

In Proc. of the "Robotik 2002" Conference, VDI-Bericht Nr. 1679, June 2002.

Gesture Recognition in Spatial Context for Commanding of a Domestic Robot

Jörg Illmann, Boris Kluge, and Matthias Strobel
{illmann, kluge, mstrobel}@faw.uni-ulm.de

Research Institute for Applied Knowledge Processing,
Helmholtzstr. 16, 89081 Ulm, Germany

Abstract

The use of human gestures for interacting with robot systems in domestic environments is investigated. Special attention is paid to the recognition of the user's intent behind a gestural action. The main advantage of our approach is that the human's movement together with valuable information extracted from a spatial scene representation is directly considered while trying to uncover the intention behind a human's gesture. To uncover the intention of dynamic human gestures we are using continuous density hidden Markov models. As application example, instructing a domestic service robot is considered. An event driven control architecture permits easy context switching and meets the demands of an interactive robot assistant.

Keywords: intuitive robot programming, gesture recognition, spatial modelling, domestic service robot.

1 Introduction

Housekeeping robots focus on the employment of assistive systems in everyday domestic settings. There are different motivating factors for the employment of robots at home: on one side, comfort factors and a changing societal framework favor the employment of man-made personnel; on the other side, an increasing number of households include inhabitants that require physical support in day-to-day life due to sickness or age. Robot systems will work directly with people in this area, thus placing a central importance on making interactions between people and machines as natural as possible.

A domestic service robot should work together with and support the human user, e.g. in fetch-and-carry duties and tasks such as setting, clearing, and cleaning the table. At the FAW Ulm, we developed the prototype service robot *CleaningAssistant* (see Figure 1) which is able to perform these tasks. In Section 2 of this article, design and sensor setup of this cleaning assistant is envisaged. The robot system already possesses a sufficient base of high-level and specialized skills to avoid expendable and tedious instruction and programming of rudimentary skills. E.g. considering the task of cleaning several surfaces in a kitchen, it should be sufficient to show the robot which surfaces it has to clean and not to teach the whole cleaning trajectory. The main purpose of human interaction with the robot assistance system will be the commanding and teaching of the robot, but it also offers interesting possibilities of increasing the performance of the entire system. According to our conviction, natural instruction, interaction and programming of robots will play a decisive role in the entering of such systems in home environments. Therefore we decided to use techniques based on human gestures, compliant motion and optional speech. They can be expected to be natural and convenient. A detailed treatment of

the interactions with our domestic robot based on compliant motion can be found in [Marrone et al., 2002]. During the last years great efforts have been made to make human gesture recognition possible and fairly robust. See e.g. [Pavlovic et al., 1997] for a review of the work done in visual recognition of hand gestures. In our work, special attention is payed to the recognition of the user's intent behind a gestural action (Section 3). The main advantage of our approach is that the human's movement together with valuable information extracted form a spatial representation of the online monitored scene is directly considered while trying to uncover the intention behind a human's gesture. Finally, the event driven control architecture on *CleaningAssistant* which permits easy context switching and meets the demands of an interactive robot assistant is introduced in Section 4.

2 *CleaningAssistant* – A Domestic Cleaning Robot



Figure 1: *CleaningAssistant*: A domestic service robot developed at the FAW Ulm.

The *CleaningAssistant*, shown in Figure 1 consists of a mobile base and a manipulator on top of it. The manipulator joints as well as the differential drive system for the mobile base are built of modular drive components. The 7 DOF manipulator is based on a vertical linear axis used to enhance the vertical workspace of the system. Next a SCARA-like chain of revolute joints are mounted on the linear axis. An additional degree of freedom is used to switch between the horizontal and vertical arrangement of the SCARA-like chain. Intermediate configurations are also allowed. The advantage of this arrangement is that it allows to use the advantages of the horizontally mounted chain (low energy consumption and high dynamics for horizontal arm movements) without being limited to horizontal movements only. The end-effector is either a two-finger gripper or a special end-effector, e.g. for cleaning non-textile surfaces like working places in kitchens or sinks. A discussion of the kinematics together with a solution for the inverse kinematic problem can be found in Marrone and Strobel [2001]. Sensory feedback is provided by (a) a new kind of compliant force-torque-sensor, developed at the German Aerospace Center (DLR) which is mounted between the wrist and the end-effector of the manipulator, (b) a 2D range laser-scanner being used for position estimation as well as for obstacle detection and avoidance while navigating the mobile base, (c) a trinocular stereo-vision system for gesture and object recognition and localization, (d) a magnetic field tracking system for capturing movements of the human's forearm, back of the hand, forefinger and thumb during the execution of dynamic gestures, and (e) a touchscreen for additional gesture input, e.g. for the qualitative path specification or object selection in a displayed scene representation.

