June 1988

Redundant Multi-Modal Integration of Machine Vision and Programmable Mechanical Manipulation for Scene Segmentation

Constantine J. Tsikos
University of Pennsylvania

Ruzena Bajcsy
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-41.

Redundant Multi-Modal Integration of Machine Vision and Programmable Mechanical Manipulation for Scene Segmentation

Abstract

The main idea in this paper is that one cannot discern the part-whole relationship of three-dimensional objects in a passive mode without a great deal of *a priori* information. Perceptual activity is exploratory, probing and searching. Physical scene segmentation is the first step in active perception. The task of perception is greatly simplified if one has to deal with only one object at a time. This work adapts the non-deterministic Turing machine model and develops strategies to control the interaction between sensors and actions for physical segmentation. Scene segmentation is formulated in graph theoretic terms as a graph generation/decomposition problem. The isomorphism between manipulation actions and graph decomposition operations is defined. The non-contact sensors generate the directed graphs representing the spatial relations among surface regions. The manipulator decomposes these graphs under contact sensor supervision. Assuming a finite number of sensors and actions and a goal state, that is reachable and measurable with the available sensors, the control strategies converge. This was experimentally verified in a real, noisy, and dynamic environment.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-41.

REDUNDANT MULTI-MODAL INTEGRATION OF MACHINE VISION AND PROGRAMMABLE MECHANICAL MINPULATION FOR SCENE SEGMENTATION

Constantine J. Tsikos
Ruzena K. Bajcsy

MS-CIS-88-41
GRASP LAB 144

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

June 1988

Acknowledgements: This research was supported in part by U.S. Postal Service grant 104230-87-M-0195, NSF-CER grant MCS-8219196, U.S. Army grants DAA29-84-K-0061, DAA29-84-9-0027, LORD Corporation and IMB Corporation grants.

REDUNDANT MULTI-MODAL INTEGRATION OF MACHINE VISION AND PROGRAMMABLE MECHANICAL MANIPULATION FOR SCENE SEGMENTATION

Constantine J. Tsikos and Ruzena K. Bajcsy

Computer and Information Science Department
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104-6389

ABSTRACT

The main idea in this paper is that one cannot discern the part-whole relationship of three-dimensional objects in a passive mode without a great deal of a priori information. Perceptual activity is exploratory, probing and searching. Physical scene segmentation is the first step in active perception. The task of perception is greatly simplified if one has to deal with only one object at a time.

This work adapts the non-deterministic Turing machine model and develops strategies to control the interaction between sensors and actions for physical segmentation. Scene segmentation is formulated in graph theoretic terms as a graph generation/decomposition problem. The isomorphism between manipulation actions and graph decomposition operations is defined. The non-contact sensors generate the directed graphs representing the spatial relations among surface regions. The manipulator decomposes these graphs under contact sensor supervision. Assuming a finite number of sensors and actions and a goal state, that is reachable and measurable with the available sensors, the control strategies converge. This was experimentally verified in a real, noisy, and dynamic environment.

INTRODUCTION

In the past, [BAJCSY 1985], [BAJCSY/TSIKOS 87], [TSIKOS 87] argued for active sensing as opposed to the traditional static analysis of passively sampled data. In the robotics and computer vision literature, the term active sensor, generally refers to a sensor that transmits radiation, (e.g., sonar, radar, lidar, etc.) into the environment and measures the returned signals.

The use of active sensors is not a necessary condition for active sensing. Active sensing can be performed with passive sensors (that only receive and do not transmit information), employed actively. The term "active" is used not to denote a time-of-flight type sensor, but to denote a sensor employed in an active fashion, purposefully changing the sensor's state parameters according to sensing strategies. Hence the problem of active sensing can be stated as a problem of intelligent control strategies applied to data acquisition process which will depend on the current state of the data interpretation including recognition.

In this paper we shall concentrate on the task of scene segmentation within the framework of active perception. The main issue in active perception is control and the interaction between sensors and actions. For that we must identify the initial state, the state space, the transition functions, and finally the goal state. The scene varies not only initially but also during the segmentation process. This leads to a non-deterministic system. Another major issue is the identification of the goal state of the perceptual system. The goal state must be not only reachable but also measurable with the available sensors. Examples of such goal states may be: an empty scene, an ordered set of objects, a segmented set of N objects, successful matching of an object with an expected shape, and so on.

BACKGROUND

During our literature search it became obvious that most of the research on object description makes the following assumptions: a) Objects are rigid, i.e. made from solid materials. b) Objects have non-flexible parts. c) If two or more objects are attached to each other, the recognition whether they are

one or more objects is guided by a priori information of the shape or size of the object. Research has been concentrating more on the assembly process than the disassembly process, yet to understand the structural composition of an object, unless a priori given, one needs to decompose it. The closest to our thinking has come [YAMADA et. al. 87] in building an expert system which can generate all possible procedures of disassembling objects from 3-D models.

GOAL

Our ultimate goal is disassembly in a non constrained environment. We believe that passive perception is not sufficient to discern whether objects in a scene are mechanically bound, rigidly, flexibly, or not bound at all. The binding force is that force which holds two or more objects together. The binding can be rigid (like glue), flexible (like a hinge, or spring), or non existent (objects form touching or overlapping arrangements held together by gravity and friction). Physical scene segmentation should be the first step in disassembly. The task of perception is greatly simplified if one has to deal with only one object at a time. We must first eliminate the "obvious" forces (gravity and friction) that keep a collection of objects together, before we proceed to find other forces. Our immediate goal is to to develop a model of sensing-action-interaction and use this model for segmentation of random object arrangements.

