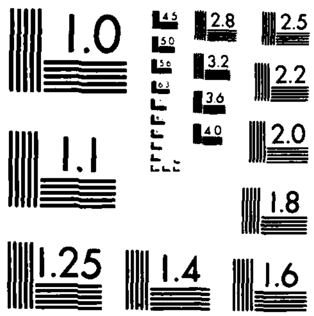
AD-A142 173 THE DEVELOPMENT OF A MATHEMATICAL FOUNDATION FOR CELLULAR IMAGE PROCESSING(U) FLORIDA UNIV GAINESVILLE DEPT OF MATHEMATICS G X RITTER FEB 84 AFOSR-TR-84-0472 1/1

UNCLASSIFIED AFOSR-83-0065 F/G 14/5 NL

END
DATE
FILMED
7-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

U
SECURIT

AD-A142 173

DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 84-0472
6a. NAME OF PERFORMING ORGANIZATION: University of Florida / 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code): Department of Mathematics, Gainesville, FL 32611	7b. ADDRESS (City, State and ZIP Code): Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR / 8b. OFFICE SYMBOL (If applicable): NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR-83-0065
8c. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	D-9	

11. TITLE (Include Security Classification): The Development of a Mathematical Foundation for Cellular Image Processing

12. PERSONAL AUTHOR(S): Gerhard X. Ritter

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
FINAL	FROM 1 MAR 83 TO 29 FEB 84	Feb 84	19

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

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

The primary result of this research effort has been the development of an image algebra that can serve as the foundation of a common algebraically based image processing language. In comparison to other existing image algebras, this algebra is capable of expressing common image processing algorithms and transforms in terms of its operators.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr. Robert N. Buchal	22b. TELEPHONE NUMBER (Include Area Code): - 4939 / 22c. OFFICE SYMBOL: NM

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

84 06 18 167

SECURITY CLASSIFICATION OF THIS PAGE

AFOSR-TR. 84-0472

The Development of a Mathematical Foundation

for Cellular Image Processing

Final Technical Report

Air Force Grant No. AFOSR-83-0065

3/1/83 through 2/29/84

Principal Investigator:

Gerhard X. Ritter

Department of Mathematics

University of Florida

Gainesville, Florida 32611

84 06 18 167

Abstract

The primary result of this research effort has been the development of an image algebra that can serve as the foundation of a common algebraically based image processing language. In comparison to other existing image algebras, this algebra is capable of expressing common image processing algorithms and transforms in terms of its operators.

Introduction

Current image processing algorithm development is not based on an efficient mathematical structure that is designed specifically for image manipulation, feature extraction and analysis. In general, each researcher develops his own set of ad-hoc image processing tools, thereby increasing research and development costs accordingly. The vast increases in image processing activities throughout the military, industrial and academic communities are resulting in an immense proliferation of different operations and architectures that all too often perform similar tasks. There are probably as many image processing languages as there are architectures, and all of them differing in capabilities.

In view of this ever-increasing diversity of image processing architectures and languages, the principal investigator proposed to develop a standard image processing algebra to serve as a mathematical basis for a common image processing language. The relational formalism of an algebraic image processing language would constitute an invaluable aid in the design, development, optimization and testing of image processing algorithms and hardware configurations.

This research effort succeeded in defining a universal image algebra that could serve as the origin from which a common image processing language could evolve. As compared to other existing image algebras, no problems have been encountered in translating common image processing operations into the language of this algebra. Because of this enormous success, research efforts are now continuing on four fronts: (1) Further theoretical development of the algebra; (2) Formulation and compilation of a new image processing language based on this algebra; (3) Optimization of FLIR algorithms; (4) Design of reconfigurable VLSI architectures for image processing based on this algebra.

Technical Aspects of the Research and Research Results

This research, under Air Force grant No. AFOSR-83-0065, has been involved with the development of a rigorous formulation of a mathematical foundation for image processing algorithms and operations. The research proceeded along the following guidelines:

(1) Investigation of existing algebraic structures for image processing.

(2) The cataloguing, according to task similarity of existing image processing operations.

(3) Investigation of the relationships between the basic components of the catalogued operations.

(4) Extraction of a minimal set of operators in order to form an algebra capable of expressing all image transform operations.

(5) Establishment of some basic relationships and theorems governing the algebra.

(6) Investigation of the algebra's potential to serve as a foundation of a common algebraically based image processing language.

The investigation of existing image algebras turned out to be somewhat disappointing. Extracting the mathematical and conceptual core of existing algebras (ref. 6,11,12), resulted in only one structure mathematicians would dare call an "algebra". This structure is equivalent to the Minkowski algebra of sets (4). Although the literature abounds with so called new techniques - i.e. erosions, dilation closings, openings, rolling ball algorithms, etc. - the algebraic relationship provided by these techniques can all be found or easily derived from the algebraic relationships in (4).

Even though many neighborhood operations can be expressed in terms of the Minkowski algebra, the algebra is extremely limited in performance (6), incapable of expressing global transforms and various neighborhood transforms and can, therefore, never serve as a universal image algebra. It became clear at the initial stage of this research that a new algebra needed to be defined. In order to accomplish this task, cataloguing and investigating existing image operations became a necessity.

Because of the "one year" time constraints some image transforms were more thoroughly investigated than others. In particular, emphasis was placed on linear transforms, non-linear smoothing and enhancement techniques, edge detection schemes, image segmentation and background removal. W.K. Pratt's book on digital image processing (7) proved to be the most valuable resource for this particular task.

