(12) LEVEL III

RADC-TR-81-190, Vol II (of two)
Final Technical Report
September 1981

AD A108793

# ADVANCED WEAPON SYSTEM (AWS) SENSOR PREDICTION TECHNIQUES STUDY

General Electric Company

C. F. R. Weiman

(AWS) SENSOR PREDICTION TECHNIQUES STUDY, VOL II

DTIC FILE COPY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
ELECTE
S DEC 23 1981 D
B

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

81 12 23 077

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

Because of the size of this document, it has been divided into two volumes. Volume I contains Sections 1 through 5. Volume II contains Appendix IV.

RADC-TR-81-190, Volume II (of two) has been reviewed and is approved for publication.

APPROVED: Robert H. La Salle

ROBERT H. LA SALLE
Project Engineer

APPROVED:

OWEN R. LAWTER, Colonel, USAF
Chief, Intelligence and Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRP) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
RADC-TR-81-190, Vol II (of two)

2. GOVT ACCESSION NO.
AD-A108 793

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
ADVANCED WEAPON SYSTEM (AWS) SENSOR PREDICTION TECHNIQUES STUDY. Volume II

5. TYPE OF REPORT & PERIOD COVERED
Final Technical Report
14 Feb 80 - 11 May 81

6. PERFORMING ORG. REPORT NUMBER
N/A

7. AUTHOR(s)
C.F.R. Weiman

8. CONTRACT OR GRANT NUMBER(s)
F30602-80-C-0082

9. PERFORMING ORGANIZATION NAME AND ADDRESS
General Electric Company
P O Box 2500
Daytona Beach FL 32015

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62702F
45941718

11. CONTROLLING OFFICE NAME AND ADDRESS
Rome Air Development Center (IRRP)
Griffiss AFB NY 13441

12. REPORT DATE
September 1981

13. NUMBER OF PAGES
161

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Same

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
Same

18. SUPPLEMENTARY NOTES
RADC Project Engineer: Robert H. LaSalle (IRRP)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
AWS, Digital Data Base, Sensor Simulation, Digital Landmass Simulator, Image Prediction, FLIR, Computer Graphics, CIG

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This study investigated, identified, and developed technology necessary to support efficient computer simulation of Advanced Weapon Systems (AWS) sensor images associated with low altitude flight. A main objective of the study was to delineate approaches to the solution of data base storage and processing problems characteristic of low altitude flight sensor image simulation. The tasks carried out to achieve this objective included analysis of scene content requirements, a survey of relevant computer

14

graphics and image processing technology, and evaluation of experimental image generation algorithms carried out by General Electric. The functional description of a research laboratory system for sensor display simulation is presented.

One of the most significant conclusions drawn from the analysis of technology and experimentation is that grid data base representation and display may dramatically improve image quality and reduce cost of sensor simulations vis-a-vis polygon data base representation in the near future. This is ude to the recent advances in VLSI architecture and the discovery of new parallel algorithms in computer graphics.

## TABLE OF CONTENTS - VOLUME 2

Accession For
NTIS GRA&I ☑
DTIC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A

## APPENDIX IV. 1

## SCENE ANALYSIS: A SURVEY

Presented in this appendix is a survey which traces the development of scene analysis by computer from its origins in digitized picture processing and pattern recognition, and discusses geometric concepts related to projection. New approaches based on projective geometry and neurophysical models are suggested.

Courant Computer Science Report #9

December 1975

# Scene Analysis: A Survey

Carl Weiman

Courant Institute of
Mathematical Sciences

Computer Science Department

New York University

Report No. NSO-9 prepared under Contract
No. N00014-75-C-0571
with the Office of Naval Research

## COURANT COMPUTER SCIENCE PUBLICATIONS

Price

### COURANT COMPUTER SCIENCE NOTES

Programming Languages and Their Compilers,
J. Cocke and J. T. Schwartz, 2nd Revised Version,
*April 1970, iii+767 pp.* $19.25

On Programming: An Interim Report on the SETL Project.
Part I: Generalities.
Part II: The SETL Language and Examples of Its Use.
(Parts I and II are consolidated in this volume.)
J. T. Schwartz, Revised *June 1975, xii+675 pp.* 17.25

A SETLB Primer. H. Mullish and M. Goldstein, *1973, v+201 pp.* 5.25

Combinatorial Algorithms, E. G. Whitehead, Jr., *1973, vi+104 pp.* 2.75

### COURANT COMPUTER SCIENCE REPORTS

No. 1 ASL: A Proposed Variant of SETL
Henry Warren, Jr., *1973, 326 pp.*

No. 2 A Metalanguage for Expressing Grammatical Restrictions in Nodal Spans Parsing of Natural Language,
Jerry R. Hobbs, *1974, 266 pp.*

No. 3 Type Determination for Very High Level Languages,
Aaron M. Tenenbaum, *1974, 171 pp.*

No. 4 A Comprehensive Survey of Parsing Algorithms for Programming Languages, Phillip Owens, *Forthcoming.*

No. 5 Investigations in the Theory of Descriptive Complexity,
William L. Gewirtz, *1974, 60 pp.*

No. 6 Operating System Specification Using Very High Level Dictions, Peter Markstein, *1975, 152 pp.*

No. 7 Directions in Artificial Intelligence: Natural Language Processing, Ed. Ralph Grishman, *1975, 107 pp.*

No. 8 A Survey of Syntactic Analysis Procedures for Natural Language, Ralph Grishman, *1975, 94 pp.*

No. 9 Scene Analysis: A Survey, Carl Weiman, *1975, 62 pp.*

A catalog of SETL Newsletters and other SETL-related material is also available. Courant Computer Science Reports are available upon request without charge Prepayment is required for Courant Computer Science Notes. Please address all communications to

COURANT INSTITUTE OF MATHEMATICAL SCIENCES
251 Mercer Street
New York, N. Y. 10012

COURANT INSTITUTE OF MATHEMATICAL SCIENCES

Computer Science NSO-9

# SCENE ANALYSIS: A SURVEY

Carl Weiman

Report No. NSO-9 prepared under
Contract Number N00014-75-C-0571
with the Office of Naval Research

## Table of Contents

# CHAPTER 1. INTRODUCTION

Computers are often considered extensions of the mind in the same sense that manual tools are extensions of the hand. In this machine-man analogy, physical measuring devices (photocells, pressure sensitive switches, etc.) are the sensory cells of organs of sensation and the computer receiving messages from them functions as an organ of perception. In many applications of the analogy, e.g. process control, the computer exhibits mere sensation and reflexes. In artificial intelligence (henceforth AI), however, computer systems are designed expressly to exhibit perception, comprehension, curiosity, and intention. Though rather anthropomorphic, this aim is a primary motivation for most AI researchers and raises many important philosophical and psychological questions whose discussion has already changed our way of looking at human intelligence. Naive popular interpretations of AI include conjurations of evil or benevolent robots or, no less anthropomorphic, the amoral oracle which answers questions truthfully but refuses ("stubbornly, proudly") to consider the good or evil consequences. The escalating impact of computers on society elevates the importance of philosophical questions about AI from resolving esoteric problems to making decisions about social policy. The reader is directed to an eloquent examination of the philosophical questions by Turing (BIBGEN 1947)* and an investigation of the social impact of AI by Firschein et al. (BIBGEN 1973). Such philosophical questions are far beyond the level of this survey. However, the psychological analogy is useful in introducing a unified analysis of artificial visual systems whose characteristics would otherwise appear quite diverse. Since artificial visual systems are often aimed at imitating, improving on, or generalizing from natural visual systems, the analogy is not as far-fetched or poetic as it might seem.

* See Chapter 4 for key to bibliographic references in the text.

For the purposes of this paper I will define scene analysis as the computer processing of two-dimensional projected images of three-dimensional scenes, usually typical of what humans see in everyday life, to yield a data structure which somehow captures the individual identity of and spatial relations between, objects in the 3-D world. Scene analysis is then computer visual perception and comprehension. This rather narrow definition deliberately excludes holography which, though it can give an accurate reversible (i.e. yielding original 3-D surface), representation of three dimensional objects, treats the visual world as a mathematical surface devoid of meaning. I also exclude picture processing which though often applied to enhance images of objects to be treated by human or machine viewers as three dimensional, (e.g. stereophotos in cartography) has not until recently addressed itself directly to the data structure mentioned in this paragraph's opening sentence. The two preceding areas are not meant to be slighted by omission; they are among the most fruitful application areas of computer processing of pictorial information. They are certainly in a more advanced state of application than scene analysis. Their strong relation to scene analysis and the area of overlap will be discussed. In this survey, I restrict the scope more to semantic, relational aspects involved in the representation of objects rather than the purely numerical, mathematical surface defined by the locus of surfaces intersected by the line of sight. "Semantics" can mean a number of things, so let me exclude also the most common meaning in AI, viz., semantic data structures such as Winograd's (BIBGEN 1972) where physical objects are represented devoid of their geometric coordinates in terms of labelled items and their relations in list structures for the purpose of manipulation and inference. This too is a very important and closely related area whose connections with scene analysis will be discussed though not thoroughly surveyed. It is fraught with many of the open-ended problems of the general representation of knowledge in AI independent of vision. We then wedge ourselves narrowly between applied mathematics (physics, geometry and computer processing of pictures) and semantics (knowledge, meaning), using both for support. I hope to show that many important advances in scene analysis implicitly embody the curious semantics of projective geometry, but often expressed in forms specifically suited to optics and digital computation rather than the general axiomatic form used in mathematics.

This survey consists of four chapters and a bibliography. Chalter 1, which you are now reading, is a short introduction to scene analysis and guide to the rest of the paper. Chapter 2 is the most important one. It consists of a history and state of the art description of scene analysis techniques. It is organized according to type of approach, in roughly historical order. Chapter 3 is a conclusion which contains a critical overview of the techniques and a preview of methods which might be successfully applied in the future. Chapter 4 is a guide to the bibliography. This includes a description of the kinds of sources available and their relation to research institutions. Chapter 4 should be skimmed immediately in order to understand the format of references to the bibliography in the text of this survey.

# CHAPTER 2. SCENE ANALYSIS METHODS

## 2.1 Introductory Remarks

An important goal of scene analysis research is to build artificial eyes connected with control systems which enable a robot to manipulate and/or maneuver in its environment. Applications include the design of visual systems for industrial robots, automatic pilots for vehicles, mechanical extraterrestrial explorers, and other devices. To organize the discussion of diverse approaches, scene analysis methods are subjectively divided into several areas in this chapter. Two major areas are line analysis and region analysis. The former is based on classifying and relating to each other the 2-D images of 3-D edges and vertices of polyhedra in a scene. The latter is based on merging adjacent picture regions with similar properties. These and other areas will be described and compared. The order of discussion will be historical except when that conflicts with organization by area.

Many of the techniques discussed appear superficially to differ greatly from each other. A deeper analysis in terms of projective geometry, however, often points out implicit exploitation of similar phenomena. This underlying connection is not often discussed in the literature but will be brought out to unify description of techniques in this chapter and suggest fruitful areas for new research in the next.

Scene analysis owes much to earlier work in picture processing and pattern recognition. These were the first areas in which methods were developed for representing, analyzing, and manipulating pictorial (i.e. 2-D) information by computer. This essentially geometric information processing is quite different from more traditional number crunching or text manipulation computer applications. Among the differences are that picture manipulation requires much more processible memory, and the processes are conceptually two-dimensional rather than one-dimensional. The geometric problems and techniques of pattern recognition and scene analysis have much in common but also some crucial differences. In the psychological model of mechanical vision, the aim of scene analysis is to perceive and understand 2-D images of 3-D scenes. The meaning of this analogy can be clarified using a rudimentary informational model; this yields a natural hierarchy from physical measurement through pattern recognition to scene analysis. The nature of this hierarchy is examined in the following discussion of picture processing and pattern recognition. The reader can find comprehensive bibliographies of picture processing accompanied by clear descriptions of techniques in Rosenfeld's excellent surveys (BIBPIC 1969,1972-1975). Any of the titles containing the words "Pattern Recognition" in BIBPIC can introduce the reader to that topic.

## 2.2 Pattern Recognition and Scene Analysis

A physical measuring instrument can be considered to be a finite state device, each of whose states corresponds to a range of physical conditions of the object it is measuring. Rothstein (BIBGEN 1956) has shown how the amount of information about the physical object derived from such a measurement can be formally defined in terms of thermodynamic entropy, thereby linking computational and physical concepts. Though the measuring device is finite state, and information consists of reduction of uncertainty about which of these states correctly describes the measured object, the physical state space is usually thought of as a one-dimensional continuum with a metric. That is, the finite set of states is well-ordered and can be divided into subsets with that same property without limit (until the quantum uncertainty limit) and differences between values of state variables have meaningful (comparable) magnitudes. The psychological analogue of such information could be termed a sensation such as the sensation of temperature, brightness, or pressure; these quantities are usually considered as belonging to one-dimensional intensity continua. The problem of classifying such information is usually trivially solved by dividing the continuum into ranges and ordering observations relative to the limits of these ranges.

Pattern recognition involves the classification of patterns consisting of the cartesian products of many (usually) measurements into a small (usually) number of sets. The large dimensionality of measurements not only obliterates the simple ordering relation described in the preceding paragraph but also yields astronomical numbers of distinct possible patterns. A common approach to simplifying classification is to pre-process the information by extracting features, abstractions of simple properties defined by relations between members of subsets of the pattern space. The number of features is usually far smaller than the number of measurements and they are chosen so that they measure some pattern properties related to the desired classification. The cartesian product of a number of features constitutes feature space; points in this space derived from patterns which are to be assigned to the same set in pattern classification ought to cluster together more than points derived from differing patterns. By defining a distance function between points in feature space, classification is achieved by assigning a new point to the nearest cluster. The kinds of features chosen are crucial to effective clustering and highly task specific. Many of the tools of pattern recognition are derived from sophisticated statistical decision theory applied to the assignment of points in feature space to clusters. Tou and Gonzalez (BIBPIC 1974) contains detailed description of these tools and a good bibliography for other sources in pattern recognition. In our psychological analogy, if measurement is associated with sensation, pattern recognition is associated with perception. Assigning large numbers of distinct patterns to a single class corresponds to recognizing a form or percept in any one of a large number of differing particular presentations.

In visual (also called optical or pictorial) pattern recognition, the patterns to be classified are usually gray-level (also called grey-scale) pictures. These are 2-D pictures which have been divided into cells by a regular grid. Each cell is characterized by a single number (measurement) corresponding to the integrated intensity of light throughout that cell. Any picture is then a pattern or point in a space with as many dimensions as there are cells in the grid. Hence, for an $n \times n$ grid, the pattern space has dimension $n^2$. The earliest efforts in pictorial pattern recognition were directed to the problem of recognizing printed characters such as numerals or letters of the alphabet by template matching, described as follows. The gray-level in each cell is either 1 or 0 depending on whether ink (darkness) is present or absent in that part of the picture. The ideal example of any character is called a template and is represented by the $n^2$ component vector of 1's and 0's. Classification of other samples is achieved by identifying each with the ideal (template) vector it best matches; by "best" is meant that for which the largest number of corresponding components match in the two vectors. The method fails completely if the sample to be recognized is not almost identical in size, position and orientation to the template. Two-dimensional transformations "scramble" the $n^2$-dimensional vector. This limitation can be overcome by exploiting the fact that the $n^2$ vector can be "unfolded" into the two-dimensional picture space. In this much more tractable space, applying simple geometric transformations can invert the effects of 2-D translation, rotation, scale change and other one-to-one transformations to yield a pattern in registration with any template. Such techniques have been applied to yield optical character recognition (OCR) devices capable of reading printed text into a computer automatically.

When geometric transformations are unpredictable, not invertible or too complex, for example in recognizing handwritten characters, the methods described in the preceding paragraph are inadequate. In more complex problem domains, feature extraction can be very helpful. Unlike the case of general pattern recognition where feature extraction operates over the many dimensional space of the cartesian product of measurements, in pictorial pattern recognition, feature extraction usually operates over the two-dimensional picture space. That is, features in the latter case often represent simple geometric properties. Some correspond to local relations such as shape, orientation, curvature, contrast and number of lines at an intersection while others reflect more global relations such as topological connectivity, visual texture, repetitiveness, size or alignment of parts. Spatial frequency analysis including hologram analysis can be regarded as the extraction of global features related to geometric symmetries. Presence of such features corresponds to peaks in the frequency domain. Families of orthogonal square wave functions such as Walsh, Haar and Hadamard (e.g. Gerardin and Flament, BIBPIC 1969) functions are computationally much simpler for spatial analysis than the trigonometric functions used in Fourier analysis partly because inner products in the former case can reduce to boolean operations and also because their digital nature is more appropriate for grey-scale (digitized) pictures. However, the Fourier transform itself can be cheaply and rapidly effected using optical holography. A useful characteristic of spatial frequency analysis is that geometric distortions and their inverses in the picture domain often correspond to easily expressed changes in the transform domain, greatly simplifying the problem of picture registration in generalizations of template matching. Another useful characteristic is that the image degrading effects of high frequency noise resulting from digitization of poor optics can be effectively countered by low-pass filtering in the frequency domain. Also, spatial integration and derivative taking are expressed in simple algebraic terms in the transform domain. The disadvantages of spatial frequency analysis are computational expense and inflexibility in choice of features. The technique is useful only if certain kinds of global geometric symmetries are suitable for distinguishing pattern classes. Then, special purpose hardware can overcome the computational expense of general purpose softeware. Fingerprint classification and visual texture analysis are examples of tasks where this is true. One advantage in extracting features of whatever type is that pattern classification can often be accomplished by template matching in the 2-D transform space more easily than in the $n^2$ dimensional picture space.

Contrast features have been particularly useful in a wide variety of applciations, including scene analysis. These are local features corresponding to small, simple neighborhoods exhibiting strong variation in the gray levels of their constituent cells. In simple pictures composed of relatively large uniformly light areas differing from each other in gray level, such high contrast neighborhoods outline the regions. The picture can then be represented as a collection of chains of high contrast boundary curves. In a sense this corresponds to storing only the locations of non-zero gray-level gradients rather than all gray levels. Since gradients are zero in large uniform regions, far less information is needed to specify the picture than the gray-level representation, though the latter can be reconstructed by a computation analogous to integration. This economy is vital because the gray-level representation often requires more central computer memory than is available at one time.

Representing any curve, including a boundary, by encoding differences between coordinates of successive cells along the curve rather than listing locations is called chain encoding. This difference in coordinates corresponds roughly to the digitized derivative or slope and economizes storage for reasons analogous to those given in the preceding paragraph describing the gray-level to gradient transition. In addition, however, chain encoding yields digitized descriptions of geometric shape, independent of position, which can be used to compute such measures as area, curvature and perimeter as well as express the changes in patterns resulting from geometric transformations. Such digitial computations have been developed by Freeman (BIBPIC 1974), Weiman and Rothstein (BIBTR 1972), and Rothstein and Weiman (BIBPIC 1976). Applications range from image processing to computer graphics. The discrete expression of these geometric computations is often completely different from their expression in continuous mathematics. The processes of geometric abstraction outlined above play an important role in pattern recognition in part because of the resulting computational economies. One such economy is realized in approximating curves by straight line segments. Since the curve is completely determined by its endpoints, only they need be stored. In scene analysis, straight lines also play an important role because of the projective geometric relation between the 3-D scene and its 2-D image.

Dividing a picture into regions using any criteria, including high-contrast boundaries, leads to an abstract description in terms of region adjacency. That is, each region can be represented as a node in a graph and edges connect nodes representing regions sharing a common border. Besides being informationally economical, the picture is described in terms of topological relations rather than the exact positions and shapes of its parts. Such a description is invariant under a large group of geometric and other transformations. This approach has been generalized to what are called linguistic methods in pattern recognition (see Miller and Shaw, BIBPIC 1968 and Kaneff, BIBPIC 1970). In the linguistic approach the relational description of a picture is more important than simple classification. The relations may be much more general than topological connectivity but the description is still graphical. The graph can be modified by rules from a graph grammar to either parse or generate "legal" graphs, i.e. those corresponding to valid pictures. This is a "more intelligent" process than classification of pictures in the same sense that parsing a sentence (word) in a formal language is a "more intelligent" process than simply recognizing its constituent words (symbols). It has the important advantage over classification that a potentially infinite class of pictures, i.e. including pictures never seen before, can be recognized or described. This kind of relational analysis is an important charactertistic of many scene analysis methods. In our psychological model, it corresponds to comprehension of a whole as more than the sum of its perceived or recognized constituent parts.

The scene is comprehended as an organization, rather than merely a collection, of objects.

There are phenomena in the process of projecting 3-D scenes into their 2-D images which traditional 2-D pattern recognition techniques do not address. Paramount among these is foreshortening, a distortion in which the 2-D distance between image points depends on the orientation of the line connecting the corresponding 3-D points relative to the line of sight and also on their distance from the image plane. The extreme case of foreshortening occurs when the line connecting two 3-D points is a line of sight (goes through the point of projection). For opaque objects, the usual case, this yields occlusion, a discontinuity that abruptly erases part of an object's image that may contain important features for recognition. This phenomenon causes hidden surfaces and on 3-D rotation of an object in a scene yields a succession of 2-D views that cannot be continuously transformed into each other. Projection is a many-to-one function and therefore not invertible. This loss of information and the discontinuities in 2-D images of continuously transformed 3-D objects cannot be overcome by ordinary feature extraction, 2-D geometric transformations, or template matching.

The information loss in projection is structured in the sense that any two 3-D points that are mapped into one image point lie on a straight line through the point of projection, i.e. a line of sight. If there are two points of projection then, the intersection of two lines of sight uniquely determines the position of a point in three dimensions, a property useful in binocular stereoscopic vision. Geometrically related to this property is the fact that a line in three dimensional space together with the point of projection determine a plane in three space. This plane intersects the image plane in a straight line; hence 3-D straight lines have 2-D images in the picture plane that are also straight lines. This fact, together with the informational economy in representing straight lines discussed in the paragraph on chain-encoding earlier, make it natural that many of the earliest efforts of scene analysis were directed to analyzing scenes* containing straight lines. Such scenes contained polyhedra, 3-D objects bounded by planar faces which therefore intersect along straight line edges. The 2-D images of the faces are regions bounded by straight lines. In simple lighting situations a planar face is usually uniformly bright, but the different orientations of faces relative to the light source and the observer yield image regions of different gray levels. Thus, the images of edges may be found by using contrast features. We shall briefly examine some of the techniques used for finding these straight lines in gray-level pictures before discussing the analysis of polyhedral scenes.

Identifying straight lines in gray-level pictures was a much more difficult task than originally anticipated by researchers in scene analysis. It appeared that one need only trace along adjacent high-contrast features in pictures of polyhedra to get straight lines, but in reality such features were often spuriously present in noisy picture regions not near straight lines and difficult to detect in some places along faint straight lines. Though smoothing (or low pass filtering) reduces noise, it also suppresses contrast. Special contrast feature extractors can be designed, however, which paradoxically enhance contrast while suppressing noise; they combine spatial smoothing and the taking of spatial derivatives. Smoothing a function can be accomplished by taking the convolution of the function with a fixed smoothing function such as the gaussian function. In the discrete version of the process, the convolution integral reduces to a weighted average of values in a local neighborhood. The result in either version is "blurring" or smoothing. Now, regions of high contrast are characterized by extreme values of derivatives. In a picture, derivatives are defined as the limits of ratios of intensity to distance between points at which the intensities are observed.

*From now on, the word "scene" refers to the 3-D configuration whose image is projected onto a 2-D picture.

In the discrete version the limit cannot proceed below grid cell size, which if taken to equal unity reduces the derivative to a kind of weighted finite difference. Smoothing and derivative taking in the continuous case can be conceptually unified by considering their representations in Laplace transform theory. There, convolution in transform space corresponds to multiplication of the pattern transform by the transform of the smoothing function. The transform of the derivative of a pattern is simply a constant times the transform of the pattern. Since products commute, the same result is effected by taking the derivative of a smoothed pattern as by taking the convolution of the pattern with the derivative of the smoothing function. In discrete form, the latter coresponds to taking a weighted average of the picture points in a neighborhood where the weights may be negative as well as positive. Thus, smoothing and derivative taking can be accomplished in one step no more complex computationally than smoothing. Smoothing is analogous to a 0th order difference, and various higher order derivatives are analogous to higher order differences. In the 2-D case partial derivatives correspond to finite differences or weighting functions over neighborhoods rather than intervals. The choice of diameters of these neighborhoods is critical in effective feature extraction. Ordinarily "noise" grain is much finer than "signal" (or true picture) grain. If the diameter of averaging neighbhoods falls between these two grain sizes, both noise suppression and contrast enhancement can occur simultaneously resulting in a better signal-to-noise ratio.

Almost all contrast feature detectors can be considered as weighted averagers as described in the preceding paragraph. Examples are Shirai's contrast detector (BIBSA 1973) which resembles a smoothing and first order derivative taken in a direction perpendicular to a line. The Laplacian operator is a second order partial derivative; Horn (BIBTR 1972 and BIBPIC 1974) discusses its discrete analog. Hueckel's (BIBPIC 1971, 1973) operators are combinations of finite differences of various orders; for that reason they are particularly versatile in detecting and describing a broad range of types of contrast; however they are somewhat more expensive computationally than simpler differencing schemes.

Smoothing and taking spatial derivatives of a picture of a polyhedral scene yields distinctive ridges and valleys along the images of polyhedral edges and yields flat regions where high frequency noise or uniform intensity prevail. The TRACK program package developed at MIT and described by Lerman and Woodham (BIBTR 1973) reduces such a picture to a line drawing. It uses a Shirai type contrast detector to locate points of high contrast; with these it associates more distant points lying close to the best straight line fitting the original points, progressing from point to point. A straight line is thus extended in the direction best fitting (according to some mean square error criterion) those points already assigned to it. When no more feature points are found in that direction, the line is terminated. Thresholds can be set to "tune" the program to various contrast levels and acceptable line lengths. In the end, the data structure that TRACK presents to scene analysis programs consists of lines specified by their endpoints. Neither gray-levels nor features remain.

Using statistical mathematical models for noise, edge image blurring, and light intensity variations over polyhedral surfaces Griffith (BIBPIC 1971, 1973) derived theoretically the acceptance criteria for straight line determination that are found experimentally (by "tuning" thresholds) in the TRACK program package. Duda and Hart (BIBPIC 1972) use a rather different method for recognizing straight lines in noisy situations which was applied by O'Gorman and Clowes (BIBPIC 1973). It involves first finding high-contrast neighborhoods and then scanning these in two perpendicular directions with feature extractors resembling first order derivatives. These two "directional derivatives" are components of the local light intensity gradient. The direction of any local edge is perpendicular to this gradient, and can be considered as a small segment of a straight line. The so-called Hough Transform is applied to each small line segment (local edge) to represent it as a point in a space whose two dimensions correspond to the angle the local edge made with the x-axis in the picture space and the distance of the line on which it might lie from the origin. Any extended line in picture space will yield a small dense cluster of such points in Hough Transform space. The degree of clustering is therefore used as a criterion in deciding whether a line exists in the picture; its endpoints can be found by locating extreme values of the coordinates of the neighborhoods corresponding to the edges in question. A crucial difference between this method and tracking methods is that orientation of a feature and its existence anywhere on the line locus are used as evidence rather than feature continuity with respect to tracking order. The result is that a line can be recognized even if interrupted by many large gaps or crossed by other curves; in such situations tracking could fail. The end result of the Duda-Hart process is also a data structure representing straight line segments whose end points are specified.

If, in our psychological analogy, light intensity measurement corresponds to sensation, then pattern recognition, including straight lire recognition, could be called perception. The next level up in the hierarchy is then comprehension. In artificial vision we shall here consider comprehension as understanding something about the 3-D relations between parts whose 2-D images have been perceived. The goal of scene analysis is that kind of comprehension. We will now examine some of the first approaches to recognizing scenes consisting of polyhedra based on the line drawings of their images.

## 2.3 Approaches to Scene Analysis

### 2.3.1 Analysis of Straight Line Representations of Polyhedra

In the history of scene analysis Roberts (BIBSA and BIBTR 1965) is credited with taking the first step in computer interpretation of a 2-D picture as a monocular view of a 3-D scene. His method was an extension of known picture processing and pattern recognition techniques. First, the grey-scale picture is reduced to a line drawing by fitting lines to contrast features. Racognition is accomplished by geometric transformation and generalized template matching, but in the 3-D realm rather than the 2-D picture space. Picture regions are considered to be the images of polygonal faces of polyhedra. The polyhedra in the 3-D scene are restricted to be examples of certain known models, hence the 2-D images of faces can only be of certain known types, subjected to projective distortions. The final image is a product of rigid motions of the polyhedra in three dimensions and the projective transformation. In cartesian coordinates the former are linear and the latter nonlinear transformations. Roberts overcame the computational difficulties in expressing this mixture by using homogeneous coordinates which permit a unified linear representation of all relevant transformations. This notation originated in projective geometry (see Coxeter BIBGEN 1964) where it greatly simplified and unified description of phenomena difficult or impossible to describe in Euclidean geometry. The next step in Roberts' recognition process is to invert the transformations which yielded the image to identify the visible faces of the 3-D polyhedra. Recognition consists in matching the resulting 3-D structure with a known model, a generalization of template matching.

Roberts' method cannot correctly label picture regions as faces of blocks in a 3-D scene when that scene contains unknown types of polyhedra or known types which partially occlude each other. Guzman (BIBTR and BIBSA 1968) tried to overcome these weaknesses as well as avoid the computationally expensive numerical approach by using more linguistic, relational methods. Given the line drawing of a scene, Guzman's SEE program does not use the exact coordinates of points and lines as does Roberts' program but rather the geometric relations between them. SEE concentrates on points at which several picture lines intersect. These junctions are the images of polyhedral vertices, the meeting points of several faces. The acuteness or obtuseness of the angles between these junction lines depends on the viewer's position relative to the planes of the polyhedron. For example, when looking at a corner of a cubical building from the street, the two visible roof edges and the intersection of the walls at the corner form an image junction which resembles an upward pointing arrow. After a vertical helicopter ride to a position above the plane of the roof, the junction resembles a two pronged fork; the angle between the shaft of the arrow and its barbs has changed from acute to obtuse and the roof surface has become visible. These and several other junction configurations are recognized by Guzman's program. From observing cases, heuristics are derived for inferring whether or not regions on either side of a junction line should be considered as images of adjacent faces of a polyhedron. An abstract model of the scene is constructed by representing each region as a node in a graph and face adjacency is denoted by linking appropriate nodes. The phase just described yields a representation based on information local to junctions. A more globally plausible structure is derived in the next phase by examining the graph and ruling out certain unlikely configurations such as incorrect connections between distinct polyhedra.

The graphical representation of distinct bodies by Guzman has a linguistic flavor and is a higher level abstraction than Roberts' representation of examples of models. This permits analysis of much more complex scenes with many objects partially occluding each other and containing objects never seen before. One key to the power of this approach is that junction properties exploited by the region-linking heuristics are invariant under a large class of 2-D and projective transformations. Therefore, transformation inversion which Roberts needed, is never necessary. Junctions are informationally rich because they capture at a single point the relations between several extended parts (the faces) of polyhedra.

Guzman's program was a first step in junction analysis, and it had several unforeseen weaknesses. After a more careful analysis of scenes and their images, Rattner (BIBTR 1970) embodied an improved set of heuristics in his program SEEMORE. One of the weaknesses of Guzman's program was that in scenes with many objects close to each other, shadows are thrown across faces and their edges can be misinterpreted as the images of polyhedral edges. Orban (BIBTR 1970) studied these configurations and developed heuristics for detecting and removing such lines from junctions.

Waltz (BIBTR 1972) attacked the same scene analysis domain as Guzman, Orban, and Rattner but instead of using a set of heuristics based on empirical observations he analyzed geometric relations between 3-D vertices and the junctions that are their projective images. His fundamental object of analysis was also the junction type, but instead of considering only shape, he examined the 3-D configurations that could give rise to such shapes. He then labelled junction lines with descriptors of the 3-D properties they could correspond to. These labels were originated independently by both Huffman (BIBSA 1971) and Clowes (BIBSA 1971) to distinguish between 3-D edges which are concave, convex, occluding bordering surfaces or cracks between objects. Huffman used these four labels to determine whether or not line drawings could be interpreted as pictures of "real" objects. This involves attaching labels to lines and progressing to other lines through junctions until all lines are labelled. The existence of one kind of label at a junction constrains the label of other lines. Any line can only be assigned a single label. If it is impossible to label a picture consistent with these constraints, it could not be the picture of a real object. An example is the "devil's pitchfork", a picture which locally looks "real" but is inconsistent globally. Waltz exhaustively examined all possible views of vertices where three (and in a few cases, more) polyhedral faces met and labelled the lines accordingly. For any particular junction shape, a large number of possible labellings, each corresponding to a distinct 3-D configuration, is possible.

Waltz' program associates with each junction in a picture a list of all its possible labellings. Those are ruled out for which no adjacent junction has a labelling compatible along the line they share. Waltz originally thought a tree search would be necessary to examine all consistent labellings, with the accumulation of interacting constraints leading to a single or few possible complete picture labellings. To his surprise, most junction labellings were ruled out in simply progressing from junction to junction. The reason was that the apparent increase in descriptive complexity is more than offset by the fact that the number of geometrically realizable labellings of any particular junction shape is far smaller than the number of combinations possible if the lines could be independently labelled. Thus compatibility between adjacent junctions is much less likely than if lines were labelled independently. Waltz next introduced shadow labellings for lines; exactly the same phenomenon occurred resulting in even better scene analysis with no tree searching necessary. Shadows, instead of interfering with scene analysis as in previous approaches, actually contribute information. Newborn (BIBTR 1974) embodied Waltz's picture labelling algorithms in New York University's high-level set theoretic language SETL.

The descriptive labels on the lines of an analyzed picture can easily be used to derive object identity. Waltz goes beyond Guzman, however, in that even more geometric information about the 3-D scene is found. Crude shape descriptors for edges tell something about the relative positions of planes in the 3-D scene. A drawback is that instead of a small number of junction labels, a large dictionary, with thousands of entries, must be stored or computed.

Waltz's improvement over Guzman is a result of generalizing Guzman's heuristics which were seen as special cases of the geometric semantics relating 3-D vertices and their 2-D images (junctions). Mackworth's program POLY (BIBSA 1973) is an extension in this tradition. His approach is based on the following geometric representation. Any 2-D picture of a 3-D scene consists of the projection of the latter's visible points through the image plane to a single special point called the viewpoint. In a camera, for example, the photographic film occupies the image plane and the viewpoint lies in the lens. The 2-D image of a line in the 3-D scene is also a line. This image line and the viewpoint determine a plane which must also contain the line in the 3-D scene which gave rise to the image. This is called the "plane of interpretation" by Mackworth; the polygonal image of a polyhedral scene yields a bundle of planes of interpretation all of which contain the viewpoint. Now, if the scene consists of polyhedral objects, their faces intersect along straight line edges whose images determine the planes of interpretation. The relations between the orientations of these sets of planes can be exploited to infer the former from the latter. The cumbersome scene representation of planes as sets of points is avoided by representing each plane as a single point in dual space. In brief, any plane in three space can be characterized by the direction of its normal and its distance from the origin; thus in dual space each plane is represented by a point at the tip of a vector starting at the origin pointing in the direction of the normal and having magnitude equal to the distance of the plane from the origin. Relations between planes have the following representations in dual space. A set of parallel planes corresponds to a set of points lying on the same line through the origin. The intersection of a number of planes in a single point (polyhedral vertex) corresponds to the vertices of a planar polygon in dual space. All planes of interpretation are represented as points in dual space lying in a single plane; that plane is the dual of the viewpoint. The edges bounding a single face of a polyhedron correspond in dual space to lines passing through a single point, the dual of the planar face.

Mackworth's program POLY operates on the dual space representation of scenes and their images to determine edge types (such as Huffman and Clowes types), surface orientation, body identification, and hidden structure. It tries to link neighboring image regions by labelling the lines separating them as the images of "connect" edges in the scene, i.e. intersections of the planes whose images are [illegible] regions. In dual space, identifying a connect edge between two planes corresponds to drawing the dual of the edge (a line) through the duals of the planes (two points). Progressing from region to region in the image corresponds to connecting points with straight lines in dual space. Constraints imposed by incidences in the image have their counterpart constraints in dual space, preventing many edges from being interpreted as connect edges. Since the background ordinarily consists of a single plane, the program begins by attempting to connect it to all regions bordering it. There, the large number of simultaneous constraints greatly reduces the amount of backtracking necessary in searching for correct picture interpretations. The eventual aim is a picture in which every edge is labelled as connecting or occluding two neighboring regions. Polyhedral objects in the scene correspond to polyhedra in dual space; completing missing portions in the latter corresponds to solving hidden line and surface problems in the former. Pictures for which this is impossible are generalizations of Huffman's illegal or "nonsense" sentences, for example, the "devil's pitchfork" mentioned in the discussion of Waltz' approach.