ASSUMPTIONS

We assume a finite number of sensors and their processing modules. Currently we consider a range sensor, force/torque and other contact sensors. Additional sensors such as: stereo, texture, tactile, proximity, etc., can be incrementally added. We assume a finite number of actions. Currently we consider acquisition (grasping/picking), local displacement (pushing), and global displacement (shaking). This list can be expanded to include actions such as: compliant move, turn, etc. We assume that a random arrangement of objects can be separated by the manipulation actions of pick, push, and shake.

METHODOLOGY

Our methodology of segmentation is based on graph-theoretic operations. The segmentation problem is formulated as a graph generation/decomposition problem. The sensors are used as the graph generator, and the manipulator as the decomposing mechanism of this graph. An isomorphism exists between the manipulation actions and graph decomposition operations [TSIKOS 87]. Our approach is to close the loop between sensing and manipulation. The manipulator is used to simplify the scene by decomposing the scene into visually simpler scenes. The manipulator carries the contact sensors to the region of interest and performs the necessary exploratory movements that will determine the nature of the mechanical binding between objects in the region.

THE MODEL OF SENSING, MANIPULATION, AND CONTROL

The model of sensing, manipulation and control is a Non-Deterministic Turing Machine (NDTM). See Fig. 1. The physical world (scene) is the "tape" of the machine, the "read_from_tape" actions are the sensing actions, and the "write to_tape"actions are the manipulation actions. The model is a Turing machine because actions constantly change the physical environment (tape) and therefore its own input. The control automaton of the NDTM is non- deterministic because of the non-determinism of the sensors and the actions. The actions and strategies are modeled as non-deterministic, finite state automata. There are many advantages in using this well-known model.

The first advantage, [ALBUS et. al. 82], is that the sense-compute-act formalism allows the control problem to be partitioned in time and complexity. At any given time, the system deals only with present state and present input, produces an output which is a function of current state and current input and moves to a new state. Current state encodes information about past history of states and actions of the machine and its environment. Current sensory input is not deterministic (noise in sensory data). The next state of the NDTM is not deterministic because the machine modifies its tape via actions whose outcome cannot be known a priori (push and shake actions).

The second advantage is that the theoretical tools needed to prove correctness of the machine's behavior have long been established and tested. Path sensitization and graph de-cyclization algorithms exist, [HARTMANIS/ STERNS 66], [KOHAVI 70], [DEO 74], to prove that, the goal state is reachable, and the state transition diagram does not contain deadlock states, or cycles.

The third advantage is that it facilitates error handling. If additional states need to be defined to deal with non-anticipated error conditions, then these states can be simply inserted. The fourth advantage is that is modular and allows insertion of new sensors, actions and feedback conditions. The fifth advantage is that it makes debugging easy. The sixth advantage is that it allows a system to be developed incrementally.

One disadvantage is that the number of states and transitions needed to represent the machine and its environment increases as more sensors are added. Addition of more sensors implies increased complexity.

Definition: An NDFSA, is a quadruple (I, O, S, T) where:

I: Inputs from a variety of contact and non-contact sensors, such as:
Vision (range, stereo), force/torque, gripper distance, vacuum, etc.

O: Outputs = Actions, such as: Shake, Push, Pick, Look, Stop, etc.

S: States = Set of states, such as: manipulator states, gripper states, sensor states, and states of the environment (as perceived by the sensors).

T: State Transition Function, (I x Sc) —-> Sn,
where, the next state Sn is a function of current state Sc and current input I.

INPUTS (Sensory Inputs)

The non-contact sensory input to the NDTM is vision (in the form of

range images). The scene is segmented into what appears to be spatially-connected surface regions. For each region, we compute the position (X, Y, Z) of the center of gravity, the orientation (O, A, T) of the surface normal at the center of gravity, an estimate of size (L, W, H) of the smallest parallelepiped bounding the region, and an estimate of the maximum curvature (C). From these measurements, the objects are initially classified into one of three generic shapes such as: flat, box, and tube/roll.

The On-Top-Of relation between all pairs of visible regions in the scene is computed and the directed graph representing this relation is constructed. Vertices represent visible, connected, surface regions. Directed edges represent the spatial relations between the vertices. See Fig. 2, 3, 4, and 5.

Top-most surface segments are important in physical scene segmentation because they may belong to top-most objects in the scene. Top-most objects are important because they usually have more surfaces exposed (more ways to be grasped). The forces required to extract them from the scene are less and therefore the chances of loosing positional information after the object is being grasped are minimized. Furthermore, manipulating the top-most object keeps scene disturbances to a minimum.

A partially dispersed scene corresponds to a disconnected digraph. An efficient algorithm based on "fusion" of adjacent vertices is given in [DEO 74]. A totally dispersed scene, (as well as a singulated scene), correspond to a null graph (a graph with vertices and no edges). Efficient graph theoretic algorithms exist (testing the digraph's adjacency matrix for all zero entries) for singulation verification. Finding the top-most objects in the scene corresponds to topological sorting of the digraph.