Investigation of the basic components of the various image transforms made it clear that we were dealing with only four basic operations, namely two arithmetic and two logic operations. However, these four operations are applied differently in cellular (neighborhood) processing than in non-cellular processing, resulting in an algebra of eight binary operators. The definition of these operators can be found in Appendix 2 and details concerning the algebra are given in (8).

In order to demonstrate the potential of this algebra to serve as a basis for a common image processing language, we showed that the algebra is capable of describing all commonly used image processing functions. We considered such diverse processing operations as linear transforms (Fourier transforms, Walsh transforms, etc.), non-linear filtering and enhancement techniques, a variety of well-known edge detection algorithms, gray scale averaging, thresholding, and histogram equalization. In order to minimize the number of operations in the algebraic formulation of algorithms, we also established some basic relationships governing the algebra. As mentioned earlier, no great difficulties were encountered when translating image operations into our algebraic formulation. A special bonus of the translation task was the discovery of some new and powerful image enhancement techniques (9).

Discussions with my colleague Dr. S. Chen of the National Science Foundation led to the discovery that the algebra is an "image processing machine" in the abstract sense, and can thus be used to define and model real architectures. Building on this idea, we defined a language based systems architecture where the algebraic algorithms are expressed as data flow graphs that are mapped to a reconfigurable distributed system (3). In view of recent advances in VLSI technology, such architectures are now feasible.

In our system, the user inputs image data through a front-end computer to a distributed network which leads to various operation modules. The active operation modules drive parallel processing elements that carry out the elementary algebraic operations and transformations. Configurations for our variable neighborhood definition are formed through the control of arbitration networks. Modularity and redundancy will enable the system to be fault tolerant and expandable.

The main improvement of this system over some existing architectures, such as the cytocomputers or the CLIP series, is the ability to handle variable neighborhoods and to perform certain image processing algorithms in parallel that are not feasible on current cellular array computers.

A side benefit of this research was the development of software for the VAX-11/780 computer to enable the printing of gray level images on standard dot-matrix printers. The extremely low resolution, spatial distortion and slowness of the print routines provided by Government furnished print routines necessitated this development. This software will be made available to AFATL personnel at Eglin AFB. Appendix 1 provides examples of the improved image displays.

Finally, this one-year research effort resulted in four publications (1,2,9,10) and four invited talks and lectures. In October 1983, the principal investigator was an invited speaker and session chairman at the IEEE International Conference on Computer Design held in New York. At the 1984 annual Spring meeting of the Mathematical Association of America's Florida section in Tampa, an invited lecture was given on the connection between digital topology and the image algebra. Two talks concerning the algebra were given in April 1984, one at the Conference on Intelligent Systems and Machines in Rochester, Michigan, and the other at the 1984 Southeast Regional ACM Conference in Atlanta, Georgia.

Summary and Recommendations

We constructed an algebra for image manipulations consisting of eight binary operators. All image processing techniques investigated during this research were expressible in terms of this algebra, and it is our opinion that most, if not all, current image manipulation techniques are translatable into the language of this algebra. However, this needs to be more thoroughly documented.

Since very little is known about algebraic structures containing more than two operators, theorems, corollaries, identities, and laws concerning compositions of different operations need to be established. Such laws will not only be useful in terms of algorithm simplification but also provide deeper insight and a better understanding of the algebra. Following the establishment of such relationships and laws, a natural "next step" could be the application of these algebraic relationships for the optimization and testing of Government furnished FLIR algorithms, and also as an aid in the development of new image processing algorithms. The importance of such a fully developed algebra with regard to military applications cannot be overestimated.

References

1. L.A. Ankeney and G.X. Ritter, "Applications of Cellular Topology in Image Processing," Int'l J. of Computer and Inf. Science, 12(6) 1983, 433-456.

2. S. Chen and G.X. Ritter, "Image Processing Architectures and Languages," Proc. IEEE Int'l Conf. on Computer Design: VLSI in Comp. New YOrk, 1983, 723-726.

3. S. Chen and G.X. Ritter, "A Reconfigurable Architecture for Image Processing," pre-print (submitted for publ.)

4. H. Hadwiger, Vorlesungen über Inhalt, Oberfläche und Isoperimetrie, Springer-Verlag, Berlin, 1957.

5. P.E. Miller, An Investigation of Boolean Transformations, Ph.D. Thesis, Ohio St. University, Columbus, OH, 1978.

6. P.E. Miller, "Development of a Mathematical Structure for Image Processing," Perkin-Elmer Optical Div., TR, Danbury, CT, 1983.

7. W.K. Pratt, Digital Image Processing, Wiley and Sons, New York, 1978.

8. G.X. Ritter, "On the Foundations of a Common Image Processing Algebra," TR, AFATL, EO-Terminal Guidance Branch, Eglin AFB, 1983.

9. G.X. Ritter, "The Algebra of Images," Proc. Conf. on Intelligent Systems and Machines, Oakland U. of Rochester, MI, 1984.

10. G.X. Ritter, "The Language of Massively Parallel Image Processing Computers," Proc. Southeast Reg. ACM Conference on Friendly Systems, Atlanta, GA, 1984.

11. J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London, 1982.

12. S. Sternberg, "Languages and Architectures for Parallel Image Processing," Proc. Conf. on Pattern Recognition in Practice, North-Holland Publ., 1980.

