BUILDING A BRIDGE BETWEEN AI AND ROBOTICS

Hirochika Inoue

Department of Mechanical Engineering
The University of Tokyo
Hongo, Bunkyo-ku, Tokyo, JAPAN

ABSTRACT

About fifteen years ago, the hand eye system was one of the exciting research topics in artificial intelligence. Several AI groups attempted to develop intelligent robots by combining computer vision with computer controlled manipulator. However, after the success of early prototypes, research efforts have been splitted into general intelligence research and real world oriented robotics research. Currently, the author feels there exists a significant gap between AI and robotics in spite of the necessity of communication and integration. Thus, without building a bridge over the gap, AI will lose a fertile research field that needs real-time real-world intelligence, and robotics will never acquire intelligence even if it works skillfully. In this paper, I would like to encourage AI community to promote more efforts on real world robotics, with the discussions about key points for the study. This paper also introduces current steps of our robotics research that attempt to connect perception with action as intelligently as possible.

I. INTRODUCTION

About fifteen years ago, hand eye system was one of the exciting research topics in artificial intelligence. Several AI groups concentrated large efforts to create their prototype intelligent robots by combining computer vision with computer controlled manipulator. The performance of early hand eye system was quite limited. Vision system could recognize only simple block world. Control algorithm of manipulator was also simple and sensory interaction was very primitive. Many researchers who succeeded in building up the prototype concluded that individual subsystem should be pursued further before designing new system. And, research efforts have been devoted to explore general aspects of vision, planning, and control. Since then, the attempt of system integration has been left behind in AI community.

Although early hand eye systems could carry out only simple jobs in very simple block world, its success presented promising perspectives to manufacturing industries. During the last decade, industries have concentrated their efforts to advance industrial robots, and they succeeded in introducing robots into various production lines. The progress of hardware implementation including microelectronics is remarkable. And now, industries are keen to implement intelligent capability into their robots.

Robotics is the study of intelligent connection of perception to action.[1] Although artificial intelligence involves many disciplines, it is the study of intelligence from the standpoints of computation. Thus, both robotics and AI can share common interests. However currently, the author feels there exists significant gap between them. That is, AI deals mainly abstracted world, while robotics concerns real world, and the interests of those approaches seem to pass each other. As robotics involves many difficult problems which would not be solved without AI approach, it is very important to build a bridge that connects AI to robotics. In this paper, the author will discuss about key problems for integrating several disciplines into an intelligent robot system. And, current steps of our robotics research will be introduced as an example for building a bridge that connects robotics to AI.

II. KEY PROBLEMS

Robot is a versatile intelligent system that can interact with real world through sensors and effectors. Generally, it is consisted of four major subsystems; perception subsystem, action subsystem, thinking subsystem, and user friendly interface. Perception subsystem recognizes situations of real environment by means of vision, force sensing, and touch sensing. Action subsystem changes environment situation by moving objects or walking around. Thinking subsystem makes a plan of robot behavior, monitors its execution, and evaluates the results. User interface provides communication channel between human operator and robot. During last decade, a large number of researches on vision, manipulation, and planning have been done, and the performance of each subsystem has progressed very much. However, research on system integration has been left behind. We must remind that more studies of system architecture of robot are required if we wish to make robots smarter.

I would like to stress the importance of system oriented research of robotics, because

intelligent interaction of perception with action in real world is the main concern of robotics. It opens very interesting and promising research field for not only robotics but also AI. General priciples of intelligence gained from AI research would help us to synthesize total robot system. However, the problems which have been dealt with in theoretical AI research are oversimplified when we consider its application to dealing with complex constraints in real world. On the other hand, researches on robot mechanisms, control and sensors have been focussed to advance physical performance, and lack semantic aspects of robot behavior. Although there exists such a gap between AI and robotics, attempts to connect them together will be needed to explore fertile research field where intelligent machines can behave in real world.