Visual information may be sufficient to accurately describe simple objects and non-overlapping scenes. However, it is not sufficient to distinguish between overlaps caused by two different objects in the scene and overlaps caused by a single, self-occluding object. For example, a thin flat object supported by and totally occluding a smaller box-shaped object can be mistaken as a large box-shaped object. Therefore, machine vision alone is not enough.

We use additional information from contact sensors. Some of the contact sensors are: Two force/torque sensors (mounted on the gripper jaws) are used in closed loop feedback during manipulation. Force feedback is used to provide force servoing to the gripper, to sense collisions, to measure the weight of objects, and to determine if an object or tool is properly grasped. A finger position sensor is used in a closed-loop feedback manner during manipulation. Position feedback is used to provide basic position servoing

to a gripper, and to refine size estimates of objects (computed from vision). A vacuum sensor is used to verify proper grasp, to differentiate between small size, non- penetrating cavities, from holes which penetrate an object. Vision and tactile sensors have difficulties in reaching into confined places. Surface porosity can be measured using a vacuum sensor. Porosity is very difficult to measure using either vision, force/torque or tactile data.

OUTPUTS (Actions)

The manipulation actions are composed hierarchically from simpler actions. The hierarchy of actions is in terms of composition of complex actions from simpler actions and does not apply to the execution of these actions. The hierarchy of action composition is given in [TSIKOS 87]. Some manipulation actions are modeled as deterministic finite state automata (FSA), while others are modeled as non-deterministic, finite state automata (NDFSA). The lowest level in the hierarchy of actions consists of very simple actions. These actions are used as state-to-state transitions in the construction of a more complex action automaton. This automaton is used as a transition (action) at the next higher level. An action at a given level is represented as an automaton at the next lower level. The advantages of hierarchical construction are modularity, testability, and incremental growth.

The manipulation actions are: Acquisition (pick), Local displacement (push), and Global displacement (shake). The pick action is used to break the vertex connectivity of the digraph by removing vertices. Several tools may be used to implement this action. An object may be picked and removed from the scene using the gripper, or it may be picked by selecting a tool (i.e. suction tool). The push action is used to break the edge connectivity of the digraph representing the on-top-of relation. Several tools may be used to implement this action. An object may be pushed using the gripper, or it may be pushed by selecting a push tool (such as a spatula or the suction tool). Complete planning of the push actions is very complicated, [LOZANO-PEREZ 80], [LOZANO- PEREZ 81], [MASON 82], [MASON 86], and requires knowledge of the friction coefficients of all objects in the scene as well as knowledge of the spatial relations of all objects in the scene to decide where and how far to push.

STATES

The states of the NDFSA controlling the Turing machine are partitioned into the following classes. Robot states, gripper states, sensor states, states describing the environment (as perceived by the sensors), and pathological states.

The robot states are: tool acquisition and release locations, object acquisition locations, object displacement locations, and object release locations. The gripper states are: gripper open holding nothing, gripper closed holding tool, gripper closed holding object, and gripper closed holding nothing. The states of the environment as perceived by the sensors are: Empty, Dispersed, Overlapped, Ambiguous, and Unstable. This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of the above five states can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations.

A scene is EMPTY if the digraph of the on_top_of relation is an empty digraph. (I.e. a graph with no vertices, and no edges). A scene is DISPERSED if the digraph is null. (I.e. a graph with vertices and no edges). See Fig. 2. A scene is OVERLAPPED if the digraph contains no directed cycles. See Fig. 3. A scene is AMBIGUOUS, if the digraph contains directed cycles. See Fig. 4. A scene is UNSTABLE, if it is either overlapped, or ambiguous and contains vertices labeled "Tubes/Rolls". See Fig. 5.

For the immediate goal of scene segmentation the goal state is the EMPTY state. This state must be not only reachable but also measurable with the current sensors. In other words, for the machine to halt the system must have sensors to sense that the goal state has been entered. In this work the empty state is both reachable and measurable (all range values are zero, i.e. no surface segments and therefore no objects exist in the scene).

The pathological states are listed in order of severity (most severe first). For more details see [TSIKOS 87]. The pathological states are: Sensor damaged, Unable to get tool, Tool and object lost, Lost tool, Lost object above the workspace, Lost object away from the work space, Unable reach object, Unable to Pick, and Unable to Push. As more sensors and actions are added into the system, more pathological states must be defined. A

finite number of sensors and a finite number of actions results in a finite number of pathological states.

SEGMENTATION STRATEGIES

The control structure of the NDTM is a NDFSA. The input to the automaton is the digraph of the on-top-of relations of surface regions in the scene generated by the vision system. It is important to emphasize that this digraph represents relations of only the visible surface segments and not the physical objects that these surface segments belong to. The physical on-top-of and other spatial relations as well as the part-whole relations of the objects is not known.

A heap of objects is defined in graph theoretic terms. The control problem is transformed into the problem of topological sorting of object arrangements. The manipulation actions of object acquisition (pick) and local displacement (push) are defined as decomposition operations on digraphs representing the on-top-of relation of objects in the arrangement. Upper and lower bounds on the number of actions needed for graph decomposition are established, see [TSIKOS 87].

