Distributed Control in the Multi-sensor Kernel System[1]

Tom Henderson, Bir Bhanu and Chuck Hansen

The University of Utah, Computer Science Department

Abstract

The Multi-sensor Kernel System (MKS) has been introduced as a convenient mechanism for specifying multi-sensor systems and their implementations. In this paper, we demonstrate how control issues can be handled in the context of MKS. In particular, the Logical Sensor Specification is extended to include a control mechanism which permits control information to (1) flow from more centralized processing to more peripheral processes, and (2) be generated locally in the logical sensor by means of a micro-expert system specific to the interface represented by the given logical sensor. Examples are given including a scheme for controlling the Utah/MIT dextrous hand.

Introduction

The Multi-sensor Kernel System (MKS) has been introduced as a means for obtaining and organizing data from several sensors.[1,2] The system is composed of three parts:

1. a low-level representation - the inputs from the sensors are organized in a spatial proximity graph[3]

2. a sensor specification - the sensors are described in terms of a characteristic output type produced as a result of computation on the output of other sensors[4], and

3. a high-level model - the objects to be recognized are defined in terms of their features or structures[5,2].

In this paper, we extend the sensor specification to permit the introduction of control of the sensors.

Logical Sensor Specification

Logical sensor specification (LSS) provides a design methodology for sensor systems. Such a methodology is necessary due to the emergence of multi-sensor systems of great complexity. Many of the systems require that the sensors be widely distributed both physically and computationally. Moreover, it may be necessary to define a hierarchical relationship between the sensors and/or the algorithms applied to them. For example, a dextrous hadn is made up of several fingers each of which is composed of several phalanges requiring position and tactile sensors. LSS exploits many well-known software design principles including abstraction, modularity, and separates implementation from specification. Finally, these features permit the dynamic reconfiguration of sensing resources for either fault tolerance or for a sort of adaptive sensing in response to the situation.

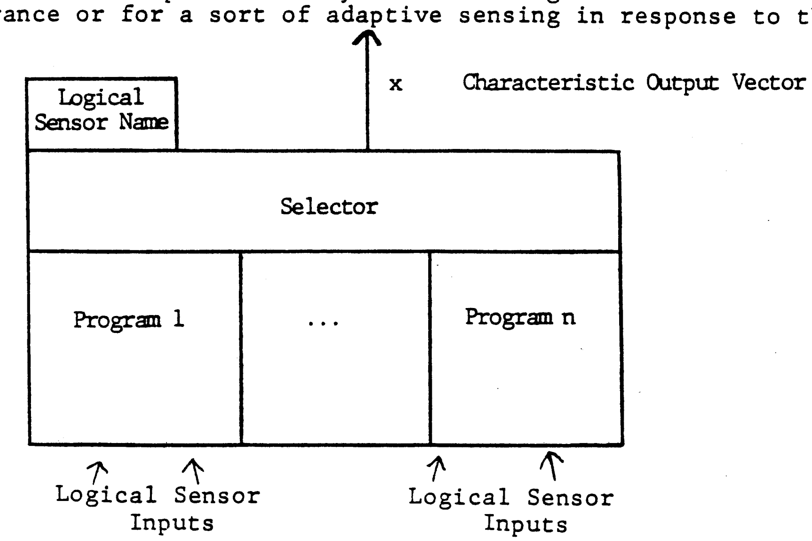

Figure 1. Logical Sensor Specification

LSS specifications are translated into s-expressions. Currently, the resulting s-expressions can be compiled into either UNIX shell script or Function Equation Language

Specification of Control

In order to solve most recognition and manipulation problems, it is necessary to be able to reposition sensors (e.g., aim cameras) and adapt rapidly to changing conditions (e.g., if an object is slipping from the grasp of a robot hand, perhaps more force should be applied). Thus, in addition to a stream of sensed data flowing from physical sensors on up through some hierarchy of logical sensors, there may also be a stream of control commands (or signals) flowing in the reverse direction. We therefore propose the following modification to the logical sensor schema:

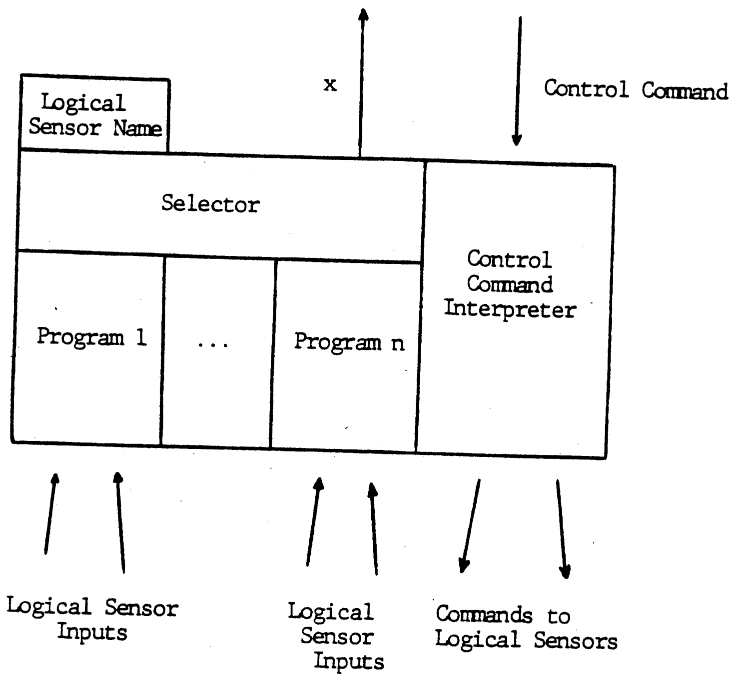

Figure 2. Logical Sensor Specification with Control

Each logical sensor now must also have a program to interpret the control commands coming from a level up in the hierarchy and to send commands down to logical sensors lower in the hierarchy. Moreover, the select function can now play a more sophisticated role in the logical sensor. Namely, the select function may now be allowed to monitor both the sensor data going up and the command stream to be issued. Given the command (or commands) to be executed and the sensor data being produced locally, the select function may be able short circuit the path back to the root logical sensor and to modify the commands to be issued directly. Such a function may be viewed as a micro-expert system which knows all about the interface represented by the logical sensor in which it is located. Thus, a logical sensor acquires some of its meaning now not simply as a sensor/algorithm combination, but also as an interface between two layers of sensing and analysis.

Viewed in this way, it becomes possible to envision the possibility that the selector function can learn some of the knowledge required to successfully administer its task. Also, it becomes clear how transfer and coordination can occur between logical sensors at the same relative level and position. Finally, it is possible for some very high level system (which has knowledge of the specification of a given selector) to modify the selector. We believe that these properties are very crucial for successful sensor system schemes.

As a specific example consider Figure 3:

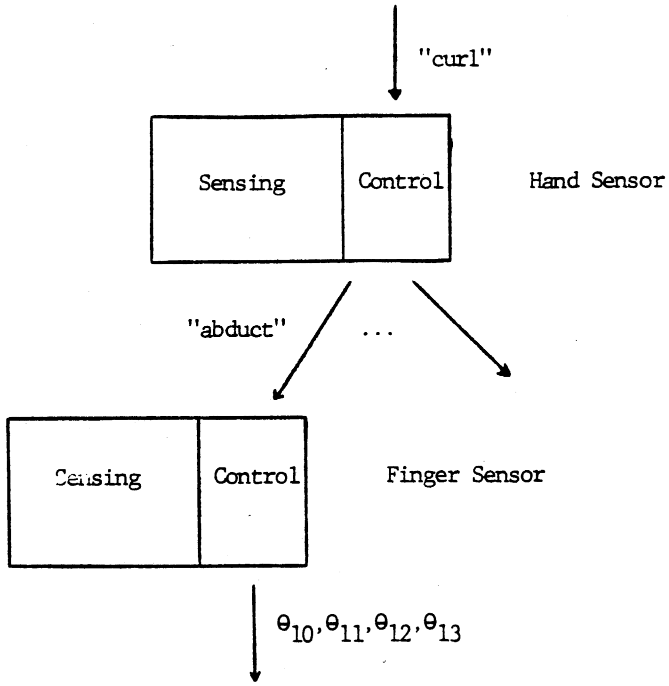

Figure 3. Part of a Robot Hand Specification

Shown here are some of the logical sensors which comprise the specification of the sensor and control scheme for the UTAH/MIT dextrous robot hand. The robot hand has four fingers each with four degrees of freedom[6]. The high level commands for hand control are interpreted as a set of commands to a lower-level right on down to the control of the joint positions of each finger which define the configuration of the robot hand.

Conclusions

We have shown how the Multi-sensor Kernel System can be extended to permit the specification of control distributed throughout the sensing system. This permits local control commands to be issued in response to local conditions in order to override commands generated from a more global source which has outdated information. It is also possible to permit some form of learning to take place in a local micro-expert system which delivers the data up the hierarchy and which emits control commands down the hierarchy. These properties are necessary if real-time performance goals are to be achieved with complex sensor and manipulation systems.

References

1. Henderson, T.C. and Wu So Fai, MKS: A Multi-sensor Kernel System, IEEE Transactions on Systems, Man, and Cybernetics to appear, 1984.
2. Henderson, Thomas C. and Wu So Fai, A Multi-sensor Integration and Data Acquisition System, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 274-280, IEEE, June, 1983.
3. Henderson, T.C., Efficient 3-D Object Representations for Industrial Vision Systems, IEEE Transactions on Pattern Analysis and Machine Vision PAMI-5(6):609-618, November, 1983.
4. Henderson, T.C. and E. Shilcrat, Logical Sensor Systems, Journal of Robotic Systems, 1(2):169-193, 1984.
5. Henderson, T.C. and Wu So Fai, The 3-D Hough Shape Transform, Pattern Recognition Letters 2:235-238, June, 1984.
6. Jacobsen, S., D.F. Knutti, K. Biggers, E.K. Iverson and J.E. Wood, An Electro-pneumatic Actuation System for the Utah/MIT Dextrous Hand, In Proceedings of the Fifth C/SM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, June, 1984.

PROCEEDINGS
Of SPIE-The International Society for Optical Engineering

Volume 521

Intelligent Robots and Computer Vision

David P. Casasent/Ernest L. Hall
Chairmen/Editors

Cooperating Organizations
Carnegie-Mellon University, Robotics Institute • Catholic University of Leuven (Belgium)
Massachusetts Institute of Technology, Artificial Intelligence Laboratory
University of Cincinnati, Center for Robotics Research • University of Rhode Island, Robotics Research Center
UCLA Computer Science Department

November 5-8, 1984
Cambridge, Massachusetts