Vision Based Robot Behavior: Tools and Testbeds for Real World AI Research

Hirochika Inoue
Department of Mechano-Informatics
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, JAPAN

Abstract

Vision is a key function not only for robotics but also for AI more generally. Today real-time visual processing is becoming possible; this means that vision based behavior can become more dynamic, opening fertile areas for applications. One aspect of this is real-time visual tracking. We have built a real-time tracking vision system and incorporated it in an integrated robot programming environment. Using this, we have performed experiments in vision based robot behavior and human-robot interaction. In particular, we have developed a robotic system capable of "learning by seeing". In general, it is important for the AI community not to lose sight of the problems and progress of robotics. After all, an AI system which acts in real-time in the real-world is no less (and no more) than an intelligent robot.

1 Introduction

A robot is a versatile intelligent machine which can carry out a variety of tasks in real-time. The interaction with the outside world is the essential aspect which distinguishes robotics from ordinary AI. In order to make this interaction more intelligent, a robot needs functions such as: the ability to understand the environment by visual recognition, the ability to perform dexterous manipulation using force, tactile, and visual feedback, the ability to plan task procedures, the ability to naturally communicate with humans, the ablity to learn how to perform tasks, the ability to recover from errors, and so on. All of these are required for robot intelligence to be realized.

From the earliest days of AI research, aspects of robot-related intelligence have been tackled; these include the principles for problem solving, planning, scene understanding, and learning. Whereas AI research generally takes the quest for the basic principles of intelligence as its goal, in robotics, the results of task planning or scene understanding are not the ultimate goal, but are rather the means for acting and reacting properly in the real world.

Visual information plays a very important role for robot-environment interaction. If provided with visual sensing, the potential repertoire of robotic behavior becomes very rich. To actually experiment with such behaviors, we need very fast visual information processing. Section 2 sketches our efforts towards high speed robot vision. This system is implemented as a multi-processor configuration, greatly enhancing the performance.

We have combined the real-time tracking vision system with a radio control system for wireless servo units, giving us a robot development system. In this approach, the robot body consists of mobility, manipulator and vision. The robot does not carry its own computer; rather it is connected with the powerful vision system and computer by radio link. Thus, this approach enables very compact robot bodies which actually behave in the real world. Section 3 describes this remote-brained approach and several environments for robot behavior research.

High speed visual tracking capability opens up another important way to make human-robot interaction smarter: "learning by seeing". Section 4 explains our preliminary experiments on this. Despite the limited performance of the vision system, the system as a whole can observe and understand pick-and-place sequences which a human acts out for the benefit of robot vision.

Section 5, discusses some future directions for real world AI research and speculates on the possiblity of developing humanoid robots.

2 A Real-time Visual Tracking System

Vision is an essential sense for robots. In particular, robot vision requires real-time processing of visual information about motion and depth. Motion information should include recognition of moving objects, tracking, and ego-motion understanding. Depth information is always necessary for a robot to act in the three-dimensional real world. Flexible interaction between visual data and motion control is also important for attaining vision based intelligent robot behavior.

2.1 Using correlations between local image regions

The fundamental operation on which our system is based is the calculation of correlation between local image regions. It computes the correlation value between a region R in image F_1 and subregions s within a search area S in image F_2, where F_1 and F_2 are either part of the same