The strategies are modeled as NDFSAs. The states are scene descriptions generated by the vision system. The transitions are manipulation, sensing, and error recovery actions. A strategy is generated by applying the following rules: 1) No action is allowed to be repeated on the same object, consecutively, more than x number of times. 2) Acquisition follows topological order. 3) Acquisition has priority over. 4) Non-graspable objects are displaced. 5) Ambiguous scenes are first displaced globally then locally. 6) Unstable scenes are displaced globally. 7) Partially visible objects are displaced. Displacement proceeds in an outermost first, order. 9) Flat objects are picked/pushed with a suction tool, applied at the center of gravity of the visible surface segment. 10) Box-shaped objects and tubes/rolls are picked either with the gripper or a suction tool and are pushed at the edges.

For physical scene segmentation, the NDTM is a very general model. This model is sufficient to describe every strategy for the following reasons:

1. There is a finite number of objects in any arrangement.

2. There is a finite number of internal states and a finite number of entropy reducing actions (given the manipulator, a finite set of sensors and tools).

3. There exists a measurable stopping criterion. I.e. there is a reachable goal state (empty scene, where entropy is zero).

We have developed four segmentation strategies, see Figures 6, 7, 8, 9. The control structure of strategy 1 is shown in Fig. 6. The strategy does not use local displacement (push). The general idea is to look, pick the top-most object, and look again. If the scene is ambiguous or unstable, it shakes the heap. If shaking fails, it continues with the pick action. This strategy is simple and very effective in dealing with scenes where all objects are graspable with the set of acquisition tools. The strategy eliminates ambiguities via the shake and pick actions. If the shake action fails to remove the ambiguity then non-topmost objects are picked up. This causes objects to be lost during acquisition. For the strategy to succeed the sensor thresholds must be raised to enable the system to tolerate higher torques caused by picking objects off the center of gravity. When the threshold is raised, the probability of tool losses increases as well as the probability of damaging the sensors. Therefore, the probability of entering the fatal error state is increased. If the weight of the objects is low, the probability of damaging the sensors (even if the system picks objects supporting other objects) is low, and the strategy converges. For a more comprehensive description of the strategies and proofs of convergence see [TSIKOS 87].

IMPLEMENTATION and EXPERIMENTAL RESULTS

The system block diagram is shown in Fig. 10. It consists of a range imaging system, a linear stage, a PUMA 560 robot, a LORD Corp. servoed instrumented gripper, a micro-VAX-II computer, a support structure, several tools, tool fixtures, and accessories.

All experiments run on the real system. No simulation results are reported. The domain was mostly objects found in the mail stream, such as: parcels, flats, tubes and rolls. A number of additional experiments were conducted with objects containing holes, cavities, and some porous objects. The heaps were created by stacking these objects at random to an average

of five object layers per heap. The weight of every object was under one pound. During all experiments the heap was observed to transform and to enter all five states: ambiguous, unstable, overlapped, dispersed, and empty.

EXPERIMENTS

The purpose of this group of experiments was to evaluate strategy 1. The strategy performed well on unstable, overlapped, and dispersed heaps. Difficulties were observed with ambiguous configurations. The shake action was not very effective in removing ambiguities. One reason is that the action was implemented using the linear stage in a vibration mode at maximum speed and acceleration. These speeds and accelerations were not enough to produce a significant change in the scene. Using the pick action to remove ambiguities resulted in an increased number of tool and object losses. The shake action failed to eliminate the ambiguities caused by configurations of flats. This is because flats form stable configurations. However, because flats are rather lightweight and flexible, it was possible to use the pick action to break-up cyclic object configurations without many tool or object losses. Strategy 1 failed to converge when the heap contained porous objects. Other experiments have been performed to evaluate Strategies 2, 3, and 4, see [TSIKOS,87].

CONCLUSIONS

We introduced the paradigm of iterative, interactive scene segmentation and simplification via vision, manipulation, force/torque and other sensory input. We developed a methodology of scene simplification based on graph theoretic operations. We formulated the scene simplification problem as a graph decomposition problem and defined the isomorphism between the pick and push manipulation actions and the graph decomposition operations of vertex and edge removal. We have shown that the sensors can be used as the partial graph generators, and the manipulator as the decomposing mechanism of this partial graph. We have modeled the actions and strategies as non-deterministic, finite state automata that decompose these graphs

under sensor supervision. We have proved that the strategies converge, we identified the pathological states and developed several error recovery actions.

We have integrated a vision system, a manipulator, force/torque and other sensory input into an experimental robot work cell and conducted experiments to test convergence, error recovery and graceful degradation of four different strategies. We have found that many of these strategies can recover from pathological states, tolerate errors in the sensory data, recover from un- successful actions, and converge. What we have learned during this work is:

1. In an unstructured environment, where there is uncertainty and incomplete information, a detailed, sophisticated plan is not enough. The plan must constantly change and adapt to the sensory input.

2. Redundancy of Actions (in addition to redundant sensors, tools, etc.) is needed for exploration of unstructured environments.

3. In the domain of heap segmentation, it is possible to reach the goal state via iteration and interaction of a few sensors, tools and a few simple, short-range manipulation actions.

ACKNOWLEDGEMENTS

This research was funded in part by the United States Postal Service, BOA Contract: 104230-87-H-0001/M-0195, DARPA/ONR grant N0014-85-K-0807, NSF grant DCR 8410771, Air Force grant AFOSR F49620-85-K-0018, Army grant DAAG-29-84-K- 0061, by DEC Corporation, IBM Corporation, and LORD Corporation.

