accessed via two virtual pointers called *focus* and *zoom*. The focus points on the task relevant part of the model, i.e. on a single object. After "focussing" the joint states of this object are accessible. The zoom influences the result of the feature prediction in a way comparable to a camera zoom: After pointing it on a node of the object–tree only features of the downward parts are predicted. The most important reading access is the request of a feature prediction, but also the attributes or states of objects like the opening angle of a door can be queried. Writing accesses include changing the state of an object, updating the boundary or feature description and inserting newly explored elements.

3 Determination of joint states

The presented recursive object structure allows a likewise recursive algorithm for the determination of joint states: First the static part of the object is registrated using the object recognition system *MORAL* [4], and with it the joint axes of its member–objects. Then the joint state of the first member–object is determined, followed by those of its own member–objects.

To identify a joint state three task–specific types of features have been found to be appropriate: *Radial variant edges* (edges whose starting point coincides with the axis), *parallel variant edges* (edges that are parallel to the axis) and *variant corners*, which connect a radial and a parallel variant edge. To determine a joint value it

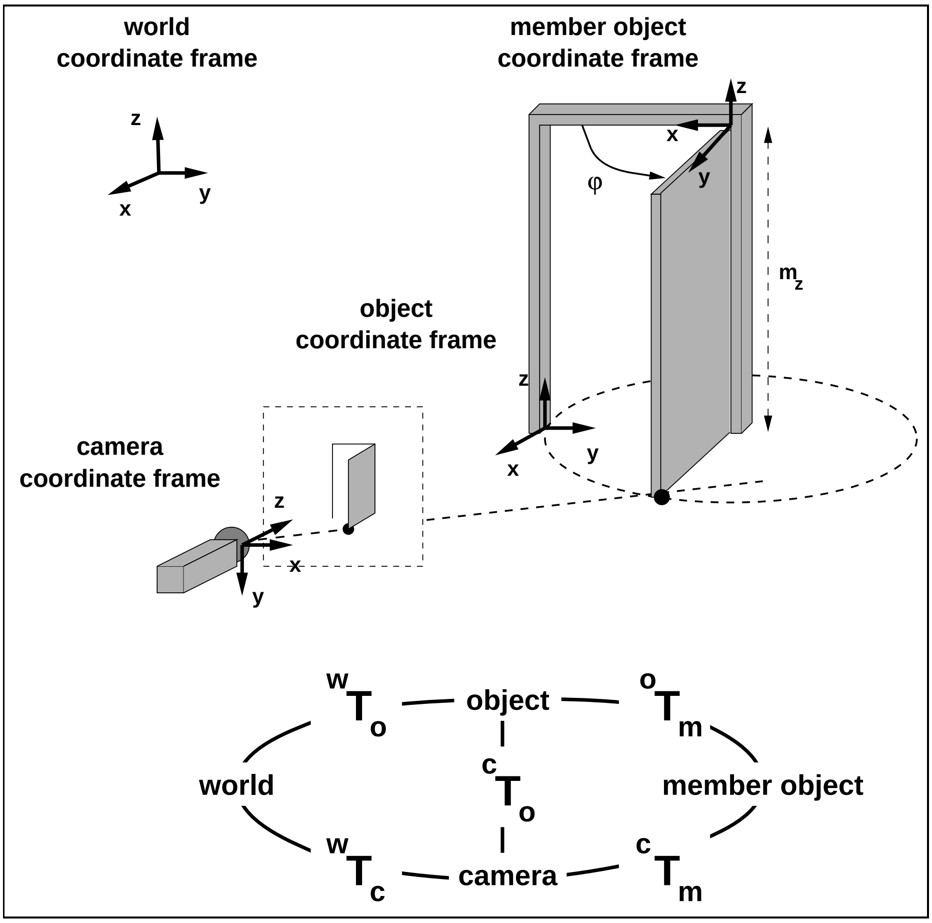

Figure 1: Determining the state of a rotary joint

is sufficient to locate one variant corner in the image; the state can then easily be calculated by intersecting the path of the variant corner (circle in the case of a rotary, straight line in the case of a translatorical joint) with its corresponding projection ray. Figure 1 illustrates this at the example of a door, an object with one rotary degree of freedom.
For the corresponding intersection equation

$$\begin{pmatrix} 0 \\ 0 \\ m_z \end{pmatrix} + r \cdot \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{pmatrix} = {}^m\mathbf{T}_c \cdot \lambda \cdot \begin{pmatrix} x_{2D} \\ y_{2D} \\ f \\ 1 \end{pmatrix}$$

x_{2D}, y_{2D} : pixel coordinates of the corner $\qquad f$: focal length

the transformation matrix between camera and member coordinate frame ${}^m\mathbf{T}_c$ has to be known. It can be computed following the kinematic chain as

$${}^m\mathbf{T}_c = \left({}^w\mathbf{T}_o \cdot {}^o\mathbf{T}_m\right)^{-1} \cdot {}^w\mathbf{T}_c = \left({}^c\mathbf{T}_o \cdot {}^o\mathbf{T}_m\right)^{-1}$$