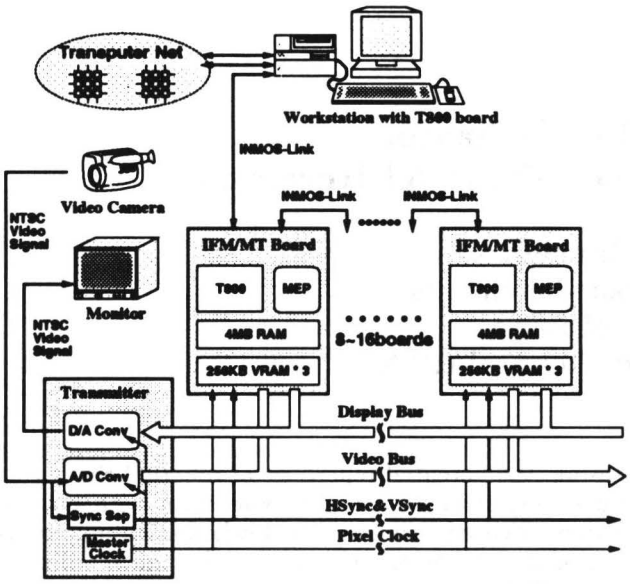

Figure 1: Visual tracking system

image, two time-consecutive image frames or left/right images at the same sampling time, and finds the best matching subregion, namely, that which minimizes the value $Cor(R, s)$.

The correlation between R and s is given by the equation

$$Cor(R, s) = \sum_{R(x) \in R, s(y) \in s} |R(x) - s(y)|$$

Although correlation is generally defined as a sum of products, we employ this simpler equation (the Mean Absolute Error criterion) to decrease the computation time.

2.2 Hardware organization

Figure 1 diagrams the organization of the hardware [Inoue 1992]. The system is implemented as a transputer-based vision system augmented with a high speed correlation processor. The transputer vision board is equipped with three image-frame memories, each of which can be used simultaneously for image input, image processing, and image display. Thus, the system can devote all its computation power to image processing without waiting for image input or display. The vision board also incorporates an off-the-shelf chip (MEP: Motion Estimation Processor [SGS 1990]), designed for image compression, but used here as a correlation processor. Using this chip, we have developed a very fast correlation based robot vision system. This system can also be used in a multi-processor configuration, greatly increasing performance. The transputer controls the image data stream for $R \subset F_1, S \subset F_2$, this data is transferred to the correlation chip, and the results are returned to the transputer.

2.3 Visual tracking based on local correlation

Real-time visual tracking is an important requirement for robot vision. In the usual approach, various feature parameters of objects, such as region center or edge information, are computed for the input image data and the objects represented by these parameters are tracked. Such approaches are simple and fast enough, however they sometimes have the drawback of over-sensitivity to noise, lighting conditions, and background image characteristics. Our method is simpler: local correlation is used to search for the corresponding location between the last and current image or between the reference image and the current input image. Until now, this method has been considered much too computation-intensive, but by using the powerful correlation chip this computation can be performed in real-time if the reference frame is of moderate size.

The tracking process repeats the following two step procedure: (1) search the reference image in the local neighborhood around the current attention point, and determine the location of the highest correspondence. (2) move the point of attention to this location.

We performed a simple experiment using the vision hardware described in the previous section. The target region R was 16×16, the search area S was 32×32, and the search region D was $\{(i,j) | -8 \leq i,j \leq +7\}$. We found that tracking for a 16×16 reference region is performed in 1.15 msec, significantly faster than possible with the transputer alone. Further, the hardware configuration using the MEP chip is very simple, compact, and inexpensive.

Using this system we can track more than 20 regions at video rate; which is more than sufficient for many real-time tracking applications. If it is necessary to track more regions a multi-processor system can be used; the number of trackable regions increases linearly with the number of processors.

2.4 Real-time optical flow computation

Although optical flow provides a very attractive way to detect motion by vision, its computation also has been extremely time consuming. Using the correlation processor, we managed to speed-up the calculation of optical flow.

The input image is divided into a set of small patch regions, each of which is correlated with the image taken at the time dt later, and the flow vector is determined as the vector from the patch region in the previous image to the best corresponding region on the subsequent image.

By using a single MEP chip the optical flow vectors for 12×12 points were computed in 51 msec. The processing time for local correlation was less than 1 msec; the rest of the time was consumed by the transputer for data dispatch from image frame memories to the MEP chip. If the data dispatch were done by a dedicated circuit the computation time would be much faster.

2.5 Stereo and depth map generation

We next attempted depth map generation based on binocular stereo matching. In the experiments, the depth map at 10×15 points was generated. The 10×15 measurement points were fixed on the left view image. The reference window to be located at each measurement point was defined as a 64×8 pixel local image.