Mackworth's purely geometric, projective approach appears superficially to be a step backward from the abstract, symbolic approaches of Guzman and Waltz to the numerical methods of Roberts. That this is not so stems from Mackwroth's dual space representation in which points correspond not to points in the scene but to sets of points (planes) bearing some relation to each other. Roberts' approach was based essentially on transformation and template matching, though with a novel twist; in Mackworth's approach object identification results from interpreting the dual graph in terms of connectivity rather than shape and position of picture parts. This extraction of high-level relations is in the spirit of Guzman and Waltz and is one property which characterizes artificial intelligence approaches as opposed to pattern recognition approaches. In addition, however, the dual space representation can yield geometric information about the scene inaccessible to Guzman and Waltz. Distances and angles in dual space correspond to quantities describing geometric relations between entities in the scene. Among these are orientations of faces and edges; Waltz's classification of these was restricted to binary distinctions such as convex vs. concave. Waltz's dictionary of junction types is a list of special cases of phenomena Mackworth can represent in their entirety. Waltz's surprise at the small number of realizable parsings of sentences over words in this dictionary would have vanished if he had known about the general constraints of positioning points and lines in dual space. Mackworth's dual space representation is capable of yielding an infinite dictionary,including all the difficult cases of polyhedral vertices formed by more than three coincident planes. Though he did not include it, there is room for representing shadows as projections of edges through the light source onto a surface in the scene and then through the viewpoint to the image plane. This new projection point (the light source) ought to add many of the scene analysis capabilities of binocular visual systems; the dual space equivalent of this point is a plane which would further constrain relations between dual space points.

The polyhedral scene analysis programs just described (from Roberts through Mackworth) are characterized as "bottom up". That is, before scene analysis proper begins, small regions of high local contrast (features) are found and linked together by least squares or similar statistical methods into straight line segments. These line segments are the primitives on which the scene analysis programs operate. A serious drawback of this approach is that feature detection is very sensitive to threshold and appropriate thresholds may differ in different parts of the picture depending on the high-level (semantic, 3-D geometric) structure of the scene. This high-level information cannot be known in advance of the low-level or line finding phase of the operation. In using a uniform threshold throughout the picture, high sensitivity yields spurious feature detection where noise is present. Noise is inevitably high due to camera imperfections, bad lighting, uneven reflective properties of objects in the scene and digitization truncations (HORN, BIBTR 1969). Lowering sensitivity of feature finders to avoid false positive detections results in failure to detect faint lines between regions which are the images of nearly equally bright adjacent faces of a polyhedron.

Falk's (BIBSA 1971,1972 and BIBTR 1970) approach to solving the problems described in the preceding paragraph was to append a "top-down" phase to the bottom-up phase. Spurious lines are removed and missing lines are added after the initial bottom-up phase by considering the initial line drawing as the projected image of a scene consisting of a collection of polyhedra of known types. Inverting the scene-to-image projection in the manner of Roberts and solving hidden line and surface problems leads to a best model of the scene in terms of the known polyhedral types. Missing or extra lines resulting from the bottom-up phase are added or deleted respectively to make the final line drawing conform to the model of the scene. Thus, it differs from Roberts' approach in that a Guzman-like bottom-up phase first predicts the scene objects and only after that relatively cheap computation (list processing rather than matrix operations) does the projective analysis occur. The advantage is not only greater simplicity, but the Guzman-like phase of the operation is far superior in analyzing scenes with large numbers of objects which occlude portions of each other. One drawback of this method is that the scene is limited to collections of objects from a fixed set. The hidden line and surface algorithms are also computationally expensive.

Shirai (BIBTR 1972 and BIBSA 1973) overcame many of the drawbacks of the bottom-up approach using a method that looks deceptively low-level, but succeeds as a result of efficient use of heuristics based on high-level properties of polyhedral scenes. His overall strategy is to proceed from the most distinct (highest contrast) edges to the least, searching for lines only in places suggested by earlier, stronger evidence. Lines are classified into three categories. Those in the first, contour lines, are characterized at the low (picture processing) level as consisting of picture points whose grey levels contrast greatly with local neighbors. At the high (3-D scene property) level, contour lines correspond to the separation between, for example, a light object seen against a dark background. The second category consists of boundary lines which are characterized at the low level by slightly smaller contrast values; at the high level they correspond to the separation between an object and another object, or between an object and the background. Contour lines are special cases of boundary lines. The third category, internal lines, contains those exhibiting least contrast; they correspond to lines separating adjacent faces of a body.

Shirai's process is begun by sampling a grey scale picture once in every 8 × 8 cell region. Samples whose grey levels differ greatly from the picture average are singled out as contour points; full resolution is restored and the same criterion is applied to find contour points missed by the original sampling in the 8 × 8 neighborhoods of those found by the original sampling. The chains of contour points resulting from this process are traced and broken into straight lines whose endpoints are local maxima of hain curvature. These constitute a set of contour lines, strong candidates for delineating the outlines of objects in the image of a scene because an object and its background usually differ more from each other in optical properties and position relative to the light source than adjacent faces of the same object or nearby faces of neighboring objects of a similar type. This is an example of the semantics of projective geometry relating 3-D objects and their 2-D images embodied in Shirai's heuristics. Most of the other nine heuristics involve attempts to extend lines from their endpoints or find lines going in new directions starting at the endpoints of other lines. The semantic justification for these heuristics is obvious from looking at the junction types of Guzman or Waltz. That is, lines in a picture are images of polyhedral edges. Edges terminate in vertices with other edges. Therefore their images are junctions of lines, the logical place to look for extensions of old lines or beginnings of new ones. Since an occluded neighboring object often contrasts less with its occluder than the occluder with the background, the extension of a contour line along the top of a TEE junction is usually a boundary line. This is embodied in heuristics by increasing the sensitivity of the line finder when searching for the extension. Similarly motivated heuristics are used to find all boundary lines. Finally, the sensitivity is increased even more to find internal lines at likely places such as the stems of arrows.

The final product of Shirai's system is a line drawing of a scene, with lines labelled as contours, boundaries, and internal lines. Individual objects (polyhedra) are identified by tracing contour lines and their extensions into boundary lines; this usually yields a simple closed path that is the outline of an object. The selective use of increased line finder sensitivity in places determined by earlier stronger evidence results not only in considerable computational economy but also in the avoidance of false positive line detection in noisy regions. There are two underlying principles behind Shirai's approach. The first is careful exploitation of the projective semantics which relate straight line edges in a scene to their straight line images. The other is the constant interplay between high level knowledge of scene properties and low level line following. This interplay is an example of heterarchical rather than hierarchical control. The term heterarchy was used by McCulloch (BIBGEN 1945) to describe neural networks in which feedback destroys the usual distinction between high and low level control. In analogy with linear systems theory, feedback often gives systems greater dynamic range, flexibility, and computation power than their hierarchical counterparts. Minsky (BIBTR 1970) suggested that heterarchical control might be valuable in AI systems. This contention is strongly borne out by Shirai's system which not only often yields far better line drawings with less computation than conventional bottom-up approaches but also achieves rather good high level separation of objects and identification of edge types. Remarkably, this higher level "semantic" type of information is acquired without the use of tree searches, dictionaries of junction types, or projective transformations characteristic of other approaches. These omissions prevent Shirai's system from separating neatly stacked bodies and detecting certain kinds of concavity, i.e. situations which are not pointed to by the images of convex edges; the lack of 3-D semantics is a limitation. However, Shirai's program could be used as a superior (to conventional line-finders) preprocessor for bottom-up programs, though such service violates the spirit of heterarchy!

This completes discussion of polyhedral scene analysis based on straight lines and their junctions. Combinations of this approach with semantic methods embodying knowledge of physical structure and purpose will be discussed after describing other methods which can also be so combined. The first of these is region analysis.

### 2.3.2 Region Analysis

Minsky and Papert (BIBGEN 1967) suggested avoiding the difficulties of line finding by constructing the regions that might be bounded by such lines directly, joining neighboring cells which have similar light intensities. Brice and Fenema (BIBSA 1970) incorporated this suggestion in a scene analysis system. The first step is to scan a picture, introducing a short boundary between any two neighboring cells whose intensity values differ by more than some threshold. The magnitude of the difference in intensities is stored in association with each such boundary. In terms of our earlier discussion of contrast features, this process yields first order differences, analogous to first order partial derivatives with respect to the coordinate axes. The resulting boundaries resemble the edge-elements of O'Gorman and Clowes, yielding a picture with many short-perimeter regions bounded by these edge elements. Brice and Fenema's departure from previous contrast detecting methods at this point consists in merging these small regions according to heuristics based on qualities of regions rather than line-fitting. The first of these, the phagocyte heuristic, merges adjacent regions, erasing their common boundary if the contrast along it is low and the ratio of its length to the perimeter of the shorter region is large. The result is a less choppy picture with many of the former small enclosed or almost enclosed regions being merged with larger ones. Next, a weakness heuristic is applied, merging regions if the ratio of the length of low contrast common boundary to length of common boundary is high. This differs from the first in operating without regard to perimeter length. Thus the phagocyte heuristic cleans up small enclosed islands which are unlikely to be real region outlines and the weakness heuristic reduces the tendency to see spurious large region outlines where intensity gradients are present but weak.

The next step in Brice and Fenema's scheme is to find vertices (places where three regions meet) and join adjacent ones along their connecting region-boundary with straight line masks. The latter are thin rectangles anchored at vertices; their widths represent limits of acceptable deviation of boundary points from a straight line joining vertices. After successful fitting, each mask is replaced by a line and the picture is represented as a line drawing. Guzman-like techniques are applied to propose objects, which are assumed to be wedges, cubes, wall or floor. The last two constituents are also sought on the basis of their vertical location in the picture. Line drawings are corrected in a manner similar to Falk's; missing or extra lines are added or deleted respectively on the basis of models of objects. This lack of object generality, is not the fault of region-growing but crudeness of higher-level processing. This is one of the first attempts at region-growing and predates the work of Waltz and Mackworth, who had more sophisticated high-level representations.

In deriving the line drawings of polyhedral scenes, region growing and line-finding should yield the same result. A picture specified by a line drawing can just as well be specified by the regions those lines enclose. However, region-growing ought to be more robust than line-finding because the heuristics which merge regions account for global properties of intensity distributions throughout a region. This is usually a much larger area than the small neighborhoods used in line-finding so local high spatial frequency noise is much less disruptive.

The usefulness of the heuristics in the region-growing method just described is predicated on the existence of rather large regions with simple boundaries, not necessarily straight lines. Line fitting was the result of a later pass designed specifically for polyhedral scenes. An attractive feature of region growing is that it may be applied to more natural scenes whose images contain no straight lines. Regions may correspond to irregularly shaped areas with characteristics distinct from the surround. These characteristics need not be restricted to light intensity but may include texture, color , or binocular disparity between two pictures. The characteristic used need not have constant value throughout the region but simply be free from sharp discontinuities. Systems embodying general region analysis will be described in the following paragraphs.

Krakauer (BIBTR 1971) embodied a unique kind of region analysis in a program to distinguish between various fruits such as apples and pears. This task is representative of those in which humans easily recognize objects though examples within a class may differ in ways difficult to express explicitly. Recognition cues seem to include gross 2-D image shape, 3-D shape and reflective properties inferred from intensity gradients and roughness or texture. All of these are incorporated in a simple tree structure derived from an intensity contour map. The tree's structure can be visualized by considering an intensity contour map as a set of stacks of planar regions of uniform thickness, each level corresponding to picture points with higher light intensities than the threshold associated with that level. As examples, a uniformly bright disk of light corresponds to a stack of poker chips and a disk with progressive darkening away from its center corresponds to a conical stack. Each distinct connected region (and there may be many in a complex picture) at any particular level corresponds to a distinct node in the tree at that level. If one region supports a region above (that is if the boundary of the former encloses the boundary of the latter) a path connects the corresponding nodes. Thus, the tree structure describes set-nesting relations of contour curves. A picture of a "speckled" object corresponds to a "bumpy" contour map with many disconnected regions; this in turn corresponds to a tree with many branches. Krakauer distinguishes between types of such texture by plotting number of branches against tree level. This profile has characteristics immune to geometric transformation of the object being recognized, changes in illumination, and irrelevant texture details. In addition to the structure of the tree, each node can carry information about the size or shape of the corresponding region. For example, pears can be distinguished from apples by the eccentricity or elongation of regions at most levels. The advantage of this kind of measure is that statistics can be gathered on regions at fixed levels easily even if their boundaries are not "clean". That is, ragged edges, holes, and sharp turns need not be eliminated by region growing heuristics in order to measure useful properties of regions.

Barrow and Popplestone (BIBSA 1971) used region analysis in a somewhat more general manner than Krakauer to recognize pictures of household objects including a teacup, hammer, wedge, spectacles, pencil and the like. Regions are found using a merging algorithm in which all cells in any particular region must have gray-levels within a narrow range. When ranges overlap, regions may overlap. Instead of defining a tree using set-containment and intensity thresholds, Barrow and Popplestone define a graph which expresses more general relations between regions. Each region corresponds to a node and links between nodes correspond to relations between regions such as relative position (above, beside), relative size (bigger), distance, and shape of adjoining boundary. In addition, each node carries descriptive information about the corresponding region such as shape and brightness. These cues were subjectively chosen to correspond to those which might distinguish common objects in cartoon-like drawings; such drawings resemble the result of region growing programs operating on gray-level pictures.

Recognition in the Barrow and Popplestone approach consists in matching the relational graph of a picture with that of the model it resembles. This involves matching not only graph shape but also the attributes attached to nodes and links. The model comes from training sessions in which samples of known objects are processed to gather statistics on the resulting graphs. The system performed 85% correct recognition in distinguishing nine household objects, using an average of five minutes of computer time per picture on an ICL 4130.

Region analysis and other approaches can be combined with semantic methods to improve scene analysis. These topics will be discussed immediately after the next section which deals with numerical approaches exploiting geometric semantics.

### 2.3.3 Numerical Region Analysis Methods Using Geometric Semantics

Though the region analysis methods just described allow greater flexibility in scene types and problem solving tools than line analysis methods, the loss of projective semantics relating scenes to their images is a serious drawback. For example, a shadow thrown across a region leads to its erroneous separation into distinct regions in the graphical description of the picture. Occlusion of objects by others has similar undesirable results. For this reason the methods of Krakauer, and Barrow and Popplestone, fail to correctly identify objects when applied to scenes with many objects. The ad-hoc heuristics on which recognition is based are not easily extended to account for such projective phenomena. An approach by Horn (BIBTR 1970) analyzes regions but in a strictly numerical, geometric way, accounting precisely for the projective relations between object and image. The goal is to infer 3-D shape of smooth objects from gentle (as opposed to edges) gradations in light intensity (shading) in their images. The underlying assumption is that the object's surface is optically uniform. That is, in approximating any small area of a surface by its tangent plane, the relative light intensity of its image is a function of the relative directions of its normal and lines to the light source and observer. Different models of surface optical properties lead to different functions. To invert such a function and infer the 3-D surface shape requires solving a first-order nonlinear partial differential equation which can be reduced to a system of five ordinary differential equations. Horn's rigorous, mathematical approach not only enables him to incorporate heuristics to tune the parameters of numerical methods to solve the system of equations efficiently, but also leads to interesting observations on the significance of shading in human vision. The roles of cosmetic makeup and lighting techniques in photography are discussed in terms of the mathematical model. Horn's approach is strictly mathematical; it doesn't produce the abstract, relational data structures usually associated with AI. Just as in Mackworth's approach, however, such data structures can be derived from the inferred 3-D shape. For example, the visual boundaries of any smooth convex object are characterized by surface normals perpendicular to the line of sight; thus the normals could be used to identify individual objects. Contour maps of 3-D depth of objects should provide a more reliable base for analyzing relations between picture regions than simple intensity contour maps because the former are invariant under 3-D transformations with suitable change in viewing position alone, while the latter are not. Shadows cast by objects onto others are used to infer shapes and relative positions; they do not interfere as in the other region analysis methods.

High contrast loci (edges or region boundaries in earlier methods discussed earlier) are the terminators of Horn's solution paths rather than being the objects of attention as in line-finding methods. In Ramer's approach (BIBTR 1973 and BIBPIC 1975) they play the latter role, but in scenes which can be as general as Horn's. This suggests fruitful combination of the two methods for identifying (isolating) distinct objects in real world scenes and inferring their 3-D shapes. The capabilities of the two methods are complementary. Ramer uses Hueckel's operator (BIBPIC 1971,1973) to detect the positions, orientations, and strengths of high-contrast edges in a picture. He uses Freeman's chain-encoding to approximate their orientations and links short chains into longer ones using heuristics based on statistical confidence criteria such as strength, length of chain, local signal to noise ratio, and directions of adjacent neighbors. Long chains resulting from this process are classified as shadow edges, cracks, texture, boundaries of bright specular reflection (highlights) and object boundaries on the basis of constituent edge properties and relations to other chains. For example, shadow edges are generally less sharp than boundaries of objects in a scene; the chains corresponding to the former are therefore characterized by lower contrast features and a broader range of local variation in direction of constituent edges. Resulting chains are fitted to quadratic curves; this is the simplest computational extension of linear interpolation. The increase in computational complexity yields more general boundary curve detection. Quadratic curves are particularly important because they are the images of sections of quadric surfaces (spheres, cylinders, ellipsoids, cones, etc.) which bound objects in a more general class than polyhedra. Methods for fitting quadric surfaces to objects are described by Agin (BIBTR 1972) and Agin and Binford (BIBSA 1973). In a sense, Ramer's approach does for this class of objects what Shirai's method did for polyhedra; namely, it yields a classification of curved edges which can be used to determine 3-D structure of objects viewed. The tools used in these two cases are quite different, however.

Another purely numerical method exploiting the projective semantics relating a 3-D scene to its 2-D image is used by Hannah (BIBTR 1974). While Horn's method may be characterized as region-growing based on extending solution paths of partial differential equations, Hannah's method consists of region growing based on intensity profile correlations between two pictures of the same scene taken from slightly different positions.

This process is related to binocular depth perception. In the first phase, a small region (window) with distinctive statistical characteristics (e.g. large gray-level variance) is found in one picture. Looking only at parts of the other picture with the same characteristics, an attempt is made to correlate the original window with a window in the other picture. When this succeeds, the two windows are considered as being in registration with each other, i.e. they correspond to two views to the same part of the scene. A region is grown around this window by moving the windows in both pictures the same distance and direction. When an abrupt change in 3-D depth occurs, such spreading will fail because of binocular disparity, a shift in the images relative to their original registration positions. Thus each region eventually found corresponds to a surface in space bounded by abrupt depth changes. Correlation and search for regions in registration with each other is extremely expensive computationally. Hannah greatly reduces the number of computation steps necessary by exploiting statistical properties of the picture, using fast algorithms such as the Fast Fourier Transform, and restricting the search to areas deemed likely by scene and image geometry. The latter involves employment of a Camera Model, using techniques developed by Sobel (BIBTR 1970 and BIBSA 1973,1974). This is a mathematical model which relates camera positions to image properties. For example, an image point in one picture corresponds to a point in three space which could be located anywhere on a line of sight in three space. The image of this line of sight in a second picture is a line rather than a point if the camera is not in the same position in both cases. In searching the second picture for the corresponding image of a point in the first then, one need only look along this line even if the depth of the point in the scene is completely unknown. Guesses about its depth correspond to restricting the search to segments of this line. Thus, the fact that straight lines are preserved as a class under projective geometric transformations can be exploited to great advantage even when there are no straight lines in the 3-D scene or its images.

A property that makes Hannah's approach more powerful than any discussed so far is that her region analysis is not impaired by camouflage, shading, texture, sharp shadows or intensity contrast edges because it is not based on intensity levels and their local variations but on correlations between arbitrarily structured intensity distributions in two pictures. She has exploited a binocular geometric semantics which appears to be much more powerful than the monocular geometric semantics of other methods. We will now examine some other kinds of semantics used in scene analysis.

### 2.3.4 Incorporation of General Semantics and Mechanical Aids

Feldman and Yakimovsky (see BIBTR and BIBSA, both authors, either order and several titles relating to Semantic Region Growing) improved the results of region growing by using merging rules based on semantic or (real world) phenomenon in the scene. The first step is to conservatively apply non-semantic region growing rules with merging based on similarity of color as well as light intensity. The result is a large number of small regions. Next, semantic meanings, depending on the problem domain, are attached to regions. For example, in a typical view through the windshield of a car, blue regions at the top of the picture are considered sky; they are merged with each other and while (cloud) regions which may intervene. At the bottom of a picture, green, yellow and brown regions are interpreted as grass and merged together. Regions in the middle of the road with distinctive shapes are identified as cars; in this case merging proceeds as long as the geometric shape of the regions better approximates that of known cars. Similar criteria are evaluated in merging regions in other parts of the picture. The criteria are based on training sessions in which the trainer evaluates results of certain merges. In the actual recognition process, a Bayesian strategy is followed to maximize the probability of making the best choice. The end result is a partitioning of the picture into regions, each of which is labelled according to its meaning in the real world. Regions are identified much more realistically than in methods where semantics are not employed in the region growing rules.

In the VISIONS system, Hanson and Riseman (BIBTR 1974) incorporate a heterarchical combination of high-level (semantic) and low-level (geometric feature extraction) methods using parallel computation. At the semantic level, objects which might be in a scene are modelled by abstract relational structures similar to those of Barrow and Popplestone or Yakimovsky. Low and high-level processes interact in the following manner. First, low-level processors detect features which suggest a subset of objects which might be present in a scene. The corresponding semantic models are selected; they direct low-level processors to seek features which would confirm or deny model validity. In the case of denial, new features found could be used to suggest other models, and so forth. The low-level feature detection capabilities are very flexible and include detection of local color, texture, brightness, and contrast. The selection of paticular features at the direction of high level models greatly improves efficiency by restricting the search to relevant information.

Winston (BIBTR 1970) developed a system which learns to recognize complex high-level structures constructed of blocks (polyhedra). Feature extraction and line detection are low-level processing whereas identifying polyhedra, as in the programs of Guzman and those who followed, is considered high-level processing. An even higher level description of a scene is one in which the blocks form structures such as arches, tables and towers. Such structures are represented in Winston's system by graph data structures whose major nodes correspond to substructures and links between nodes are relations such as is-supported-by or is-a-part-of. Additional nodes correspond to blocks and are linked to other nodes describing their attributes (posture, type of polyhedron). Different kinds of structures have different kinds of graphs, but many times one type of structure can have variations yielding different graphs. The goal is to train the system to distinguish which of these variations are crucial in order to correctly identify structures consisting of arrangements of blocks. It is accomplished by presenting the learning program with "near miss" examples, those for which small changes transform one kind of structure into another. Thus instead of using a statistical training approach which would blur critical distinctions, the latter are singled out for special attention.

Winston's approach is closedly related to Minsky's idea of frames (BIBTR 1974), a relational data structure for representing certain global situations or scenes. The frame is subject to transformations which are considered to be minor perturbations of a single global gestalt. The nature of the transformations is complex, however, involving alterations of the list structure that may be interdependent and difficult to anticipate. The semantic relations transcend those of the real world in AI. This very interesting and difficult topic lies at the heart of much current AI research but is beyond the intended scope of this survey; for further discussion the reader is referred to the general AI texts in BIBGEN, to Minsky, Papert and Winston (BIBTR, all references to any of the names) and in particular to Winston (BIBSA 1975).

The aim of scene analysis is machine comprehension of real world scenes; hopefully under natural lighting conditions. Sometimes, however, an immediately useful application is desired, for example in robot assembly of parts in a factory, and one must fall back on less sophisticated methods for determining 3-D configurations. These include touch sensors attached to probes to mechanically measure distance, sonar, and artificial lighting situations. The latter exploit highly structured lighting which is usually used to infer distance by triangulation between the known position and direction of the light source, position and direction of the camera, and the image of the light on an object. Agin and Binford (BIBSA 1973) use a laser in such a system at Stanford University. Shirai and Suwa (BIBSA 1971) use a slit beam for range finding by triangulation in scenes with polyhedra. Will and Pennington (BIBSA 1971) illuminate a scene with striped lighting and infer planar face orientation from properties of the 2-D spatial fourier transform. All of these methods represent a retreat from the tradition of analyzing natural scenes under natural lighting. In part, this retreat is an admission of the failure of attempts to correctly analyze any but the most artificial of scenes. Successful scene analysis, as modest as it is, is achieved only at great computational expense. Evidence for the latter is expressed by Smith and Coles (BIBGEN 1973) who report that analyzing a single complete scene requires ten minutes of central processor time on a large, high-powered computer at Stanford University. This is hopelessly slow for a practical robot to skillfully navigate a fork-lift loader, let alone drive a car. In the next chapter possible ways of overcoming this frustrating limitation will be examined.

# CHAPTER 3. OVERVIEW

The performance of artificial visual systems is far below even that of rather primitive natural visual systems. Though machine recognition of objects could be improved by combining several of the methods discussed previously, the result would be a cumbersome system requiring large amounts of computer time and memory space. The expense and physical size of such a system could of course be significantly reduced by the currently decreasing cost and size of computer components. However, speed cannot be significantly improved using conventional methods; without a thousandfold speedup, mobile robots which interact with reasonable environments are impossible. This is precisely the kind of impasse which haunts many other areas in AI; methods successful in toy worlds do not generalize effectively to real ones without drastic reduction in performance. The combinatoric explosion which occurs when complexity increases in generalizing to real world situations is often combatted by embodying semantics of the real world in heuristics. The way in which such semantics are represented is crucial and there are no general rules for choosing either the semantics or their best representation. This situation applies to the linguistic approach to scene analysis as well. There, objects, structures, and functional aggregates of objects correspond to nodes in a graph whose links correspond to relations between these objects. As the number of nodes increases in generalizing to more complex situations, the number of potential links increases exponentially. The problem of deciding which links are relevant becomes increasingly difficult and, in dynamic situations, relevance may change in nonobvious ways. Minsky's concept of frames represents an attempt to isolate subparts [illegible] the real world and thereby reduce linkage complexity.

The problems discussed above are general to AI and it appears, regrettably, that many artificially intelligent systems cannot be significantly generalized from their toy worlds. Chandrasekaran and Reeker (BIBGEN 1972) and Dreyfus (BIBGEN 1972) discuss some apparent limitations in AI research. However, in scene analysis it appears that breakthroughs can be realized by proper exploitation of geometric semantics. Since the phenomena of projective transformations are completely expressible mathematically, their semantics are much more tractable than those of other AI areas. Any mathematical system, however, can be expressed in a large number of different ways equivalent formally but enormously different in terms of the complexity of carrying out certain kinds of computations. Roberts, Mackworth, Horn and Hannah explicitly used projective semantics but in completely different representations, and for different purposes. Is there a better, more universal representation? Biological examples may help answer that question. Examination of biological processes has not been profitable in most areas of AI because very little is known about neurophysiological correlates of conceptual phenomera, e.g. words, thoughts or logic. In vision however, much is known about the neurophysiology and its low level computational capabilities. Computational geometry is a term used by Minsky and Papert (BIBSA 1969) to describe the logical mechanisms which are used to deduce geometric properties in quantized spaces. They discussed computations for topological geometry. The computational geometry of some affine transformations is discussed in Weiman and Rothstein (BIBTR 1972). Lateral inhibition is a good example of biologically observed geometric computation. This is the inhibition of activity of neighboring nerve cells by an active cell in a network. It is a phenomenon common to many sensory nerve networks; in vision it provides a mechanism for smoothing and taking spatial derivatives just like the finite differencing operators mentioned in the discussion of contrast features. Marr and Pettigrew (BIBTR 1973) and Marr (BIBTR 1974) have examined more subtle geometric computation capabilities of neurophysiological structures in the visual system.

Some of the problems of inferring 3-D scenes from 2-D pictures involve the solution of partial differential equations (PDE's) relating geometric quantities. In Horn's approach the differential quantities were components of tangents to surfaces and gradations in image intensity. Though Hannah's method does not use PDE's explicitly, the binocular disparities used as a basis for region analysis are angle differentials which are related to depth differentials. In fact, the relations between those quantities have been expressed in terms of differential geometry by Luneburg (BIBGEN 1947) to yield some rather remarkable and counter-intuitive conclusions about our binocular perception of visual space. A closely related topic that is virtually unexplored but probably fruitful is projective differential geometry (Wilczynski BIBGEN 1905). A computational advantage of representing pictures in discrete space (for example a retina or a digitized picture) is that derivatives of various orders correspond to finite differences and the operations of calculus are easily approximated by simple arithmetic operations as we saw in the discussion of contrast detectors. This fact is exploited in numerical analysis using grids for the numerical solution of PDE's where the analytic solution is not known or intractable. These numerical solutions are usually solved sequentially, one cell at a time. In the biological case, many nerve cells are active simultaneously. This kind of parallel processing gives biological organisms the capability of reacting much faster than artificial visual systems despite the fact that biological components are much slower. Parallel computation cannot only increase speed but often completely changes the expression of algorithms (Traub, BIBGEN 1973). In many cases, complex sequential algorithms can be expressed in terms of a large number of simple, cooperating, simultaneously active (parallel) algorithms. This is particularly relevant to the solution of PDE's; algebraically complicated global functions are often the solutions of simple differential (local) equations. Recently formal models of biological developmental processes have been generalized and studied in the theory of formal languages. Called L-systems after the biologist Lindenmayer who originated them, these systems are important to the theory of parallel computation as well as biological models (Herman and Rozenberg BIBGEN 1975). Horn (BIBPIC 1974) has recently incorporated parallel, discrete algorithms in a system for inferring illumination of surfaces from image intensities. Since image intensity is a function of surface orientation and illumination, this problem is closely related to his Ph.D. thesis topic discussed earlier. Remarkably, his approach explains perceptual phenomena previously poorly modelled and points to actual neurophysiological components of the mammalian retina which could be carrying out the parallel algorithms.

The parallel, numerical, PDE approaches just discussed in a sense deal with what could be called lower level data in vision. That is, they show how 3-D depth contours can be inferred but do not deal directly with the higher concepts of shape recognition, isolation of distinct bodies, and determination of relations between objects. Global abstractions are considered in gestalt psychology as being the result of associating parts on the basis of something they have in common (Haber and Hershenson, BIBGEN 1973). This includes grouping parts that are geometrically close to each other and grouping together parts that are similar to each other. Lester (BIBTR 1974 and 1975) has quantified the concepts of proximity and similarity in programs which group together very general kinds of genetric objects. Though he only considered 2-D pictures, application of this kind of idea to inferred 3-D structures could be useful. Hannah's approach implicitly joins regions on the basis of lateral and depth proximity while other region growing methods use a combination of similarity and proximity. If one generalizes "common fate" gestalt grouping concepts to include "common transformation" the role of motion in vision becomes very important. This is an area which has been virtually ignored in artificial vision which usually concentrates on analyzing form in static pictures and regards motion as an interpolation between two static pictures. This ignores completely the fact that motion detectors are present in large numbers in all mammallian visual nervous systems and that when images are stabilized on the human retina, vision ceases. The roles usually ascribed to motion detectors are providing feedback for eye movements and attracting the eye to parts of a scene which may move. Though there is certainly great adaptive value in these roles, motion detectors could also play a fundamental part in the perception of geometric form and inferring 3-D shape. Platt (BIBGEN 1960) has shown how null responses from motion detectors when a pattern is moved on a retina can be used to recognize straight lines and circles as a result of self-congruence properties of such figures under certain transformations. These particular figures are of central importance in projective geometry. Generalizing this motion to three dimensions, Johannsen (BIBGEN 1975) presents strong experimental evidence that human assumption of 3-D self-congruence of objects in a scene plays a powerful role in interpreting motion visible only on a 2-D display. This characteristic is exploited in the kinetic depth effect in computer graphics. When a 2-D image is transformed as it would be if the rigid 3-D object it represents were rotated in space, the image immediately "looks" 3-D. (Newman and Sproull BIBGEN 1973). Human perception of moving objects does not appear to be a succession of static frames; in real life stagecoach wheels do not appear to rotate backwards due to frame strobing as in motion pictures.

In summary, perceiving motion directly could provide a method for separating the effects of projective transformations and 3-D motions. Once these have been isolated, scene organization can be based on "common 3-D motion" as a gestalt grouping mechanism. For example, a still picture of a flock of brown birds flying together between the branches of a large brown tree would be very difficult to analyze purely by the methods of either region analysis, edge analysis or binocular disparity analysis. There are too many significant small regions and sharp edges to analyze, and many weak edges separate tree branches from birds. A motion detecting method, however, would group together all edges sharing the same speed and direction. This idea is very much in keeping with Gibson's (BIBGEN 1966) who proposes that sensory systems measure invariance of certain stimulus properties under transformations rather than properties in isolation. Mechanisms for detecting such invariances could be incorporated in integral geometry schemes. These are generalizations of the Buffon needle problem from statistics and probability and have the advantage that geometric properties can be measured to any desired accuracy independent of coordinate systems using statistical inference (Novikoff BIBGEN 1962). A pool of edge detectors would be an ideal candidate for embodying such mechanisms. Combining the ideas of this paragraph with the discussion of PDE's in analyzing static scenes, motion is the derivative of position. Contrast feature detectors linked by delays can be used as motion detectors. Relative motion of eye and image reduce motion detection and edge detection to the same kind of computation. Now, the systems of PDE's for static picture analysis are augmented by equations in which derivatives with respect to time are involved. This added constraint and the integral relation between motion and distance ought to be explored for possible embodiment in artificial visual systems.

The fact that the new approaches suggested in the preceding paragraphs do not strongly resemble traditional approaches is not meant to discredit the latter. All of them have elements which are important stepping stones incorporated into the new suggestions. Linguistic, traditionally AI methods have not been greatly discussed because I feel there are many geometric semantic problems which can be solved to put us far ahead of where we are now; at that time the conceptual, semantic structures such as Minsky's and Winston's will be even more important than they are now. Vision and intelligence can be treated separately, though intelligent vision is more powerful than vision alone. For example, the lowly housefly has excellent vision which is used to navigate at high speed, avoiding collision in environments containing far more complex objects and lighting extremes than those current robot vision systems maneuver through. It is doubtful that high level concepts and abstractions are involved. It appears quite feasible to build a system with similar capabilities. If in addition we incorporate knowledge of real world objects and relations into such a system, so much the better, but the latter is not necessary for excellent machine vision.

## CHAPTER 4. DESCRIPTION OF BIBLIOGRAPHY

The bibliography is divided into four parts whose names are also the names of indirect access permanent files on the CDC 6600 at the Courant Institute. The first three letters of the names of these files are always BIB; remaining letters are acronyms for the type of contents of each file. The format of these files is such that copies onto Hollerith cards yield symbols compatible with most computer systems, permitting portability. Each bibliographic reference is formatted for easy visual perusal and elementary information retrieval or updating using string processing languages or standard text editors. To simplify these tasks, all references are in a standard form that encompasses the wide diversity of publication types which range from dissertations to textbooks. In this standard form, each reference occupies three lines. The first three or four columns of each line are reserved for keys, handy links for information retrieval or quick visual reference. The first line of each entry contains the author's name(s), last name first followed by initials followed by the date of publication in parentheses (see BIBGIDE file which follows this section for other format details). References in the text are given by author(s) followed by the file name and date.

The division of the bibliography into four files is based on document accessibility and topic. The file BIBSA contains references closely related to scene analysis (hence SA in its acronym) and publically available in journals (key JP) books and conference proceedings. The key JP stands for journal paper, in which case the third line (key S for source) gives the name of the journal, volume number, and pages in that order. The journals Artificial Intelligence (American Elsevier, New York or North-Holland, Amsterdam) and Computer Graphics and Image Processing (Academic Press) contain the bulk of scene analysis literature in this form. Another important source are in the set of Proceedings of the International Joint Conferences on Artificial Intelligence referred to as PIJCAI in the bibliography and described at the bottom of the file BIBSA. Also important is the Machine Intelligence series, proceedings of international workshops in AI held in Edinburgh, Scotland. Information about this collection is also listed at the bottom of the file; references to it in the bibliography are abbreviated as MI in the third or source line. Often papers in both these collections are reworked and publishad in the journal Artificial Intelligence as can be deduced by looking at authors and titles; this redundancy is useful if one of the sources is unavailable. Finally, these collections also contain descriptions of robots which incorporate scene analysis and other methods in problem solving systems.

Two books (key BK in the second line of a reference) should be singled out for special attention in BIBSA. The first, a textbook by Duda and Hart (BIBSA 1973) gives comprehensive coverage of the areas in its title, excellent overviews, and good bibliographies. The second is Winston (BIBSA 1975) which could be subtitled "Scene Analysis at MIT." The coverage is not comprehensive over the field of scene analysis but contains versions of Minsky's paper on the concept of Frames, and Horn's and Waltz's dissertations; previously these were only available as technical reports.