Appendix 1

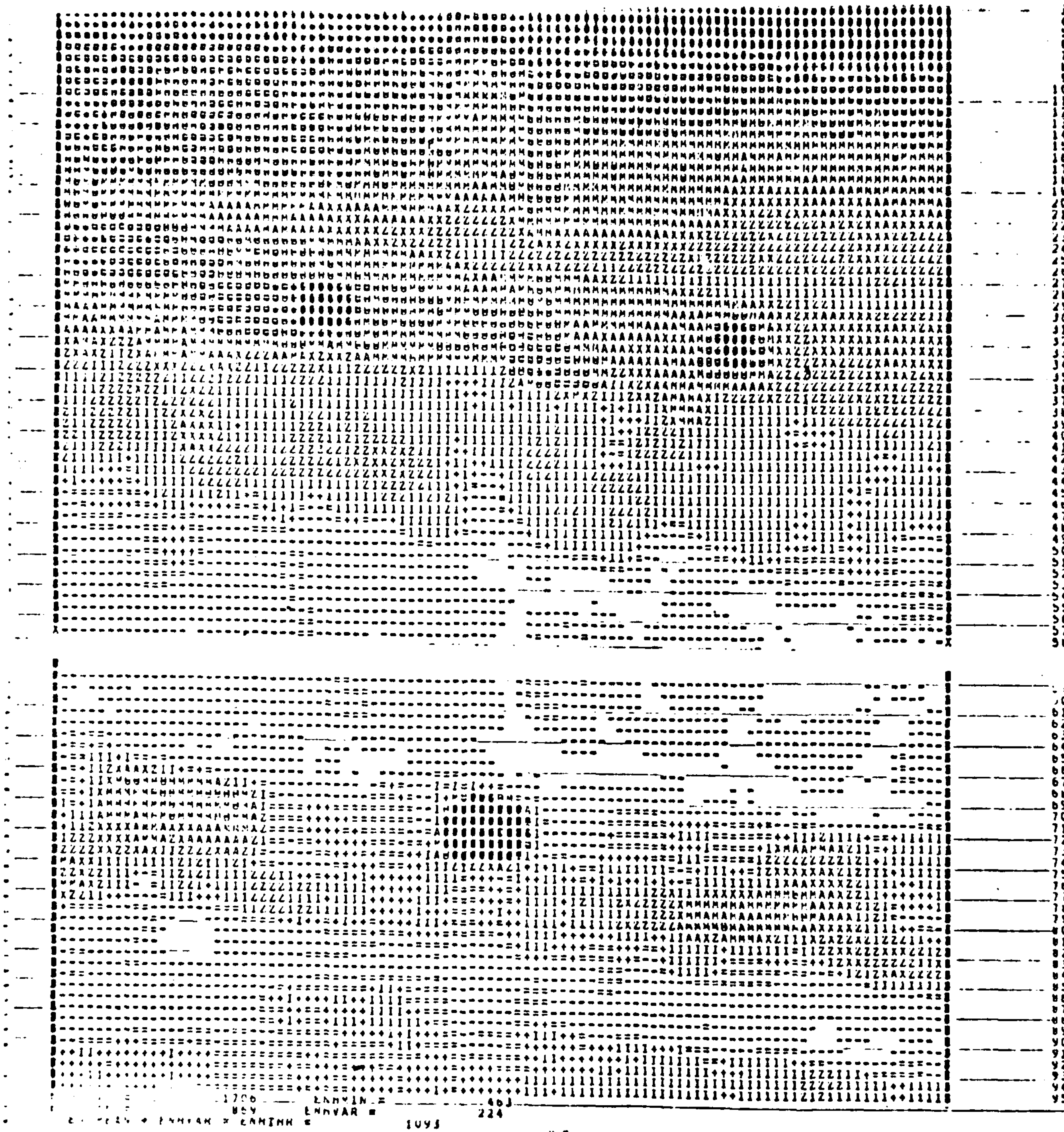
::: GRAY LEVEL IMAGE :::

```
ENHMAX  =               1706     ENHMIN  =              463
ENHMEAN =                869     ENHVAR  =              224
  ENHMEAN + ENHVAR = ENHTHR =              1093

THE FOLLOWING IS THE ENHTHR THRESHOLD IMAGE #  1
```