Depending on the utilized algorithm, the localisation process either yields the position of the door-frame relative to the camera (${}^{c}\mathbf{T}_{o}$) or the position of the camera in the world (${}^{c}\mathbf{T}_{w}$); the model supplies matrices ${}^{w}\mathbf{T}_{o}$ and ${}^{o}\mathbf{T}_{m}$.
Since a corner represents two parameters for only one degree of freedom, the remaining one can be used to estimate the confidence of the computed state. The conditioning of the intersection equation itself directly reflects the 'perceptability' of a variant feature; singularities correspond to cases when a change of the joint state doesn't result in a change of the image feature. This can be used as a criterion for the selection of features or even suitable camera poses.

4 Experimental results

Our experimental setup consisted of a single off–the–shelf grey–scale CCD–camera without dedicated image processing hardware. Figure 2 shows video images of a whiteboard, an object with one translatory and two rotary degrees of freedom; extracted features are inserted in white, the model features in black.

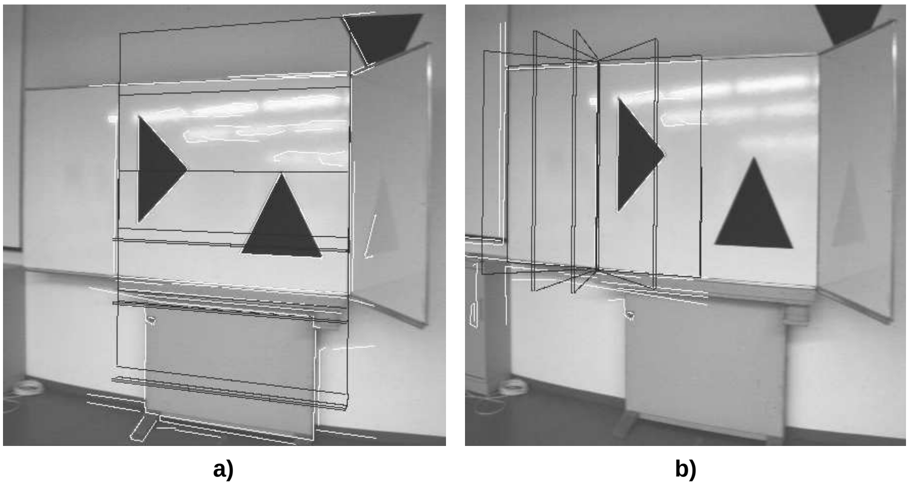

Figure 2: Determining the joint states of a whiteboard

At the beginning all states are unknown. After the localisation of the invariant part (not shown) the first member–object is zoomed on and predictions for several states are requested (figure 2a). For each "snapshot" model and image edges are matched, using a standard matching algorithm that weights the differences in length, angle and distance and computes a total match value for the object; the corners of the snapshot with the best total match value are used to calculate the joint state. After updating the model the focus is moved to the middle part, the zoom to its first member–object, the left wing, and again snapshots are requested (figure 2b). The right wing is treated analogously.

In contrast to iterative solutions this recursive method treats only one free parameter at a time, which simplifies the computational task considerately. Dependencies between joints are taken into account automatically by updating the model after each step. The definition of task–specific features allows to concentrate on the relevant model information which limits the number of possible correspondences and reduces the probability of mismatches. Back–tracking is facilitated by the hierarchical model structure.

Even without dedicated image processing hardware, the algorithm has been found to be quite fast, the determination of one joint state currently taking less than $0.4s$.

5 Conclusion

We have presented an intuitive algorithm for the determination of joint states of articulated objects from single grey-scale CCD images. Image interpretation is facilitated by extensive use of information stored in a hierarchical object model.

Further research will concentrate on developing strategies for the case of self-occlusion, that is the occlusion of an object by its member-object. Long term goal is the integration of this method into the object recognition and localisation process of MORAL, which is currently limited to objects without internal degrees of freedom.

References

[1] A. Hauck and N. O. Stöffler. A Hierarchic World Model supporting Video–Based Localization, Exploration and Object Identification. In *2. Asian Conference on Computer Vision*, pages 176–180, 1995.

[2] A. Hauck and N. O. Stöffler. A Hierarchic World Model with Sensor- and Task-Specific Features. In *International Conference on Intelligent Robots and System (IROS'96)*, Osaka, Japan, 1996.

[3] Y. Hel-Or and M. Werman. Constraint–Fusion for Interpretation of Articulated Objects. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pages 39–45, june 1994.

[4] S. Lanser and C. Zierl. On the use of topological constraints within object recognition tasks. In *13th International Conference on Pattern Recognition, Wien*, pages 580–584, 1996.

[5] D. G. Lowe. Fitting parametrized 3-d models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:441–450, 5 1991.