CyberFacts, Inc.
101 Moore Road, Sudbury, Massachusetts 01776 (617) 443-8900

DEVELOPMENT OF A MODEL OF MACHINE "HAND-EYE" COORDINATION AND PROGRAM SPECIFICATIONS FOR A TOPOLOGICAL MACHINE VISION SYSTEM

Final Report

June 30, 1972

Prepared for

National Aeronautics and Space Administration
Washington, D. C. 20546

Contract NASW - 2243

DEVELOPMENT OF A MODEL OF MACHINE "HAND-EYE" COORDINATION AND PROGRAM SPECIFICATIONS FOR A TOPOLOGICAL MACHINE VISION SYSTEM

Final Report

Prepared for

National Aeronautics and Space Administration
Washington, D.C. 20546

Contract NASW-2243

Dr. Leo C. Driscoll
Research Director

Cyberfacts, Inc.

June 30, 1972

DEVELOPMENT OF A MODEL OF MACHINE "HAND-EYE" COORDINATION AND PROGRAM SPECIFICATIONS FORA TOPOLOGICAL MACHINE VISION SYSTEM

Abstract

A unified approach to computer vision and manipulation is developed which is called choreographic vision. In the model, objects to be viewed by a projected robot in the Viking missions to Mars are seen as objects to be manipulated within choreographic contexts controlled by a multi-moded remote, supervisory control system on Earth. A new theory of context relations is introduced as a basis for choreographic programming languages. A topological vision model is developed for recognizing objects by shape and contour. This model is integrated with a projected vision system consisting of a multiaperture image dissector TV camera and a ranging laser system. System program specifications integrate eye-hand coordination and topological vision functions and an aerospace multiprocessor implementation is described.

TABLE OF CONTENTS

SECTION I. OVERVIEW OF RESEARCH

SECTION II. EYE-HAND COORDINATION MODEL

SECTION I. OVERVIEW OF RESEARCH*

1.1 Introduction

The value of unmanned teleoperator or robotic flights has been demonstrated by the Soviet Union in the Luna 16 Flight in which a machine system retrieved a lunar sample at an estimated 1/50th of the cost of a similar manned mission.

A prerequisite for automatic assembly of space station and space laboratory components both for near-earth and interplanetary missions is a machine "hand-eye" coordination system. The present research derives its rationale from this need as well as requirements scaled to machine vision implementations that can be forecast for unmanned spaceflight in the 1977 era.

Research in the U.S. on machine vision and computer control of mechanical arms and hands is proceeding principally at M.I.T., Stanford University, Stanford Research Institute and Battelle Institute. University labs such as the Artificial Intelligence Lab at M.I.T.'s Project MAC usually use a DEC PDP-10 class of computer for these investigations and their objectives are largely theoretical. They are investigating the general problem of computer vision and seek a phenomenology of machine vision derived from a better understanding of how machines can emulate humans in analyzing scenes.

The specific objective of the present contract is to provide NASA with an alternative approach that can later be evaluated in a test-bed environment. This work develops a unified approach to computer vision and manipulation that we term choreographic vision. Results of the research can be scaled to limited computation resources -- aerospace multiprocessors. This work is at once an approach toward a new theory of machine perceptual-motor functioning and an applications study directed toward context-dependent perceptual-motor tasks for unmanned space missions.

1.2 NASA Context

The original NASA context toward which this research was oriented is the need for robotic elements in interplanetary flights. Research contributions are focused on projected unmanned missions to Mars currently being planned by NASA/Ames and the Jet Propulsion Laboratory.

*Supported by National Aeronautics and Space Administration, Washington, D C. under Contract NASW - 2243.

Our development of a logical and mathematical model of robot vision can be regarded as a continuation of earlier research undertaken for NASA by the M.I.T. Instrumentation Laboratory.[1] This work, conducted by Sutro and Kilmer (with the collaboration of Dr. Warren S. McCulloch) was also oriented toward defining robotic elements for the exploration of Mars.[2]

Sutro and Kilmer have pointed out that the need for robots (aside from questions of cost and feasibility for equivalent manned missions) is closely related to the up-link and down-ling transmission times involved in earth command and control of a Martian lab vehicle. A single bit of information from Mars to Earth requires several minutes for transmission. This requirement rules out anything but pure robotic elements for real time control. For example, transmission delays of this magnitude rule out the exclusive use of teleoperators in remote supervisory control modes. Unless the robot has autonomous perception and manipulation, perceptual-motor loops cannot be closed.

An interesting strategy for commanding and controlling a robot on the Martian surface would be the use of pure robotic elements in conjunction with the use of symbolic, supervisory control for specifying plans, experiments, and schedules of perceptual-motor activity. Perceptual motor relations in particular could be specified. This would correspond to transmission of coded commands to the robot's multi-moded relational computer on Mars. An aerospace multiprocessor would then command the broad-bandwidth, perceptual-motor behavior loops of the robot. Such a stratification of command and control exists in biological systems.

1.3 Approach to Problems of Machine Hand-Eye Coordination

Research on computer vision and computer control of electro-hydraulic mechanical manipulators has generally proceeded in parallel paths with little or no-cross fertilization of strategies. They are conceptualized as separate problems. However, it is becoming increasingly important to study the problems of computer eyes and computer hands in a unified way in an attempt to develop the beginnings of a theory of machine perceptual-motor functioning. Research on computer vision alone--

[1] NASA Grant, NGR 22-009-140, to the Instrumentation Laboratory, MIT, from NASA Electronics Research Center in 1968.

[2] R-582, Assembly of Computers to Command and Control a Robot, L. Sutro and W. Kilmer, February, 1969, MIT Instrumentation Laboratory, Cambridge, Massachusetts.

analysis of complex visual scenes by computers using television eyes--is an extremely valuable scientific program that will lead to advances in artificial intelligence. However, in the interim we need applications-oriented research in which a limited computer can control a sensor and an actuator in certain restricted contexts of machine perceptual-motor tasks.

Out of the M.I.T. lab studies of machine vision of natural objects such as fruit, there has emerged a realization of the crucial role played by context in seeing.[1] For example, if a machine knows beforehand that an apple has more of a specular surface than a pear, it can interpret highlights more efficiently. The role of context in the specialized vision that we shall study in this research is even more critical. As a result, context will be formally modelled here as an operator in evaluating functions and mappings. Context will be taken not as the visual field or surroundings of the imaged object, nor even the lighting conditions or the direction of the light source. Rather, context will be modelled choreographically in terms of the interactive eye-hand elements in a manipulation scene.

For example, a nut on a threaded shaft will be viewed in terms of rotation and torque. An occluded object will invite a pushing action of the hand to separate the scene into non-occluded objects. Thus an object will have as a set of properties not only its visual properties such as surfaces, outlines, shadows, but a set of properties as an object to be manipulated.

Interactive properties will be sought which can be hypothesized and verified by topological tests. In this respect, we shall differ considerably from M.I.T. heuristics which are frequently derived from the artificial geometric context determined by regular polyhedra that can be modelled as line drawings "or subsets of the abstract retina." Thus vertices and edges may be of lesser importance than shape properties that can be induced by deformation-testing.

One of the recent thrusts of M.I.T. research has been the formation of concepts on the basis of visual information.[2] The work of Winston has used relation-constructs for machine representations of 3-dimensional

[1] Cf. Lawrence J. Krakauer, MAC TR-82, "Computer Analysis of Visual Properties of Curved Objects," M.I.T. Artificial Intelligence Lab, May 1971, p. 10.

[2] Cf. Patrick H. Winston, MAC-TR-76, "Learning Structural Descriptions from Examples," M.I.T. Artificial Intelligence Lab, September, 1970.

structures such as arches. This approach will be further formalized by constructing eye-hand relations using the diagrams of category algebra and mappings derived from contextual logic.[1] In this way, we shall realize in limited applications, a calculus of relations which will permit the formation of hypotheses about the interactive space for the eye and the hand.

From the viewpoint of methodology, the present work relies less on anthropomorphic models of machine vision than does M.I.T. They seek a common thread between the phenomenology of human vision and machine vision. This is a valid approach in constructing a laboratory for machine vision. In our circumstances, we shall modify this strategy somewhat and depend more on machine-specific heuristics for seeing. We do not have the luxury of large-scale machines and software. Also in the NASA context, the computer environment for spaceflight will also tend to be austere.

In the place of attempts to approach the versatility of human vision, our efforts will be directed toward realizing practical machine vision for limited, albeit mission-oriented, contexts. Instead of searching for the universal intelligent machine eye, we are constructing models for reasonable machine perception and arm/hand control based on perception.

In place of the extensive geometric heuristics at M.I.T. we shall depend on topological heuristics that are more adapted to seeing natural objects (viz. not confined to regular polyhedra) and which can be expected to require less of the computer in terms of speed and storage. Topological encodings will form a common bridge for the eye and hand language and will permit economical machine implementations of the resulting choreographic language of the eye-hand interactions.

Programs using geometric processing of image data tend to grow without bound. For example the program of Krakauer for recognition of curved objects requires 5,200 36-bit words of PDP-10 MIDAS assembly language code plus 1,700 words of fixed buffer and tablets plus the dynamically allocated array and list structures which can grow to an arbitrary size.

Our approach to machine vision is active vision. Other investigators have been impressed with how much an eye can see without extensive

[1] Cf. John C. Kotelly, "Context Logic I: Fundamental Concepts, Notations, and Derived Notions," Notre Dame Journal of Formal Logic, Vol. XI, No. 4, October, 1970, pp. 431-446.

special preparation or interaction with objects in the visual field.[1] We are equally impressed by the activity required for visual hypothesizing at all levels of processing visual information.

Everyone agrees today that the secret of vision is intimately connected with the ability to selectively throw away masses of visual data that tend to generate model mismatches. However, this ability to throw away is not to us a passive phenomenon on the part of the eye, but is the payload of very active hypothesizing in which competing hypotheses are allowed to contend for the honor of being the hypothesis that the eye will adopt relative to the scene before it... Seeing is thus a form of hypothesizing and a machine model must use the logic of hypothetical inference as its strategy. This accounts for our efforts to formalize hypothetical inference--a subject that has not received enough attention in studies of artificial intelligence.

1.4 Approach to Development of Choreographic Vision Model

The dominant idea of choreographic vision is that we are not interested in "scenes to be analyzed" or still-frames of objects as entities that have interest to the machine retina or the machine lateral geniculate body. "Choreography" denotes a sequence of frames in which there are changes in the positions and relative orientations of the objects. By generalizing this concept of motion sequences, an object to be viewed is seen as an object to be manipulated and instead of the visual scene that is the usual problem-solving format for computers-with-vision we substitute the choreographic context.

For example, in the task-context of a robot performing automatic assembly operations in automotives, a large circular object with radially spaced projections is to be seen as a wheel to be mounted onto a car. The wheel-choreogaphy would then be represented by a sequence of manipulator actions culminating in rotations with associated torque quantities which would correspond to hand-eye coordination in the " mount wheel context."

The list of object properties in any choreographic frame must include the frame-transformation variables that are required to bring the object to the next choreographic frame. Thus typically,object orientation and motion-path and force-vector would be part of the choreographed model of the object to be manipulated.

[1] Cf. M.I.T. Project MAC Report VI, p. 15.

To grasp the significance of this approach, consider that what the eye-hand system has to do is to implement a "cine" sequence in which by eye-hand choreographic interactions it creates its own "filmstrip." If the "filmstrip" matches internally stored choregraphed models, then the choreographic sequence matches an eye-hand task sequence. Otherwise, it will continue to construct the required "filmstrip" by adopting and confirming and rejecting alternative hypotheses about the object to be manipulated.

For example, suppose that in a "mount wheel context," robot operation of a wrench does not produce the desired sequential goal frame of a mounted wheel. An alternative hypothesis would then be tested that a thread might be stripped and some eye-hand test would be called to verify the choreographic subsequence of discovering if a thread is stripped. A stripped thread could perhaps be detected by simply viewing the wheel lug as a helix or a broken spring.

One of the significant discoveries of the M.I.T. research program on artificial intelligence is the superiority of vertical system structures in a program vs. horizontal system structures. Earlier horizontal processing of visual data from a television camera proceeded by sequential operations such as preprocessing to establish a favorable signal-to-noise ratio, detecting of edges, abstraction of features, finding contours etc.

In a horizontal or hierarchical system, there is an implicit division into lower level processing and higher level processing and there is little iteration or recursion in the sequence. As a result, earlier errors get propagated such that higher level heuristics must become inordinately powerful to process the visual data.

The new approach to this problem is to treat vision as a heterarchical system in which there is considerable communication of cues and information among the various vision-processing routines at all levels. This result is particularly important in this study since we are aiming at a model of eye-hand coordination. We envision a vertical structure which has considerable feed forward of information to higher level processing and considerable feed backward to lower level processing. Further, at all levels, a unifying topological language and hypothetical inference framework will be used to verify the choreographic context operators. The vision model consists in an integration of the eye choreography and the hand choreography. This integration gets realized by a vertical program structure. This software structure is thus a mirror of feedback structures in the hardware.

One of the characteristics of horizontal systems is their contextual-poverty. As Krakauer notes, "....there will always be ambiguities in finding significant pattern features which can only be resolved through the use of context."[1] We propose to use the vertical structures which tend to naturally propagate contextual information among the various vision processing routines, as a formal system for imbedding context logic.[2]

A by-product of a formal contextual vertical program structure, will be the relaxation of hardware requirements. Television sensors of a standard type and relatively imprecise preprocessors can be used since the precision of the system will be derived from the logical framework and the ability of the system to emulate the hypothetical inference that is found in human and probably animal vision.

The problem-solving domain of the choreographic vision model will be an interactive space. By this we mean an eye-hand coordination space which is coordinately described in terms of eye properties that indicate the existence of a choreographic frame and hand properties that transform that frame into a next-eye-frame.

Formalisms have been developed to serve as a basis for modelling perceptual-motor loops.[3] These loops are modelled by entailment diagrams involving hand-eye data, choreographic properties (characteristics of objects-to-be-manipulated), and choreographic models.

A formal system for tracking entailments in context is extended to provide a formalism for modes of hypothetical inference.[4] The hypothetical inference schema of Polya[5] is also formalized in order to account for the inter-dependency of likelihoods among perceptual-motor events using process variable information from a Kalman filter. The entailment diagrams which a robot must construct in order to command and control perceptual-motor behavior loops via its relational computer are formalized.

[1] Krakauer, op. cit. p. 10

[2] J.C Kotelly, "Context Logic I," Notre Dame Journal of Formal Logic, Fall, 1970.

[3] Quarterly Report No. 1: Logical Development of Choreographic Vision Model, Cyberfacts, Inc., October 15, 1971.

[4] J. C. Kotelly, "Context Logic I," Notre Dame Journal of Formal Logic, Vol. XI, Number 4, October, 1970, pp. 431-446.

[5] Polya, Patterns of Plausible Inference, Vol. II, Princeton University Press, 1954.

The logical choreographic model can be compared to theories of human perceptual-motor behavior. In the choreographic model, vision is interpreted in terms of hand/arm properties (objects to be grasped and manipulated). Coordinately, manipulation is interpreted in terms of changing the "cine" sequence of visual fields via interactions with the objects in that field and by being part of the visual field. This approach can be compared with Smith's neurogeometric theory of motion.[1] In Smith's theory, perception and motion are seen as inseparable units of behavior. Their interactions constitute the perceptual-motor behavior loop.

The choreographic model stresses three components of robot behavior: interaction, context, and relational modelling.

To paraphrase A. R. Johnson, grasp will be a more fundamental perceptual mode than mere touch for a robot on the Martian surface. Looking will be more essentially descriptive of vision than mere seeing. It will be the total involvement of the robot that will be paramount.[2]

For a Martian robot, sensory input (eye-hand data) will be metaphoric. There will be an hypothesized comparison of eye-hand data and the event which produced it. The sensory input itself will not be modelled as the event. It will simply be interpreted as a metaphoric statement about the event or object. It will remain for the robot to discern the meaning of the metaphor through follow-on-eye-hand behavior (which will be again more than simply a set of passive, sensory observations). Meaning will be found in the response which the robot finds appropriate (in accordance with stored models of contexts and filters that enable the robot to select out contexts). The context of the event will act as an operator in the relational computer of the robot.[3] It will assign meaning to the manifold of sensory metaphor. Discernment and matching of context will be a continuous, active process in the relational computer, and eye-hand data will be matched with internally stored choreographic models via modelling of choreographic properties of objects.

[1] L. E. Smith and J. S. Harrison, "Comparison of the Effects of Visual, Motor, Mental, and Guided Practice Upon Speed and Accuracy of Performing a Simple Eye-Hand Coordination Task," Research Quarterly, 33, 299-307, 1962.

[2] A. R. Johnson, "Organization, Perception, and Control in Living Systems," Industrial Management Review, Winter 1969, Vol. 10, No. 2, p.2.

[3] H. T. Hermann and J.C. Kotelly, "An Approach to Formal Psychiatry," Perspectives in Biology and Medicine, Vol. 10, No. 2, Winter 1967, 272-308.

The entailment/relational diagrams (eye-hand data → choreographic properties → choreographic models) will be the blue-print for the robot's perception. What does perception do?

> Keeping up to date the internal organizing system, that represents the external world, by an internal matching response to the current impact of the world upon the receptive system is called perception.[1]

How will the robot realize perception?

> In a sense, the internal organizing system is continuously making fast-time predictions on what is going on out there and what is likely to happen out there, and it takes anticipatory action to counter the small errors that might threaten its overall stability.[2]

The choreographic Vision model is based on a calculus of relations for the robot's relational computer. A context relation logic has been developed for the first time. The objective is to provide a firm basis for future choreographic programming languages. This theory will be to a choreographic programming language what the λ-calculus of Church was to development of LISP.[3]

The crucial importance of a powerful theory of relations and a correspondingly machine-implementable calculus was recognized by the late Dr. Warren S. McCulloch. McCulloch was searching for a theoretical basis for nueurophysiological models that could serve as models for robots as "descriptions of life applicable to either men or machines." The recent valuable work of Winston shows the power of a relational approach to internal machine modelling.[4]

---

[1] W. M. Brodie and N. Lindgren, "Human Enhancement Beyond the Machine Age," (Describing the work of D. MacKay) IEEE Spectrum, February, 1968.

[2] ibid.

[3] A. Church, The Calculi of Lambda-Conversion, Princeton University Press, Princeton, 1941.

[4] P. H. Winston, MAC-TR-76, Learning Structural Descriptions from Examples, MIT Artificial Intelligence Lab, September, 1970.

Sutro and Kilmer have described the functions of a relational computer by making analogies to the functions of the cerebral cortex. They looked on the relational computer as the lateral geniculate memory in the hierarchy of computers to command and control a robot.

> Visual memory will store the relations found in.... stimulus patterns by the visual subsystem. Hence the term: relational computer. An object will be stored, not as a picture, but as a structure of relations, or model, which cause the robot to do something: run from the object, pick it up, experiment with it. Such a model can either be built in or learned.[1]

1.5 Approach to Development of Toplogical Vision Model

Two new, complementary approaches to topological machine vision form a basis for subsequent system program specifications.

A shape-recognition model takes advantage of the inherent edge-enhancement and shape-detection properties of multiaperture image dissector television cameras. A shape encoding schema is developed with which we can associate activation of subsets of the multichannel aperture matrix with graph elements out of which shapes can be constructed. The rationale of this encoding schema is to provide economical machine representations of the shapes of both artifacts and natural objects both for processing in a limited computer environment and for transmission to earth-based computers for image-reconstruction.

The model also presents an alternative approach to storing and transmitting shape-encodings. A variant of grass-fire encoding is explored which should minimize storage of a large number of shapes and should provide faster correlation of an observed shape with a stored shape. More significantly, grass-fire encoding, which is a technique for characterizing a shape in terms of structural transformations of that shape in Euclidean space, may offer a practical strategy for minimizing bandwidth requirements for transmittal of shape information on the downlink to earth.

A contour-recognition model is developed for processing inputs from the projected laser ranging systems for the Viking missions. This model provides a method for encoding laser-derived images of 3-dimensional, smooth, compact objects. A graph-representation of contours leads to graph-representations of a sequence of contours obtained by scanning the

[1] L. Sutro and W. Kilmer, op. cit., p. 13.

laser through planar cross-sections perpendicular to the line-of-sight. The feasibility of the method is based on a theory of Morse which limits both contour sampling and scanning from single viewing angles and from multiple viewing angles. An approach is also given to the problem of object-identification under various orientations or in different positions.

SECTION II. EYE-HAND COORDINATION MODEL

2.1 Hypothetical Inference Processes of Choreographic Vision

The problem of hypothetical inference for a choreographic vision system is to substitute for a complicated tangle of predicates or properties attached to an object in the field of vision, a single conception of an object to be manipulated.

From the viewpoint of the choreographic processing system, the process of hypothesizing looks like this. The internal machine model of a class of objects to be manipulated, O, has for its contextual properties (without breaking down into distinct subcontexts) a set of predicates, P_1, P_2, P_3, etc. As a result, I infer that the object in the field of view belongs to O.

The program will not admit or form a hypothesis, unless it can develop a model from it that will account for both eye data and hand data. Following Peirce, the form of inference is as follows: "The surprising fact, C, is observed; but if A were true, C would be a matter of course. Hence there is reason to suspect that A is true."[1]

For example, the outline of a printed circuit board has suddenly become an irregular polygon. If the printed circuit board is placed behind other components, then this would account for the sudden change in contour. As a result, there is reason to hypothesize that the printed circuit board is occluded by other objects in the field of view.

Another example involves hand-data. The gripper is slipping when it trys to grasp the target object to be manipulated. If the object is covered with oil, then it will have specular rather than matted properties when differential light regions and contours are constructed. The object does indeed have highly specular components in the image. There is reason to hypothesize that the gripper is in contact with the correct object, but it has become coated with oil.

The rationale for hypothetical inference is covering a smaller bet with a larger bet. The program places a high degree of likelihood in the condition "if A were true, C would be a matter of course." As a result it hypothesizes as likely that A is true (where C is the larger bet and A is the

[1] C. S. Peirce, Collected Papers, Vol. 5, paragraph 189, Harvard University Press, Cambridge, Mass. 1960.

smaller bet). It has been observed by Peirce and others that this process has the form of modus ponens where if p is hand or eye data and q is a set of choreographic properties and r is a choreographic model:

$$(p \longrightarrow q, \quad q \longrightarrow r) \Longrightarrow (p \longrightarrow r).$$

The modellability of the middle term q permits the machine to model the eye-hand data, p.

Now since machine vision requires selection of modellable properties and does not have, as we do, an innate intelligence for hypothesizing correctly over the long run, there is required another form of hypothetical inference which rejects lower level processing of visual data as the basis of rejecting the modellability of a set of choreographic properties. This inference mode is modus tollens, This will take a machine form as follows: "The surprising fact, C, is observed, but if A were true, C would be a matter of course. But in certain contexts, A is not true. As a result, in those contexts C does not offer any information to the choreographic program."

Modus ponens is thus the machinery for forming hypotheses and modus tollens is the machinery for deselecting hypotheses.

For example, highlights are observed which would correspond to the curved surface of a capacitor. However, in the context determined by handling integrated circuit boards, it is unlikely that a capacitor-scale object exists. As a result, it is unlikely that capacitor highlights are being observed.

Since hypothetical inference is a form of machine betting on the likelihood of model match or model mismatch, we need a formal system for balancing likelihoods or for introducing a contextual awareness at the outset which biases a vision system toward one of the other choreographic responses. There can be several approaches in solving this problem. Hypothetical inference is currently being investigated by logicians with renewed interest since it is recognized as a very neglected and critical part of logic. There are schemes in Polya, Hintekka, Suppes, Tarski, Carnap. We shall provisionally adopt a scheme in Polya. Also instead of casting it into a logistic system (such as the probabilistic logics of Lukasiewicz), we shall adopt a form of natural inference as opposed to axiomatic inference and its deductive overtones.

2.2 Formalization of Polya Plausible Inference Scheme

Polya introduced his plausible inference scheme to provide an alternative to deductive or axiomatic inference systems.[1] In artificial intelligence studies today, schemes such as Polya's provide an alternative to theorem proving since they offer natural inferential schemes that can be generalized and formalized to provide machines with hypothetical inference.

We shall use the Polya scheme as a starting point to build a logical model of choreographic vision. His scheme will be generalized and formalized to provide a mechanism of inference that can lead to machine algorithms and heuristics in later phases of this research.

We start by assigning graduate levels of likelihood. Consider the following graduated scheme of levels and their interpretations.

L_1: "will be modelled"

L_2: "is more likely to be modelled"

L_3: "is somewhat more likely to be modelled"

L_4: "is somewhat less likely to be modelled"

L_5: "is less likely to be modelled"

L_6: "will not be modelled"

where "modelled" means realizing an effective match between the program's knowledge of the choreographic context and the program's perception of choreographic elements, and where "not modelled" means a mismatch. A choreographic element is defined as a property (or set of properties) of an object to be manipulated.

We define a set of necessary conditions, $N(x_i)$ for the modellability of a choreographic elements, x_i.

$$N\ (x_i) = \{n_1 \wedge n_2 \wedge n_3 \wedge n_4 \wedge \ldots \wedge n_n\}$$

We also define a set of sufficient conditions, $S(x_i)$ for modelling a choreographic element, x_i.

$$S\ (x_i) = \{s_1 \vee s_2 \vee s_3 \vee s_4 \quad \ldots \vee s_n\}$$

[1] G. Polya, Patterns of Plausible Inference, Vol. II, Princeton University Press, 1954.

In the Polya plausible inference scheme, we infer (modus tollens) from levels of confidence in the realization of a set of necessary conditions $N(x_i)$ to levels of confidence in the modellability of a choreographic element.

Major Premise $x_i \longrightarrow N(x_i)$ (if x_i is modelled, then we infer that the set of necessary conditions $N(x_i)$ is realized.)

Minor Premises	$L_1 N(x_i)$	$L_2 N(x_i)$	$L_3 N(x_i)$	$L_4 N(x_i)$	$L_5 N(x_i)$	$L_6 N(x_i)$
Conclusions	$L_2(x_i)$	$L_3(x_i)$	I*	$L_4(x_i)$	$L_5(x_i)$	$L_6(x_i)$

Correlatively, we infer (modus ponens) from levels of confidence in the realization of a set of sufficient conditions, $S(x_i)$ to levels of confidence in the modelling of a choreographic element, x_i.

Major Premise $x_i \longleftarrow S(X_i)$ (if a set of sufficient conditions, $S(x_i)$ is realized, then we infer that the choreographic element is modelled.)

Minor Premises	$L_1 S(x_i)$	$L_2 S(x_i)$	$L_3 S(x_i)$	$L_4 S(x_i)$	$L_5 S(x_i)$	$L_6 S(x_i)$
Conclusions	$L_1(x_i)$	$L_2(x_i)$	$L_3(x_i)$	I*	$L_4(x_i)$	$L_6(x_i)$

2.3 Kalman Filtering of Eye-Hand Data

Derivation of likelihoods for a hypothetical inference scheme must be either based on preprogrammed or learned assignments or must be dynamically derived from sensor/actuator data. We shall assume that the robot will have such a mechanism. One approach of deriving likelihoods is Kalman filtering.

2.3.1 Sensor Utilization-Optimum Parameter Estimation

We assume in this discussion that the robot control subsystem can receive measurement or observation inputs, m, from the sensor subsystem. The measurement, m, is a function of sensor process or actuator process variables to be estimated and subsequently autonomously controlled.

The robot's control logic must accomplish this process of estimation and control with minimum error in order to perform complex tasks.

*I is an indeterminate conclusion.

The robot's problem is as follows. How can it use an ensemble of observations or measurements in such a way that the robot's control subsystem has a model of the sensor or actuator task to be controlled?

Recall at this point that the robot is internally representing to itself aspects of its own behavior as it interacts with the behavior of objects within its sensor or actuator space.

Our approach to the mechanism that the robot will use to give itself an optimum estimate of process variables to derive Polya likelihoods for model match or mismatch is the Kalman Filter.

The Kalman Filter gives the optimum linear minimum variance estimate of process variable x_i from a set of measurements or observations m_i.

From an optimum estimate of process variable, the robot can compute a response function, u, to control (1) its own manipulatory and vision processes and (2) the Mars mission experiment process. The goal is to arrive at the desired values of the process variables that the robot is estimating.

The robot is confronted with the problem: what has it measured or observed and what is the contextual relevance of each measurement. The recursive filtering approach given by the Kalman computation is a process whereby the robot will use each new piece of incoming data to form an improved estimate of all the estimated quantities.

2.3.2 Model of the Problem

To illustrate an acceptable way for the robot to handle the two different classes of process variables it will encounter, let us imagine a process in which the robot must control two variables. One variable, A, is essentially constant with time. The other variable, B, is time-variant.

A point to be stressed is that a good a priori knowledge of the variables to be controlled must be either programmed into the robot (on the basis of prior computer simulation) or must be acquired via its own ability to perform hypothetical inference and receive information from the earth-based symbolic supervisory control system. Let us assume that our robot is the beneficiary of the results of computer simulation of the process (using conventional Monte Carlo methods in conjunction with a replication of the Kalman Filter in a lab computer). Simulation could also occur in training sequences. At any rate the robot does not become adaptive instantaneously but starts with some a priori models.

The time-variant variable, B, can be expressed:

$$B\ (t) = \frac{1 + b\,t}{c + d\,t} \qquad (1)$$

Thus in our two-variable problem (A and B), the robot will need to acquire the continuous time-history of a, b, c, and d, where

$$\begin{aligned} A &= (a,\ t) \\ B &= f\ (b,\ c,\ d,\ t). \qquad (2) \end{aligned}$$

in order to estimate the process variables, A and B. This is the first critical requirement for a Kalman Filter Robot.

The quantities to be estimated are collected into a state vector, $\underline{X}$. With the vector expressed in a row or a transposed form, the state vector in this problem is

$$\underline{X}^T = [a\ b\ c\ d] \qquad (3)$$

The value of a is the current value of A. The quantities, b, c, d, are the coefficients in the B variable model given in equation (1). These quantities can be taken as constants although it is to be expected that somewhat different values of these parameters would fit the local process behavior better at different time periods. So, to model the fact that these parameters need some freedom, a differential equation of state is used to express the rate of change of $\underline{X}$ as a random noise.

$$\underline{\dot{X}} = \underline{n} \qquad (4)$$

The effect of this noise is to hold up the gains of the recursive filter so that the estimates are more sensitive in later measurements than they would be without $\underline{n}$. For this purpose, the simplest model of the noise will be a white noise and this will serve the robot well for most control situations. Then by definition,

$$\underline{\bar{n}}\ (t) = 0 \qquad (5)$$

$$\overline{\underline{n}\ (t)\ \underline{n}\ (\tau)^{\ T}} = P\ (t)\ \delta\ (t-\tau) \qquad (6)$$

The best way to evaluate the P matrix is to note that it is numerically equal to the correlation matrix for the changes that we might expect in $\underline{X}$ over a unit time interval. As a result,

$$P\,(t) \;=\; \overline{\Delta \underline{X}\;(t)\;\;\Delta \underline{X}\;(t)^T} \qquad (7)$$

with

$$\Delta \underline{X}\;(t) = \underline{X}\;(t+1) - \underline{X}\,(t) \qquad (8)$$

We now have to derive an expression that maps the robot's physical measurement, m, into the variables of the state vector. By defining the calculated value of the measurement to be

$$M = f\,(a,\; b,\; c,\; d) \qquad (9)$$

we see that the next critical requirement is to be able to analytically express the measurement as a function of the elements of the state variable. The linearized theory requires the linearized relationship between the estimated quantities and the sensor subsystem measurements. In practice, this linearization is performed about the current value of the estimates. Hence at our current estimate of a, b, c, and d (current value of $\underline{X}$ in equation (3), M is calculated from equation (9).

Let us now define a measurement column-vector, $\underline{h}$, where

$$\underline{h}^T = \begin{bmatrix} \dfrac{\partial M}{\partial a} & \dfrac{\partial M}{\partial b} & \dfrac{\partial M}{\partial c} & \dfrac{\partial M}{\partial d} \end{bmatrix} \qquad (10)$$

From a knowledge of the current value of $\underline{X}$, equation (10) can be evaluated.

The actual measurement telemetered by the robot subsystem, measurement, m, is the ideal measurement given by equation (9) plus the measurement noise. That is

$$m = M + n \qquad (11)$$

The third critical requirement is for the robot to determine whether measurement, m, accurately represents the true value of M. To find this out, $\sigma^{2}m$ is specified as the variance of the random value of n. The robot can let this quantity vary for each telemetered sensory input in some way that is tailored to the characteristic of the individual sensory channel and its usual behavior over time for the range of industrial contexts it is expected to operate in. What is critical is that the robot can describe the accuracy of each individual measurement. Once this is accomplished, the robot has a model of the process.

To summarize the specification of a process model, an adaptive robot will have to be given some knowledge of the process variables to be controlled (as well as their approximate time profiles) probably by prior computer simulation of the process. Then, the robot should be able to express any measurement as a function of the process variables. Finally, it should be able to form a model of the accuracy of each individual measurement.

2.3.3 The Kalman Estimator

The "genius" of the adaptive robot is its recursive activity as a way of achieving self-reference. The recursive estimator for the current problem has the form

$$\underline{\hat{X}_K} = \hat{X}_{K-1} + \frac{E_{K'} \; \underline{h}_{K-1}}{\underline{h}_{K-1}^{T} \; E'_K \underline{h} \; K-1 + \sigma^2 mK} \quad (m_K - M \; (\hat{X}_{K-1}) \qquad (12)$$

$$E'_K = E_{K-1} + P_K \; (t_K - t_{K-1}) \qquad (13)$$

$$E_K = E_{K'} - \frac{E'_K \underline{h}_{K-1} \underline{h}_{K-1}^{\;T} E'_K}{\underline{h}^T{}_{k-1} \quad E'_K \underline{h}_{K-1} + \sigma^2 mK} \qquad (14)$$

where $\hat{X}_K$ = estimate of $\underline{X}$ after the Kth measurement.
E_K = covariance matrix for the errors in X_K.
m_K = measured value of Kth measurement telemetered by sensor.
$M \; (\hat{X}_{K-1})$ = calculated value of measurement based on last estimated state vector.

The computational steps are as follows.

(1) initialize $\underline{\hat{X}}$ to get $\underline{\hat{X}}_0$ by assuming inital values for a_0, b_0, c_0, and d_0 (based on prior simulation).

(2) Assume initial values for E_0. (This accuracy of this step is not critical if the robot is to be receiving a large number of measurements from a given telemetry channel associated with a given sensor over a period of time.)

(3) Assume a constant rate of change for the covariance matrix. A constant value for P_K is usually adequate for most processes.

(4) Calculate M $(\underline{\hat{X}}_0)$ and h $(\underline{\hat{X}}_0)$.

(5) Calculate E'_1 at time of first measurement (t_1 of m_1).

(6) Calculate E_1.

(7) Calculate $\underline{\hat{X}}_1$, then calculate A and B.

(8) Repeat (4) through (7) as process to be controlled continues to obtain the current value $\hat{X}_K$ at time of current measurement, t_K.

2.3.4 Synthesis

It is assumed that the process to be controlled is a function of a set of variables such as rates of movement, positional changes, etc. These variables are designated $q_1, \ldots\ldots, q_n$. Then in the generalized case

$$A = f\,(q_1, q_2, \ldots\ldots, q_n)$$

$$B = f\,(q_1, q_2, \ldots\ldots, q_n)$$

and

$$dA = \frac{\partial A}{\partial q_1} dq_1 + \frac{\partial A}{\partial q_2} dq_2 + \ldots\ldots + \frac{\partial A}{\partial q_n} dq_n \qquad (15)$$

A similar formulation holds for B, the variable variant with time. A nominal profile of A (t) and B (t) is stored within the robot's memory. The current estimate of $\hat{\underline{x}}$ permits computation of A and B. The difference between the estimate and the derived nominal n is obtained. The dq's are solved for in the context of affecting the desired change in the process variables. The result of this computation is fed back to complete the loop in real time to the sensor/actuator subsystems.

To return to the discussion of synthesis, the robot compares measurement information to the optimal profile within its memory. It thus is in a position to act cybernetically--as a governor--to bring the present response surface of process variables closer to the desired surface in the next time window, t + 1. Specifically, as a result of the preceding computations, the robot makes changes to the control functions (the q's) that are driving its control loop. When the robot must take very positive actions to stabilize this control loop, the partial derivatives in equation (15) can be changed in value so as to change the overall system's gains in controlling process variables.