There are so many difficult problems to be studied for creating smarter robot system. Before entering into each discussion, we have to consider the level of complexity of real environment to be dealt by several disciplines. Considering current performance of each related field, I propose a world of simple electro-mechanical assembly. Object shapes are supposed to be generated from general blocks and cylinders. Objects may have holes and threaded holes. Screws, gears and wires are also included in the repertoire of objects.

Major problems to be studied are listed below with short comments and discussions.

A. Abstracted Definition of Manipurator System

So far, research interests on manipulator system have been focussed onto the theory of advanced dynamic control for fast and precise motion. On the other hand, AI program assumes manipulator as very simple static object mover that always succeed in precise motion. Control discipline prefers aspects of dynamics, and ignores semantic aspect. AI approach pays its attention on semantic aspects of action rather than dynamic control. Thus, the interests of the two approach pass each other. In order to bridge over the gap, it seems necessary to provide AI people with an abstracted framework of manipulator control system so as to encourage them to construct smarter control structure. For this purpose, the author proposes a reasonable sets of abstracted functions for manipulator system. The seven basic functions are:

put-arm-reference (position, orientation)
get-arrn-coordinate (position, orientation)
put-hand-opening (opening-width)
put-grasp-force (grasping-force)
get-hand-opening (opening-width)
get-wrist-force (force, moment)
get-touch-sensor (sensor-state)

Above seven functions are defined in abstracted coordinates system, and are free from actual mechanical configuration. They work as if they were the instruction sets for computer. The cycle time is supposed to be very short, equivalent to the sampling period of servo-mechanism. Each function completes its operation in a few mili second. For instance, if we trigger put-arm-reference with the arguments about reference position and orientation in 3D world coordinates, then this function calculates corresponding 6 joint angles and sends the data to joint servo within single sampling period.

It is also necessary to provide a flexible means to combine the functions together. For such framework, I propose a fast real time operating system with concurrent process execution capability. If we connect above mentioned function sets by means of the concurrent real time OS commands such as start-process, stop-process, do-function, signal, wait, delay, and so on, we can describe any sensor based operations of manipulator.[2] I would like to encourage AI community to describe semantic aspects of manipulation on the above mentioned abstraction.

B. Theory of Manipulation

Sensor based robot programs are very difficult to write in general form. So far, several basic robot operations such as pin-into-hole, block-in-corner and rope-into-ring have been demonstrated. However, most of those program synthesis are based on ad hoc strategies, where geometric structures of parts relationship are assumed a priori, and lack generality. Lozano-Perez et al pointed out that small changes in the geometry of parts can have significant impact *on* fine-motion strategies, and they proposed a formal approach to the automatic synthesis of a class of compliant fine-motion strategies. Their approach uses geometric descriptions of parts and estimates of measurement and motion errors to produce fine-motion strategies. [3] Mason presented a theoretical explanation of the mechanics of pushing and demonstrated application of the theory to the analysis and synthesis of manipulator operations.[4] Those approach open a quite important novel field for future robotics research. More efforts should be devoted to the general study of manipulation strategy, in order to explore theoretical foundations for the program synthesis of complex sensor based manipulation.

C. Control of Visual Attention.

Although we are not conscious in every instance, we employ so many kinds of visual attentions during task execution, to find objects, to monitor situation changes of environment, to guide the motion of object, and to verify preconditions as well as postconditions for every piece of action. If we can provide a robot with fast and flexible capability of such visual attentions, the level of intelligent interaction of robot with real world will evolve dramatically. However, neither AI oriented vision research nor sensor based robotics research can tell very little about this problem. Can we formulate basic repertoire of visual attentions? What is the good framework to relate visual attentions to actions?

How should we synthesize the control strategies of visual attentions? Those questions seem to overlap the previous discussions on theory of manipulation. General study on visual attentions and their control strategy would provide one of the unexplored rich research field for connecting perception with action in intelligent way.