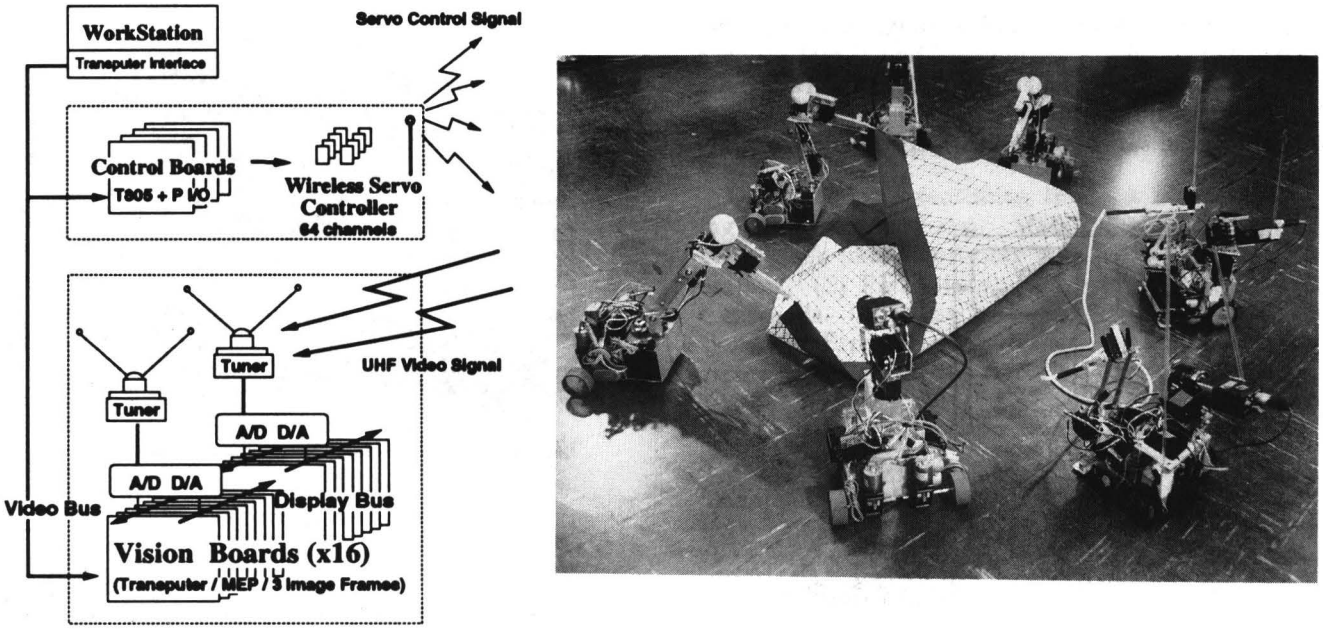

Figure 2: Robot world in remote brained approach

The reference window on the left view was matched to sub-regions within a search window of 144 × 24 pixels on the right view.

3 Applying visual tracking to robot behavior control

When the speed of visual processing reaches real-time, the nature of sensor interaction can be made dynamic instead of static. In particular, the performance of our tracking vision system enables us to perform new experiments in real-time intelligent robot behavior, such as game playing between a computer-controlled robot and a human-controlled robot.

3.1 Experimental setup: the remote-brained approach

In order to advance the study of vision based robot behavior, we built a system to serve as a base for experiments. Figure 2 shows how this system is constructed using a transputer-based multi-processor organization. It is intended to provide a high performance, flexible system for implementing vision based behavior experiments. Each workstation is interfaced with the vision unit and the motion control unit. The transputer/MEP based vision system in multi-processor configuration provides powerful sensing means for behavior control. For the controller interface, we use radio control servo units, which are available as parts for radio controlled model kits. In our system there are 64 wireless channels for servo units. The video signal is transmitted by UHF radio from onboard cameras to the vision processor. We can say that, rather than lugging its brain around, the robot leaves it at a remote computer and talks with it by radio [Inaba 1992].