FLIR Image #1

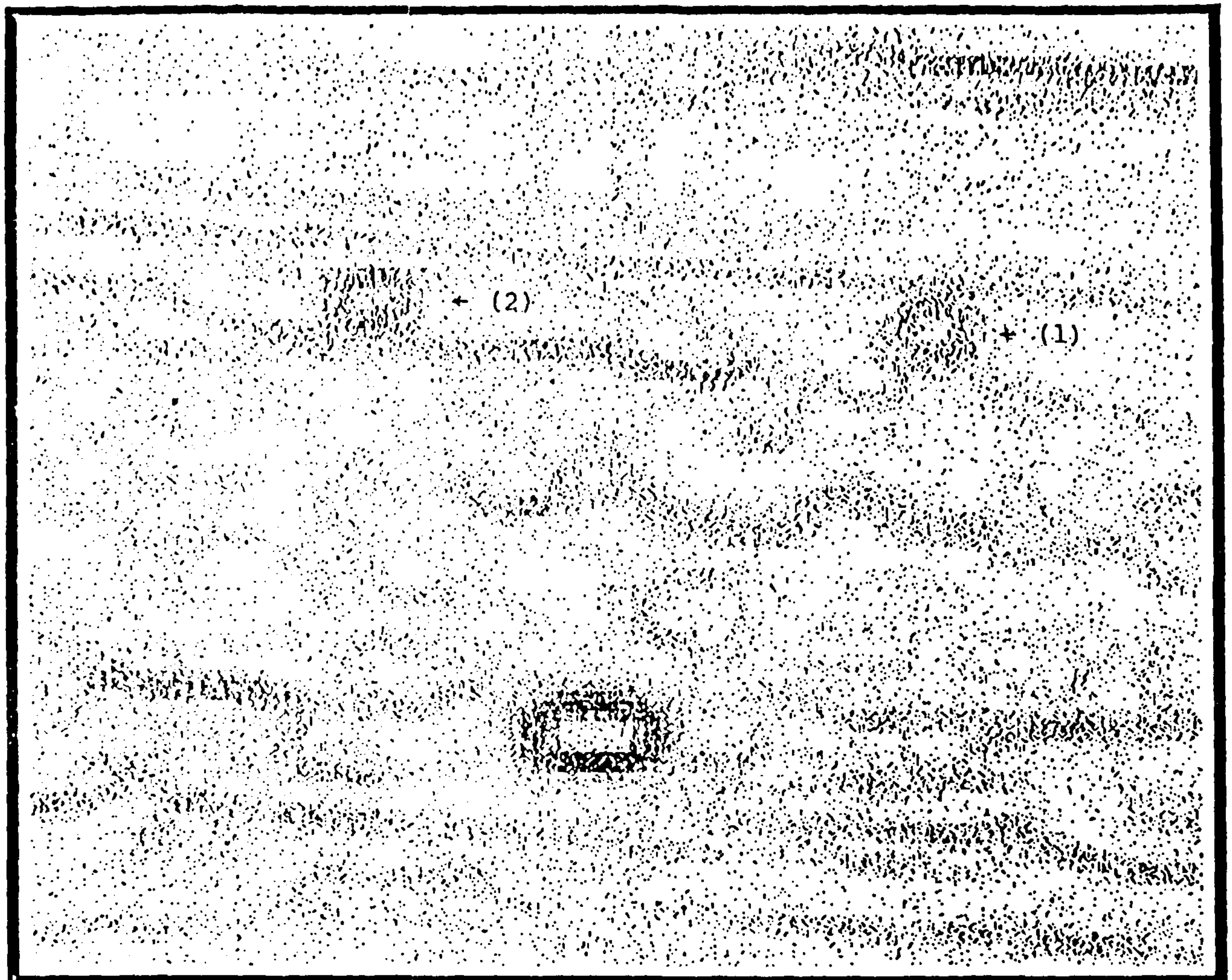
University of Florida



FLIR Image #1

Eglin AFB

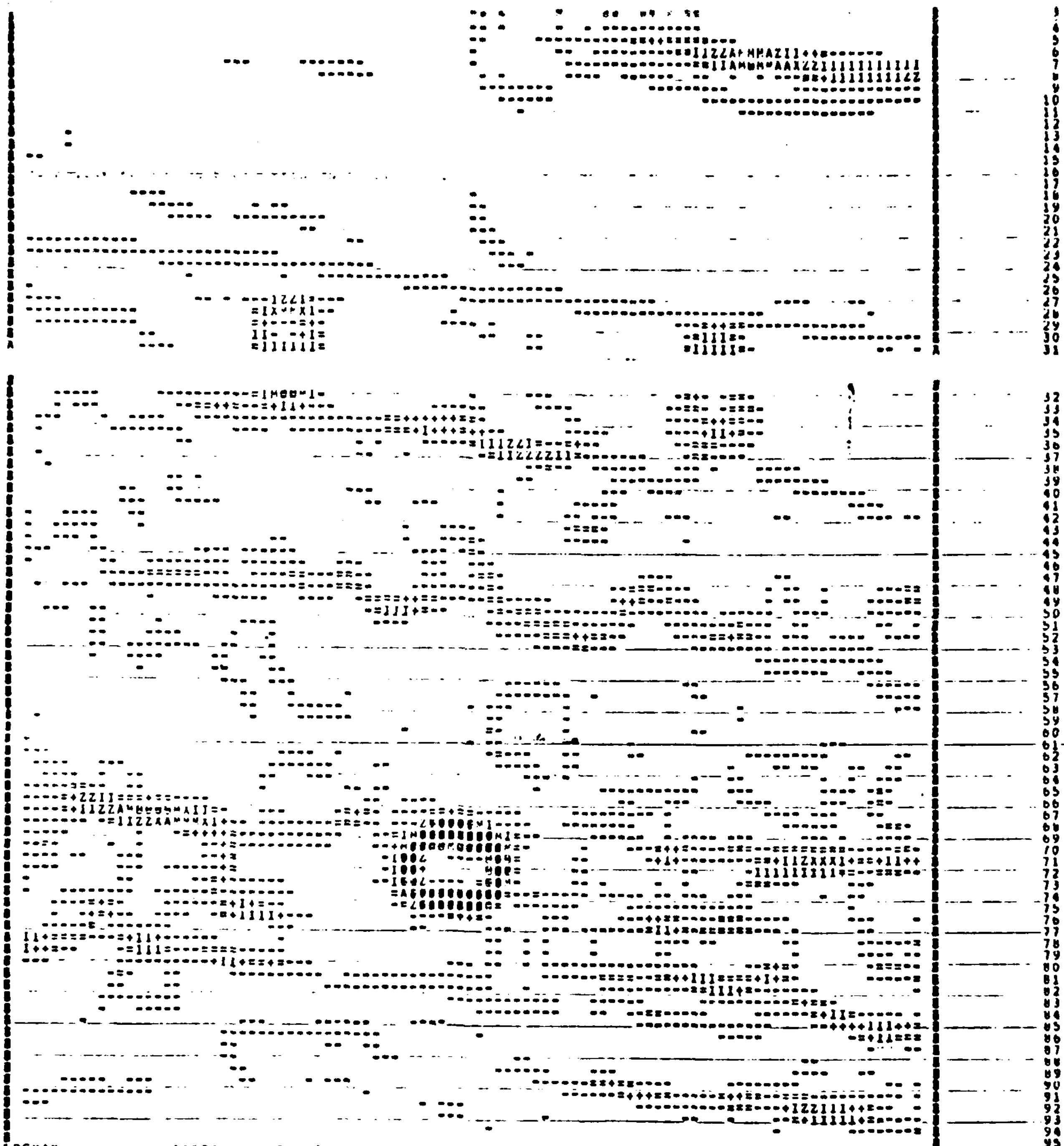
::: GRAY LEVEL IMAGE :::