D. Automated Planning in Robot Language.

The main purpose of high level robot language is to simplify the programming of complex robot tasks. We can classify robot language into three levels such as manipulation oriented, parts oriented, and goal oriented. Like AL, manipulation oriented language requires to describe? robot tasks in explicit manipulation procedures.[8] In parts oriented language, the task is described as the state transition sequence of geometrical relationship between parts. RAPT is an example of this kind of language. Tt does not require explicit description of coordinate data of object motion in source code, instead, it attempts to solve those data from the descriptions about the logical constraint expressions among surfaces and axes of objects.[5] Goal oriented robot language requires only the goal description in source code, and generates action sequence by means of automated planning technique. In order to discriminate parts oriented language from goal oriented language, we assume the former takes complete state transition sequence that covers all subgoals so as to avoid the use of problem solver. From the viewpoint of programming simplicity, the goal oriented language is the ultimate goal of robot language development. But, the performance of problem solving today is too primitive to cope with complex robot environment. Tt deals very simple state description like (ON A B), while actual robot environment needs more complex state descriptions as RAPT shows. Moreover, object structures such as ARCH or TOWER which are dealt by current problem solver are far from actual assembly environment in its complexity. The author hopes AT will promote researches on powerful problem solver that can deal with complex real world constraints.

E. Sensor Based Environment Model Management.

Presumably, the environment model management system would be a core of an integrated intelligent robot system. It manages accumulated data base about the description of current environment situation, conceptual definitions of various structures and objects to be handled, and general strategies for primitive operations. Studies on robot language tell us the environment model management plays very important role in compiling manipulation procedures. Every piece of manipulation changes the environment situation in some extent. Therefore, when language processor compiles source codes of robot tasks, it must simulate situation changes in order to generate right codes with right data. The higher the performance of model management is, the easier the programming becomes. Tf no model management is provided, we are forced to write complicated task program keeping all the details of situation changes in mind.

If the initial state of environment is not known, the scene analysis program would be invoked to recognize the real world and the results are returned to the model. During the execution phase of planned action sequence, the execution monitor can know the expected situation changes in advance, and compares it with actual environment changes. Doing the motion under inconsistent circumstance between real world and its computer model sometimes causes a serious damage. In order to avoid such disaster, the consistency between the two should be checked all the time by means of sensors. Generally, the capacity to recover from action errors reflects the level of intelligence of robots. Suppose a situation where a robot drops object during the move and the object hits a tower under construction. Recovery from this accident needs the following procedure. First, robot must aware the happening by vision. Next, the damaged situation is recognized by scene analyzer, which is considered as the initial state. The goal state can be obtained from the model as the state description before that accident. Problem solver plans an action sequence that converts the initial state to the goal state. It is really difficult to implement such a robot system even for simple block world, however this example covers most of the robot component which should be equipped in future robots with high intelligence. Model management system is related to many aspects of robotics such as scene analysis, robot language, automated planning, motion execution, and control of visual attention. Finding a common rich framework that can be employed compatibly in those aspects would be the most important problem.

III. COSMOS: AN EXAMPLE OF INTEGRATED SYSTEM

This chapter introduces COSMOS, an example of interactive programming environment for the study of intelligent connection of perception to action in real world.[61 COSMOS is, in another word, a Lisp system that can interact with real environment by means of vision and manipulation.

Figure 1 shows hardware organization of COSMOS. We have two arms. One is a small universal arm driven by seven DC motors. Single board microcomputer which includes servo routine, communication routine, and diagnostic routine controls this arm. Trajectory data and control parameters are supplied from host minicomputer through GPIB interface. Another arm is a conventional 6 axes industrial robot of DC motor drive. As this arm can handle large pay load, it is used to move heavy experimental attachments such as TV camera or multiple axes wrist mechanism. Simple touch sensors and 6 axes force sensor are also interfaced to the system. As visual input device, we employed precision TV camera. Video

signal is digitized and stored into image frame memory device consisting of 768X512 pixels with 8 bits intensity scale. Host computer system consists of a 32 bits minicomputer, Data General ECLIPSE MV/4000 with 2 MB memory, and a 16 bit minicomputer, ECLIPSE S/140.