2.3.5 Error Propagation

The covariance matrix, E, gives the statistics of the errors in estimates of the parameters, a, b, c, and d which are the elements of $\underline{X}$. It has the form

$$E = \begin{bmatrix} \sigma^2 a & \sigma ab & \sigma ac & \sigma ad \\ \sigma ba & \sigma^2 b & \sigma bc & \sigma bd \\ \sigma ca & \sigma cb & \sigma^2 c & \sigma cd \\ \sigma da & \sigma db & \sigma dc & \sigma^2 d \end{bmatrix} \qquad (16)$$

What we are immediately interested in are the statistics of the errors in the estimates of the process parameters, A and B. The linearized relationship between these errors is as follows:

$$\begin{bmatrix} \text{error in A} \\ \text{error in B} \end{bmatrix} = G \quad \text{error in } \underline{\hat{X}} \qquad (17)$$

where

$$G11 = \frac{\partial A}{\partial a} = 1 \qquad G21 = \frac{\partial B}{\partial a} = 0$$

$$G12 = \frac{\partial A}{\partial b} = 0 \qquad G22 = \frac{\partial B}{\partial b} = t\ (c + dt)^{-1}$$

$$G13 = \frac{\partial A}{\partial c} = 0 \qquad G23 = \frac{\partial B}{\partial c} = -\ (c + dt)^{-1}\ B$$

$$G14 = \frac{\partial A}{\partial d} = 0 \qquad G24 = \frac{\partial B}{\partial d} = \sigma 23\ t$$

Then the covariance matrix for the errors in the estimates of A and B is

$$E_{A\ B} = G\ E\ G^{T} = \begin{bmatrix} \sigma^2 A & \sigma AB \\ \sigma BA & \sigma^2 B \end{bmatrix} \qquad (18)$$

2.3.6 Simulation

A critical step towards realization of an adaptive robot will be acquiring the profile of parameter variations for the process histories of the planetary exploration or experimentation. We have to acquire a measure of the quality of the sensor subsystem measurements in specific experimental contexts. The objective is to determine the estimation errors for the experimental process.

The actual mechanization of an adaptive system is complex. The estimator may be very powerful and flexible, but in the last analysis it is limited by its knowledge of the process to begin with. It should be emphasized that by "process" we mean the product of the interactions of the robot's sensor and actuator subsystems with the special machine process variables under adequate time-histories to create an adequate measurement and synthesis model.

2.3.7 Performance Summary for Kalman Robots

Conventional automatic batch-processing control systems determine a system's dynamic state and associated uncertainties of measurement of the state at many different points simultaneously. A Kalman Filter Robot would differ by its recursive behavior. Its basic cycle yields the state vector estimate at time, t_K, together with the statistical uncertainties associated with the measurement at t_K, the error propagation equations relating t_K to some earlier time, t_{K-1}, and the earlier state vector estimate at t_{K-1}, together with its uncertainties. Each piece of measurement or observation data is processed as soon as it is received from the sensor/actuator subsystems.

There exists an essential difference between a Kalman Filter Robot and a robot that incorporates a weighted least square batch processor. Both are optimum estimators in the sense that their selection of the best estimate of the state minimizes the statistical dispersions around the estimate. Abstractly, they can be shown to be equivalent, but in practice they produce different behavior. A significant advantage of Kalman filtering is the savings in throughput time and storage thus making it ideal for the complex real-time applications of aerospace robots. Kalman filtering is currently used for adaptive navigation problems. Its rapid acceptance for real-time applications has however uncovered certain weaknesses and potential problem areas.

The primary problem-area is divergence or failure of the estimated quantities to track the nominal quantities. However, the general cause of divergence can be traced to inadequate modelling and simulation of the process to be controlled. Any attempt at economizing on this simulation activity can lead to systems failure if divergence is unacceptable within the control criteria required by the process.

Most real time applications will entail dynamics which are not known in great detail. Further the interaction of robot actuator subsystems will introduce new dynamics which will certainly not be known beforehand in detail. Since advanced adaptive robots will be taking observations of their own interactions with outside processes, divergence can lead to pathological behavior on the part of the robot. Let us examine the divergence phenomenon.

The robot's estimation algorithm operates on perturbation equations derived from a linearization about some reference state. Neglected nonlinearities associated with the process to be controlled or the robot's own interactions with this process, make the linearized equations only approximations to the actual physical control problem. Unless process noise is added in the filtering algorithm, to account for inherent uncertainties, the Kalman filter "thinks" it has an excellent solution and the actuator subsystem is commanded to act in accordance with this solution. As a result, as the robot continues to control a process, it eventually will tend to exhibit sloppy control behavior. The statistical uncertainties associated with the latest state vector estimate (which must be compared with the assumed uncertainties corrupting the measurements at each observation or measurement step) becomes overly optimistic. A statistical inconsistency then develops and divergence follows. The filter acts in a solipsistic fashion--much as some humans--and tends to give less and less weight to subsequent measurements since it "thinks" that it has already obtained an accurate model of the system to be controlled.

In our example of controlling processes A and B, we have created a mechanism to defeat divergence within the robot by adding a process noise matrix to the covariance of the state vector uncertainties between every two measurements. This results in a filtering algorithm that is somewhat stoical about its control task and places more weight on recent observations.

Other schemes can of course be devised in which weighting schemes or "tracking logics" can be given to robots in training sequences and can be specifically tailored to the demands of the process to be controlled. In this area, there will be no substitute for experience and at this point we have no experience in the rigors of space-exploration, adaptive robot control and have extrapolated from applications of adaptive control in aerospace problems such as navigation.

2.4 <u>Contextual Formalization of Modus Ponens and Modus Tollens</u>

The Polya scheme introduced in 2.2, even when formalized into a procedure that can be computed has a serious drawback. It is a context-free inferential schema (like nearly all formal systems). However, as realized by investigators in artificial intelligence, the ability to handle context is of critical importance in a computer-vision system of any kind. The next step then is to provide the Polya scheme (as we have formalized it) a recognition of contextual selection and deselection of hypotheses.

We now return to the context-free form of <u>modus ponens</u> where if p is hand or eye data and q is a set of choreographic properties and r is a choreographic model:

$$(p \rightarrow q, \quad q \rightarrow r) \Longrightarrow (p \rightarrow r)$$

We now put this expression into a standard form of MP. Given A, $A \rightarrow B$, then B. Thus A stands for the given premises $(p \rightarrow q,\ q \rightarrow r)$, B stands for the conclusion, $(p \rightarrow r)$ and $A \rightarrow B$ stands for the entire expression $((p \rightarrow q,\ q \rightarrow r) \Longrightarrow (p \rightarrow r))$.

Now context-free MP would have it that if A, and $A \rightarrow B$, then always and inflexibly, B. Returning to an earlier example, if p is an observation that a large circular object with radial projections is in the field of view, and q is an interpretation that the radial projections are bolts onto which nuts can be screwed, then always and inflexibly we must correlate observations of large circular objects with radial projections with a "mount-wheel chreographic model." However, if the robot is to be more versatile than SAM at General Motors (which is preprogrammed only to weld car frames), there most likely will be encountered other automotive (or machine tool) parts that also fit the hand/eye data, p. As a result, to resolve the ambiguity concomitant with context-free exercise of MP inference, there must be freedom to conclude to other choreographic models such as in this case "balance-wheel model," "change wheel model," "calibrate-wheel model," "adjust-brake model" etc. or more dramatically, "mount differential housing."

Following the context logic of Kotelly, a contextual MP form would have the following diagram

$$\begin{tikzcd} & B_1 \\ A \arrow[ur] \arrow[r] \arrow[ddr] & B_2 \\ & \vdots \\ & B_n \end{tikzcd}$$

such that given A, $A \rightarrow B_i$, then B_i.

In our model, this means that the premises, "$p \rightarrow q$, $q \rightarrow r$" must contain contextual clues such that a program can select out the appropriate conclusion, $p \rightarrow r$. Since p, and r appear in the conclusion to be selected out, the contextual clues must be assigned to the term that does not appear in the conclusion, q. It will be recalled that q was defined as a set of choreographic properties--properties of an object to be manipulated.

However, in Kotelly's treatment of context, the entire premise, A or ($p \rightarrow q$, $q \rightarrow r$), must be asserted in a specific context, in order for MP to be contextual. This corresponds with the veritcal structures described earlier, where at the lower levels of processing hand and eye data, context must provide interpretation as an operator. Context also operates in the middle level of q--assignment of choreographic interpretations--as well as at the higher level of discovering model match and mismatch.

At this junction, we can introduce the new formalism for contextual MP. We shall first present some essentials of the context C* logic.[1] The system uses the language of category algebra and its diagrams.[2] The universe of discourse in this formalism consists of the following primitives:

( ) corresponds to a place-holder with no specific restrictions
$\longrightarrow$ stands for the concept of mapping or entailment ("arrow")
= denotes the undefined equality between mappings
Ø denotes that a symbol appearing on the left is the name of the "arrow" on the right, i.e. Ø: $(\;)\longrightarrow(\;)$ or $(\;)\xrightarrow{\varnothing}(\;)$

Capital letters with stars represent context-selection operators, i.e. A, B*, C*, E* etc.
'the prime is applied to a ( ) giving a ( )' such that ( ) need not be equal to ( ).'

(A6) A context operator C* assigns an arrow to the origin-endpoint of an identify map. $\curvearrowright$

Given $\begin{tikzcd} (\;) \arrow[loop above] \end{tikzcd}$, then $\begin{tikzcd} (\;) \arrow[d, "C^*(\;)"] \\ (\;) \end{tikzcd}$

where the extremity of C* ( ) is either a place-holder or an arrow, i.e. the origin of C* ( ) is the origin of the identity map.

Df. A map Ø is defined to be a map <u>in the context C*</u> if the following diagram is commutative:

$$\begin{tikzcd} (\;) \arrow[rr, "\varnothing"] \arrow[dr, "C^*(D(\varnothing))"'] & & (\;) \arrow[dl, "C^*(R(\varnothing))"] \\ & (\;) & \end{tikzcd} \qquad (1)$$

i.e., $C^*(D(\varnothing)) = C^*(R(\varnothing)) \circ \varnothing$

At this point, we can return to MP in the Context Logic. Asserting an A (the premises of MP, viz. ($p\longrightarrow q$, $q\longrightarrow r$) corresponds to the assertion that 1_A is in the context C*, i.e. asserting the diagram:

[1] J. C. Kotelly, "Context Logic I," <u>Notre Dame Journal of Formal Logic</u>, Vol. XI, Number 4, October 1970, pp. 431-446.

[2] Cf. Saunders MacLane, <u>Homology</u>, Springer-Verlag, Berlin, 1963; Peter Freyd, <u>Abelian Categories</u>, Harper and Row, New York, 1964; and Barry Mitchell, <u>Theory of Categories</u>, Academic Press, New York, 1965.

$$\begin{tikzcd} 1_A \arrow[d, "C^* (1_A)"] \\ (\ ) \end{tikzcd} \tag{2}$$

If we assert $A \longrightarrow B$ (the whole expression $((p \longrightarrow q;\ q \longrightarrow r) \Longrightarrow (p \longrightarrow r))$, this corresponds to making a prediction on the above diagram exemplified by the following diagram:

$$\begin{tikzcd} 1_A \arrow[r, "Pd\ (C^*\ (1_A)) = \emptyset", ""{name=U, below}] \arrow[d, "C^*\ (1_A"'] & (\ ) \\ (\ ) \arrow[to=U, "Pd"'] & \end{tikzcd} \tag{3}$$

such that the domain of Pd $(C^*\ (1_A)) = \emptyset$ is 1_A.

MP is then an <u>operator</u> whose value results, in context, from application of the above diagram, i.e. MP (Pd $C^*\ (1_A)) = \emptyset = 1_B$ such that the following diagram would be commutative:

$$\begin{tikzcd} 1_A \arrow[rr, "\emptyset"] \arrow[dr, "C^*\ (1_A)"'] & & 1_B \arrow[dl, "C^*(1_B)"] \\ & (\ ) & \end{tikzcd} \tag{4}$$

The hierarchy of operators is accordingly: C*, Pd, MP, represented by the following diagram:

$$\begin{tikzcd} 1_A \arrow[r, ""{name=U, below}] \arrow[d, "C^*(1_A)"', ""{name=L}] & (\ ) \\ (\ ) & \arrow[from=L, to=U, "Pd"', ""{name=P}] \arrow[from=P, to=1-2, "MP"'] \end{tikzcd} \tag{5}$$

Df. Pd is an operator on C* such that Pd $(C^*(1_A)) = \emptyset$ and such that $D\ (\emptyset) = 1_A$.

Df. MP is an operator on the diagram

$$\begin{tikzcd}
1_A \arrow[r, "Pd\ (C^*\ (1_A))"] \arrow[d, "C^*\ (1_A)"'] & (\ ) \\
(\ ) \arrow[ur, "Pd"'] &
\end{tikzcd} \qquad (6)$$

whose value is $1_B = R\ (Pd\ (C^*\ (1_A))) = R\emptyset$ such that:

$$\begin{tikzcd}
1_A \arrow[rr, "\emptyset"] \arrow[dr, "C^*\ (1_A)"'] & & 1_B \arrow[dl, "C^*\ (1_B)"] \\
& (\ ) &
\end{tikzcd} \qquad (7)$$

is commutative.

In the formalism of the Context Logic, there is no explicit treatment of Modus Tollens. The form can easily be derived, however, consistent with the $\emptyset^*$ Logic, and consistent with the treatment we have given to MT up to this point.

MT will simply be an inference form that recognizes that the operator MP does not hold for a particular diagram. This does not imply that the context operator Q* becomes a "zero" context operator such that $Q^*\ (1_A)$ reduces to A, $Pd\ (Q^*\ (1_A))$ reduces to $A \longrightarrow B$, and $MP\ (A,\ A \longrightarrow B) = B$--- the context-free form of Modus Ponens. Contrariwise, MT is a form of context-dependent inference and its force is to recognize a breakdown in $\emptyset^*$ operators when applying MP in $\emptyset^*$.

It will be recalled that on page , MT was described as follows: "The surprising fact, C, is observed, but if A were true, C would be a matter of course. But in certain contexts, A is not true. As a result, in those contexts C does not offer any information to the choreographic program." We can represent this symbolically

$$(p \longrightarrow q,\ \ q \longrightarrow r) \not\Rightarrow (p \longrightarrow r).$$

where p is hand or eye data and q is a set of choreogrphic properties and r is a choreographic model.

The modellability of the middle term q permits the machine to detect a model mismatch for eye-hand data, p.

Now to return to the Context Logic, MT need not be defined as a new operator, but simply the non-realization of MP. A necessary and sufficient condition for the non-realization of MP as an operator is the condition that the diagram (7) is verified <u>not</u> to be commutative, i.e. $C^* (1_A) = C^* (1_B) \circ \emptyset$. This is equivalent to the condition that $C^* (D(\emptyset)) = C^*(R(\emptyset)) \circ \emptyset$ in diagram (1) i.e. that diagram does not commute. As a result, MT is the inference mode that detects that a map $\emptyset$ is not a map in the context C*. It concludes that at least for C*, MP does not have the value true. What this means then is that the middle operator in the hierarchy of operators, Pd does not hold for C*. We can now define MT more precisely in the $\emptyset$* Logic as an inference formed from detecting that Pd $(C^* (l_A) = \emptyset$ <u>or</u> $D (\emptyset) = l_A$. (See diagram (5). Now if Pd is discovered <u>not</u> to be an operator on C*, we conclude it may be an operator for D*, E*, F*, etc. but not for <u>this</u> context, C*. One final mechanism is involved to check whether C* is contextually nested in another context D* and if so, to infer MT that Pd is not an operator for D*.[1]

Up to this point, MP and MT have been considered only as inference modes in which deductions can be made in context. Since our interest is focused on having a machine hypothesize in context, we need to develop a hypothetical inference form of MP and MT in the $\emptyset$* Logic. We shall do this via the Polya Plausible Inference Scheme.

Earlier, the graduated levels of likelihood, L_i, were defined on sets of choreographic elements, x_i, sets of necessary conditions, $(N (x_i)$ for the modellability of choreographic elements, and sets of sufficient conditions, $(S (x_i)$. Correlatively, in the MP mode, we inferred from levels of confidence in the realization of a set of sufficient conditions, $(S (x_i))$, to levels of confidence in the modelling of choreographic elements, x_i.

Translating this into the $\emptyset$* Logic, on diagram (5) we can instantiate the place-holder, l_A, as p, eye-hand data. Similarly, we instantiate the place-holder, l_B, as r, a choreographic model. The prediction arrow or operator, Pd, will be interpreted as the double-entailment, "p ⟶ q, q ⟶ r," where q again is interpreted as a set of choreographic properties.

Now we interpret the contextual assertion of p via the map, $C^* (l_p)$.

$$\begin{tikzcd} 1_p \arrow[d, "C^* (1_p)"] \\ (\ ) \end{tikzcd} \qquad (8)$$

We also interpret the arrow or mapping, $C^* (1_p)$ as the assertion of the existence of a subset of the set of sufficient conditions for modelling p.

[1] Cf. Context Logic, <u>loc. cit.</u> p. 437 for the conditions of contextual-nesting.

Similarly, in the right-hand place-holder, we place 1_r, and as in the diagram (7) we assert the contextual identity of r via the map (C* (1_r) as the assertion of the existance of r in the context C*.

$$\begin{tikzcd}
1_r \arrow[d, "C^* (1_r)"] \\
(\ )
\end{tikzcd} \qquad (9)$$

We now define C* (S (x_i)) as a contextual set of sufficient conditions for p to be modelled, where C* (S (x_i)) can now be precisely defined as the composition of the two maps C* (1_A) o C* (1_B) or in our case, C* (1_p) o C* (1_r). Thus the mapping Ø which is the implication $\Longrightarrow$ in $(p \longrightarrow q,\ q \longrightarrow r) \Longrightarrow (p \longrightarrow r)$ is defined in terms of sufficient conditions via the commutativity of the diagram; i.e. 1_p Ø 1_r = C* (1_p) o C* (1_r). Now for this composition to be realized, the extremity of both contextual identification maps is a range of C* properties which we shall interpret as q, choreographic properties (in the context C*).

At this juncture, we can assign levels of confidence in the contextual identity maps to infer the level of confidence in the implication Ø which we can reinterpret as "p has a choreographic model in r." For example, using the table developed in 2.2 under MP, C*(1_p) oL_5 C* (1_r) = L_5 Ø, if we adopt the convention that $L_i \wedge L_j = L_i$ if $L_i \geq L_j$.

We can now give representative diagrams involving various formalized Polya assignments for MP.

$$\begin{tikzcd}
1_p \arrow[rr, "L_6\ (\emptyset)"] \arrow[dr, "L_1(C^*(1_p))"'] & & 1_r \arrow[dl, "L_6\ (C^*\ (1_r))"] \\
& (q) &
\end{tikzcd} \qquad (10)$$

$$\begin{tikzcd}
1_p \arrow[rr, "I^*\ (\emptyset)"] \arrow[dr, "L_2(C^*(1_p))"'] & & 1_r \arrow[dl, "L_4(C^*(1_r))"] \\
& (q) &
\end{tikzcd} \qquad (11)$$

where I* is an indeterminate assignment of confidence to the implication Ø.

[1] Except for a value, L_5, on the identity context maps, which yields

$$\begin{tikzcd} 1_p \arrow[rr, "L_4(\emptyset)"] \arrow[dr, "L_3(C^*(1_p))"'] & & 1_r \arrow[dl, "L_5(C^*(1_r))"] \\ & (q) & \end{tikzcd} \qquad (12)$$

$$\begin{tikzcd} 1_p \arrow[rr, "L_2(\emptyset)"] \arrow[dr, "L_2(C^*(1_p))"'] & & 1_r \arrow[dl, "L_2(C^*(1_r))"] \\ & (q) & \end{tikzcd} \qquad (13)$$

The next step in completing the hypothetical inference machinery for the Context Logic is providing a hypothetical MT operator just as we arrived at a hypothetical MP operator. However, as remarked earlier no distinct MT operator was needed for hypothetical MT in the $\emptyset^*$ Logic. What is involved is establishing that Pd is not an operator on C*. Now the nexessary condition for this is either Pd $(C^*(1_p)) = \emptyset$ or D $(\emptyset) = 1_p$. (Refer again to the second diagram on page 26), we can infer MT that "p does not have a choreographic model in r." Now a necessary condition for D $(\emptyset) = 1_p$ or $1_r = R\ (\emptyset)$ would be the condition that a <u>contextual shift</u> has occured and we are actually processing distinct, that is contextually distinct, choreographic properties, qi. As a result, diagram (4) could appear as follows:

$$\begin{tikzcd} 1_p \arrow[r, "\emptyset"] \arrow[d, "C^*(1_p)"'] & 1_r \arrow[d, "D^*\ E^*\ \ldots\ Z^*(1_r)"] \\ (q) & (q_i) \end{tikzcd} \qquad (14)$$

such that the diagram no longer commutes.

Placing this result in the formalized Polya framework, hypothetical MT proceeds by detecting the conditions of non-commutativity in the application of MP, or more precisely balancing the likelihoods that an MP diagram will <u>not</u> be commutative.

Thus if L_2 $((C^*(1_p))$ L_2 $(C^*(1_r))$, then (following the assignments on the table on page 15), if we define N (x_i) as $(C^*(1_p))$ and $(C^*(1_r))$, then we conclude L_3 (p), i.e. "the eye-hand data is somewhat more likely to have a choreographic model in r." Or if L_4 $(C^*(1_p))$ and L_5 $(C^*(1_r))$, then we infer L_5(p), i.e. "the eye-hand data, p, is less likely to have a choreographic model in r."

Specific machine procedures will be required to test for the existence of both the sets of necessary and sufficient conditions for MP and MT respectively and for the assignment of levels of confidence to the contextual identity maps, $(C^*(1_p))$ and $C^*(1_r))$ in the diagrams of the type shown in (10) through (14).

One mechanism for accomplishing this is to represent the diagrams as relations that must be constructed. Thus the entailment map, Ø, can be represented as a relation in extension to be constructed, R. Following the standard theory of PM,[1] 1_p is D(R) and 1_r is R(R). R therefore represents in extension the set of ordered pairs (p, r) that satisfy the relation R. Now if we interpret the contextual identity maps, ((C* ( )) as relative descriptions, we obtain the following diagram where $(\fint)$ is a set of intensions or relative properties of the relation taken in extension.

Consider the following relation, for example, R, interpreted as "son of." Suppose we have to find the domain of R given the range of R, some individual, J. We construct the following diagram, where the contextual identity map is represented as a relative description of the domain of R: by this we mean that D or R has a mapping to a set of relative properties ( ). This set of relative properties, in turn, has a mapping as a relative function of J. Preserving the structure of the C* diagrams, we say the (( ) R J) = ( ) $\text{Int Id}(\fint)$ o $(\fint)$ rf (J). This means that to instantiate D (R) we must model the set of intensions or properties of the individual or individuals comprising the set, D (R) such that that set of intensions or relative properties can function as a relative function of the known or given R (R), viz. the individual J. Thus we are required to identify the range of R as a relative of J, where J is the independent variable, such that D (R) = rf (J).

$$\begin{tikzcd}
(\ ) \arrow[rr, "C^*(R)"] \arrow[dr, "C^*\ \mathrm{Int\ Id}"'] & & J \\
& (\fint) \arrow[ur, "C^*(rf)"'] &
\end{tikzcd} \qquad (15)$$

where C* Int Id is an identity mapping from an object or set of objects in extension to a set of properties of that object or set of objects in intension, where $(\fint)$ is a set of intensions, and where C* (rf) is a relative function, mapping from the set of intensions to the given object, J.

Now suppose that C* (R) denotes the relation "biological son of." Suppose that we are given the correlative of C* (R) as an individual J. To

[1] Cf. Whitehead and Russell, Principia Mathematica, Oxford Univ. Press.

construct C* (R) we must find an individual that is the domain of C* (R). Suppose we can lead J into a room with a candidate set that is a candidate domain of R. We now make general and specific and perhaps even physiological measurements of J to construct a set of relative properties ( ) that are a function of J in the C* context of biological paternity. We next construct the mapping C* Int Id as a selection of a subset of the boys in the room that can be relatively described by the set of intensions ( ). This process is iterated until we select a single boy (assuming that we know beforehand that only one individual is in the domain of R). We designate this individual K. We now say that K (R) J iff K Int Id ( ) is a relative function of J. By this procedure we have constructed a relation in extension by intensional tests. Such an intensional construction procedure, when applied to the problem of MP and MT inference, using the diagrams of the C* Logic provides a construct that will lead to a machine procedure for hypothetical inference. This development will be continued during the next phase of this research.

2.5 Context Relation Logic

The theory of relations is an ancient logico-mathematical discipline. Its crucial importance in realizing advances in artificial intelligence was first recognized by W. S. McCulloch at MIT Research Laboratory for Electronics. McCulloch with R. Moreno-Diaz published some preliminary notes as a trail balloon to spur on a concerted effort to develop a useful calculus of relations for cybernetics.[1] Unfortunately, the attempt exclusively represented relations in extension (according to sets) and did not account for relations in intension (according to attributes, qualities, or characteristic relative properties of objects in extension). The notes were further weakened by a premature attempt to develope an n-adic calculus of relations using structures from the tensor and matrix calculi.[2]

The theory of relations probably originated with Aristotle,[3] was given a rigorous foundation by C. S. Peirce[4] and Schröder.[5] It was refined by

[1] R. Moreno-Diaz, W. S. McCulloch, "On a Calculus for Triadas," Artoga, Communication 101, May 1967.

[2] J. Simoes da Fonseca and J Mira, "A Calculus of N-Adic Relations" (Communication to W. S. McCulloch, April 9, 1969).

[3] Aristotle, Topics.

[4] C. S. Peirce, Collected Papers, Harvard University Press, Vol. III, 1960.

[5] E. Schröder, "Über die Algebra der binären Relative," Math. Annalen, 46, 1895, 144-158.

Russell and Whitehead,[1] and investigated in many forms by Tarski,[2] Wiener,[3] Lyndon,[4] Jonsson,[5] Birkhoff,[6] and others. This writer, while at NASA ERC, directed the work of J. Rial,[7] in an attempt to develop for the first time an intensional relation algebra.

Classical theories of relations (as well as newer treatments aimed at structures for machine intelligence) have treated relations exclusively in extension as sets or classes. The relations are also treated as context-free (CF). The present research is directed toward overcoming some of these shortcomings. In the present precis of context relation theory (which is a nucleus for a later in-depth formulation) relations are taken both extensionally and intensionally as constructible (by man or machine) through formal context operators that assign meaning or interpretation to relational structures.

2.5.1 Definitions

In the following development of a context relation logic as a basis for the future development of choreographic programming languages, definitions from PM (Principia Mathematica) of Russell-Whitehead will be used as a frame of reference and a means to introduce the context relation logic. P.M. definitions will be identified as CF (Context-Free) descriptions of relations in extension. The new developments will be identified immediately after each CF entry as C* (Context-Dependent).

[1] A. N. Whitehead and B. Russell, Principia Mathematica, Cambridge University Press, 1927.

[2] A Tarski, "On the Calculus of Relations," Journal of Symbolic Logic, Vol. 6, No. 3, September 1941.

[3] N. Wiener, "A Simplification of the Logic of Relatives," Proceedings of the Cambridge Philosophical Society, 17, 1914, Part 5, 387-390.

[4] R. Lyndon, "The Representation of Relation Algebras," Annals of Mathematics, Vol. 51, 1950, 707-729 also Annals, Vol. 63, 1956, 294-307.

[5] B. Jonsson, "Modular Lattices and Relation Algebras," Transactions of the American Mathematical Society, Vol. 92, No. 3, 1959, 449-464.

[6] G Birkhoff, Lattice Theory, American Mathematical Society Colloquium Publications, Vol. XXV, 1948.

[7] J. F Rial, Intensional Relation Algebras, The MITRE Corporation, Bedford, Mass., 1969 (prepared under NASA Contract No. NASA 12-644).

CF A dyadic relation is defined: $x \, R \, y = \hat{x} \, \hat{y} \{\Phi \, (x, y)\}$ i.e. "x has the relation R to y" denotes in extension the set, the x's and the y's such that $\Phi \, (x, y)$ or the couples that satisfy the relation, R.

C* A dyadic relation is defined: $x \, \mathcal{R} \, C^* \quad y = \hat{x} \, \hat{y} \{\bar{\Phi}_{C^*}(x, y)\}$ i.e. "x has the relation $\mathcal{R}$ to y in the context C*" denotes in extension a set, the x's and the y's such that $\bar{\Phi}$ C* (x, y), and such that $\bar{\Phi}$ C* is a relative mapping of x to y determined by the context, C*. The $\bar{\Phi}$ C* mapping is constructed via the following commutative diagram, where λ represents x taken intensionally and μ represents y taken intensionally.

$$\begin{tikzcd}
x \arrow[r, "\bar{\Phi}_{C^*}"] & y \arrow[d, "\text{INTIDC}^*"] \\
\lambda \arrow[u, "\text{EXTIDC}^*"] \arrow[r, "\text{REL FUNCT}", "C^*"'] & \mu
\end{tikzcd}$$

In this diagram, the range of $\bar{\Phi}$ C* is the domain of the contextually determined identity mapping, INT ID C.* The domain of $\bar{\Phi}$ C* is the range of a contextually determined identity mapping EXT ID C* from a set of intensions of relative properties of x, λ , or the domain EXT ID C*. REL FUNCT C* is a relative function mapping whereby the range of REL FUNCT C* is an independent variable, μ , and the domain of REL FUNCT C* is a set of properties or intensions of x, λ , that constitute a function, viz. $\lambda = f \, (\mu)$.

The $\mathcal{R}$ relation is defined in the context C* by the commutative diagram:

$$x \xrightarrow{\bar{\Phi} C^*} y_{,} = y \xrightarrow{\text{INT ID } C^*} \mu \circ \mu \xrightarrow{\text{REL FUNCT } C^{*-1}} \lambda \circ \lambda \xrightarrow{\text{EXT ID } C^*} x$$

$$x \xrightarrow{\bar{\Phi} \; C^*} y = x \xrightarrow{\text{EXT ID } C^{*-1}} \lambda \circ \lambda \xrightarrow{\text{REL FUNCT } C^*} \mu \circ \mu \xrightarrow{\text{INT ID } C^{*-1}} y$$

Consider an example $\mathcal{R}$ relation to be constructed.[2] Let C* be a "mount-wheel context." Suppose also we are given the hypothetical relation x $\bar{\Phi}$ y, i.e. x stands in the relation of turning y. Then we say that this relation must be constructed in the mount-wheel context, in order for the relation to hold. The construction procedure is as follows.

[1] Suppose that C* happens to be a mathematical context, and y stands for the number two. Then y in extension is the class of all sets, each of which stands in 1-1 correspondence to the set $\{0, 1\}$. Then y in intension (represented by μ) is the property of being even prime.

[2] A more detailed exposition of the construction procedure is given in Relative Product, 2.5.11.

The object in extension y is identified in C* with a specific set of intensions or properties, μ . Another (distinct) set of properties λ , is a function of μ , and is extensionally identified with an object x.

Thus if y in C* is given as a radially spaced set of projections, and y can be identified with properties such as "threaded" "two-inches long," "doughnut-shaped," "hexagonal," rotating clockwise," etc. and another distinct set of properties such as "having a clamping force of 10 pounds," "having a torque of 20 foot-pounds," "clamping opposing straight surfaces," rotating clockwise" etc. is a function of the first set of properties, and if we can extensionally identify the second relative property set with an object x, then we can construct the relation $\mathcal{R}_{C^*}$, "robot hand turning lug-nut on a wheel."

The important difference in the CF and C* forms of relations can be seen in this example. Construction of an $\mathcal{R}$ relation proceeds by hypothetical inference. In the classical CF form, the relation R is given as a set of ordered pairs. In the C* form, the relation is constructed by extensional and intensional tests, and by determining that one set of properties is a relative function of the second set. Such a procedure is the basis of machine perception in the present study. In the C* system, given one member of the relation $\mathcal{R}$ or given a relative in extension, the problem is to find a correlative in extension, such that they are $\mathcal{R}$ -related in the context C*. If the diagram can be constructed such that the diagram commutes, then the inference is made that the objects are $\mathcal{R}$ -related in C*. If the commutative diagram cannot be constructed, then they may be $\mathcal{R}$ -related in another context, D*, E*, F*......but not in the context, C*.

CF The antecedent of R is the object which has the relation R to something. The consequent of R is the object to which something has the relation R.

C* The antecedent of $\mathcal{R}$ is the object which in the context C* has the relation $\mathcal{R}$ to something by being a relative of something. The antecedent is a relative term because intensionally it is a relative function of something. The consequent of $\mathcal{R}$ is the object to which in the context C* something has the relation $\mathcal{R}$. The consequent is correlative term because intensionally it is an argument of something.

2.5.3 Relations Between Relations

CF -R is the complementary relation of R and is defined $\hat{x}\hat{y}(\sim x\mathcal{R}y)$

It denotes in extension all couples not joined by the relation R. For example, if R denotes "north of" (without specifying whether the context is geography or bridge or whatever), -R denotes all couples that are R-related in any other way except in a north-south relationship.

C* $-\mathcal{R}$ denotes in extension the set of all couples that are $\mathcal{R}$ -related by a different context, D*, E*, F*......but are not $\mathcal{R}$ -related in the context, C*. For example, if C* is the linear order relation, "longer than," then $-\mathcal{R}$, denotes a class of linear order relations determined by contexts other than C*, such as say, D*, "heavier then," E* "older than," F* "faster than,"etc. Now the class of linear order relations constitutes the contextual universe of $\mathcal{R}$ and each context is a Model of $\mathcal{R}$.[1] All models of $\mathcal{R}$ must satisfy the following where $\mathcal{L}$ is the set of linear ordered couples:

$$
\begin{aligned}
&(1)\quad (x, y) \in \mathcal{L} \Rightarrow x \neq y.\\
&(2)\quad (x, y) \in \mathcal{L} \Rightarrow (y, x) \notin \mathcal{L}\\
&(3)\quad (x, y) \in \mathcal{L} \qquad (y, z) \in \mathcal{L} \Longrightarrow (x, z) \in \mathcal{L}.
\end{aligned}
$$

Now to construct a relation $\mathcal{R}$, we must determine if the relation holds for a specific set (x, y) in a specific Model of $\mathcal{R}$, i.e. in a specific context, C*. Now if the relation diagram is not commutative (as shown in the following three types of non-commutativity), then we construct $-\mathcal{R}$. Thus $-\mathcal{R}$ connotes a shift in context, or a model mismatch, rather than the negation of $\mathcal{R}$, by the affirmation that x and y can be $\mathcal{S}$ -related, or $\mathcal{T}$ -related, etc.

[1] Here we are considering context as a model relative to a universe of models. More generally in the C* system, "model" refers to the relational maps of the C* relation diagram such that model is an ordered set denoted by Mod (B, X, h) read "B [illegible] models X relative to h," where (1) X is the set being modelled, (2) B is the modelling set, and (3) h is a map from X to B. Cf. Herman and Kotelly, op. cit. p. 273 ff.

$$\begin{tikzcd}
X \arrow[rr, dashed, "-\Phi c^*"] & & Y \arrow[d, dashed, "\text{INT ID C*}"] \\
\lambda \arrow[u, dashed, "\text{EXT ID C*}"] \arrow[r, dashed, "\overset{\text{REL FUNCT}}{-C^*}"] & \mu_j & \mu_i
\end{tikzcd}$$

- $\mathcal{R}$ Type 1: Diagram Does Not Commute

$$\begin{tikzcd}
X \arrow[rr, dashed, "-\Phi c^*"] & & Y \arrow[d, dashed, "\text{INT ID C*}"] \\
\lambda_i \arrow[u, dashed, "\text{EXT ID IC*}"] & \lambda_j \arrow[r, dashed, "\overset{\text{REL FUNCT}}{-C^*}"] & \mu
\end{tikzcd}$$

- $\mathcal{R}$ Type 2: Diagram Does Not Commute

$$\begin{tikzcd}
X \arrow[r, dashed, "-\Phi c^*"] & Y \arrow[d, dashed, "\text{INT ID C*}"] \\
\lambda_i \arrow[u, dashed, "\text{EXT ID C*}"] & \mu_i \\
\lambda_j \arrow[r, dashed, "\overset{\text{REL FUNCT}}{-C^*}"] & \mu_j
\end{tikzcd}$$

- $\mathcal{R}$ Type 3: Diagram Does Not Commute

CF $R\dot{\cup}S$ for $\hat{x}\hat{y}$ $(x\ R\ y \vee x\ S\ y)$. (union of R and S)

C* Case 1: $(R\dot{\cup}S)$is in a single context, C* such that $\lambda = f\ (\mu_i \vee \mu_j)$.

$$\begin{tikzcd}
y \arrow[d, "\text{INT ID } C^*"'] & x \arrow[l, "\Psi_{C^*}"'] \arrow[r, "\Phi_{C^*}"] & y \arrow[d, "\text{INT ID } C^*"] \\
\mu_j & \lambda \arrow[u, "\text{EXT ID } C^*"] \arrow[l, "\text{REL FUNCT } C^*"] \arrow[r, "\text{REL FUNCT } C^*"'] & \mu_i
\end{tikzcd}$$

C* Case 2: $(R\dot{\cup}S)$ is in different contexts, C* and D* Such that
$\lambda = f\ (\mu_i) \vee g\ (\mu_j)$

$$\begin{tikzcd}
y \arrow[d, "\text{INT ID } C^*"'] & x \arrow[l, "\Psi_{D^*}"'] \arrow[r, "\Phi_{C^*}"] & y \arrow[d, "\text{INT ID } C^*"] \\
{} & {} \arrow[u, "\text{EXT ID } C^* \vee D^*"] \arrow[l, "\text{REL FUNCT } D^*"] \arrow[r, "\text{REL FUNCT } C^*"'] & {}
\end{tikzcd}$$

CF $R\dot{\cap}S$ for $\hat{x}\hat{y}$ $(x\ R\ y \cdot x\ Sy)$. (intersection of R and S)

C* Case 1: $(R\dot{\cap}S)$ is in a single context, C* such that $\lambda = f\ (\mu_i \wedge \mu_j)$.

$$\begin{tikzcd}
x \arrow[r, "\Phi_{C^*} \wedge \Psi_{C^*}"] & y \arrow[d, "\text{INT ID } C^*"] \\
\lambda \arrow[u, "\text{EXT ID } C^*"] \arrow[r, "\text{REL FUNCT } C^*"] & \mu_i \wedge \mu_j
\end{tikzcd}$$

C* Case 2: $(R\dot{\cap}S)$ is in different contexts, C* and D* such that $\lambda = f\ (\mu_i \wedge \mu_j)$.

$$\begin{tikzcd}
x \arrow[r, "\Phi_{C^*} \wedge \Psi_{C^*}"] & y \arrow[d, "\text{INT ID } C^* \wedge D^*"] \\
\lambda \arrow[u, "\text{EXT ID } C^* \wedge D^*"] \arrow[r, "\text{REL FUNCT } C^* \wedge D^*"] & \mu_i \wedge \mu_j
\end{tikzcd}$$

CF $R \dot{\parallel} S$ for $\hat{x}\,\hat{y}$ (x R y $\mid$ s S y) (strong disjunction)

C* Case 1: $(\mathcal{R} \dot{\cap} \mathcal{S})$ is in a single context, C,* such that $\lambda_i = f(\mu_i)$ $\textcircled{v}$ $\lambda_j = f(\mu_j)$.