3 Intuitive Gesture Interface

There are two alternative gesture recognition methods running on the *CleaningAssistant*: One method uses the stereo vision system and is able to capture a set of static hand and arm gestures. This method is convenient, since no sensor setup to be worn by the user. However, low frequency rates and limited tracking accuracy of this vision-based system does not allow to use this sensor setup for dynamic gesture recognition. Therefore a magnetic field tracking system is used for capturing movements of the human's forearm, back of the hand, forefinger and thumb during the execution of

dynamic gestures.

In both methods, the recognition of the user's intent behind a gestural action is supported by the usage of a spatial representation of the actual monitored scene. Section 3.1 gives a description of our approach to get online information about the object distribution and topology in the current scene. The recognition system contains an offline generated 3D environment model consisting of the stationary furnishing and floor plans. Haertl et al. [2001] show a semi-automatic way of 3D model generation from a real scene using a visual laser radar.

3.1 Online Scene Analysis

The purpose of the online scene analysis is to detect those objects that are not yet contained in the environment model. Our scene analysis procedure starts with a segmentation of the scene into elementary parts. A stereo camera provides color and range information both of which are used for figure-ground separation. Elementary parts are required to be compact in space and separated from each other. Since we are using disparity information, objects have to have a certain texture. To compute disparity information corresponding pixels are computed by maximization of correlations of the local gradient within a local window. The estimated disparity is validated by analyzing the local texture, the uniqueness of the correlation value and whether adjacent pixels have close disparity. Finally the model of the calibrated camera is used to compute positions of all image pixels in the 3D workspace.

To locate objects and to support geometry-based object recognition several geometrical features are computed from range information. The objects extent is described by the three dimensional bounding box of the object. To characterize the objects location geometrical features are fitted to top down projections of all points which are assigned to the object.

Based on geometry information and color distribution parts of the scene are recognized as known objects. The estimated geometry has to match predefined values which are specified with some uncertainty. To distinguish between objects with a similar geometry the color distribution is considered. The color of an object is represented by a two-dimensional histogram of hue and saturation values of all pixels which are assigned to the object. The estimated color histogram is compared to histograms computed from images of known objects. The object is finally assigned to the class with the best matching histogram.

3.2 Static Gesture Recognition

People use deictic gestures to reference objects or areas of the environment where certain actions have to take place. To interact with a robotic system we mainly use pointing gestures to express our intention. Once gestures are detected the interpretation is performed depending on the task and spatial context.

To detect pointing gestures of a user we search skin colored regions of required shape, compute the pointing direction and evaluate whether the gesture is static. The detection of the users hand starts with a 3D scene segmentation. Region growing in the 3D workspace provides a set of connected components. A skin color validation procedure is applied to each of these regions. Skin color is defined by a color histogram [Feyrer and Zell, 1999], which is computed from some skin color training images. The number of skin color pixels is counted for each candidate region. If this number exceeds a predefined threshold the position of the skin region is computed. A subsequent 3D segmentation based on a simple geometrical hand model provides the hand region. After a shape validation we choose the closest detected hand region for further processing.

To estimate the pointing direction a line in the three-dimensional workspace is fit to all points which were assigned to that region. This line is used as an estimate of the the pointing direction and to normalize the orientation of the hand to get viewpoint independent view of the hand.

The normalized view of the hand region is classified by comparison of orientation histograms against a set of predefined hand gesture. We have chosen a set of six different static gestures (pointing, stop, up, down, star and fist) to interact with the robot (see Strobel et al. [2001]).

A pointing gesture can not be recognized based only on the hand shape. To be valid the posture has to be static, the pointing direction should not change. Therefore we verify the pointing direction of subsequently detected gestures. If the direction does not move for a certain amount of time the pointing is valid.