Figure 2 shows software configuration of COSMOS. Currently, nine software modules are imbedded into top level Lisp programming environment. Brief summary of each module is described below.

Top Level Lisp is the core of COSMOS system. Until 1983, we used Eclisp interpreter which was implemented on ECLIPSE S/140. Eclisp had a serious address space limitation caused by the 16 bit architecture of the S/140 processor. In order to solve this limitation, we decomposed a large program into several processes and connected them together by means of the inter process communication facility of AOS (Advanced Operating System of ECLIPSE). In 1983, we updated our host computer to a 32 bit virtual address machine ECLIPSE MV/4000, and Lisp system is also updated to Lisp/MV and its successor ETALisp, both of which are originally implemented in C on VAX at Electrotechnical Laboratory.[7] Eclisp, Lisp/MV and ETAlisp are designed upward compatible, so, all the early programs implemented on Eclisp still run on our current Lisp system.

Manipulation system has been built up as a hierachy of Task Level Robot Language, AL/L Language, Trajectory Calculator, Arm Control System, and Servo Controller, from top to bottom. AL/L is a manipulation oriented language. The syntax of AL/L is similar to AL which is developed at Stanford, but it is implemented in Lisp.[8] The

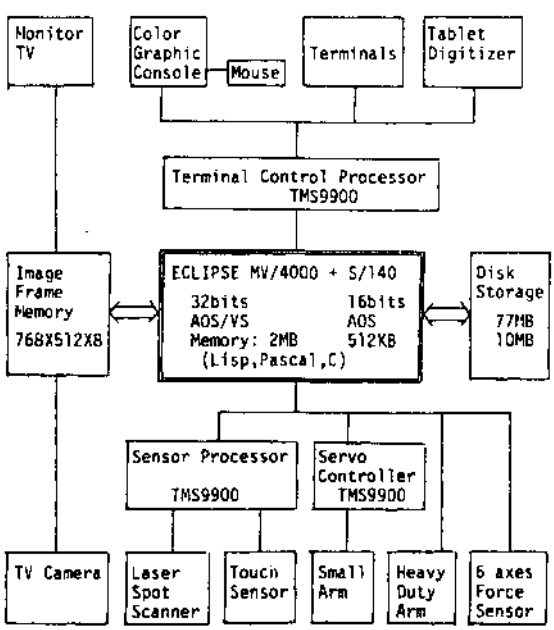

Fig.1 Hardware Configuration of COSMOS

model management part of AL/L is written in FRL to explore AI oriented study of robot language.[9] We have also attempted to extend robot language into parts oriented and task oriented level. In order to describe stractural constraints of objects, we adopted RAPT like representation. Providing conceptual description of goal structure such as arch or tower in RAPT like notation, and employing a simple problem solver, our task oriented language plans action sequence to perform a given goal like (BUILD ARCH), and generates the AL/L source code for the task.[10]

Vision system is conventionally devided into two modules; Vision Primitives, and 3D Vision. The former includes a set of primitive image processing functions such as data aquisition, image operators, feature extracters, planer logic operators, and display functions. The latter covers recognition level programs. Robot vision requires three dimensional recognition. We are mainly studying on stereo vision that analyses binocular images. As AI oriented vision research, we are exploring two competitive approaches for scene analysis. One is a FRL based line finder, in which frames control visual recognition process and finds right line drawings for simple blocks.[11] Another attempt employs production system for simple scene analysis.[12]

COSMOS is an open purpose project. We are just climbing up step by step towards smarter intelligent robot system. When I initiated this project in 1978, I intend to create general purpose research tool for exploring intelligent connection of perception to action. At first, we implemented Lisp interpreter. Then, we developped AL/L robot language onto the Lisp. Next, we developped general purpose robot arm and interfaced it to AL/L. Almost at same time, we added primitive vision facility to our Lisp system. Thus, we succeeded in constructing simple hand eye system within Lisp environment. It actually provided us very convenient research environment for intelligent robotics. After a system was integrated, we enjoyed COSMOS very much, updated it, and accumulated new features of robotics, year by year. We do not think that COSMOS configuration is the best solution. Rather, we consider that COSMOS reflects our research history and future direction in computer executable form.