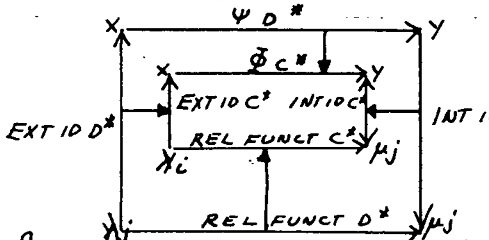
$$\begin{tikzcd}
x \arrow[r, "\Phi_{C^*} \textcircled{v} \Psi_{C^*}"] & y \arrow[d, "INT\ ID\ C^*"] \\
\lambda_i \textcircled{v} \lambda_j \arrow[u, "EXT\ ID\ C^*"] \arrow[r, "REL\ FUNCT\ C^*"] & \mu_i \textcircled{v} \mu_j
\end{tikzcd}$$

C* Case 2: $(\mathcal{R} \dot{\cap} \mathcal{S})$ is in different contexts, C* and D,* such that $\lambda_i = f(\mu_i)$ $\textcircled{v}$ $\lambda_j = g(\mu_j)$.

$$\begin{tikzcd}
x \arrow[r, "\Phi_{C^*} \textcircled{v} \Psi_{D^*}"] & y \arrow[d, "INT\ ID\ C^* \textcircled{v} D^*"] \\
\lambda_i \textcircled{v} \lambda_j \arrow[u, "EXT\ ID\ C^* \textcircled{v} D^*"] \arrow[r, "REL\ FUNCT\ C^* \textcircled{v} D^*"] & \mu_i \textcircled{v} \mu_j
\end{tikzcd}$$

CF $R \Subset S$ for $(\hat{x},\hat{y})$. x R y $\supset$ x S y. (containment)

C* Case 1: $(\mathcal{R} \dot{\cup} \mathcal{S})$ is in different contexts and C* is contextually nested in D,* viz. (C* $\subset$ D*) such that D* ⟶ C* or equivalently, Mod (B, X, h) in D* entails Mod (B, X, h) in C,* where (1) X is the set being modelled, (2) B is the modelling set, and (3) h is a map from X to B, written $X \xrightarrow{h} B$.



C* Case 2: $(\mathcal{R} \dot{\cup} \mathcal{S})$ is in a single context, C,* such that if $\mathcal{S}$ can be constructed, $\mathcal{R}$ can be constructed.

$$\begin{tikzcd}
x \arrow[r, bend left, "\Phi_{C^*}"] \arrow[r, "\Psi_{C^*}"'] & y \arrow[d, "INT\ ID\ C^*"] \\
\lambda \arrow[u, "EXT\ ID\ C^*"] \arrow[r, "REL\ FUNCT\ C^*"] & \mu
\end{tikzcd}$$

2.5.4 Universal and Null Relations

CF $\dot{\vee}$ for $\hat{x}$, $\hat{y}$ $(x = x \cdot y = y)$ (universal relation)

C* The universal relation is the zero-context or the CF relation of PM. Thus the universal relation in C* form is subject to the axioms, theorems, and laws of the relation x R y in PM. The universal C* relation denotes in extension a set (x, y) that is ;R-related by any possible context. As a result, its diagram always commutes, since all mappings are determined by an infinite disjunction of context operators, C,* D,* E,* F*.........

$$\begin{tikzcd}
x \arrow[r, "\Phi\dot{\vee}"] & y \arrow[d, "INT\ ID\ C^* \vee D^* \vee E^* \vee \cdots"] \\
\lambda \arrow[u, "EXT\ ID\ C^* \vee D^* \vee E^* \vee \cdots"] \arrow[r, "REL\ FUNCT", "C^* \vee D^* \vee E^* \vee \cdots"'] & \mu
\end{tikzcd}$$

CF $\dot{\wedge}$ for $\hat{x}\,\hat{y}$ $(x \neq x \cdot y \neq y)$ (null relation)

C* The null relation is the universal context relation in the C* system. The null relation denotes in extension a set (x, y) that is ℛ -related in every context. This set is empty since no ℛ-relation holds in every context. As a result, its diagram never commutes, since all mappings are determined by an infinite conjunction of context operators, C,* D,* E,*, F*...........

$$\begin{tikzcd}
x \arrow[r, dashed, "\Phi\dot{\wedge}"] & y \arrow[d, dashed, "INT\ ID\ C^* \wedge D^* \wedge E^* \wedge \cdots"] \\
\lambda \arrow[u, dashed, "EXT\ ID\ C^* \wedge D^* \wedge E^* \wedge \cdots"] \arrow[r, dashed, "REL\ FUNCT", "C^* \wedge D^* \wedge E^* \wedge \cdots"'] & \mu
\end{tikzcd}$$

2.5.5 Existence of a Relation

CF $\exists ! R$ for $(E \hat{x} \hat{y}) \cdot x R y$

C* The structure of the C* relation, $\mathcal{R}$, is the commutative diagram. When the diagram is verified as commuting by instantiation of the extensional place holders (x, y) and the intensional place-holders (λ, μ), $\exists ! \mathcal{R}_{C*}$.

Example: where C* is a "mount-wheel context."

$$\begin{tikzcd}
x=\text{robot-hand} \arrow[r, "\text{turner}_{C*}\text{(of)}"] & y=\text{ lug-nut} \arrow[d, "\text{INT ID } C*"] \\
\lambda = \{\text{rotating CW, 20 foot-lbs of torque, etc.}\} \arrow[u, "\text{EXT ID } C*"] \arrow[r, "\text{REL FUNCT } C*"] & \mu = \{\text{rotating CW, hexagonal, threaded, etc.}\}
\end{tikzcd}$$

The existence of a relation can be seen in more detail in 2.5.11 Relative Product where the reader can instantiate the extensional and intensional place-holders of the $\mathcal{R}$ -relation in the C* context of time, $x \Gamma y$, "x precedes y"

2.5.6 Individual Descriptions

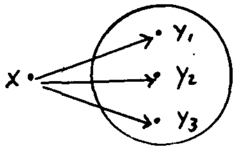
CF $R ' y$ for $(\text{℩} x)$ $(x R y)$ (read "R of y" is the individual description since it describes only one individual x which is R-related to one other object, y. Example: if "R" is the relation "author of" and y is "Cybernetics," $x = R ' y$= Norbert Wiener.

C* The CF individual description is the individual relative in C*. The antecedent of the relation $\mathcal{R}$ is an individual relative when x and y are individuals and the set of properties of y are an argument for nothing else but a set of properties of x. Such an individual relative is called restricted in C*.

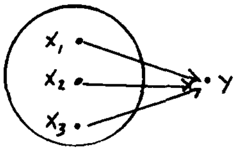
Plural Descriptions

CF $\overrightarrow{R}{}'y$ for $\hat{x}$ (x R y) (the class of objects which has the relation R to a given individual and $\overleftarrow{R}{}'x$ for $\hat{y}$ (x R y) (the class of objects to which a given individual has the relation R). Graphically:

$\overleftarrow{R}{}'x$ or Gs ' R



$\overrightarrow{R}{}'y$ or Sg ' R



C* The CF plural descriptions are either relatives or correlatives of $\mathcal{R}$, viz. antecedents or consequents of a relation $\mathcal{R}$, which are sets of individuals. The relative of $\mathcal{R}$ corresponds to the R plural description (sg ' R) when the set of properties of an individual y are an argument for nothing else but a set of properties of the class x. The correlative of $\mathcal{R}$ corresponds to the $\overleftarrow{R}$ plural description when the set of properties of an individual x are an argument for nothing else but a set of properties of the class, y. Such relatives and correlatives are called restricted in C*.

2.5.7 Converse of Relations

CF x $\breve{R}$ y for y R X. Cnv ' R for $\breve{R}$ (R-converse).

C* To construct the converse relation, $\breve{\mathcal{R}}$, the dual diagram of $\mathcal{R}$ is constructed. In the dual diagram, intensional and extensional identity maps are interchanged and REL FUNCT is changed to REL FUNCT $^{-1}$. The set of intensions μ , becomes a function of λ .

$$\mathcal{R} = \begin{tikzcd} x \arrow[r, "\Phi\ C^*"] & y \arrow[d, "INT\ ID\ C^*"] \\ \lambda \arrow[u, "EXT\ ID\ C^*"] \arrow[r, "REL\ FUNCT\ C^*"'] & \mu \end{tikzcd}$$

$$\begin{tikzcd} x \arrow[r, "\Phi^{-1}\ C^*"] \arrow[d, "INT\ ID\ C^*"'] & y \\ \lambda \arrow[r, "REL\ FUNCT^{-1}\ C^*"'] & \mu \arrow[u, "EXT\ ID\ C^*"'] \end{tikzcd}$$

CF $R \doteq S \doteq \breve{R} \doteq \breve{S}$

C* Two relations $\mathcal{R}$ and $\mathcal{S}$ are equal iff $\mathcal{R}$, $\mathcal{S}$ are constructed by the same diagram. If two relations are constructed by the same diagram, than their converses will be constructed by the same diagram. Hence their converses $\breve{\mathcal{R}}$ and $\breve{\mathcal{S}}$ will be equal.

CF Cnv ’ Cnv ’ R $\doteq$ R

C* Since $\breve{\mathcal{R}}$ is constructed by the dual diagram of $\mathcal{R}$, then by interchanging the identity maps and taking the inverse of REL FUNCT,[1] the resulting diagram is the diagram of $\mathcal{R}$. Since both diagrams are equal, then $\breve{\mathcal{R}} = \mathcal{R}$.

CF $\exists !$Cnv ’ R.

C* If the relation $\mathcal{R}$ exists (cf. section 2.4 on p. 31), then the converse, $\breve{\mathcal{R}}$ exists. Given the diagram of $\mathcal{R}$, the diagram of $\breve{\mathcal{R}}$ can be constructed. Given the diagram of $\exists ! \mathcal{R}$, the diagram of $\exists ! \breve{\mathcal{R}}$ can be constructed.

CF Cnv ’ (R $\dot{\cup}$ S) = Cnv ’ R $\dot{\cup}$ Cnv ’ S.

C* Case 1: ($\mathcal{R} \dot{\cup} \mathcal{S}$) is in the same context, C* The same diagram constructs Cnv ’ ($\mathcal{R} \dot{\cup} \mathcal{S}$) and Cnv ’ $\mathcal{R}$ $\dot{\cup}$ Cnv ’ $\mathcal{S}$.

$$\begin{tikzcd}
x \arrow[r, "\Phi^{-1} C^*"'] \arrow[d, "INT\ ID\ C^*"'] & {} & x \arrow[l, "\Psi^{-1} C^*"] \arrow[d, "INT\ ID\ C^*"] \\
\lambda_i \arrow[r, "REL\ FUNCT^{-1}"] & \mu \arrow[u, "EXT\ ID\ C^*"] & \lambda_j \arrow[l, "REL\ FUNCT^{-1}"']
\end{tikzcd}$$

Case 2: ($\mathcal{R} \dot{\cup} \mathcal{S}$) is in different contexts, C* and D.* The same diagram constructs Cnv ’ ($\mathcal{R} \dot{\cup} \mathcal{S}$) and Cnv ’ $\mathcal{R}$ $\dot{\cup}$ Cnv ’ $\mathcal{S}$.

$$\begin{tikzcd}
x \arrow[r, "\Phi^{-1} C^*"'] \arrow[d, "INT\ ID\ C^*"'] & {} & x \arrow[l, "\Psi^{-1} D^*"] \arrow[d, "INT\ ID\ D^*"] \\
\lambda_i \arrow[r, "REL\ FUNCT^{-1}"] & \mu \arrow[u, "EXT\ ID\ C^* \wedge D^*"] & \lambda_j \arrow[l, "REL\ FUNCT^{-1}"']
\end{tikzcd}$$

CF Cnv ' (R $\dot{\cap}$ S) = Cnv 'R $\dot{\cap}$ Cnv ' S.

C* Case 1 ($\mathcal{R}\dot{\cap}\mathcal{S}$) is in the same context, C.* The same diagram constructs Cnv'($\mathcal{R}\dot{\cap}\mathcal{S}$) and Cnv'$\mathcal{R}$ $\dot{\cap}$ Cnv'$\mathcal{S}$.

$$\begin{tikzcd}
X \arrow[r, bend left, "\Phi^{-1}_{C^*}"] \arrow[r, "\psi^{-1}_{C^*}"] \arrow[d, "INT\ ID\ C^*"'] & Y \\
\lambda i \vee \lambda j \arrow[r, "RELFUNCT^{-1}\ C^*"] & \mu i \vee \mu j \arrow[u, "EXT\ ID\ C^*"']
\end{tikzcd}$$

C* Case 2: ($\mathcal{R}\dot{\cap}\mathcal{S}$) is in different contexts, C* and D*. The same diagram constructs Cnv ' ($\mathcal{R}\dot{\cap}\mathcal{S}$) and Cnv'$\mathcal{R}$ $\dot{\cap}$ Cnv'$\mathcal{S}$.

$$\begin{tikzcd}
X \arrow[r, "\Phi^{-1}_{C^*}"] \arrow[r, bend right, "\psi^{-1}_{D^*}"'] \arrow[d, "INT\ ID\ C^* \wedge D^*"'] & \dot{Y} \\
\lambda i \wedge \lambda j \arrow[r, bend left, "RELFUNCT^{-1}\ C^*"] \arrow[r, "RELFUNCT^{-1}\ D^*"'] & \mu i \wedge \mu j \arrow[u, "EXT\ ID\ C^* \wedge D^*"']
\end{tikzcd}$$

CF Cnv - R = - Cnv R

C* - $\mathcal{R}_{C^*}$denotes in extension the set of all couples that are $\mathcal{R}$-related by a different context, D*, E*.......Z*, but are not $\mathcal{R}$ -related in the context, C*. Corresponding to -$\mathcal{R}$ is a set of diagrams that construct -$\mathcal{R}$. Corresponding to -$\check{\mathcal{R}}$ is a set of diagrams that construct -$\check{\mathcal{R}}$. Corresponding to (- $\mathcal{R}$) is the dual set of diagrams that construct (-$\check{\mathcal{R}}$). This set of diagrams is the same set that constructs - ($\check{\mathcal{R}}$). Therefore, since two relations are equal iff their diagrams are equal, Cnv'=$\mathcal{R}$ = - Cnv'$\mathcal{R}$.

CF $R = \check{S} = S = \check{R}$

C* Two relations are equal iff they are constructed by the same diagram. $\mathcal{R}$ then is constructed by the dual diagram of $\mathcal{S}$, and $\mathcal{S}$ is constructed by the dual diagram of $\mathcal{R}$.

CF $R \subset \breve{S} = S \subset \breve{R}$, $R \subset \breve{S}$ for $(\hat{x}, \hat{y})\ x\ R\ y \supset x\ \breve{S}\ y$ and $S \subset \breve{R}$ for $(\hat{x}, \hat{y})$ $x\ S\ y \supset X\ \breve{R}\ y$.

C* Case 1: $\breve{\mathcal{R}} \subset \mathcal{R}$ and $\breve{\mathcal{S}} \subset \mathcal{S}$ such that $\mathcal{R}$ is in C*, then $\breve{\mathcal{R}}$ is in C*; $\mathcal{S}$ is in D* then $\breve{\mathcal{S}}$ is in D*. Then if $\mathcal{R} \subset \breve{\mathcal{S}}$, $\mathcal{R} \subset \mathcal{S}$, and if $\mathcal{S} \subset \breve{\mathcal{R}}$, $\mathcal{S} \subset \mathcal{R}$. Hence $\mathcal{R} = \mathcal{S}$, and $\mathcal{R} = \breve{\mathcal{S}}$.

Case 2: $\mathcal{R}$ is in C*, and $\breve{\mathcal{R}}$ is in D*, then if $\mathcal{S}$ is in E* and $\breve{\mathcal{S}}$ is in F*, C* $\subset$ E* $\subset$ D*. (contextual nesting). Example of $\mathcal{R}$ is in C*, and $\breve{\mathcal{R}}$ is in D*: "The man is father (C*) to the boy." "The boy is father (D*) to the man."

Case 3: $\breve{\mathcal{R}} \subset \mathcal{R}$ such that if $\mathcal{R}$ is in C*, then $\breve{\mathcal{R}}$ is in C*, $\mathcal{S}$ is in E* and $\breve{\mathcal{S}}$ is in F*. Then C* $\subset$ F* and E* $\subset$ C*, then E* $\subset$ F*.

Case 4: $\mathcal{R}$ is in C* and $\breve{\mathcal{R}}$ is in D*, $\mathcal{S}$ is in E*, and $\breve{\mathcal{S}}$ is in E*, then C* $\subset$ E*, E* $\subset$ D*, then C* $\subset$ D*.

2.5.8 Domains and Fields

CF D'R for $\hat{x}\{(E y)\ x\ R\ y\}$ D'R is the domain of R, i.e. the set of objects which stand in the relation R to any object.

C* D'$\mathcal{R}$ is the set of objects in extension which in the context C* stand in the relation to an object. INT D'$\mathcal{R}$ (intensional domain of $\mathcal{R}$) is the set of properties, λ of D'$\mathcal{R}$ which in C* are a relative function of a set of properties of an object to which D'$\mathcal{R}$ is $\mathcal{R}$-related.

CF Ɑ'R for $\hat{y}\ \{(E x)\ x\ R\ y\}$ Ɑ'R is the converse domain of R, i.e. the set of objects to which any other objects stand in the relation R.

C* Ɑ'$\mathcal{R}$ is the converse domain of $\mathcal{R}$, or the set of objects in extension to which any other objects stand in the relation $\mathcal{R}$, in the context C*. INT Ɑ'$\mathcal{R}$ (intensional converse domain of $\mathcal{R}$) is the set of properties, μ, of Ɑ'$\mathcal{R}$, which in C* are a relative argument of set of properties of an object which is the D'$\mathcal{R}$.

CF C'R is the field of R, i.e. the logical sum of the domain and converse domain of R. C'R for D'R $\dot{\cup}$ Ɑ'R.

C* C'$\mathcal{R}$ is the field of $\mathcal{R}$ or the logical sum of the domain and converse domain of $\mathcal{R}$. C'$\mathcal{R}$ for D'$\mathcal{R}$ $\dot{\cup}$ Ɑ'$\mathcal{R}$ INT C'$\mathcal{R}$ is the logical sum of the INT D'$\mathcal{R}$ and INT Ɑ'$\mathcal{R}$.

CF $(\hat{x}, \hat{y})$: x R y $\supset$ x ϵ D'R $\cdot$ y ϵ Ɑ'R.

C* $(\hat{x}, \hat{y})$: x $\mathcal{R}$ y $\supset$ x ϵ D'$\mathcal{R}$ y ϵ Ɑ'$\mathcal{R}$, and λ ϵ INT D'$\mathcal{R}$$\cdot\mu\epsilon$INT Ɑ'$\mathcal{R}$.

CF (y) $\cdot$ $\overrightarrow{R}$'y $\subset$ D'R.

C* (y) $\cdot$ $\overrightarrow{\mathcal{R}}$'y in C* = D'$\mathcal{R}$ in C*.

CF (x) $\overleftarrow{R}$'x $\subset$ Ɑ'R.

C* (x) $\overleftarrow{\mathcal{R}}$' x in C* = Ɑ' R in C*.

CF C'R = C'$\breve{R}$.

C* C'$\mathcal{R}$ = C'$\breve{\mathcal{R}}$ and INT C'$\mathcal{R}$ = INT C'$\breve{\mathcal{R}}$ where both $\mathcal{R}$ and $\breve{\mathcal{R}}$ are in C*.[1]

CF D'R $\subset$ Ɑ'R = Ɑ'R = C'R.

[1] C* Condition for $\mathcal{R}$ and $\breve{\mathcal{R}}$ to be in the same context, C*: C'$\mathcal{R}$= C'$\breve{\mathcal{R}}$ and INT C'$\mathcal{R}$ = INT C'$\breve{\mathcal{R}}$.

C* $\mathrm{D}'\mathcal{R} \subset a'\mathcal{R} = a'\mathcal{R} = \mathrm{C}'\mathcal{R}$ and INT $\mathrm{D}'\mathcal{R} \subset$ INT $a'\mathcal{R}$ = INT $a'\mathcal{R}$ = INT $\mathcal{C}'\mathcal{R}$.

CF $a'\mathrm{R} \subset \mathrm{D}'\mathrm{R} = \mathrm{D}'\mathrm{R} = \mathrm{C}'\mathrm{R}$.

C* $a'\mathcal{R} \subset \mathrm{D}'\mathcal{R} = \mathrm{D}'\mathcal{R} = \mathrm{C}'\mathcal{R}$ and INT $a'\mathcal{R} \subset$ INT $\mathrm{D}'\mathcal{R}$ = INT $\mathrm{D}'\mathcal{R}$ = INT $\mathrm{C}'\mathcal{R}$.

CF $\mathrm{C}'\mathrm{R}$ is the field of R, i.e. the logical sum of the domain and converse domain of R. $\mathrm{C}'\mathrm{R}$ for $\mathrm{D}'\mathrm{R} \cup a'\mathrm{R}$.

C* $\mathrm{C}'\mathcal{R}$ is the field of $\mathcal{R}$ or the logical sum of the domain and converse domain of $\mathcal{R}$. $\mathrm{C}'\mathcal{R}$ for $\mathrm{D}'\mathcal{R} \dot{\cup} a'\mathcal{R}$. INT $\mathrm{C}'\mathcal{R}$ is the logical sum of the INT $\mathrm{D}'\mathcal{R}$ and INT $a'\mathcal{R}$

CF $(x, y): \mathrm{X}\,\mathrm{R}\,y \supset x \varepsilon \mathrm{D}'\mathrm{R} \cdot y \varepsilon a'\mathrm{R}$.

C* $(x, y): x\,\mathcal{R}\,y \supset x \varepsilon \mathrm{D}'\mathcal{R} \cdot y \varepsilon a'\mathcal{R}$, and $\lambda \varepsilon$ INT $\mathrm{D}'\mathcal{R} \cdot \mu \varepsilon$ INT $a'\mathcal{R}$

CF (y) $\overrightarrow{\mathrm{R}}'y \subset \mathrm{D}'\mathrm{R}$.

C* (y) $\overrightarrow{\mathcal{R}}'y$ in C* = $\mathrm{D}'\mathrm{R}$ in C*.

CF (x) $\overleftarrow{\mathrm{R}}'x \subset a'\mathrm{R}$.

C* (x) $\overleftarrow{\mathcal{R}}'x$ in C* = $a'\mathrm{R}$ in C*.

CF $C'R = C'\breve{R}$.

C* $C'\mathcal{R} = C'\breve{\mathcal{R}}$ and INT $C'\mathcal{R}$ = INT $C'\breve{\mathcal{R}}$ where both $\mathcal{R}$ and $\breve{\mathcal{R}}$ are in C*[1]

CF $D'R \subset a'R = a'R = C'R$.

C* $D'\mathcal{R} \subset a'\mathcal{R} = a'\mathcal{R} = C'\mathcal{R}$ and INT $D'\mathcal{R} \subset$ INT $a'\mathcal{R}$ = INT $D'\mathcal{R}$ = INT $a'\mathcal{R}$.

CF $a'R \subset D'R = D'R = C'R$.

C* $a'\mathcal{R} \subset D'\mathcal{R} = D'\mathcal{R} = C'\mathcal{R}$ and INT $a'\mathcal{R} \subset$ INT $D'\mathcal{R}$ = INT $D'\mathcal{R}$ = INT $c'\mathcal{R}$.

2.5.9 Relations with Limited Domains, Converse Domains, and Fields

CF $\alpha \upharpoonleft R$ for $\hat{x}\hat{y}\,(x \in \alpha \cdot x\,R\,y)$ denotes the relation R limited in its domain to the class α. $R \upharpoonright \beta$ for $\hat{x}\hat{y}\,(y \in \beta \cdot X\,R\,y)$ denotes the relation R limited in its converse domain to the class β $\alpha \upharpoonleft R \upharpoonright \beta$ for $\hat{x}\hat{y}\,(x \in \alpha \cdot y \in \beta \cdot x\,R\,y)$ denotes the relation R limited in its domain to the class α and in its converse domain to the class β. $R \upharpoonleft\!\downharpoonright$ for $\alpha \upharpoonleft R \upharpoonright \alpha$ denotes the relation R is limited in its field to the class α.

C* The domain, converse domain, and field of a relation $\mathcal{R}$, are limited inversely proportionately with the intensional domain, intensional converse domain, and intensional field of $\mathcal{R}$. The contravariance of intension and extension has been noted by logicians from Peirce to Carnap. It can be seen intuitively in an example relation. If the domain of a relation consists of a set of actuators, then its intensional domain may simply consists in

[1] C* Condition for $\mathcal{R}$ and $\breve{\mathcal{R}}$ to be in the same context, C*: $C'\mathcal{R} = C'\breve{\mathcal{R}}$ and INT $C'\mathcal{R}$ = INT $C'\breve{\mathcal{R}}$.

the properties of forces. If the domain is limited to electric actuators, then the intensional domain may include the properties of harmonic drive sets or linear induction motors.

2.5.10 <u>One-One Relations</u>

CF $1 \longrightarrow \mathrm{Cls}$ for $\hat{R}$ $\{(x, y, z)\colon x\ R\ z\ y\ R\ z \supset x = y\}$ denotes the one-many relation (restricted in its domain to unit classes or singletons). $\mathrm{Cls} \longrightarrow 1$ for $\hat{R}$ $\{(x, y, z)\colon x\ R\ y \cdot s\ R\ z \supset y = z\}$ denotes the many-one relation in which the converse-domain is restricted to unit classes. $1 \longrightarrow 1$ for $(1 \longrightarrow \mathrm{Cls}) \cap (\mathrm{Cls} \longrightarrow 1)$ denotes the one-one relation.

C* Restriction of the domain, converse domain, and field of a relation to a unit class or unit classes results from the selection of context. For example, in the context of industrial robots today, "controller of a robot hand" is a one-one relation, whereas in the context determined by future time-shared robots, "controller of a robot hand" will be a one-many relation. Another example can be taken from the context of biology. The relation, "father of" is one-many for the same father can have several children, but a child can have only one father. In the context of therapy of psychological guidance or counseling, the relation "father of" can be ordinary or many-many. Because extension and intension of a relation are contravariant, when a contextual shift occurs and a relation changes from ordinary or many-many to say, a one-one relation, the intensional field of the relation will expand. Shifts of context are characterized by corresponding expansions and contractions of extensional and intensional fields of a relation, expecially one a relation is transformed into one of the above relations involving unit classes.

2.5.11 <u>Relative Product</u>

CF $R|S$ for $\hat{x}\,\hat{z}\,(\mathcal{E}y)\ (x\ R\ y \cdot y\ S\ z) \supset x\ T\ z.$

C* The formation of relative product of relations is a critical operation in mechanizing hypothetical inference (as discussed in the First Quarterly Report). In its CF form, what is required is the existence of the converse domain of a first factor relation, R, and the existence of the same term as the domain of a second factor relation. This extensional requirement can be seen in the formation of relative product in the time-domain context. The operation is valid because in this context, the real-line is dense in itself, i.e. in extension there always exists a non-zero length interval of the real-line to guarantee the existence of the term that is the converse domain of the first factor relation (x R y) and the domain of the second factor relation (y S z). This holds for formation of linear relative products in the time-context, viz. where in the above definition of the CF relative product, x = z, relative products (x = z), whose formation does not violate properties of the real-line partial ordering.[1]

An analysis of the formation of relative product requires in addition to considering the properties of the field of the relation, an analysis of the intensional field of the relation. In the C* system, the formation or relative product is equivalent to constructing diagrams for the factor relations $\mathcal{R}$ and $\mathcal{S}$, then constructing a derivative diagram for their product relation $\mathcal{T}$.

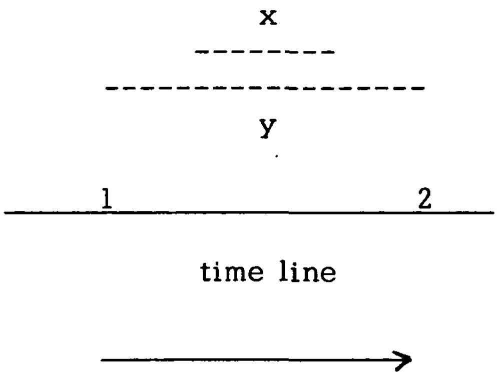
We shall consider an example from the time-context to demonstrate this procedure. We shall use a set of primitives or intensional properties of events or activities that were developed by this writer and L. Suyemoto to generate a time-relation algebra.[2]

A time relation "x is contained in y" or in symbols, $x \check{\Sigma} y$, is defined by the relative function (REL FUNCT), $\lambda (x) = \alpha^{-1}\beta(\mu(y))$, where λ is the set of properties in the time-domain context, C*, of x, and μ is the set of properties of y in the same context and where α^{-1} is an atomic time relation such that $p(y) < p(x)$, where p ( ) is the start-time of a non-zero length activity, β is an atomic time relation such that $q(x) < q(y)$, where q ( ) is the finish-time of a non-zero length activity. The REL FUNCT mapping between λ (x) and μ (y) is the conjunction of the functions $\alpha^{-1}\beta$.

So far, we have hpothesized that $x \check{\Sigma} y$, and we have defined a corresponding REL FUNCT map between the intensions or relative properites of x and y, which would hold if indeed $x \check{\Sigma} y$.

In the next step in constructing the relation diagram for $x \check{\Sigma} y$, we construct an INT ID map between y and its set of properties in C*, μ . Since μ is the independent variable, we select a value for μ relative to the time-line: $p(y) = 1$, $q(y) = 2$. (Note that $p(y) < q(y)$ by definition of y as a non-zero length activity). Then by REL FUNCT $= \alpha^{-1}\beta$, we derives values for x: $p(x) > 1$, $q(x) < 2$. The relation $x \check{\Sigma} y$ can now be constructed. First however, we visualize it by representing it relative to the time-line.

[1,2] Cf. L. C. Driscoll and L. Suyemoto, "Heuristics for Resolution of Logical Scheduling Conflicts," Proceedings of the Fourth International Conference on Operational Research, Ed. by D. B. Hertz and J. Meles, Wiley-Interscience, New York, 1966, 651-680.



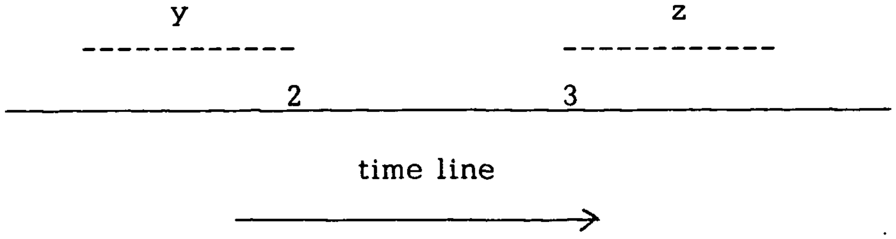
The first factor relation in the relative product is constructed by the diagram.

$$\begin{tikzcd}
x \arrow[r, "\check{\Sigma} C^*"] & y \arrow[d, "INT\ ID\ C^*"] \\
\lambda: p(x) > 1,\ q(x) < 2 \arrow[u, "EXT\ ID\ C^*"] \arrow[r, "REL\ FUNCT\ C^*", "\alpha^{-1}\beta"'] & \mu: p(y) = 1,\ q(y) = 2
\end{tikzcd}$$

For our second factor relation, we define a time relation, "y precedes z" or in symbols, y Γ z, by the REL FUNCT, λ (y) = δ (μ(z)), where δ is an atomic time relation such that q (y) < p (z). The REL FUNCT mapping between λ (y) and μ (z) is the function δ .

We now construct an INT ID map between z and its set of properties in C*. Since z is the independent variable, we again select a value for μ relative to the time-line: p (z) = 3, q (z) > 3 (since by definition, p (z) < q (z)). Then by REL FUNCT = δ ( ), we derive values for y: q (y) = 2, p (y) < 2.

The relation y Γ z can be represented relative to the time line, and can be constructed as a C* relation.



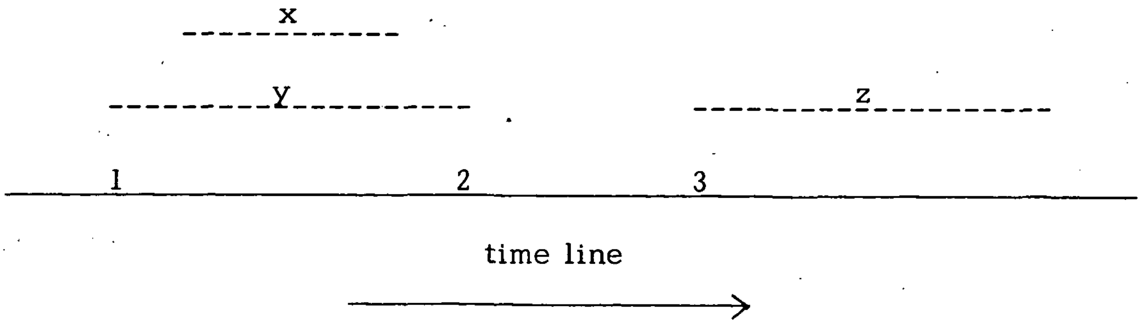
$$\begin{tikzcd}
y \arrow[r, dash, "\Gamma C^*"] \arrow[d, dash, "EXT\ ID\ C^*"'] & z \arrow[d, dash, "INT\ ID\ C^*"] \\
\lambda: q(y) = 2,\ p(y) < 2 \arrow[r, dash, "REL\ FUNCT\ C^*", "\delta"'] & \mu: p(z) = 3,\ q(z) > 3
\end{tikzcd}$$

The next step in the procedure of constructing the relative product of $\breve{\Sigma}$ and Γ is to construct an identity relation for the term that is the converse domain of the first factor relation and the domain of the second factor relation, viz. y. The reason for this step is purely formal: (1) to explicitly represent the intensional identity properties of y (2) to be consistent with the procedure of having λ (y) = REL FUNCT (μ(z)). In the C* context of time-relations, the identity relation is "y co-occurs with y" or in symbols, y Φ y. This identity relation is defined by the REL FUNCT =$\eta\theta$: p (y) = p (y), q (y) = q (y), p (y) < q (y).