EDGMAX = 11094 EDGMIN = 27
EDGMEAN = 928 EDGVAR = 1035
EDGMEAN + EDGVAR = EDGTHR = 1963

THE FOLLOWING IS THE EDGTHR THRESHOLD IMAGE # 1

FLIR Image #2, University of Florida
Note the targets labeled (1) and (2)
and compare with FLIR Image #2, Eglin
AFB



EDGMAX = 11894 EDGMIN = 27
EDGMEAN = 928 EDGVAR = 1035
EDGMEAN + EDGVAR = EDGTHR = 1963

FLIR Image #2

Eglin AFB

Appendix 2

THE LANGUAGE OF MASSIVELY PARALLEL IMAGE PROCESSING COMPUTERS

Gerhard X. Ritter

Department of Mathematics
University of Florida
Gainesville, Florida 32611

ABSTRACT

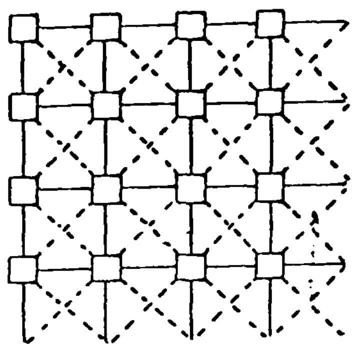
In this paper we define an image algebra whose operators serve as the basis of a new image processing language. In comparison to other existing image algebras, this algebra is capable of expressing most common image processing algorithms and transforms in terms of its operators. The development of this algebra has been influenced by the architectures of massively parallel image processing systems.

1. INTRODUCTION

About 25 years ago, Unger (16) proposed that algorithms for image processing and analysis could be implemented in parallel using "cellular array" computers. Recent advances in VLSI technology now permits the realization of such array computers. A detailed description concerning the diversity and genealogy of cellular array computers can be found in (10).
For our purposes it suffices to observe that cellular architectures implement variations of von Neumann's automaton (15).

NASA's massively parallel processor or MPP (2), and the CLIP series of computers developed by Duff (4) represent the classic embodiment of von Neumann's original automaton. The CLIP4 which constitutes the latest in the series of CLIP computers, consists of an array of 9216 (96x96) processors. Employing VSLI technology, sets of eight processing elements are integrated on a single chip.

The MPP also integrates eight processing elements per chip in an assemblage of 128x132 processing elements. In distinction to the CLIP, where each processing element has the capability of communicating with its eight immediate neighbors, an MPP processing element has connection to only four immediate neighbors as indicated by the solid lines in Figure 1.



Cellular image processing automaton of identical processors with nearest neighbor connection

Figure 1

Using these types of hardwired communication links between neighboring processors, each processor is responsible for one pixel (or one element of the image), and is capable of performing operations on the image via its communications links. These local operations can be expressed in terms of neighborhood operators or neighborhood funtions and are performed in parallel on the whole images and neighborhoods (i.e. sub-images induced by the local windows). In this sense then array processors impose a natural algebra on the set of images and window configurations.

Several image algebras employing these concepts already exist. Among these, there are only three that mathematicians would call "algebras" (7,13,14). However, despite their profound accomplishments, these algebras are not capable of expressing most common image processing operations such as Fourier transformations, gray scale averaging, and various edge detection techniques. In fact, the failure of these algebras to express a fairly straightforward U.S. Government furnished FLIR algorithm has been well documented (8).

In contrast, the image algebra developed by this author is capable of expressing most common image processing operations in terms of its operators. The development was motivated by the Air Force's need for translating image processing algorithms into a common mathematical language for performance characterization, documentation, and algorithm simplification.

In the next section we provide a rigorous mathematical definition of this algebra, endowing it with sufficient flexibilty for implementation on future reconfigurable neighborhood computers (3) as well as conventional serial image processing machines.

2. FUNDAMENTAL TERMS

Henceforth, Z and C shall denote the sets of integers and complex numbers, respectively. Although we could just as well have used the set of reals instead of complex numbers, we obtain a mathematically more useful and extensive structure by employing the latter.

DEFINITION 2.1.

(1) $S = \{(x,y,z): x,y \in Z,\ z \in C\}$.