3.3 Dynamic Gesture Recognition

Different to the common method of recognition, classification and recording of static gestures over time we are not interested in the gesture classification at every discrete time step. Instead, a dynamic gesture is characterized by a spatio-temporal sequence of features resulting from a human hand's movement. This features are twofold and can be subdivided into

- (a) features derived only from the movement of the hand and other tracked extremities. So far, we consider the hand, forefinger, thumb and forearm position and orientation.
- (b) *spacial context features*, which are features derived from the position and orientation of the (pointing) forefinger and the spatial geometric model of the environment using methods from computer graphics [Foley et al., 1996].

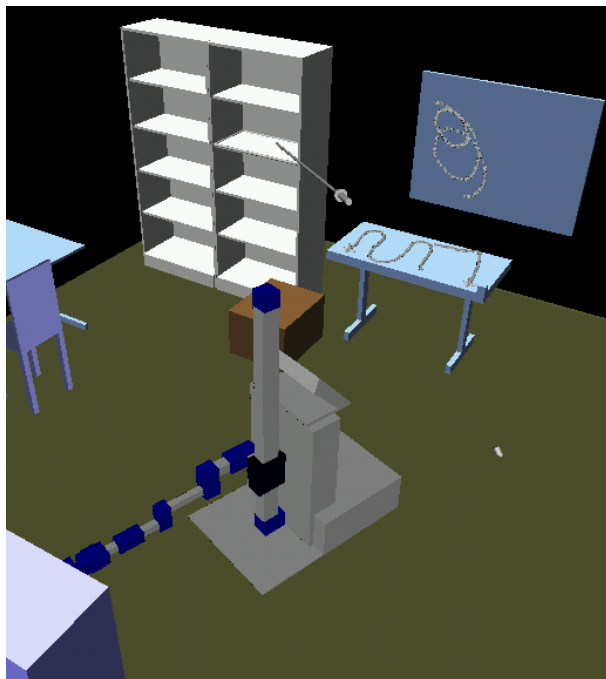


Figure 2: Projections onto scene objects resulting from human's hand movements.

model identifies the gesture. Different to the way HMM are used in most applications, our HMM topology is not strictly what can be summarized as left-right models, since a lot of isolated sub-gestures are cyclic or can be performed in different ways.

Based on this spatio-temporal feature sequence our aim is to find the intention of the human's overall movement within the spatial context. To uncover the intention of the human's movement producing the spatio-temporal feature sequence we are using continuous density *hidden Markov models* (HMM). A tutorial on hidden Markov models together with selected applications in speech recognition can be found in [Rabiner, 1989]. The elegance and power of the HMM framework is that given a set of training examples corresponding to a particular model, the parameters of the model can be determined automatically by a robust and efficient re-estimation procedure. Thus, provided a sufficient number of representative examples of each gesture was collected, a HMM can be constructed which implicitly models all of the many sources of variability inherent in human gesture execution. The use of HMMs for (isolated) gesture recognition can be summarized as follows: Firstly, a HMM is trained for each isolated gesture using a number of examples of that gesture duration. Secondly, to recognise some unknown gesture, the likelihood of each model generating that gesture is calculated and the most likely

Figure 2 shows an example of a scene with projections onto scene objects resulting from human's hand movements during a instruction run. The intention of the human there is to instruct the robot to (a) clean the whole blackboard with circular cleaning motion primitives, (b) clean the tables surface using parallel paths movements and wipe along the tables outside margin, and (c) show the robot a place in a rack where it has to put objects into.

Another advantage off using HMM is that their temporal segmentation (given through the resulting internal state sequence of a successful recognition run) can be used for robust parameter extraction from the gesture. E.g. consider the parallel path movements from the above example: Here the temporal sequence is segmented into *left turns*, *straight line movements* and *right turns*. Considering only the straight line movements the parameters *preference direction* and *desired displacement* for the parametrization of the cleaning skill are extracted from the noisy human gesture execution.

4 Control Architecture for the *CleaningAssistant*

The *CleaningAssistant* has a set of basic skills at its disposal that are provided by a set of server software components. The purpose of the control component is to put each of the servers into a mode of activity according to the current overall task context as well as to provide the active servers with the information they currently need to accomplish their tasks.

