

Gesture Recognition in a Spatial Context for Commanding a Domestic Service Robot

Matthias Strobel, Jörg Illmann, Boris Kluge, and Fabrizio Marrone

Research Institute for Applied Knowledge Processing (FAW)

Helmholtzstr. 16, 89081 Ulm, Germany

{mstrobel, illmann, kluge, marrone}@faw.uni-ulm.de

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 0 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 modeling, domestic service robot.

1 INTRODUCTION

Housekeeping robots focus on the employment of assistant systems in everyday domestic settings. There are different motivating factors for the employment of robots at home. On the one hand, comfort factors and a changing societal situation favor the employment of artificial personnel. On the other hand, an increasing number of households include inhabitants that require physical support in everyday life due to sickness or age. In this domain, robot systems will cooperate directly with people, attaching central importance to rendering interactions between people and machines as natural as possible.

A domestic service robot should cooperate with and support the human user, for example in fetch-and-carry duties and tasks such as setting, clearing, and cleaning a table. At the FAW Ulm, we developed the *CleaningAssistant*, a prototype of a service robot (see Figure 1) which is able to perform these tasks. In Section 2 of this article, design and sensor setup of the cleaning assistant are presented. The robot system already possesses a sufficient base of high-level and specialized skills to avoid expendable and tedious instructing and program-

ming of rudimentary skills. Considering for example 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. While the main purpose of interaction with the robot assistant system is the commanding and teaching of the robot, it also offers interesting possibilities to increase 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 (see Section 3.2 and 3.3), compliant motion (see [1] and Figure 2), and (optionally) speech recognition. They can be expected to be natural and convenient.

During the last years great efforts have been made to make human gesture recognition possible and fairly robust. See e.g. [2] for a review of work done in visual recognition of hand gestures. In our work, special attention is paid to the recognition of the user's intent behind a gestural action (see Section 3). The main advantage of our approach is that the human's movement together with valuable information extracted from 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 software control architecture which permits easy context switching and meets the demands of an interactive robot assistant is introduced in Section 4.

2 DESIGN OF THE *CLEANINGASSISTANT*

The *CleaningAssistant* is shown in Figure 1 and consists of a mobile base and an attached manipulator. The manipulator joints as well as the differential drive system for the mobile base are built from 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.



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

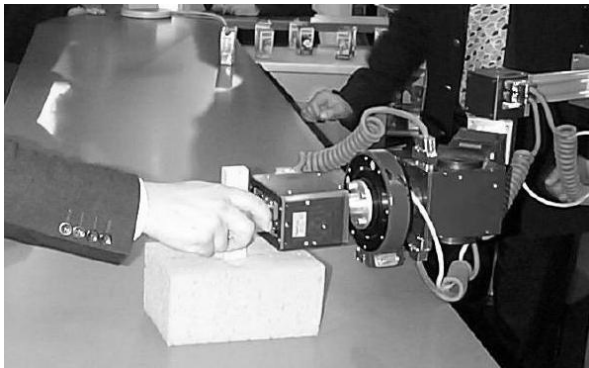


Figure 2: Compliant motion: directing the end-effector.

Intermediate configurations are also allowed. The advantage of this arrangement is that it allows to benefit from 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 [3].

Sensory feedback is provided by (a) a new kind of compliant force-torque-sensor, developed at the German Aerospace Center (DLR), and 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 touch-screen for additional intuitive input, e.g. for the qualitative path specification when commanding a motion of the base or for object selection in a displayed scene representation.

3 INTUITIVE GESTURE INTERFACE

There are two alternative gesture recognition methods employed 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 has to be worn by the user. However, low image analysis rates and limited tracking accuracy of this vision-based system do 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. Such environment models may be generated semi-automatically, for example see [4] for a 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.

Object Localization

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

Object Classification

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 in 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 their interpretation is performed depending on the task and spatial context.

Hand Detection

To detect pointing gestures of a user we search skin colored regions of the required shape, compute the pointing direction and evaluate whether the gesture is static. The detection of the user's 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 [5], 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



Figure 3: Indicating the object that is to be cleaned.

further processing.

Pointing Direction

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

Gesture Classification

This viewpoint independent description 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 [6]).

Static Pointing Gestures

A static pointing gesture can not be recognized based only on hand shape, as the shape is the same as for transient (dynamic) pointing gestures. 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.

As an example, Figure 3 shows one of the authors commanding the robot to clean the table by means of a pointing gesture during a demonstration of the system at the Hanover fair.