In order to build an experimental setup for robot behavior study, we need to work on mechanisms, on the control interface, and on software. Until everything has been integrated, we cannot do any experiments. This is one of the things that makes robotics research time-consuming. However, the remote-brained approach can help; it partitions the work on mechanism, on interface, and on software. This approach provides a cooperative environment where each expert can concentrate on his own role. For the software engineer, the definition of the control interface can be treated as the specification of just another output device. For the mechanical engineer designing the robot hardware, the wireless servo unit can be considered as just another mechano-electrical component. We believe this approach makes it easier for AI people to face up to the real world intelligence problem. Figure 2 shows a remote-brained experimental environment consisting of seven radio-linked mobile robots.

3.2 Coordination of hand-eye robots

Using the basic hardware described in the previous section, we have built an integrated experimental environment, "COSMOS-3". COSMOS-3 enhances the real-time capacity of vision system and provides an easy interface for developing experimental robot mechanisms. We have used it in several experiments in multiple robot coordination. For instance, we made two small hand-eye robots tie a knot in a rope using visual guidance. Videotapes of several other experiments will be shown at the conference.

3.3 Computer-human sumo wrestling

Figure 3 shows the system overview. Two robots face off other in the "dohyo" ring, 150 cm in diameter. One robot is controlled by a human operator via wireless controller. The control signal of the other robot is transmitted from a computer through the radio link. Each "sumo" robot is

20 cm in length and width, and its weight is under 3 kg. The two driving wheels are powered by DC motors, each of which is controlled independently through a radio link. The maximum speed of the robot is 50 cm/sec. The two robots have the same mechanical performance to make things fair.

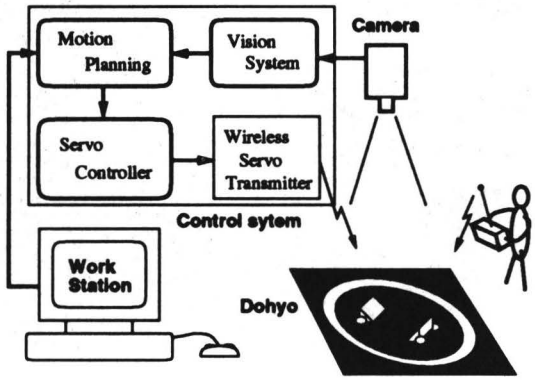

Figure 3: Robot "sumo" system

The key to the success of the experiment is the real-time visual tracking of the two battling robots. A TV camera is placed above the ring looking down at the whole environment. As the robots move in the ring, changing their position and orientation, they are observed by the vision system; their position and direction are tracked in real-time. Based on the real-time tracking of two robots's behavior, the fighting strategy and motion planning is computed.

For this application the performance of the vision system is adequate; using just one vision board the motions of both robots can be tracked completely in real-time. Experiments show that the computer controlled robot tends to beat the human controlled one. This is because the computer is quite fast in observation and control processing, and makes fewer errors in control operation than the human operator.

3.4 Towards a vision-guided autonomous vehicle

The behavior of autonomous vehicles in natural environments is another interesting goal for research on real world AI. Natural environments include not only lanes for vehicles, but also pedestrians and obstacles, both stationary and moving. We wish to develop an intelligent vehicle which behaves like an animal such as a horse. When we ride a horse, its behavior is controlled only through high-level, multi-modal communications from the human. If we let the horse free, it walks as it pleases, choosing a trail, avoiding obstacles, keeping itself safe, and interacting with other horses and moving objects. By training or teaching, a human and a horse can interact with each other for successful riding.