REFERENCES

[ALBUS et al, 82]
J. Albus, A. Barbera and M. Fitzgerald, "Programming a Hierarchical Robot Control System", 12th International Conference on Industrial Robots, Paris, France, June 1982.

[BAJCSY, 85]
R. Bajcsy, "Active Perception vs Passive Perception", IEEE Computer Society Third Workshop on Computer Vision: Representation and Control, Bellaire, MI, October 13-16, 1985.

[Bajcsy / Tsikos, 88]
R. Bajcsy and C. Tsikos, "Perception via Manipulation", 4th International Symposium of Robotics Research, (R. Bolles and B. Roth editors), MIT Press, 1988.

[DEO 74]
N. Deo, "Graph Theory with Applications to Engineering and Computer Science", Prentice-Hall, 1974.

[HARTMANIS / STEARNS 66]
J. Hartmanis and R. Stearns, "Algebraic Structure Theory of Sequential Machines", Prentice-Hall, 1966

[KOHAVI 70]
Z. Kohavi, "Switching and Finite Automata Theory", McGraw-Hill, 1970.

[LOZANO-PEREZ 80]
T. Lozano-Perez, "Spatial Planning with Polyhedral Models", Ph.D. Dissertation MIT, Department of Electrical Engineering and Computer Science, June, 1980.

[LOZANO-PEREZ 81]
T. Lozano-Perez, "Automatic Planning of Manipulator Transfer Movements", IEEE Transactions on Systems, Man, Cybernetics, Vol. SCM-11, No. 10, Oct. 1981.

[MASON 82]
M. Mason, "Manipulator Grasping and Pushing Operations", Ph.D. Dissertation, MIT, Department of Electrical Engineering and Computer Science, June, 1982.

[MASON 86]
M. Mason, "Mechanics and Planning of Manipulator Pushing Operations", International Journal of Robotics Research 5, Spring 1986.

[TSIKOS 87]
C. J. Tsikos, "Segmentation of 3-D Scenes using Multi-Modal Interaction Between Machine Vision and Programmable, Mechanical Scene Manipulation", Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania, December 1987.

[TSIKOS / BAJCSY 88]
C. J. Tsikos and R. K. Bajcsy, "Physical Scene Segmentation via Vision and Manipulation", Proceedings of the 3rd Advanced Technology Conference, Washington, DC, May 3-5, 1988.

[YAMADA et al 87]
S. Yamada, N. Abe and S. Tsuji, "Construction of a Consulting System for Structural Description of Mechanical Objects", Proceeding of the IEEE International Conference on Robotics and Automation, Raleigh NC, 1987.

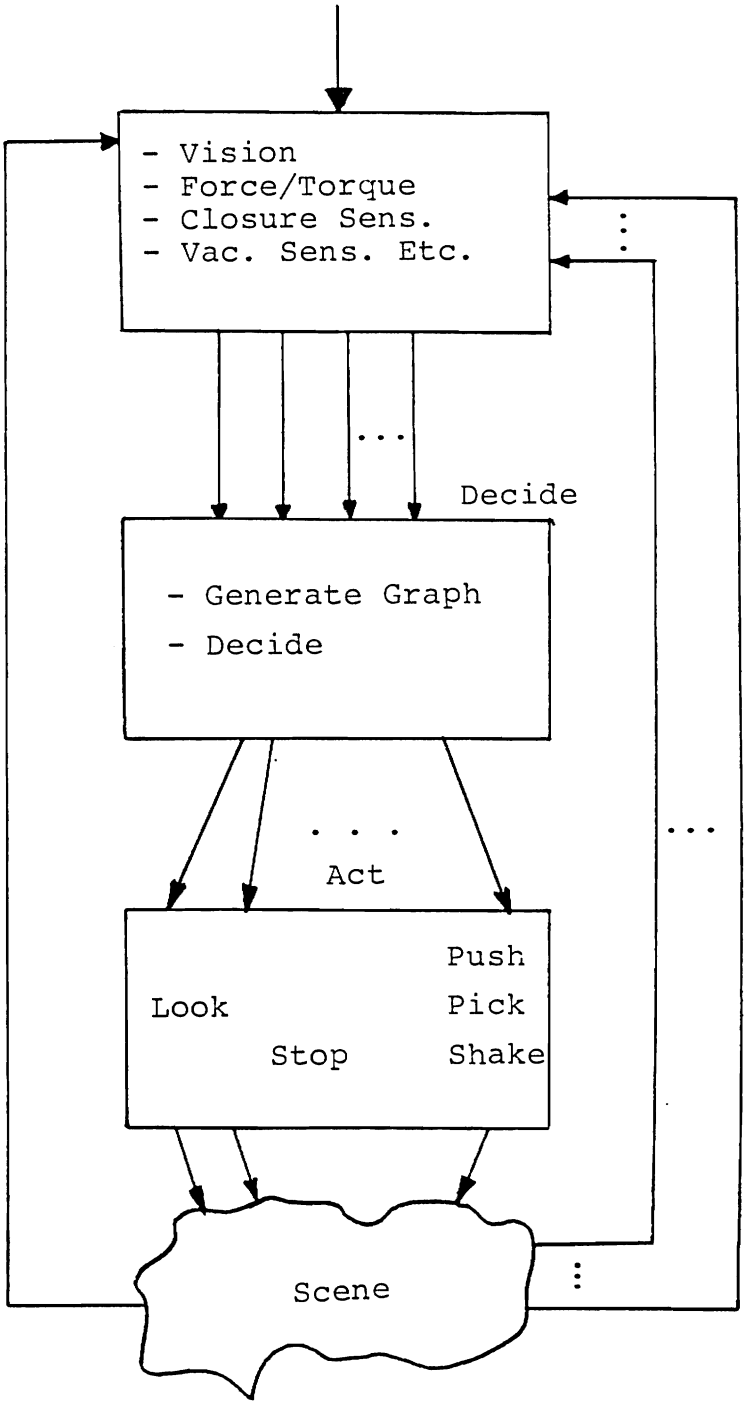

Fig. 1. The Model of Sensing, Manipulation and Control
A Non-Deterministic Turing Machine