At this point, we can construct the relative product diagram for the relative product of $\breve{\Sigma}$ and Γ. The C* values for the relative property sets of x, y, z can be listed: p (x) >1, q (x) < 2, p (y) = 1, q (y) = 2, p (z) = 3, q (z) > 3. The derived REL FUNCT mapping between λ (x) and μ (z) can now be determined since q (x) < p (z) satisfies the atomic relation δ, which it will be recalled defines the relation Γ. The existence of a REL FUNCT mapping between λ (x) and μ(z) can be seen in that:

$$\lambda\,(x) = (\text{REL FUNCT (REL FUNCT } (\mu(z)\ ))) = (\text{REL FUNCT } (\mu(z))).$$

The derived product relation between x and z is, xΓ z, "x precedes z." The relative product diagram is constructed as follows. First, we visualize it relative to the time line.



$$\begin{tikzcd}
x \arrow[r, "\breve{\Sigma}C^*"] \arrow[rrr, bend left, "\Gamma C^*"] & y \arrow[d, "INT\ ID\ C^*"] \arrow[r, "\bar{\Phi}C^*"] & y \arrow[r, "\Gamma C^*"] & z \arrow[d, "INT\ ID\ C^*"] \\
\lambda \arrow[u, "EXT\ ID\ C^*"] \arrow[r, "REL FUNCT\ C^*(\alpha^{-1}\beta)"'] \arrow[rrr, bend right, "REL\ FUNCT\ C^*(\delta)"'] & \mu \arrow[r, "REL\ FUNCT\ C^*(\eta\theta)"'] & \lambda \arrow[u, "EXT\ ID\ C^*"] \arrow[r, "REL\ FUNCT\ C^*(\delta)"'] & \mu
\end{tikzcd}$$

$\lambda(x): p(x) > 1,$ $q(x) < 2$

$\mu(y): p(y) = 1$ $q(y) = 2$

$\lambda(y): p(y) = 1$ $q(y) = 2$

$\mu(z): p(z) = 3$ $q(z) > 3.$

The C* relations in the time-context provide a simple but useful model for other contexts. The question naturally arises about the formation of relative products of relations in different contexts C* and D*. We know that intuitively we can form relative products of relations in different contexts. For example, "x is a lover of y (C* = romantic love), y is a lover of w (C* = romantic love), w is northeast of z" $\longrightarrow$ "z is southwest of a rival of x." This problem has been investigated by Rial[1] for different relation "types."

2.5.12 Ancestral Relation

CF her for $\hat{\alpha}$ $(\exists R)$ $(R``\alpha \subset \alpha)$. A class is "her" or hereditary with respect to the relation R ("her" is the class of hereditary relations), when the consequents of R in relation to the elements of α are also elements of α. Example: the class of people that can have sickle cell anemia is hereditary with respect to the relation of parents, for if x are the parents of y, and x belongs to the class of people that can have sickle cell anemia, then y belongs to that class also.

C* In the C* system, the antecedent of the relation $\mathcal{R}$ (via its intensionality or list of relative properties in the context C*) is a relative function of the consequent. In the converse form of the C* relation, the consequent in turn is a correlative function of the antecedent. The classical hereditary class of relations is a special case of the correlative being a relative function of the antecedent, such that if the mapping REL FUNCT^{-1} gives the consequent set membership in the set in which the antecedent is a member (as determined by the context C*), then the relation $\mathcal{R}$ is hereditary. The basic idea of the class of ancestral relations is that a class or a set is generated by the relation. In the C* system all sets are generated by relations since denoting set membership is itself a relation that must be constructed in the C* system. The C* "her" class of relations is a special case of a set being generated by a relation that generates a series.

CF R_* for $\hat{x}\hat{y}\,(x \in C'R :: (\alpha) : \breve{R}``\alpha \subset \alpha . x \in \alpha . \supset . y \in \alpha)$ gives precision to the concept $R^0 \dot{\cup} R^1 \dot{\cup} R^2 \dot{\cup} R^3$ etc. where R^0 is $I \upharpoonright C'R$ (identity restricted to the field of R, i.e. the relation of identity that each element of $C'R$ has to itself, and where R^1 is R, R^2 is $R | R$ (relative product of R and R), R^3 is $R^2 | R$ and R^n is $R^{n-1} | R$. This is the ancestral relation.

[1] J. F. Rial, op. cit. (Intensional Relation Algebras).

C* In the formation of relative products of series relations of the form $\mathcal{R}^{n-1} | \mathcal{R}$, the series is reduced by successively forming products of ternary strings. The product relation is defined by the functional string x = REL FUNCT (REL FUNCT (REL FUNCT (·. . . .z))). The assignment of meaning in the context C* to the resulting product relation results from evaluation of the nested functional expressions. The ancestral relation then is recursively evaluated in C* by evaluating the string of functionals. (CF 2.5.11 Relative Product).

2.5.13 First and Last Terms

CF B for $\hat{x} \hat{R} \{x \, \varepsilon \, D'R \cap - \text{G}'R$. "B" ("beginning") is the relation between the first term x of the series formed by R and R itself. This proposition states that x belongs to the domain, but not to the converse-domain of R. The class of first terms of R is sg'B'R, and that of the last terms is sg' B'Cnv'R.

C* In the C* system, all terms in a series relation of the form $\mathcal{R}^{n-1} | = \mathcal{R}^n$ are members of the intensional field of $\mathcal{R}$ except the domain of the first $\mathcal{R}$ (which is a member exclusively of the domain of $\mathcal{R}$ and the last term (which is a member exclusively of the converse domain of $\mathcal{R}$). Said another way, all terms in extension of the series relation are mapped onto a set of functionals λ_i in C* of $\mathcal{R}$, except the first term which is a function and the last term which is an argument μ_{n-1}. (Cf Relative Product in 2.5.11)

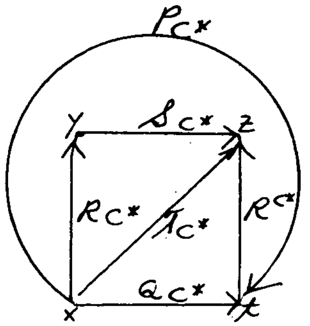
CF Min_R for $\hat{x} \hat{\alpha} \{x \, \varepsilon \, \alpha \cap C' \, R \cap - \breve{R} \, '' \alpha\}$, and Max_R for $\text{Min}_{\breve{R}}$. Min_R is B "beginning" restricted to one class. It is the minimum of this class with respect to R Max_R is the maximum.

C* In C* system, the first term is the function term, λ_1 of the series relation $\mathcal{R}$ n-1/ $\mathcal{R}$ and the last term is the argument term, μ_{n-1}. In the series relation $\breve{\mathcal{R}}^{n-1} | \breve{\mathcal{R}}$, the last term becomes the function term λ_1 and the first term becomes the argument term, μ n-1.

2.5.14 Isomorphic Relations

CF $R \dagger S$ for $R|S|\breve{R}$. The relation $R\dagger S$ is isomorphic: it holds between x and t when the following holds: $(Ey, z)\ x\,R\,y\ \ y\,S\,z\ \ z\,\breve{R}\,t \supset x\,P\,t$.

C* The "isomorphy" of $\mathcal{P} = \mathcal{R}\dagger\mathcal{S}$ simply reflects the fact that whatever the derived relation or relative product between x and t is in context C*, $\mathcal{P}^{c}*$, this relation must be the same or metaphorically identified with the relation, $\mathcal{Q}c*$, between x and t that may be given in the same context C*. This can be shown graphically: $x\,\mathcal{P}_{C^*}\,y$ is isomorphic to $x\,\mathcal{Q}_C{*}y$, where $\mathcal{P}_{C^*} = \mathcal{T}_{C^*} / \breve{\mathcal{R}}_{C^*}$.



The isomorphy of terms induced by the diagram above is more interesting. Since x is a first term in the relative product, λ(x) = functional of an argument of t; similarly, since t is the first term in the converse relative product, λ (t) = functional of an argument of x. As a result, x is related to t as co-functionals of each other's arguments. Example: "x is a lover of someone who is northeast of z," ⟶ "t is a lover of someone who is southwest of y." Also: "x is a lover of someone who is northeast of someone loved by t" ⟶ " t is a lover of someone who is southwest of someone loved by x." As a result, x is co-related to t (in extension) because of their intensional symmetry.

2.5.15 Reflexivity

CF refl for $\hat{R}$ ($R^{0} \subset R$): refl is the class of reflexive relations in which if an x belongs to the field of R and $R \subset \hat{R}$, then x R x holds.

C* Every context has nested within it an identity context which is the metaphorical identity context in relation to the context containing it. Consider the identity relation of the time algebra discussed in 2.5.11 Relative Product, page 52). The relation is a time-relation in the context C* of time. In that context, it is defined by REL FUNCT to be "x co-occurs with y" or "x and y are simultaneous." The identity context D* of C* is nested in C* (D* $\subset$ C*), and the relation in D* is defined by REL FUNCT as identity "x is identical with x." As a result, if REL FUNCT has the component functions P (x) = P (y), Q (x) = Q (y) it defines the relation $\overline{\Phi}$ as "co-occur with;" if REL FUNCT has the component functions P (x) < Q (y), P (x) = P (y), Q (x) = Q (y), then necessarily x = y, such that x $\overline{\Phi}$ x is interpreted in D* $\subset$ C*, x is identical with x (with respect to the context time, C*). The class of refexive relations in C* is co-extensive with the class of identity or metaphorical identity relations contextually nested in C*, D*, E*,..........

CF irr for $\hat{R}$ ($R^0 \subset$ - R). irr is the class of irreflexive relations in which if an x belongs to the field of R and R $\subset \hat{R}$, then if x R y, x $\neq$ y.

C* -$\mathcal{R}$ denotes in extension the set of all couples that are $\mathcal{R}$ -related by a different context, D*, E*, F*,.......but are not $\mathcal{R}$-related in the context, C*. (CF 2.5.2 <u>Relations between Relations</u>). The class of irreflexive relations in the C* system is co-extensive with the class of relations that do not have an identity context in C*, but have an identity context in D*, E*, F*.......Example: if C* is the context of parental love and is the relation "lover of," then if x $\mathcal{R}_{C*}$ y, if x belongs to the field of $\mathcal{R}_{C*}$, then x $\neq$ y. If a relation does not have an identity context nested in a given context C*, then this results from the condition that it does not have an intensional identity context nested in C*.

CF R ε refl = (x): x ε C ' R $\supset$ x R x and R ε irr = (x) $\sim$ x R x.

C* $\mathcal{R} \varepsilon$ refl iff the INT C'$\mathcal{R}$ is such that REL FUNCT defines an identity in C* between the domain of $\mathcal{R}$ and the converse domain of $\mathcal{R}$. $\mathcal{R} \varepsilon$ irr iff INT C'$\mathcal{R}$ is such that REL FUNCT does not define an identity in C* between the domain and converse domain of $\mathcal{R}$. Note: if reflexivity and irreflexivity are taken as context-free, then there exist ambiguous relations that can be both reflexive and irreflexive and are usually treated as neither reflexive or irreflexive. An example is this is "x cooks food for y." However, if context is taken into account, then this fuzzy class is eliminated. Thus in the context determined by automation "x cooks food for y" would require that the relation "cook," $\mathcal{R}$, is irreflexive on the assumption that x is a robot (x $\neq$ y). In the context determined by the sole survivor of an air crash in a life-raft, "x cooks food for y" would require that $\mathcal{R}$ is reflexive (x = y).

2.5.16 Symmetry

CF sym for $\hat{R}$ ($\breve{R} \doteq R$). 'as' for $\hat{R}$ ($\breve{R} \doteq - R$). sym is the class of symmetrical relations; as is the class of asymmetrical relations.

C* A relation $\mathcal{R}$ is symmetrical in a context C*, iff INT $C'\mathcal{R}$ = INT $D'\mathcal{R}$ = INT $a'\mathcal{R}$ i.e. x $\mathcal{R}$ y is symmetrical in C* if REL FUNCT = REL FUNCT^{-1} in C*. A condition for this is that $\lambda(x) = \mu(y)$, in C*. A relation $\mathcal{R}$ is assymetrical in a context C*, iff INT $C'\mathcal{R}$ = INT $D'\mathcal{R}$ = INT $a'\mathcal{R}$ i.e. x $\mathcal{R}$ y is assymetrical in C* if REL FUNCT $\neq$ REL FUNCT^{-1} i.e. $\lambda(x) \neq \mu(y)$ in C*.

CF R $\mathcal{E}$ sym = (x) x R Y = y R x. R $\mathcal{E}$ as (x) x R y $\overset{\sim}{=}$ y R x.

C* In the contexts of ancestral relations and in the context of linear order relations, amny relations are clearly assymetrical. For example, "x is greater than y," "x is smaller than y" "x is the father of y." Other contexts yield symmetrical relations "x is a neighbor of y," "x is a colleague of y." There exists also in CF logic, relations that are taken as neither symmetrical or assymetrical. For example, the relation "listen" is summetrical or assymetrical when context is taken into account and neither in a context-free system. In the context determined by a telephone conversation "x listens to y" the relation $\mathcal{R}$ listen is symmetrical. In the context determined by a television set, $\mathcal{R}$ is asymmetrical.

2.5.17 Transitivity

CF trans for R ($R^2 \subset R$) intr for R ($R^2 \subset -R$). A relation is transitive iff $R \,|\, R \subset R$.

C* A transitive relation $\mathcal{R}^2 \subset \mathcal{R}$ in C* is a relation restricted to contexts connoting identification, metaphoric identification, likeness, model, order, etc. The inclusion of REL FUNCT as a functional in REL FUNCT as a function defines the inclusion of the powers of a relation $\mathcal{R}$ in a relation $\mathcal{R}$. In REL FUNCT of a transitive relation, the property sets λ and μ are identified, metaphorically identified, compared, modelled, ordered, etc. A contemporary way to view the transitive relation class is to identify it with the context of modelling. Thus we have $x \mathcal{R} y \wedge y \mathcal{R} z \supset x \mathcal{R} z$ iff μ (z) models μ (y) and μ (y) models λ (x) such that μ (z) is a modeller of a modeller of λ (x) ⟶ μ (z) is a modeller of λ (x). The INT C' $\mathcal{R}$ if $\mathcal{R}$ is a transitive relation consists in relative modelling properties of objects in extension. The chain of functionals being included in the function induces an identity on the λi and μi by recursion. For where REL FUNCT is symbolized as f, λ (x) = f (f(z)). But if f is a modelling function (defining a transitive relation), λ (x) = f (z). Therefore λ (x) = f (λ (x)) and λ(x) = f (f (λ (x))) etc.

CF R $\mathcal{E}$ trans = (x, y): (E z) x R z • z R y $\supset$ x R y. R $\mathcal{E}$ intr = (x, y, z(x R y • y Rz $\supset \sim$ x R z.

C* The linear order relations and all relations expressing equality and identity are clearly of the class of transitive relations. In CF, however, there are relations which are taken as neither transitive or intransitive. The relation "love" is one of these. However, if "love" is put into a modelling context it can be verified as a transitive relation. For example: "x is a lover of everybody loved by y and y is alover of everybody loved by z" ⟶ "x is a lover of everybody loved by x." In intension, z models y, y models x ⟶ z models s.

CF trans $\cap$ sym $\subset$ refl.

C* The intersection of the classes trans and sym is the class of relations, $\mathcal{R}^2 \subset \breve{\mathcal{R}}$ The class of reflexive relations is the class in which $\mathcal{R}^0 \subset \mathcal{R}$ where $\mathcal{R}^0$ is I $\restriction$ C'$\mathcal{R}$. (identity restricted to the field of $\mathcal{R}$, i.e. the relation of identity that each element of C'R has to itself). As a result, trans $\cap$ sym defines a class of relations restricted to contexts of equality, similarity, metaphorical identity, identity, which are nested within the individual identity context. The individual identity context is nested within every context, such that the identity is a metaphorical identify relative to the context containing it. (Cf. 2.5.15 Reflexivity). The CF law above then expresses in C* the containment of contexts: where trans $\cap$ sym is restricted to C* as a similarity model of $\mathcal{R}$, where refl is restricted to D* as a metaphorical identity model of $\mathcal{R}$, and where E* is a context that is any arbitrary model of $\mathcal{R}$: C* D* E*.

CF as $\subset$ irr.

C* as is defined as $\hat{\mathcal{R}}$ ($\breve{\mathcal{R}} = -\mathcal{R}$). irr id defined as $\hat{\mathcal{R}}(\mathcal{R}^0 \subset -\mathcal{R})$ such that $\mathcal{R} \in$ as = (x) x $\mathcal{R}$y in C* =$\sim$y $\mathcal{R}$x in C* and $\mathcal{R} \in$ irr = (x) $\sim$ x R x in C*. $\mathcal{R}$ is assymetrical in C* iff INT C'$\mathcal{R}$ $\neq$ INT D'$\mathcal{R}$ $\neq$ IND σ'$\mathcal{R}$ i.e. λ (x) $\neq$ μ(y). in C* $\mathcal{R}$is irreflexive in C* iff λ (x) $\neq$ REL FUNCT μ(x) in the identity context of C*. $\mathcal{R}$is assymetrical in C* iff λ (x) = REL FUNCT μ (y) $\longrightarrow$ λ (y) $\neq$ REL FUNCT μ (x). Now λ (x) $\neq$ REL FUNCTμ(x) in the identity context of C* $\longrightarrow$ - R in C*, and R in D*, E*, F*......... (cf. 2.16 p.) Similarly, λ (y) $\neq$ REL FUNCT μ(x) in C* $\longrightarrow$ - R in C*, and R in D*, E*, F*.......If we denote irr as Mod_{id} as the set of contexts or models of $\mathcal{R}$ in which $\mathcal{R}$is reflexive, and if we denote Mod_{sym} as the set of contexts or models of $\mathcal{R}$in which $\mathcal{R}$is symmetrical then $\text{Mod}_{id} \subset \text{Mod}_{sym}$, $\therefore$ - $\text{Mod}_{sym} \subset -\text{Mod}_{id}$. Therefore in C*, as $\subset$ irr.
Counter example: In the complete lattice of the time relation algebra referenced in 2.5.11, there are 28 time relations or time relation contexts of time relations that can hold between two activities. Of this number, 7 are reflexive, 21 are irreflexive, 6 are symmetrical, 22 are asymmetrical. As a result, the basis for this proposition in CF and C* requires further study.

2.5.18 Similarity and Equality

CF sim for sym $\cap$ refl. The class of relations sim is the class of relations of similarity, i.e. "nearly the same," "approximate," "nearly equal," etc. All such are symmetrical and reflexive.

C* A relation $\mathcal{R}$ is symmetrical iff REL FUNCT = REL FUNCT^{-1} and relexive iff INT C'$\mathcal{R}$ is such that REL FUNCT defines an identity in C* between the domain of $\mathcal{R}$ and the converse domain of $\mathcal{R}$. This identity is a metaphorical identity is most reflexive relations (viz. an identity context which is metaphorical through the C* relation). The class of relations $\mathcal{R}$ is sim iff REL FUNCT = REL FUNCT^{-1} and REL FUNCT defines a metaphorical identity between D'$\mathcal{R}$ and $a'\mathcal{R}$.

CF aeq for trans $\cap$ sym. is the class of relations of equality, i.e. of the same form, size, color, etc. These relations are transitive and symmetrical.

C* A relation $\mathcal{R}$ is aeq iff for $x\mathcal{R}y$ and $y\mathcal{R}z$, μ(z) models μ (y) and μ(y) models λ (x) such that μ(z) is a modeller of λ(x), in C*, and REL FUNCT = REL FUNCT^{-1} in C*.

CF aeq$\subset$refl and aeq$\subset$sim

C* If a relation $\mathcal{R}$ is irreflexive, i.e. does not have a metaphorical identity context in C*, but has a metaphorical identity context in D*, E*, F*...... and we denote this set of contexts Mod$_{\text{Id}}$ and if a relation $\mathcal{R}$ is not aeq (i.e. not trans and sym) and we denote the set of contexts for which $\mathcal{R}$ is not aeq. as Mod$_{-\text{aeq}}$ then Mod$_{\text{id}}$ $\subset$ Mod$_{-\text{aeq}}$. and Mod$_{\text{aeq}}$ $\subset$ Mod$_{\text{refl}}$. Similarly, if a relation is irreflexive in C* and assymetrical in C*, and we denote the set of contexts in which it is reflexive and symmetrical as Mod$_{\text{id}}$ $\cap$ Mod$_{\text{sym}}$ and we denote the set of contexts for which it is not aeq. as Mod$_{\text{aeq}}$, then Mod$_{\text{id}}$ $\cap$ Mod$_{\text{sym}}$ $\subset$ Mod$_{-\text{sim}}$ and Mod$_{\text{sim}}$ $\subset$ Mod$_{\text{id}}$ $\cap$ Mod$_{\text{sym}}$.

2.5.19 Connexity

CF connex for $\hat{\breve{R}}$ (J $\downharpoonleft$ C' R $\subset$ R $\dot{\cup}$ $\breve{R}$). A relation R is "connex" or connected when R or $\breve{R}$ always holds between any two objects in the field of R.

C* In the time-relation algebra referenced in 2.5.11, the relation $\mathcal{R}$: $x \mu y$, "x finishes after y starts" or the relation $\breve{\mathcal{R}}$; $x \breve{\mu} y$, "y finishes after x starts," can be constructed between any two activities for which we can construct time relations. If we consider each of the 28 time relations in this algebra as a model of time relations, or a context in which we model time relations, then the μ-context is the connex context of time.

CF ser for irr $\cap$ trans $\cap$ connex. A relation generates a series when it is irreflexive, transitive, and connex.

C* A relation $\mathcal{R}$ is ser or generates a series iff REL FUNCT of $\mathcal{R}$ does not define an identity in C* between D' $\mathcal{R}$ and $\text{G}'\mathcal{R}$, and where $x \mathcal{R} y \wedge y \mathcal{R} z, \supset x R z$, in C*, $\mu(z)$ models $\lambda(x)$, and $x \mathcal{R} y$ or $x \breve{\mathcal{R}} y$ can be constructed between any objects in the field of $\mathcal{R}$, in C*. All models of $\mathcal{R}$ must satisfy the following, where $\mathcal{S}$ is a series:

(1) $(x, y) \in \mathcal{S} \longrightarrow x \neq y$
(2) $(x, y) \notin \mathcal{S} \longrightarrow (y, x) \in \mathcal{S}$
(3) $(y, x) \in \mathcal{S} \longrightarrow x \neq y$
(4) $(x, y) \in \mathcal{S}$, $(y, z) \in \mathcal{S} \longrightarrow (x, z) \in \mathcal{S}$
(5) $(y, x) \in \mathcal{S}$, $(x, z) \in \mathcal{S} \longrightarrow (y, z) \in \mathcal{S}$

CF ser $\subset$ irr, ser $\subset$ as

C* The context of series is contained in the context of relations that do not express metaphoric identity as well as the context of relations where REL FUNCT $\neq$ REL FUNCT^{-1}.

CF $R \in$ Ser $= \breve{R} \in$ ser

C* If $\mathcal{R}$ is irr, tran, and connex, $\breve{\mathcal{R}}$ is irr, tran, and connex.

2.5.20 Definition of Triadic Relation

No CF definitions will be given since there does not exist a CF calculus of polyadic relations. The present C* definition is based on Peirce and is an extension of the C* system.

A triadic relation is defined as an ordered triple of relative functions ($\mathbf{x}$, σ, π) whose arguments form an ordered triple of terms (x, y, z) such that the following diagrams commute in C*. It is symbolized:

$$R\ C^*\ (\mathbf{x}\ (y,z),\quad \sigma(x,\ z),\ \pi(x,\ y))\ \text{or}\ RC^*\ (\mathbf{x},\sigma,\pi).$$

Diagram 1: x is a term in extension that has an INT ID map to the conjugative relative $\mathbf{x}$ which is a function of two arguments, y an element in extension and z an element in extension. The map REL FUNCT 1 is between x and y: $\mathbf{x}$ = REL FUNCT 1 (y). The map REL FUNCT 2 is between $\mathbf{x}$ and z: $\mathbf{x}$ = REL FUNCT 2 (z).

Note: what distinguishes the triad from the dyad is the functions $\mathbf{x}$, σ, π each require two arguments for their evaluation.

Interpretation: $\mathbf{x}$ is a function such that x is an intermediary between y and z. Example: $\mathbf{x}$ = giver of y (REL FUNCT 1); $\mathbf{x}$ = giver to z (REL FUNCT 2). Thus RC* ($\mathbf{x}$,σ, π) is interpreted "x is a giver of y to z." When y and z, the arguments, are determined, then the context in which "x is a giver of y to z" is determined. Thus if y is a gift and z is a hospital, then x is a benefactor. If y is a ticket and z is a motorist, then x is a law-enforcer, etc.

The triad generates two dyads x R y and x S z. The triad is not reducible to these dyads because $\mathbf{x}$ is simultaneously a function of two arguments and not two distinct functions of two arguments. The sub-diagrams of the two dyads commute only as parts of the overall diagram and not separately. The reason for this is that the context in which x stands in the relation R to z and x stands in the relation S to y is defined by the double REL FUNCT maps, <u>each of which is outside the other's diagram</u>.

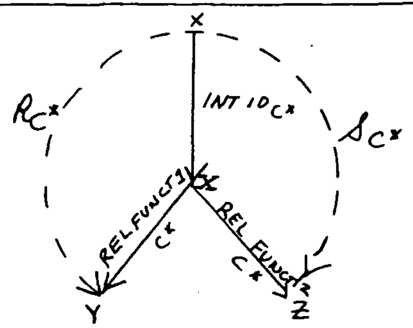


Diagram 1 Construction of triad R C* ($\mathbf{x}$,σ,π)

Diagram 2 is a cyclic permutation of Diagram 1. z is a term in extension that has an INT ID map to the conjugative relative σ which is a function of two arguments, x, and element in extension, and y, an element in extension. The map REL FUNCT 3 is between σ and x: σ = REL FUNCT 3 (x). The map REL FUNCT 4 is between σ and y: σ = REL FUNCT 4 (y).

Interpretation: σ is a function that is intermediary between x and z. Diagram 2 is a cyclic permutation of Diagram 1. Therefore, its interpretation is in the context determined by REL FUNCT 1 and REL FUNCT 2 of diagram 1. Diagram 1 was interpreted "x is a giver of y to x." Then Diagram 2 is interpreted: "z is a receiver of y from x." If the context of Diagram 1 was determined by REL FUNCT 1 and REL FUNCT 2 to be philanthropy, then if y is a check and x is the President of IBM, by REL FUNCT 3 and by REL FUNCT 4, z is the Director of the Hospital.

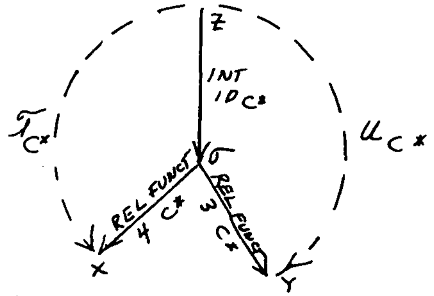


Diagram 2 Construction of Cyclic Permutation of Triad R C* (x, σ, π)

Diagram 3 is a cyclic permutation of Diagram 1. In this permutation y is a term in extension that has an INT ID map to the conjugative relative π which is a relative function of two arguments, x and element in extension and z, an element in extension. The map REL FUNCT 5 is between π and x; π = REL FUNCT 5 (x). The map REL FUNCT 6 is between π and z: π = REL FUNCT 6 (z).

Interpretation: π is a function that is intermediary between x and z. Diagram 3 is a cyclic permutation of Diagram 1. Preserving the context defined by REL FUNCT 1, REL FUNCT 2, REL FUNCT 3, and REL FUNCT 4, then Diagram 3 is interpreted: "y is a gift from x to z." If the context has been more precisely determined by REL FUNCT 3 and REL FUNCT 4 to be public relations, then if x is the Secretary of the President of IBM, and z is a 3 - year old child in-patient at the hospital, by REL FUNCT 5 and by REL FUNCT 6, y is a teddy bear with a check pinned to its ribbon.

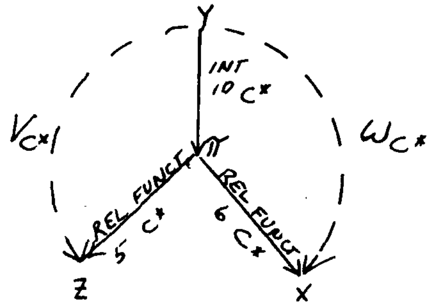


Diagram 3. Construction of Cyclic Permutation of Triad R C* (x, σ, π)

The triad offers a richer structure for hypothetical inference, but is described here to suggest that the C* system is a general theory of relations and not restricted to dyadic relations. Full development of a working calculus of polyadic relations will await a separate research program.

SECTION III. TOPOLOGICAL VISION MODEL

3.1 Object Identification by Recognition of Shape

3.1 Approach

In the original proposal for this study, it was specified that the topological vision model would be based on the EIA standard 525 line vidicon TV system. The present model is based on more realistic system elements that are likely to be specified for the Viking missions to Mars. We shall assume that a robotic system will have a multiaperture image dissector television camera and an image processing laser system. The image dissector will prove adequate for conventional TV functions since it can be used in a scanning mode to produce a normal television image. Its compelling advantage in providing an auxiliary "eye" for a robotic system is its capability to provide edge-tracking and shape-detection. This will be a significant advantage since it will reduce the complexity of the computer environment as well as the software and hardware/software integration. A laser ranging system will similarly reduce computer requirements since it can provide 3-dimensional information directly to the topologist program. The combination of the image dissector and laser is an attractive system solution for realizing computer vision via the class of computers that can be expected for Viking missions.

3.1.2 Multiaperture Image Dissector

Research at MIT for detecting edges and recognizing shapes has relied on complex programs. [1,2,3,4,5,6,7]. Although immensely valuable as studies of pre-processing of image data for subsequent laboratory scene-analysis, these approached do not appear scaled to the requirements of an autonomous or semi-autonomous robotic system for Martian surface exploration.

The multiaperture image dissector and follow-on developments such as the "Dissecticon" developed by Bendix for the Navy are ideally suited to

[1] H. Blum, "A Transformation for Extracting New Descriptions of Shape," Models for the Perception of Speech and Visual Form, W. Wathen-Dunn (Ed.), MIT Press, Cambridge, 1967.

[2] A. Herscovits, "On Boundary Detection, "Project MAC AI Memorandum 183, 1970.

[3] A. Guzman-Arena, MAC TR-59, "Computer Recognition of Three-Dimensional Objects in a Visual Scene," December, 1968.

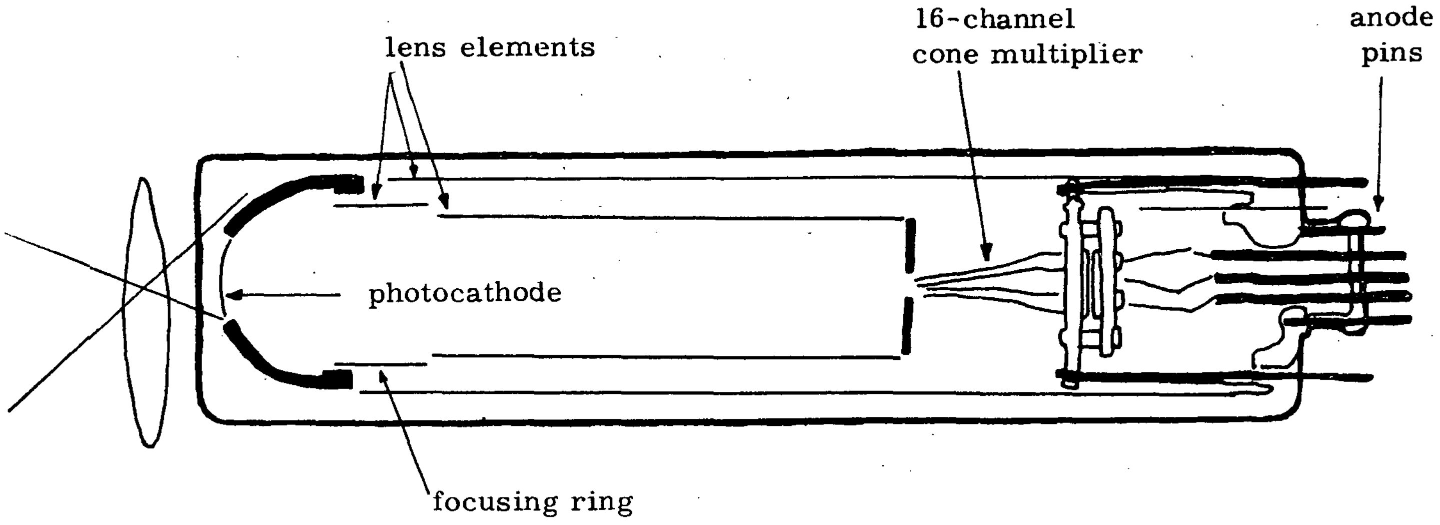


Figure 3-1. Diagram of Multiaperture Image Dissector Tube

limited computing environments. By integrating these cameras with edge-detection and shape-detection logic, they will simplify robot vision on Mars. In effect, functions usually performed by a computer can be expected to be performed in the electro-optics and associated logic.

The multiple aperture image dissector is an electrostatically focused, magnetically deflected image dissector. An image-dissector is a television camera tube in which an image projected on the photocathode produces electrons that are focused through a deflection system so that the current from a selected point of the image plane falls through a small hole and is measured.

The main features of the multiple aperture image dissector tube are shown in the following diagram. A lens projects an optical image on a curved photo-cathode. The emitted electrons are distributed in accordance with the light intensity at each part of the image. The electron-optical image is then focused onto a flat metal plate at the rear of the lens. The focus ring varies the positive voltage to obtain a sharp focus. The flat plate has a small aperture behind which channel electron multipliers are inserted. In our envisioned application, there will be sixteen multipliers arranged in a four-by-four matrix. Each of the 16 multipliers receives an electron current proportional to the light intensity of the projected image. Each channel multiplier amplifies current falling on it independently of the other multipliers. Sixteen separate anodes collect the multiplied currents. The matrix can receive the electron image from any part of the photocathode by magnetic deflection fields.

The multiaperture tube has an interesting property that can be used to detect edges and shapes. The device can look at adjacent parts of an image simultaneously with or without a scanning field applied.

[4] A. Herscovits and T.O. Binford, "On Boundary Detection," Project MAC AI Memorandum 183, July 1970.

[5] B. Horn, "Shape from Shading: A Method for the Determination of the Shape of a Smooth Opaque Object from One View," MAC-TR-79, Nov. 1970.

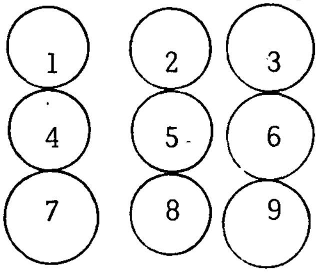
[6] A. K. Griffith, "Computer Recognition of Prismatic Solids," MAC-TR-73, Aug. 1970.

[7] L. G. Krakauer, "Computer Analysis of Visual Properties of Curved Objects," MAC-TR-82, May 1971.

The tube produces signals proportional to the rate of change of intensity in any direction. Signals from two adjacent apertures can be differenced and the difference between these two outputs will be proportional to the rate of change of intensity with distance, i.e. dI / d X. This difference is a function of the intensity differential and is independent of scanning speed. Two apertures at right angles will produce a signal corresponding to the differential d I / d Y.

Bendix Research Laboratories, Southfield, Michigan, has developed a simple edge enhancement system based on this property. This is shown in the simple examples of differential images of a checkerboard input image. The checkerboard pattern is shown in Figure 3-2 using the signal from one aperture in a conventional scan mode. Figures 3-3 and 3-4 show the differential image for horizontally adjacent and vertically adjacent apertures, respectively. Figure 3-5 shows the results of processing the differential signals in a "window gate" which in effect adds the gate signals from both directions.

Bendix also found that the tube could be used to detect shapes. In this sense, we could characterize it as a "line-finder." In this mode, when the tube is scanned across a scene, logic circuitry would detect line patterns. Bendix found that with a 3 X 3 aperture format, it could devise logic circuits to detect bright vertical lines in an image when channels 2, 5, and 8 are bright and all other channels are dark.



Bendix 3 X 3 Aperture Format

This result can be extended to find black-to-white and white-to-black transitions in an image that generate a line with a direction. The multichannel matrix will require more elements than a 3 X 3 format. If we quantize line directions and use a 4 X 4 aperture format, the tube can with associated logic provide a detection of shapes that can be encoded in accordance with the activation of the channels. A sequence of these codes can be used to build up a complete object shape over successive selective scans in which different lines are found by the combination of channels.