Figure 4 shows the design of our vehicle. Our purpose is to develop an semi-autonomous vehicle with horse-level abilities. We adapted a compact electric scooter originally designed for senior citizens. It is battery pow-

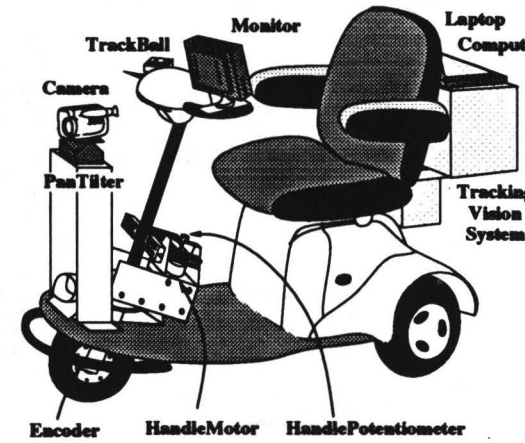

Figure 4: Hyper scooter

ered, carries a single driver, and has a maximum speed of 6 km/h. We modified it for computer control. The steering is powered by a servo-mechanism. A video camera is mounted at the front. We put a trackball and a monitor TV on the steering bar to give instructions and to communicate. At the back, we installed a high speed robot vision system and control computer. We have built this experimental prototype and have just begun preliminary experiments. Our long-term challenge is built an autonomous vehicle which can behave like a mechanical animal in being teachable/trainable.

4 Seeing, understanding and repeating human tasks

As a step towards an integrated robot intelligence, we have built a prototype system that observes human action sequences, understands them, generates robot program for the actions, and executes them. This novel method for robot programming we call "teaching by showing" or "learning by seeing" [Kuniyoshi 1990,1992]. It includes various aspects of intelligent robot behavior.

4.1 Experimental Setup

Figure 5 shows the hardware setup of the system. The system is implemented on COSMOS-2, a network based robot programming system.

(1) Camera Configuration: Task presentation is monitored by three monochrome video cameras (two for stereo and one for zoom-in) connected to the network-based robot vision server.

(2) Vision Server: Special vision hardware is connected to a host workstation. The host runs a server which accepts commands, controls the vision hardware and transmits the extracted data through a socket connection over the ethernet. The vision hardware consists of a high speed Line Segment Finder (LSF) and a Multi Window Vision System (MWVS). The LSF extracts lists of connected line segments from a gray scale image (256X256) within 200 msec [Moribe 1987]. The MWVS is a multi processor hardware component that

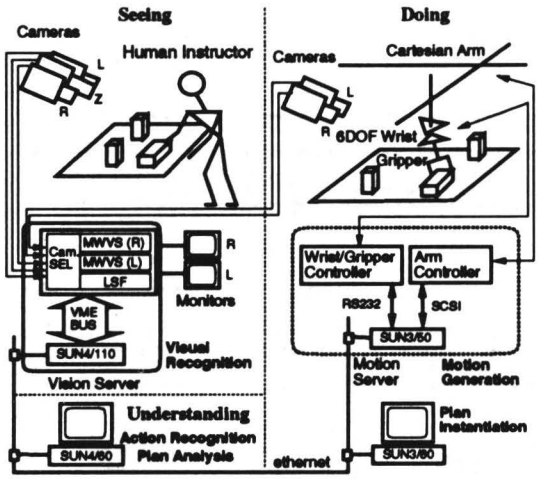

Figure 5: System for teaching by showing

extracts various image features at video rate from within rectangular "windows" of specified size, sampling rate and location [Inoue 1985b]. It can handle up to 32 windows in parallel for continuous tracking and detection of features.

(3) High-level Processing Servers: Two workstations are dedicated for action recognition and plan instantiation. The action recognizer consists of an action model, an environment model and an attention stack. It extracts visual features by actively controlling the vision server and generates a symbolic description of the action sequence. Plan instantiation involves matching this "action plan" against the environment model, which is updated by visual recognition of the execution environment. From this plan, motion commands for the manipulator are generated and sent to the motion server. The programs are written in EUSLISP, an object-oriented Lisp environment with geometric modeling facilities.

(4) Motion Server: A cartesian type arm with a 6 DOF wrist mechanism supporting a parallel-jaw gripper is used for task execution. The host workstation interprets robot commands from the ethernet and sends primitive instructions to the manipulator controller.