IV. HAND EYE EXPERIMENT ON COSMOS

This chapter introduces one example of hand eye experiment which is performed on COSMOS.[13] The experiment is on rope handling, which is a difficult task to do without visual feedback, because a flexible rope cannot hold a well defined shape.

As explained earlier, full COSMOS is a toolbox which covers all our robotics software. It is really convenient, but sometimes it is heavy for a particular experiment. In order to improve

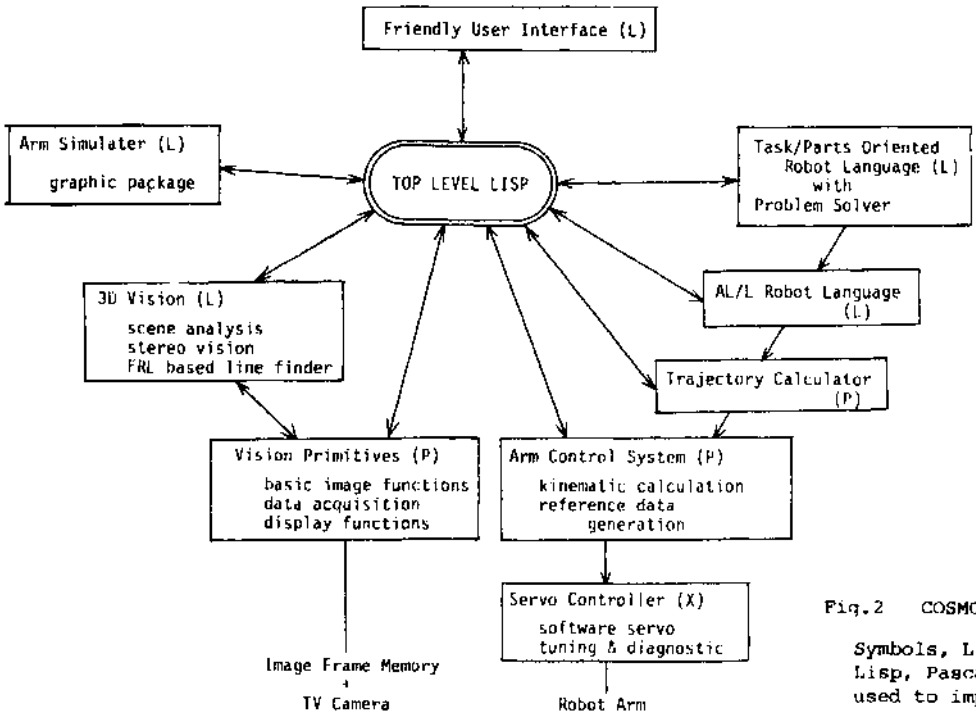

Fig.2 COSMOS Software

Symbols, L,P,X stand for Lisp, Pascal, Cross Assembler used to implement each module.

run time efficiency, COSMOS software is rearranged and tuned for the rope handling experiment. The tuned system consists of Top Level Lisp, Vision subsystem, Arm subsystem, and Hand Eye Calibration. Vision subsystem involves three modules; EYE, WATCH, and STEREO.

Top Level Lisp: Top level Lisp allows us interactive hand eye programming. Command sets are tuned for this experiment. Robot Language AL/L is not employed. Instead, adequate arm commands which directly manages arm control system are built in. All the following vision modules can be invoked by Lisp commands.