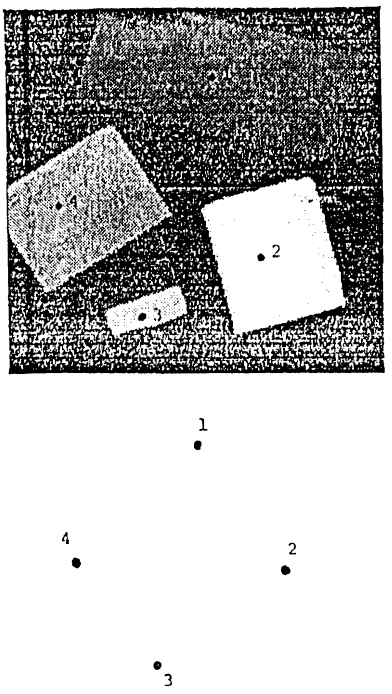

Fig. 2. Range Image of a Dispersed Scene and the Corresponding Digraph.

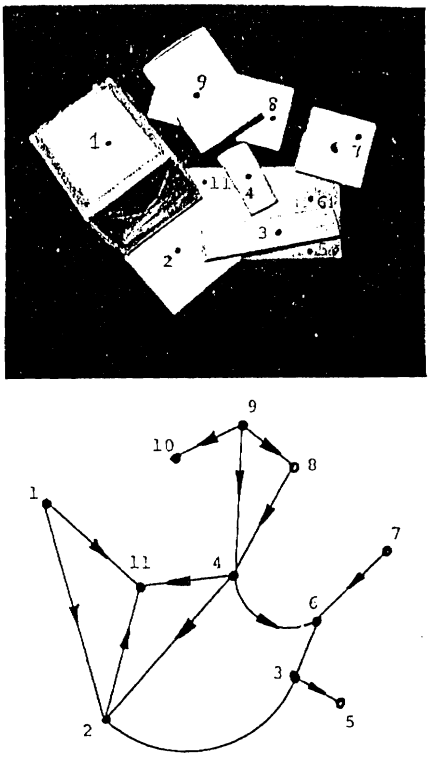

Fig. 3. Intensity Image of an Overlapped Scene and the Corresponding Digraph.

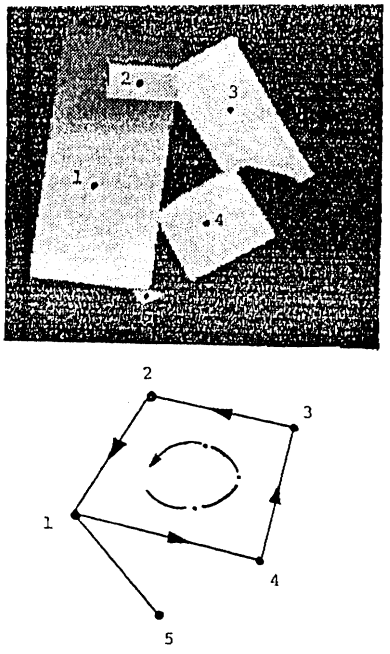

Fig. 4. Range Image of an Ambiguous Scene and the Corresponding Digraph.

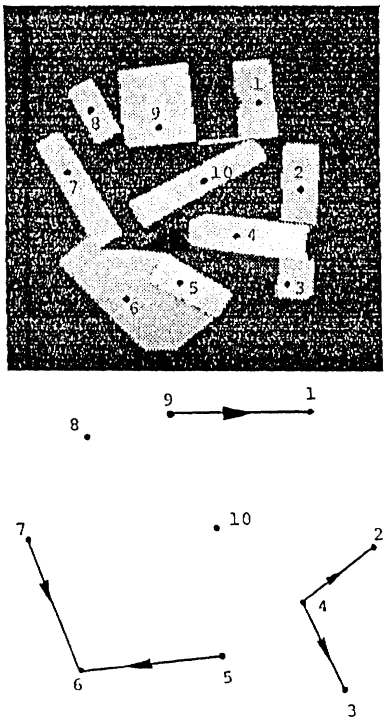

Fig. 5. Range Image of an Unstable Scene and the Corresponding Digraph.