(2) $P(S) = \{A: A \subset S\}$.

The power set P(S) will represent our universe of discourse. In particular, images will be viewed as elements of P(S).

Henceforth $s = (x,y,z)$ and $s' = (x',y',z')$ will denote elements of S and A, B subsets of S.

DEFINTION 2.2.

(1) s and s' are said to be <u>related</u>, denoted by $s \sim s'$, if $x = x'$ and $y = y'$, otherwise s is <u>not related</u> to s', which is denoted by $s \nsim s'$.

(2) $A_B = \{a \in A: a \nsim b \text{ for any } b \in B\}$ and $A^B = \{a \in A: a \sim b \text{ for some } b \in B\}$.

(3) A is <u>related</u> to B, denoted by $A \sim B$ if $A^B = A$ and $B^A = B$.

(4) A called an <u>image</u> if whenever $a,b \in A$ and $a \sim b$, then $a = b$.

If c is a complex number then the magnitude of c will be denoted by $|c|$ and the real part of c by $r(c)$. Given two complex numbers c and c', we define the <u>maximum</u> and <u>minimum</u> of c and c' as $c \nabla c' = \max\{r(c), r(c')\}$ and $c \Delta c' = \min\{r(c), r(c')\}$, respectively.

DEFINITION 2.3

(1) The <u>gray level sum</u> of s and s' is defined as $s(+)s' = (x,y,z+z')$ and the <u>gray level product</u> as $s(x)s = (x,y,zz')$.

(2) The <u>maximum</u> of s and s' is defined as $s(\nabla)s' = (x,y,z\nabla z')$ and the the <u>minimum</u> as $s(\Delta)s' = (x,y,z\Delta z')$.

(3) If f is a real or complex valued function on C, then the pixel function $\underline{f}$ induced by f is defined as $\underline{f}(s) = (x,y,f(z))$. In particular, we denote the <u>absolute value</u> or <u>magnitude</u> of s by $|s| = (x,y,|z|)$, and <u>exponentiation</u> and <u>scalar multiplication</u> by a complex number c by $s^c = (x,y,z^c)$ and $cs = (x,y,cz)$, respectively.

For finite subsets of S we also define the following four operations.

(4) $s(+)A = s(+)\ (\underset{a \in A}{+})a$

(5) $s(x)A = s(x)\ (\underset{a \in A}{x})a$

(6) $s(\nabla)A = s(\nabla)(\underset{a \in A}{\nabla})a$

(7) $s(\Delta)a = s(\Delta)(\underset{a \in A}{\Delta})a$

Here the operation $(\underset{a \in A}{+})a$ means to add using (+), all of $a \in A$.

Several comments are in order. First, observe that the operations defined in (1) and (2) are not commutative. To further clarify (3), consider examples such as $\underline{\sin}(s) = (x,y,\sin z)$ $\underline{\exp}(s) = (x,y,\exp(z))$ and $\underline{\mathrm{Ln}}(s) = (x,y,\ln z)$. Finally, note the order in which the a's are added in (4) is immaterial since s is added on the "left". Thus, if $A = \{a,b,c\}$, then $s(+)A = s(+)a(+)b(+)c = s(+)c(+)a(+)b$. The same observation holds for operations (5) through (7).

DEFINITION 2.4

A <u>neighborhood function</u> or <u>neighborhood</u> for S is a function $N:S \to P(S)$. The mathematical image N(s) of a point $s \in S$ is called the <u>N neighborhood</u> of s or, simply, a <u>neighborhood</u> of s. The restriction of N to a subset A of S will be denoted by N_A and is called a <u>neighborhood</u> for A. The <u>deleted neighborhood</u> N'(s) is defined as $N'(s) = N(s) - \{s\}$.

We are now in the position of defining a universal algebra which operates on subsets of S.

3. IMAGE ALGEBRA

As mentioned earlier, the operands of our

The algebra $\mathfrak{C}$, defined in the previous section, initially evolved from the four neighborhood operations (Definition 3.3) as a need for mathematically describing image processing routines that are "natural" to cellular architectures. Addition of the remaining four arithmetic-logic operators, however, yields a more flexible algebra. In fact, the extended algebra provides a uniform method for describing most standard image transforms and image processing techniques in terms of algebraic formulae involving only images and the operations defining the algebra. This is accomplished by analyzing the basic components and operations constituting a given manipulation or transform and then translating them into the language $\mathfrak{C}$. Due to limited space, we present translations of only a few well-known techniques, omitting more complicated algorithms, proofs and verifications, all of which can be found on (1ϕ).

4.1 EDGE DETECTION

The edge detection and enhancement techniques described in (8) are easily translated into the language of $\mathfrak{C}$. We provide two standard examples.

Defining the deleted neighborhood function B_i by $B_i'(a) = 5M'_{i,i+1,i+2}(a)(+) - M'_{i+3,\ldots,i+7}(a)$ permits us to express the Kirsch edge detection algorithm (6) as

$$K = IV\Big(\bigvee_{i=0}^{7}\Big)\Big|\,A(+)B_i' - A\,\Big|\Big),$$

where K denotes the enhanced image obtained from A. The logarithmic edge detection scheme as defined by Wallis (8,p.489) translates into:

$$W = (1/4)\mathrm{Ln}\Big\{A\times\Big(A^3(\times)\Big(\prod_{i=0}^{3}\Big)(M'_{2i})^{-1}\Big)\Big\},$$

where A and W denote the input and output images, respectively.