EYE: This module provides low level vision primitives, such as image operators, feature extracters, logic operators, and display functions. Usually, vision functions are applied on the window area of 64X64 pixels. This module is used for rough, global analysis of a scene.

WATCH: In order to perform fine local analysis efficiently, scan-line based image processing is employed. When we use linear scan, the position, direction, and length of the scan-line are controlled. When we use circular scan, the center and diameter of scan circle are controlled. Zero crossing operator along those scan extracts precise edge point on the scan line. This module provides a simple and efficient visual recognition for feedback.

STEREO: In order to obtain stereo image, we attached mirror adapter onto TV camera. It composes binocular image onto single TV frame. Left half of the frame corresponds to left eye image, and right half of the frame corresponds to right eye image. Both images are analysed separatedly, and the coordinates of corresponding points on both images are calculated. Then, three dimensional position of the point is calculated by the principle of triangulation.

Hand Eye Calibration: In order to mate manipulator system with vision system together, we must know precise numerical relationship between the two coordinate systems. It is called hand-eye calibration. This module calculates precise trnsformation matrix between the two coordinates by means of vision facility automatically.

The experiment includes a task of rope-into-ring and tying a knot around the ring. The scenario is as follows. At first, robot finds a rope by vision and grasps it. Next, robot finds a ring and puts the rope into ring by visual feedback. Then, robot release the rope, regrasps the rope from opposite side of the ring, and pull the rope out. Finally robot ties a knot around the ring. Figure 3 shows flow diagram of the experiment. The block number in circle indicates vision processing, while the block number in square indicates manipulation. The experiment itself will be shown by movie.

v. VISION SYSTEM WITH MULTIPLE ATTENTIONS

The use of visual monitoring, visual feedback, and visual verification is necessary in execution phase of action plan. They accomplish various interaction between vision and action in real time. In most cases, above mentioned vision program can be constructed so as to focus its attention onto local regions in which the existence of key features are expected. Usually, points of attention should be multiple, although processing for each attention is simple. As the processing time of such visual interaction is very critical for real time use, those multiple attentions must run in parallel. In order to realize such kind of visual interaction in real time, we designed and implemented new versatile multi window vision processor. A rectangular local region to be processed is referred to as a window. In our system, the location, shape, and resolution of each window can be independently controlled by hardware.

Figure 4 shows a block diagram of multi window vision system. It consists of one digitizer/transmitter unit and a large number of window processing units. Digitizer/transmitter unit converts NTSC composite color signal into digital video signals and broadcasts them onto the video bus. The signals *on* the video bus are R, G, B, and Y, each of which has 8 bits intensity scale. During the video scan, pixel address increments in video scanning rate within 320X240 pixel area. In order to inform the current scanning position, the horizontal and vertical pixel address are also broadcasted onto the video bus. By observing the current pixel address, each window unit picks up only the necessary data inside the designated window area, and stores them