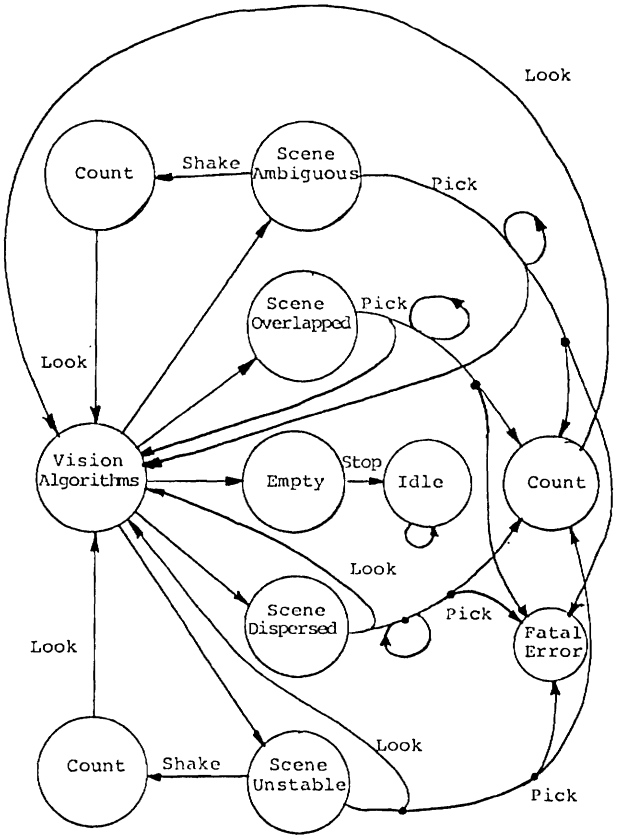

Fig. 6. Strategy-1 Action Automaton (Look, Pick, Look, ...)

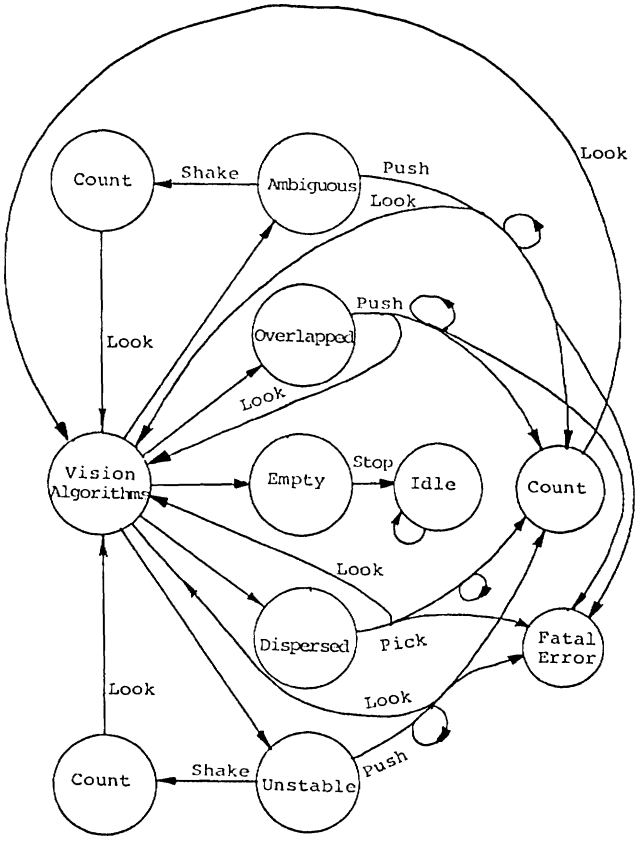

Fig. 7. Strategy-2 Action Automaton
(Look, Push_Until_Dispersed, Pick, Look)

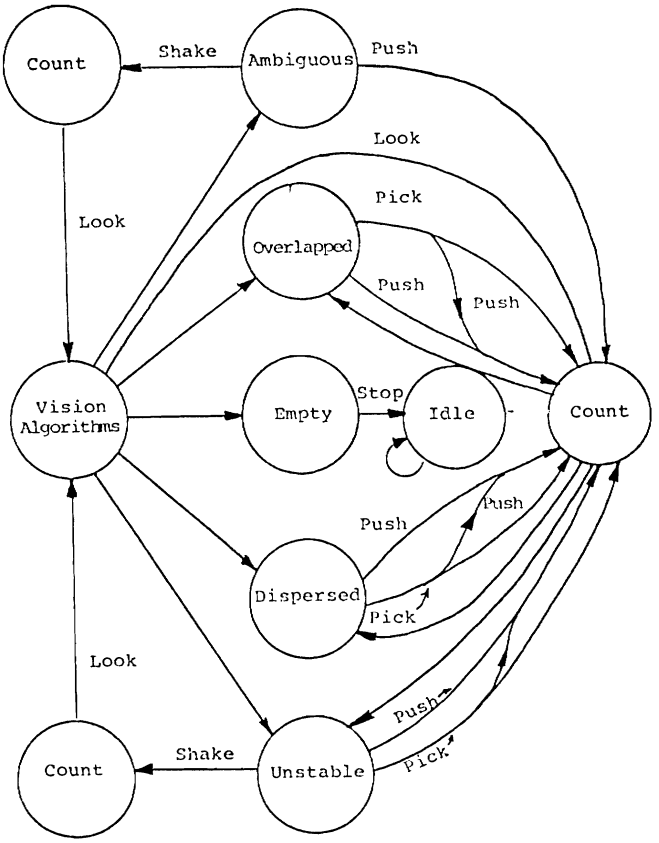

Fig. 8. Strategy-3 Action Automaton (Look, Pick/Push, Look, ...).