3.1.3 Shape Encoding

The multiaperture tube's capability to detect edges and shapes suggests

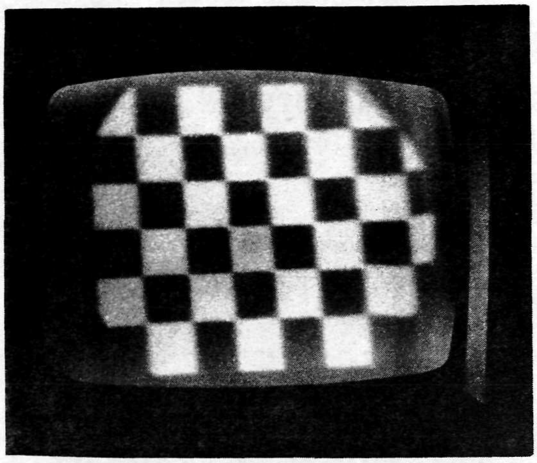


Figure 3-2. Checkerboard Image from Single Aperture

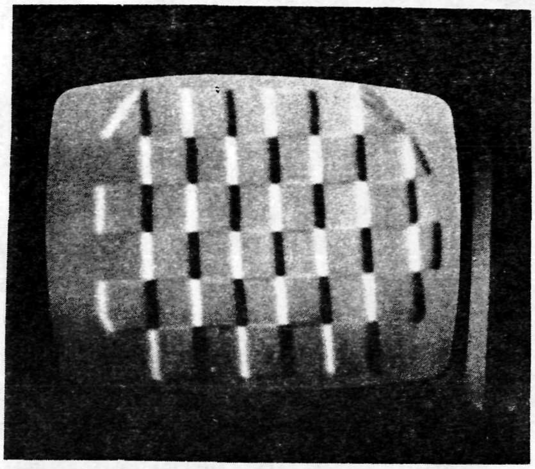


Figure 3-3. Differential Image from Two Adjacent Horizontally Stacked Apertures

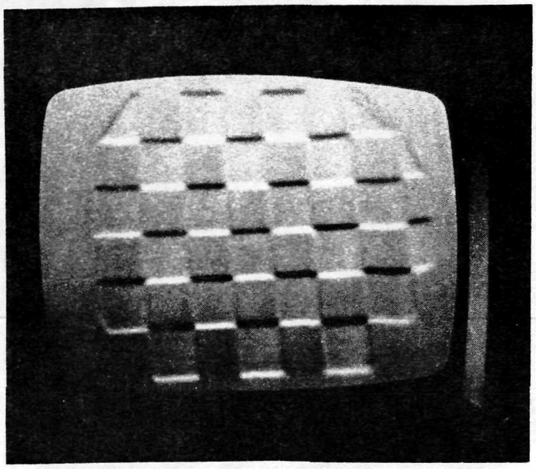


Figure 3-4. Differential Image from Two Adjacent Vertically Stacked Apertures

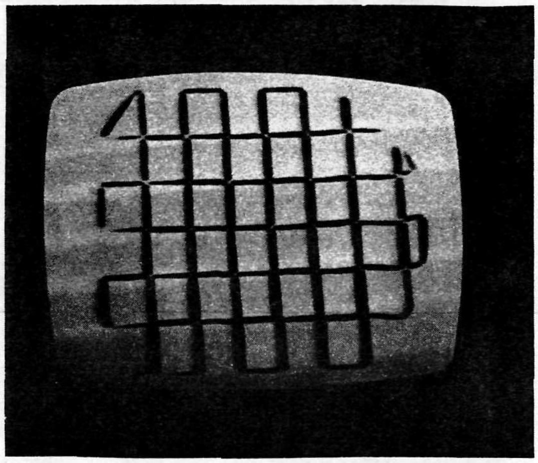
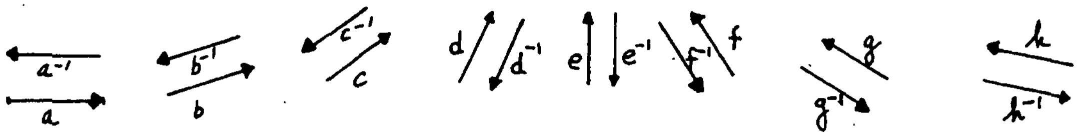


Figure 3-5. Output from Bendix Edge Ehhancement System with Checkerboard Input Image

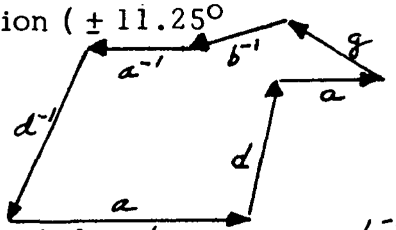
a shape-encoding scheme in which we associate activation of subsets of the multichannel aperture, matrix with graph elements out of which shapes can be constructed.

Figure 3-6 shows a 4 X 4 aperture format with basic straight line directions and basic curves. Channel activation is assumed to take place by detection of black-to-white (B——▸W) transitions or white-to-black (W——▸B) transitions. Stacking of the transitions vertically for example corresponds with detection of the 90° line direction in the image.

Since we are interested in detection of a shape under various orientations relative to the camera's viewing angle, we can assign a code to each direction and its negative.

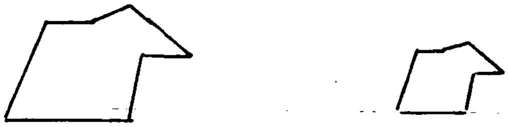


Thus given any shape made up of straight lines, we can encode it by our gross quantization (± 11.25°

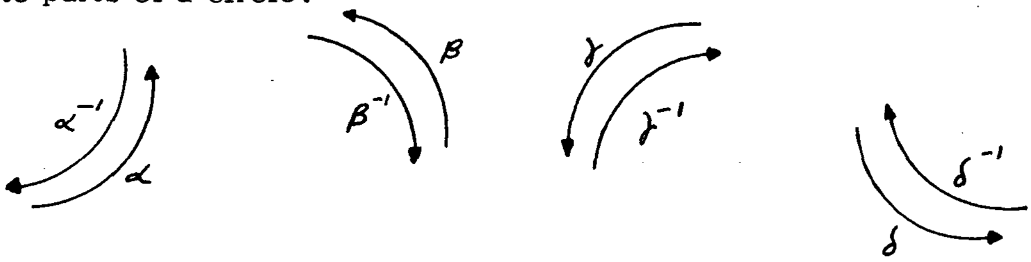


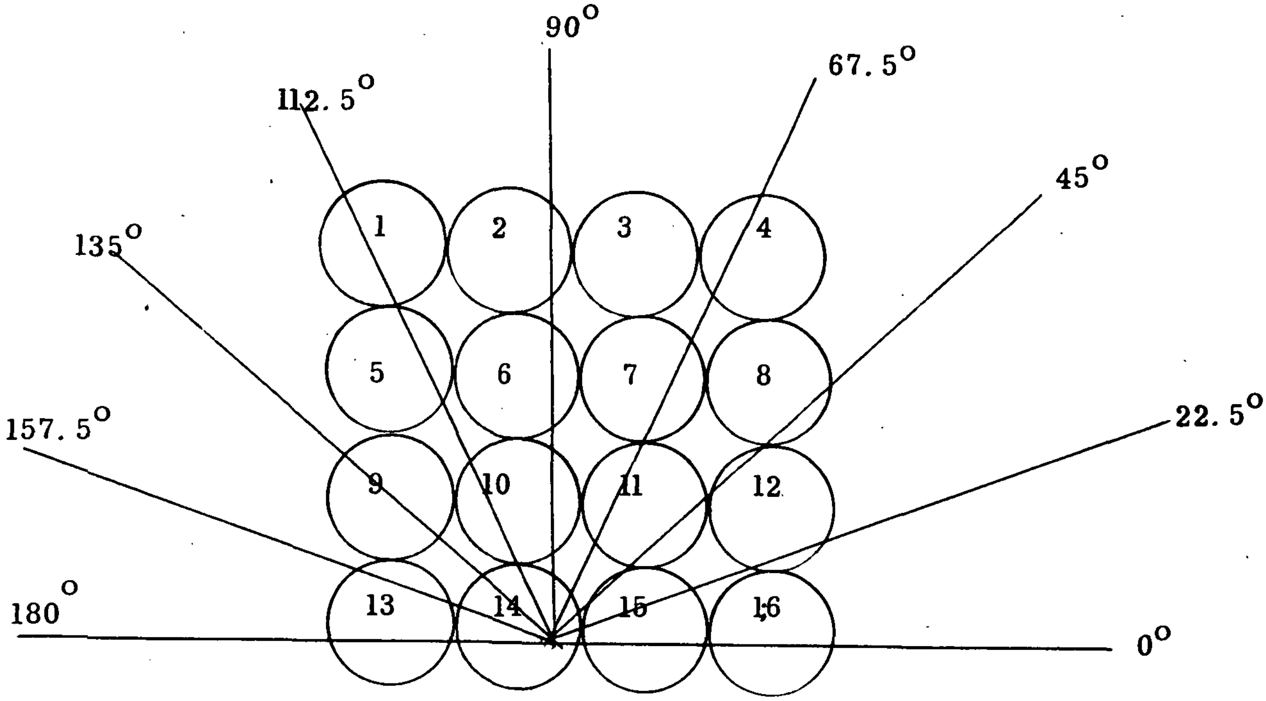
The shape is encoded as $(a\ d\ a\ g\ b^{-1}a^{-1}d^{-1})$.

We would obtain the same coding for the following shape, since we have not yet accounted for size.



In a similar way, we can encode curved shapes by assigning a code to parts of a circle.

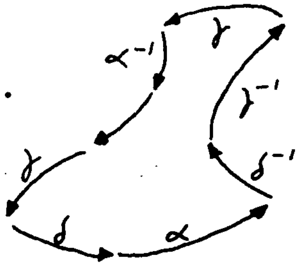




B→W, W→B Channels	Set	Basic Straight Line Directions
13, 14, 15, 16	A	180°
14, 15, 12 ---	B	22.5°
14, 11, 12, 8	C	45°
14, 11, 7, 4	D	67.5°
14, 10, 6, 2	E	90°
14, 10, 5, 1	F	112.5°
14, 10, 9 ---	G	135°
14, 13, -----	H	157.5°
B→W, W→B Channels	Set	Basic Curves
14, 15, 12, ---	I	α
8, 4, 3, -----	J	β
2, 1, 5 ------ -	K	γ
9, 13, 14 ------	L	δ

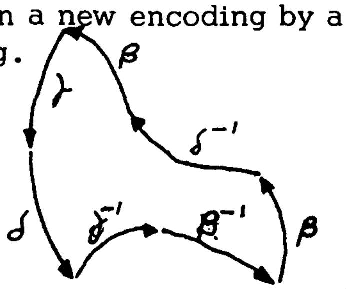
Figure 3-6. 4 X 4 Aperture Format Channels and Associated Basic Straight Line Directions and Basic Curves

We can then encode curved shapes as follows.



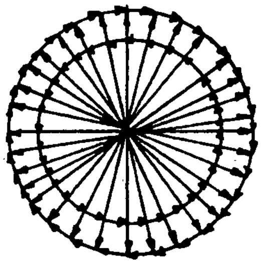
This shape is encoded as ($\delta \; \alpha \; \delta^{-1} \gamma^{-1} \gamma \; \alpha^{-1} \gamma$)

If this shape is rotated 90° clockwise, we obtain a new encoding by a 90° clockwise transformation of the old encoding.

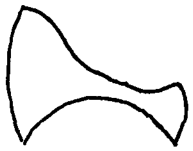


The shape in this new orientation is encoded as ($\gamma^{-1} \beta^{-1} \beta \delta^{-1} \beta \; \gamma \; \delta$).

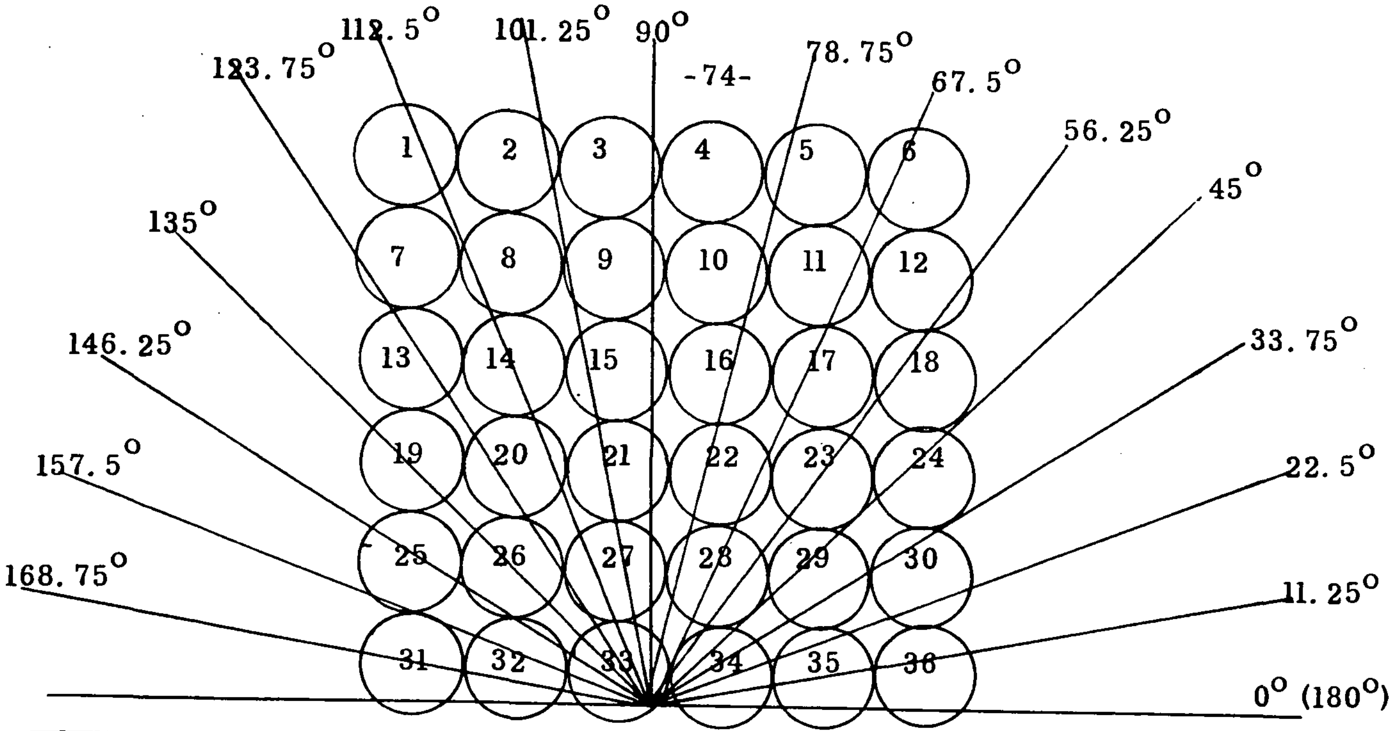
If we use a 6 X 6 aperture format, as shown in Figure 3-7, we can achieve a finer quantization of straight line directions and curves in the image. If we superimpose all the associated codings about a center, we obtain the "wagon-wheel" figure. The set of encodings forms a group under rotations by 22.5° and by reflections about any quantized direction line. As a result, we can encode a given shape as S $= \sigma_1 \sigma_2 \cdots \sigma_n$, and then drive the image dissector line-finder logic by transforming S under the group transformations, T_μ , where T_μ (S) = $T_\mu(\sigma_1)$ $T_\mu(\sigma_2)$, ..., T_μ (σ_n).



Since we have not accounted for sizing of a shape, we would obtain the same encoding representation if we were to stretch a curved shape. This is shown with our example shape.



Clearly, an approach is needed to the sizing problem to identify specific objects instead of just classes of objects, or shapes.



B → W, W → B Channels	Sets	Quantized Straight Line Directions
31, 32, 33, 34, 35, 36	A	0° (180°)
34, 35, 36	B	11.25°
34, 35, 30	C	22.5°
34, 29, 30	D	33.75°
34, 29, 24	E	45°
34, 28, 23, 18, 12	F	56.25°
34, 28, 22, 17, 11, 6	G	67.5°
28, 22, 16, 10, 4	H	78.75°
33, 27, 21, 15, 9, 3	I	90°
33, 27, 21, 15, 8, 2	J	101.25°
33, 27, 20, 14, 8, 1	K	112.5°
33, 27, 26, 20, 13	L	123.75°
33, 26, 19	M	135°
33, 26, 25	N	146.25°
33, 32, 25	O	157.5°
33, 32, 31	P	168.75°

B → W, W → B Channels	Sets	Quantized Curves
34, 35	Q	α
30, 24	R	β
18, 12	S	γ
5, 4	T	δ
3, 2	U	ε
7, 13	V	ζ
19, 25	W	η
32, 33	X	θ

Figure 3-7. 6 X 6 Aperture Format Channels and Associated Quantized Straight Line Directions and Curves

3.1.4 Approach to Sizing Problem[1]

Because our characterization in terms of curves and straight line directions is independent of metrics -- specific lengths of individual parts of a shape and the relative proportion of individual shape parts -- we now need a way to recognize shape in terms of size.

The use of the image dissector with multiple aperture can permit rapid correlation of an image stored by image points in the computer and an image in view. Since, however, the computer storage is limited and since maximizing a correlation function in matching images is time-consuming even with,say 36 correlation circuits associated with the 36 channels, we need further a way to store and process size information for rapid real-time object identification.

Let us assume that we have an effective logic for detecting black-to-white and white-to-black transitions. Such a logic would be a modification of the Binford-Horn Linefinder, for example, developed at MIT.[2] This class of line-finders should prove effective with the highly sensitive and high resolution "Dissecticon"variants of the image dissectors. The three types of intensity transitions (step, roof-shaped, flat with a peak on the edge associated with the edge-effect)[3] would be mapped into the straight line and curve codes. Since it is recognized today that a line-finder should not generate vertices in order to represent a shape, vertices would be constructed on the basis of contextual information (natural objects vs. artifacts such as instruments which may be classed as polyhedra). We thus assume that there is a strong interaction with the contextual processing of the relational program. Since the present study has a model of eye-hand coordination, contextual clues will also be developed by exploratory grasp of the object in order to derive more precise shape information and to correct the critical generation of vertices needed to encode a shape.

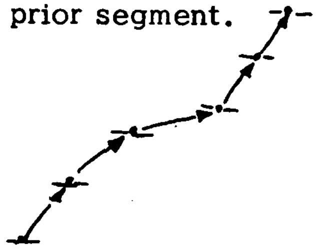
[1] Alternatively, the sizing problem can be solved in pre-processing shape date from the image dissector (as would be the case if a LINE DRAWER Progam (variant of Binford-Horn LINEFINDERis used.

[2] B. K. P. Horn, "The Binford-Horn LINEFINDER," Vision Flash 16, MIT Artificial Intelligence Laboratory (undated memo.)

[3] A. Herskovits and T. O. Binford, "On Boundary Detection," Artificial Intelligence Memo 183, Artificial Intelligence Laboratory, MIT, July, 1970.

The "wagon-wheel" encoding pass developed via the multichannel detection of straight line and curved portions of the shape to be recognized can be supplanted by a more finely quantized "wagon-wheel." This will now be done by computation, as opposed to using the shape-detection properties of the image dissector. An initial or origin x, y point is established, and the program constructs a vector to the "next" point (x_1, y_1). This process is iterated to generate the vectors, V_{i+j} = $(x_i + 1 - x_i)$ i + $(y_i + 1 - y_i)$ j. A new "wagon-wheel" encoding is thus derived from the changes in the derivative or slope of the vectors $v_i + 1$.

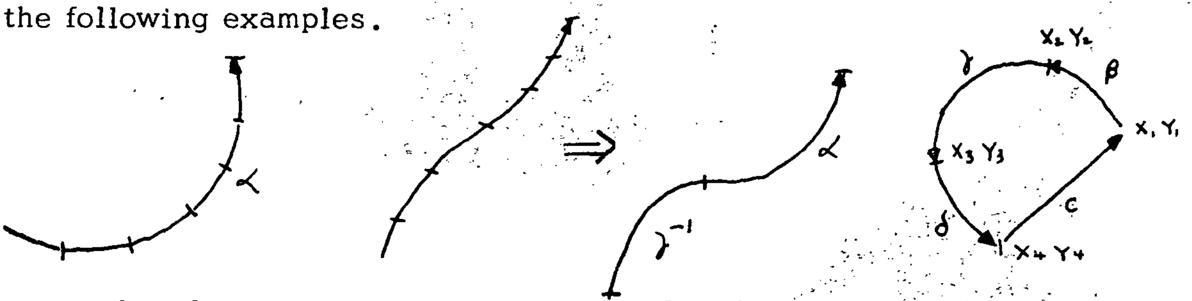
This process is shown in the following diagram. A processing rule will require that changes in the second derivative mode will provide inputs to the "wagon-wheel" construction routine which will consider that point to be a termination of a prior segment.



For example, a c-coded segment (straight line with 45° direction) would be characterized by a non-change slope of approximately $\pi/4$. An α - coded curved segment would be characterized by an increase in slope up to where the slope is zero or ∞ , as measured from the horizontal. Slope changes for the basic segments can be listed as follows.

Segment	Change in Slope
a, a^{-1}	zero
b, b^{-1}	zero
c, c^{-1}	zero
d, d^{-1}	zero
e, e^{-1}	zero
f, f^{-1}	zero
g, g^{-1}	zero
h, h^{-1}	zero
β	positive increase
β^{-1}	negative increase
γ	negative increase
γ^{-1}	positive decrease
δ	negative decrease
δ^{-1}	positive increase
α	positive increase
α^{-1}	negative decrease

Thus as a boundary or edge tracing program tracks along the B ⟶ W, W ⟶ B intensity transition points, it establishes the direction (quadrant) of the track, and the mode (slope constant, increasing, or decreasing). When a slope of zero of ∞ is reached, a change of mode is flagged and this point is characterized as a "segment endpoint." This is shown in the following examples.



Thus for image matching, for example, the program quickly traces around a boundary and stores only the set of initial and end points associated with a coded segment, thus eliminating intermediary points. For example, for a shape, $S = (\beta \gamma \delta c)$, we have a shape-point encoding, $S = ((x_1, y_1) \beta , (x_2, y_2) \gamma (x_3, y_3) \delta (x_4, y_4) c (x_1, y_1))$.

The shape-point encoding can now be used to develop sizing parameters. There are clearly several possible approaches to sizing using shape-point encoding. If the objective is to develop pointing information for the laser, an algorithm is needed to develop the scale of the shape within the field of view of the laser. If shape information is used to direct the manipulator hand, then this algorithm should provide grasping parameters. We shall assume that a reasonable configuration for a Viking mission manipulator is a "soft-hand" derived from the current Baer Automated Systems pressure-operated gripping devices. In this case only gross sizing information may be useful. For example, with several of these prehensile fingers which can be automatically "docked" with the arm mechanism, the robot selects a hand for the gripping, lifting, or manipulation task on the basis of the sizing information. Interestingly, with soft hands, shape is of minor importance because the fingers can close under differential pressure until the object is completely in contact with the fingers.

A nominal solution to the sizing proglem can however be presented. The shape-point encoding can be used to size the workpiece or natural object to be manipulated. This information can be used to "box in" in the shape segment, such that a rectilinear shape of known dimensions can be used to size the constructed shape.

For example suppose we are given a shape-point set $(x_1, y_1, \alpha, x_2, y_2)$. The program defines new coordinates (x_1, y_2) and (x_2, y_1) to construct the new shape-point set $((x_1, y_1) a (x_1, y_2) e (x_2, y_2) a^{-1} (x_1, y_2) e^{-1} (x_1, y_1))$. By iteratively constructing these rectilinear shape-point sets,

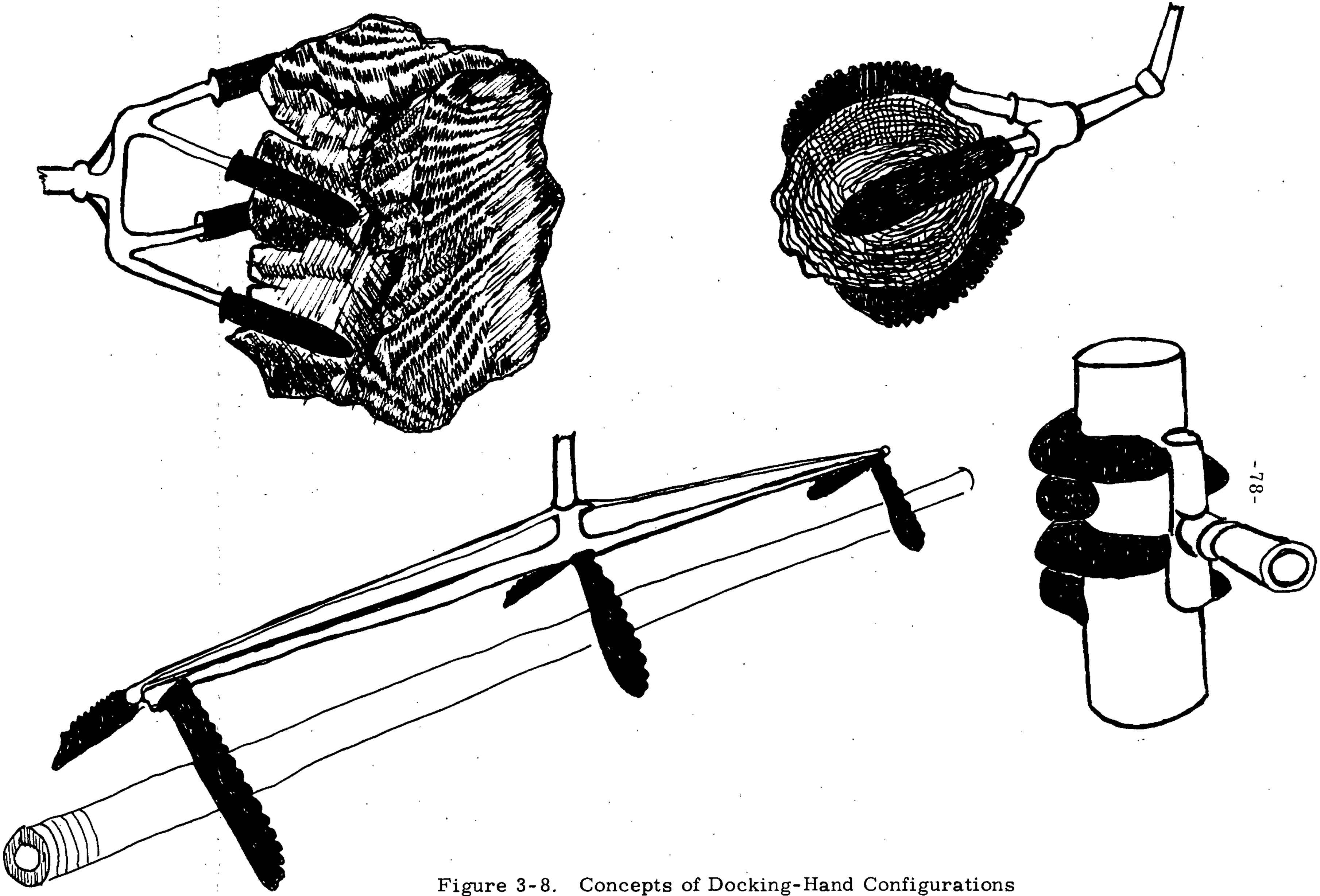
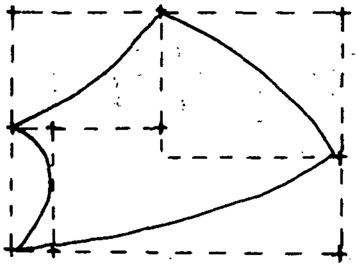


Figure 3-8. Concepts of Docking-Hand Configurations (After "Manifolding Concepts" of Baer Automated Systems, Inc., Lakeland, Florida, Oct. 31, 1971).

a final shape-point set is constructed which can be used to scale or size the shape in the image. This information can then be correlated with stored metric shape-point sets or used to compare or match images in successive looks or scans from different viewing points. Metric-shape changes can be described as a subsequence of adopting various viewing angles for a given object. In practice this might be done by rotating the object by the robot hand for different viewing angles.



This section on image-dissector and "wagon-wheel" shape-encoding should be considered only a first-stage in identification of objects by recognizing their shapes in various orientations. The next section will present a method for form homotopic encoding in order to simplify the internal machine preprocessing and representation of objects.

Figure 3-9 provides a final example of shape-encoding, this time using the 6 X 6 aperture format channels and associated straight line directions and quantized curves (as shown in Figure 3-7). The example shows two cross-sectional views of the shape of a Baer inflatable finger. The upper encoding and diagram of the finger line drawing is in the unflexed state and the lower encoding and diagram represents the flexed state.

3.1.5 Grass-Fire Encoding

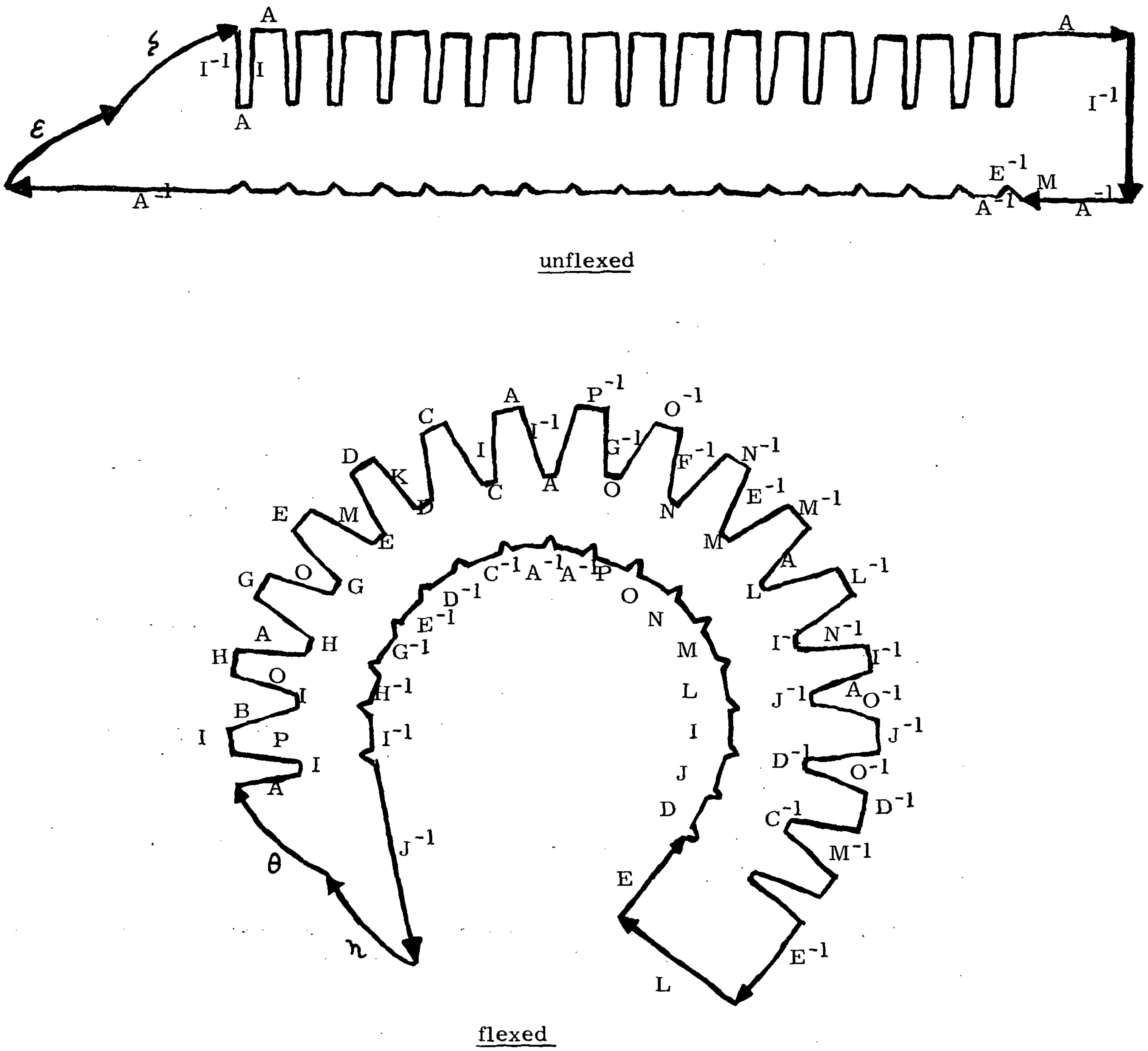
Grass-fire encoding is a technique derived from earlier studies of pattern-recognition in which a pattern or a shape is uniquely characterized in terms of structural transformation of that shape in Euclidean space.[1,2,3,4]

[1] H.Blum, "An Associative Machine for Dealing with the Visual Field and Some of its Biological Implications,: AFCRL-62-63, February 1962, Air Force Cambridge Research Laboratories, Hanscom Field, Mass.

[2] J.C. Kotelly, "A Mathematical Model of Blum's Theory of Pattern Recognition," AFCRL-63-164, April, 1963.

[3] H.Blum, "A Transformation for Extracting New Descriptors of Shape," Symposium on Models and Perception of Speech andVisual Forms, Boston, Mass. Nov. 1964.

[4] L. Calabi and W. E. Hartnett, "Shape Recognition, Prairie Fires, Convex Deficiencies, and Skeletons," 5711-SR-1, Parke Mathematical Laboratories, Inc., Carlisle, Massachusetts, Feb. 1966.



unflexed

flexed

Figure 3-9. Illustration of Shape Encoding of Inflatable Finger in Flexed and Unflexed States (6 X 6 aperture format)

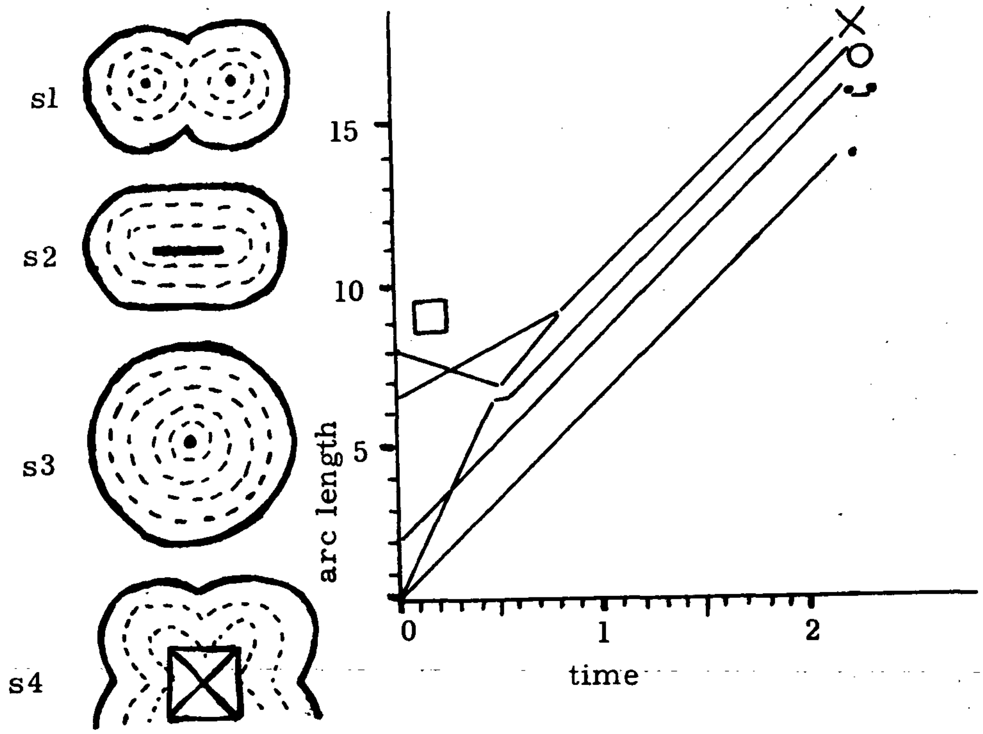


Figure 3-10. Arc-length of Wavefronts vs. Time for Different Shapes (from Blum (1962)).