The file BIBPIC contains a small selection of references in picture processing methods which were precursors of or are connected with scene analysis. Most important are the surveys by Rosenfeld which lead to further references in picture processing in a comprehensive and clearly presented way.

BIBGEN contains general references in mathematics, biology, and AI related to the discussion in Chapters 1 and 3 primarily. The Handbook of Sensory Physiology, Volume VII (the first entry in BIBGEN) is the most complete source on the neurophysiology of vision and can lead the reader to further references in that area. Most of the titles of other references in BIBGEN are self-explanatory.

BIBTR lists technical reports, dissertations and memos (keys TR, DI and ME, respectively, on the title line) and similar documents related to scene analysis but not easily accessible.

The source line (key S) lists the institution, department, degree if dissertation, and report number if any, in that order. Most of the work listed in BIBTR was carried out at the AI labs at MIT or Stanford University, California. These two institutions have strong professional interconnections and in the past have received considerable federal funding for AI research. Hence dissertations usually become technical reports, the most accessible type of document in BIBTR. Even if the technical reports are unavailable, the dissertation titles give the reader an idea of the major research interests of their authors; seeing the names later in journal articles can lead to a good guess about the topic of an article. In addition, revised versions of papers listed in BIBTR often find their way into those listed in BIBSA; thus, to find a more publically accessible form of a paper one need only match authors and seek similar titles in BIBSA.

The least accessible documents in BIBTR are memos, papers intended only for internal circulation within the source institution. Those of the artificial vision research group at MIT are often called Vision Flashes (abbreviated VF in the bibliography); other MIT memos in AI are often called Working Papers (abbreviated WP in the bibliography).

For information on obtaining the bibliography files as a deck of Hollerith cards, write to Malcolm Harrison, the principal investigator of the sponsoring grant, at the address on the title page of this report. In the listing of the deck which follows each single quotation mark (') comes out as a not-equal sign (≠); the former was removed from the printer chain at the Courant Institute to make room for special symbols. The card code is a 4-8 punch which is interpreted as a single quotation mark by most systems.

## BIBGIDE: GUIDE TO FILES FOR SCENE ANALYSIS SURVEY BIBLIOGRAPHY.

| | |
|---|---|
| BIBGEN | - BIBLIOGRAPHY OF GENERAL TOPICS SUCH AS MATH AND NEUROPHYSIOLOGY RELATED TO THIS SURVEY |
| BIBPIC | - BIBLIOGRAPHY OF PICTURE PROCESSING AND EDGE DETECTION |
| BIBSA | - BIBLIOGRAPHY OF EXPLICIT SCENE ANALYSIS PAPERS AND BOOKS, PUBLICALLY MARKETED (JOURNALS AND PUBLISHERS) |
| BIBTR | - BIBLIOGRAPHY OF TECHNICAL REPORTS AND SIMILARLY HARD TO GET DOCUMENTS CLOSELY RELATED TO SCENE ANALYSIS |

1) IN ANY FILE EACH REFERENCE CONSISTS OF THREE LINES; THE FIRST ONE OR TWO COLUMNS OF EACH LINE HOLD A KEY WHICH REFERS TO THE KIND OF INFORMATION CONTAINED ON THAT LINE. THE FIRST LINE ALWAYS CONTAINS THE AUTHOR≠S NAME(S), LAST NAME FIRST STARTING IN COLUMN FOUR, FOLLOWED BY INITIALS AND THEN THE YEAR OF PUBLICATION IN PARENTHESES; THE KEY TO THE FIRST LINE IS ≠A≠ (FOR AUTHOR) IN COLUMN 1.

2) THE SECOND LINE IS ALWAYS THE TITLE ENCLOSED IN SINGLE QUOTES STARTING IN COLUMN SIX, BUT THE TWO LETTER KEY IN COLUMNS ONE AND TWO GIVE THE TYPE OF PUBLICATION AS FOLLOWS:

| | |
|---|---|
| BK | FOR BOOK, |
| CF | FOR CONFERENCE PROCEEDINGS (UNLESS PUBLISHED AS A BOOK WHICH IS PUBLICALLY MARKETED) |
| DI | FOR A DISSERTATION OR THESIS |
| IN | FOR A PAPER IN SOME OTHER WORK |
| JP | FOR JOURNAL PAPER |
| ME | FOR MEMO |
| TR | FOR TECHNICAL REPORT |

3) THE THIRD LINE GIVES THE PUBLISHING SOURCE (KEY S IN COLUMN ONE) OF A WORK STARTING IN COLUMN SIX. EXAMPLES OF THIS ARE THE PUBLISHER FOR A BOOK, THE NAME OF A JOURNAL, THE INSTITUTION FOR A TECHNICAL REPORT OR THESIS, THE AUTHOR(EDITOR) AND TITLE FOR A COLLECTION A PAPER APPEARS IN.

WHERE CONTINUATION LINES ARE NECESSARY, THEY APPEAR WITHOUT KEYS. A BLANK LINE FOLLOWS EACH REFERENCE. SEE MODELS BELOW:

```
A  LASTNAM1,I.J., LASTNAM2,K.L. AND LASTNAM3,M.N. (YEAR)
JP   ≠TITLE OF PAPER≠
S    JOURNAL TITLE, VOL 3, PAGES 234-345.

A  LASTNAM,I.J. (YEAR)
TR   ≠TITLE OF TECHNICAL REPORT≠
S    INSTITUTION, DEPARTMENT, LOCATION, TECH REPORT NUMBER.
```

BIBGEN: GENERAL REFERENCES IN MATH, BIOLOGY, NEUROPHYSIOLOGY,
ARTIFICIAL INTELLIGENCE, PSYCHOLOGY, AND PHILOSOPHY.

**************************************************************

A  ASSORTED EDITORS (1972 AND LATER FOR VARIOUS PARTS)
BK  ≠HANDBOOK OF SENSORY PHYSIOLOGY≠, VOLUME VII, VISION
S  SPRINGER-VERLAG, BERLIN AND NEW YORK.

A  BUSEMANN, A. AND KELLY, P.J. (1953)
BK  ≠PROJECTIVE GEOMETRY AND PROJECTIVE METRICS≠
S  ACADEMIC PRESS, NEW YORK.

A  CHANDRASEKARAN, B. AND REEKER, L.H. (1972)
TR  ≠ARTIFICIAL INTELLIGENCE- A CASE FOR AGNOSTICISM≠
S  OHIO STATE UNIV, COMP AND INFO SCI DEPT, CISRC-TR-72-9.

A  COXETER, H.S.M. (1964)
BK  ≠PROJECTIVE GEOMETRY≠
S  BLAISDELL PUB CO, NEW YORK.

A  DEUTSCH, S. (1966)
JP  ≠CONJECTURES ON MAMMALIAN NEURON NETWORKS FOR VISUAL
PATTERN RECOGNITION≠
S  TRANS IEEE, SSC2, DECEMBER, PAGE 81.

A  DODWELL, P.C. (1970)
BK  ≠VISUAL PATTERN RECOGNITION≠
S  HOLT, RINEHART, AND WINSTON, NEW YORK.

A  DREYFUS, H.L. (1972)
BK  ≠WHAT COMPUTERS CAN≠T DO - A CRITIQUE OF ARTIFICIAL REASON≠
S  HARPER AND ROW, NEW YORK.

A  FIRSCHEIN, O., FISCHLER, M.A., COLES, L.S. AND TENENBAUM, J.M. (1973)
IN  ≠FORECASTING AND ASSESSING THE IMPACT OF AI ON SOCIETY≠
S  PIJCAI 3, STANFORD, PAGES 105-120.

A  GIBSON, J.J. (1966)
BK  ≠THE SENSES CONSIDERED AS PERCEPTUAL SYSTEMS≠
S  HOUGHTON-MIFFLIN, BOSTON.

A  HABER, R.N. AND HERSHENSON, M. (1973)
BK  ≠THE PSYCHOLOGY OF VISUAL PERCEPTION≠
S  HOLT, RINEHART, AND WINSTON, NEW YORK.

A  HERMAN, G.T. AND ROZENBERG, G. (1975)
BK  ≠DEVELOPMENTAL SYSTEMS AND LANGUAGES≠
S  NORTH-HOLLAND, AMSTERDAM.

```
A   HEWITT, C.  (1971)
IN    #PROCEDURAL EMBEDDING OF KNOWLEDGE IN PLANNER#
S     PIJCAI 2, LONDON, PAGES 167-184

A   JOHANNSON, G. (1975)
JP    #VISUAL MOTION AND PERCEPTION#
S     SCIENTIFIC AMERICAN, VOL 232, JUNE, PAGES 76-88.

A   JULESZ, B. (1971)
BK    #FOUNDATIONS OF CYCLOPEAN PERCEPTION#
S     UNIVERSITY OF CHICAGO PRESS, CHICAGO, ILL.

A   LUNEBURG, R.K.  (1947)
BK    #MATHEMATICAL ANALYSIS OF BINOCULAR VISION#
S     DARTMOUTH EYE INSTITUTE, HANOVER, NEW HAMPSHIRE.

A   MCCARTHY, J. AND HAYES, P.J. (1969)
IN    #SOME PHILOSOPHICAL PROBLEMS FROM THE STANDPOINT OF AI#
S     MI 4, PAGES 463-502.

A   MCCULLOCH, W.S. (1945)
JP    #A HETERARCHY OF VALUES DETERMINED BY THE TOPOLOGY OF NERVOUS NETS#
S     BULLETIN OF MATHEMATICAL BIOPHYSICS, VOL 7, PAGES 89-93.

A   MCCULLOCH, W.S. (1965)
BK    #EMBODIMENTS OF MIND#
S     MIT PRESS, CAMBRIDGE,MASS.

A   MINSKY, M. AND PAPERT, S. (1967)
BK    #PROJECT MAC PROGRESS REPORT IV#
S     MIT PRESS, CAMBRIDGE, MASS.

A   NEWMAN, W.M. AND SPROULL, R.F. (1973)
BK    #PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS#
S     MCGRAW-HILL, NEW YORK.

A   NILSSON,N.J. (1971)
BK    #PROBLEM SOLVING METHODS IN AI#
S     MCGRAW-HILL, NEW YORK.

A   NOVIKOFF, A. (1962)
IN    #INTEGRAL GEOMETRY AS A TOOL IN PATTERN PERCEPTION#
S     VON FOERSTER AND ZOPF (EDS), #PRINCIPLES OF SELF-ORGANIZATION#
      PERGAMON PRESS, MACMILLAN, NEW YORK, PAGES 347-368.

A   PLATT, J.R.  (1960)
JP    #HOW WE SEE STRAIGHT LINES#
S     SCIENTIFIC AMERICAN, VOL 202, JUNE, PAGES 121-129.

A   RATTLIFF, F. (1965)
BK    #MACH BANDS: QUANTITATIVE STUDIES ON NEURAL NETWORKS IN THE RETINA#
S     HOLDEN-DAY, SAN FRANCISCO.
```

```
A   REISS, R.F. (EDITOR) (1964)
BK    ≠NEURAL THEORY AND MODELLING≠
S     STANFORD UNIV PRESS, STANFORD, CALIF.

A   ROTHSTEIN, J. (1956)
BK    ≠INFORMATION, LOGIC AND PHYSICS≠
S     PHILOSOPHY OF SCIENCE, VOL 23, PAGES 31-35.

A   SMITH, M.H. AND COLES, L.S. (1973)
IN    ≠DESIGN OF A LOW COST, GENERAL PURPOSE ROBOT≠
S     PIJCAI 3, STANFORD, PAGES 324-335.

A   SPRINGER, C.E. (1964)
BK    ≠GEOMETRY AND ANALYSIS OF PROJECTIVE SPACES≠
S     W. H. FREEMAN, SAN FRANCISCO.

A   TRAUB, J.F. (1973)
BK    ≠COMPLEXITY OF SEQUENTIAL AND PARALLEL NUMERICAL ALGORITHMS≠
S      ACADEMIC PRESS, NEW YORK.

A   TURING, A.M. (1947)
IN    ≠INTELLIGENT MACHINERY≠
S     MI 5, 1970, PAGES 3-23.

A   VON SENDEN, M. (1960)
BK    ≠SPACE AND SIGHT≠
S     THE FREE PRESS, GLENCOE, ILLINOIS.

A   WILCZYNSKI, E.J. (1905)
BK    ≠PROJECTIVE DIFFERENTIAL GEOMETRY OF CURVES AND RULED SURFACES≠
S     CHELSEA, NEW YORK, (1961 REPRINT OF 1905 TEUBNER PUBLICATION).

A   WINOGRAD, T. (1972)
BK    ≠UNDERSTANDING NATURAL LANGUAGE≠
S     ACADEMIC PRESS, NEW YORK.

**********************************************************************
```

```
BIBPIC: BIBLIOGRAPHY OF PAPERS, BOOKS, AND CONFERENCES ON
        PICTURE PROCESSING, PATTERN RECOGNITION AND EDGE DETECTION.
*******************************************************************

A   DAVIS, L. S. (1975)
JP    *A SURVEY OF EDGE DETECTION TECHNIQUES*
S     COMPUTER GRAPHICS AND IMAGE PROCESSING, VOL 4, PAGES 248-270.

A   DUDA,R.O. AND HART,P.E.,(1972)
JP    *USE OF THE HOUGH TRANSFORM TO DETECT LINES AND CURVES IN
      PICTURES*
S     CACM, VOL 15, PAGES 11-15.

A   FREEMAN, H. (1974)
JP    *COMPUTER PROCESSING OF LINE DRAWING IMAGES*
S     ACM COMPUTING SURVEYS, VOL 6, PAGES 57-97.

A   GERARDIN, L.A. AND FLAMENT, J. (1969)
IN    *GEOMETRICAL PATTERN FEATURE EXTRACTION BY PROJECTION ON
      HAAR ORTHONORMAL BASIS*
S     PIJCAI 1, WASH DC, PAGES 65-78.

A   GRASSELLI, A. (ED)  (1969)
BK    *AUTOMATIC INTERPRETATION AND CLASSIFICATION OF IMAGES*
S     ACADEMIC PRESS, NEW YORK.

A   GRIFFITH, A. K. (1971)
IN    *MATHEMATICAL MODELS FOR AUTOMATIC LINE DETECTION*
S     PIJCAI 2, LONDON, PAGES 17-26.

A   GRIFFITH, A. K. (1973)
JP    *MATHEMATICAL MODELS FOR AUTOMATIC LINE DETECTION*
S     JACM, VOL 20, PAGES 62-80.

A   HORN, B.K.P. (1974)
JP    *DETERMINING LIGHTNESS FROM AN IMAGE*
S     COMPUTER GRAPHICS AND IMAGE PROCESSING, VOL 3, PAGES 277-299.

A   HUECKEL, M.H. (1971)
JP    *AN OPERATOR WHICH LOCATES EDGES IN DIGITIZED PICTURES*
S     JACM, VOL 18, PAGES 113-125.

A   HUECKEL, M.H.  (1973)
JP    *A LOCAL VISUAL OPERATOR WHICH RECOGNIZES EDGES AND LINES*
S     JACM, VOL 20, PAGES 634-647.

A   KANEFF, S. (ED)  (1970)
BK    *PICTURE LANGUAGE MACHINES*
S     ACADEMIC PRESS, NEW YORK.

A   LEVINE, M.D., O*HANDLEY, D.A. AND YAGI, G.M. (1973)
JP    *COMPUTER DETERMINATION OF DEPTH MAPS*
S     COMPUTER GRAPHICS AND IMAGE PROCESSING, VOL 2, PAGES 131-150.
```

```
A  LIPKIN, B.S. AND ROSENFELD, A. (EDS) (1970)
BK    #PICTURE PROCESSING AND PSYCHOPICTORICS#
S     ACADEMIC PRESS, NEW YORK.

A  O#GORMAN F. AND CLOWES, M.B. (1973)
IN    #FINDING EDGES THROUGH COLLINEARITY OF FEATURE POINTS#
S     PIJCAI 3, STANFORD, PAGES 543-555.

A  MILLER, W. AND SHAW, A. (1968)
IN    #LINGUISTIC METHODS IN PICTURE PROCESSING - A SURVEY#
S     AFIPS PROC FALL JOINT COMP CONF, VOL 33, PAGE 279.

A  PINGLE, K.K. AND TENENBAUM, J.M. (1971)
IN    #AN ACCOMODATING EDGE FOLLOWER#
S     PIJCAI 2, LONDON, PAGES 1-7.

A  QUAM, L.H. (1971)
DI    #COMPUTER COMPARISON OF PICTURES#
S     STANFORD UNIV, COMPUTER SCI DEPT, PH D, AIM-144.

A  RAMER, U.E. (1975)
JP    #EXTRACTION OF LINE STRUCTURES FROM PHOTOGRAPHS OF CURVED OBJECTS#
S     COMPUTER GRAPHICS AND IMAGE PROCESSING,VOL 4, PAGES 81-103.

A  REED, S.K. (1973)
BK    #PSYCHOLOGICAL PROCESSES IN PATTERN RECOGNITION#
S     ACADEMIC PRESS, NEW YORK.

A  ROSENFELD, A. (1969)
BK    #PICTURE PROCESSING BY COMPUTER#
S     ACADEMIC PRESS, NEW YORK.

A  ROSENFELD, A.  (1969)
JP    #PICTURE PROCESSING BY COMPUTER#
S     ACM COMPUTING SURVEYS, VOL 1, PAGES 147-176.

A  ROSENFELD, A. (1972)
JP    #PICTURE PROCESSING: 1972#
S     COMPUTER GRAPHICS AND IMAGE PROCESSING, VOL 1, PAGES 394-416.

A  ROSENFELD, A. (1973)
JP    #PROGRESS IN PICTURE PROCESSING: 1969-1971#
S     ACM COMPUTING SURVEYS, VOL 5, PAGES 81-108.

A  ROSENFELD, A. (1974)
JP    #PICTURE PROCESSING: 1973#
S     COMPUTER GRAPHICS AND IMAGE PROCESSING, VOL 3, PAGES 178-194.

A  ROSENFELD, A. (1975)
JP    #PICTURE PROCESSING: 1974#
S     COMPUTER GRAPHICS AND IMAGE PROCESSING, VOL 4, PAGES 133-155.
```

```
A   ROTHSTEIN, J. AND WEIMAN, C.F.R. (1976)
JP    ≠PARALLEL AND SEQUENTIAL SPECIFICATION OF A CONTEXT SENSITIVE
      LANGUAGE FOR STRAIGHT LINES ON GRIDS≠
S     COMPUTER GRAPHICS AND IMAGE PROCESSING (TO APPEAR).

A   SMITH, M.W. AND DAVIS, W.A. (1975)
JP    ≠A NEW ALGORITHM FOR EDGE DETECTION≠
S     COMPUTER GRAPHICS AND IMAGE PROCESSING, VOL 4, PAGES 55-62.

A   STAMOPOULOS, C.D. (1975)
JP    ≠PARALLEL IMAGE PROCESSING≠
S     IEEE TC, VOL C-24, PAGES 424-433.

A   TOU, J.T. AND GONZALEZ, R.C. (1974)
BK    ≠PATTERN RECOGNITION PRINCIPLES≠
S     ADDISON-WESLEY, NEW YORK.

A   TOUSSAINT, G.T. (1975)
JP    ≠SUBJECTIVE CLUSTERING AND BIBLIOGRAPHY OF BOOKS ON
      PATTERN RECOGNITION≠
S     INFORMATION SCIENCES, VOL 8, PAGES 251-257.

A   UHR, L. (1973)
BK    ≠PATTERN RECOGNITION, LEARNING, AND THOUGHT≠
S     PRENTICE-HALL, NEW YORK.

A   YOUNG, J. F. (1973)
BK    ≠ROBOTICS≠
S     BUTTERWORTHS, LONDON

******************************************************************
```

```
BIBSA: BIBLIOGRAPHY OF BOOKS, JOURNAL ARTICLES AND SIMILAR
       PUBLICALLY MARKETED PUBLICATIONS CONTAINING EXPLICIT
       SCENE ANALYSIS MATERIAL.
**************************************************************

A  AGIN, G.J. AND BINFORD, T.O. (1973)
IN   #COMPUTER DESCRIPTION OF CURVED OBJECTS#
S    PIJCAI 3, STANFORD, PAGES 629-640.

A  BARROW, C.R. AND POPPLESTONE, R.J. (1971)
IN   #RELATIONAL DESCRIPTIONS IN PICTURE PROCESSING#
S    MI 6, PAGES 377-396.

A  BRICE, C. R. AND FENEMA, C. L. (1970)
JP   #SCENE ANALYSIS USING REGIONS#
S    ARTIFICIAL INTELLIGENCE, VOL 1, PAGES 205-226.

A  CLOWES,M.B., (1971)
JP   #ON SEEING THINGS#
S    ARTIFICIAL INTELLIGENCE, VOL 2, PAGES 79-116.

A  DUDA, R.O. AND HART, P.E. (1973)
BK   #PATTERN CLASSIFICATION AND SCENE ANALYSIS#
S    WILEY, NEW YORK.

A  FAHLMAN, S.E.,(1974)
JP   #A PLANNING SYSTEM FOR ROBOT CONSTRUCTION TASKS#
S    ARTIFICIAL INTELLIGENCE, VOL 5, PAGES 1-49.

A  FALK, G. (1971)
IN   #SCENE ANALYSIS BASED ON IMPERFECT EDGE DATA#
S    PIJCAI 2, LONDON, PAGES 8-16.

A  FALK, G. (1972)
JP   #INTERPRETATION OF IMPERFECT LINE DATA AS A
     THREE DIMENSIONAL SCENE#
S    ARTIFICIAL INTELLIGENCE, VOL 3, PAGES 101-144.

A  FELDMAN, J.A., FELDMAN, G.M., FALK, G., GRAPE, G., PEARLMAN, J.,
   SOBEL, I., AND TENENBAUM, J. M. (1973)
IN   #THE STANFORD HAND-EYE PROJECT#
S    PIJCAI 1, WASH DC, PAGES 521-526.

A  FELDMAN, J.A. AND YAKIMOVSKY, Y.  (1974)
JP   #DECISION THEORY AND ARTIFICIAL INTELLIGENCE: I. A SEMANTICS-BASED
     REGION ANALYSER#
S    ARTIFICIAL INTELLIGENCE, VOL 5, PAGES 349-371.

A  GUZMAN, A. (1968)
IN   #DECOMPOSITION OF A VISUAL SCENE INTO THREE DIMENSIONAL BODIES#
S    PROC OF THE FJCC, VOL 33, PAGES 291-304.
```

```
A  HUFFMAN, D.A. (1971)
IN   #IMPOSSIBLE OBJECTS AS NONSENSE SENTENCES#
S    MI 6, PAGES 295-323.

A  MACKWORTH, A.K.  (1973)
JP   #INTERPRETING PICTURES OF POLYHEDRAL SCENES#
S    ARTIFICIAL INTELLIGENCE, VOL 4, PAGES 121-137.

A  MACKWORTH, A.K. (1973)
IN   #INTERPRETING PICTURES OF POLYHEDRAL SCENES#
S    PIJCAI 3, STANFORD, PAGES 556-563.

A  MINSKY, M. AND PAPERT, S. (1969)
BK   #PERCEPTRONS, AN INTRODUCTION TO COMPUTATIONAL GEOMETRY#
S    MIT PRESS, CAMBRIDGE, MASS.

A  NEVATIA, R. AND BINFORD, T. (1973)
IN   #STRUCTURED DESCRIPTIONS OF COMPLEX OBJECTS#
S    PIJCAI 3, STANFORD, PAGES 641-647.

A  NILSSON, N.J. (1969)
IN   #A MOBILE AUTOMATON: AN APPLICATION OF AI TECHNIQUES#
S    PIJCAI 1, WASH DC, PAGES 509-520.

A  ROBERTS, L.G. (1965)
IN   #MACHINE PERCEPTION OF THREE-DIMENSIONAL OBJECTS#
S    TIPPET, J.T. ET AL (EDS) #OPTICAL AND ELECTRO-OPTICAL INFORMATION
     PROCESSING#, MIT PRESS, PAGES 159-197.

A  SHIRAI, Y. (1973)
JP   #A CONTEXT-SENSITIVE LINE FINDER FOR RECOGNITION OF POLYHEDRA#
S    ARTIFICIAL INTELLIGENCE, VOL 4, PAGES 95-120.

A  SHIRAI, Y. AND SABURO, T. (1971)
IN   #EXTRACTION OF THE LINE DRAWINGS OF 3-D OBJECTS BY SEQUENTIAL
     ILLUMINATION#
S    PIJCAI 2, LONDON, PAGES 71-87.

A  SHIRAI, Y. AND SUWA, M. (1971)
IN   #RECOGNITION OF POLYHEDRONS WITH A RANGE FINDER#
S    PIJCAI 2, LONDON, PAGES 80-87.

A  SOBEL, I. (1973)
IN   #ON CALIBRATING COMPUTER CONTROLLED CAMERAS FOR PERCEIVING
     3-D SCENES#
S    PIJCAI 3, STANFORD, PAGES 648-652.

A  SUBEL, I.  (1974)
JP   #ON CALIBRATING COMPUTER CONTROLLED CAMERAS FOR PERCEIVING 3-D
     SCENES#
S    ARTIFICIAL INTELLIGENCE, VOL 5, PAGES 185-198.

A  TOMITA, F. AND MASAHIKO, Y. (1973)
IN   #DETECTION OF HOMOGENEOUS REGIONS BY STRUCTURAL ANALYSIS#
S    PIJCAI 3, STANFORD, PAGES 564-571.
```

A  WILL, P.M. AND PENNINGTON, K.S. (1971)
JP    #GRID CODING: A PREPROCESSING TECHNIQUE FOR ROBOT AND MACHINE
      VISION#
S     ARTIFICIAL INTELLIGENCE, VOL 2, PAGES 319-329.

A  WILL, P.M. AND PENNINGTON, K.S. (1971)
IN    #GRID CODING: A PREPROCESSING TECHNIQUE FOR ROBOT AND MACHINE
      VISION#
S     PIJCAI 2, LONDON, PAGES 66-70.

A  WINSTON, P. H. (1972)
IN    #THE M.I.T. ROBOT#
S     MI 7, PAGES 431-465.

A  WINSTON, P. (1975)
BK    #THE PSYCHOLOGY OF COMPUTER VISION#
S     MCGRAW-HILL, NEW YORK.

A  YAKIMOVSKY, Y. AND FELDMAN, J.A. (1973)
IN    #A SEMANTICS-BASED DECISION THEORY REGION ANALYSER#
S     PIJCAI 3, STANFORD, PAGES 580-588.

*****************************************************************

A  INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
CF    PROCEEDINGS OF FIRST  (PIJCAI 1), WASHINGTON D.C. (1969)
      PROCEEDINGS OF SECOND (PIJCAI 2), LONDON (1971)
      PROCEEDINGS OF THIRD  (PIJCAI 3), STANFORD, CALIF. (1973)
S  THESE CONSIST OF THREE SEPARATE VOLUMES; FOR THE 3RD
   AND INFORMATION ON THE OTHERS WRITE TO
      STANFORD RESEARCH INSTITUTE, PUBLICATIONS DEPARTMENT
      333 RAVENSWOOD AVENUE
      MENLO PARK, CALIFORNIA 94025

------------------------------------------------------------------

THE UNIVERSITY OF EDINBURGH, GREAT BRITAIN, HELD ANNUAL AI WORKSHOPS. PROCEEDINGS WERE PUBLISHED UNDER THE TITLE #MACHINE INTELLIGENCE#, VOLUMES 1 THROUGH 7 BY AMERICAN-ELSEVIER, NEW YORK. THE EDITORS, VOLUME NUMBERS, AND DATES OF PUBLICATION ARE GIVEN BELOW:

| | | |
|---|---|---|
| COLLINS AND MITCHIE | MI 1 | (1967) |
| DALE AND MITCHIE | MI 2 | (1968) |
| MITCHIE | MI 3 | (1969) |
| MELTZER,MITCHIE,SWANN | MI 4 | (1969) |
| MELTZER AND MITCHIE | MI 5 | (1970) |
| MELTZER AND MITCHIE | MI 6 | (1971) |
| MELTZER AND MITCHIE | MI 7 | (1972) |

------------------------------------------------------------------

```
BIBTR: TECHNICAL REPORTS AND SIMILAR DOCUMENTS NOT EASILY
       ACCESSIBLE OR WORSE.
**********************************************************

A  AGIN,G. (1972)
DI   #REPRESENTATION AND DESCRIPTION OF CURVED OBJECTS#
S    STANFORD UNIV,AI PROJECT, PH D, AIM-173.

A  BAUMGART,B.  (1975)
DI   #COMPUTER GRAPHICS AND VISUAL PERCEPTION#
S    STANFORD UNIV,COMP SCI DEPT, PH D.

A  BOBERG,R.  (1972)
DI   #GENERATING LINE DRAWINGS FROM ABSTRACT SCENE DESCRIPTIONS#
S    MIT, EE DEPT, MASTERS.

A  FALK, G. (1970)
DI   #MACHINE ANALYSIS OF MULTI-BODY SCENES#
TR   #COMPUTER INTERPRETATION OF IMPERFECT LINE DATA AS
     A THREE DIMENSIONAL SCENE#
S    STANFORD UNIV, COMP SCI, PH D, CS-180 OR AIM-132.

A  FININ, T. (1971)
ME   #FINDING THE SKELETON OF A BRICK#
S    MIT, AI LAB, WP 19.

A  FREUDER, E.C. (1972)
ME   #RECOGNITION OF REAL OBJECTS#
S    MIT, AI LAB, VF 33 (ALSO IN WINSTON, BIBTR 1973).

A  FREUDER, E. C., (1973)
ME   #SUGGESTION AND ADVICE#
S     MIT, AI LAB, VF 43.

A  FREUDER, E.C. (1973)
ME   #ACTIVE KNOWLEDGE#
S    MIT, AI LAB, VF 53.

A  GILL, A  (1972)
DI   #VISUAL FEEDBACK AND RELATED PROBLEMS IN COMPUTER CONTROLLED
     HAND-EYE COORDINATION#
S    STANFORD UNIV, COMP SCI DEPT, PH D, AIM-178.

A  GRAPE, G. (1973)
DI   #MODEL BASED (INTERMEDIATE LEVEL) COMPUTER VISION#
S    STANFORD UNIV, COMP SCI DEPT, PH D, AI MEMO 201.

A  GRIFFITH, A. (1970)
DI   #COMPUTER RECOGNITION OF PRISMATIC SOLIDS#
S    MIT, PROJECT MAC, PH D, MAC-TR-73.
```

A GUZMAN A. (1967)
DI #SOME ASPECTS OF PATTERN RECOGNITION BY COMPUTER#
S MIT, EE, MASTERS, MAC-TR-37.

A GUZMAN, A. (1968)
DI #COMPUTER RECOGNITION OF THREE-DIMENSIONAL OBJECTS IN A VISUAL SCENE#
S MIT, EE, PH D, MAC-TR-59.

A HANNAH, M.J. (1974)
DI #COMPUTER MATCHING OF AREAS IN STEREO IMAGES#
S STANFORD UNIV, COMP SCI, PH D, AIM-239 OR CS-74-438.

A HANSON, A.R. AND RISEMAN, E.M. (1974)
TR #PREPROCESSING CONES: A COMPUTATIONAL STRUCTURE FOR SCENE ANALYSIS#
S U OF MASS, COMP AND INFO SCI DEPT,TR C-7.

A HART, P.E., FIKES, R.E., GARVEY, T. D., NILSSON, N.J., NITZAN, D., TENENBAUM, J. M. AND WILBUR, B.M. (1972)
TR #ARTIFICIAL INTELLIGENCE - RESEARCH AND APPLICATIONS#
S SRI, ANNUAL TR TO ARPA, OCT 71 - OCT 72.

A HERSKOVITS, A. AND BINFORD, T. (1970)
ME #ON BOUNDARY DETECTION#
S MIT, AI LAB, MEMO 183.

A HOLLERBACH, J.M. (1972)
ME #THE PROJECTIVE APPROACH TO OBJECT DESCRIPTION#
S MIT, AI LAB, WP 37.

A HORN, B.K.P. (1969)
ME #THE IMAGE DISSECTOR #EYES# #
S MIT, AI LAB, MEMO 178.

A HORN, B. K. P. (1970)
DI #SHAPE FROM SHADING: A METHOD FOR OBTAINING THE SHAPE OF A SMOOTH OPAQUE OBJECT FROM ONE VIEW#
S MIT, EE DEPT, PH D, MAC TR-79 (ALSO IN WINSTON, BIBSA 1975).

A HORN, B.K.P. (1972)
ME #VISNEM: A BAG OF #ROBOTICS# FORMULAE#
S MIT, AI LAB, VF34 (ALSO IN WINSTON, BIBTR 1973).

A HORN, B.K.P. (1973)
ME #THE BINFORD-HORN LINE-FINDER#
S MIT, AI LAB, AI MEMO 285.