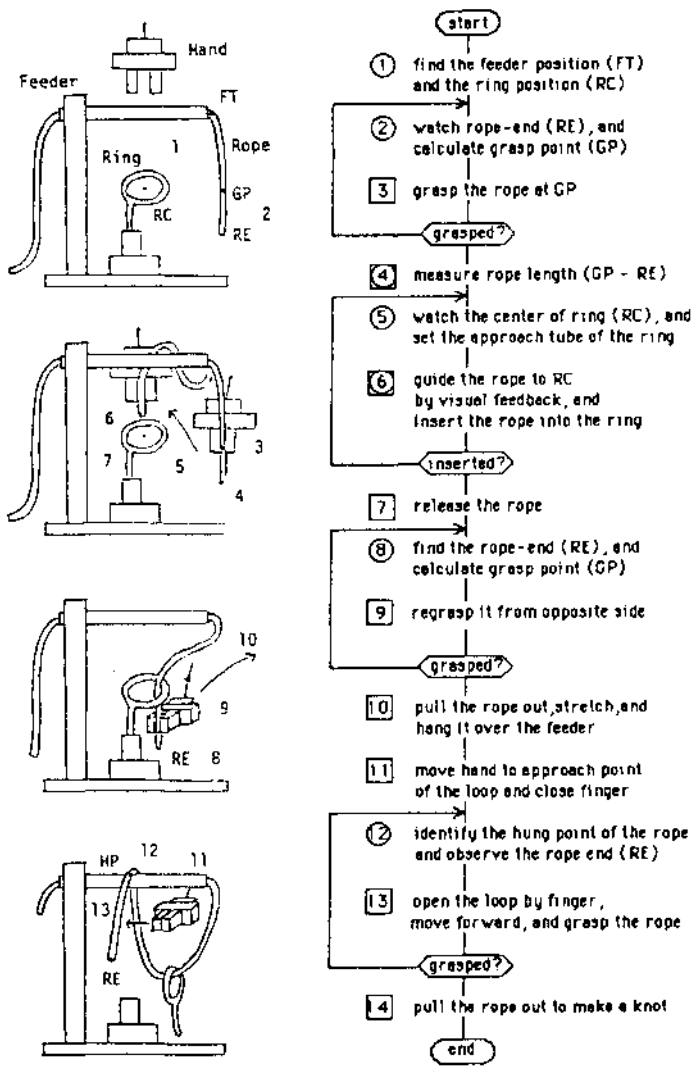

Fig.3 Flow Diagram for Rope Handling

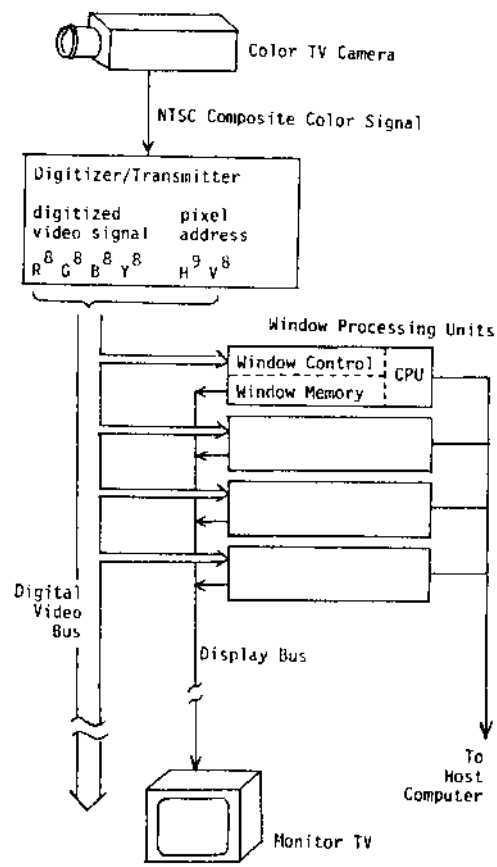

Fig.4 Block Diagram of Multi Window Vision System

in its window memory, which is processed by a microcomputer independently.

The signal on the video bus is one way, and read only. So, a large number of window unit can aquire the window data independently and simultaneously. In our current implementation, window memory has only 4 KB. It corresponds to 64X64 pixels if it is used as square window. Memory size is small, however its usage is very flexible. Firstly, the location of each window is controllable. So, we can locate each window anywhere inside TV screen. Secondly, the aspect ratio of each window is programmable. So, we can change window shape into square, rectangular, and linear. Thirdly, spatial data resolution, in another word, pixel sampling rate is variable. We can read every pixel data, one pixel data from every 2X2 neighbors, from every 4X4 neighbors, or from every 8X8 neighbors. If we use low resolution mode, we can obtain rough data over large window area. If we need precise data, we use high resolution mode for small window area. Finally, we have four kinds of input signals such as red, green, blue, and brightness. We can choose one signal from the four.