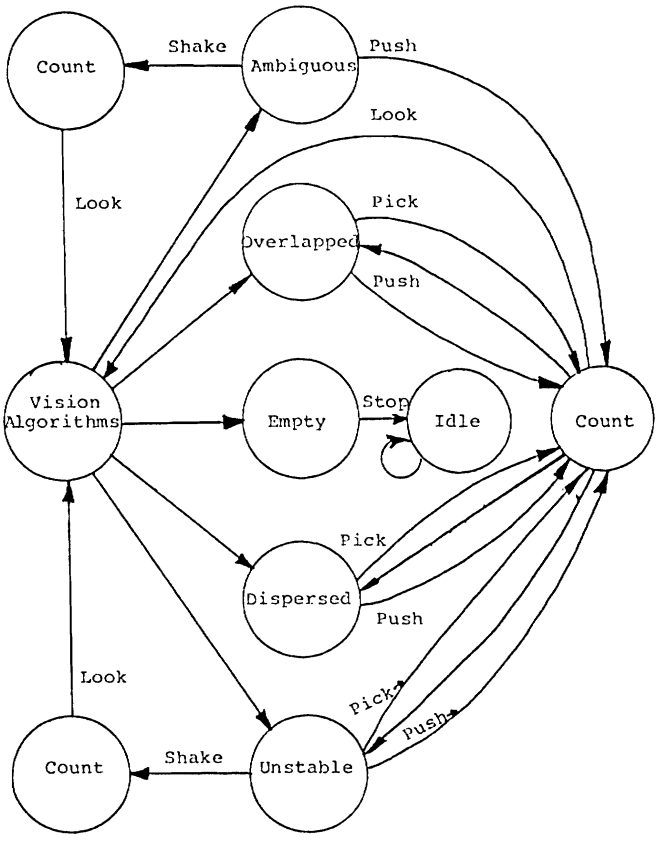

Fig. 9. Strategy-4 Action Automaton (Look, Push_Partially_Visible, Pick, Push, Look, ...).

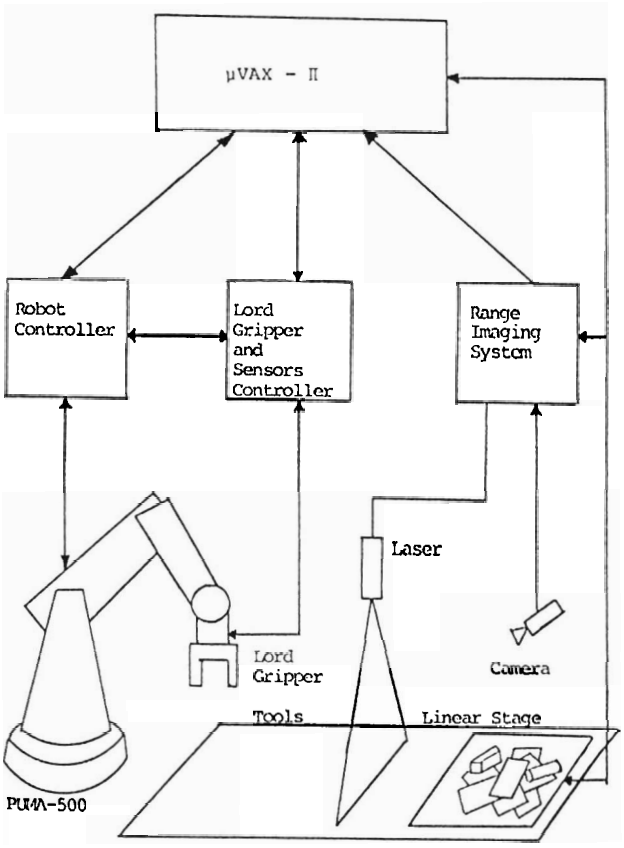

Fig. 10. Experimental System Block Diagram.

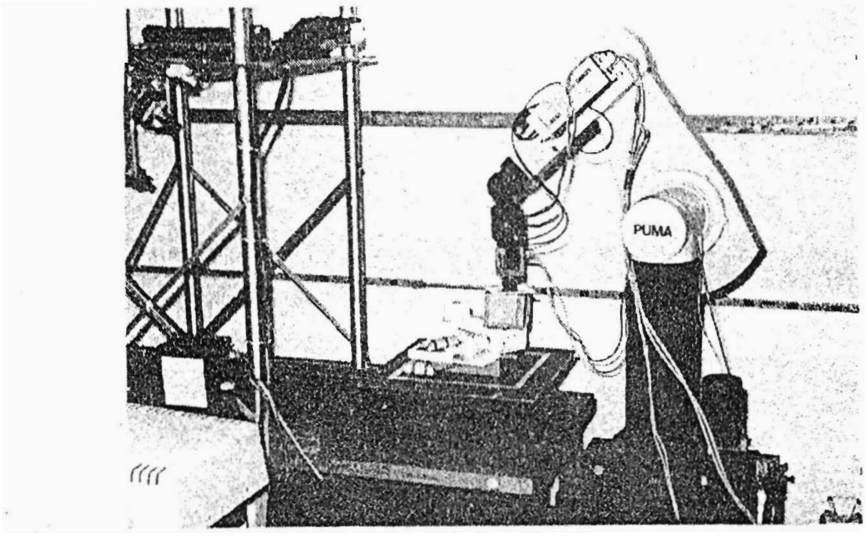

Figure 10a. An Example of the "PICK" Action