A HORN, B.K.P. (1973)
ME #ON LIGHTNESS#
S MIT, AI LAB, AI MEMO 295.

```
A  KRAKAUER, L.J. (1971)
DI   ≠COMPUTER ANALYSIS OF VISUAL PROPERTIES OF CURVED OBJECTS≠
S    MIT, EE DEPT, PH D, MAC TR-82, AI-TR-234.

A  LERMAN, J. B. (1973)
DI   ≠COMPUTER PROCESSING OF STEREO IMAGES FOR AUTOMATIC
     EXTRACTION OF RANGE≠
S    MIT, EE DEPT, MS.


A  LERMAN, J.B. AND WOODHAM, R. J. (1973)
ME   ≠THE ≠TRACK≠ PROGRAM PACKAGE≠
S    MIT, AI LAB, VF49.

A  LESTER, J.M. (1974)
DI   ≠EDGE DETECTION IN COMPLEX SCENES BASED ON GESTALT PRINCIPLES
     OF PROXIMITY AND SIMILARITY≠
S    UNIV OF WISC, COMP SCI DEPT, TR-229.

A  LESTER, J.M. (1975)
DI   ≠A CLUSTERING MODEL OF BOUNDARY FORMATION≠
S    UNIV OF WISC, COMP SCI DEPT, PH D, TR-239.

A  MARR, D. (1974)
ME   ≠AN ESSAY ON THE PRIMATE RETINA≠
S    MIT, AI LAB, AI MEMO 296.

A  MARR, D. AND HEWITT, C. (1973)
ME   ≠VIDEO ERGO SCIO, AN ESSAY ON SOME THINGS WE WOULD LIKE A VISION
     SYSTEM TO KNOW≠
S    MIT, AI LAB, WP 60.

A  MARR, D. AND PETTIGREW, J.D. (1973)
ME   ≠QUANTITATIVE ASPECTS OF THE COMPUTATION PERFORMED BY VISUAL
     CORTEX IN THE CAT WITH A NOTE ON A FUNCTION OF LATERAL INHIBITION≠
S    MIT, AI LAB, WP 55.

A  MINSKY, M. AND PAPERT, S. (1970)
ME   ≠PROPOSAL TO ARPA FOR RESEARCH ON AI AT MIT≠
S    MIT PROJECT MAC, AI MEMO 185.

A  MINSKY, M. AND PAPERT, S.  (1972)
ME   ≠ARTIFICIAL INTELLIGENCE PROGRESS REPORT≠
S    MIT, AI LAB, MAC PROGRESS REPORT VIII, AI MEMO 252.

A  MINSKY, M. AND PAPERT, S. (1972)
ME   ≠RESEARCH ON INTELLIGENT AUTOMATA≠
S    MIT, AI LAB, PROPOSAL TO ARPA, OCT 1972.

A  MINSKY, M.  (1974)
ME   ≠A FRAMEWORK FOR REPRESENTING KNOWLEDGE≠
S    MIT, AI LAB, MEMO 306 (ALSO IN WINSTON, BIBSA 1975).
```

```
A  NILSSON, N.J., AGIN, G., BARROW, H., BRAIN, A.E., DEUTSCH, B.,
   FIKES, R.E., GARVEY, T.D., LYNCH, D.C., SACERDOTI, E.D. AND
   TENENBAUM, J.K. (1974)
TR    #ARTIFICIAL INTELLIGENCE - RESEARCH AND APPLICATIONS#
      (PROGRESS REPORT TO ARPA, 1972-1974)
S     STANFORD RESEARCH INSTITUTE, MENLO PARK, CALIF.

A  NEWBORN, M. (1974)
DI    #COMPUTER VISION: THE ANALYSIS OF SCENES CONTAINING POLYHEDRA#
S     NYU, COMP SCI, MASTERS.


A  ORBAN, R. (1970)
ME    #REMOVING SHADOWS FROM A SCENE#
S     MIT, AI LAB, B.S., MEMO 192.

A  PAUL, R., FALK, G. AND FELDMAN J.A.  (1969)
ME    #THE COMPUTER REPRESENTATION OF SIMPLY DESCRIBED SCENES#
S     STANFORD UNIV, AI PROJECT, AI MEMO 101.

A  PERKINS, D. (1970)
DI    #STEREO PERCEPTION OF LINE DRAWING PAIRS#
S     MIT, PH D.

A  PRICE, K. (1974)
ME    #THE DETECTION AND REPRESENTATION OF MOTION BY COMPUTER
      USING A THREE-DIMENSIONAL MODEL#
S     CARNEGIE-MELLON UNIV, PITTSBURGH, COMPUTER SCIENCE THESIS PROPOSAL

A  RAMER, U. E. (1973)
DI    #COMPUTER EDGE EXTRACTION FROM PHOTOGRAPHS OF CURVED OBJECTS#
S     NYU, EE, PH D, TR CRL-34.

A  RATTNER, M. H. (1970)
DI    #EXTENDING GUZMAN#S #SEE# PROGRAM#
S     MIT, AI LAB, B.S., MEMO 204.

A  RISEMAN, E. M. AND HANSON, A.R. (1974)
TR    #DESIGN OF A SEMANTICALLY DIRECTED VISION PROCESSOR#
S     U MASS, COMP SCI, TR 74C-1.

A  ROBERTS, L.G. (1965)
TR    #HOMOGENEOUS MATRIX REPRESENTATION AND MANIPULATION OF
      N-DIMENSIONAL CONSTRUCTS#
S     MIT, LINCOLN LABS, MEMO 1405.

A  SHIRAI, Y.  (1972)
ME    #A HETEPARCHICAL PROGRAM FOR RECOGNITION OF POLYHEDRA#
S     MIT, AI LAB, MEMO 263 (ALSO IN WINSTON, BIBTR 1973 AND BIBSA 1975).
```

```
A  SOBEL, I.E. (1970)
DI   #CAMERA MODELS AND MACHINE PERCEPTION#
S    STANFORD UNIV, EE, PH D, AIM-121.

A  WALTZ, D.  (1972)
ME   #SHEDDING LIGHT ON SHADOWS#
S    MIT, AI LAB, VF 29.

A  WALTZ, D.L. (1972)
DI   #GENERATING SEMANTIC DESCRIPTIONS FROM DRAWINGS OF SCENES WITH
     SHADOWS#
S    MIT, EE, PH D, AI LAB TR-271 ( IN WINSTON, BIBTR 1973, BIBSA 1975)

A  WEIMAN, C.F.R. AND ROTHSTEIN, J. (1972)
TR   #PATTERN RECOGNITION BY RETINA-LIKE DEVICES#
S    OHIO STATE UNIV, COLUMBUS, COMPUTER AND INFO SCI DEPT, PH D,
     OSU-CISRC-TR-72-8 (AD 214 665/2).

A  WINSTON, P. H.   (1970)
DI   #LEARNING STRUCTURAL DESCRIPTIONS FROM EXAMPLES#
S    MIT, EE, PH D, MAC-TR-76 ( IN WINSTON, BIBTR 1973 AND BIBBSA 1975)

A  WINSTON, P.H.   (ED)  (1973)
TR   #PROGRESS IN VISION AND ROBOTICS#
S    MIT, AI LAB, TR-281.

A  YAKIMOVSKY, Y. (1973)
DI   #SCENE ANALYSIS USING A SEMANTIC BASE FOR REGION GROWING#
S    STANFORD UNIV, COMP SCI DEPT, PH D, AIM-209.

************************************************************
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NSO-9

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Scene Analysis: A Survey

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Carl Weiman

8. CONTRACT OR GRANT NUMBER(s)
N0C014-75-C-0571

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Courant Institute of Mathematical Sciences
New York University, 251 Mercer Street,
New York, N. Y. 10012

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Department of the Navy
Arlington, Virginia 22217

12. REPORT DATE
December 1975

13. NUMBER OF PAGES
62

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
scene analysis, computer vision, artificial intelligence, picture processing, bibliography

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This survey traces the development of scene analysis by computer from its origins in digitized picture processing and pattern recognition. Discussion of diverse approaches is unified by geometric concepts related to projection. A concluding overview contains suggestions for new approaches based on projective geometry and neurophysiological models. The bibliography, available as a deck of Hollerith cards, is organized according to topic and document accessibility.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# APPENDIX IV.2

## TWO ALGORITHMS FOR CONSTRUCTING A DELAUNAY TRIANGULATION

This appendix presents a comprehensive description of methods used to create a type of Voronoi tessellation known as Delaunay triangulation. This presentation unifies material from a variety of sources. The algorithms which are presented here provide for efficiency in representing, storing, and displaying ground surface coloration and texture.

GENERAL ELECTRIC

GSD
GROUND SYSTEMS DEPARTMENT
DAYTONA BEACH, FLORIDA

TIS CLASS 1

# TWO ALGORITHMS FOR CONSTRUCTING A DELAUNAY TRIANGULATION

79ASD007 © JULY 1979

GROUND SYSTEMS DEPARTMENT
SPACE DIVISION
DAYTONA BEACH, FLORIDA

General Electric Company
Space Division
TECHNICAL INFORMATION SERIES

## MINIMUM DISTRIBUTION

- Government Unclassified Material in General Electric Classes 1 and 2.

| COPIES | TO MANAGERS CF ENGINEERING |
|---|---|
| 1 | Re-entry Systems, Philadelphia, Pennsylvania |
| 1 | Research and Engineering Space Systems Organization, Valley Forge Space Technology Center |
| | **OTHERS** |
| 1 | Electronics Library, Electronics Park, Syracuse, New York |
| 1 | Librarian, Research Laboratory, Schenectady, New York |
| 5* | Technical Data Center, Schenectady, New York |
| 1 | Librarian, Space and Re-entry Systems Division, Philadelphia, Pennsylvania |
| 2 | Library, Daytona Beach, Florida |

*Plus 6 title pages.

- For Classes 3 and 4, the Technical Data Center, Schenectady, will receive one copy plus six title pages. Any additional distribution will be controlled by the author and the Library.

- Government Classified Material, Secret, or Confidential in General Electric Classes 1, 2, 3, or 4 will be the following, title page only:

| | |
|---|---|
| 1 | Classified Section, Electronics Library, Electronics Park, Syracuse, New York |

J. Blackburn
T. Brewer
W.M. Bunker
W. Diszenza
S. Dix - Wichita
B.F. Fishman
M.G. Gilliland
J. Hartigan
R.A. Heartz
J. Lambert
D.T. Lee (5) - Northwestern Univ.
(c/o B.J. Schachter)

J. Mundy - Schnectady
T.L. Neimeyer
R. Pester
R.H. Quist - Valley Forge
B.J. Schachter (10)
R. Schroth
B.J. Shinn
S. Simons
E. Sims
R.E. Stearns - Schnectady
J.T. Van Loo

# Two Algorithms for Constructing a Delaunay Triangulation

By

D. T. Lee* and B. J. Schachter**

CURRENT ADDRESSES

* Northwestern University
Dept. of Electrical Eng. and Computer Science
Evanston, Illinois 60201

** General Electric Co.
P.O. Box 2500
Daytona Beach, Florida 32015

This work was supported in part by the National Science Foundation under grant MCS-76-17321 and the Joint Services Electronics Program under contract DAAB-07-72-0259.

© D.T. Lee and B.J. Schachter June 1978, revised July 1979

GENERAL ELECTRIC

# TECHNICAL INFORMATION SERIES

Title Page

| | | |
|---|---|---|
| AUTHORS<br>D.T. Lee<br>B.J. Schachter | SUBJECT<br>Triangulation | NO. 79ASD007<br>DATE July, 1979 |
| TITLE<br>Two Algorithms for Constructing a Delaunay Triangulation | | G. E. CLASS 1<br>GOVT. CLASS |
| REPRODUCIBLE COPY FILED AT Technical Publications<br>P. O. Box 2500, Daytona Beach, Florida 32015 | | NO. PAGES 41 |

SUMMARY

This paper provides a unified discussion of the Delaunay triangulation. Its geometric properties are reviewed and several applications are discussed. Two algorithms are presented for constructing the triangulation over a planar set of N points. The first algorithm uses a divide-and-conquer approach. It runs in O(NlogN) time, which is asymptotically optimal. The second algorithm is iterative and requires $O(N^2)$ time in the worst case. However, its average case performance is comparable to that of the first agorithm.

KEY WORDS

Delaunay triangulation, triangulation, divide-and-conquer, Voronoi tessellation, computational geometry

INFORMATION PREPARED FOR __________

TESTS MADE BY __________

AUTHOR __________

COMPONENT __________

APPROVED __________

## CLASSES OF GENERAL ELECTRIC TECHNICAL REPORTS

### CLASS 1—GENERAL INFORMATION

Available to anyone on request. Patent, legal, and commercial review required before issuance.

### CLASS 2—GENERAL COMPANY INFORMATION

Available to any General Electric employee on request. Available to any General Electric Subsidiary or Licensee, subject to existing agreements. Disclosure outside General Electric Company requires approval of originating component and patent, legal, and commercial review before release. T.I.C. will monitor secondary distribution.

### CLASS 3—LIMITED AVAILABILITY INFORMATION

Original distribution to those individuals within General Electric with specific need for the information. Disclosure outside General Electric Company requires approval of originating component and patent, legal, and commercial review before release. Subsequent Company Availability monitored and controlled by the Technical Information Center.

### CLASS 4—HIGHLY RESTRICTED DISTRIBUTION

Original distribution to those individuals personally responsible for Company's interests in the subject. Material content and knowledge of existence restricted to copy holders. Subsequent Company availability requires approval of originator or technical manager of originating component. Report copies serially numbered, assigned and recorded by name. T. I. C. will monitor and control secondary distribution except for serially numbered reports. Responsibility for the latter will remain with the originator.

# TABLE OF CONTENTS

# 1. INTRODUCTION

In this paper we consider the problem of triangulating a set of points in the plane. Let V be a set of $N \geq 3$ distinct points in the Euclidean plane. We assume that these points are not all colinear. Let E be the set of $\binom{N}{2}$ straight line segments (edges) between vertices in V. Two edges $e_1, e_2 \in E$, $e_1 \neq e_2$, will be said to properly intersect if they intersect at a point other than their endpoints. A triangulation of V is a planar straight-line graph G(V,E') for which E' is a maximal subset of E such that no two edges of E' properly intersect [16].

There is no conceptual difficulty involved in constructing a triangulation. Any set of points can be triangulated if edges are added with the proviso that no new edge intersects an existing edge. We will investigate a particular triangulation called the Delaunay triangulation [3]. It has the property that the circumcircle of any triangle in the triangulation contains no point of V in its interior.

This paper is the result of a recent study whose objective was to develop an efficient algorithm for fitting triangular faceted surfaces to digital terrain data. A piecewise planar surface is used as a terrain model by all visual flight simulators. It was concluded that the Delaunay triangulation is an excellant choice for this application, based on the initial objectives of minimizing computation time and producing a good visual display.

In the next section, we will formally define the Delaunay triangulation and review its properties. Then in Section 3, we will provide two algorithms for its construction. Section 4 will cover some applications of the triangulation.

## 2. Definition and Properties of the Delaunay Triangulation

Suppose that we are given a set $V=\{v_1,\ldots,v_N\}, N\geq 3$, of points in the Euclidean plane. Assume that these points are not all colinear, and that no four points are co-circular. Let $d(v_i,v_j)$ denote the Euclidean distance between points $v_i$ and $v_j$. The bisector $B(v_i,v_j)$ of vertices $v_i$ and $v_j$ is the locus of points equidistant from them; i.e. $B(v_i,v_j)=\{x\varepsilon E^2|d(x,v_i)=d(x,v_j)\}$. The half-plane $H(v_i,v_j)$ is the locus of points closer to $v_i$ than to any $v_j$, $j\neq i$; $H(v_i,v_j)=\{x\varepsilon E^2|d(x,v_i)\leq d(x,v_j);\ j\neq i\}$. The polygonal region $VD(v_i)=\bigcap_{j\neq i}^{N} H(v_i,v_j)$ is the locus of points closer to vertex $v_i$ than to any other vertex. Each vertex $v_i$ therefore defines a region $VD(v_i)$ called the Voronoi [37] (Dirichlet, Wigner-Seitz , Theissen [36], or "S" [24]) polygon associated with the vertex. The collection of these N Voronoi polygons is referred to as the Voronoi diagram VD(V) of the set V of points [28,32].

Voronoi polygons may be thought of as the cells of a growth process. Suppose that we let each vertex in V be the nucleus of a growing cell. Cells will propagate outward from their nuclei, simultaneously and at a uniform rate. The border of a growing cell will freeze in place at its points of contact with the border of another growing cell.

Eventually, only the cells whose nuclei are on the convex hull of V are still expanding. The remaining cells have completely partitioned (or tessellated) a region of the plane into a set of non-overlapping closed convex polygons, one polygon about each nucleus. These closed polygons, together with the open polygons on the convex hull define a Voronoi tessellation of the entire plane.

Let us take a closer look at this process. Since all cells expand at the same rate, the first point of contact between two cells must occur at the midpoint between their nuclei. Likewise, every point of continuing contact must be equidistant from the two nuclei. These points are on the commom edge (called a *Voronoi edge*) of two developing Voronoi polygons. This edge continues elongating until it encounters the border of a third expanding cell. The point of contact (called a *Voronoi point*) of this edge and the border of the third cell must be equidistant from the growth centers of all three cells. It is therefore the circumcenter of the triangle defined by the three nuclei.

Voronoi cells which share a common edge are called *Voronoi neighbors*. The aggregate of triangles formed by connecting the growth centers of all Voronoi neighbors tessellates the area within the convex hull of the point set. This tessellation is called the *Delaunay triangulation* DT(V) of V. An example of a Voronoi tessellation and its dual is shown in Figure 1.

Each Voronoi point corresponds to a triangle and each Voronoi edge to a Delaunay edge. Since the number of Voronoi points and edges are both $O(N)$[1], the number of Delaunay triangles and edges are $O(N)$. To be more precise we have the following.

[1] Note: We say that $g(n)=O(f(n))$ if $|g(n)|\leqslant cf(n)$ for some constant c and all sufficiently large n. We say that $g(n)=\Omega(f(n))$ if $|g(n)|\geqslant cf(n)$ for some constant $c>0$ and all sufficiently large n. We say that $g(n)=\theta(f(n))$ if both $g(n)=O(f(n))$ and $g(n)=\Omega(f(n))$.

Lemma 1 Given a set V of N points, any triangulation T(V) nas the same number of triangles, $N_t=2(N-1)-N_h$, and the same number of edges, $N_e=3(N-1)-N_h$, where $N_h$ is the number of points on the convex hull of V.

Proof By induction, see [11] for example.

Now we will state without proof some properties of the Delaunay triangulation DT(V).

Lemma 2 Given a set $V=\{v_1,...,v_N\}$ of points, any edge $(v_i,v_j)$ is a Delaunay edge of DT(V) if and only if there exists a point x such that the circle centered at x and passing through $v_i$ and $v_j$ does not contain in its interior any other point of V.

Corollary Given a set $V=\{v_1,...,v_N\}$ of points, the edge $(v_i,v_j)$ on the boundary of the convex hull of V is a Delaunay edge.

Lemma 3 Given a set $V=\{v_1,...,v_N\}$ of points, $\Delta v_i v_j v_k$ is a Delaunay triangle of DT(V) if and only if its circumcircle does not contain any other point of V in its interior.

The proofs of these lemmas can be found in [13, 32]. The latter property is called the circle criterion. It is often used as a rule for constructing a triangulation. Triangulations may also be constructed according to the MAX-MIN angle criterion, i.e. the minimum measure of angles of all the triangles in the triangulation is maximized. We shall investigate the relationship between these two criteria.

The following analysis follows that given by Lawson in [ 9 ] and [10] (see also Sibson [33]). Consider a very simple triangulation, that over the vertices of a strictly convex quadrilateral. A quadrilateral is called strictly convex if its four interior angles are each less than $180^0$. A quadrilateral can be partitioned into two triangles in two possible ways. Each of the criteria described above can be thought of as a rule for choosing a preferred triangulation.

By examining the case of four co-circular points (Fig. 2 ), one can show that the two criteria are equivalent. Suppose that for this example line segment $(v_2,v_3)$ is shorter than $(v_3,v_4)$, $(v_4,v_1)$ and $(v_1,v_2)$. Let the angular measure of the arc $v_2,v_3$ be $2\theta$. Then the angles $<v_3,v_1,v_2$ and $<v_3,v_4,v_2$ are each $\theta$. Thus the two possible triangulations over the four points have the same minimum angle. The choice of a preferred triangulation is then arbitrary according to the MAX-MIN angle criterion. The Voronoi tessellation of the quadrilateral also exhibits a tie case. All four Voronoi polygons meet at a single point.

Further analysis shows that moving one point, say point $v_4$ in Fig. 2 , inside the circle causes $<v_3,v_4,v_2$ to increase and points $v_2$ and $v_4$ to be the growth centers of Voronoi neighbors. Consequently, the two criteria and the Voronoi tessellation of the polygon all now prescribe the connection of vertices $v_2$ and $v_4$.

A fourth criteria has been studied, that of choosing the minimum length diagonal. Shamos and Hoey [32] claim that a Delaunay triangulation is a minimum edge length triangulation. Lawson [9 ] and Lloyd [16] prove by counterexample that this is not the case (Fig. 3).

Lawson [10] gives the following local optimization procedure (LOP) for constructing a triangulation. Let e be an internal edge (in contrast to an edge on the convex hull) of a triangulation and Q be the quadrilateral formed by the two triangles having e as their common edge. Consider the circumcircle of one of the triangles. This circle passes through three vertices of Q. If the fourth vertex of the quadrilateral is within the circle, replace e by the other diagonal of Q, otherwise no action is taken. An edge of the triangulation is said to be locally optimal if an application of the LOP would not swap it. Since for any set of vertices, the number of triangles in any triangulation is a constant, a linear ordering over the set of all triangles can be defined as follows. To each triangle in the triangulation we assign a value, which is the measure of its minimum angle. Let $N_t$ denote the number of triangles. For each triangulation we have an indicator vector of $N_t$ components, each component corresponding to the minimum angle of its corresponding triangle. Triangles are sorted in nondecreasing order. Given any two triangulations T and T', define T<T' if and only if the associated indicator vector of T is lexicographically less than that of T'.

Theorem 1 [10,13] Given a triangulation T, if an application of the LOP to an edge e results in a swapping of the edge with any other edge e', thus producing a new triangulation T', then T<T'; i.e. T' strictly follows T in the linear ordering defined above.

Proof Let I be an indicator vector for T. The measures of the smallest angles in the two triangles of T sharing the edge e occur as two of the components of I, say $I_j$ and $I_k$ with j<k and thus $I_j \leq I_k$. Since a swap

was made when applying the LOP, the smallest angle in both of the two new triangles of T' sharing the edge e' must be strictly greater than $I_j$. It follows that the indicator vector I for T is lexicographically less than the indicator I' for T' and thus T<T'.

Theorem 2 [10,13] All internal edges of a triangulation T of a finite set V are locally optimal if and only if no point of V is interior to any circumcircle of a triangle of T.

Proof If no point of V is interior to the circumcircle of any triangle of T, then the application of LOP to any edge will not swap it. Thus all edges are locally optimal. If all edges are locally optimal, then we show that no point of V is interior to the circumcircle of any triangle. Suppose that the circumcircle K of a triangle $\Delta abc$ contains a point v of V. Let $\delta$ be the distance of v to its nearest edge, say (a,c) as shown in Fig. 4. Assume that among all triangles of T whose circumcircles contain v as an interior point, none has an edge which is at a distance less than $\delta$ from v. Since v is on the opposite side of (a,c) from b, the edge (a,c) is not a boundary edge of T. Thus, there is another triangle $\Delta acq$ sharing an edge with triangle $\Delta abc$. The vertex q cannot be interior to the circle K as this would contradict the hypothesis that edge (a,c) is locally optimal. The vertex q cannot be in the cross-hatched region of the diagram, or $\Delta acq$ will contain v in its interior. Suppose that edge (c,q) is the nearest edge of $\Delta acq$ to v. Note that the distance from (c,q) to v is less than $\delta$. Since the circumcircle of $\Delta acq$ also contains v in its interior, we have a contradiction that $\Delta abc$ is the triangle with an edge at the smallest distance from v.

By the above theorem, the edges of the Delaunay triangulation of a finite set V of points are locally optimal. If we assume that no more than three points are co-circular, the uniqueness of the Delaunay triangulation [33] suggests the following theorem.

Theorem 3 A triangulation T(V) is a Delaunay triangulation if and only if its indicator vector is lexicographically maximum, i.e. no triangulation follows it in the linear ordering.

Proof If the triangulation T is lexicographically maximum, then all the edges of T must be locally optimal, which implies that no circumcircle of any triangle will contain any other point of V in its interior (Theorem 2). Thus, T is the Delaunay triangulation DT(V). To prove the converse, suppose that the Delaunay triangulation is not maximum in the linear ordering. That is, there exists another triangulation T(V), such that $DT(V)<T(V)$. Repeatedly applying the LOP to T(V) will produce a triangulation T'(V) in which all the edges are locally optimal. Since $DT(V)<T(V)<T'(V)$, T'(V) would also be a Delaunay triangulation by Theorem 2 and Lemma 3. However, since the Delaunay triangulation is unique, $T'(V)=DT(V)$, a contradiction.

Corollary 1 The Delaunay triangulation of a set of points satisfies the MAX-MIN angle criterion. (Note that a triangulation which satisfies the MAX-MIN angle criterion is not necessarily a Delaunay triangulation.)[2]

We have shown that the Delaunay triangulation of a set of points is a maximum in the linear ordering over the set of possible triangulations. Now we are ready to describe two algorithms for its construction.

[2] Consider a triangulation T whose indicator vector is $(i_1,\ldots,i_t)$ where $i_1$ is the minimum angle. Suppose that $\Delta abc$ is the triangle with the smallest angle. If applying LOP to the edges of ab,bc,ca will not result in a swap, then T satisfies the MAX-MIN angle criterion. Since we have only checked one triangle in the triangulation, we cannot say that T is a Delaunay triangulation. However, if we apply the LOP to all edges of T, until no more swapping occurs, the resulting triangulation will be Delaunay.

## 3. TECHNIQUES FOR CONSTRUCTING THE DELAUNAY TRIANGULATION

We will present two algorithms for constructing a Delaunay triangulation. The first is rather involved, but its running time is asymptotically optimal. The second algorithm is simpler to understand, simpler to program, and requires less overhead. These features make it especially attractive for small and medium size data sets (≈500 points or less).

The first algorithm is comparable to the one given by Lewis and Robinson [15] in that it divides the original data set into disjoint subsets. After obtaining a solution for each of these subsets, it combines solutions to yield the final result. In [15], Lewis and Robinson claim, without proof, that their algorithm runs in $O(N \log N)$. But in fact[3], the running time of their algorithm is $O(N^2)$. The second algorithm that we will present is iterative. It follows the idea proposed by Lawson [9]. Nelson [23] gives a similar method.

### 3.1 DIVIDE - AND - CONQUER

Shamos and Hoey [32] present an $\theta(N \log N)$ algorithm for constructing the Voronoi diagram for a set of N points in the plane. Green and Sibson [7] also implement an algorithm for this purpose, but the running time is $O(N^2)$ in the worst case. Lee [14] modifies the procedure given by Shamos and Hoey and extends the method to any $L_p$ metric, $1 \leqslant p \leqslant \infty$.

Once the Voronoi diagram is obtained, its dual graph (i.e. the Delaunay triangulation) can be constructed in an additional $O(N)$ time. To eliminate the need for a two step procedure, we have developed

[3] There are at least four triangulation algorithms in the computer literature which claim to be $O(N \log N)$, but which are in fact $O(N^2)$. A counter-example for several of these is given in the Appendix.

the following algorithm which constructs a triangulation directly. The running time of the algorithm is shown to be O(NlogN). Shamos and Hoey [32] show that the construction of any triangulation over N points requires $\Omega$(NlogN) time. Thus, our algorithm is optimal.

We will begin by describing the data structures and notation to be used in the sequel. For each point $v_i$, we keep an ordered adjacency list of points $v_{i1},\ldots,v_{ik}$, where $(v_i,v_{ij})$, j=1,...k, is a Delaunay edge. The list is doubly-linked and circular. PRED$(v_i,v_{ij})$ denotes the point $v_{ip}$ which appears clockwise (CW) of and immediately after the point $v_{ij}$. The counter-clockwise function SUCC operates in a similar manner. For example in Fig. 5 , $v_5$=PRED$(v_1,v_6)$ and $v_5$=SUCC$(v_1,v_4)$.

If the point $v_i$ is on the convex hull CH(V), then the first entry on its adjacency list is the point denoted FIRST$(v_i)$, which appears after $v_i$ if we traverse the boundary of CH(V) in a CCW direction. Let $\ell(v_i,v_j)$ denote the line segment directed from $v_i$ to $v_j$.

Now we are ready to construct the Delaunay triangulation. First, we sort the given set V of N points in lexicographically ascending order and rename the indices so that $v_1<v_2<\ldots<v_N$, such that $(x_i,y_i)=v_i<v_j$ if and only if either $x_i<x_j$ or $x_i=x_j$ and $y_i<y_j$. Next we divide V into two subsets $V_L$ and $V_R$, such that $V_L=\{V_1,\ldots,v_{\lfloor N/2 \rfloor}\}$ and $V_R=\{v_{\lfloor N/2 \rfloor+1},\ldots,v_N\}$. Now we recursively construct the Delaunay triangulations DT$(V_L)$ and DT$(V_R)$. To merge DT$(V_L)$ and DT$(V_R)$ to form DT$(V_L \cup V_R)$, we make use of the convex hull CH$(V_L \cup V_R)$. The convex hull can be obtained in O(N) time [ 26 ] from the union of the convex hulls CH$(V_L)$ and CH$(V_R)$. The convex hulls can also be computed recursively. Forming the union of CH$(V_L)$ and CH$(V_R)$ will result in

two new hull edges which are the lower and upper common tangents of CH($V_L$) and CH($V_R$). These two common tangents are known to be in the final Delaunay triangulation.

The following subroutine finds the lower common tangent of CH($V_L$) and CH($V_R$). For each convex hull CH(S), we maintain two points LM(S) and RM(S), which are the leftmost and rightmost points of S, respectively.

## SUBROUTINE HULL

```
(Comment:  HULL is input two convex hulls.  It finds their lower
           common tangent (Fig. 6).  The upper common tangent can
           be found in an analagous manner.)

(Comment:  $\ell(X,Y)$ denotes the line directed from X to Y.)

X←RM($V_L$); Y←LM($V_R$)

Z←FIRST(Y); Z'←FIRST(X); Z"←PRED(X,Z')

A:     IF (Z is-right-of $\ell(X,Y)$)
       Z←SUCC(Z,Y)
       Y←Z
       ELSE
              IF (Z" is-right-of $\ell(X,Y)$
              Z"←PRED(Z",X)
              X←Z"
              ELSE
              RETURN (X,Y)
              ENDIF
       ENDIF
       GO TO A
END HULL
```

The lower common tangent will be used as an input to the following subroutine which merges the two triangulations DT($V_L$) and DT($V_R$).

## SUBROUTINE MERGE

(Comment: MERGE is input two triangulations and the upper and lower common tangents of their convex hulls. It merges the two triangulations, starting with the lower common tangent, zig-zagging upward until the upper common tangent is reached.)

(Comment: Initially, the points adjacent to the endpoints of the lower common tangent are examined. Using the circle criterion, we either connect:

(i) the left endpoint (in $V_L$) of the lower common tangent to a point adjacent to the right endpoint (in $V_R$) of the lower common tangent, or

(ii) the right endpoint (in $V_R$) of the lower common tangent to a point adjacent to the left endpoint (in $V_L$) of the lower common tangent.

The above process is repeated with the newly found edge taking the place of the lower common tangent, and each succeeding new edge taking the place of that. The subroutine continues in this manner until the upper common tangent is reached.)

(Comment: The adjacency list of the right (left) endpoint in $V_R$($V_L$) of the current edge being considered is examined in a CW (CCW) direction starting with the left (right) endpoint of the edge.)

(Comment: INSERT(A,B) inserts point A into the adjacency list of B and point B into the adjacency list of A at proper positions. DELETE(A,B) deletes A from the adjacency list of B and B from the adjacency list of A.)

(Comment: QTEST(H,I,J,K) tests the quadrilateral having CCW ordered vertices H,I,J,k. It returns TRUE if the circumc rcle of $\triangle HIJ$ does not contain K in its interior, and returnsFALSE otherwise.)

```
step 1:  INITIALIZATION
         BT←lower common tangent
         UT←upper common tangent
          L←left endpoint of BT
          R←right endpoint of BT
step 2:        DO UNTIL (BT equals UT)
step 3:        A←B←FALSE
step 4:        INSERT(L,R)
step 5:        $R_1$←PRED(R,L)
step 6:              IF ($R_1$ is-left-of $\ell$(L,R))
step 7:              $R_2$←PRED(R,$R_1$)
step 8:                    DO UNTIL (QTEST($R_1$,L,R,$R_2$))
                           DELETE (R,$R_1$)
                           $R_1$←$R_2$
                           $R_2$←PRED(R,$R_1$)
                           END DO UNTIL
step 9:              ELSE
                     A ←TRUE
                     ENDIF
step 10:       $L_1$←SUCC(L,R)
step 11:             IF ($L_1$ is-right-of $\ell$(R,L)
step 12:             $L_2$←SUCC(L,$L_1$)
step 13:                   DO UNTIL (QTEST(L,R,$L_1$,$L_2$))
                           DELETE (L,$L_1$)
                           $L_1$←$L_2$
                           $L_2$←SUCC(L,$L_1$)
                           END DO UNTIL
step 14:             ELSE
                     B←TRUE
                     ENDIF
step 15:             IF (A)
                     L←$L_1$
step 16:             ELSE
step 17:                   IF (B)
                           R←$R_1$
step 18:                   ELSE
step 19:                         IF (QTEST(L,R,$R_1$,$L_1$))
                                 R←$R_1$
step 20:                         ELSE
                                 L←$L_1$
```

```
                                ENDIF
step 21:                    ENDIF
step 22:                ENDIF
step 23:            $BT \leftarrow \ell(L,R)$
                    END DO UNTIL
step 24:        END MERGE
```

To show that the above algorithm merges two triangulations correctly, it is sufficient to show that each edge we insert into the triangulation is a Delaunay edge (step 4). Initially, the first edge (L,R) is a lower common tangent and is known to be a Delaunay edge. Steps 5-8 delete the edges in $DT(V_R)$ which are not Delaunay edges in $DT(V)$ by determining if L is within the circumcircle of $\Delta R,R_1,R_2$. If so, the edge $(R,R_1)$ is not a Delaunay edge and must be deleted. Steps 10-13 perform the same operation on the edges in $DT(V_L)$. An example of this process is shown in Fig. 7a. Now the circumcircle $K_R$ of $\Delta L,R,R_1$ does not contain any point of $V_R$ in its interior and the circumcircle $K_L$ of $\Delta R,L,L_1$ does not contain any point of $V_L$ in its interior. Now as shown in Fig. 7b, we have a choice of either connecting $L_1$ to R or $R_1$ to L. Step 19 chooses the correct edge by applying the circle criterion. In Fig. 7 , $K_L$ contains $R_1$ in its interior, so we choose the edge $(L,R_1)$. All that we have left to do is to show that the edge $(L,R_1)$ is a Delaunay edge, or equivalently that the circle $K_R$ does not contain any point of $V_L$ in its interior. Since the edge (L,R) is known to be a Delaunay edge, by Lemma 2 there exists a circle K passing through L and R which contains no point of V in its interior. It is also known that the circle $K_L$ contains no point of $V_L$ in its interior. Since the circle $K_R$ lies inside the union of K and $K_L$, it follows that $K_R$ contains no point of $V_L$ in its interior. We combine this result with the fact that $K_R$ does not contain any point of $V_R$ in its interior to conclude that $K_R$ contains no point of V in its interior. Thus, the edge $(L,R_1)$ is a Delaunay edge. The edge $(L,R_1)$ can now be inserted to replace the edge (L,R). In showing that the next edge to be added after $(L,R_1)$ is a Delaunay edge, the circle $K_R$ plays the same role as the circle K just did.

Not let us analyze the algorithm MERGE. We first note that during the merge process, once an edge is deleted, it will never be re-examined. Since the total number of edges deleted is O(N) and the total number of edges added is also O(N), the time needed for the merge is O(N). Let t(N) denote the time required to construct the Delaunay triangulation for a set of N points. We have the following recurrence relation

$$t(N) = 2t(N/2) + M(N/2,N/2)$$
$$t(1) = 0,$$

where M represents the time required for the merge process. Since $M(N/2,N/2)=O(N)$, $t(N)=O(N\log N)$.

## 3.2 Iteration

The algorithm presented in this section iteratively triangulates a set of points within a rectangular region. If the point set does not include all four vertices of the rectangle, the missing vertices are implicitly added. The algorithm uses the swapping approach developed by Lawson in [9] and [10]. Since the convex hull of our point set is a rectangle and is known in advance, we need not compute it, as is done in Lawson's algorithm. The initial ordering of the point set also differs from that used by Lawson.

This algorithm was developed with the terrain fitting problem in mind. Terrain regions are processed in rectangular blocks. Adjacent terrain regions must fit together without any gaps. Once a triangular facetedsurface is fit to a terrain region, the accuracy of the fit is usually computed. If the approximation surface does not meet the given accuracy constraints, additional vertices are added and the triangulation is updated. An iterative triangulation algorithm is ideal for updating.

### Algorithm SWAP

#### INITIALIZATION

step 1: Given a set V of N points within a rectangle, remove any points which fall on the vertices of the rectangle.

step 2: Partition the rectangle into approximately $N^{1/2}$ bins (smaller rectangular regions).

step 3: Re-order the points by bins, starting at some bin and proceeding to neighboring bins (see Fig. 8 ).

step 4: Place the first point into the rectangle. Connect the point to the four corners of the rectangle to produce an initial triangulation.

ITERATION

(Comment: The remaining points in V will be iteratively added to the triangulation. After each point is added, it will be connected to the vertices of its enclosing triangle. The triangulation will tnen be re-structured so that the MAX-MIN angle criterion is globally satisfied. [See Fig. 9.] )

step 1: Input the next point to the existing triangulation. Connect this point to the vertices of its enclosing triangle.

step 2: Step 1 will produce up to four strictly convex quadrilaterals. (Four quadrilaterals only occur when a newly introduced point falls on the edge of the triangulation.) Each of these quadrilaterals has an alternate diagonal. Swap a diagonal with its alternate, if doing so is required to satisfy the MAX-MIN angle criterion within the quadrilateral (i.e. use the LOP within the quadrilateral).

step 3: Each swap performed in step 2 may result in two new quadrilaterals that need to be tested. If one of these quadrilaterals doesn't satisfy the MAX-MIN angle criterion, swap its diagonal with its alternate.

step 4: This swapping procedure may propagate further outward. Lawson [10] has shown that this process will always terminate.

step 5: If all points in V have been used then stop, otherwise go to step 1.

END SWAP

Step 1 requires the identification of the enclosing triangle of a point. This can be accomplished by the following very simple subroutine [ 10]. The subroutine assumes that the triangulation is stored using a variation of Lawson's data structure (given in the Appendix).

## SUBROUTINE TRIFIND

(Comment: The triangulation is stored in the form given in Appendix 2.This subroutine locates the triangle $\tau$ which encloses the point $(x_0,y_0)$.)

(Comment: $X(\tau,i)$ denotes the x value of the i-th vertex of triangle $\tau$.)