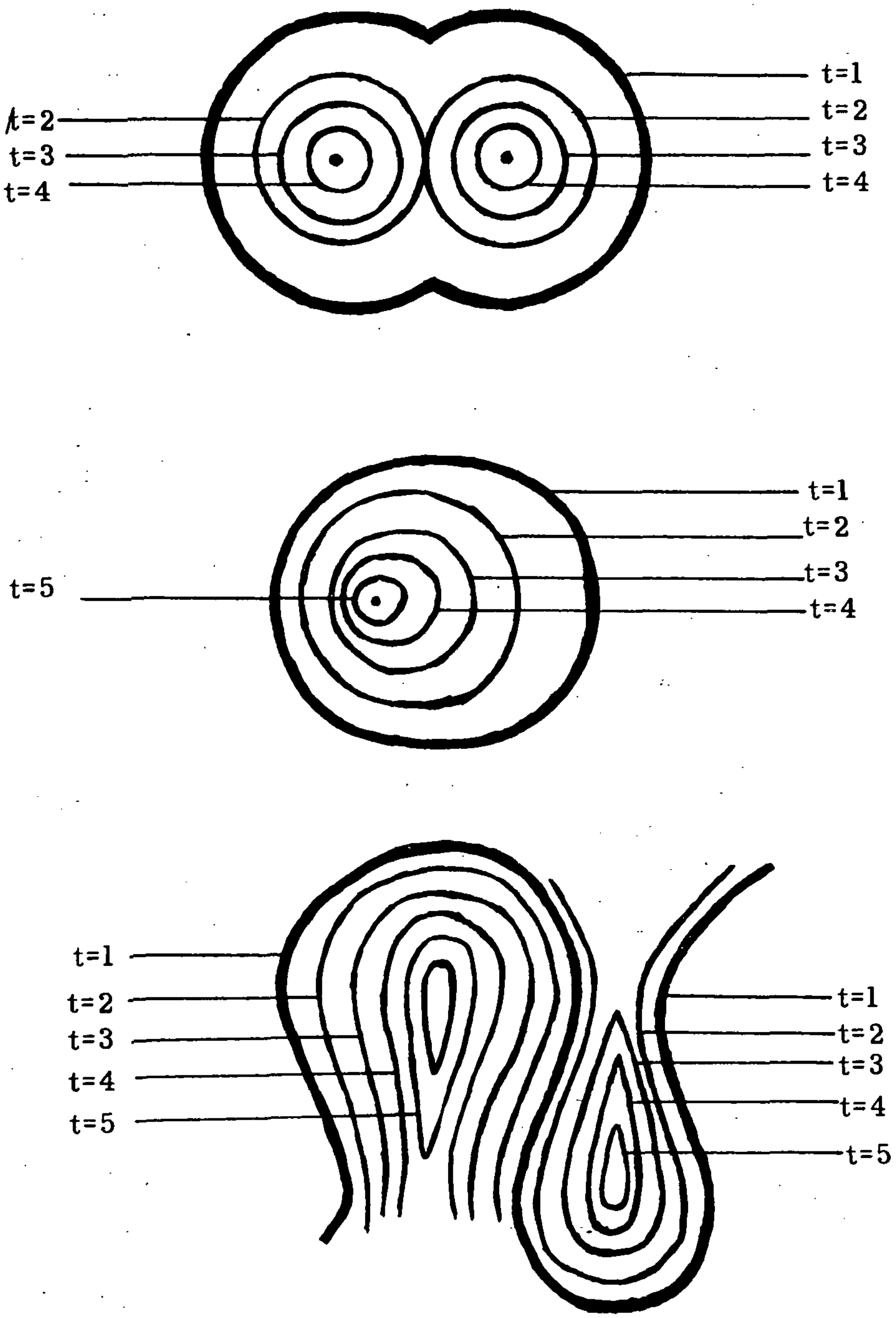


Figure 3-11. Characteristic Wavefront Patterns for Different Shapes

In the present model, we imagine that the image-plane of the image-dissector camera is an ideally homogeneous field that is dry and ready to burn. The variant of the Binford-Horn line-finder mentioned in 3.1.4 defines a boundary or a stone wall within this plane. If we spray kerosene inside this wall and then arrange to set it afire all at once, the wall acts as a fire-cut and the fire develops toward the center of the shape. If we observe how the fire progresses, we will see homotopically-generated wavefronts. In this model, we do not have any portions of the wavefronts passing through each other, since we cannot burn a space or a region twice.

These Huyghens-like wavefronts have an interesting property first observed by Blum.[1] If we map the arc lengths of the wavefront as a function of burning-time, we obtain a graphical method for characterizing shapes. Usually, this method is analyzed in terms of mapping of the unit interval onto the wavefront plane. For example in Figure 3.10 we reproduce Blum's diagram in which an X-shape, a circle, a line defined by two terminal points, and a single point are the origins of the fire and the wavefronts burn outward to define a characteristic shape as a function of time. Clearly, we can also assume that we start with the outer shape and burn inwards to the center, i.e. the generation of wavefronts is reversible as a process.

Some examples of wavefront-generation are shown in Figure 3-11 for various shapes.

We can derive an encoding method from this process which should prove useful in minimizing storage of a large number of shapes, in correlating an observed shape with a stored shape, and perhaps more significantly, in minimizing the bandwidth required for transmittal of shape information on the downlink to earth.

Shapes can be encoded in terms of (1) arc-length of the wave front vs. time (Fig. 3-10) (2) times at which wavefronts meet (3) time for the disappearance line (at the end of the burning) (4) the characteristic "stick-figure" defined by the disappearance lines of the shape.

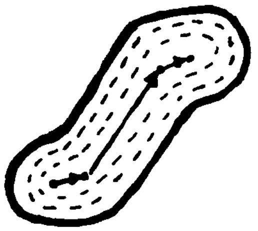
If we utilize the shape-point metric information discussed in section 3.1.4, the characteristic stick-figures can be encoded in the same way that the original boundary shape was encoded using the "wagon-wheel"

[1] H.Blum, "An Associative Machine for Dealing with the Visual Field and Some of its BiologicalImplications,: AFCRL-62-63, February 1962, Air Force Cambridge Research Laboratories, Hanscom Field, Mass.

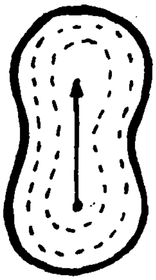
encoding technique. In this way, the shape can be reconstituted or reconstructed on earth on the basis of the burning-time and wave-meeting time parameters and the straight-line segment codes. For example, the stick-figure of shape 1 (an assymetric shape) would receive an encoding different than the symmetric shape 2.

Wavefront transformations at times t can also be encoded by "wagon-wheel" codes since they can be regarded as quantized or discrete samplings of a continuous homotopic deformation of the original shape.

shape 1



shape 2



$S_1 = ((x_1\ y_1)\ C\ (x_2\ y_2)\ G\ (x_3\ y_3)\ C\ (x_4\ y_4))$. $S_2 = (x_1\ y_1)\ I\ (x_2\ y_2))$.

3.2 Object Identification by Recognition of Contours

3.2.1 Approach

Research at Stanford University's Vision Laboratory points to the feasibility of a new approach to computer vision using a laser coupled to a vidicon TV system. Parallel developments are underway at Stanford Research Institute to adapt a surveyor-type laser for robot vision. The use of lasers as image processing inputs to robots promises solutions to the problems of 3-dimensional machine representations of solid objects. Lasers can also lead to simplification in software systems -- a prerequisite for realistic system designs for the Viking missions.

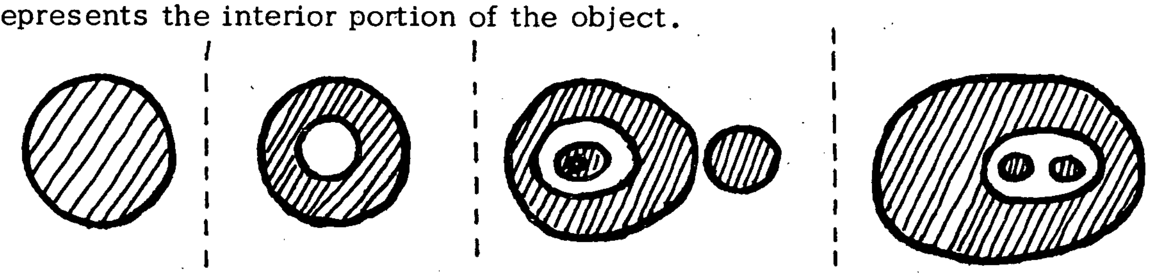
This section outlines an approach to encoding a laser-derived image of a 3-dimensional, smooth, compact body. Through eye-hand coordination (via an ancillary TV eye), objects whose size and mass permit manipulation can be presented to the laser in such a way as to obtain representative viewing angles. For other objects, a rover-type robot can obtain different viewing angles by moving about the object.

The laser systems will be scanned to derive 2-dimensional contours of the object corresponding to object cross-sections in a plane perpendicular to the laser's optical axis or the line-of-sight. Alternatively, the laser input can be thought of as "contour-map" information. The contours will appear at regularly sampled intervals of length along the line-of-sight. These intervals will be variable as required for finer quantization of the "contour-map."

3.2.2 Image-Encoding From a Single Viewing Angle

Contours derived by the laser imaging system will represent compact, 2-dimensional objects in the plane bounded by curves. Contours can be classified as non-singular and singular. A non-singular contour is defined by the condition that the boundary curves consist of several simple, closed curves with no intersections or self-intersections.

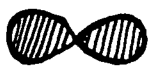
An example of non-singular contours is shown below in which shading represents the interior portion of the object.



If we assume that the object is smooth, then by a well-known theorem of Sard[1] we know that the contours determined by the various distances along the line-of-sight will be non-singular (except for a set of distances which form a subset of the real line of measure zero). Thus the probability of any randomly sampled contour being singular is zero.

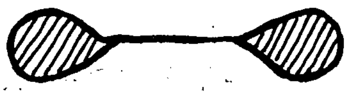
In addition to providing boundary curves, the contours also give "shading" or depth, i.e. contours indicate which regions of the plane are interior and which exterior to the object. The assumption that the object is bounded uniquely specifies interior and exterior regions since given the boundary curves, one simply alternates shaded and unshaded regions as one moves inwards.

A singular contour is simply any other contour than a non-singular contour. The following are examples of some singular contours.



A

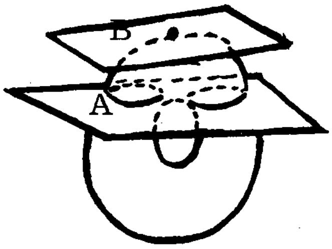
B



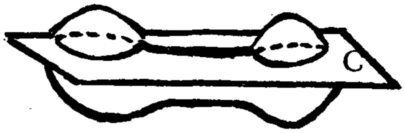
C

[1] cf. V.Milner, Topology from the Differentiable Viewpoint.

These singular contours occur from the labeled sections of the following objects.



torus



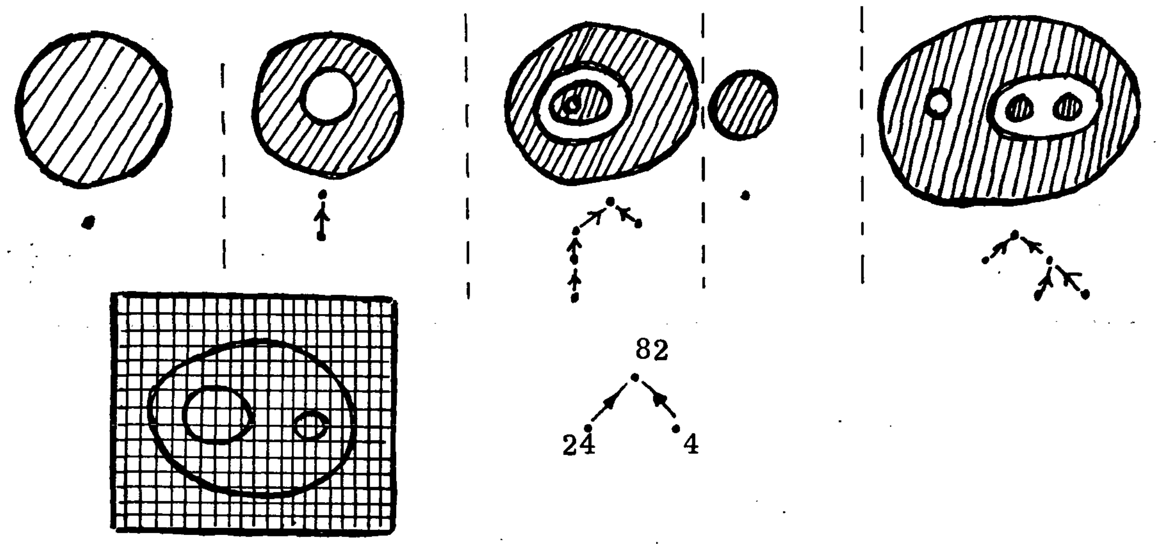
"dogbone"

In fact, whenever two non-singular contours of an object are different topologically, there must be a singular contour at some intermediate distance along the line-of sight. Usually, these singular contours are isolated and they are always of measure zero.

A simplest machine strategy would be to discard singular contour information since it's occurrence would be rare. If by chance, there should occur two successive singular contours, then the program would direct the search for an intermediate non-singular contour.

The simplest approach to encoding non-singular contours is to store only two parameters: (a) the number of contour-components (i.e. simple closed curves); (b) the depth or range-ordering of components (i.e. which components are inside or outside relative to other components). This information can be symbolically represented by a graph with oriented line segments. The vertices correspond to the components and the edge connects two vertices if one of the corresponding components is inside the other. The edge is directed from the inside vertex to the outside vertex.

An example of this encoding schema can be given for the examples of non-singular contours previously given.

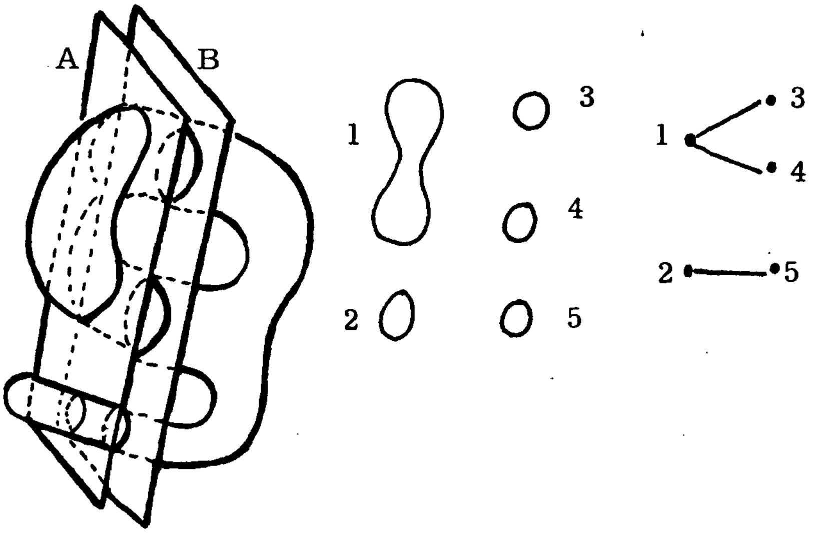


These basic graphs can be used to build up a metric representation of the contour instance -- information such as contour quantization (distance between the components) and the sizes of the boundary-curves. For example, in the preceding graph representation, we assign to each vertex a weight measuring the area enclosed by that curve. The graph indicates that the represented object-contour has two holes and that the contour area is 54 and the hole areas are 24 and 4.

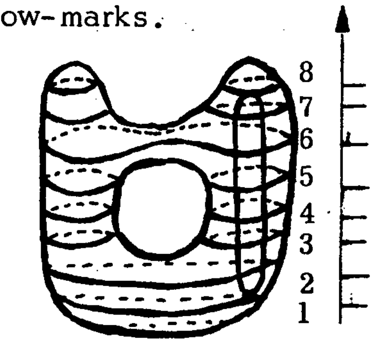
3.2.3 Assembly of a Sequence of Contours

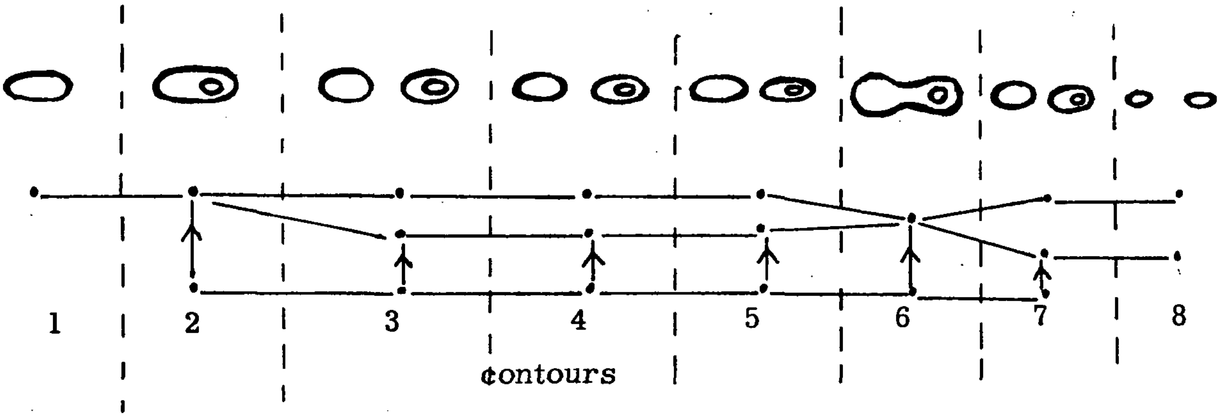
We can now encode a sequence of non-singular contours obtained from planar cross-sections perpendicular to the line-of-sight and sampled at regular range intervals along the line-of-sight. Range-gating can be variable, however, when as mentioned earlier, singular contours happen to be acquired. We now want to represent the contiguity between adjacent contours. We need to indicate the connectivity of curves associated with closer range contours with curves associated with contours at a greater range via the slice of the object lying between the corresponding planar sections. More precisely, consider that portion of the object lying between the two planar cross-sections, and imagine it breaking up into several components each of which connects certain points of one contour with certain points of the contiguous contour. Or alternatively, we could just consider the shape or boundary of the object from our viewing angle. If we consider that part of the boundary lying between the contour-sections, it would consist of several pieces of smooth surface with boundary curves lying in one or both of the planar sections. The problem now can be stated: which curves bound the same piece of intermediary surface.

This can be visualized in the following example. There are two pieces of surface between planes A and B. One piece connects contour 1 with contours 3 and 4. The other piece connects contours 2 and 5. We represent this situation graphically by connecting the corresponding vertices of the two graphs associated with the cross-sections via an undirected edge.

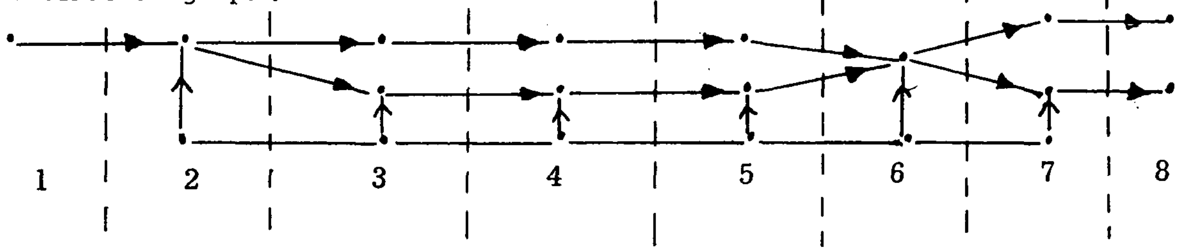


In another example, we can indicate contour information and contour-connectivity information. The example shows a contour-map defined by eight planar sections perpendicular to the laser line-of-sight. The sections are indexed by the arrow-marks.



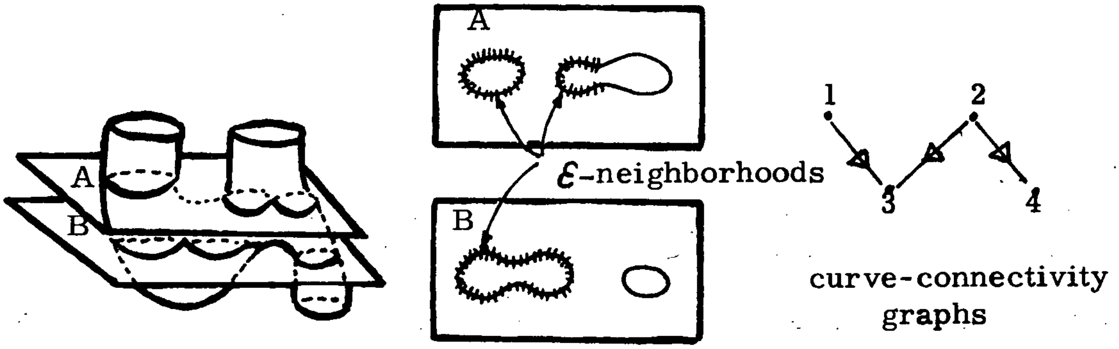


This graph is constructed of both directed and undirected edges that correspond with individual cross-sections of the object. This is done by stacking vertices vertically when they belong to the same cross-section. In a <u>connected</u> graph (one in which any two vertices are joined by some sequence of edges or line segments), the presently undirected segments can be directed from left to right (corresponding to the sequential stacking of the planar sections perpendicular to the laser line-of-sight). If a graph happens to be disconnected, then it is a representation of multiple objects or equivalently a disconnected object (consisting of more than one piece). By assigning solid arrows to indicate contour-connectivity and open arrows to indicate contour structure, we can now represent the above example as a directed graph.



This graph represents the connectivity of contours as well as the structure of individual contours. The problems still remains to represent the connectivity of contour-components, i.e. curves, between contiguous contours in the sequence of cross-sections.

One approach is suggested by the condition that the distance between contiguous sections is less than the distance between different curves in the same section. If this is the case, if we were to superimpose sections, then curves of different sections would be connected. More precisely, we would select a small real number $\varepsilon>0$ and consider the ε -neighborhood of each curve in each contour. This neighborhood is defined as the area in the plane consisting of all points of distance $< \varepsilon$ from the curve. Then whenever the curve in one contour lies within the ε -neighborhood of a curve in an adjacent contour, they are to be connected by an edge. In the following example, curve 1 is contained in the ε -neighborhood of curve 3, and curve 4 is contained in the neighborhood of curve 2. This gives us a connectivity-graph for contour-curves between different adjacent contours.

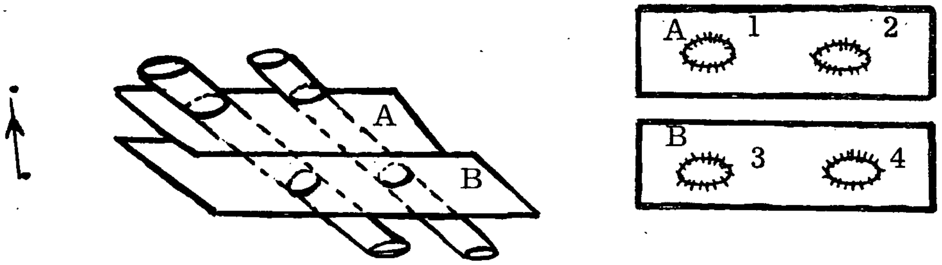


curve-connectivity graphs

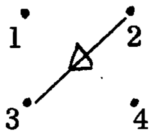
contours

If ε is a value too small, some curves may fail to connect with a curve from the adjacent contour. If ε is chosen too large, then some curves will be contained in the ε -neighborhood of another curve in the same contour. The selection of the value for ε clearly has to be determined experimentally for the class of objects that we expect to be identified by the robot. Since case 2 (selection of too large a value for ε) can be nullified by the condition or rule that the connectivity-arrow for curves cannot connect curves within the same contour, then the threshold-value for ε can be increased to avoid case 1 (failure to connect curves at all).

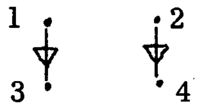
The next example illustrates another difficulty likely to be encountered by a program processing contour-sequences.



If we superimpose contour section A onto contour-section B, then the curve-connectivity graph by our method of $\mathcal{E}$ -neighborhoods of the curves, would be the following.



However, a true representation would have the following graph.



This error is induced by the fact that the contour-sections are not sufficiently close together. This implies that an optimal contour-section sampling would approach continuous sampling. Clearly, continuous sampling would avoid the above misrepresentation of curve-connectivity between sections. However, approaching continuous sampling means that inordinate demands will be placed on the resolution of laser-scanning. It also means that inordinate storage and processing demands will be made on the contour vision program to process the correspondingly large number of graphs that would be generated by laser-scanning that approaches a continuous scan. One approach to solving this problem is to maximize the scanning resolution of the laser, but to discard redundant contour-sections as they are inputted from the laser and associated logic.

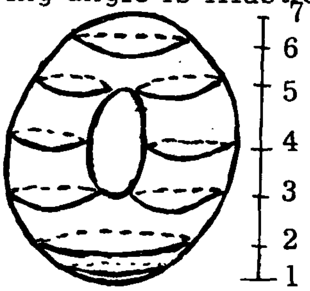
Redundant contour-sections can be detected by checking whether contours are isomorphic. A contour is siomorphic to another contour when there is a one-to-one correspondence between vertices such that a directed edge or segment joins the corresponding vertices in the other contour, and in the same direction. (In our graphs, at most one edge connects any two vertices.)

An example of isomorphic contours can be found on page . Contours 3, 4 and 5 are isomorphic. This set is also isomorphic to contour 7 and further, contours 2 and 6 are isomorphic, but the program would not label any contours isomorphic unless they belong to adjacent sections.

Our program procedure to discard isomorphic contours is as follows. Once the laser has scanned a window of k-contour sections, this contour-window or contour-sequence is temporarily stored. The k+1 contour is then read into storage and tested for isomorphism with the kth contour. If the kth and k+1 contour are verified as isomorphic, then the kth contour will be discarded. But first, the k+1 contour is checked for isomorphism with the k-1 contour. If k-1 is isomorhpic to k+1, then the program discards k-1 and retains contour k, otherwise it discards contour k. This enables the program to store the most recent contour to determine connections to the present contour.

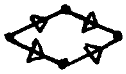
As a consequence of Morse Theory[1]the number of contours the program will store will have an upper bound beyond which however closely the sections are taken (however finely the laser is scanned), the contour-sequence will remain the same. More precisely, if we change the viewing angle by an infinitesmal amount, the following situation will occur (1) A finite number of points on the line-of-view will determine singular contours. (2) The non-singular contours arising from points in the interval between two adjacent singular points will be isomorphic. We call this a Morse View (although in fact a Morse View is even more restrictive.) We can conclude from this that there exists a well-defined sequence of graphs that is independent of the particular sections chosen, as long as they are sufficiently close, and dependent only on the object and the line of view. A further consequence of Morse's Theory is that nearby lines of view will yield the same sequence of graphs. The objective of the contour-sequence processing program then is to construct the well-defined sequence of graphs representative of the contour-connectivity in the object. From Morse Theory, we know that this can be done by discrete scanning and that there exists a practical upper bound on laser scanning resolution beyond which we will not obtain a different graph-sequence. However, there is a lower bound also, as we have seen, below which we will not obtain contour-sections that are close enough for computation of the connectivity of contours.

The unique correspondence of a graph sequence representative of the object from a particular viewing angle is illustrated in the following example.

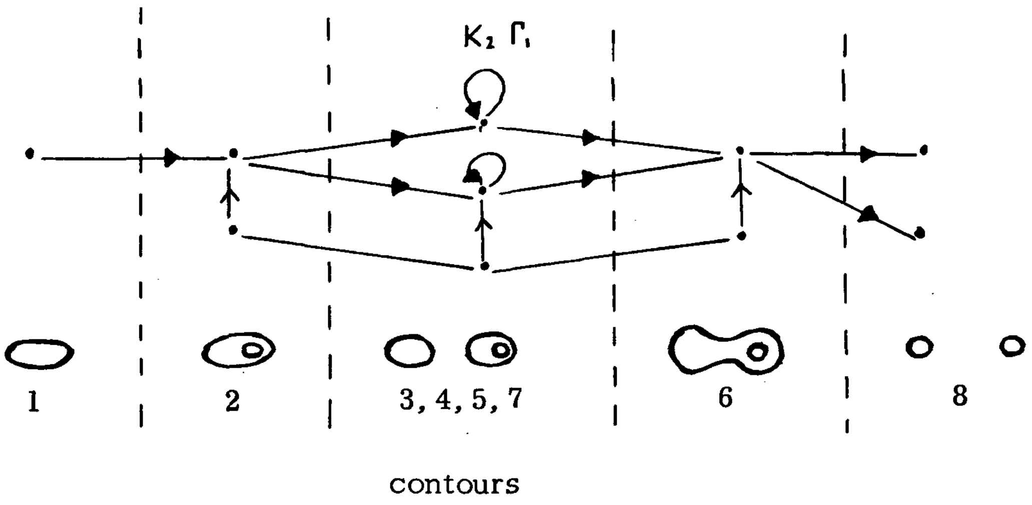


[1] J. Milner, Morse Theory, Annals of Mathematical Studies No. 51, Princeton University Press, 1963.

The singular contours appear at points 1, 3, 5, and 7. Any contour between contours 1 and 3 looks like contour 2 (i.e. its graph is isomorphic to the graph of contour 2). Similarly, the set of contours between contours 3 and 5 look like contour 4, and the set between 5 and 7 look like contour 6. Thus the graph of this object from this viewing angle will always have the following representation.



More compact coding of graph sequences can be realized by indexing the number of times a given graph appears in succession. For example, on page 27 the program would record only one graph for contours 3, 4, and 5, but would assign it a series number 3 to indicate that the graph represents a sequence formed by an immediate succession of contour 3, 3 times. Further, we can indicate succession by a relational symbol, Γ and immediate succession by a symbol K . Thus the graph that is the beginning of a series would be indexed to denote the multiplicity of its immediate successions and successions. In this same example, contour 3 is succeeded by contour 7 (another instance of contour 3). The sequence of graphs on page can now be represented by a directed graph in which the directed edges between contour edges between contour sections are replaced by scanning -relations, K and Γ , whenever a contour is repeated in a graph sequence.



3.2.4 Image-Encoding from Multiple Viewing Angles

Up to this point, we have given an approach to image-encoding from a single viewing angle. Since different objects may present identical contour-sequences from given, single viewing angles, we clearly need an identification procedure that is based on multiple viewing angles of the same object.

We shall assume in this section that the object is in a fixed position (i.e. has not moved relative to its position during laser scanning from the first viewing angle).

Our present problem then is to select multiple viewing angles that will satisfy two requirements: (1) the ensemble of views will distinguish different objects; (2) different ensembles of views of the same object can be recognized as views of the same object. Requirement (1) relates to identification of an object so as to distinguish it from different objects that may be indistinguishable from a single viewing angle. Requirement (2) relates to recognition of an object on the basis of a different selection of viewing angles and to avoid erroneous identification of the object as a new object different from the object previously represented by a set of different viewing angles.

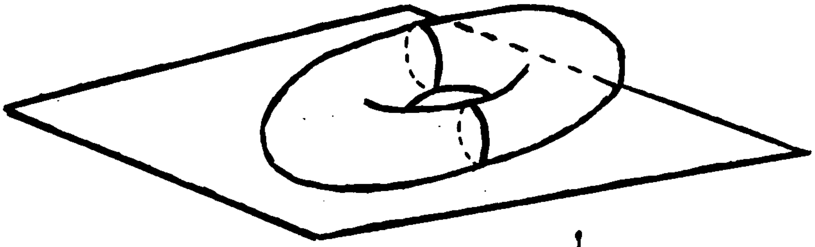
We assume that the object is resting on a horizontal plane in a certain position or orientation. We shall describe a procedure for obtaining a finite selection of views of the object in this given position. The resulting set of views will depend upon the object in a particular position and should distinguish the object from a different object or the same object in a different position. We shall then consider the problem of recognizing the object in different positions.

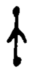
The first view of the object will be taken vertically (from directly overhead). It may be noted in passing that this is the normal viewing angle of the laser/TV system being developed by Stanford University. The cross-sections of the object will be planes parallel to the horizontal plane on which the objects lies. The remaining views will be along horizontal lines and can be obtained as follows. The laser will be moved in a circular path about the object in a horizontal plane. From each point of view it views the object along the horizontal line which passes through the origin of the circle. In this way, a continuum of horizontal views is obtained (actually a large number of views from regular small intervals along the circle.)

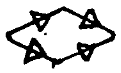
It follows from properties of the function space that we may assume that all but a finite number of views along the circle will be Morse views. Thus the Morse views occur at the points of a finite number of open intervals along the circle. The views from the points of one of these intervals will all give the same contour representation. It follows that once our sampling along the circle is fine enough, we will obtain a certain finite number of different views which remain fixed independently of increasingly finer samplings. The laser has only to travel 180° and the final view will be the reverse of the initial view.

This procedure provides us with a well-defined sequence of views that are independent of the technique used to obtain the samplings.

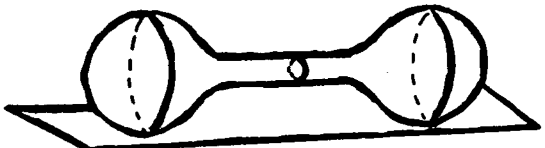
Consider the following example. Suppose we have a torus lying on a horizontal plane.

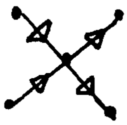


The top view will give the graph 

The horizontal views will all be the same 

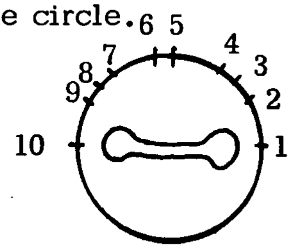
Another example is a "dogbone" lying along its length.



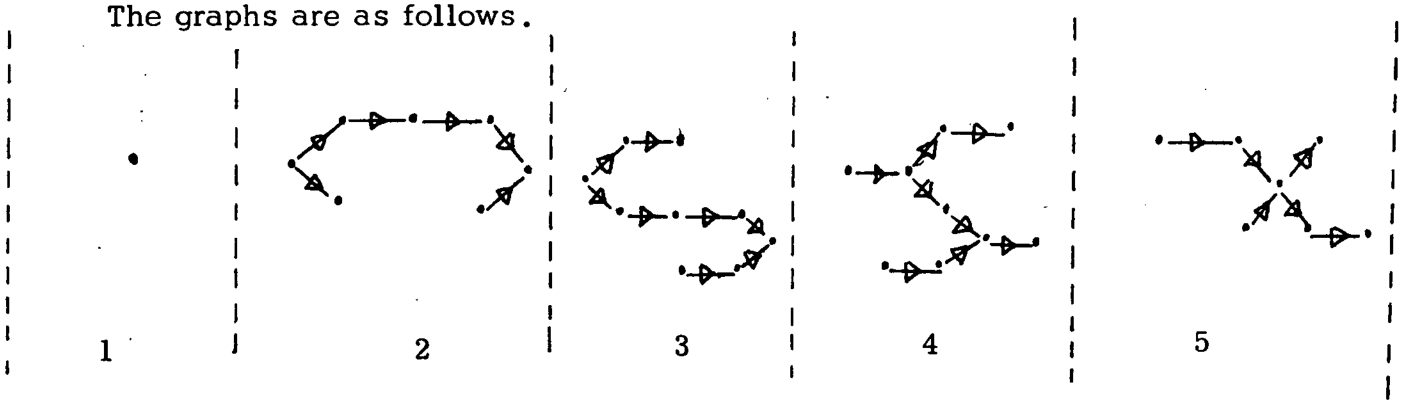


The top view is represented as follows.

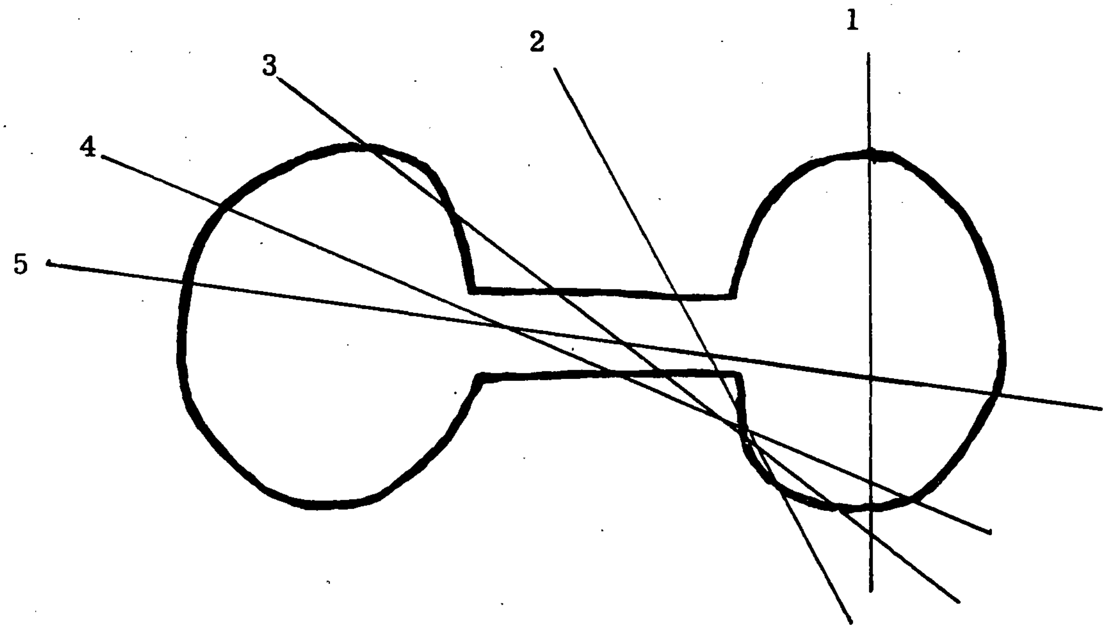
Moving along the horizontal circle give us 10 different views which occur at positions along the circle.



Some of these views are identical to others. They group together as follows: 1, 10; 2, 9; 3, 8; 4, 7; 5, 6. The last group contains adjacent views which are identical, even though they are separated by a non-Morse view. Nevertheless, our final representation would not contain view 6. The graphs are as follows.



Sample cross-sections for each of these five views (as seen from the top or overhead) would appear as follows.