4.2 Required Functions

Seeing, understanding and doing must be integrated. Our approach is to connect these at the symbolic level. As shown in Figure 5, the system consists of three parts (divided by dotted lines in the figure), for seeing, understanding and doing. The following functions are performed by each of these parts:

Seeing : (1) Recognizing the initial state and constructing the environment model. (2) Finding and tracking the hand. (3) Visually searching for the target of the operation. (4) Detecting meaningful changes around the target and describing them qualitatively.

Understanding : (1) Segmentation of the continuous performance into meaningful unit operations. (2) Classification of operations based on motion types, target objects, and effects on the targets. (3) Dependency analysis of observed task procedures to infer subprocedures consisting of temporally dependent operations.

(4) Bottom-up plan inference to generate abstract operators for each subprocedure and to gather target objects and state changes descriptions from the lower-level operators.

Doing : (1) Instantiating the task plan. Recognizing the given initial state. Matching the result with the stored task plan to produce goal positions for each operation. (2) Path planning and generation of motion commands. (3) Using sensor feedback for guiding motions. (4) Error detection by vision and performance of recovery actions for the error.

4.3 Example: Recognizing a pick and place sequence

The detailed technical content will not be be described here, however, to give the flavor of teaching by showing, the process of recognition of a "PLACE" operation is sketched in Figure 6. The top arrow is the time axis annotated with scene descriptions. "Attention" lines represent continuous vision processing executing in parallel. Marks on the "Events" line show when the various events are flagged. Intervals on "Motion" lines denoted segmented assembly motions. Two types of "Snapshots" at segmentation points and their "Changes" are also shown: "(Sil.)" snapshots are gray-scale silhouettes and "(Junct.)" snapshots are connectivity configurations of local edges around the target face of an object.

(1) Recognition of Transfer: First a motion-detector is invoked. When a large movement is detected, an event "Found-moving" is raised, signaling the start of a "Transfer" motion. At the same time, a hand-tracker is invoked to track and extract motion features. For explanatory purpose, we assume that a PICK Operation was completed and a Transfer motion was started during the break marked by wavy lines.

(2) Initial point of LocalMotion: When the hand starts to move slowly downward, a "Moving-down" event is raised. This event invokes a visual search procedure. When the target object is found, a "Near" event is raised. This signals the end of the "Transfer" motion and the start of a "LocalMotion". The environment model remembers that the hand is holding an object, a fact recorded when the system recognized the previous motion as a PICK. This information gives rise to an anticipation that the held object is going down to be placed on the target object just found. A change-detector is invoked to extract and store a snapshot around the expected PLACE position.

(3) Final point of LocalMotion: The hand starts to move again. When it gets far enough away from the target object, a "Far" event is detected. This signals the end of the "LocalMotion" and the start of the next "Transfer". The change-detector takes another snapshot and finds that the area of the silhouette of the target has significantly increased. This results in identification of the operation as a "PLACE-ON-BLOCK" (if there were no change in silhouette area, it would be identified as a "NO-OP", and if there were a decrease, as a "PICK".)