```
       τ←last triangle ι.ɜated
LOOP:  DO FOR I←1 to 3
       IPLUS1←I(mod 3)+1
            IF [(y0-Y(τ,I))*(X(τ,IPLUS1)-x0) .GT. (x0-X(τ,I))*(Y(τ,IPLUS1)-y0)]
            Comment: If (x0,y0) is not in τ, jump to the neighbor of τ which
            is in the direction of the point.
            τ←N(τ,IPLUS1)
            GO TO LOOP
            ENDIF
       END DO FOR
       RETURN (τ)
       END TRIFIND
```

Each iteration of algorithm SWAP requires an O(N) searcn performed by TRIFIND, followed by an O(N) swapping procedure. Thus, the algorithm is $O(N^2)$.

An empirical examination of algorithm SWAP yields somewhat better results. If the initial point set is uniformly distributed within its enclosing rectangle, then the number of data points in each bin will be approximately $O(N^{\frac{1}{2}})$. Thus the search procedure will be $O(N^{\frac{1}{2}})$. The swapping procedure which updates the triangulation can be propagated many times. We have found that two levels of swaos are nearly always sufficient to globally satisfy the MAX-MIN angle criterion. Thus the algorithm is roughly $O(N^{3/2})$, empirically.

## 4. Examples and Applications

### (1) Random Delaunay Triangulation

A two-dimensional Poisson process of intensity $\lambda$ can be used to describe a random distribution of points in the plane. This process is characterized by the property that the expected number of points within a region of area A is $\lambda A$, irrespective of the shape or orientation of the region.

Suppose we let the points chosen by a Poisson process seed a Delaunay triangulation. The resulting network of triangles inherits the properties of homogeneity and isotropy from the driving point process.

A random Delaunay triangulation is probably the only non-regular triangulation which is mathematically tractable. Many of its statistical properties have been derived by Miles [20]. Its principal first order statistics are given below, the paper by Miles also provides the associated second order statistics.

$$\begin{aligned} &E[\text{cell area}] = \pi/\lambda \\ &E[\text{cell circumference}] = 32/[3\pi(\lambda/2\pi)^{\frac{1}{2}}] \\ &E[\text{cell in-radius}] = (8\lambda/\pi)^{-\frac{1}{2}} \end{aligned}$$

Miles has also derived the probability density function $f(\alpha)$ for an arbitrary angle $\alpha$ in the triangulation.

$$f(\alpha) = 4\{(\pi-\alpha)\cos\alpha+\sin\alpha\}\frac{\sin\alpha}{\alpha}$$

For certain applications, we would like a triangulation with as few small angles as possible. The distribution $f(\alpha)$ provides a characterization of the "goodness" of a triangulation.

### (2) Interpolating Functions of Two Variables

A major application of triangulations is the interpolation of functions of two variables, where the function is initially defined only at an irregular set of locations. These locations are used as the vertices of a triangulation. The value of the function at a point, other than a vertex, is computed by performing an interpolation within the triangle containing the point. A triangulation composed of nearly equiangular triangles is considered good for this purpose. McLain [19] and Lawson [10] have used the Delaunay triangulation for this purpose. Also see Powell and Sabin [25].

### (3) Decomposition of Polygons into Convex Sets

An algorithm for decomposing polygons into triangles may be based upon the concept of the Delaunay triangulation. There are applications in pattern recognition and computer graphics for which one wants to combine adjacent triangles into larger convex sets. An O(mN) algorithm for decomposing a non-convex polygon of N sides and m reflex angles into convex sets is given by Schachter [29]. (An O(NlogN) polygon triangulation algorithm not based upon the Delaunay triangulation is given by Garey et al. [5].)

### (4) Terrain Fitting

A rectangular terrain region may be represented by an array of elevation values. A two-dimensional digital filter (e.g. a Wiener filter) can be applied to this data to extract local extrema (i.e.

peaks of mountains and "pits" of valleys) and ridge line segments.

We would like the local extrema to be the vertices of a triangulation and the ridge line segments to fall on the edges of the triangulation. This structure can be obtained as follows. Let L denote the set of local extrema and E the set of endpoints of the ridge segments. For each element of LUE falling within the smallest circumscribing circle of a ridge segment, construct a normal projection onto the segment. Let P denote the projection point set. Now, let the points in LUEUP seed a Delaunay triangulation. Each element of LUEUP has an associated elevation. The triangulation therefore defines a piece-wise planar approximation to the terrain surface. For certain applications, an approximated surface must fit the original data to a given error tolerance. If this error bound is not satisfied, additional vertices are added, and the triangulation is updated. An iterative algorithm is well suited for updating.

The above technique assumes that the line segment set is sparse in the plane. A good solution for the more general problem of triangulating any planar set of points and line segments is given by Lee [13].

A Delaunay triangulation is an excellent choice for the terrain fitting and display problem. Triangles with very small angles produce a poor computer graphics display. Maximizing the minimum angle within the triangulation insures the best possible visualization of the data. Further requirements are met concerning speed of construction.

## (5) SPATIAL PATTERN MODELS

The Voronoi and Delaunay tessellations have been extensively used to model spatial patterns in a wide range of fields including astronomy [3], bio-mathematics [1,18,24,35], computer science [4,11,12,30-32], geography [2,17,27], meteorology [36], metalurgy [6], numerical analysis [10,24,25], and packing and covering [22,28]. We will briefly consider two somewhat representative examples.

Suppose that we are given a collection of sites, where each site has a random variable associated with it. Let these sites seed a Voronoi tessellation. The statistical dependence between (Voronoi) neighboring sites may be specified in terms of the Delaunay edge length between sites and the Voronoi border length between their cells. See Besag [ 1] for details and references.

Stearns [34] poses the following problem:

"A domain wall in ferromagnetic materials can be considered as a two-dimensional membrane which, when subject to an r.f. field, will oscillate in a manner determined by the boundary conditions. One possible set of boundary conditions would correspond to pinning the wall at impurities whose positions are random in the wall. In describing wall motion, we must know the area distributions of triangles formed from three impurity sites. These triangles will contain no other impurity pinning points in their interior and will be called 'good' triangles. What is the probability distribution of the areas of the resulting network of 'good' triangles formed by choosing N points distributed uniformly in a given area?"
Miles [21] interprets 'good' to mean Delaunay, and proposes a solution.

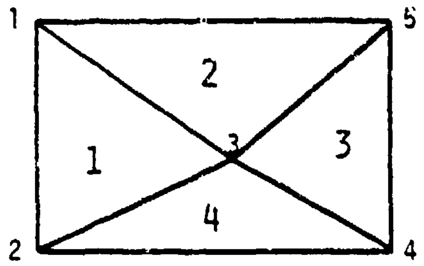
## 5. Discussion

We have presented two algorithms for constructing the Delaunay triangulation for a set of N points in the Euclidean plane. The first algorithm is based upon a divide-and-conquer approach. It runs in O(NlogN) time, which is asymptotically optimal. The second algorithm iteratively adds points to an existing triangulation, updating the triangulation to include each newly introduced point as a vertex. Although it could take $O(N^2)$ time for a worst case, it runs fairly well for the average case.

ACKNOWLEDGEMENTS: We would like to thank S. Dix, C. L. Lawson, D. Milgram, R. E. Stearns, and one of the reviewers for their thoughtful comments on this paper.

## APPENDIX 1: Data Structure for a Triangular Network

The data structure used by the iterative algorithm will be described by an example.



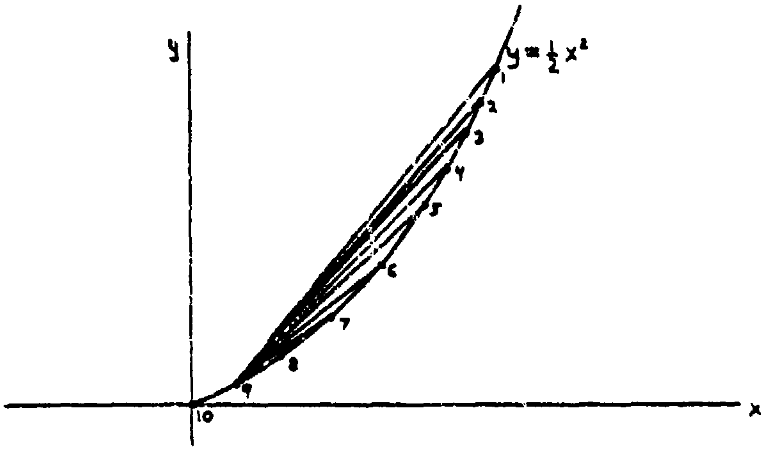
| vertex<br>T | X<br>T | Y<br>T |
|---|---|---|
| 2 | 1 | 31 |
| 3 | 18 | 17 |
| 4 | 31 | 31 |
| 5 | 31 | 1 |

| triangle | neighboring triangles (counter-clockwise) | | | vertices (counter-clockwise) | | |
|---|---|---|---|---|---|---|
| T | N(T,1) | N(T,2) | N(T,3) | V(T,1) | V(T,2) | V(T,3) |
| 1 | 2 | 0 | 4 | 1 | 2 | 3 |
| 2 | 1 | 3 | 0 | 3 | 5 | 1 |
| 3 | 4 | 0 | 2 | 4 | 5 | 3 |
| 4 | 1 | 0 | 3 | 2 | 4 | 3 |

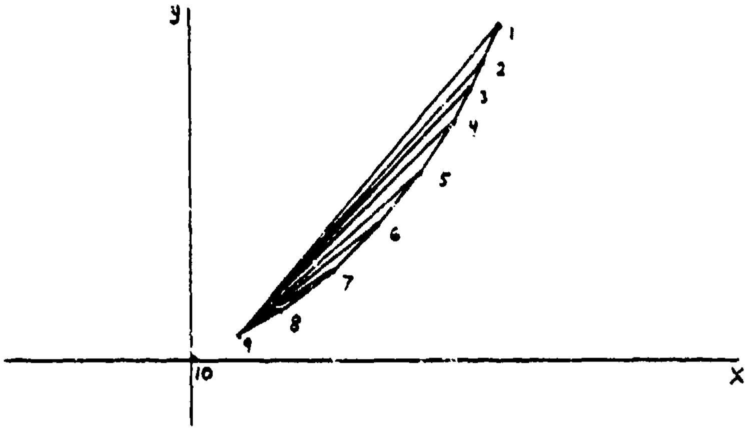
The following conventions are used: Triangles N(T,1), N(T,2) and T meet at vertex V(T,1). Zero denotes a null triangle.

## APPENDIX 2: An Example for Which Iterative Algorithms Work in $O(N^2)$ Worst Case Time.

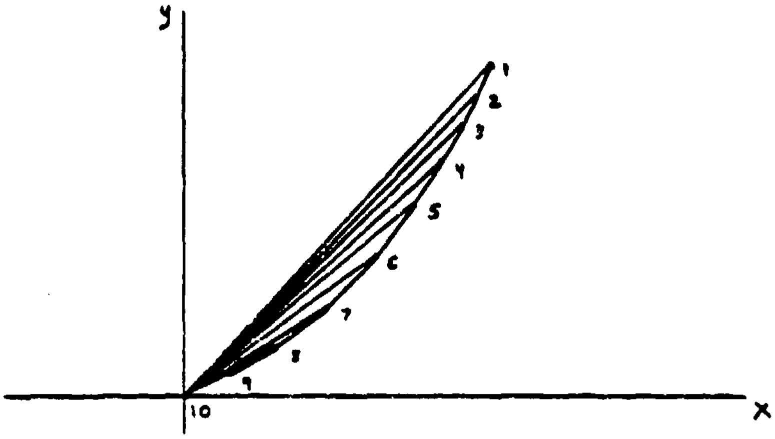
Consider 10 points on the parabola $y=\frac{1}{2}x^2$. The points in the diagram are numbered in the order in which they are added to the existing set.



Let S={1,2,...,9}. DT(S) is given below.



Now when point 10 is added, all the edges incident with point 9 are deleted. The resultant triangulation is given below.


y
1
2
3
4
5
6
7
8
9
10
x

## REFERENCES

1. J. Besag, Spatial interaction and the statistical analysis of lattice systems, *J. Royal Stat. Soc. B 36(2)*, 192-236, 1974.
2. D. J. Bogue, The structure of the metropolitan community, University of Michigan Contributions of the Inst. of Human Adjustment Social Science, U. of Mich., Ann Arbor, MI, 1949.
3. B. Delaunay, "Sur la sphere vide, *Bull. Acad. Science USSR (VII), Class. Sci. Mat. Nat.*, 793-800, 1934.
4. H. Fuchs and Z. M. Kedem, The "highly intelligent" tablet as an efficient printing device for interactive graphics, U. of Texas (Dallas), Program in Math Science, 1979.
5. M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan, Triangulating a simple polygon, *Info. Processing Letters 7*, 175-179, June 1978.
6. E. N. Gilbert, Random subdivision of space into crystals, *Ann. Math. Stat. 33*, 958-972, 1962.
7. P. J. Green and R. Sibson, Computing Dirichlet tesselations in the plane, *Comp. Journal 21(2)*, 168-173, July 1978.
8. T. Kiang, Random fragmentation in two and three dimensions, *Zeitschrift fur Astrophysik 64*, 433-439, 1966.
9. C. L. Lawson, Generation of a triangular grid with applications to contour plotting, Tech. Memo. 299, Jet Propulsion Lab., Pasadena, CA, Feb. 1972.
10. C. L. Lawson, Software for $C^1$ surface interpolation, In *Mathematical Software III*, J. Rice, Ed., Academic Press, N. Y., 1977.

11. D. T. Lee, On finding k-nearest neighbors in the plane, Tech. Report. Eng. 76-2216, Coordinated Science Lab., University of Illinois, Urbana, IL, 1976.

12. D. T. Lee and C. K. Wong, Voronoi diagrams in $L_1(L_\infty)$ metrics with two-dimensional storage applications, Research Report RC(6843) IBM T. J. Watson Research Center, Yorktown Heights, NY, 1977.

13. D. T. Lee, Proximity and reachability in the plane, PH. D. Thesis, Coordinated Science Lab. Rep. ACT-12, University of Illinois, Urbana, IL, 1978.

14. D. T. Lee, Two dimensional Voronoi diagrams in the $L_p$-metric, submitted for publication.

15. B. A. Lewis and J. S. Robinson, Triangulation of planar region with applications, Computer Journal 21(4), 324-332, Nov. 1978.

16. E. L. Lloyd, On triangulation of a set of points in the plane, Tech. Report MIT/LCS/TM-88, Lab. for Computer Science, Mass. Inst. of Tech., Cambridge, MA, May 1977.

17. P. E. Lloyd and P. Dicken, Location in Space: A Theoretical Approach to Economic Geography, Harper and Row, NY, 1972.

18. B. Matern, Spatial variation, Medd. Statens Skogsforskninginstit. 36(5), 1-144, 1960.

19. D. H. McLain, Two dimensional interpolation from random data, Computer Journal 19(2), 178-181, 1976. Errata in Computer Journal 19(4), 384, 1976.

20. R. E. Miles, On the homogeneous planar Poisson point-process, Math. Biosciences 6, 85-127, 1970.

21. R. E. Miles, Solution to problem 67-15 (Probability distribution of a network of triangles), SIAM Review 11, 399-402, 1969.

22. J. Molnar, Packing of congruent spheres in a strip, Acta Math. Acad. Sciences Hungaricae Jomus 31(1-2), 173-183, 1978.

23. J. M. Nelson, A triangulation algorithm for arbitrary planar domains, Appl. Math. Modelling 2, 151-159, Sept. 1978.

24. E. C. Pielou, Mathematical Ecology, Wiley, NY, 1977.

25. M. J. D. Powell and M. A. Sabin, Piecewise quadratic approximations on triangles, ACM TOMS 3(4), 316-325, Dec. 1977.

26. F. P. Preparata and S. J. Hong, Convex hulls of finite sets of points in two and three dimensions, Comm. Assoc. Comp. Mach. 20(2), 87-93, Feb. 1977.

27. D. Rhynsburger, Analytic delineation of Thiessen polygons, Geographical Analysis 5(2), 133-144, April 1973.

28. C. A. Rogers, Packing and Covering, Cambridge University Press, 1964.

29. B. J. Schachter, Decomposition of polygons into convex sets, IEEE Trans. Comp. C-72(11), 1078-1082, Nov. 1978.

30. B. Schachter, A. Rosenfeld, and L. S. Davis, Random mosaic models for textures, IEEE Trans. Systems, Man, and Cybernetics 8(9), 694-702, Sept. 1978.

31. M. I. Shamos, Computational Geometry, Springer-Verlag, NY, 1977.

32. M. I. Shamos and D. Hoey, Closest-point problems, Proc. 16th Annual Symp. on Foundations of Computer Science, 151-162, Oct. 1975.

33. R. Sibson, Locally equiangular triangulations, Computer Journal 21(3), 243-245, Aug. 1978.

34. B. Stearns, Probability distribution of a network of triangles (Problem 67-15), SIAM Review 11, 399, 1969.

35. P. Switzer, Reconstructing patterns from sampled data, Annals Math. Stat. 38, 138-154, 1967.

36. A. H. Thiessen, Precipitation averages for large areas, Monthly Weather Review 39, 1082-1084, 1911.

37. G. Voronoi, Nouvelles applications des parametres continus a la theorie des formes quadratiques. Deuxieme Memoire: Recherches sur les parallelloedres primitifs, J. Reine Angew. Math. 134, 198-287, 1908.

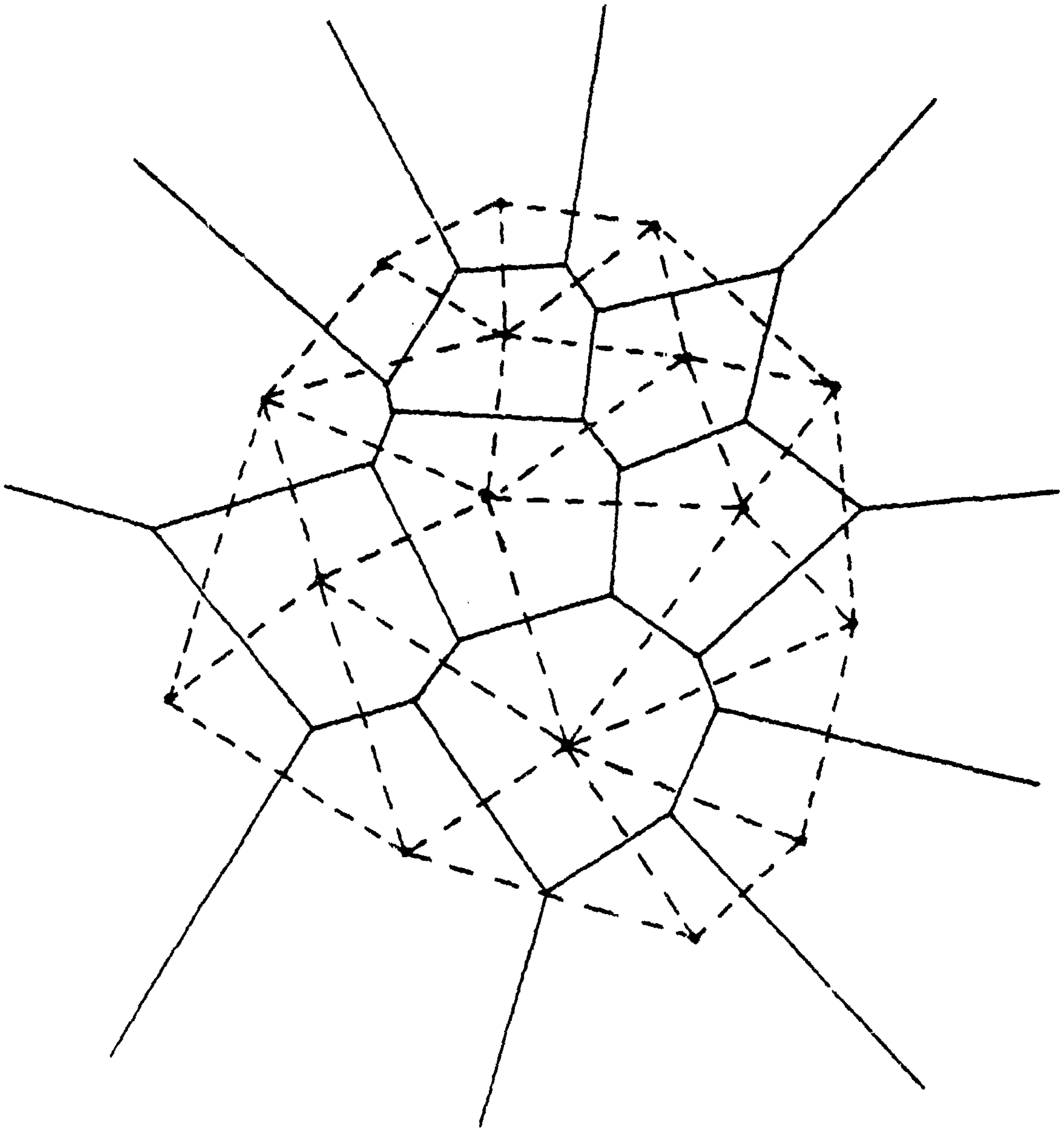


Figure 1 Voronoi diagram for a set of 16 points (solid lines). Delaunay triangulation (dashed lines).

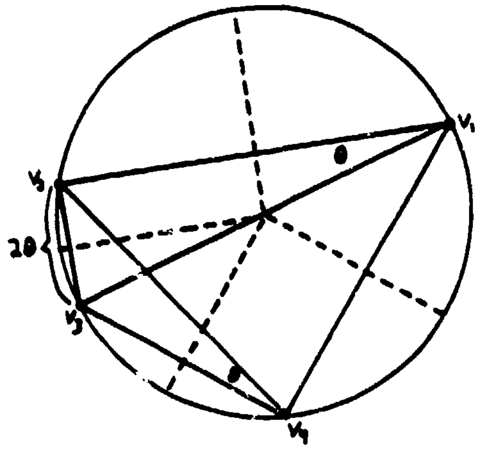


Figure 2 Lawson's example showing a triangulation over four co-circular points. The Voronoi tessellation is shown as dashed lines.

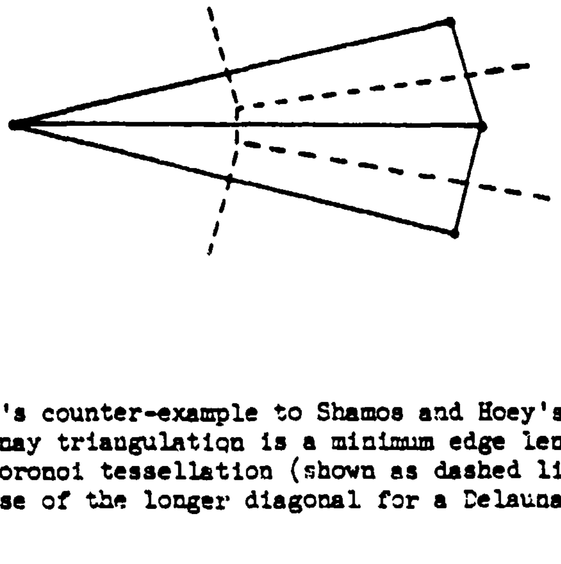


Figure 3 Lloyd's counter-example to Shamos and Hoey's claim that a Delaunay triangulation is a minimum edge length triangulation. The Voronoi tessellation (shown as dashed lines) indicates the use of the longer diagonal for a Delaunay triangulation.

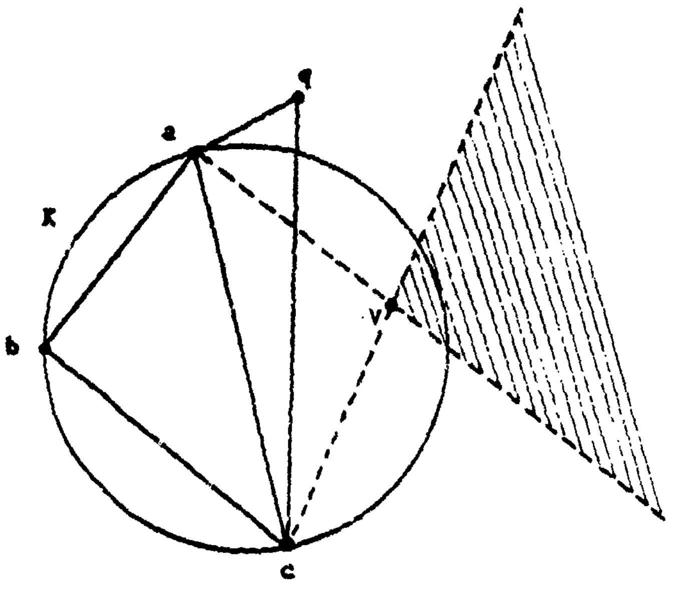


Figure 4 Illustration of the proof of Theorem 2.

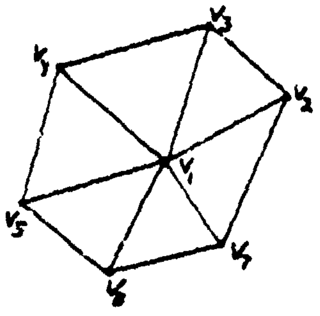


Figure 5 Example illustrating a doubly-linked circular list.

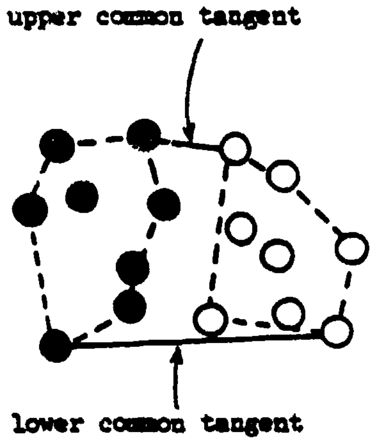


Figure 6 Illustration of the upper and lower common tangents of two convex hulls.

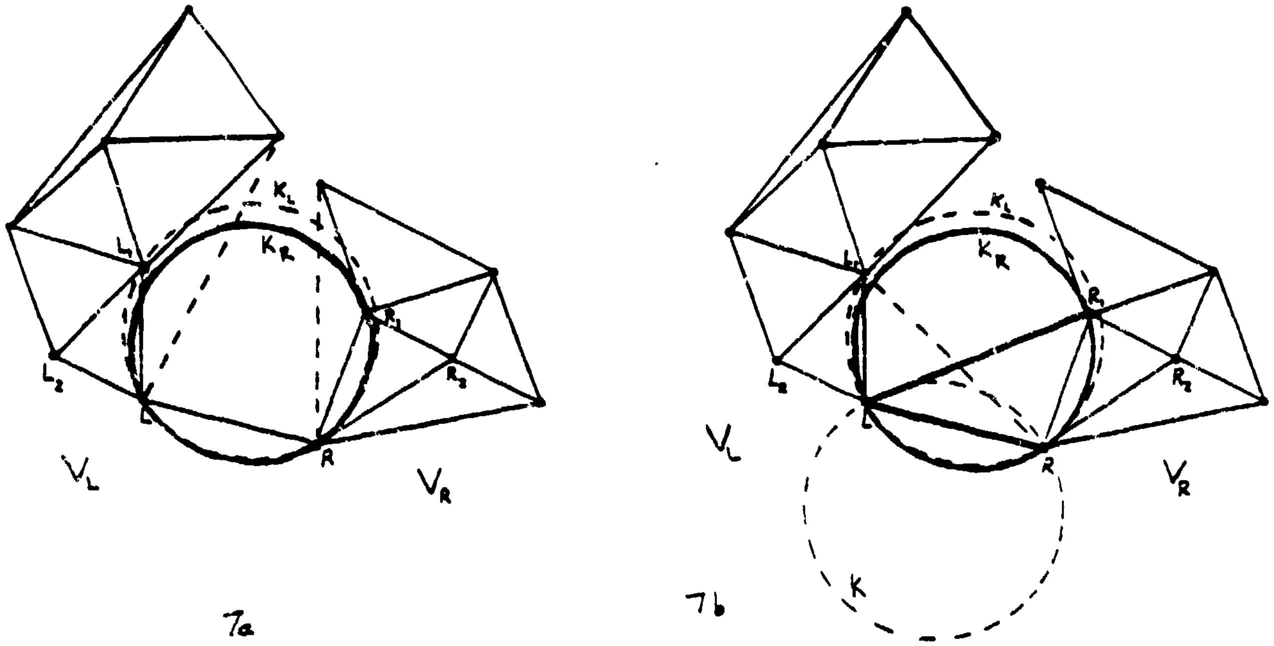


Figure 7 Illustrtion for the merge process. Bold lines are new Delaunay edges introduced by the MERGE algorithm.

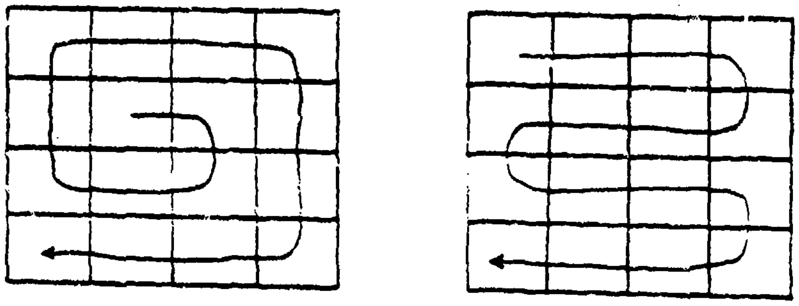


Figure 8 Two possible bin ordering schemes.

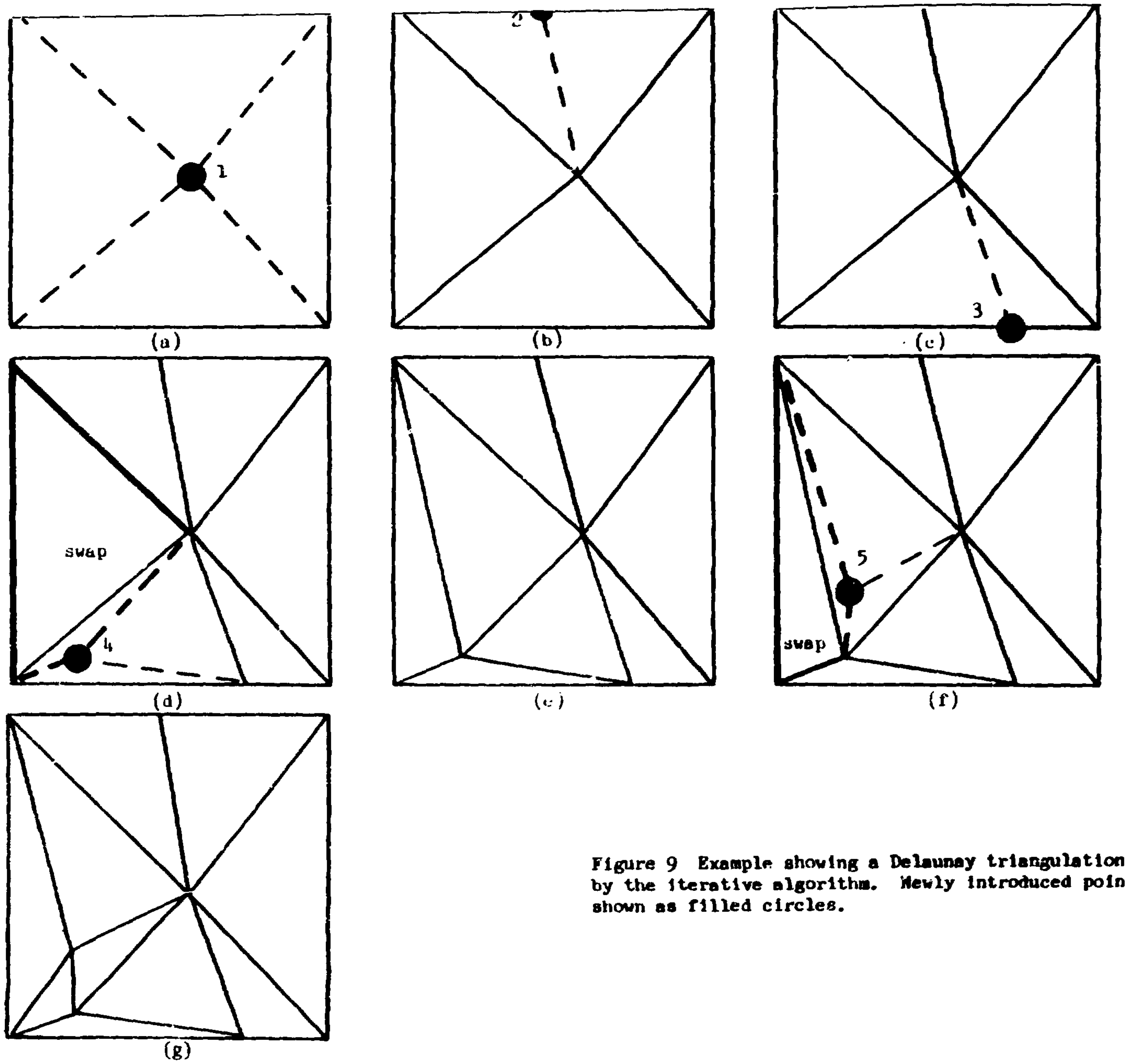


Figure 9 Example showing a Delaunay triangulation produced by the iterative algorithm. Newly introduced points are shown as filled circles.

# APPENDIX IV.3

## A SURVEY OF COLOR VIDEO FRAME BUFFER DICPLAY SYSTEMS FOR DESIGN GRAPHICS RESEARCH

This appendix presents the results of a comprehensive color display system survey conducted at the General Electric Corporate Research and Development Center during 1980. The information contained in the survey report should be highly useful in selecting a display configuration for a sensor prediction techniques research system. It should be noted however, that this survey was conducted for purposes of serving specific needs by a specific General Electric organization in the area of design graphics research. Hence, much of the author's commentary is given in that context.

GENERAL ● ELECTRIC

General Electric Company
Corporate Research and Development
Schenectady, New York

# TECHNICAL INFORMATION SERIES

| | | |
|---|---|---|
| AUTHOR<br>Atherton, PR | SUBJECT<br>computer graphics | NO. 80CRD051<br>DATE May 1980 |
| TITLE<br>A Survey of Color Video Frame Buffer Display Systems for Design Graphics Research | | GE CLASS 1<br>NO. PAGES 53 |
| ORIGINATING COMPONENT<br>Automation and Control Laboratory | | CORPORATE RESEARCH AND DEVELOPMENT<br>SCHENECTADY, N.Y. |

SUMMARY

The purpose of this survey is to investigate the frame buffer marketplace as it is today and project near future developments in order to recommend a suitable replacement for the DeAnza 2000 series system presently in use. Video frame buffer technology has been very dynamic in recent years resulting in some very powerful systems being offered at dramatic cost reductions.

After studying the requirements for design graphics research, the following criteria were established to restrict the survey to those vendors that would successfully satisfy these requirements:

- Refresh color video display type system
- Minimum of 512 × 512 pixel resolution
- Minimum of 8 bits memory per pixel
- Minimum of 5 bits per color gun in the color map
- Workable interface to PDP-11 and VAX computers

In terms of design graphics requirements, the real differences in cost among the viable systems seem to vary directly with the sophistication of the related processor. Some information on available hard copy units, video disks, and NTSC encoders is also included in this report.

KEY WORDS

computer graphics, color video display, image processing frame buffer, design graphics

INFORMATION PREPARED FOR ______________________________

______________________________

Additional Hard or Microfiche Copies
Available From

Technical Information Exchange
Bldg. 81 Room A133, Schenectady, N.Y., 12345

RD-54 (1/80)

# A SURVEY OF COLOR VIDEO FRAME BUFFER DISPLAY SYSTEMS FOR DESIGN GRAPHICS RESEARCH

P.R. Atherton

## INTRODUCTION

The Design Graphics Program, which is part of the Information Technology Branch of the Automation and Control Laboratory at General Electric Corporate Research and Development, has been working with a DeAnza 2000 series color video frame buffer system for over a year. Image displays have consisted mostly of output from SynthaVision* and MOVIE.BYU† imagery for various applications primarily oriented toward CAD/CAM development. Recently, some new and very exciting frame buffer systems have been developed by various vendors and are being offered at excellent price/performance ratios. With the appearance of these new powerful higher resolution systems, it was deemed necessary that Design Graphics should update their system to provide a more suitable environment for color video computer graphics development.

The color video frame buffer marketplace is a very dynamic environment which has made information gathering a difficult task in that various sources will often have different responses, and what is said today may be obsolete tomorrow. The author of this report welcomes any questions, comments, or added information pertaining to color video frame buffer systems, especially from those people who have had experience on any of the related systems.

*SynthaVision is a three-dimensional modeling system developed by the Mathematical Applications Group, Inc. (MAGI) of Elmsford, NY.

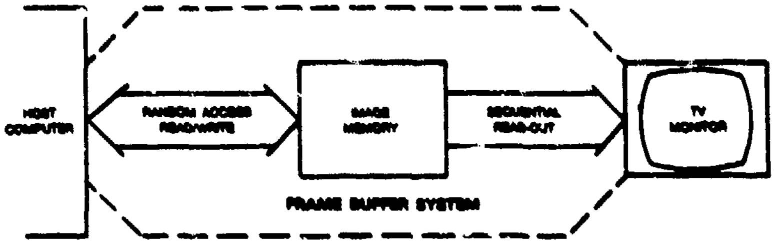
†MOVIE.BYU is a polygonal display program primarily used for movie animation developed in the Civil Engineering Department at Brigham Young University in Provo, Utah.

Manuscript received February 12, 1980

## VIDEO FRAME BUFFER DISPLAY SYSTEMS FOR DESIGN GRAPHICS RESEARCH

**Definition:**

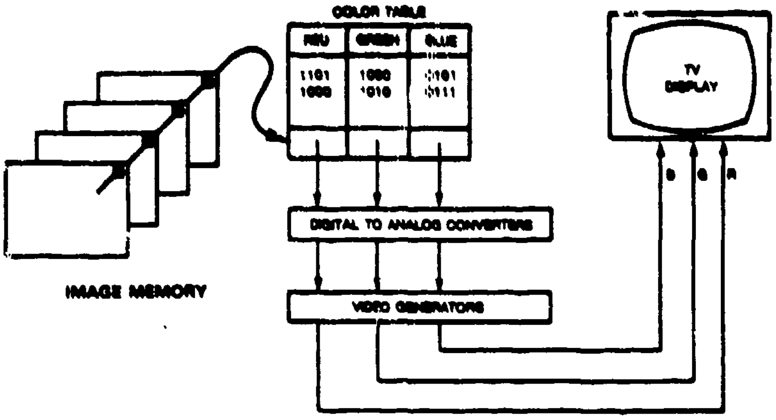
A video frame buffer system is basically a video raster scan display driver centered around a large piece of memory which contains image information.



The frame buffer system allows the user to update and read back from the image memory, while at the same time, the entire digital image information is sequentially read-out and converted to analog video signals for display on a TV monitor at refresh rates (from 20 to 60 times per second).

### IMAGE MEMORY AND COLOR MAP ORGANIZATION

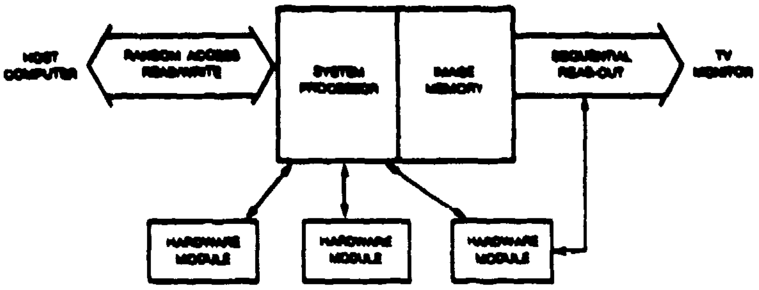
For most of the systems discussed in this survey, the image memory structure is essentially the same. A modular section of memory is dedicated to an entire display, such that one bit of information corresponds to a single pixel location on the screen. The number of memory modules corresponds to the number of bits of information dedicated to each individual pixel. The pixel bit information describes an address in a color map table.



The color table consists of a list of intensity values. Each location in the table contains relative intensities for each red, green, and blue color gun. During the sequential pixel scan-out (read), these values are taken from the table in the order that the image memory dictates, converted to analog signals then to video for display. The whole image memory is read out, converted, and displayed during a single refresh. There are some variations and enhancements to this basic design which will be noted in the system descriptions.

## SYSTEM PROCESSORS

Many frame buffer systems utilize micro- and/or array-processors to handle commands passed down from the host and to speed up the execution of various graphics operations. The processor normally sits between the image memory and the host computer with access to various special purpose hardware modules.



The system processor will accept commands from the host computer, translate them, and perform the appropriate operations to the image memory. These tasks may range in complexity from simply writing a pixel to enhancing all the visible edges in a digitized photograph. To speed up the execution of some of these processes, many vendors have provided special purpose hardware modules. A prime example of a hardware module useful for Design Graphics is the vector generator. Some vector generators are capable of rendering vectors on a color video display at rates of up to 16,000 vectors per second.

Many of the system processors can also be user programmable in micro-code. User programmability may prove to be a nice feature for Design Graphics in that it may allow us to put in the processor capabilities to do such things as fast parametric surface rendering — an operation no vendor offers today.

System processors can become very big and powerful and that power is usually reflected in the cost of the frame buffer system. In this survey, a lot of discussion will be devoted to the system hardware, firmware, and software facilities. In most cases, firmware and hardware facilities will reflect the power of the system processor and related hardware modules, while software facilities will describe code that exists on a host computer and is often FORTRAN callable. User programmability will describe capabilities for the user to down load or directly program the system processor.

## DESIGN GRAPHICS CONSIDERATIONS

The requirements for a video frame buffer system for the Design Graphics Program range from basic scan-line displays and interactive operations to sophisticated surface imagery and enhancement procedures. Some of the considerations include:

1. 2D and 3D model space control
    - Viewport
    - Window
    - Rotation
    - Translation
    - Scaling
2. Display of design and manufacturing models
    - Wire frame drawings
    - Sculptured surfaces
    - Complex solids
    - Finite element results
    - Text generation and control
3. Capabilities for various image generation techniques
    - Pixel by pixel
    - Scan-line via run-length encoding
    - Polygon and area fill
4. Ease of program development and implementation
    - FORTRAN callable software
    - High level graphics language
    - SIGGRAPH CORE graphics standard
5. Speed and ease of user interaction
    - Hardware or firmware facilities (i.e., zoom, scroll, or vector generation)
    - Peripheral devices (i.e., tablet, light pen or trackball)
6. Animation capabilities (i.e., cutter path or assembly visualization)
    - Run-length decode facilities
    - Image memory and color map controls
    - Video disk facilities
7. Image enhancement capabilities
    - Edge detection
    - Anti-aliasing
    - Contrast enhancement
    - Hue, intensity, and brightness controls

**8. Transportation of image displays outside of laboratory area**

- **Direct hard copy**
- **Polaroid or 35 mm**
- **Video recorder**
- **Video disk**

## PRELIMINARY DESIGN GRAPHICS REQUIREMENTS

The foremost consideration involved in the selection of a video frame buffer system is the capability of displaying the desirable images (SynthaVision, MOVIE.BYU, Sculptured Surfaces, etc.) in a more effective manner than the existing DeAnza ID2000 series system. The most obvious improvement would be the increase in image resolution from 256 × 256 (medium resolution) to 512 × 512 (high resolution) or to 1024 × 1024 (ultra-high resolution). Another improvement that could significantly enhance image display is the increase of the color table depth to allow for a greater variation of intensities over the range of a specific hue. In particular, changes across a curving surface would appear much smoother, and the effectivity of various lighting, texture, edge-smoothing, and anti-aliasing models would be greatly enhanced.

To restrict the survey to those vendors that could successfully fulfill our needs, preliminary requirements for selecting a video frame buffer display system are described:

1. A refresh display system to allow continual visual feedback of image updates.
2. A workable interface to the PDP11/70 and VAX computers.
3. Minimum of 512 × 512 pixel resolution.
4. Minimum of 8 bits of information per pixel, thus allowing 256 colors to be displayed simultaneously.
5. Minimum of 5 bits per color gun in the color map.

Note that this survey does contain some systems that do not satisfy requirement 5. These systems were included for completeness in that some people might choose to live with the 4 bits of color intensity if the system is extremely successful in regard to other Design Graphics considerations.

## SYSTEM CATEGORIZATION

In terms of Design Graphics requirements *versus* system design *versus* price criteria there seem to be 4 basic types of frame buffer systems available.

1. High resolution (512 × 512)
2. Ultra-high resolution (1024 × 1024)
3. High resolution upgradable (modular) to ultra-high resolution
4. Image processing systems for both high and ultra-high resolutions

In some cases, there is difficulty in discerning the image processing systems (4) from the image display systems (1, 2, and 3) since some systems try to bridge that gap to entice both markets. However, this is to our advantage in that the Design Graphics considerations previously discussed do reflect a need for some image processing capabilities such as edge detection and anti-aliasing. Therefore, I have classified as image processing only those systems for which a high price would be paid for powerful facilities that would do little to satisfy our needs.

A major point of contention pertains to the high *versus* ultra-high resolution decision. After some market study, it appears that the ultra-high resolution systems are not quite ready for consumption. Most vendors believe that the ultra-high resolution systems will not really be effective until the 64k chips become readily used. The significance of this probably will not be felt in the frame buffer market for about 2 or 3 years. Many hardware and firmware facilties offered with the high resolution systems simply are not available with ultra-high resolution. Peripherals, particularly video recording hardware, simply does not exist for 1024 × 1024 displays. Finally, software that was developed for high-resolution pixel data executes much slower and often demands more host memory on ultra-high resolution systems. On the other hand, the high resolution systems offer a very cost effective solution in that it is the same resolution of standard television. Thus, most of the video equipment related to the system has been in use for years and is greatly refined. It is also relatively less expensive. The high resolution frame buffer systems have also been in use for many years so that there are now many viable vendors who are operating in a very competitive marketplace. All these considerations make it pretty apparent that we should be directing ourselves toward the high resolution systems while keeping a close eye on the ultra-high resolution market developments.

## VENDOR SURVEY

This initial survey was specifically designed to obtain the general system configurations and capabilities in order to determine what types of systems satisfy our needs, to what extent, and at what cost. A number of decisions have already been made regarding the minimal requirements and the high *versus* ultra-high resolution question. We can now cut through a lot of extraneous information and concentrate on the system configurations that really concern Design Graphics.

For each system, pertinent information has been extracted and entered on a form sheet which can be found in Appendix A. Following each form sheet is a system diagram produced by the vendor (if one was available). Note that all technical and cost information has been based on high resolution (512 × 512) systems. The basic system cost refers to a simple system containing:

- 512 × 512 Image memory with at least 8 bits depth at each pixel
- System processor
- Interface to host computer (PDP 11/70)
- 19-Inch color monitor ($4000 if not offered directly)

Added to the basic system cost are the optional peripherals and facilities that would help satisfy our needs resulting in a total system cost.

The table following this section is an attempt to squeeze the tables from Appendix A onto a single sheet of paper to give the reader a *very* generalized overview of the systems surveyed. Some of the things to look for are:

**Approximate Total Price:** How much system power is needed in relation to the time of next foreseeable purchase?

**Memory Configuration:** Modular bit planes (512 × 512 × 1) are preferred to allow a more flexible initial purchase with easy lower cost upgrading later. Remember the 8 bits depth per pixel requirement (no. 4, page 7) mentioned earlier.

**Pixel Access:** The time required to read or write a pixel is very important in user interaction and critical for repeated operations.

**Color Map:** The greater the number of intensities allowed per Red-Green-Blue color gun, the smoother a color change can be made. Note the 5 bits per color gun requirement (no. 5, page 7). Also, the longer the color map (the number in parenthesis), the greater the number of colors that can be displayed on the screen for a single image if the corresponding number of bits per pixel are provided.

**Programmable Processor:** May allow the user to do specialized fast processing of image generation.

**Host Interface:** Look for DMA, because extended memory configurations could cause troubles going to the VAX or other host computer. Also, extended memory systems that work on other PDP 11s are more difficult to implement on the 11/70.

**Desirable Facilities:** Purely a personal reaction to the hardware, firmware, and software facilities a system has to offer. The reader should consider the requirements, examine Appendix A, and establish his own reaction about the desirable/offered facilities

**1024 × 1024 Display Upgrade:** If there is a strong near-term desire to display images at ultra-high resolution, even for just test purposes, we should definitely value this category as a very high priority item. If we want to hold off for 2 to 3 years until more viable systems are available, this factor can be considered in very low priority.

VIDEO FRAME BUFFER SYSTEMS (TABLE BRIEF)

| | Approximate Price | Memory Configuration | Pixel Access | Color Map | Programmable Processor | Host (PDP-11) Interface | Desirable Facilities | 1024 x 1024 Display Upgrade |
|---|---|---|---|---|---|---|---|---|
| DeAnza ID5000 | $19,525 | Restricted 8 or 12 | 1.8 Ms | 4-4-4 (1024) | No | DMA Unibus | Weak | N/A |
| AED 512 | $19,820 | Modular 8 Max | 1.0 Ms | 8-8-8 (256) | No | DMA Unibus | Good | N/A |
| 3 Rivers CVD | $20,000 | RLE Format | | 5-5-5 (64) | No | Extended Memory | Weak | N/A |
| Grinell GMR-27 | $22,000 | Modular 32 Max | 1.5 Ms | 8-8-8 (1024) | No | DMA DR11-B | Weak | Rough |
| Grinell GMR-270 | $23,200 | Modular 32 Max | 1.5 Ms | 8-8-8 (3-256) | No | DMA DR11-B | Weak | Rough |
| ADI Light-50 | $23,300 | Modular 16 Max | 1.2 Ms | 5-6-5 (256) | Yes-download TMS9900 | DMA like DR11-B | Good | N/A |
| Lexidata 3400 | $25,800 | Modular 16 Max | 1.0 Ms | 8-8-8 (256) | Difficult | DMA Unibus | Good | Trade-in |
| Genisco GCT-3000 | $31,050 | Modular 14 Max | 1.5 Ms | 8-8-8 (256) | Yes 2k RAM 6CT 3011 | DMA Unibus | Good | Fair |
| DeAnza VC 5000 | $34,950 | Restricted 16 only | 1.2 Ms | 4-4-4 (1024) | Yes LSI-11 | DMA via high speedport | Weak | Very Rough |
| Aydin 5216 | $42,500 | Modular 16 Max | 1.0 Ms | 8-8-8 (256) | 1 Mega Word Intel 8086 (Forth) | DMA DR11-B | Excellent | Fair |
| Norpak VDP | $44,000 | Modular 32 Max | 1.5 Ms | 8-8-8 (256) | Yes-download Bit-Slice Micro | DMA DR11-B | Very Good | N/A |
| DeAnza IP 5000 | $48,000 | Restricted 24 only | 800 Ns | 8-8-8 (3-256) | User Programmable Pipeline Array Proc. | Extended Memory | Fair | N/A |
| Ikonas | $48,000 | Modular Max 40 | 400 Ns | 8-8-8 (1024) | Fast 32 bit Micro | DMA DR11-B | Very Good (excellent Poss; | Built-in |
| Ramtek 9400 | $53,100 | Modular 16 Max | 1.12 Ms | 8-8-8 (1024) | Not Recommended Z-80 | DMA DR11-B | Excellent | Rough |
| Comtal Vision One/20 | $40K to $700K | Modular Max 512 | 1.5 Ms | 8-8-8 | LSI Micro | DMA DR11-B | Good | Rough |

**BRIEF SYSTEM COMMENTARIES**

This section contains a commentary on each video frame buffer system in which I will try to emphasize the strong and weak points of each system *relative to the needs of Design Graphics*.

**DeAnza ID5000** is the big brother of the DeAnza 2000 now in use. Relative to the other newer systems in this survey, most commentaries must be directed to the system's shortcomings which include:

- Limited memory configurations
- Restricted and slower memory access
- Only 4 bits of intensity per RGB in the color table
- Very limited facilities

The only real positive thing about the DeAnza ID5000 is its low price.

**AED 512** was one of the big stars of the SIGGRAPH '79 vendor exposition. It can act as a reasonably powerful pixel addressable frame buffer display system, or as a sophisticated character oriented graphics terminal and also comes with a Tektronix emulator. In terms of price versus performance, the AED 512 is very hard to beat in the lower price range of this survey. The only limitations (within the lower price range) that I see are the maximum of 8 bits depth per pixel.

**Three Rivers CVD** is basically designed for the play back of animation sequences. The major limitation of its Run-Length-Encoded (RLE) based memory is that many color variations across a single scan line cannot be executed. Smooth surface imagery generated by SynthaVision or sculptured surfaces would have to be approximated resulting in low quality imagery. The CVD does not even attempt to utilize most of the facilities we need and can only display 64 colors at one time (equivalent to 6 bits of depth per pixel).

**Grinell GMR27** is a pretty good low cost image display type of frame buffer including a 1024 long color map. However, the GMR27 lacks the facilities of some newer comparable systems.

**Grinell GMR270** is the image processing version of the GMR27. Basically, we would be paying more for the GMR270 than the GMR27 for image processing facilities that we have little use for.

**ADI Light-50** is a newcomer to the frame buffer marketplace but it appears to be quite viable and it is the only system that includes a NTSC encoder. The RGB intensity control in the color map is somewhat limited and memory planes can only be acquired in 512 × 512 × 4 units. However, the system does provide a way for the user to down load micro code to the TMS 9900. At this point in time, the ADI Light-50 has not yet been tested as a consumer product since most recent efforts have involved software interfacing with CHILD, Inc.

**Lexidata 3400** is a very flexible modular system that offers some nice facilities in a pretty comfortable price range. Some of the options include:

- Integer zoom controller
- Multiple scroll controllers

- 1024 Long color table with 8 bits per RGB
- Real-time edge smoother for 2X and 4X zoom
- Multiple overlays

**Genisco GCT-3000** does offer some good facilities at a fairly reasonable price. However, aside from the Grafmac II software package, the system offers little more than the AED 512 which costs almost half as much; and the GCT-3000 cannot compete with some of the higher price systems in terms of processor power. Considering this middle-market price/performance position Genisco has taken along with rumors of hardware unreliability, it would be more judicious for Design Graphics to avoid the GCT-3000.

**DeAnza VC5000** looks like an attempt to move away from the norm of frame buffer system design . . . but not in our direction. The user has to buy a 512 × 512 × 16 image memory configuration and at the same time be limited to only 4 bits of intensity per RGB color gun. There is a reasonable vector generator and character controller, but little else in the way of facilities . . . and a relatively high price tag.

**Aydin 5216** is the system to beat in terms of hardware, firmware and software facilities which include 3D object transformations and Z-sort hidden surface removal. The system even offers a user programmable Intel 8086 with one Megaword of memory and the Forth programming language. There is a very long (2048) color map, and with an extra video card (costing about $3000) can offer 8 bits depth of intensity per color gun as a non-standard configuration. However, at this point in time, the software packages are not yet complete.

**Norpak VDP** is a Canadian company that is new to the high level frame buffer market. They offer many nice features but it seemed as though every time I wanted to get some detail, I got a response like "well . . . its not quite complete yet." For that kind of money I'd want to see it completed and tested before buying.

**DeAnza IP5000** is good image processing system at a reasonable price, but really quite overpriced for Design Graphics use.

**Ikonas** is probably the best frame buffer system on the market to do computer graphics image display research on. The possibilities for their 32 bit microprocessor along with some of the built-in hardware facilities are really quite interesting. Ikonas claims to have developed a system that is modular enough to keep them on top of the research graphics market for at least the next few years. Engineers from other frame buffer companies say to look for good things in the near future from Ikonas.

**Ramtek 9400** probably has the best vector generator in the market (although Aydin's untested hardware shows promise). It also offers some other nice facilities like 2D rotation, entity detection and down load list processing. For an additional $6100 of hardware they will also offer 8 bits of color depth intensity per RGB which is non-standard.

**Comtal Vision One/20** is the top of the line for image processing systems. They offer a real-time pan of a 4098 × 4098 × 8 bit image and a movie capability to viewing a 512 × 512 × 512 bit array. It is basically a very powerful high priced system to perform operations we do not really need.

## RECOMMENDED SYSTEMS

In terms of price/performance criteria, I see six video frame buffer systems that stand out from the others. In order of approximate price they are:

| | |
|---|---|
| AED 512 | $19,820 |
| ADI Light-50 | $23,200 |
| Lexidata 3400 | $25,800 |
| Aydin 5216 | $42,500 |
| Ikonas | $48,000 |
| Ramtek | $53,100 |

The six systems seem to fall into two price ranges:

lower price range: $19,820 to $25,800

higher price range: $42,500 to $53,100

The lower price range systems are good, fast modular systems that will satisfy our needs quite nicely. The higher price systems are quite similar, except that they have much bigger processors, more hardware facilities and are upwards compatible to the ultra-high resolution displays.

In the higher price range, Aydin and Ramtek both provide the best hardware support modules, and extensive firmware/software facilities. Ramtek seems to offer a superior vector generator, but Aydin does support 3D object transformations and offers an easily programmable micro which could certainly prove valuable for surface generation. One the other hand, Ikonas provides a more state-of-the-art technology with the 32 bit 200 Nsec cycle processor with fast hardware multiplier and various special purpose hardware facilities including 3D transformations. Ikonas also has its image memory configured in such a way that an ultra-high resolution upgrade would only require a monitor change and setting a software switch. All other available systems require some hardware changes. So at the higher price range it comes down to a preference between Aydin's already developed facilities to Ikonas's more advanced engineering.

At the lower price range, the task of selecting a better system becomes more difficult. In my opinion, either the AED 512, ADI Light-50 or the Lexidata 3400 would be good selections that would stand us in good stead for the next two to three years. To help with this decision, I believe that we should look further into any special deals or company relationships that might provide more incentive one way or another. The following section will discuss a suggested approach to making the final selection.

## THE NEXT STEP

I believe the next logical step would be to evaluate our present needs and place into perspective our near term goals and our long term goals. Using that criteria, we should determine whether we need to put down the money for a high powered system, or whether we can satisfactorily pursue our goals with a lower cost system. In either case, the following questions should be asked of each critical vendor and responses requested in writing along with a formal detailed quote:

1. What is the delivery lead time?
2. What are the levels and costs of factory and field service?
3. What is the actual mechanical packaging (i.e., parts supplied)?
4. Are there any special company relationships to consider?
5. Are there any special price cuts to consider?
6. What are the planned future capabilities and to what extent or cost will they be available to us?

The responses to these questions may well provide the thrust to select one vendor over another. For example, Genigraphics received a handsome discount from Lexidata with the intent of future quantity buying. Some of the vendors suggested a company contract that would allow us to utilize their software packages on many in-house systems for one set price. Ikonas, for one, has stated that it is virtually impossible to deliver a system before 1980. We must understand all these factors, evaluate them, and then compare the tradeoffs in order to make a final selection.

## OTHER HARDWARE TO CONSIDER

There are some other devices related to video frame buffer systems that may help satisfy some of our requirements. They all represent ways by which we may record our images for communication or special applications.

### NTSC Encoders

The purpose of an NTSC encoder is to transform the RGB video signals that are displayed on high quality monitors, to standard television signals so we may record imagery directly. I found only two encoders that sold for under $3000.

**Lenco Inc.** offers NTSC encoder that "was specifically designed to encode high resolution color graphic computer displays irregardless of scan rates." They do claim that it will encode 1024 × 1024 resolution displays. Cost: $1595

**Video Modular Systems** offers an NTSC encoder that does not presently handle 1024 × 1024 resolution displays, but they suggest that they will have that capability in 2 to 3 months. Cost: $940

### Video Disks

A video disk is a disk unit especially designed to store and play back video images. It is presently being used quite successfully for computer animation in that it allows the storing of images at a slow rate and will play them back at a real-time rate. This allows for a much more flexible recording system than the traditional movie frame-by-frame photography method because there is no wait for film processing, and also because the animator may selectively edit random frames. The major problem with video disks is that most of them use laser technology making them quite expensive. A price tag above $100,000 is not unusual for a good digital system. However, there are some alternatives.

**Oktal** offers an analog video disk system for $40,000 which is being used in various places including Cornell's lab for Computer Graphics. As I understand it, the analog nature of Oktal's system requires so much tweeking that a video engineer should be on hand most of the time.

**Eigen Video** recently announced a lower quality low cost solution in the form of a magnetic disk. The monochrome recorder costs about $16,000, and the additional time base corrector for color recording boosts that system's cost to $24,500. The Eigen system can record up to 300 frames which is good for about 10 seconds of animation. The magnetic cassettes last approximately 100 hours before they must be rebuilt at a cost of $10 each.

The GE television development group in Portsmouth is looking into video disks and are planning on buying one already developed elsewhere. If animation is a definite requirement, I would recommend finding it at a system level because the costs of video disks are so high.

Some frame buffer systems now or soon will offer disk controllers in combination with hardware run-length decoders which may well satisfy most play back animation speed requirements. A lower cost solution would be the Three Rivers' CVD frame buffer system which is especially built for animation (discussed earlier in this survey). It could be purchased as a second frame buffer for $15,000 and used totally for animation display.