4.2 THRESHOLDING

Since $(rI^A) \wedge A = \{(a,b,rc): (a,b,c)\ A\}$, the image obtained from thresholding A at r given by $B = (rI^A) \wedge A$.

4.3 AVERAGING FILTER

Suppose $N(a)$ denotes the window with with center pixel a used for averaging A. Then since $A(+)N' = \{a(+)N'(a): a \in A\}$, the averaging filter can be translated as $B = n^{-1}(A(+)N')$, where n represents the number of pixels in the window N and B the output obtained from the image A.

4.4 GEOMETRIC FILTERS

As mentioned in the Introduction, a fairly successful Boolean image neighborhood algebra, based on two fundamental operators, was independently developed by P.E. Miller (7), J. Serra (13) and S. Sternberg (14). The two fundamental operators of this algebra correspond to the Minkowski addition and subtraction of sets in Euclidean space (12) and (5). In the image processing literature the Minkowski operations are commonly referred to as the <u>expansion</u> or <u>dilation</u> and the <u>erosion</u> or <u>shrinking</u> operators. It turns out that if N is any neighborhood configuration and A an image, then the dilation of A by N tranlates into $A(\nabla)N$ and the erosion of A by N into $A(\Delta)N$. For example, if $N(a) = M_{0246}(a)$ and A is as shown in Figure 2(a), then $A(\nabla)N$ and $A(\Delta)N$ are as shown in Figure 2(b) and (c). It follows that our algebra generalizes the Minkowski image algebra.

Two of the most basic and far reaching combinations of the Minkowski operations have become known as the <u>closing</u> and <u>opening</u> operations. A closing is an expansion followed by a shrinking while an opening is a shrinking followed by an expand. For A and N as in the last example, the closing of A by N correspnds to $(A(\nabla)N)(\Delta)N$ and is shown in Figure 2(d). Observe that the result of a closing is a smoother image, with the interior "holes" (zeros) removed. In many cases the closing filter exceeds the local averaging filter on performance (1).

```
00000000000000      00000010000000
00000010000000      00000111101100
00000111101100      00111111111110
00111111111000      01111111111100
00001100011000      00111111111100
00000111111100      00001111111110
00000000010000      00000111111100
00000000000000      00000000010000

     (a)                 (b)

00000000000000      00000000000000
00000000000000      00000010000000
00000010000000      00000111101100
00000100000000      00111111111000
00000000000000      00001111111000
00000000010000      00000111111100
00000000000000      00000000010000
00000000000000      00000000000000

     (c)                 (d)
```

Figure 2

4.5 THE FOURIER TRANSFORM

As a final example we examine the discrete Fourier transform (DFT) of a nxn image array. The usual definition of the DFT is given by

$$F(u,v) = (1/n)\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}\{f(x,y)\ \exp[(-i\pi/n)(u,y+vy)]\},$$

where $f(x,y)$ represents the gray level at position (x,y).

In order to express this transformation in terms of whole images, we need to define some special images and neighborhoods:

algebra wil be images and neighborhoods (subimages and windows). Before defining the operators of this algebra, we have need to define certain special images.

DEFINTION 3.1.

(1) The <u>zero image</u> is defined as $0 = \{(x,y,z) \in S: z = 0\}$.

(2) The <u>unit image</u> is defined as $I = \{(x,y,z) \in S: z = 1\}$.

(3) If f is a real or complex valued function on C, then the <u>induced image function</u> F on P(S) is defined as $F(A) = \{\underline{f}(a): a \in A\}$, for each subset A of S.

Here $\underline{f}$ denotes the pixel function induced by f. Thus $\overline{Exp}(A) = \{\underline{exp}(a): a \; A\}$ and $Ln(A) = \{ln(a): a \in A\}$. In particular, the <u>magnitude</u> of A is given by $|A| = \{|a|: a \in A\}$, $cA = \{ca: a \in A\}$ and $A^c = \{a^c: a \in A\}$, where $c \in C$. Thus, $-A = \{-a: a \in A\} = \{(x,y,-z): (x,y,z) \in A\}$ and $A^{-1} = \{a^{-1}: a \in A\} = \{(x,y,1/z): (x,y,z) \in A\}$. Note that the latter is defined only if $A \cap 0 = \phi$.

We are now ready to define the first four binary operators of our algebra.

DEFINITION 3.2

Let A and B be images and $E = A_B \cup B_A$.

(1) The <u>grav level sum</u> of A and B is defined as $A + B = \{a(+)b: a \in A, b \in B, \text{ and } a \sim b\} \cup E$.

(2) The <u>grav level product</u> of A and B is defined as $AxB = \{a(x)b: a \in A, b \in B, \text{ and } a \sim b\} \cup E$.

(3) The <u>maximum</u> of A and B is defined as $A \nabla B = \{a(\nabla)b: a \in A, b \in B \text{ and } a \sim b\} \cup E$.

(4) The <u>minimum</u> of A and B is is defined as $A \Delta B = \{a(\Delta)b: a \in A, b \in B \text{ and } a \sim b\} \cup E$.

In contrast to the operations defined in Section 2, all the above operations are commutative and associative. In fact, $A+0^A = A$, $AxI^A = A$, $A+(-A) = 0^A$, $AxA^{-1} = I^A$ and $A^n = AxAx...xA$, where n is a positive integer and the product consists of n-factors. Further properties and theorems concerning these operations can be found in (10).