(4) Updating the environment model: The environ-

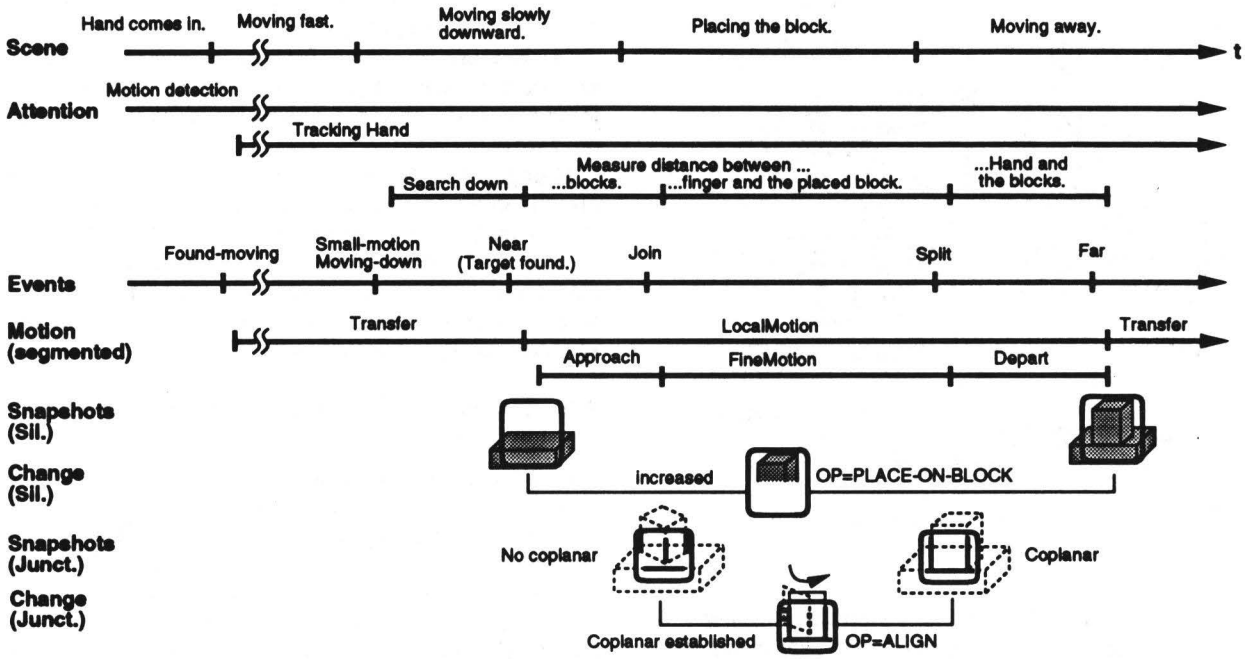

Figure 6: Recogintion flow of PLACE operation

ment model is updated, based on the operation identified, to reflect the current state of the environment. To be specific, the "Holding" relation between the hand and the placed object is deleted and an "On" relation between the placed object and the target object is added. The target position of the operation is initially estimated by measuring the center of area found by differentiating the stereo images. Then, the vertical position of the placed object is recalculated, based on knowledge of the type of operation (from the action model) and the dimensions of the objects (from the environment model), and this information is stored. Copies of environment model nodes corresponding to the hand and the object are made and stored in the "final-state" slot of the current node of the action-model.

(5) Recognition of FineMotion: A finer level of recognition proceeds in parallel with that of the "LocalMotion". The relative positions of the held object and the target object are continuously monitored by vision. When they touch each other, a "join" event is established; this signals the start of "FineMotion". A coplanar-detector is invoked and gives the result "Non-Coplanar", because the faces of the objects are not alligned at this point.

When the fingers release the placed object, an event "Split" is detected, signaling the end of "FineMotion". This time the coplanar-detector detects the "Coplanar" state. Comparing the initial and final states, the "FineMotion" is identified as an "ALIGN" operation. The coplanar relation defines the relative orientation of the objects, which is stored in the environment model.

5 Concluding Remarks : Robot behavior and real world computing

At an invited talk at IJCAI-85 I presented a system intended to help bridge the gap between AI and robotics [Inoue 1985a]. That system, called COSMOS, is a Lisp-based programming environment which integrated a 3D vision system, a geometric modelling system, and a manipulator control system. The early COSMOS was built in a mini-computer based centralized configuration. Its successor, COSMOS-2, is implemented in a network-based configuration consisting of several robot-function servers. Using COSMOS-2, we built the intelligent robot system, mentioned above, which can observe a human-performed task sequence, understand the task procedure, generate a robot program for that task, and execute it even in a task space different from the one in which it was taught. As described in section 2, we recently succeeded in devleloping a very fast robot vision system, and COSMOS-3 is the extension of this to a multi-transputer configuration, greatly enhancing its real-time capacity.

This paper has focused on our current efforts towards intelligent robots as real-world AI. The remainder of this paper presents some of our hopes and plans for the robots of the future.