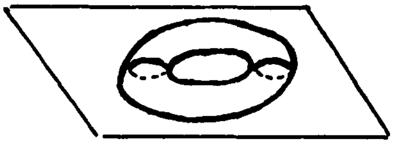


Our results up to now can lead to a program procedure for identifying an object on the basis of contour information via unique encodings provided that the object happens to be in a fixed position. A method is also needed to recognize an object whoseposition or orientation has changed. Within the context of the projected robot application in the Mars mission, this capability is especially needed to recognize artifacts such as tools or instruments that can be expected to assume positions different from those encountered in laboratory training sessions or in first sightings on the surface of Mars. Furthermore, the robot eye should be able to recognize natural objects whose positions have changed either as a result or prior manipulation or transfer or as a result of intervening environmental changes such as the effects of dust-storms or landslides.

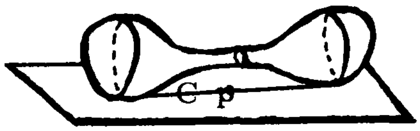
Clearly, there are several possible approaches to solving the problem of seeing an object in various positions such that the robot recognizes it as the same object and not several different objects. The program could look up the shape-encodings developed earlier to correlate positions with boundaries. Let us assume initially that we restrict the computer vision to contours alone. One approach would be to encode the contours of an object in all possible positions. Using the present topological approach, there would be only a finite number of different encodings required since slight perturbations in position do not induce a corresponding change in a given contour encoding. In fact, machine implementation of our method would indeed require a slewing mechanism to make slight changes in the viewing angles of the laser optics to perturb the views. This will probably be required in order to obtain the requisite Morse Views which form the basis for finite sampling of contours and finite assembly of contour sequences. We can expect that selection of perturbations will be critical since different perturbations will probably yield different representations of contour sequences thus leading to erroneous object identifications. However, pointing algorithms can be readily devised to produce systematic perturbations to obtain valid Morse Views.

As is usual in artificial intelligence, an optimal solution to problems of ambiguity in computer vision should invoke context to remove ambiguities. Instrumentation with its relational structures (reminiscent of the structural relations of Winston at Project MAC) will have contextual positions. More cogently, objects will have *natural* positions. For example, we may restrict ourselves to positions in which the object can stand unsupported and stable (such that a slight push will not upset it). To state this requirement mathematically, a stable position will be one in which the center of mass of the object projects to a point p in the horizontal plane P which lies in the interior of the convex hull, C, of the subset of P formed by the points of contact with the object. These are called *stable.* A *semi-stable* position will be one defined by p belonging to the *frontier* of C, i.e. p belongs to C but not to its interior.

The following is an example of a stable position. The points of contact form a circle. C is the region enclosed by the circle (and boundary) and p is the center of the circle.



The next example shows a semi-stable position. The points of contact are two points. C is the line joining these two points and p is the midpoint of the line, C.



The object may have no stable position (as in the case of the "dog-bone"), but there will always exist a semi-stable position. Contour-encodings of stable or semi-stable positions would eliminate the necessity for storing contour-encodings of objects in all possible positions. However, this problem requires further study.

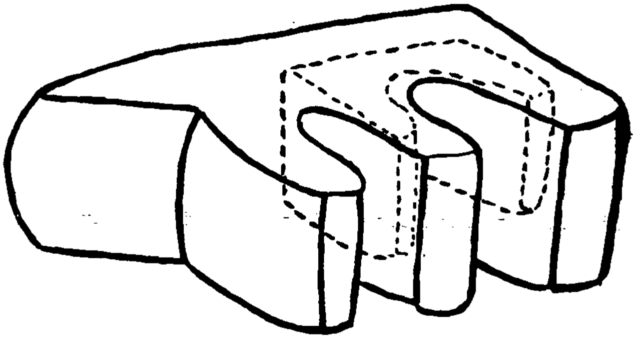
Other solutions may offer better alternative approaches to recognizing an object that may have moved relative to the "training set" position or the first sighting. One such solution would be to utilize color information (at least for artifacts) such that a change in position would present a corresponding color transformation to the TV system. This approach has certain drawbacks since the resolution of the color TV system may not be sufficient for the overall system resolution.

Alternatively, if we assume that the imaging sources (both active and passive) can move about the object, a search procedure can be initiated for a reference contour-encoding sequence. The system would try to correlate a stored encoding with encodings taken by moving about the object. A rule could be used that no object will be interpreted as a new object unless a zero correlation exists with a set of stored contour encodings (graph-statements.) If a correlation is met, then the program can infer that the object has a certain identity with a new position.

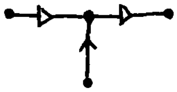
3.2.5 Example of Contour-Encoding of a Robot Finger

We shall conclude this section on identifying objects on the basis of recognizing contours by giving an example of our procedure for the inflatable robot finger discussed in the section on shape-encoding on page .

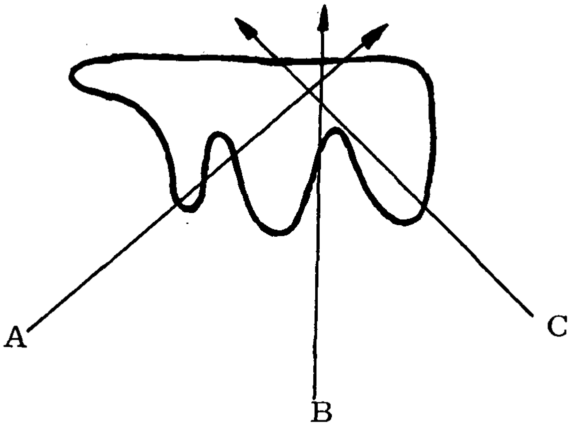
We encode a portion or segment of the robot finger in a supine or unflexed position. The following shows the segment under consideration.

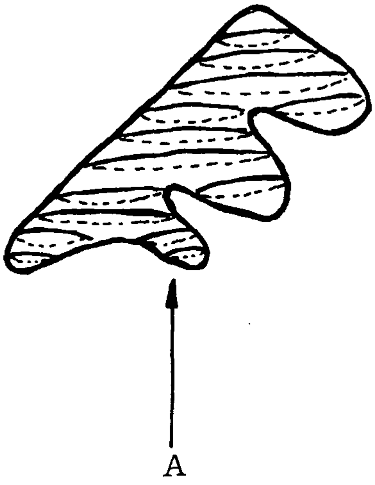


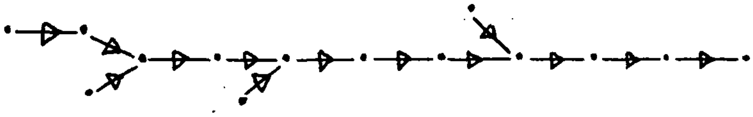
The dotted outline shows the hollow interior of the finger. The view of the finger from on top has the following simple graph.



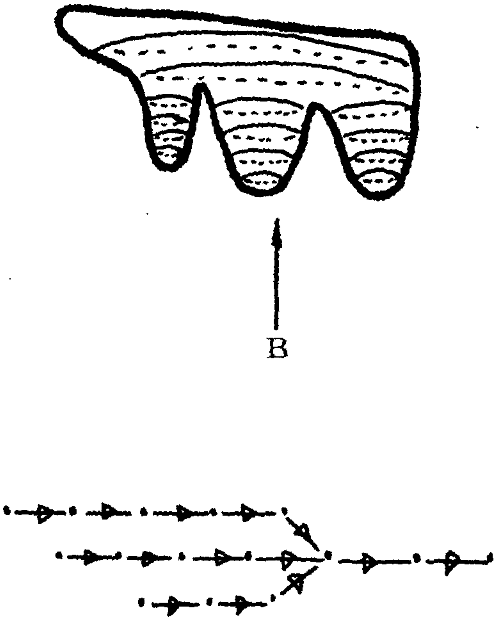
There are many different horizontal views. We can indicate three of these views along the direction indicated (as shown in a view of the finger from on top.)



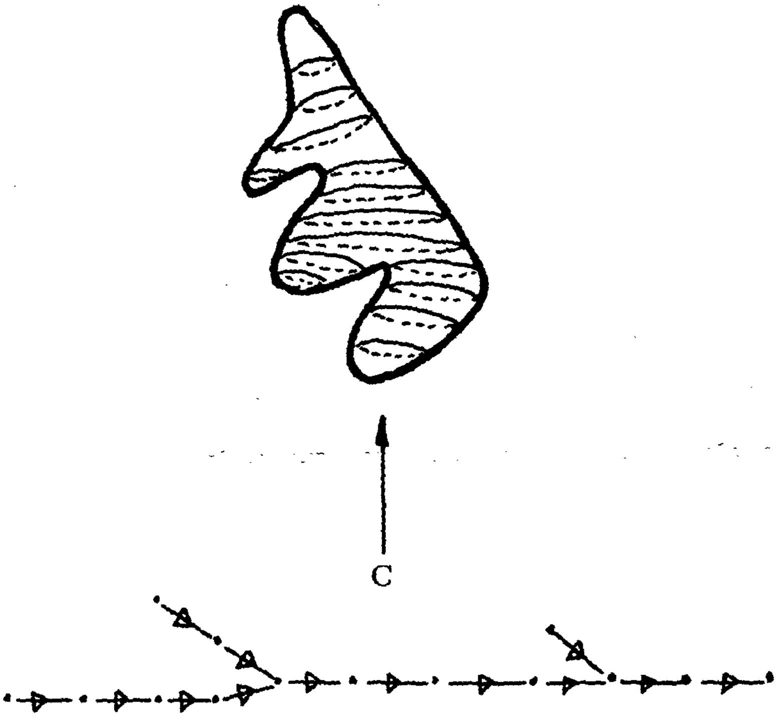




Contour-graph from direction A



Contour-graph from direction B



Contour-graph from Direction C

SECTION IV. SYSTEM PROGRAM SPECIFICATIONS

4.1 System Description

Figure 4-1 shows the system diagram. The system is vertically or heterarchically structured such that higher-level programs are involved at nearly all levels of processing of eye-hand-object data. This is shown in Figure 4-2.

4.1.1 Topological Vision Subsystem

The topological vision subsystem consists of the following programs or systems.

(1) Multiaperture image dissector TV control system. The multiaperture image dissector television device will probably be a further development of the image-dissecticon camera tube developed by Bendix Research Laboratories under contract to the Naval Weapons Research Center in Corona, California.[1, 2, 3, 4] The image-dissection is a high-sensitivity camera tube. It consists of an image dissector (described in section 3.1.2) combined with an image-intensifier, all in the same vacuum envelope as shown in Figure 4-3. Photoelectrons from the input photocathode are focused onto the microchannel plate, where they are amplified and projected onto the phosphor screen. The light emitted from the screen is imaged onto a second photocathode by a fiber-optic plate. The photocathode feeds the signal into an electro-static multiple aperture image dissector, with continous resistive-surface-deflection plates to deflect the image across the dissecting apertures. This Bendix device offers a promising direction for realizing 6 x 6 multiaperture tubes needed for high-resolution shape detection. It will be able to overcome losses associated with multiple elements and provides a low-light level capability for the image-dissector. It should be emphasized that Bendix does not consider the 6 x 6 format to be now feasible but it can be forecast for the 1975-1977 period. The TV control system has four basic functions: (1) pointing and focusing control (pitch, roll, yaw, focus, zoom adjustment) (2) camera tube control; (3) mode selection (scanning mode, non-scanned mode); (4) edge and shape detection logic control.

[1] J. M. Abraham, C.E. Catchpole, G. W. Goodrich, "Image Processing with Multiaperture Image Dissector," 12th Annual SPIE Technical Symposium, Los Angeles, California, August 7-11, 1967.

[2] C. E Catchpole, "Television Camera Tube Research at Bendix," (Fourth Symposium on Photoelectronic Image Devices, London, England, September 16-20, 1968.

[3] C. E. Catchpole, "Electron Image Amplification Using Microchannel Plates," Optical Society of America Spring Meeting, Philadelphia, Pa., April 10, 1970

[4] C. E. Catchpole, Photoelectronic Imaging Devices, Vol. 2, Chapter 8, "The Channel Image Intensifier," Plenum Press, 1971, pp. 167-190.

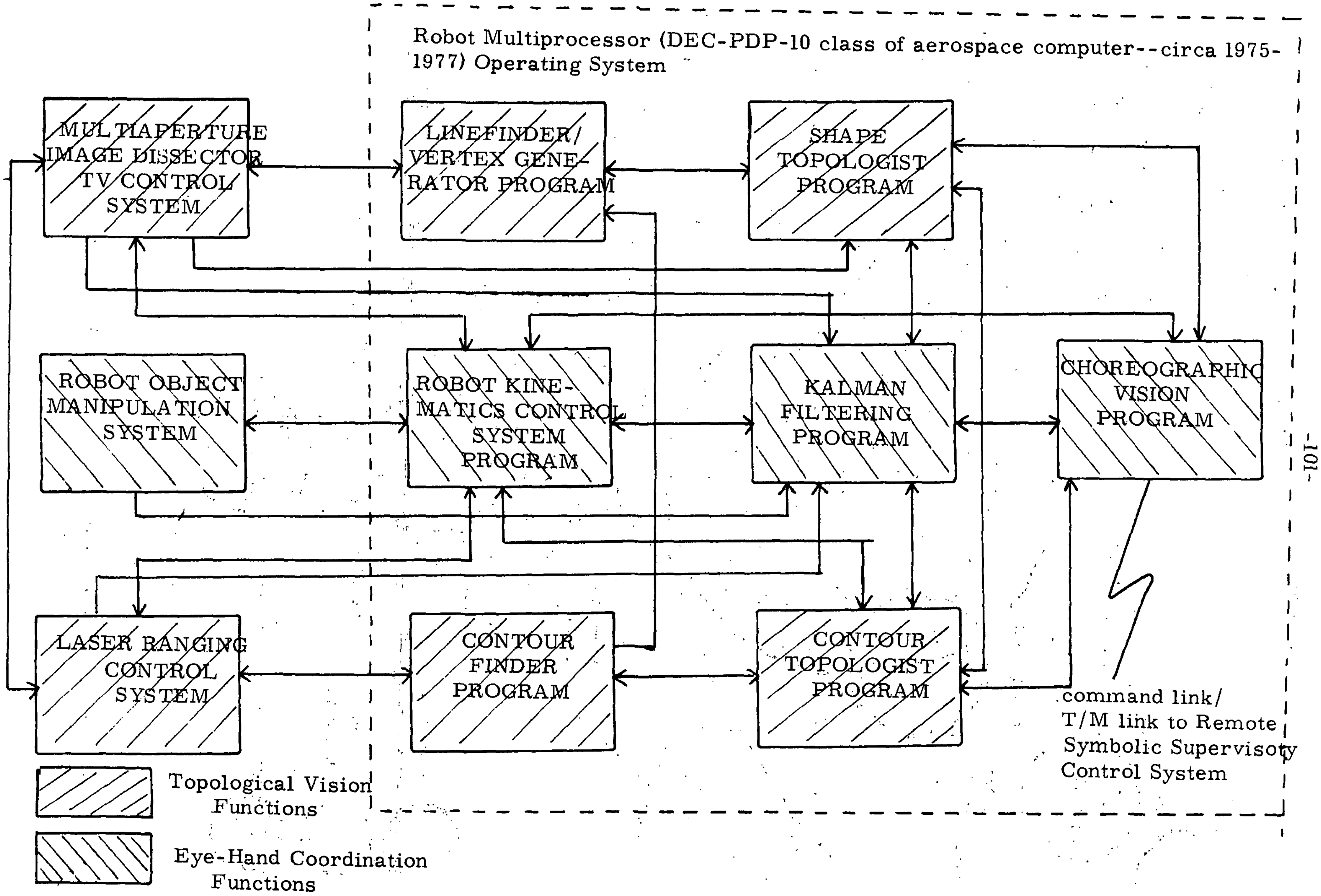

Robot Multiprocessor (DEC-PDP-10 class of aerospace computer--circa 1975-1977) Operating System
MULTIAPERTURE IMAGE DISSECTOR TV CONTROL SYSTEM
LINEFINDER/ VERTEX GENERATOR PROGRAM
SHAPE TOPOLOGIST PROGRAM
ROBOT OBJECT MANIPULATION SYSTEM
ROBOT KINEMATICS CONTROL SYSTEM PROGRAM
KALMAN FILTERING PROGRAM
CHOREOGRAPHIC VISION PROGRAM
LASER RANGING CONTROL SYSTEM
CONTOUR FINDER PROGRAM
CONTOUR TOPOLOGIST PROGRAM
command link/ T/M link to Remote Symbolic Supervisoty Control System
Topological Vision Functions
Eye-Hand Coordination Functions

Figure 4-1. System Diagram of Machine Eye-Hand Coordination and Topological Vision Model

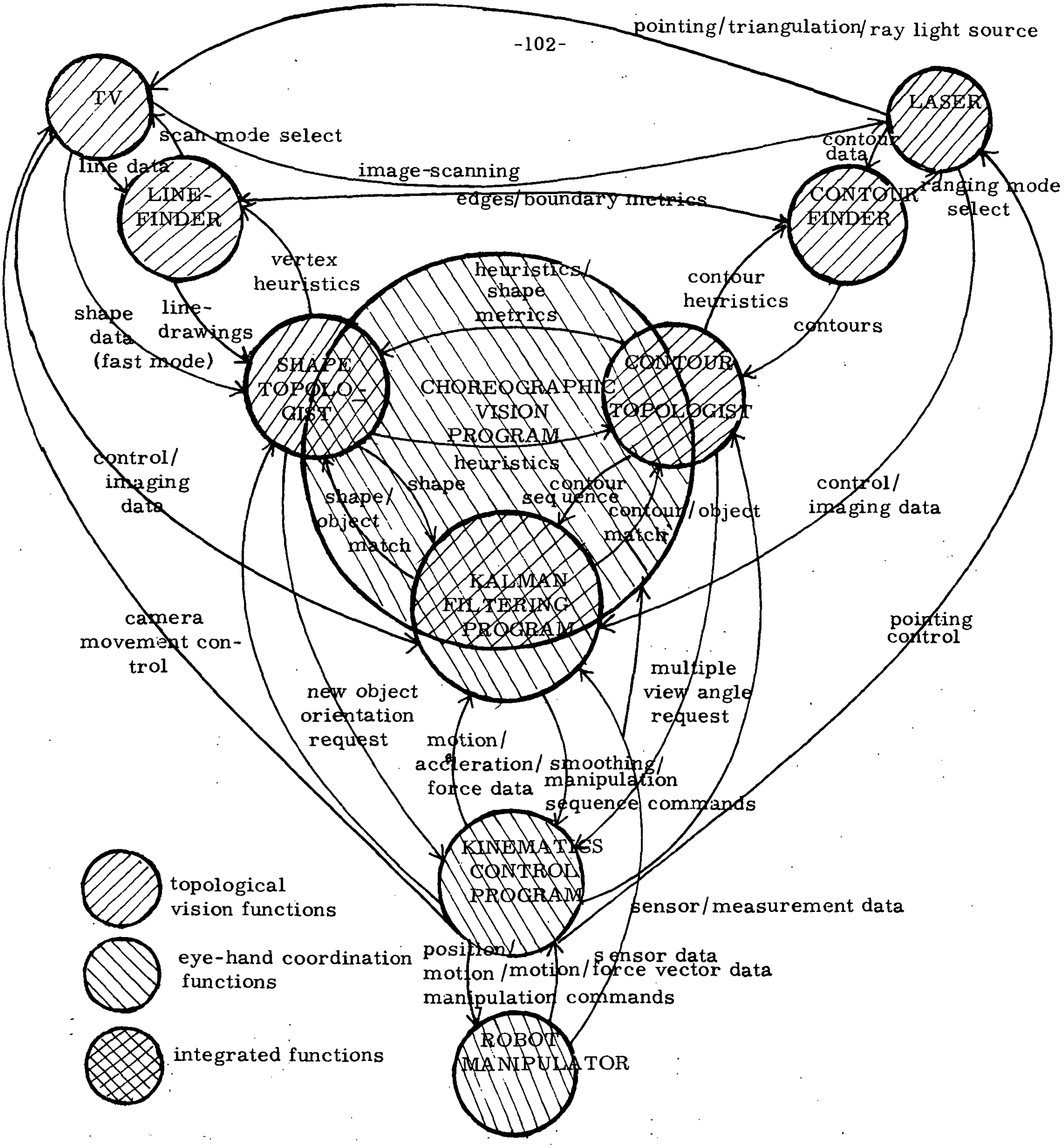


Figure 4-2. Heterarchical or Vertical System Organization

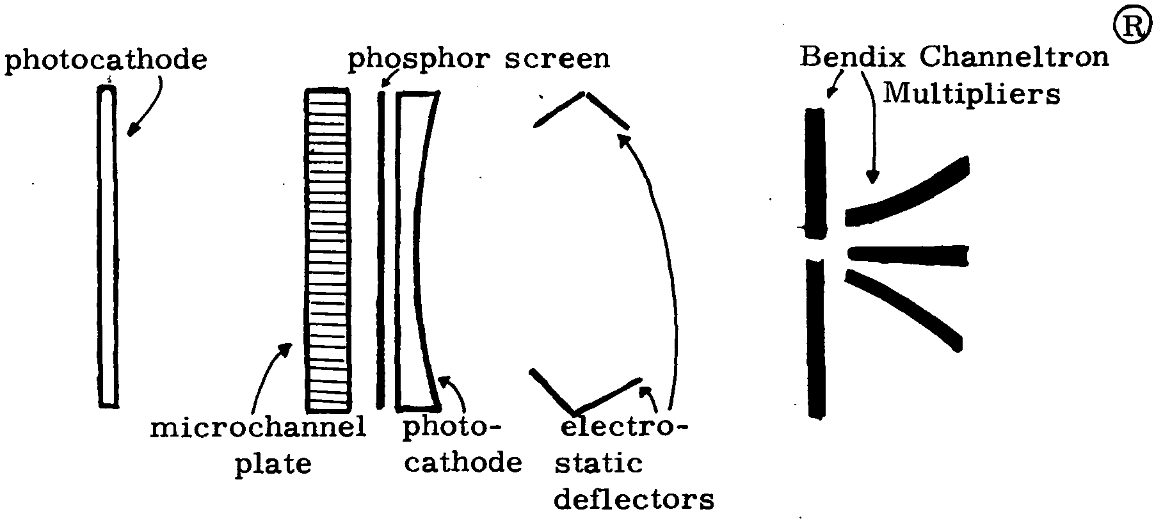


Figure 4-3. Bendix Dissecticon Tube

(2) Laser Ranging Control System. The laser ranging system will be based on testbed and prototype configurations under development at Stanford University and Stanford Research Institute. The Stanford University system can be regarded as a stereo-vision system. Corresponding points in the TV and laser views are uniquely located in space by the two angles in the two images as shown in Figure 4-4. The laser functions as a ray light source imaging a point. As a result, the TV camera follows the light source and thus imaging correspondence is obtained. In practice, the ray source is a plane source imaging a slit, such that a ray from the laser source projects along a line in the image. The Stanford configuration is shown in Figure 4-5. The Stanford experience to date indicates that color is crucial.[1] This means that the system would require a 3-color laser light source and a color-filter wheel used in conjunction with the TV image dissector cameras. The dissecticon variant of the image dissector provides (as noted earlier) low-light level capability. This means that in the projected system laser power can be correspondingly lowered. The Stanford lab system laser currently uses 30 MW. If we assume an additional ambient lighting source for hand-eye coordination tasks, then laser power can probably be lowered to about 10 MW.

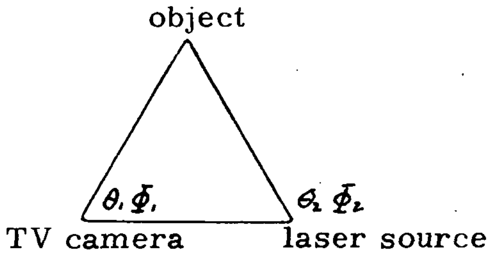


Figure 4-4 Stereo By Triangulation

(3) LINE-DRAWER. A line-finder locates lines of points having certain properties (such as being inhomogeneous or having a large light intensity gradient). A vertex finder merges concurrent lines, or creates a vertex at their meeting point.[2] The objective of this program is to produce a line-drawing with reasonably well formed vertices for subsequent shape encoding and analysis. A variant of the Binford-Horn class of linefinders will be required to fully exploit the inherent line-finding capability of the multiple aperture image dissector TV camera tube. The standard program is used with

[1] T. O. Binford, Artificial Intelligence Project, Computer Science Department, Stanford University, Notes Evaluating the Characteristics of Stanford Laser Ranging System, 30 November, 1971

[2] A. Guzman, "Computer Recognition of Three Dimensional Objects in a Visual Scene" MAC TR - 59 (Thesis), Project MAC, M.I.T. Cambridge, Mass., December, 1968, p. 60.

a random access single aperture image dissector.[1] The multiaperture format described in 3.1.2 can be used to simplify the generation of lines. Switching of the image dissector modes between scanned and unscanned modes can generate cues to discover lines and to verify the discovery of a line. Another simplification of the line-finder involves use of 3-dimensional data from the contour finder. The outside edge of an object can be correlated with the depth information from the laser-ranging processing. The proposed line-finding variant is shown in Figure 4-6.

(4) CONTOUR-FINDER in Figure 4-7 is analogous to the Binford-Horn LINE-FINDER variant. The CONTOUR-FINDER outputs to the topological contour recognition program the elements of a 3-dimensional representation. The program diagram is shown in Figure 4-7. Contours are encoded by two parameters: (a) the number of contour-components (i.e. simple closed curves) and (b) the depth or range-ordering of components (as measured along the optical axis of the laser beam). Scanning information from the LINE-DRAWER can then be used to assign metrics to the sizes of boundary curves which can be combined with metric information on depth or topographical maps derived from the laser.

(5) SHAPE-TOPOLOGIST in Figure 4-8 recognizes objects in hand-eye coordination task contexts on the basis of shape information. Two modes of operation will be available: (1) fast perceptual-motor mode in which shapes are recognized in the electro-optics by the multiaperture 6 x 6 format in conjunction with the straight line segment and curve encoding process. (2) vision mode in which shapes are analyzed in the software on a problem-solving basis by taking inputs from the LINE DRAWER. The program is structured into successive filters that correlate shape data of the object in the eye-hand contextual space with wagon-wheel shape-point set encodings or grass-fire encodings that are stored in object property lists. Successful recognition of a shape provides object identification and shape property information to the CHOREO VISION Program and outputs to the KALMAN FILTERING Program. Failure to recognize a shape activates a new object identification sequence and listing in cooperation with the Kinematics Control System which controls the eye-hand motor system. Finally, if the outcome is the identification of a new object, the grass-fire encoding of the new object shape is transmitted to the remote, symbolic supervisory control system on Earth.

(6) CONTOUR TOPOLOGIST in Figure 4-9 receives contour inputs from the CONTOUR FINDER and assembles a sequence of contours from a single viewing angle of the laser imaging system. On the basis of heuristics and Kalman filter corrections it then selects a small real number $\varepsilon > 0$ and defines the ε-neighborhood of each curve in the contour. A test for

[1] B. K. P. Horn, "The Binford-Horn LINEFINDER," Vision Flash 16, M.I.T. Artificial Intelligence Laboratory, undated.

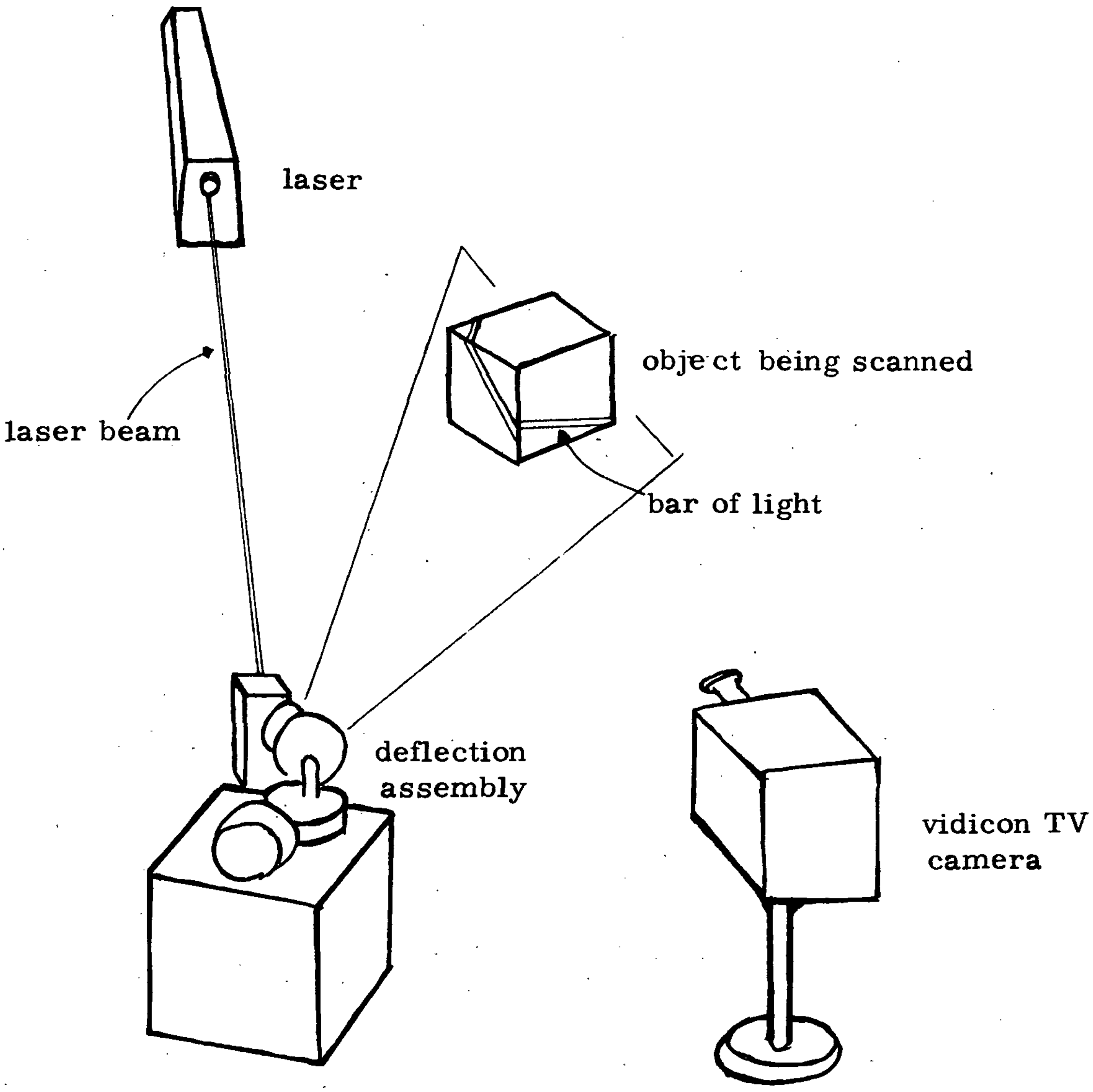


Figure 4-5. Stanford Uni versity Laser Scanning System

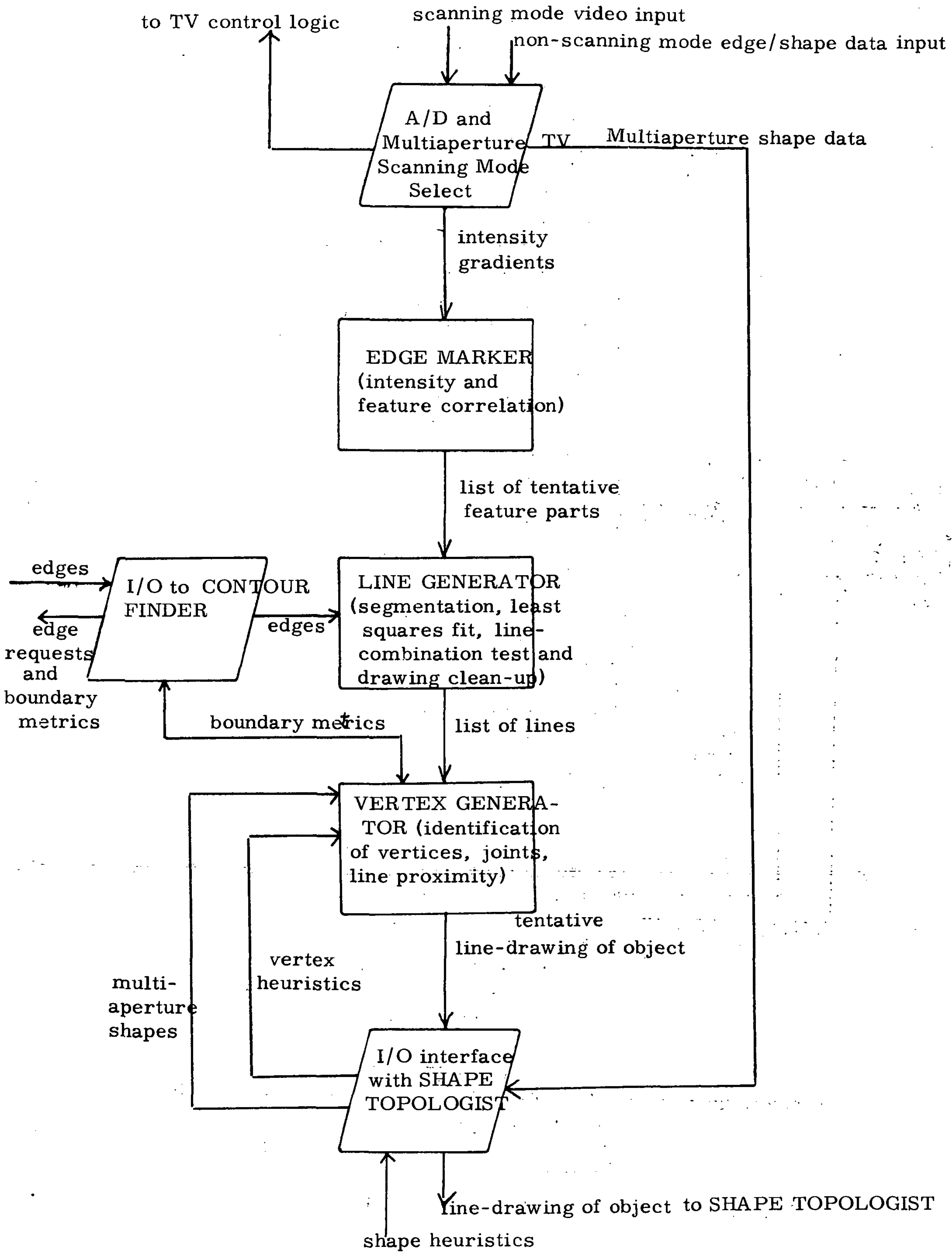

to TV control logic
scanning mode video input
non-scanning mode edge/shape data input
A/D and Multiaperture Scanning Mode Select
TV
Multiaperture shape data
intensity gradients
EDGE MARKER (intensity and feature correlation)
list of tentative feature parts
edges
I/O to CONTOUR FINDER
edge requests and boundary metrics
edges
LINE GENERATOR (segmentation, least squares fit, line-combination test and drawing clean-up)
boundary metrics
list of lines
VERTEX GENERATOR (identification of vertices, joints, line proximity)
tentative line-drawing of object
vertex heuristics
multi-aperture shapes
I/O interface with SHAPE TOPOLOGIST
line-drawing of object to SHAPE TOPOLOGIST
shape heuristics