The next set of operations we are about to define is a special set of <u>neighborhood operations</u>. However, we point out that the neighborhood N could be replaced by any subset B of S.

DEFINITION 3.3

Let N be a finite neighborhood of A. The <u>neighborhood sum, maximum</u> and <u>minimum</u> of A and N are respectively defined as:

(1) $A(+)N = \{a(+)N(a): a \in A\}$

(2) $A(x)N = \{a(x)N(a): a \in A\}$

(3) $A(\nabla)N = \{a(\nabla)N(a): a \in A\}$

(4) $A(\Delta)N = \{a(\Delta)N(a): a \in A\}$.

The universal image algebra is now defined as the pair $\Omega = (J,T)$ where $T = \{+,x,\nabla,\Delta,(+),(x),(\nabla),(\Delta)\}$ and J denotes the set of images and neighborhoods. Various properties of this algebra have been explored in (10) and (11). The next section provides but a small glance at the potential of this algebra. The examples we give should also provide sufficient insight into the natural interaction of this algebra with the architecture and operations of cellular array computers. This is particulary evident when considering the last four neigborhood operations.

4. APPLICATIONS

Eight-neighbor logic operations are some of the most common operations used in image processing. These operations lend themselves particulary well to the type of array processor architecture portrayed in Figure 1. The MPP accomplishes eight-neighbor operations by shifting over the entire array. We let M(a) denote the 3 x 3 neighborhood configuration corresponding to this wiring, where "a" denotes the center pixel. The links of a processor to the immediate neighborhoods can usually be controlled by on-off switches, allowing the configurations of different subneighborhoods. It will be convenient to label the corresponding subsets of M(a) by the counterclockwise numbering convention, illustrated by the following figures:

$$M(a) = \begin{matrix} 3 & 2 & 1 \\ 4 & a & 0 \\ 5 & 6 & 7 \end{matrix} \qquad M_{017}(a) = \begin{matrix} & 1 \\ a & 0 \\ & 7 \end{matrix}$$

$$M_{0246}(a) = \begin{matrix} & 2 & \\ 4 & a & 0 \\ & 6 & \end{matrix}$$

In particular, $M = M_{012...7}$ corresponds to CLIP's neighborhood circuitry, while M_{0246} to the neighborhood arrangement of the MPP.

(i) $X = \{(x,y,z) \in S: z = x\}$ and

$Y = \{(x,y,z) \in S; z = y\}$

(ii) $E(u,v) = \mathrm{Exp}\{(-2\pi i/n)(uX+vY)\}$,

where, $u,v \in Z$.

The <u>Fourier neighborhood function</u> of an $n \times m$ image is defined as the function $F_A: 0^A \to P(S)$, where $F_A(u,v,0) = A \times E(y,v)$.

The Fourier transformed image, $F(A)$, of A can then be expressed by the simple formula

$\underline{F}(A) = 0^A (+) F_A$.

5. ACKNOWLEDGEMENT

The support of the U.S. Air Force Office of Scientific Research under Contract F83-00-65 is greatly acknowledged.

REFERENCES

1. L.A. Ankeney and G.X. Ritter, "Applications of Cellular Topology in Image Processing," <u>Int J. of Computer and Inf. Science</u>, 12(6) 1983, 433-456.

2. K.E. Batcher, "Design of a Massively Parallel Processor," <u>IEEE Trans. Computers</u>, 29(9), 1980, 836-840.

3. S. Chen and G.X. Ritter, "Image Processing Architectures and Languages," <u>Proc. IEEE Int'l Conf. on Computer Design: VLSI in Comp.</u> New York, 1983, 723-726.

4. T.J. Fountain "CLIP4: A Progress Report," <u>Languages and Architectures for Image Processing</u>, Academic Press, London, 1981.

5. H. Hadwiger, <u>Vorlesungen ueber Inhalt, Oberflaeche und Isoperimetrie</u>, Springer Verlag, Berlin, 1957.

6. R. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," <u>Computer and Biomedical Reasearch</u>, 4(3), 1971, 315-328.

7. P.E. Miller, <u>An Investigation of Boolean Transformations</u>, Ph.D Thesis, Ohio State University, Columbus, OH. 1978.

8. P.E. Miller, "Development of a Mathematical Structure for Image Processing," Perkin-Elmer Optical Div., TR, Danbury, CT, 1983.

9. W.K. Pratt, <u>Digitial Image Processing</u>, Wiley and Sons, New York, 1978.

10. K. Preston, "Cellular Logic Computers for Pattern Recognition," <u>Computer</u>, 16(1), 1983, 36-47.

11. G.X. Ritter, "On the Foundation of a Common Image Processing Language Algebra," TR. USAF Armament Div., EO Terminal Guidance Branch, Eglin ARB, FL., 1983.

12. G.X. Ritter, "An Image Processing Algebra," Preprint.

13. J. Serra, <u>Image Analysis and Mathematical Morphology</u>, Academic Press, London, 1982.

14. S. Sternberg, "Languages and Architechtures for Parallel Image Processing," <u>Proc. Conf. on Pattern Recogition in Pratice</u>, North-Holland Publ., 1980.

15. J. von Neumann, "The General Logical Theory of Automata," <u>Cerebral Mechanism in Behavior: The Hixon Symposium</u>, Wiley and Sons, New York, 1951.

16. S.H. Unger "A Computer Oriented Toward Spatial Problems," <u>Proc. IRE</u>, 46,1958, 1744-1750.