The prototype model has sixteen windows. Four window memories are processed by single Motorola 68000. And four such processor units are connected to host computer. Image processing programs which run on window processor are written in C. They are compatible to vision primitives module of COSMOS. Host machine is also Motorola 68000, on which Lisp interpreter runs. Thus, recognition level program written in Lisp controls individual window processing, following the same philosophy of COSMOS design.

When we designed multi window vision system, we expected to operate one window as one visual demon. Therefore, multi window vision system provides very flexible hardware for pandemonium models for visual recognition. We plan to control a large number of visual demons in knowledge invoked way in order to explore sensor based environment model management system for robot. In such application, we need a large number of window hardwares. Therefore we attempted to develop special LSI. We have already succeeded in developping LSI chip for window data acquisition. We are now developping large scale multi window vision system by using that LSI.

VI. CONCLUDING REMARKS

Research on intelligent robot was born in Artificial Intelligence more than twenty years ago. However, after the development of early hand eye systems, very few efforts have been continued on system oriented research of intelligent machines that behave in real world. The author believes that many key problems for intelligent robot could not be solved without merging AI approach with robotics. Therefore, in order to encourage AI community to pay more attentions to intelligent robotics, several key points are discussed from the viewpoint of system oriented approach. Then, an attempt to build a bridge that connects AI to robotics is presented with an example of COSMOS experimental system. So far, we have concentrated our efforts onto building a Lisp programming environment that has manipulators, vision systems, robot language and other sensors. Using this total system, we are going to explore a method to connect perception with action as intelligently as possible.

Acknowledgement: COSMOS is developped at Information Systems Laboratory, in the Department of Mechanical Engineering, the University of Tokyo. The author wishes to express deep appreciations to the past and present students who contributed to implement COSMOS: T.Ogasawara, O.Naito, O.Shiroshita, T.Matsui, H.Mizoguchi, S.Kawagoe, M.Inaba, H.Matsubara, M.Fujita, A.Okano, Y.Tsusaka, Y.Murata, T.Fukuizumi, K.Kado.

REFERENCES

1 Brady,M. and R.Paul(Eds), Robotics Research, MIT Press (1984)
2 Inoue,H., "Robot System", in Mechatoronics, Iwanami-Syoten (1985) pp85~154 (in Japanese)
3 Lozano-Perez,T., M.Mason, and R.Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots", Robotics Research 3:1 (1984) 3-24.
4 Mason,M., "Mechanics of Pushing", Proc. of 2nd International Symposium of Robotics Research, Kyoto (1984) pp73~80.
5 Popplestone,R., P.Ambler, and I.Bellos, "An Interpreter for a Language for Describing Assemblies", Artificial Intelligence 14:1(1980) 79-107.
6 Ogasawara,T. and H.Inoue, "COSMOS: A Total Programming System for Integrated Intelligent Robot Study", J. of Robot Society of Japan, 2:6 (1984) 507-525 (in Japanese).
7 Ogasawara,T. and T.Matsui, ETALisp User's Manual, Electrotechnical Lab. (1984).
0 Mujtaba,S. and R. Goldman, AL User's Manual, Stanford AI Lab. (1979)
9 Inoue,H. et al, "Design and Implementation of High Level Robot Language", Proc. of 11th ISIR (1981) pp675-682.
10 Matsubara,H., A. Okano and H.Inoue, "Design and Implementation of Task Oriented Robot Language", J. of Robot Society of Japan, 3:3 (1985) (in Japanese).
11 Inaba,M., Research on Robot System with Vision, Master's Thesis, Tokyo Univ., (1983) (in Japanese)
12 Matsubara,H., Scene Analysis performed by Production System, Master's Thesis, Tokyo Univ. (1983) (in Japanese).
13 Inoue,H. and M.Inaba, "Hand Eye Coordination in Rope Handling", in (1) ppl63-174.
14 Inoue,H. and H.Mizoguchi, "A Flexible Multi Window System for Robots", Proc. of 2nd ISRR, Kyoto (1984) pp42-49.