Real world environments are full of uncertainty and change. However, a human brain can recognize and understand a situation, make a decision, predict, plan, and behave. The information to be processed is enormous in quantity and multi-modal. An real world intelligent system must perform logical knowledge processing, pattern information processing, and integration of the two. The Japanese Ministry of International Trade and Industry

(MITI) recently initiated "The Real World Computing Project", which aims to investigate the foundations of human-like flexible information processing, to develop a massively parallel computer, and to realize novel functions for a wide range of applications to real world information processing. As Dr. Otsu will present this project in an invited talk [Otsu 1993], I will merely make a few comments from the viewpoint of intelligent robotics.

A robot can be viewed as an AI system which behaves in the real world in real-time. In a robot system, various autonomous agents such as sensing, recognition, planning, control, and their coordinator must cooperate in recognizing the environment, solving problems, planning a behavior, and executing it. Research on intelligent robots thus covers most of what is involved in any real world agent. A robot can therefore be considered an adequate testbed for integrating various aspects of real world information processing.

As a concrete image for such a robot, I propose a humanoid-type intelligent robot, to serve as a base for the integration of real world AI research. I imagine a body designed to sit on a wheeled chair to move about (as legged walking is not an essential purpose for intelligent humanoids). I imagine a head equipped with binocular vision to see, a microphone to listen, and a speech synthesizer to talk. I imagine two arms, in a human-like configuration, with five-fingered hands. I imagine a brain capable of learning by seeing. Further, I intend to give this robot the ability to communicate naturally with humans.

To build such a robot we will have to deal with many issues. To mention a few: (1) visual observation and understanding of complex hand motions for object manipulation, (2) representation and control of coordinated motion of five-fingered robot hands, (3) sensor based manipulation skill, (4) direct visual feedback and forecast for dynamic motion, such as juggling, (5) handing flexible materials like ropes or clothes, (6) error recovery and reactive problem solving, (7) control of visual attention, (8) learning by seeing, and (9) recognition of and fusion of information from facial expression, gesture, and speech, allowing natural human-computer communication, among others. The tasks such a robot can perform will demonstrate its degree of dexterity and degree of intelligence. Our short-term goal is to build a robot that can play games with our children.

References

[Inaba 1992] M. Inaba, "Robotics Research on Persistent Brain with Remote-controlled Smart Body", Proc. 10th Annual Conference of Robotics Society of Japan, pp.1145-1148 (1992)

[Inoue 1985a] H. Inoue, "Building a Bridge between AI and Robotics", Proc. IJCAI-85, pp.1231-1237,(1985)

[Inoue 1985b] H. Inoue and H. Mizoguchi, "A Flexible Multi Window Vision System for Robots", Proc. of Second International Symposium on Robotics Research (ISRR2), pp. 95-102,(1985)

[Inoue 1992] H. Inoue, T. Tachikawa and M. Inaba, "Robot Vision System with a Correlation Chip for Real-time Tracking, Optical Flow and Depth Map Generation", Proc. IEEE International Conference on Robotics and Automation, pp.1621-1626, (1992)

[Kuniyoshi 1990] Y. Kuniyoshi, H. Inoue and M. Inaba, "Design and Implementation of a System that Generates Asembly Programs from Visual Recognition of Human Action Sequences", Proc. IEEE International Workshop on Intelligent Robots and Systems, pp.567-574, (1990)

[Kuniyoshi 1992] Y. Kuniyoshi, M. Inaba and H. Inoue, "Seeing, Understanding and Doing Human Task" Proc. 1992 IEEE International Conference on Robotics and Automation, pp.1-9, 1992)

[Moribe 1987] H. Moribe, M. Nakano, T. Kuno and J. Hasegawa, "Image Preprocessor of Model-based Vision System for Assembly Robots", Proc. of IEEE International Conference on Robotics and Automation,pp.366-371,1987.

[Otsu 1993] N. Otsu, "Toward Flexible Intelligence: MITI's New Program of Real World Computing", to be an invited talk at IJCAI-93, (1993)

[SGS 1990] SGS-THOMSON, STI3220 Motion Estimation Processor (Tentative Data), Image Processing Databook, SGS-THOMSON, pp.115-138, 1990.