**Hard Copy**

At this point in time, color video hard copies are hard to find in a reasonable price range relative to their monochrome counterparts. The most notable systems available today are:

**Trilog Inc.** offers a system called COLORPLOT 100 which is based on a impact printer costing $9980. It produces a copy with 100 dots per inch vertical and horizontal resolution in about 3 minutes costing about 5¢.

**Dunn Instruments** has a hard copy unit that utilizes a Polaroid camera to make high quality 8 × 10 color photographs at about $5.00 per picture. The system will also allow for 35 mm slides to be taken and costs about $16,000.

**Matrix Instruments** produces a hard copy system very similar to the Dunn but with a basic system cost of about $12,800. It has the additional capability of formatting multiple images (2, 4, 6, 9, 25) on a single 8 × 10 Polaroid print which could result in substantial film cost savings. Unfortunately, each formatter costs $1000. Additional formatters are available for 35 mm slide (1-image-$3000) and microfiche (92 images-$1500). The total cost for a good system is about $20,000.

**Xerox** makes a color copier that will accept serial computer data and output a 100 dot per inch image. The system can also produce 35 mm slides and can operate in the normal copying format. Nice system for about $25,000.

**Applicon** now advertises an ink-jet plotter for about $40,000 that will make some nice color copies. Some examples of the ink-jet plotter output are on the wall in the Design Graphics Lab.

## LOW-COST COLOR VIDEO DISPLAY SYSTEMS

At this point, I would like to note that some of the systems already mentioned may well satisfy the low-cost requirements while offering upward configuration possibilities in a modular fashion.

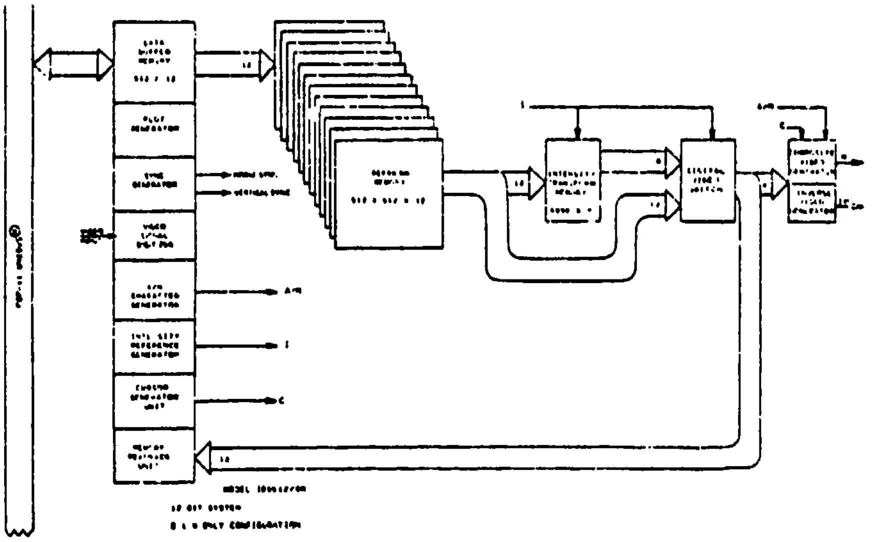
For example, Advanced Electronics Design, Inc. (AED) can configure a high resolution (512 × 512) system with 5 bits of depth at each pixel, PDP 11 interface, 14-inch color monitor, powerful firmware capabilities along with a Tektronix Plot-10 emulator for under $15,000 — not including quantity discounts. With 2 bits of depth, the cost is less than $12,500.

Applied Dynamics International (ADI) can put together a high resolution (512 × 512) system with 4 bits of depth at each pixel, PDP 11 interface, 14-inch color monitor, NTSC encoder, power firmware and a Tektronix emulator (Tek-Light) with some nice extensions for around $16,000.

These systems represent the upper-end of the low cost frame buffer spectrum, but they do offer some very nice features in a very cost effective manner.

## Appendix A

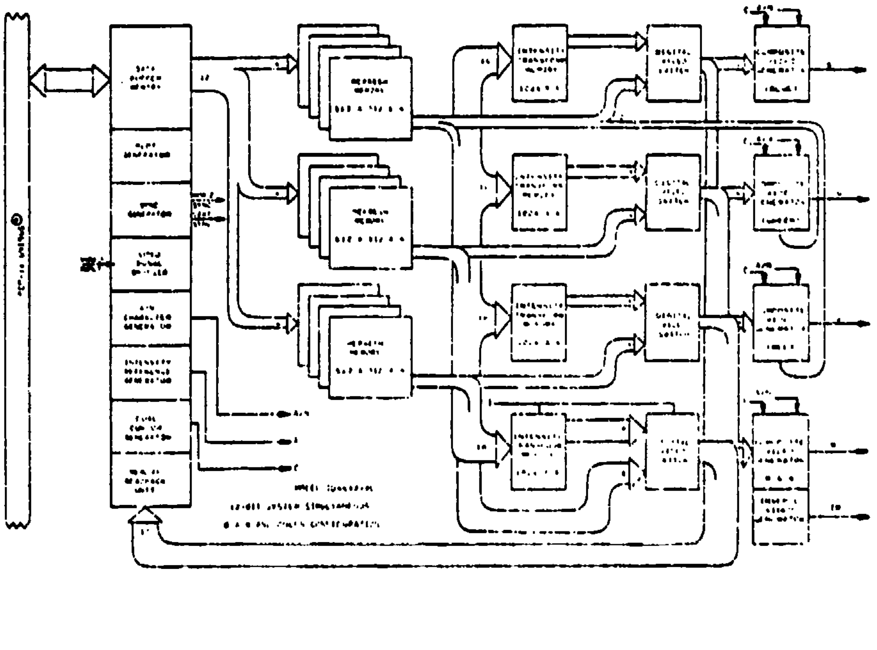
## TABLES AND DIAGRAMS DESCRIBING COLOR VIDEO FRAME BUFFER DISPLAY SYSTEMS



MODEL [illegible]

12 BIT SYSTEM

B & W ONLY CONFIGURATION



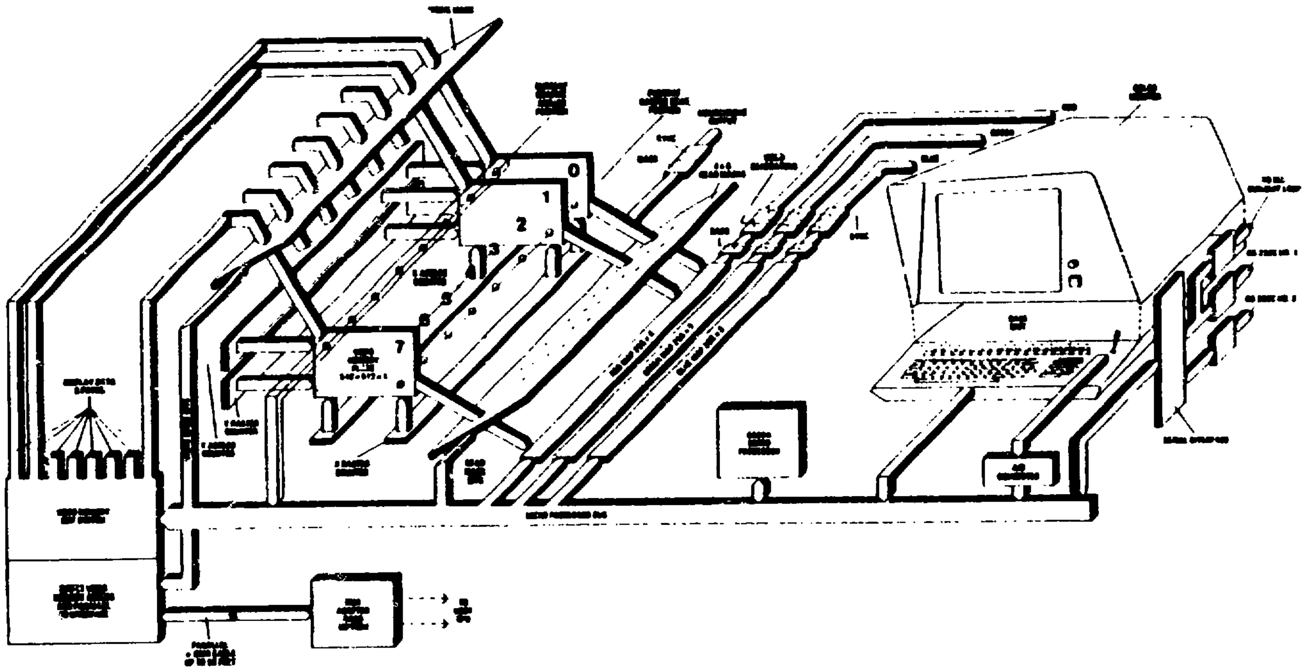
MODEL [illegible]

12-BIT SYSTEM SIMULTANEOUS

B & W AND COLOR CONFIGURATION

| | | |
|---|---|---|
| **SYSTEM**<br>DeAnza ID5000 | **CONTACTS** Rick Pizza Chuck Nordby<br>San José, CA (408) 263-7155 | |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** | 2 basic color systems<br>512 x 512 x 8 → color output 8 bits (3-R 3-G 2-B)<br>512 x 512 x 12 → color output 12 bits (4 each color) | |
| **PROCESSOR (SEE ATTACHED)** | | |
| **MEMORY ACCESS** | **ONE PIXEL I/O AFTER INITIALIZATION** 1:2 Ms (average 1.8 Ms)<br>Memory allocated line by line along DMA interface | |
| **REFRESH RATE** | 30 Hz | |
| **INTERLACE** | 2:1 | |
| **HOST** | **PDP 11/70** "Standard PDP-11 Unibus interface" - Unibus registers | |
| **INTERFACES** | **VAX** VMS I/O Driver $650 or DMA Board - | |
| **PERIPHERALS** | Joystick | $875 |
| **FACILITIES**<br>**H-HARDWARE**<br>**F-FIRMWARE**<br>**S-SOFTWARE**<br>**U-USER/PROCESSOR** | H Color Table 256 x 12 (4 per gun) for 512 x 512 x 8 System<br>1024 x 12 (4 per gun) for 512 x 512 x 12 | |
| | S VDLIB - Dec. 11 compatible<br>- Magnification function - real value - interpolated<br>- Polygon fill<br>- User interaction | $250 |
| | H Cursor<br>H Alpha - Numeric Generator | |
| **MONITORS** | | |
| **NTSC ENCODER** | | |
| **HARD COPY FACILITY** | Recommend Dunn | |
| **COST OF BASIC SYSTEM** | 512 x 512 x 12 System to allow 4 bits per RGB | ~$18,500 |
| **DESIRED EXTRAS** | VDLIB, Joystick | $19,525 |
| **TOTAL COST** | | |
| **COMMENTS** | | |

# Advanced frame buffer architecture of the AED 512

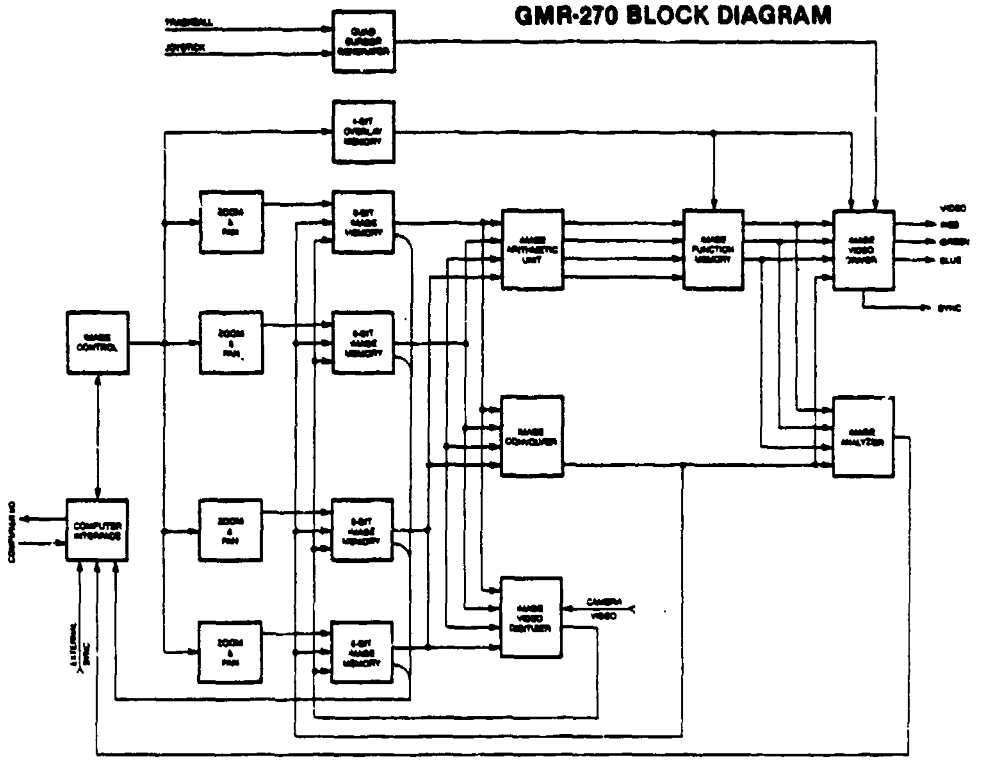


| | | |
|---|---|---|
| **SYSTEM** AED 512 | **CONTACTS** Gary Wilson (Sales)<br>Bedford, Mass. (617) 275-6400 | Pete Harris (Eng.)<br>Sunnyvale, CA (408) 733-3555 |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** | 512 x 512 x 1 bit planes - Max. 8<br>(possible to hook up 3 sets of 8) | $885 |
| **PROCESSOR (SEE ATTACHED)** | 6502A Micro - not user programmable | |
| **MEMORY ACCESS** | **ONE PIXEL I/O AFTER INITIALIZATION** Line - 50 to 100 Ms initial - 30 Ms subsequent - 1 Ms RLE - 5 Ms per pixel | |
| **REFRESH RATE** | 30 Hz | |
| **INTERLACE** | Yes | |
| **HOST** | **PDP 11/70** DMA Unibus Interface | $2,000 |
| **INTERFACES** | **VAX** | |
| **PERIPHERALS** | Keyboard with numeric pad and joystick (included) | |
| **FACILITIES**<br>**H-HARDWARE**<br>**F-FIRMWARE**<br>**S-SOFTWARE**<br>**U-USER/PROCESSOR** | F Included: Vector Generation - 9 Ms/pixel after initial<br>Scroll<br>Zoom 2x, 3x, 4x, 5x ... 16x<br>Polygon fill - after vectors<br>Area fill<br>Run Length Encode and Decode<br>Cursor - joystick control<br>"Area of interest" - similar to window<br>Circle generator<br>H Color table 256 x 24 (8 per gun) | |
| **MONITORS** | 14"<br>19" | $1,630<br>$4,750 |
| **NTSC ENCODER** | N/A Yet | |
| **HARD COPY FACILITY** | Working on Applicon and Hard Disk interfaces | |
| **COST OF BASIC SYSTEM** | 512 x 512 x 8 | $19,820 |
| **DESIRED EXTRAS** | included | - |
| **TOTAL COST** | | $19,820 |
| **COMMENTS** | Tektronix emulation mode - Unmodified Plot-10 (4000) included<br>No character generator<br>Working on floppy disk interface to unibus $4,000 | |

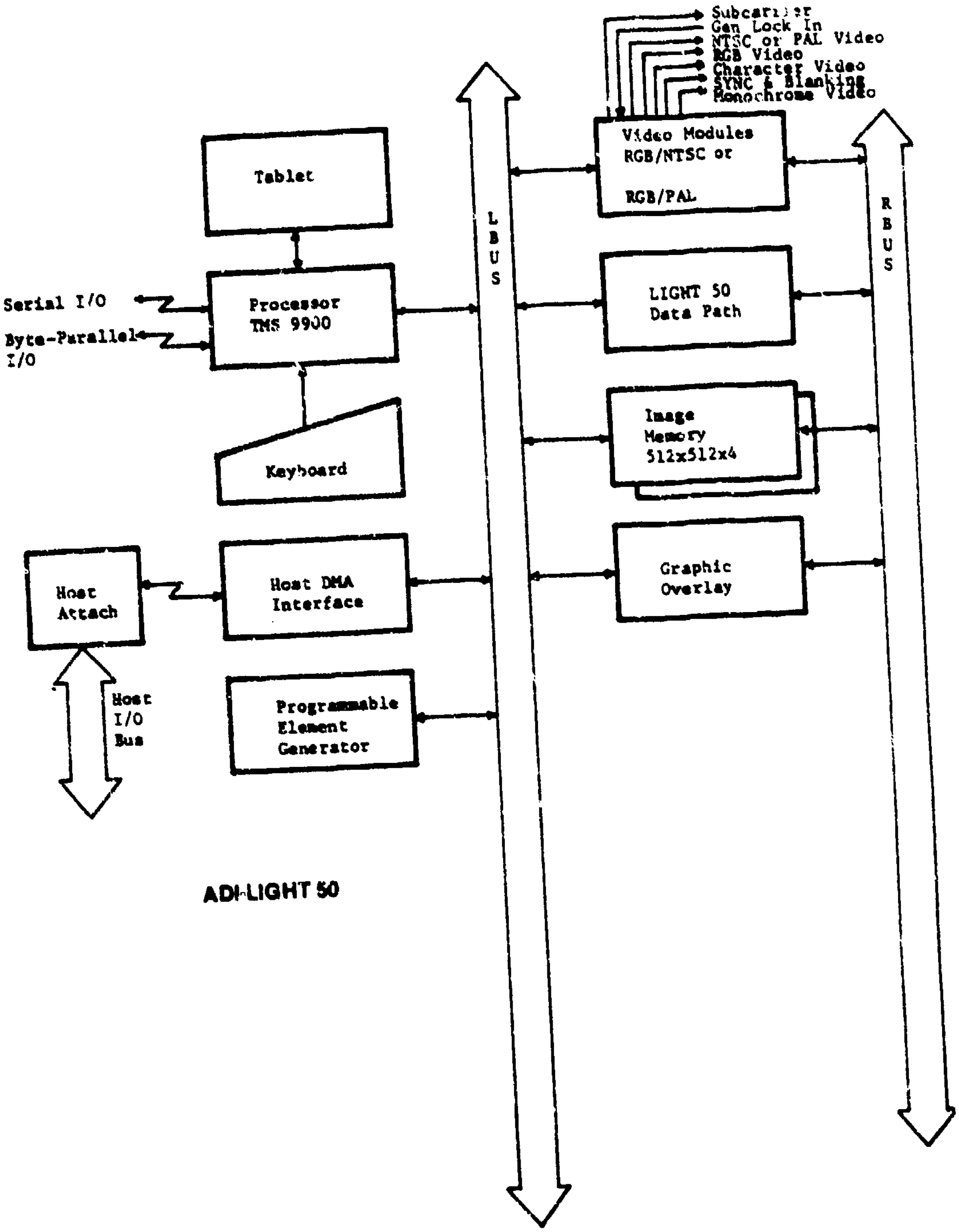
| | | |
|---|---|---|
| SYSTEM<br>Three Rivers CC | CONTACTS<br>(412) 621-6250 | Recommend purchase at McAir<br>(FMA 1978) |
| IMAGE MEMORY<br>CONFIGURATIONS AND COSTS | Memory acts as add-on to PDP-11 Memory. It is much smaller than other image memories since display information is compacted tc RLE format. | |
| PROCESSOR | Controls Run-Length encoding and decoding. | |
| MEMORY ACCESS | ONE PIXEL I/O AFTER INITIALIZATION | |
| REFRESH RATE | 30 Hz | |
| INTERLACE | | |
| HOST | PDP 11/70 Extended PDP-11 memory - difficult on 11/70 | |
| INTERFACES | VAX | |
| PERIPHERALS | Tablet | $1,500 |
| FACILITIES<br>H-HARDWARE<br>F-FIRMWARE<br>S-SOFTWARE<br>U-USER/PROCESSOR | H | Run-Length Encode/Decode |
| | H | Color Map 64 x 16 (5 per RGB, 1 for repeat line) |
| MONITORS | | |
| NTSC ENCODER | | |
| HARD COPY FACILITY | | |
| COST OF BASIC SYSTEM | | ≈ $20,000 |
| DESIRED EXTRAS | | |
| TOTAL COST | | |
| COMMENTS | Note: This system is designed for animation of simple imagery .... primarily of accounting information. It would not be suitable for display of continuous surfaces. | |

| | | | |
|---|---|---|---|
| **SYSTEM**<br>Grinell GMR-27 | | **CONTACTS** John Metzler<br>(408) 263-9920 San José, CA | |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** | | 512 x 512 x 1 bit planes<br>quantities: 1 - $800 2 - $1200 3 - $1600 4 - $2000<br>(max 32 planes) | |
| **PROCESSOR** | | Not user programmable | |
| **MEMORY ACCESS** | | **ONE PIXEL I/O AFTER INITIALIZATION** 6 Msec first pixel<br>~1.5 Msec subsequent pixels | |
| **REFRESH RATE** | | 30 Hz 60 Hz | |
| **INTERLACE** | | 2:1 1:1 | |
| **HOST** | | **PDP 11/70** Get DR11B from Dec. - Interface logic | $500 |
| **INTERFACES** | | **VAX** Similar | |
| **PERIPHERALS** | | Joystick | $700 |
| | | Trackball | $2,500 |
| **FACILITIES**<br>**H-HARDWARE**<br>**F-FIRMWARE**<br>**S-SOFTWARE**<br>**U-USER/PROCESSOR** | H | Vectors, Rectangles, Characters 1.5 Ms/pixel | included |
| | H | Scroll | included |
| | H | dot Cursor that blinks | included |
| | H | Image Function Memory Card $1600 (requires Video Drive Card)<br>- (3) 1024 x 8 color tables<br>- capability for split screen and image toggling | $1,200 |
| | H | 4 extra cursors | $1,000 |
| **MONITORS** | | buy and resell Conracs | |
| **NTSC ENCODER** | | | |
| **HARD COPY FACILITY** | | | |
| **COST OF BASIC SYSTEM** | | controler $5000, Memory $4000, Interface logic $500 & DRIIB, Monitor | $15,000 |
| **DESIRED EXTRAS**<br>**TOTAL COST** | | Image Function Memory Card, Video Drive Card, Joystick, Trackball | $22,000 |
| **COMMENTS** | | Video digitizing option $1200 | |

## GMR-270 BLOCK DIAGRAM

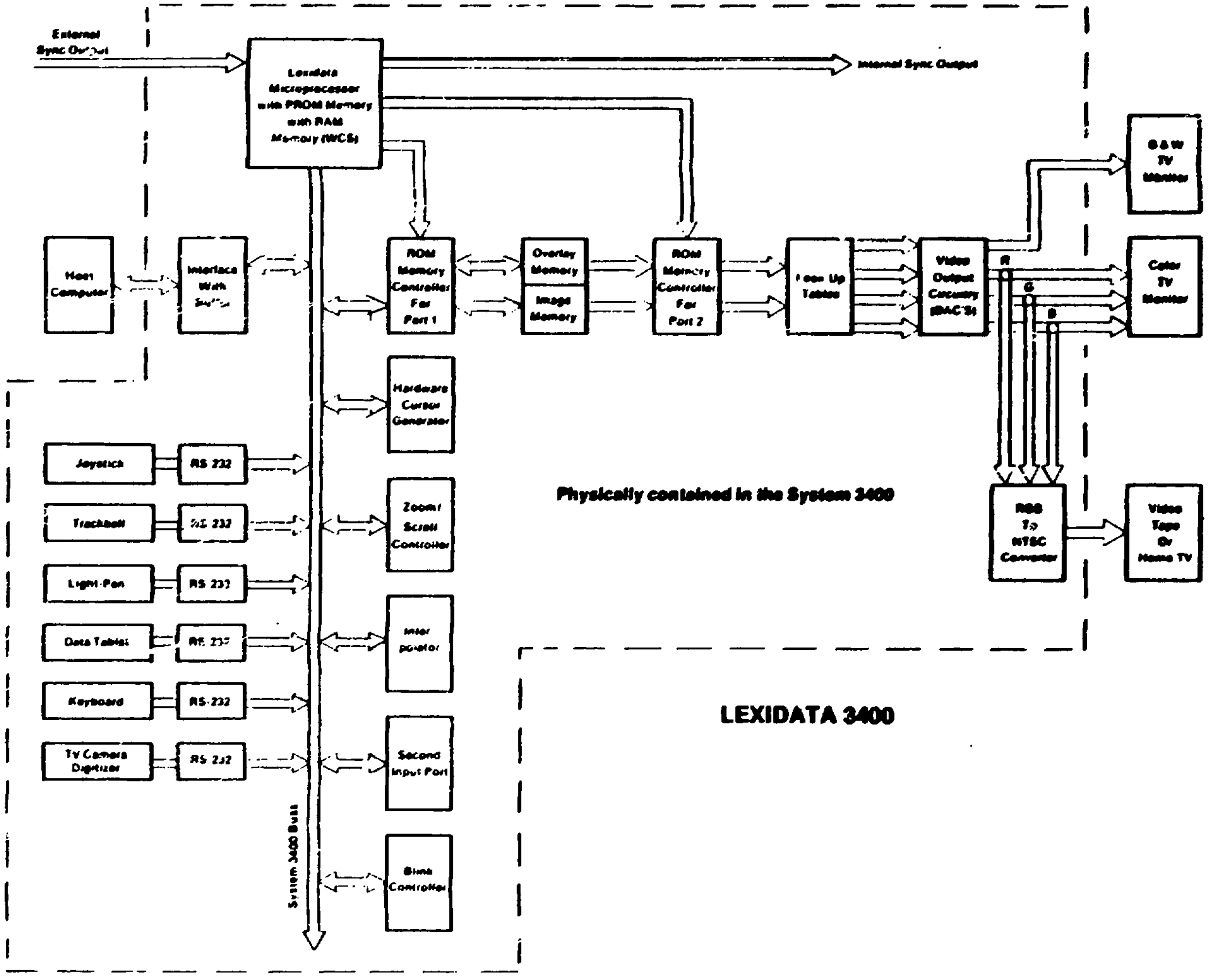


| | | | |
|---|---|---|---|
| **SYSTEM**<br>Grinell GMR-270 | | **CONTACTS** John Metzler<br>(408) 263-9920 San José | |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** | | 512 x 512 x 1 bit planes<br>quantities: 1 - $800 2 - $1200 3 - $1600 4 - $2000<br>(Max 32 planes) | |
| **PROCESSOR (SEE ATTACHED)** | | Not User Programmable | |
| **MEMORY ACCESS** | | **ONE PIXEL I/O AFTER INITIALIZATION** 6 Msec first pixel<br>1.5 Msec subsequent pixels | |
| **REFRESH RATE** | | 30 Hz 60 Hz | |
| **INTERLACE** | | 2:1 1:1 | |
| **HOST** | | **PDP 11/70** Get DR11B from Dec - Interface Logic | $500 |
| **INTERFACES** | | **VAX** Similar - | |
| **PERIPHERALS** | | Joystick<br>Trackball | $700<br>$2,500 |
| **FACILITIES**<br>**H-HARDWARE**<br>**F-FIRMWARE**<br>**S-SOFTWARE**<br>**U-USER/PROCESSOR** | H | Zoom (2x, 4x, 8x) and Pan<br>With cursor to denote screen center | $1,200 |
| | H | (3) 256 x 24 color tables (use only one at a time) | $1,600 |
| | H | Image Function Memory Card $1600 (Video Driver Card $1200)<br>- 3 1024 x 8 color tables<br>- capability for split screen and image toggling | |
| | H | Image Processor Card (multiply, divide...) | $2,200 |
| | H | Image Analyzer Card (histograms...) | $1,400 |
| | | Window read and write control | included |
| **MONITORS** | | buy and sell Conracs | |
| **NTSC ENCODER** | | | |
| **HARD COPY FACILITY** | | | |
| **COST OF BASIC SYSTEM** | | | $15,000 |
| **DESIRED EXTRAS** | | Zoom/Pan, Image Function Memory Card,<br>Video Drive Card, Joystick, Trackball | $23,200 |
| **TOTAL COST** | | | |
| **COMMENTS** | | Video Digitizing Option $1200 | |


Subcarrier
Gen Lock In
NTSC or PAL Video
RGB Video
Character Video
SYNC & Blanking
Monochrome Video
Tablet
Video Modules
RGB/NTSC or
RGB/PAL
L
B
U
S
R
B
U
S
Serial I/O
Processor
TMS 9900
Byte-Parallel
I/O
LIGHT 50
Data Path
Image
Memory
512x512x4
Keyboard
Host
Attach
Host DMA
Interface
Graphic
Overlay
Host
I/O
Bus
Programmable
Element
Generator

ADI-LIGHT 50

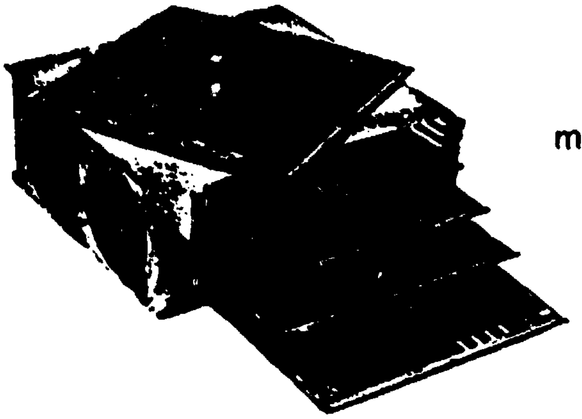
|  |  |  |  |
|---|---|---|---|
| **SYSTEM**<br>ADI - Light 50 |  | **CONTACTS** Harold F. Clearwaters (Main)<br>Bob Ray - local salesman<br>Lowell, MA '617) 459-2578 |  |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** |  | 512 x 512 x 4 bnard - Max 4 now, 8 future<br>(1024 x 1024 display not announced yet) | $2,500 |
| **PROCESSOR (SEE ATTACHED)** |  | 16 bit Micro - TMS 9900 - User can download<br>8k ROM & 4k RAM which can be increased |  |
| **MEMORY ACCESS** |  | **ONE PIXEL I/O AFTER INITIALIZATION** 1.2 Ms |  |
| **REFRESH RATE** |  | 30 Hz |  |
| **INTERLACE** |  | Yes |  |
| **HOST** |  | **PDP 11/70** interface w/micro → host attachment → host I/O bus | $2,500 |
| **INTERFACES** |  | **VAX** N/A |  |
| **PERIPHERALS** |  | Keyboard with numeric pad & 16 function switches | $600 |
|  |  | Joystick | $200 |
| **FACILITIES**<br>**H-HARDWARE**<br>**F-FIPMWARE**<br>**S-SOFTWARE**<br>**U-USER/PROCESSOR** | H | 1 pix (vertical) by 16 pix (horizontal) scroll and zoom 2x, 4x, 8x | $500 |
|  | H | Fast Element Generator (fill 2.5 Ms/pixel)(Vectors 1.3 Mx/pixel) | $3,000 |
|  | H | Graphics Overlay - RS170 (camera input)(512 x 512 x 1 plane) | $2,500 |
|  | H | Color table 256 x 16 (standard) 1024 x 16 (optional -)<br>(5-red 6-blue 5-green) |  |
|  | F | Included<br>- arbitrary (real) scaling 1x to 256x - must rebuild image<br>- generate circles, arcs, characters, rectangles, conics<br>- area fill and rectangle fill<br>- cross-hair cursor<br>- multiple views with a 2D window (function of zoom & scroll) |  |
| **MONITORS** |  | N/A |  |
| **NTSC ENCODER** |  | Yes | Included |
| **HARD COPY FACILITY** |  | Tektronix hard copy- hook-up RS170 & Child System |  |
| **COST OF BASIC SYSTEM** |  |  | $19,000 |
| **DESIRED EXTRAS**<br>**TOTAL COST** |  | Zoom/scroll, Prog. element generator,<br>keyboard, joystick | $23,300 |
| **COMMENTS** |  | Teklight - Tek emulator Prom | $850 |
|  |  | Can overlay text<br>Child System<br>S & S electronics dropped Genisco & Ramtek - ADI looks good. |  |



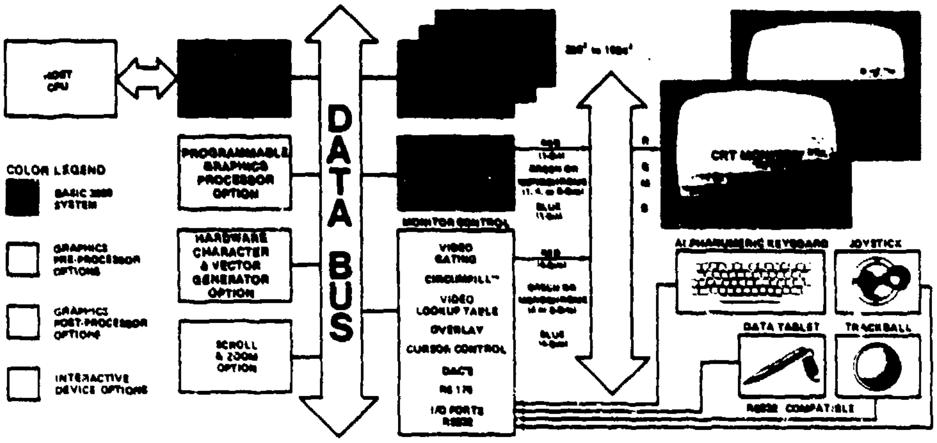
LEXIDATA 3400

|  |  |  |  |
|---|---|---|---|
| **SYSTEM**<br>Lexidata 3400 | **CONTACTS** Art Frankie<br>Burlington, Mass. (617)273-2700 |  |  |
| **IMAGE MEMORY** |  |  |  |
| **CONFIGURATIONS AND COSTS** |  | 512 x 512 x 1 plane (Max 16)<br>1024 x 1024 x 1 plane (Max 4) | $1120<br>$4480 |
| **PROCESSOR (SEE ATTACHED)** |  | 12 bit Micro with 1k-3k PROM & 1k RAM for Writable Control Store<br>difficult user programming (2k-12k) |  |
| **MEMORY ACCESS** |  | ONE PIXEL I/O AFTER INITIALIZATION ~1 Ms |  |
| **REFRESH RATE** |  | (512) 30 Hz (512) 60 Hz (1024) 30 Hz |  |
| **INTERLACE** |  | 2:1 1:1 2:1 |  |
| **HOST** |  | PDP 11/70 recommend 16 bit parallel DMA interface to Unibus | $1200 |
| **INTERFACES** |  | VAX similar |  |
| **PERIPHERALS** |  | w/RS232 joystick<br>keyboard<br>trackball<br>tablet<br>(dropped lightpen support) | $1400<br>$ 900<br>$2500<br>$2000 |
| **FACILITIES**<br>H-HARDWARE<br>F-FIRMWARE<br>S-SOFTWARE<br>U-USER/PROCESSOR | H | Zoom (1x, 2x, 3x, 16x) and Scroll | $1000 |
|  | H | for 8 bit depth 256 x 24 (8 per RGB)<br>Maximum of 1024 x 24 simultaneous | $3405 |
|  |  | Blink controller |  |
|  |  | Multiple overlays |  |
|  |  | RAM add-on for micro | $ 500 |
|  | F | Image Display Operating System - accessible via Software Driver which is resident on host.<br>-vector generation each vector : 10Ms + 2Ms per pixel<br>-ramp feature for color look-up<br>-movie feature using zoom and scroll |  |
|  |  | Cursor | $ 700 |
| **MONITORS** |  | $512^2$ resolution<br>$1024^2$ resolution | $ 3,000<br>$8,000 to $13,000 |
| **NTSC ENCODER** |  | optional | $ 3,000 |
| **HARD COPY FACILITY** |  | Tektronix hardcopy | $ 7,500 |
| **COST OF BASIC SYSTEM** |  |  | ~$20,000 |
| **DESIRED EXTRAS** |  | Joystick, keyboard, tablet, zoom/scroll, RAM add-on | ~$25,800 |
| **TOTAL COST** |  |  |  |
| **COMMENTS** |  |  |  |

# GENISCO DIGITAL DISPLAY SYSTEMS



**Setting a new criteria of modularity, display dynamics, performance, reliability, processing speed and cost-effectiveness!**



Completely programmable. Genisco Digital Graphic Display Systems are modularly expandable to cover the widest range of application requirements You specify the features and options you need Genisco graphic display experts, using functionally proven "building-block modules, tailor systems that cost-effectively answer that need . dynamically, efficiently and reliably!

**Basic 3000 System.** Consists of the proper CPU Interface fast entry MOS/RAM Refresh Memory Modules – with read/write, word or bit capabilities, automatic DMA access for block transfers to 833K 16-bit words/second – and the Video Control, that generates the basic system timing and formats the output for RS170 waveforms

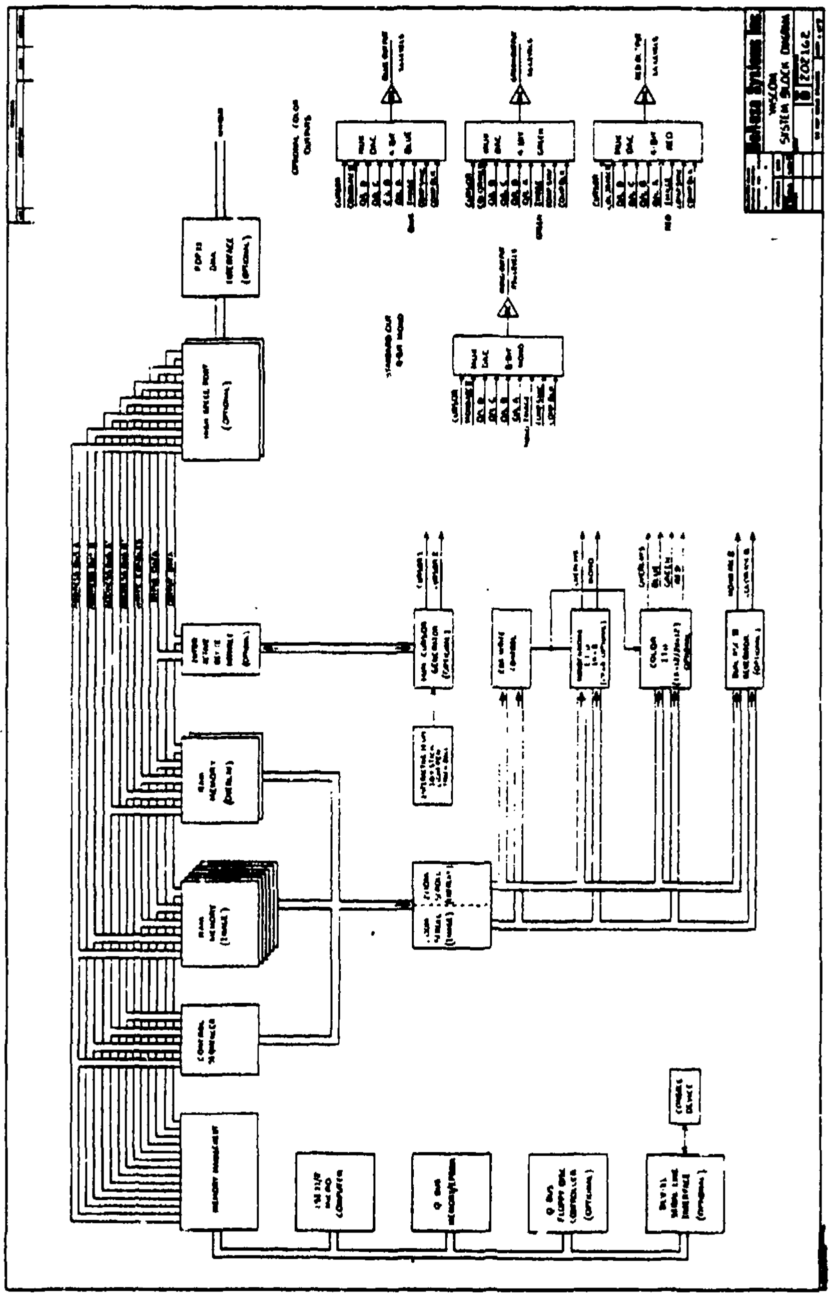
**Pre-Processor Options.** The GCT-3C11 Programmable Graphic Processor, under control of its own program that is easily modified, converts data that includes both vectors and characters, and routes it to the memory modules A Hardware Character/Vector Generator is also available for very fast dynamic applications.

**Post-Processor Options.** Monitor Control Modules, in a number of optional configurations, provide added capability to the system such as Video Gating, Circumfill™, Video Lookup and Readback Overlay, Cursor Control, DAC's, RS232 I/O Ports, and RS170 composite video waveforms. A Scroll and Zoom – by image or plane – is also optionally available.

**Interactive Device Options.** RS232 compatible interactive devices like an ASCII Alphanumeric Keyboard with 16 lighted function switches, Trackball and Joystick, and an 11" x 11" Graphic Data Tablet are available from Genisco.

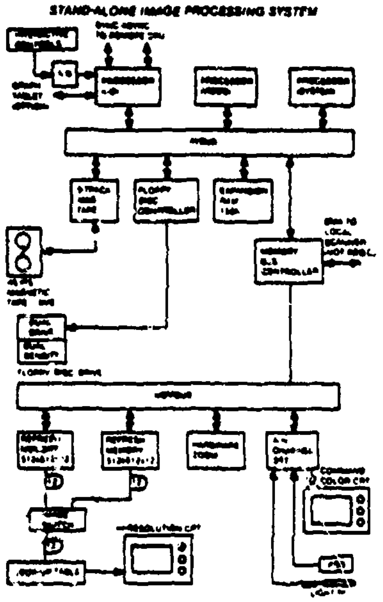
For particulars on your specific digital graphic display requirements, contact Genisco – a name that has stood for advanced technology over the past 30 years

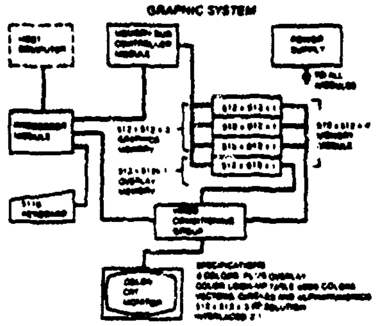
| | | | |
|---|---|---|---|
| **SYSTEM**<br>Genisco GCT-3000 | **CONTACTS** | Stu Robert (UP) Joe Tublan Bob Frey (UP)<br>Bob Ray (salesman) Dan Jones Dave Pauley<br>Lowell, Mass (617)459-2578 Irvine, CA | |
| **IMAGE MEMORY** | | | |
| **CONFIGURATIONS AND COSTS** | | 512 x 512 x 1 plane | $1500 |
| | | 1024 x 1024 x 1 plane | $2500 |
| | | Max of 14 planes in 2 chassis | |
| **PROCESSOR (SEE ATTACHED)** | | Programmable Graphics Processor (PGP)<br>Graphic Operating System takes about 1/2 of the 4k RAM | |
| **MEMORY ACCESS** | | **ONE PIXEL I/O AFTER INITIALIZATION** | |
| **REFRESH RATE** | | (512) 60 Preferred 40, 30 Hz (1024) 30 Hz | |
| **INTERLACE** | | 1:1 Yes 2:1 Yes 2:1 | |
| **HOST** | | **PDP 11/70** (RSX11-M available) Note: Driver w/demos = $560 | $1700 |
| **INTERFACES** | | **VAX** | $1700 |
| **PERIPHERALS** | | keyboard | $1350 |
| | | trackball | $2900 |
| | | joystick | $1000 |
| | | tablet | $1500 |
| **FACILITIES**<br>**H-HARDWARE**<br>**F-FIRMWARE**<br>**S-SOFTWARE**<br>**U-USER/PROCESSOR** | H | Character/vector generator ~10 Ms per pixel | $2000 |
| | H | Scroll and Zoom (2x, 4x, 8x) | $1300 |
| | H | Color Table 256 x 24 | included |
| | H | Fill Mode - will fill between vectors (max 4 planes) | included |
| | H | Cursor and blink control in lieu of second cursor | included |
| | S | Graf pac II - fortran callable graphics subroutine library includes: area fill, some 2D translations, curves, lines, vectors, text control | $3000 |
| **MONITORS** | | 512 x 512 | $ 3,240 |
| | | 1024 x 1024 | $15,200 |
| **NTSC ENCODER** | | | $ 4,500 |
| **HARD COPY FACILITY** | | | |
| **COST OF BASIC SYSTEM** | | | $20,700 |
| **DESIRED EXTRAS** | | keyboard, joystick, tablet, character/vector generator, | |
| **TOTAL COST** | | zoom/scroll, Graf pac II | $31,050 |
| **COMMENTS** | | -vectors must be erased for movement, hardware sissoring<br>-character controls: 1x...16x zoom, 90° rotation, programmable fonts<br>-plot-10 emulator available<br>-rumors of hardware problems from users | |


SYSTEM BLOCK DIAGRAM
B 202162

| | | | |
|---|---|---|---|
| **SYSTEM**<br>DeAnza VC5000 | **CONTACTS** Rick Pizza Chuck Nordby<br>San José, CA (408) 263-7155 | | |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** | All systems are 512 x 512 x 16<br>System #1. Monochrome - 8 bits intensity, 4 overlay, 4 aux.<br>System #2. Color 12 bits RGB (4 per gun), 4 overlay | | |
| **PROCESSOR** | LSI-11 totally user programmable (24K bytes) | | |
| **MEMORY ACCESS** | **ONE PIXEL I/O AFTER INITIALIZATION** 1.2 Ms | | |
| **REFRESH RATE** | 30 Hz | | |
| **INTERLACE** | 2:1 | | |
| **HOST** | **PDP 11/70** RSX11-M requires special high speed interface: Port | | $1,950 |
| | (Card) DMA | | $1,475 |
| **INTERFACES** | **VAX** | | |
| **PERIPHERALS** | Joystick (cursor) | | $875 |
| | ADM-3 Dumb Terminal | | $1,250 |
| **FACILITIES** | H | Color Tables with Image Transform Control - "Peudo Color" | |
| **H-HARDWARE** | | Monochrome 2048 x 8 | $400 |
| **F-FIRMWARE** | | Color 1024 x 12 (4 per color) | $1,950 ← |
| **S-SOFTWARE** | | Color 2048 x 12 (4 per color) | $2,000 |
| **U-USER/PROCESSOR** | H | Dual Cursor (different modes) | $1,400 |
| | H | Zoom (2x, 4x, 8x) and Scroll | included |
| | S | Vector Generation 8.5 Ms/pixel | included |
| | F | Character Control ($1,000) w/Color | $1,500 |
| **MONITORS** | | | |
| **NTSC ENCODER** | | | |
| **HARD COPY FACILITY** | Recommend Dunn | | |
| **COST OF BASIC SYSTEM** | including 1024 x 12 color table | | ~$31,400 |
| **DESIRED EXTRAS** | joystick, terminal, cursor | | $34,950 |
| **TOTAL COST** | | | |
| **COMMENTS** | -Designed to stand alone - terminal and floppy<br>-Image from floppy approx. 11 sec. | | $4,450 |

# Typical Model 5216 System Configurations





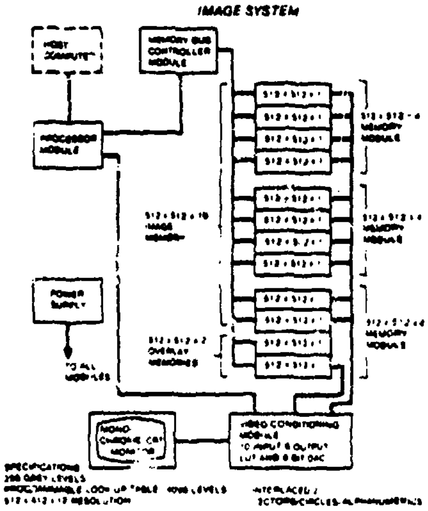
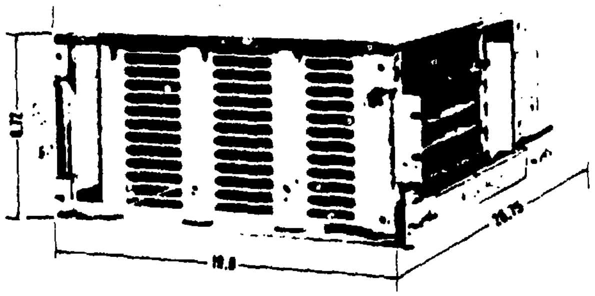


IMAGE SYSTEM
HOST COMPUTE
MEMORY BUS CONTROLLER MODULE
PROCESSOR MODULE
512 x 512 x 1
512 x 512 x 4 MEMORY MODULE
512 x 512 x 10 IMAGE MEMORY
512 x 512 x 2 OVERLAY MEMORIES
POWER SUPPLY
TO ALL MODULES
MONO-CHROME CRT MONITOR
VIDEO CONDITIONING MODULE
SPECIFICATIONS
256 GREY LEVELS
512 x 512 x 12 RESOLUTION

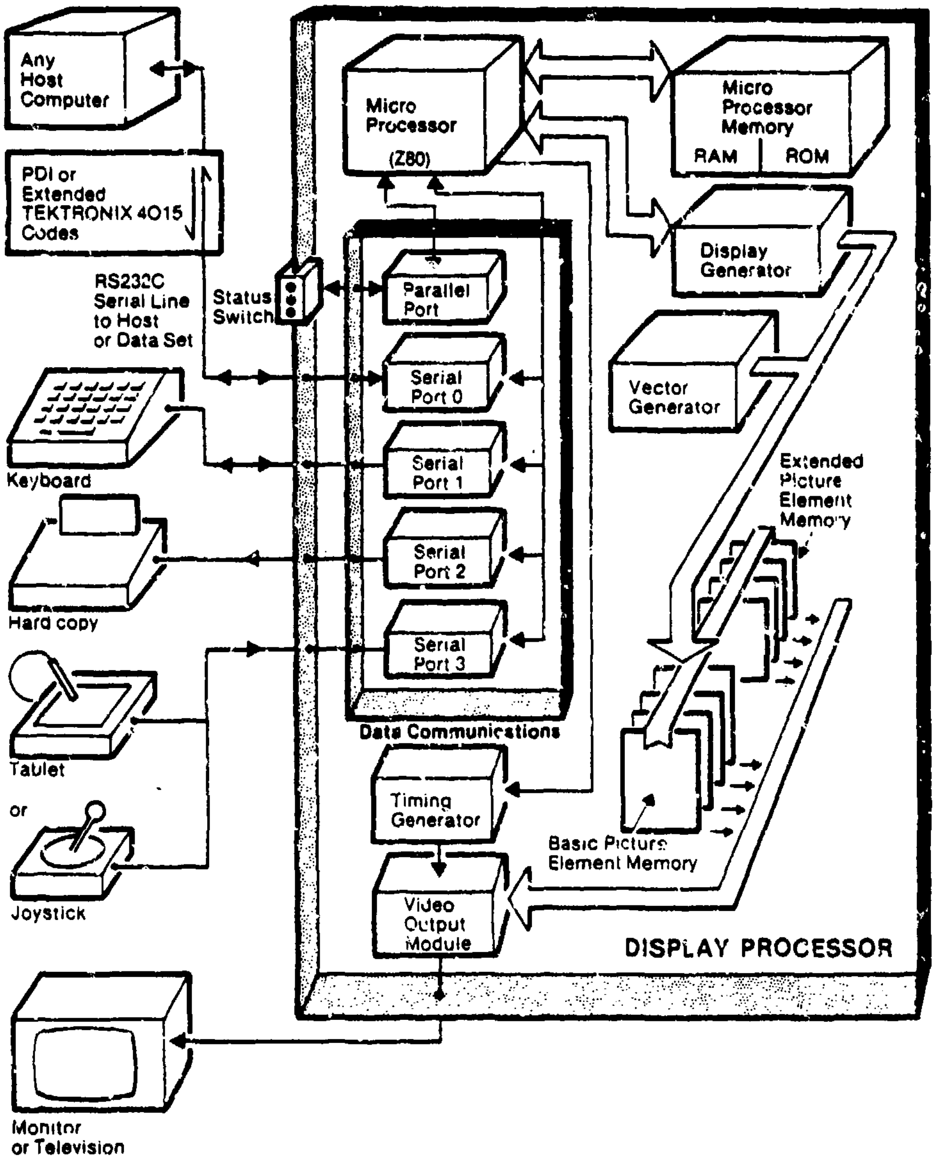
## Mechanical Configuration

The Model 5216 is packaged in a fan cooled 19 inch rack-slide mountable chassis. Eleven module or .ull card slots are provided pr r chassis — some modules require only half cards Up to three chassis may be linked together to form a 33 module configuration. Each chassis is self powered by means of a rear mounted, self-contained, power module All connections are made via bc ard-mounted connectors Side entry of plu -in modules allows full access to card-edge connections as we'l as the printed back planes when the chassis is extended on the slides

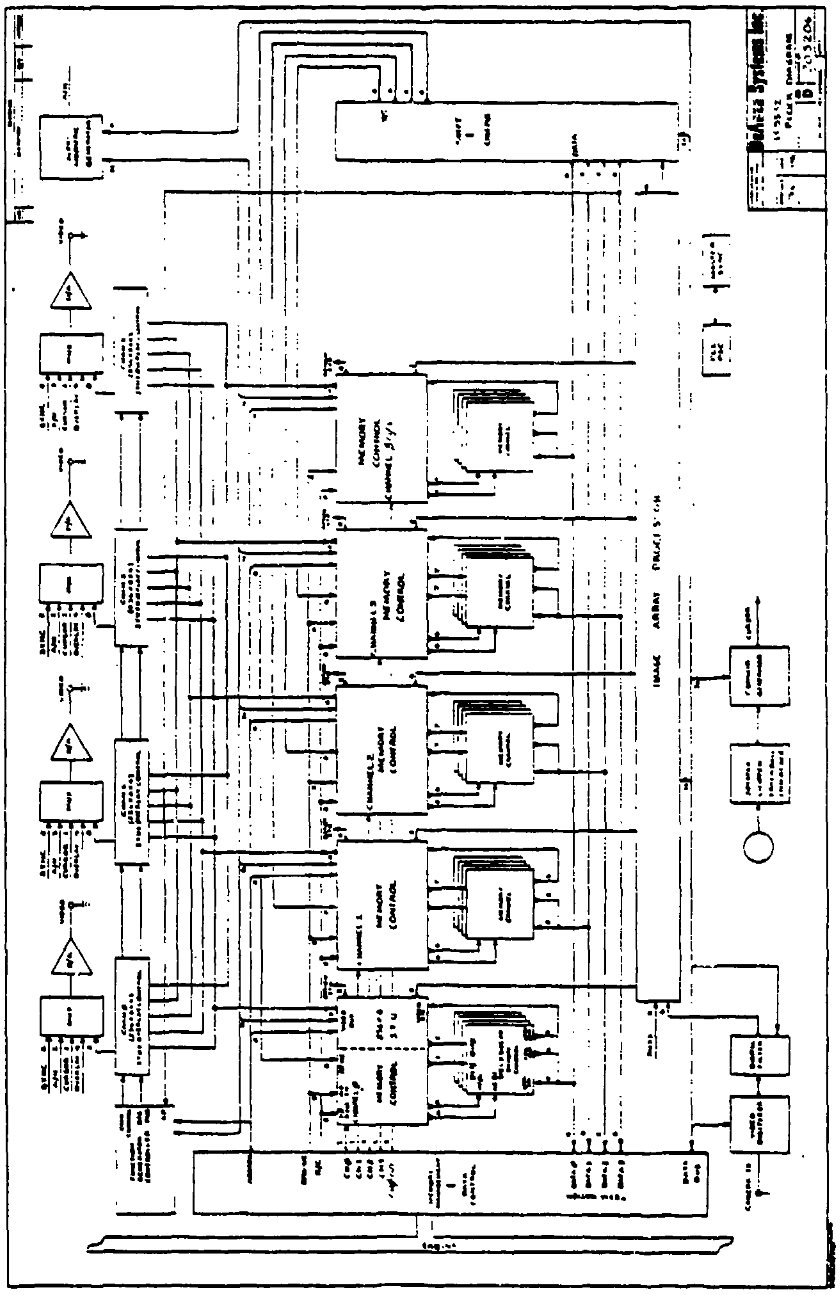


| | | | |
|---|---|---|---|
| **SYSTEM**<br>Aydin 5216 | | **CONTACTS** Ralf Hubert (Sales)<br>(617) 649-6754<br>Mass. (617) 649-7472 (home) | Customer:<br>Ron Melanson DRC<br>(617) 481-9511 Ext. 6419 |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** | | 512 x 512 x 1 bit plane (no sense)<br>1024 x 1024 x 1 bit plane<br>Maximum of 16 planes | $2000 to $2200<br>$2500 to $2750 |
| **PROCESSOR (SEE ATTACHED)** | | Intel 8086 - up to 1 Mega word user programmable using Forth language | |
| **MEMORY ACCESS** | | **ONE PIXEL I/O AFTER INITIALIZATION** ≤ 1 Ms | |
| **REFRESH RATE** | | (256x256) 60 Hz (256x256) 30 Hz (512x512) 60 Hz (512x512) 30 Hz (1024x1024) 30 Hz | |
| **INTERLACE** | | 1:1 2:1 1:1 2:1 2:1 | |
| **HOST** | | **PDP 11/70** DMA - DR11B interface | $ 850 |
| **INTERFACES** | | **VAX** Same | $ 850 |
| **PERIPHERALS** | | keyboard with 10 function keys | $ 900 |
| | | lightpen | $ 995 |
| | | 45 function keys $835 90 function keys | $1470 |
| | | joystick | $ 690 |
| | | trackball | $2895 |
| **FACILITIES** | H | Vector & Circle Generator (10x firmware speed?) | $3500 |
| **H-HARDWARE** | F & H | Zoom Control (2x, 4x, 8x, 16x) and Scroll | FQ |
| **F-FIRMWARE** | H | Alphanumeric Channel Module | |
| **S-SOFTWARE** | H | Cursor included with device controller | |
| **U-USER/PROCESSOR** | H | Color Table 2048 x 12 (or 4096 x 6) 4 per RGB<br>+Additional modules to provide 8 per RGB | ≈ $3000 |
| | F & S | AYGRAF (4 different versions) SIGGRAPH/CORE Version<br>-polygon fill with firmware<br>-Z-depth sort of polygon filled areas (hidden surface)<br>-curve fitting & generation, conics, polar coordinates<br>-color control with percent of hue, intensity & saturation<br>-2D & 3D manipulation and windowing | $7500 |
| **MONITORS** | | (8024) - 13" diagonal - 800 TV lines | $3075 |
| | | (8025) - 19" diagonal - 900 TV lines | $2395 ← |