Figure 4-6. Diagram of LINE DRAWER Program

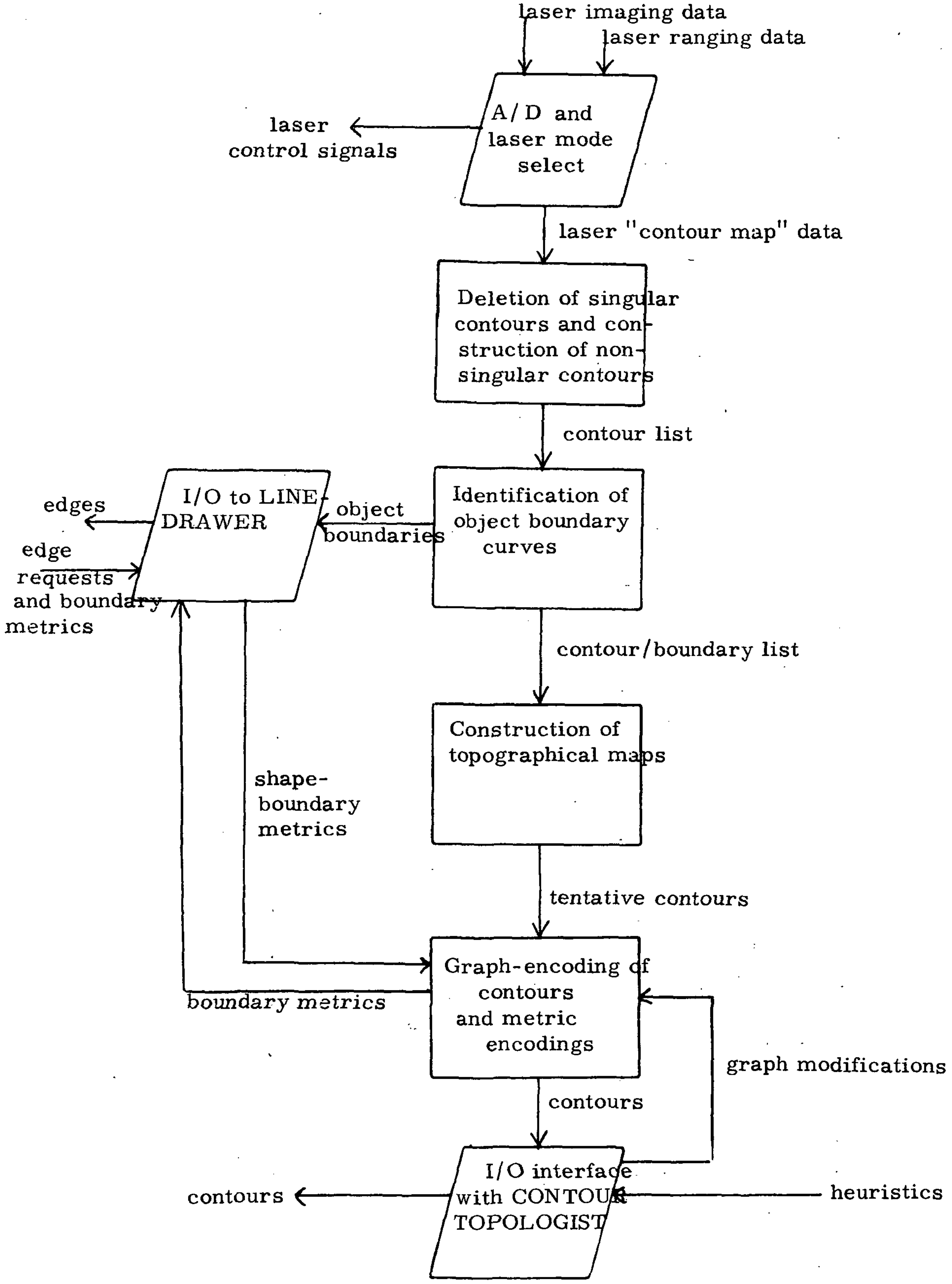

laser imaging data
laser ranging data
A/D and laser mode select
laser control signals
laser "contour map" data
Deletion of singular contours and construction of non-singular contours
contour list
Identification of object boundary curves
object boundaries
I/O to LINE-DRAWER
edges
edge requests and boundary metrics
contour/boundary list
Construction of topographical maps
shape-boundary metrics
tentative contours
Graph-encoding of contours and metric encodings
boundary metrics
graph modifications
contours
I/O interface with CONTOUR TOPOLOGIST
contours
heuristics

Figure 4-7. Diagram of CONTOUR-FINDER Program

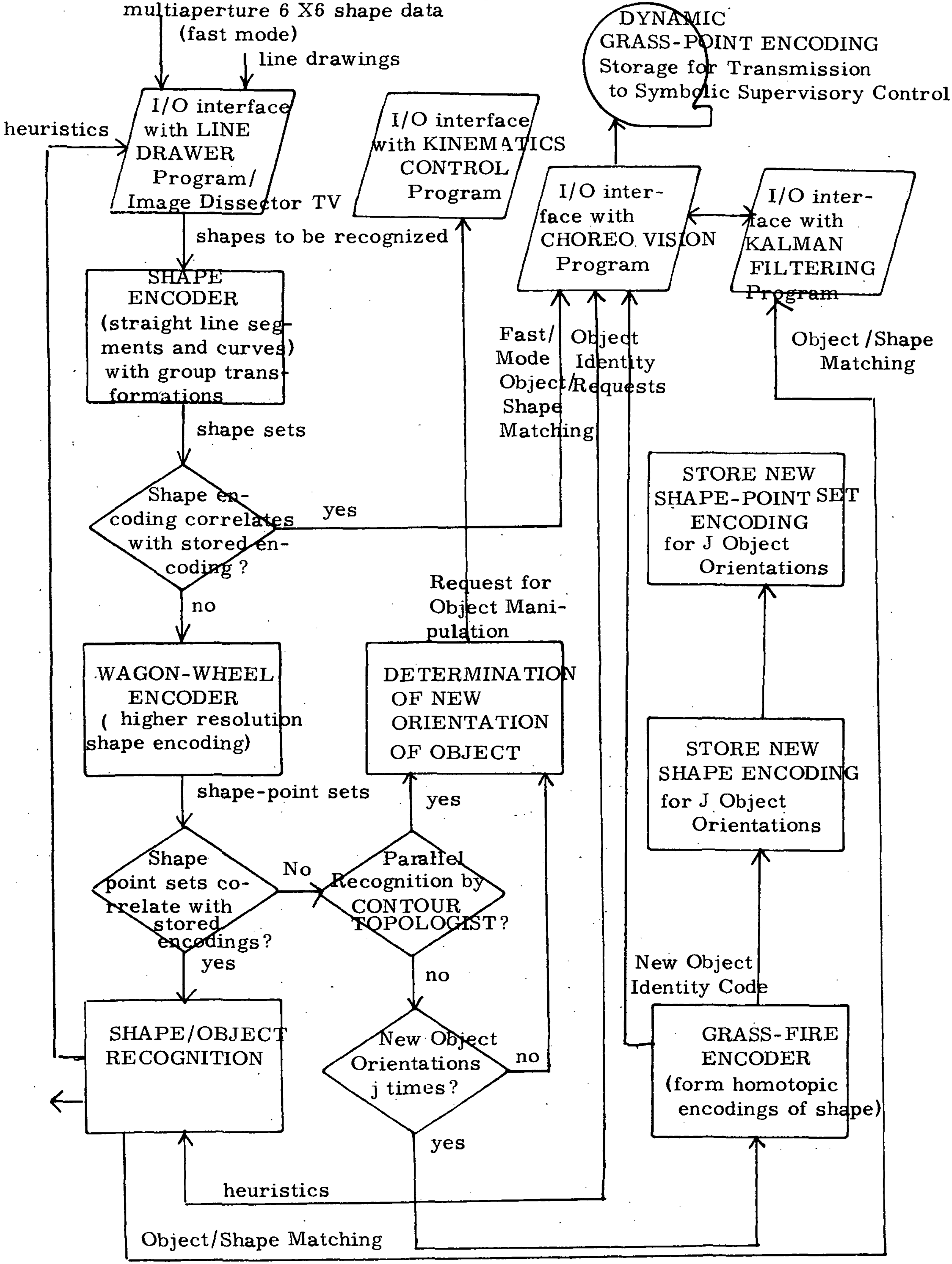

multiaperture 6 X6 shape data
(fast mode)
line drawings
heuristics
I/O interface with LINE DRAWER Program/ Image Dissector TV
I/O interface with KINEMATICS CONTROL Program
DYNAMIC GRASS-POINT ENCODING Storage for Transmission to Symbolic Supervisory Control
I/O inter-face with CHOREO VISION Program
I/O inter-face with KALMAN FILTERING Program
shapes to be recognized
SHAPE ENCODER (straight line segments and curves) with group transformations
Fast/ Mode Object/ Shape Matching
Object Identity Requests
Object /Shape Matching
shape sets
Shape encoding correlates with stored encoding ?
yes
no
STORE NEW SHAPE-POINT SET ENCODING for J Object Orientations
Request for Object Manipulation
WAGON-WHEEL ENCODER (higher resolution shape encoding)
DETERMINATION OF NEW ORIENTATION OF OBJECT
STORE NEW SHAPE ENCODING for J Object Orientations
shape-point sets
yes
Shape point sets correlate with stored encodings?
No
Parallel Recognition by CONTOUR TOPOLOGIST?
yes
no
New Object Identity Code
SHAPE/OBJECT RECOGNITION
New Object Orientations j times?
no
GRASS-FIRE ENCODER (form homotopic encodings of shape)
yes
heuristics
Object/Shape Matching

Figure 4- 8. Diagram of SHAPE TOPOLOGIST Program

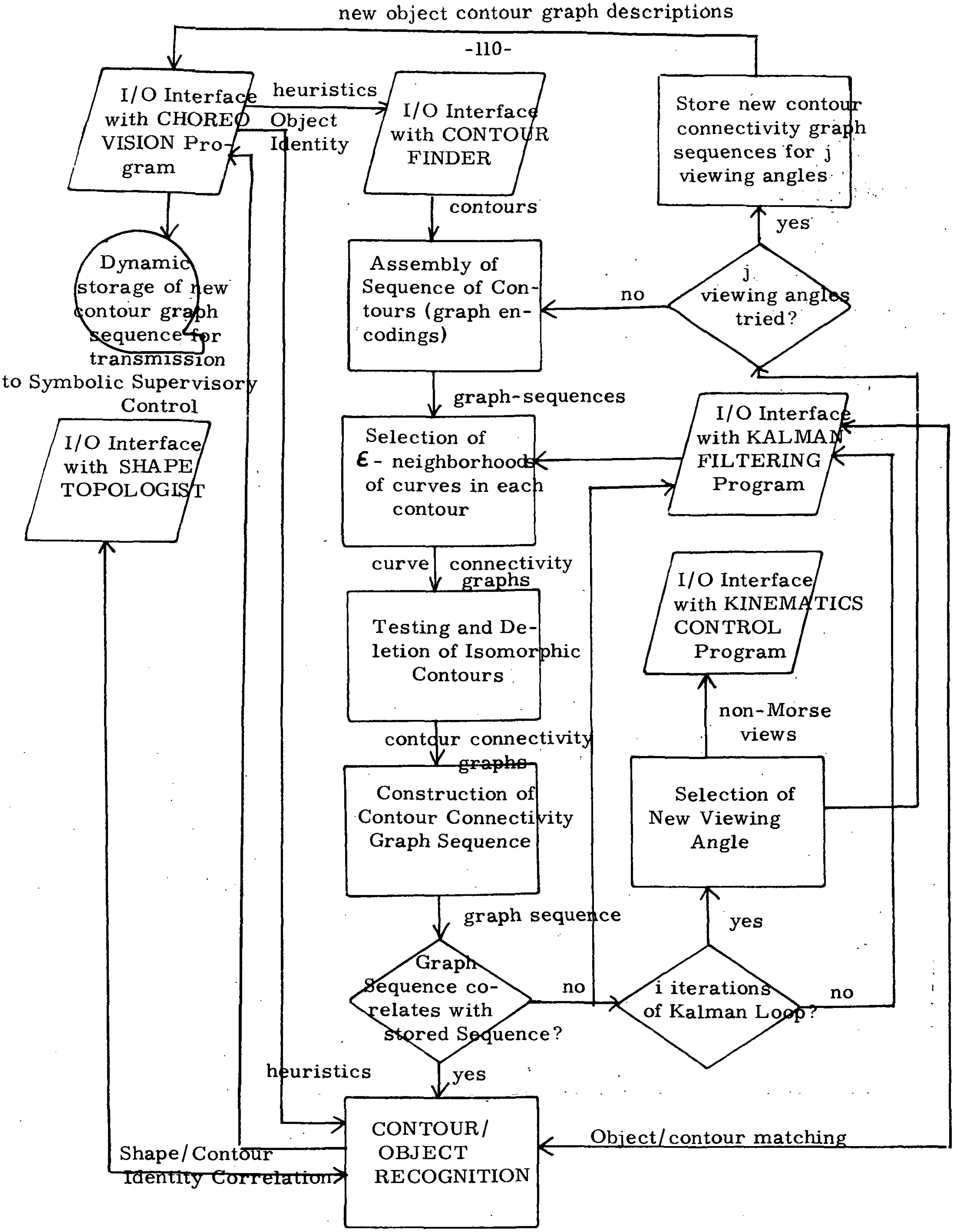

new object contour graph descriptions
I/O Interface with CHOREO VISION Program
heuristics
Object Identity
I/O Interface with CONTOUR FINDER
Store new contour connectivity graph sequences for j viewing angles
Dynamic storage of new contour graph sequence for transmission to Symbolic Supervisory Control
contours
yes
Assembly of Sequence of Contours (graph encodings)
no
j viewing angles tried?
graph-sequences
I/O Interface with SHAPE TOPOLOGIST
Selection of ε- neighborhoods of curves in each contour
I/O Interface with KALMAN FILTERING Program
curve connectivity graphs
Testing and Deletion of Isomorphic Contours
I/O Interface with KINEMATICS CONTROL Program
non-Morse views
contour connectivity graphs
Construction of Contour Connectivity Graph Sequence
Selection of New Viewing Angle
graph sequence
yes
Graph Sequence correlates with stored Sequence?
no
i iterations of Kalman Loop?
no
heuristics
yes
CONTOUR/ OBJECT RECOGNITION
Shape/Contour Identity Correlation
Object/contour matching

Figure 4-9. Diagram of CONTOUR TOPOLOGIST Program

isomorphic contours discards isomorphic contours and presents the latest non-isomorphic contour for a window of k-contour sections. The connectivity contour graph sequence is then constructed and a decision is made whether the graph matches a stored connectivity graph sequence. If no match is made, the KALMAN - $\mathcal{E}$ neighborhood selection loop is performed in a loop or iteratively until a match is made or the number i of iterations is reached. If a match is made, identification information is given to the parallel SHAPE TOPOLOGIST and the object identity and contour-properties are inputted to the CHOREO VISION program. If i iterations are performed without match, a new viewing angle is selected and the processing of contours from the new viewing angle is performed to find a match from the new angle. (This loop may also be iterated when a match has been found to verify the object identity). If, after j iterations of this contour-assembly and connectivity processing from multiple viewing angles, the object is not recognized, then a new object is identified and its contour-properties from several representative viewing angles are both entered onto tape for transmission to earth and inputted to the CHOREO VISION program.

4.1.2 Eye-Hand Coordination Subsystem

The eye-hand coordination subsystem has the following programs or systems.

(1) CHOREO VISION in Figure 4-10 is a high-level program that will use extensive heuristics. Like all such programs , CHOREO VISION will have empirically derived heuristics þrobably derived from the use of computer graphics) as well as those suggested by the logical model presented in this research. This program matches a stored or transmitted choreographic sequences (consisting of symbolic lists of eye-hand-object choreographic properties) with a list of choreogaphic properties inferred from eye-hand data. In this sense, the program as it were, matches the "cine" sequence of eye-hand coordination frames with choreographic goal frames that are represented symbolically on tape. If the system were to be a pure laboratory test bed we could use optical correlation techniques that would match a desired frame (stored picture) with the actual picture of the state of the object-hand system as derived in real-time from the computer eye. Such a "filmstrip" correlation system, however, is not feasible in terms of the transmittsion delays and bandwidth requirements for remote control of the robot on Mars. Thus we envision that the CHOREO VISION program and the entire integrated eye-hand coordination/topological vision system will be imbedded within a symbolic supervisory control system of the type being developed by M.I.T. Instrumentation Laboratory. This is shown in Figure 4-11.

CHOREO VISION forms machine concepts or machine models of the eye-hand scene and the resulting choreographic frames. This involves con-

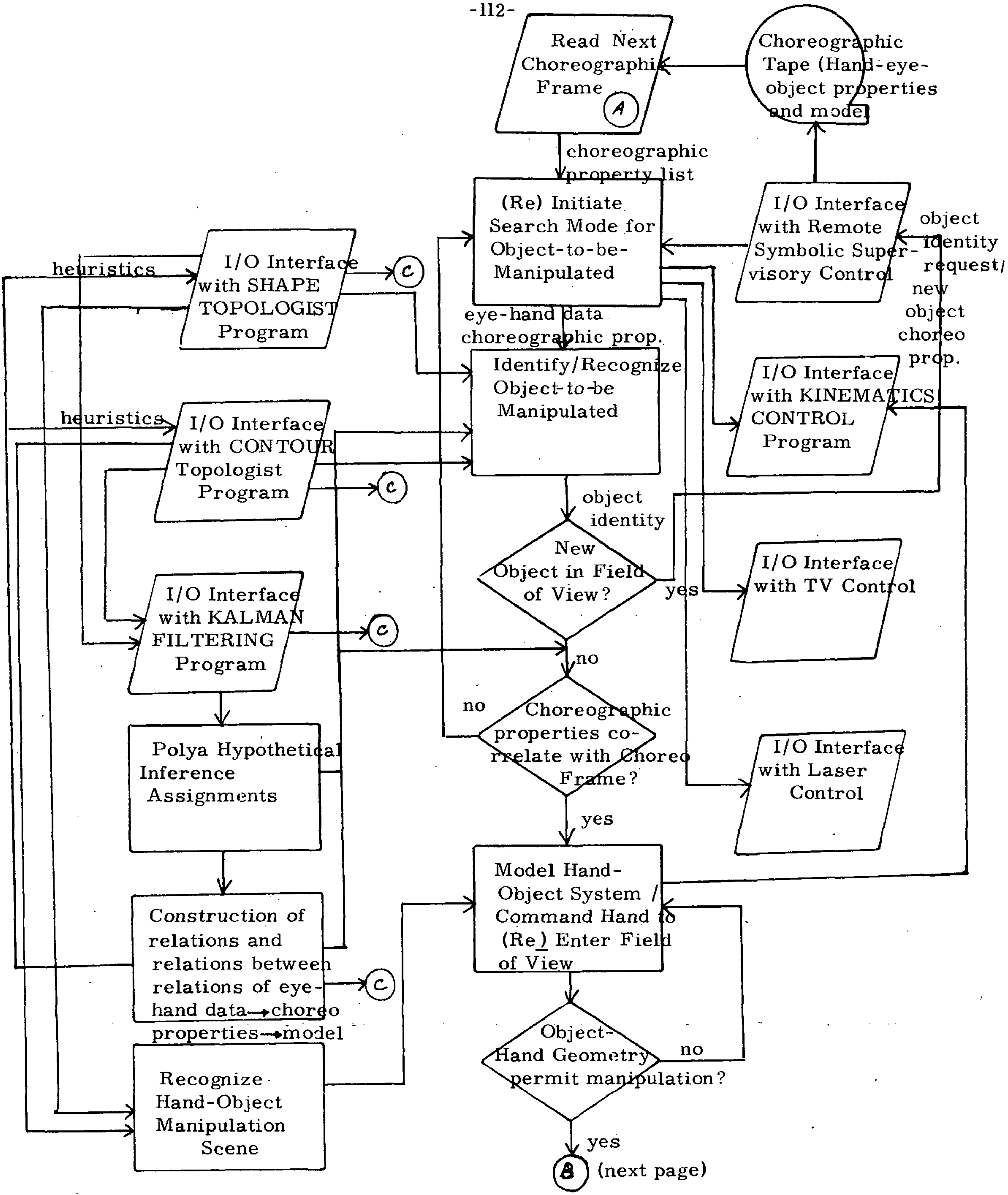

Read Next Choreographic Frame
A
Choreographic Tape (Hand-eye-object properties and model)
choreographic property list
(Re) Initiate Search Mode for Object-to-be-Manipulated
I/O Interface with Remote Symbolic Supervisory Control
object identity request/ new object choreo prop.
heuristics
I/O Interface with SHAPE TOPOLOGIST Program
C
eye-hand data choreographic prop.
Identify/Recognize Object-to-be Manipulated
I/O Interface with KINEMATICS CONTROL Program
heuristics
I/O Interface with CONTOUR Topologist Program
C
object identity
New Object in Field of View?
yes
I/O Interface with TV Control
I/O Interface with KALMAN FILTERING Program
C
no
no
Choreographic properties correlate with Choreo Frame?
Polya Hypothetical Inference Assignments
I/O Interface with Laser Control
yes
Model Hand-Object System / Command Hand to (Re) Enter Field of View
Construction of relations and relations between relations of eye-hand data→choreo properties→model
C
Object-Hand Geometry permit manipulation?
no
Recognize Hand-Object Manipulation Scene
yes
B
(next page)

Figure 4-10. Diagram of CHOREO VISION Program

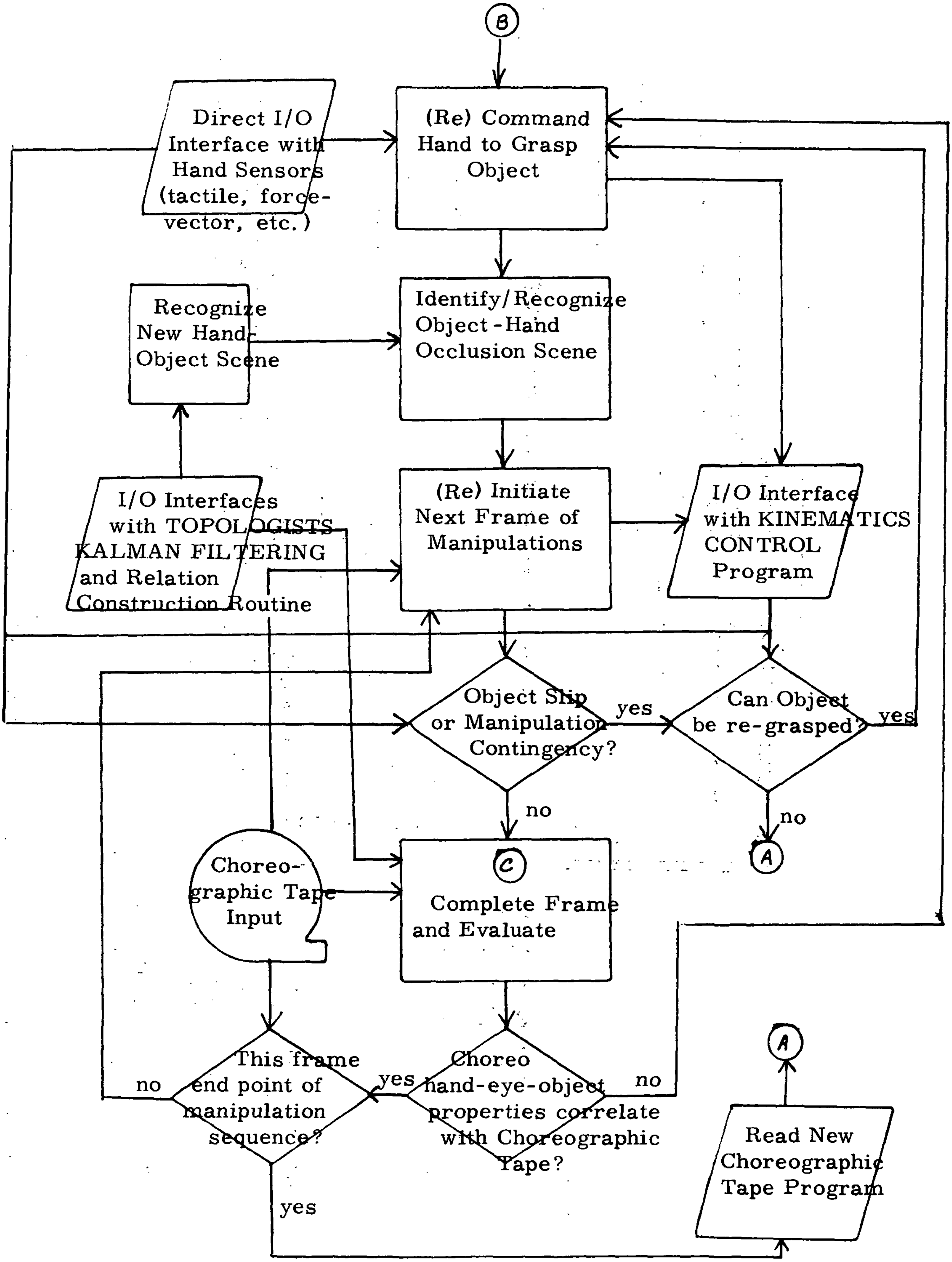

B
Direct I/O Interface with Hand Sensors (tactile, force-vector, etc.)
(Re) Command Hand to Grasp Object
Recognize New Hand-Object Scene
Identify/Recognize Object-Hand Occlusion Scene
I/O Interfaces with TOPOLOGISTS KALMAN FILTERING and Relation Construction Routine
(Re) Initiate Next Frame of Manipulations
I/O Interface with KINEMATICS CONTROL Program
Object Slip or Manipulation Contingency?
yes
Can Object be re-grasped?
yes
no
no
A
Choreo-graphic Tape Input
C
Complete Frame and Evaluate
This frame end point of manipulation sequence?
no
yes
Choreo hand-eye-object properties correlate with Choreographic Tape?
no
yes
A
Read New Choreographic Tape Program

Figure 4-10. Diagram of CHOREO VISION Program

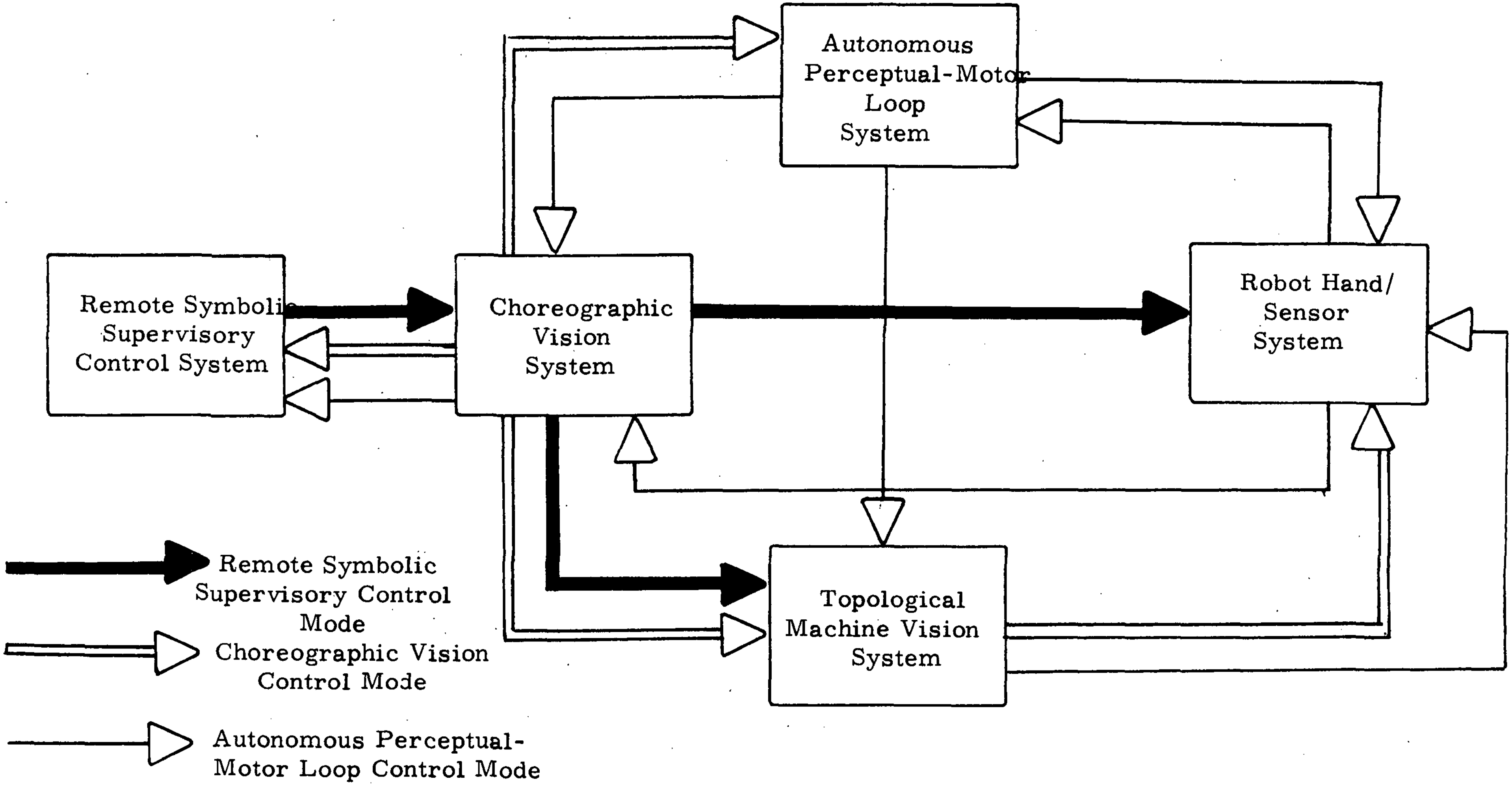

Autonomous Perceptual-Motor Loop System
Remote Symbolic Supervisory Control System
Choreographic Vision System
Robot Hand/ Sensor System
Topological Machine Vision System
Remote Symbolic Supervisory Control Mode
Choreographic Vision Control Mode
Autonomous Perceptual-Motor Loop Control Mode

Figure 4-11. Diagram of Multi-Moded Robot Control System

struction of eye-hand data ⟶ choreographic properties ⟶ choreographic models mappings via the context relation logic introduced in Section 2.5 Construction of relations and relations among relations provides a medium for machine concepts of eye-hand data in terms of choreographic properties and choreographic models.

(2) THE KALMAN FILTER ALGORITHM is described in Section 2.3. The flow-diagram is shown in Figure 4-12.

(3) KINEMATICS CONTROL is the basic digital-to-analog and analog-to-digital interface to the robot. It has the following functions: (a) transforming choreographed commands for manipulation sequences into robot hand/arm system motions, velocity, accelerations, decelerations, forces, torques; (b) transforming hand sensor and force-vector data for feedback to the CHOREO VISION Program via the KALMAN FILTER Program (including tactile, grasp, proximity, caliper, and other special purpose measurement data from the hand); (c) activation of the robot hand/locomotion system in response to commands either from the SHAPE TOPOLOGIST or CONTOUR TOPOLOGIST or the CHOREO VISION Programs; (d) integration of interactions between pointing and optics control and robot posture and arm/hand motions. This program is a high-level program that models the manipulation space as an interactive space in which choreography can be performed. Instead of commanding robot motion paths by a coordinate system, robot motions and manipulations are synthesized from the basic motion interactions of the robot hand with an object. For example, a typical command might be FIND OBJECT IN FIELD OF VIEW AND GRASP. This would be a macro which directs the basic motions of the robot through a manipulation sequence. The space in which the robot moves is thus modelled not as a volume defined by sets of coordinates for each degree of freedom, but as an interactive space in which motions can be choreographed to fit a given choreographic vision task. Development of the detailed logic for this program will require further research in choreographic programming languages.

4.2 Program Description

4.2.1 System Guidelines

At the outset of this research, it was thought that individual functions such as computer vision within an eye-hand coordination context could be implemented on a minicomputer class of machines that can be expected in the 1975 era. This, however, is clearly not the case when we consider the complexity of integrated computer vision and eye-hand coordination in the semi-autonomous mode described in Section 4.1. Our conclusion is that even with symbolic supervisory control and choreographic programming --- a system environment that minimizes the levels of artificial intelligence in the projected Viking robot system --- overall program size and real-time

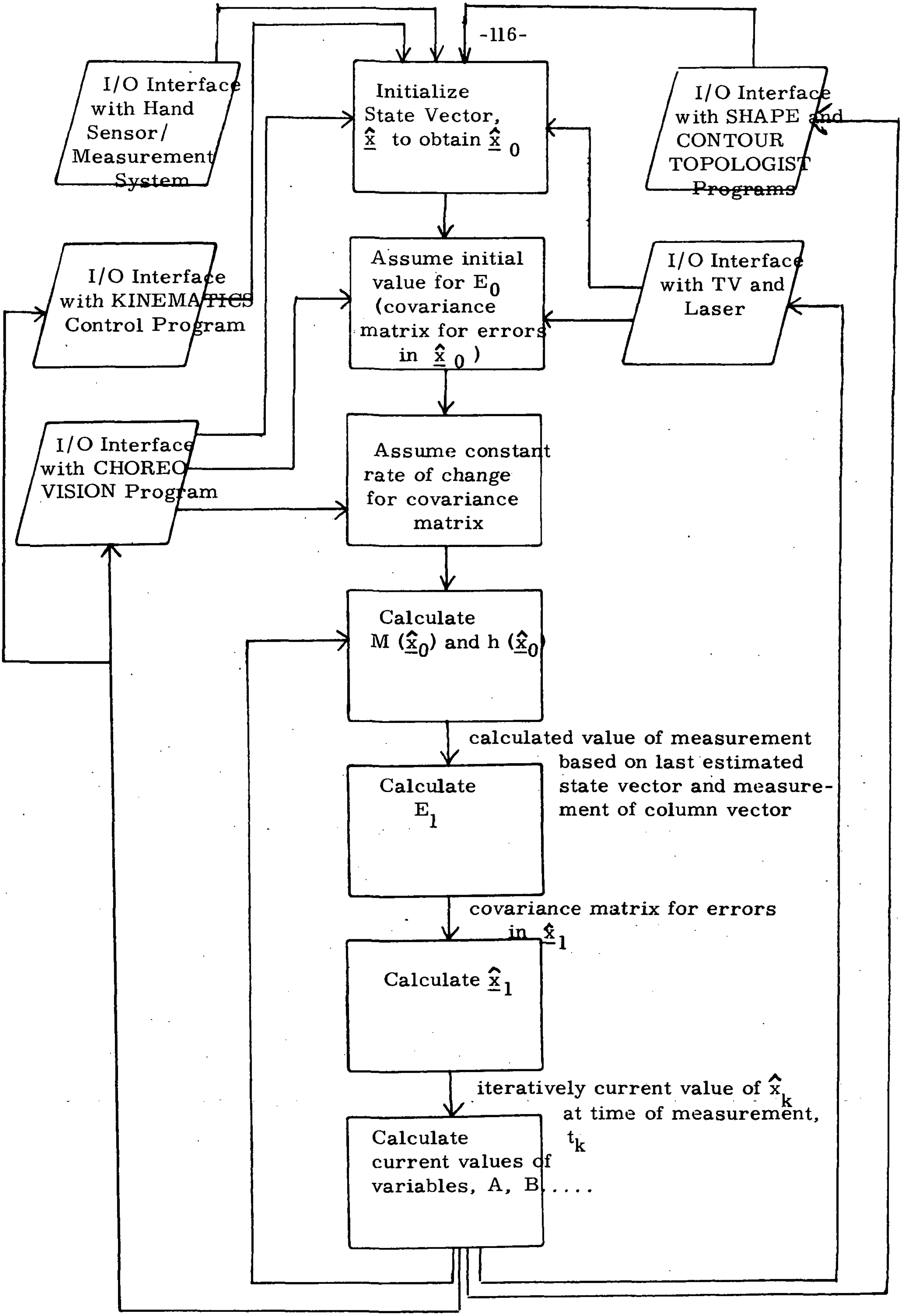

I/O Interface with Hand Sensor/ Measurement System
Initialize State Vector, $\hat{\underline{x}}$ to obtain $\hat{\underline{x}}_0$
I/O Interface with SHAPE and CONTOUR TOPOLOGIST Programs
I/O Interface with KINEMATICS Control Program
Assume initial value for E_0 (covariance matrix for errors in $\hat{\underline{x}}_0$)
I/O Interface with TV and Laser
I/O Interface with CHOREO VISION Program
Assume constant rate of change for covariance matrix
Calculate M ($\hat{\underline{x}}_0$) and h ($\hat{\underline{x}}_0$)
calculated value of measurement based on last estimated state vector and measurement of column vector
Calculate E_1
covariance matrix for errors in $\hat{\underline{x}}_1$
Calculate $\hat{\underline{x}}_1$
iteratively current value of $\hat{x}_k$ at time of measurement, t_k
Calculate current values of variables, A, B, ...

Figure 4-12. Diagram of KALMAN FILTERING Program

control requirements will require an aerospace multiprocessor whose performance can be compared to today's standard artificial laboratory computer, the DEC PDP-10. In some respects, we may require faster machines with more sophisticated real time operating characteristics. Implementation of the system described in Section 4.1 will require at least the following capabilities.

(1) 1/3 microsecond cycle time
(2) 256 K 36-bit words
(3) privileged mode instruction
(4) 16 general registers

Table 4-1 summarizes estimated requirements for the individual programs shown in Figure 4-1.

Program	Estimate % of Computer Time	Estimated Frequency of Execution	Estimated Program Size (36-bit words)	Estimated Dynamic Storage Requirements
LINE-FINDER/ VERTEX GENERATOR	5%	once per viewing angle (in slow mode)	16K	Requires 4K dynamic storage for TV data; bypassed in fast mode shape recognition when Image Dissector provides shape information.
SHAPE TOPOLOGIST	10%	once per viewing angle	20K	Grass-fire encoding option will require 4K
CONTOUR FINDER/ ENCODER	10%	once per viewing angle	32k	Requires dynamic storage for laser data.
CONTOUR TOPOLOGIST	15%	once per viewing angle	40K	Requires 8K of dynamic storage for contour data.
KALMAN FILTER	5%	continuous	20K	
CHOREO VISION	25%	continuous	64K	Requires 16K of dynamic storage for transmittal to Remote Symbolic Supervisory Control and 32K of dynamic storage for choreographical program.
ROBOT KINEMATICS CONTROL	10%	continuous during choreographic task execution	32K	
OPERATING SYSTEM	20%	continuous	32K	

Table 4-1 Estimated Program Requirements

CyberFacts, Inc.

101 Moore Road, Sudbury, Massachusetts 01776 (617) 443-8900