3.3 Dynamic Gesture Recognition

In contrast to the common method of repeated 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 spatiotemporal sequence of features resulting from a human's hand's movement. These 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 con-

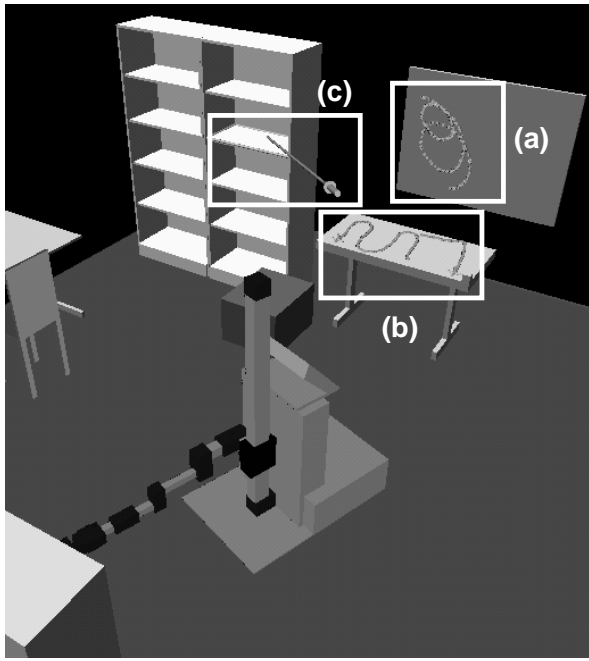


Figure 4: Projections onto scene objects resulting from human’s hand movements.

sider the hand, forefinger, thumb and forearm position and orientation.

- (b) features derived from the spatial context, for example objects or regions in the environment the forefinger is currently pointing at, which are extracted from the geometric model of the environment using methods from computer graphics [7].

Based on this spatiotemporal feature sequence our aim is to find the intention of the human’s overall movement within the given spatial context. To uncover the intention of the human’s movement producing the spatiotemporal feature sequence we are using continuous density *hidden Markov models* (HMM). A tutorial on hidden Markov models with selected applications to speech recognition can be found in [8]. 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 that 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 to 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 for that gesture. Secondly, to recognize some unknown gesture, the likelihood of each model to generate that gesture is calculated and the most likely 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.

Figure 4 shows an example of a scene with projections onto scene objects resulting from a human’s hand movements during a instruction run. Here the intention of the human is to instruct the robot to (a) clean the whole blackboard with circular motion primitives, (b) clean the tables surface using parallel paths movements and wipe along the table’s outside margin, and (c) show the robot a place in a rack where it has to put down objects.

Another advantage of 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. For example 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 parameterization of the cleaning skill are extracted from the noisy human gesture execution.

4 SOFTWARE ARCHITECTURE

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 [9] 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 *configuration*. Command, query, and update patterns are mostly used between components on the same level of the control hierarchy, event and configuration 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 [10] and allows to use FSMs as composite states in super-ordinate FSMs where they play the role

of subroutines. This helps to reduce the complexity of the description with which the human designer of the control component is confronted, which we consider an important aspect of this 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 state q of the state machine. (The most specific state of an FSM is the most specific state of its current state, and the most specific state of a state is the state itself.) 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 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.

5 CONCLUSION

In this paper we presented the design and architecture of a research prototype for a domestic service robot. Emphasis has been placed on gesture recognition as means for interaction between the user and the system. The major contribution is the employment of spatial context knowledge for the interpretation of humans' gestures. The presented system has been exhibited and successfully demonstrated to the general public at the Hanover industry fair in April 2002.

ACKNOWLEDGMENT

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

- [1] F. Marrone, F. M. Raimondi, and M. Strobel, "Compliant interaction of a domestic service robot with a human and the environment," in *Proc. of 33rd Int. Symposium on Robotics*, (Stockholm), October 2002. accepted.
- [2] V. Pavlovic, R. Sharma, and T. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. PAMI*, vol. 19, no. 7, 1997.
- [3] F. Marrone and M. Strobel, "CleaningAssistant: A service robot designed for cleaning tasks," in *Proc. Advanced Mechatronic Systems (AIM 2001)*, 2001.
- [4] 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 *Proc. of the 10th IEEE Int. Conf. on Robot and Human Interaction*, (Bordeaux-Paris, France), 2001.
- [5] 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*, pp. 83-88, 1999.
- [6] 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, Helsinki University of Technology (HUT)*, June 2001.
- [7] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Huges, *Computer Graphics: Principles and Practice*. Addison Wesley Publishing Company, Inc., 1996.
- [8] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-286, Feb. 1989.
- [9] 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*, pp. 1610-1616, 1999.
- [10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.