| | | (8026) - 19" diagonal - 1000 TV lines | $7435 |
| **NTSC ENCODER** | | | |
| **HARD COPY FACILITY** | | | |
| **COST OF BASIC SYSTEM** | | with 8 bit per RGB color table | $30,000 |
| **DESIRED EXTRAS** | | keyboard, lightpen, joystick, vector generator<br>zoom controller w/scroll, AYGRAF CORE | ≈$42,500 |
| **TOTAL COST** | | | |
| **COMMENTS** | | upgrade to ultra-high (1024x1024) resolution requires:<br>-change some PROM chips, firmware & possibly backplane<br>-monitor (changes are supposed to be minor) | |

Block Diagram

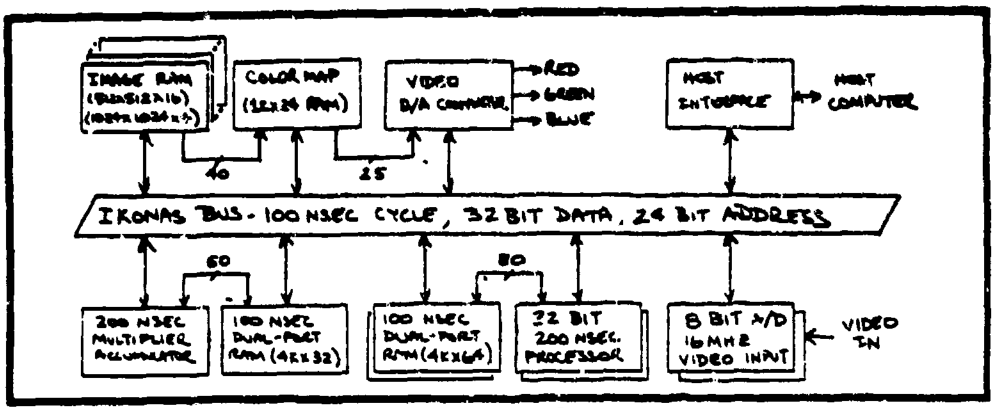


| | | | |
|---|---|---|---|
| **SYSTEM**<br>Norpak VDP | | **CONTACTS** Bill Lalond<br>Pakenham (Ottawa) 1-613-624-5507, 5570 | |
| **IMAGE MEMORY**<br>**CONFIGURATIONS AND COSTS** | | Two basic configurations<br>newer: 512 x 512 x 4 for 512 or 1024 displays<br>older: 1024 x 512 x 1 for 512 displays only | <br>$3500<br>~$1750 |
| **PROCESSOR (SEE ATTACHED)** | | Bit slice micro-Fortran calls to access micro-instructions<br>Not user programmable - can down load somewhat | |
| **MEMORY ACCESS** | | **ONE PIXEL I/O AFTER INITIALIZATION** 1.5 Ms | |
| **REFRESH RATE** | | (512) 25, 30 Hz 50, 60 Hz (1024) 25, 30 Hz | |
| **INTERLACE** | | 2:1 1:1 2:1 | |
| **HOST** | | **PDP 11/70** Modified DR11B - DMA - Limited to DEC(s) | $2800 |
| **INTERFACES** | | **VAX** | |
| **PERIPHERALS** | | keyboard with numeric/cursor pad, 32 function switches, 8 lights<br>trackball<br>joystick<br>tablet<br>touch sensitive display | $1300<br>$3300<br>$1900<br>$2000<br>(future) |
| **FACILITIES**<br>H-HARDWARE<br>F-FIRMWARE<br>S-SOFTWARE<br>U-USER/PROCESSOR | F | Firmware included<br>-Points, vectors, arcs, polygons, text ~1.5 Ms$^{+}$ per pixel<br>-Scroll (w/hardware) - each bit plane separately<br>-Zoom 2x, 4x, 8x (w/hardware)<br>-can use to do subwindows on screen (vwport)<br>-Polygon fill and area fill<br>-Run-length encode and decode<br>-Cursor in overlay | |
| | H | Color table 256 x 24 display and 256 x 4 overlay incl. | |
| | S | Fortran callable routines to access micro-instructions | $ 350 |
| | S | SIGGRAPH CORE - not complete, waiting on SIGGRAPH | |
| **MONITORS** | | Recommend Conracu | |
| **NTSC ENCODER** | | | |
| **HARD COPY FACILITY** | | Micro-controlled interactive input & Rs232 output firmware drive | $3500 |
| **COST OF BASIC SYSTEM** | | | -$35,000 |
| **DESIRED EXTRAS**<br>**TOTAL COST** | | keyboard, joystick, tablet, Fortran interface,<br>input/output drive (need for peripherals) | ~$44,000 |
| **COMMENTS** | | System is not really completed as yet, hard to pin down. | |



IP 5000 SYSTEM

| | | | |
|---|---|---|---|
| **SYSTEM**<br>DeAnza IP5000 | | **CONTACTS** Rick Pizza Chuck Nordby<br>San José, CA (408) 263-7155 | |
| **IMAGE MEMORY**<br><br>**CONFIGURATIONS AND COSTS** | | All systems are essentially 512 x 512 x 24 bits<br>System #1 2 channels (1 scratch) monochrome 8 bits<br>System #2 3 channels RGB 8 bits per color gun<br>System #3 4 channels (1 scratch) RGB w/3 overlay planes | |
| **PROCESSOR (SEE ATTACHED)** | | Pipe-line Array Processor - user programmable | |
| **MEMORY ACCESS** | | **ONE PIXEL I/O AFTER INITIALIZATION** 800 Ns | |
| **REFRESH RATE** | | 30 Hz | |
| **INTERLACE** | | 2:1 | |
| **HOST** | | **PDP 11/70** Treated as virtual memory-off UNIBUS by use of registers. | |
| **INTERFACES** | | **VAX** | |
| **PERIPHERALS** | | joystick w/interface | $ 875 |
| | | trackball w/interface | $3450 |
| | | lightpen w/interface | $2950 |
| **FACILITIES** | H | Vector generator - 2.5 Ms (estimate) per pixel | |
| **H-HARDWARE** | H | Zoom and Scroll (zoom 2x, 4x, 8x) | |
| **F-FIRMWARE** | H | Color Maps 3 256 x 24 maps - display only one | |
| **S-SOFTWARE** | | -ITU - Image Translator - secondary color control | |
| **U-USER/PROCESSOR** | H | Cursor | $1400 |
| | F | Image processing functions ... for example:<br>-can add two 512 x 512 x 8 images in 1/30 sec.<br>-multiply two 512 x 512 x 8 images in 3/10 sec.<br>-can split screen with separate look-up tables | ($4600) |
| | | Alphanumeric overlay generator | $1000 |
| **MONITORS** | | | |
| **NTSC ENCODER** | | N/A | |
| **HARD COPY FACILITY** | | Recommend Dunn | |
| **COST OF BASIC SYSTEM** | | | $4[illegible],000 |
| **DESIRED EXTRAS** | | joy_tick, lightpen, ITU, alpha/num generator overlay | $48,025 |
| **TOTAL COST** | | | |
| **COMMENTS** | | | |



Typ:cal System Block Diagram

## PROCESSOR

The IKONAS processor is fully user programmable A fast, 32 bit wide architecture gives unparalleled precision for graphics and image processing applications The IKONAS processor speeds image computation by executing many repetitive, time consuming calculations from microcode programs. Graphics and image processing performance is 'urther enhanced by allowing the host computer direct access to the image memory as well as to any other memory on the IKONAS bus (color look-up table, microcode store, etc.).

## IMAGE MEMORY

IKONAS Image Memory is bit plane organized Each module can be addressed as 1024x512x1. 512x512x2, or, for multi-pixel access, as 10Kx32. Pan, and scroll in pixel increments are standard as is zoom to any integer ratio 1 1 to 256.1 Modular nature of the units allow memory to be easily expanded from 512x512x2 up to 512x512x32 or 1024x1024x16.

## FAST HARDWARE MULTIPLIER

The multiplier accumulator module facilitates the rapid execution of many graphics and image processing tasks which require mu..iply then add or subtract cycles, e g matrix multiplication (3-D point transformation), vector dot and cross product (shade calculations), and weighted averaging (anti-aliasing) Four modules operating in parallel allow sub-microsecond 3-D point transformation

## VIDEO INPUT

Video signals may be written into the image memory in real time The high speed bus architecture of the IKONAS system allows simultaneous 10 Mbyte/sec video input, 10 Mbyte/sec video output, and 2 Mbyte/sec host data transfer

## ANIMATION

Computer graphics animation is a fast developing field with applications in physical system modeling, display of time varying data, and cartooning IKONAS systems support computer animation using color-map or run-length encoding techniques with a variety of color look-up tabler and run-length decoders Image Memory serves as a run-length animation buffer for encoded images as well as frame buffer for unencoded images. The Mass Image Storage module can hold up to 60 seconds of moderately complex animation for real-time playback or can be used to store unencoded images

## FLEXIBILITY, EXPANDABILITY

IKONAS systems are entirely modular being configured from various modules attached to a common communication bus Systems are easily expanded One cage holds 20 cards, multiple cage configurations are possible A user can begin with a simple frame buffer and add processor, image input, and hardware multiplier modules later

## CUSTOMIZED SYSTEMS

Modular design of components means that systems are configured to meet a customer's particular needs Extensive use of microprogrammed controllers in the modules means that custom modifications are easily performed in many cases A wide variety of options is available IKONAS is particularly interested in providing state-of-the-art hardware for research and special purpose graphics and image processing systems

| | | | |
|---|---|---|---|
| **SYSTEM**<br>IKONAS | | **CONTACTS** Mary Whitten or Nick England<br>Raleigh, NC (919) 813-5401 | |
| **IMAGE MEMORY** | | | |
| **CONFIGURATIONS AND COSTS** | | 1024 x 512 x 1<br>Max of 20 cards | $2000 |
| **PROCESSOR (SEE ATTACHED)** | | Big and fast 32 bit 200 Ns Microprocessor<br>(almost 50% cost of the total system) | |
| **MEMORY ACCESS** | | **ONE PIXEL I/O AFTER INITIALIZATION** 400 Nsec access<br>100 Nsec on bus (1 cycle) ... 4 cards can operate in alt. 100 Ns | |
| **REFRESH RATE** | | (512) 30 Hz (512) 60 Hz (512) 50 Hz (512) 100 Hz (1024) 30 Hz | |
| **INTERLACE** | | 1:2 1:1 1:1 1:2 | |
| **HOST** | | **PDP 11/70** DMA via DR11B | $3000 |
| **INTERFACES** | | **VAX** Same | |
| **PERIPHERALS** | | none as yet - peripherals are hung off the host | |
| **FACILITIES** | F | (Pan) Scroll & Zoom (1x, 2x, 3x, 4x, 5x, ... 256x) | included |
| **H-HARDWARE** | H | Run-[illegible] Encode/Decode ... animation possible | $2400 |
| **F-FIRMWARE** | F | Win[illegible]nd Viewporting | |
| **S-SOFTWARE** | H | Cursor | |
| **U-USER/PROCESSOR** | H | Color Table 1024 x 24 Low Speed: $2000 High Speed: $2800<br>(Required for 1024 x 1024) | |
| | U | Possible things to look for:<br>-fast vector generation (NASA)<br>-"real time" hidden-line/surface (NASA)<br>-2D & 3D model manipulation<br>-Edge detection and anti-aliasing | |
| **MONITORS** | | | |
| **NTSC ENCODER** | | Recommend Lenco | |
| **HARD COPY FACILITY** | | Recc mend Dunn | |
| **COST OF BASIC SYSTEM** | | | ~$45,000 |
| **DESIRED EXTRAS** | | RLE Encoder/Decoder | |
| **TOTAL COST** | | | ~$48,000 |
| **COMMENTS** | | "Look for good things from Ikonas ..." (AED engineer)<br>-advertised as tool for graphics research<br>-no high-level language interaction<br>-upgrade to ultra-high resolution requires one software bit change! | |

The following paragraphs briefly describe the various elements of the RM-9400 Display Generator

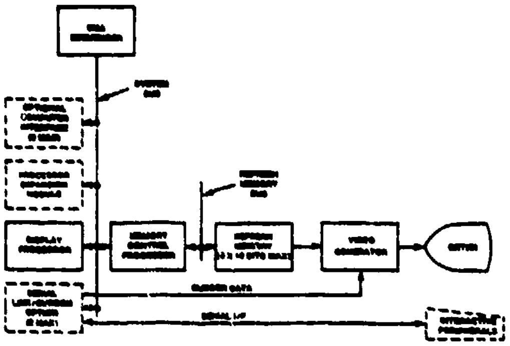


**FIGURE 1. RM-9400 FUNCTIONAL BLOCK DIAGRAM**

**· Computer Interface (RM-9000-XX)**
The Computer interface provides a high-speed link between the host computer and the RM-9400 Display Generator A general purpose interface (GPIF) is provided on the Display Processor Two additional card slots are reserved for custom interfaces Off-the-shelf interfaces are available for most minicomputers and some large mainframes All interfaces are 16-bit parallel Most incorporate or utilize direct memory access

**· DMA Sequencer**
The TTL DMA Sequencer performs high-speed non-processor transfers involving multiple devices on the System Bus (for example between the Computer interface and Display Processor or Memory Control Processor The DMA Sequencer can involve as many as 14 ports and seven subloops

**· Display Processor**
The Display Processor directly or indirectly controls each element of the display system In addition it decodes received instructions stores subpictures (command lists) and fonts performs coordinate transformations and drives the Memory Control Processor The Display Processor contains a Z80 microprocessor with 32K bytes each of EPROM and RAM a GPIF interface three serial ports a timer memory map cycle-stealing DMA and interrupt control logic The memory map accommodates up to 512K memory bytes of which 96K bytes are reserved for internal control software

**· Processor Expansion Module (RM-9400-PEM1 2 3 4)**
The Processor Expansion Module adds a high-speed math unit up to 32K bytes EPROM potential and additional user RAM to the Display Processor Memory expansion may be specified in 32K byte increments to a maximum of 128K bytes (where n = number of 32K byte increments)

**· Memory Control Processor**
The Memory Control Processor draws primitives alphanumerics graphics images etc into the refresh memory and performs clipping entity detection pan and zoom The MCP contains a special-purpose 16-bit bipolar microprocessor with dedicated ROM RAM and support logic

**· Refresh Memory (RM-Y/X Z)**
The Refresh Memory consists of solid-state MOS RAM s that store the picture(s) in raster scan dot matrix format The memory is organized as one to eight groups of up to 16-bits each Each 16-bit cell defines a single pixel on one or more CRT's Table 1 lists the possible resolutions aspect ratios, and refresh frequencies

**TABLE 1—RM-9400 RESOLUTION TABLE**

| Controller Model (RM-9400/nX) | Spatial Resolution | | Aspect Ratio (X:Y) | | Refresh Frequency (Frame Rate) |
|---|---|---|---|---|---|
| | Lines | Elements | Raster | Pixel | |
| RM-9400-1X | 256 | 640 | 4 3 | 2 1 | 50 60Hz Repeat Field |
| RM-9400-4X | 512 | 512 | 1 1(1) | 1 1 | 50 60Hz Repeat Field |
| RM-9400-5X | 512 | 640 | 4 3 | 1 1 | 50 60Hz Repeat Field |
| RM-9400 6X | 512 | 1024 | 1 1(1) | 2 1 | 50 60Hz Repeat Field |
| RM-9400 7X | 512 | 1280 | 4 3 | 2 1 | 50 60Hz Repeat Field |
| RM-9400 8X | 1024 | 1024 | 1 1(1) | 1 1 | 25 30Hz Interlaced |
| RM-9400 9X | 1024 | 1280 | 4 3 | 1 1 | 25 30Hz Interlaced |

Note (1) Active raster is centered within 4 3 CRT aspect ratio

**· Serial Link/Cursor Option (RM-9400-SLC2/4)**
The Serial Link Cursor option processes operator input from keyboards and graphic input devices. and generates two or four independent cursors that can be used to point to the face of the display without affecting the data in refresh memory The RM-9400-SLC consists of a Z80 microprocessor with dedicated ROM and RAM, four or eight serial ports and two or four 32 x 32 programmable cursor generators Support software is available for keyboards, joysticks trackballs light pens and graphic tablets.

**· Video Generator (RM-9400-Vn)**
The Video Generator transforms the stored pictures into industry compatible video signals that drive Ramtek or other commercially available high resolution CRT monitors large screen projectors and hardcopy printers All outputs are compatible with EIA Std RS-170 or RS-343-A specifications for composite video

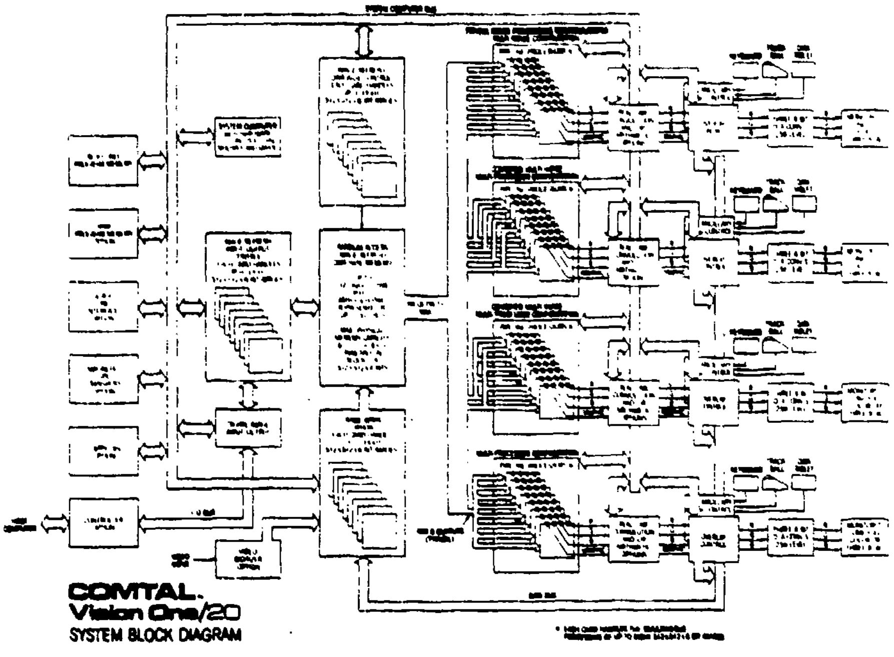
The video generators process data on a pixel-by-pixel basis through PROM or RAM defined lookup tables that assign output color and or intensity Each pixel indexes the lookup table as it is scanned from the refresh memory The contents of the addressed cell in the lookup table are then passed to the digital-to-analog converters (DAC) or video amplifiers that produce the output video signals

Cursor and overlay mixing is performed either in the lookup table or at the DAC by clamping the output voltage to minimum or maximum scale All video generators incorporate a blink frequency generator that allows selective blink

There are three off-the-shelf video generators that satisfy most applications

A The Type I Video Generator (RM-9400-V1) is designed for general graphics applications The RM-9400-V1 drives 12 two-bit (4-level) video outputs to 12 monochrome or four RGB color displays In addition the RM-9400-V1 provides hardware blink and mixes up to four independent cursors with any of the 12 output channels Color intensity overlay and blink assignment are accomplished by PROM coding Any of 64 colors or four intensities may be specified

| | | | | |
|---|---|---|---|---|
| **SYSTEM**<br>Ramtek 9400 | | | **CONTACTS** Ken Mullany<br>Boston, Mass. (617)862-7720  Sunnyvale, CA (408)735-4800 | |
| **IMAGE MEMORY** | | | 512 x 512 x 1 plane | $ 469 |
| | | | 1024 x 1024 x 1 plane | $2345 |
| **CONFIGURATIONS AND COSTS** | | | Maximum of 8 planes per chassis | |
| | | | (add $2000 for larger chassis) | |
| **PROCESSOR (SEE ATTACHED)** | | | Z-80 traffic controller ... not recommended for user programming | |
| **MEMORY ACCESS** | | | **ONE PIXEL I/O AFTER INITIALIZATION** 1.12 Ms | |
| **REFRESH RATE** | | | (512) 25, 30 Hz   50, 60 Hz   (1024) 25, 30 Hz | |
| **INTERLACE** | | | 2:1   1:1   2:1 | |
| **HOST** | | | **PDP 11/70** DR11B & C (will also quote Mass bus in future) | $2200 |
| **INTERFACES** | | | **VAX**   Similar | |
| **PERIPHERALS** | | | keyboard | $1500 |
| | | | tablet | $2000 |
| | | | joystick | $1400 |
| | | | lightpen | $2900 |
| | | | trackball | $3000 |
| **FACILITIES** | | H | Vector Generator • 16,000 Vectors/sec (50 pixels/vector) | |
| **H-HARDWARE** | | H | Zoom (2x, 3x, 4x, 5x ... 16x) and scroll | |
| **F-FIRMWARE** | | H | Color table (2) 102 x 16:4 per RGB and 4 monochrome hard copy | |
| **S-SOFTWARE** | | | → Second video board for 8 bits per RGB | $6120 |
| **U-USER/PROCESSOR** | | S | Arcs, Circle fill and Polygonfill | $ 500 |
| | | F | area fill | |
| | | F | 2D translation, rotation and scaling | $ 500 |
| | (S) | H | Viewporting | |
| | (S) | H | Decluttering - increased detail with zoom | |
| | (S) | H | Entity detect into display list and return to host | |
| | (S) | H | Down-load display list board w/list processing | $3280 |
| | | | -gives power to directly address image data commands | |
| **MONITORS** | | | 512 - 60 Hz  $3970   1024 - 30 Hz  $10,200 | |
| **NTSC ENCODER** | | | | |
| **HARD COPY FACILITY** | | | Poloroid system  $12,000  35 mm camera & adapter  $4000 | |
| **COST OF BASIC SYSTEM** | | | with 8 bits per RGB in color table | $41,100 |
| **DESIRED EXTRAS** | | | keyboard, tablet, joystick, lightpen, polygon filler, | |
| **TOTAL COST** | | | 2D transformations, download display list processor | $53,100 |
| **COMMENTS** | | | -Maintenance and training course - 4 weeks in CA | $ 1,000 |
| | | | -Upgrade to 1024 requires sync and backplane change,<br>and possibly return to shop | ~$ 2,000 |



COMTAL.
Vision One/20
SYSTEM BLOCK DIAGRAM

# Vision One/20 Technical Specifications

□ Displayed spatial resolutions of 512x512 or 1024x1024, guaranteed to meet or exceed test specifications defined in "Quantitative Evaluation of Soft Copy Displays."

- Up to 512 separate bits can represent each and every picture element in a 512x512 display presentation (128 bits per picture elements in a 1024x1024 display).
- Up to 134,000,000 bits of image refresh data have memory available in a single 6-foot electrical cabinet
- Every one of the 134 000 000 bits of the refresh memory are read out for display in 1/60 of a second
- Enough image refresh data base memory is available to allow a complete 4096x4096x8 bit array to be viewed in real-time
- Any one bit of the refresh memory can be randomly addressed and read out in 800 nanoseconds.
- Dynamic assignment of image memory for either the representation of brightness increments of an image or one bit dot map overlays
- Image refresh memory may be arrayed in spatial configurations of 512x32,768 or 32,768x512 picture elements or any other configuration in a 32,768x32,768 space
- Refresh memory configuration assignment completely dynamic and entirely under firmware control
- Full color, high fidelity color presentation (up to $2^{24}$ brightness levels image refresh and display) available in all of the spatial configurations mentioned above
- Full 512x512 resolution real-time roam of a large data base is provided in a moving window presentation with no restrictions on the direction or rate of movement of the window presentation across the refresh memory data base
- Zoomed presentation of any 256x256 or 128x128 picture element area of the refresh memory data base with full window capability as mentioned above
- Loop movie presentations of up to 64 separate 512x512 spatial resolution frames, 256 separate frames at 256x256 spatial resolution and 1024 separate frames at 128x128 spatial resolution Each frame may have up to 256 brightness levels
- Completely independent use of the refresh memory data base by up to 4 users. Each user supplied with separate keyboard control and independent full color video output
- Allocation of portions of the refresh memory data base dynamically assignable between users
- Refresh memory data base data is loaded completely independent of display presentation (dual-ported construction)
- Complete random addressability to a single picture element
- Automatic block transfer of image data provided with the ability to load sequentially from either side to either side or top to bottom or bottom to top
- Freeze frame transfer of image data synchronously into image memory at rates of 1/30 of a second
- Real-time rewriting of the refresh memory data base on the basis of processing algorithms in the output section of the display
- Iterative re-processing of the refresh stored data through system contained processing algorithms performing true "pipeline" processing of the refresh stored data with each processing step taking 1/30 of a second
- Image combining capability on the basis of plus, minus, multiply and divide
- Real-time black and white or full color image composition — allowing the non-destructive super imposition of regular or irregular shaped portions of images one upon the other with complete freedom of nondestructive translation of the super imposition section in any direction. The resulting composition can be instantly used to form and store an entirely new image

| | | |
|---|---|---|
| **SYSTEM**<br>COMTAL Vision One/20 | **CONTACTS** Harvey Raider<br>(213) 797-1175 Pasadena, CA | |
| **IMAGE MEMORY** | | |
| **CONFIGURATIONS AND COSTS** | incremental by 512 x 512 x 8 bit image groups<br>maximum of 64 groups = 512 bits per pixel | $8,000 to $10,000 |
| **PROCESSOR (SEE ATTACHED)** | Pipeline processor to recompute all picture elements<br>LSI micro to handle user interaction, system response, memory management | |
| **MEMORY ACCESS** | **ONE PIXEL I/O AFTER INITIALIZATION** 1.5 Ms (read 900 Ns) | |
| **REFRESH RATE** | 30 Hz 60 Hz | |
| **INTERLACE** | 2:1 1:1 | |
| **HOST** | **PDP 11/70** RSX11-M is available. Unibus board & DR11B | $3150 |
| **INTERFACES** | **VAX** Similar | |
| **PERIPHERALS** | keyboard<br>trackball<br>data tablet<br>magnetic tape transport<br>floppy disk | |
| **FACILITIES**<br>H-HARDWARE<br>F-FIRMWARE<br>S-SOFTWARE<br>U-USER/PROCESSOR | High powered image processing facilities - upper range<br>-can roam a 4096 x 4096 x 8 bit array<br>-134,000,000 bits of refresh memory read in 1/60 sec.<br>-animation of 64 512 x 512 x 8 images in memory<br>-real time (1/30 sec) image processing features<br>↓<br>(list is available)<br>↓ | |
| **MONITORS** | 19" high quality monitor included | |
| **NTSC ENCODER** | | |
| **HARD COPY FACILITY** | | |
| **COST OF BASIC SYSTEM** | | approx. $40,000 |
| **DESIRED EXTRAS** | Prices from approx. $40,000 to $700,000 | |
| **TOTAL COST** | | |
| **COMMENTS** | -computer built into system<br>-top-of-the-line for image processing | |

## Appendix B
## SURVEY UPDATES

Building 37, Room 509
September 26, 1979

To: V.H. Lucke, Design Graphics Personnel, J.F. Berkery, W.E. Lorensen,
J.L. Mundy, R.B. Saltzman

Subject: Update #1
Survey of Color Video Frame Buffer Systems

1. In a resent telephone conversation with Ken Anderson, of the Anderson Report, I found that no further information had been uncovered regarding Seiko's (Tokyo) digital TV display. Rumor had it that Seiko had utilized a 4 x 4 transformation matrix similar to the Evans and Sutherland Picture System II with a 512 x 512 full color frame buffer system.

2. Mr. Anderson did give me a name to contact at DEC in Nashua New Hampshire regarding their rumored 512 x 512 color video frame huffer system. DEC's System Processor is based on 2901 archietecture utilizing a 160 Ns cycle time. The system will act as a device on the Unibus with a parallel interface. The initial system will be able to draw 50,000 vectors (short or inch?) per second, but very few other facilities will be offered and the processor will not be user programable. The initial system will offer a maximum of 4 512 x 512 image memory planes, with a 19 inch color monitor and interface for approximately $14,000. It will be available for shipment around June 1980 and development will continue to improve the system.

Peter Atherton
37-509
8-1693

# MISSION
# of
# Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence ($C^3I$) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.