SmartSoft Server Components. The server components of the robot assistant are implemented in C++ using the SmartSoft framework Schlegel and Wörz [1999] which provides classes for multi-threaded server components as well as communication patterns for the interaction between components on the same and on different levels of the control architecture. SmartSoft offers the following patterns: A *command* or *query* may be sent to a server with or without status information to be sent back. The *update* patterns provides a server push mechanism to send data to interested clients. *Events* are emitted by servers and received by any client that indicated its interest. The mode of activity of a server component is controlled by setting its *state*.¹ Command, query, and update patterns are mostly used between components on the same level of the control hierarchy, event and state patterns being used between different levels.

Hierarchical Finite State Machines. The overall architecture of the control component is a hierarchical finite state machine. Hierarchical FSMs are defined recursively, the abstract class representing states of an FSM being a base class for the class representing FSMs. This conforms to the composite design pattern Gamma et al. [1997] and allows to use FSMs as composite states in super-ordinate FSMs where they play the role of subroutines. This helps to reduce the descriptive complexity with which the human designer of the control component is confronted, which we consider an important aspect of our approach.

Symbol dispatching differs from conventional FSMs where there is only one level of states. Here, the current input symbol s is first passed to the most specific state² q of the state machine. If q is able to handle s , the input symbol is consumed and dispatching terminates. Otherwise, s is recursively passed up to the FSM containing q (which is again a state in an object oriented sense). Finally, recursion terminates at the top-level FSM. Note that only the first state encountered during this recursive ascend really is a state which may handle the symbol by performing some action, i.e. sending commands to the servers. Other states encountered during ascend are in fact FSMs which possibly handle and thereby consume the symbol by performing a state transition.

Communication between Servers and Control. A state of an FSM assigns to each SmartSoft server component a specific mode of activity. Additionally, each state of the FSM defines which events

¹In order not to confuse the reader, we will refer to "mode of activity" instead of "state" when we talk about state of a SmartSoft server component below.

²The most specific state of an FSM is the most specific state of its current state. The most specific state of a state is the state itself.

from SmartSoft server modules are of interest to it and are to be activated on entering and deactivated on leaving respectively. Events from server components are turned into symbols and passed to the FSM for further dispatching as described above.

An FSM state may use any of the remaining SmartSoft patterns to communicate with the server components whenever a symbol arrives at that state or that state is to be entered or left. Commands without answer are simply sent to the server without any further processing needed. When commands with status information or queries are sent to a server, the respective answer will be turned into a symbol and passed to the FSM for dispatching. Server pushed data is not fed directly into the FSM, but the latest data is always accessible from handler methods without waiting.

This organization of communication and configuration allows to build a mostly non-modal system: the user may decide at any time to change to another major mode or task, even if the current subtask is not yet finished.

Acknowledgement: This work was supported by the German Department for Education and Research (bmb+f) under grant no. 01 IL 902 F6 as part of the project MORPHA.

References

- S. Feyrer and A. Zell. Detection, tracking, and pursuit of humans with an autonomous mobile robot. In *Proc. of International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 83–88, 1999.
- Foley, van Dam, Feiner, and Huges. *Computer Graphics - Principles and Practice*. Addison - Wesley Publishing Company, Inc., 1996.
- Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1997.
- F. Haertl, I. Heinz, and C. Fröhlich. Semi - automatic 3d cad model generation of as - built conditions of real environments using a visual laser radar. In *In Proc. of the 10th IEEE Inter. Workshop on Robot-Human Interactive Communication (ROMAN'01), Paris, France, 2001*.
- F. Marrone, F.M. Raimondi, and M. Strobel. Compliant interaction of a domestic service robot with a human and the environment. In *submitted to 33rd Int. Symposium on Robotics, Stockholm, October 7-11, (ISR 2002), October 2002*.
- F. Marrone and M. Strobel. CleaningAssistant - a service robot designed for cleaning tasks. In *Proc. Advanced Mechatronic Systems (AIM 2001)*, 2001.
- V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. PAMI*, 19(7), 1997.
- Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
- C. Schlegel and R. Wörz. The software framework smartsoft for implementing sensorimotor systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS '99*, pages 1610–1616, 1999.
- M. Strobel, J. Illmann, and E. Prassler. Intuitive programming of a mobile manipulator system designed for cleaning tasks in home environments. In *Proceedings of International Conference on Field and Service Robotics (FSR 2001), Helsinki University of Technology (HUT), June 2001*.