Title: The Development of a Mathematical Foundation for Cellular Image Processing

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.
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

Approved for public release; distribution unlimited.
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 I 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


FLIR Image #2, University of Florida
Note the targets labeled (1) and (2)
and compare with 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 original 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 CLIPs which constitutes the latest in the series of CLIP computers, consists of an array of 9216 (96x96) processors. Employing VLSI technology, sets of eight processing element are integrated on a single chip.

The MPP also integrates eight processing elements per chip in a 128x132 array of 15360 (96x160) processing elements. In distinction to the CLIP, where each processing element has the capability of communicating with all eight immediate neighbors, an MPP processor, at least has connection to only four immediate neighbors as indicated by the solid lines in 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 functions 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 them are only three that mathematicians would call "algebras" (7,13,16). 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, allowing it with sufficient flexibility 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 \mathbb{R}, z \in \mathbb{C}\}.

(2) $P(S) = A \cup C S$.

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 $S$.

DEFINITION 2.2.

(1) $s$ and $s'$ are said to be related, denoted by $s - s'$, if $x - x'$ and $y - y'$, otherwise $s$ is not related to $s'$, which is denoted by $s \neq s'$.

(2) $A_B = \{a \in A : a + b \text{ for any } b \in B\}$

and $A_B = \{a \in A : a - b \text{ for some } b \in B\}$.

(3) $A$ is related to $B$, denoted by $A - B$ if $A_B = A$ and $B_A = B$.

(4) A called ann image if whenever $a, b \in A$

and $a - b$, 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 $c$. Given two complex numbers $c$ and $c'$, we define the maximum and minimum of $c$ and $c'$ as $c = \max\{c, c'\}$ and $c = \min\{c, c'\}$, respectively.

DEFINITION 2.3

(1) The gray level sum of $s$ and $s'$ is defined as $s + s' = (x, y, z + z')$ and the gray level product as $s \cdot s' = (x, y, z'z)$.

(2) The maximum of $s$ and $s'$ is defined as $\max(s, s') = (x, y, z)$ and the minimum as $\min(s, s') = (x, y, z')$.

(3) If $f$ is a real or complex valued function on $S$, then the pixel function induced by $f$ is defined as $(f) = (x, y, f(x))$. In particular, we denote the absolute value or magnitude of $s$ by $|s| = (x, y, x)$, and exponentiation and scalar multiplication by a complex number $c$ by $s^c = (x, y, x^c)$ and $cs = (x, y, cx)$, respectively.

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

(4) $a + A = a + (\cdot a)_{a \in A}$

(5) $a \cdot A = a \cdot (\cdot a)_{a \in A}$

(6) $a(f)A = a(f) (\cdot a)_{a \in A}$

(7) $a(\cdot a) = a(\cdot a) (\cdot a)_{a \in A}$

Here the operation $(\cdot a)$ means to add using $(\cdot a)$, 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 $\sin(e) = (x, y, \sin x)$, $\exp(a) = (x, y, \exp(x))$ and $\ln(a) = (x, y, \ln x)$.

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 $\{a\} + s = a + b + c$ and $\{a\} + b + c$ $= s(a \cdot a) + (\cdot a)_{a \in A}$. The same observation holds for operations (5) through (7).

DEFINITION 2.4

A neighborhood function or neighborhood for $S$ is a function $h : S \rightarrow P(S)$. The mathematical image $h(s)$ of a point $s \in S$ is called the neighborhood of $s$, or, simply, a neighborhood of $s$. The restriction of $h$ to a subset $A$ of $S$ will be denoted by $h_A$, and is called a neighborhood for $A$. The deleted neighborhood $h^*(s)$ is defined as $h^*(s) = h(s) - \{s\}$.

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

3. IMAGE ALGEBRA

As mentioned earlier, the operators of our
The algebra 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 operations, 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 of algebra. This permits us to express the Kirsch edge detection algorithm (1) as

\[
\hat{K} = \text{IV}(W) \big[ A(+)B - A \big],
\]

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

\[
W = (1/3) \ln \left\{ \sum_{x} \sum_{y} (A(x,y))^{2} \right\},
\]

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

4.2 THRESHOLDING

Since \((f^{(m)})A = f(a,b,c) = (a,b,c) A\), the image obtained from thresholding \( A \) at \( r \) given by \( B = (f^{(r)})A \).

4.3 AVERAGING FILTER

Suppose \( N(a) \) denotes the window 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. Morill (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 opening or closing and the erosion or shrinking operators. It turns out that if \( N \) is any neighborhood configuration and \( A \), an image, then the dilation of \( A \) by \( N \) translates into \( A(+)N \) and the erosion of \( A \) by \( N \) into \( A(-)N \). For example, if \( N(a) = K_{244}(a) \) and \( A \) is as shown in Figure 2(a), then \( A(+)N \) and \( A(-)N \) 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 closing and opening 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 \( N \) corresponds to \((A(+)N)(-)N \) 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).

4.5 THE FOURIER TRANSFORM

As a final example we examine the discrete Fourier transform (DFT) of a 7x7 image array. The usual definition of the DFT is given by

\[
F(u,v) = (1/n) \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x,y) e^{-2\pi j (u/n)x + (v/n)y},
\]

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 will be images and neighborhoods (sub-images and windows). Before defining the operators of this algebra, we have to define certain special images.

**DEFINITION 3.1.**

1. The zero image is defined as \( O = \{(x,y,z) \in S : z = 0 \} \).
2. The unit image 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 induced image function \( F \) on \( P(S) \) is defined as \( F(A) = \{(a) : a \in A \} \), for each subset \( A \) of \( S \).

Here \( f \) denotes the pixel function induced by \( f \). Thus \( \mathcal{E}(a) = \{ \mathcal{E}(a) : a \in A \} \) and \( \mathcal{L}(a) \) is a finite set of \( a \) given by \( \mathcal{A} = \{(a) \in A : a \in A \} \), \( \mathcal{C} = \{a \in A : a \in A \} \), and \( \mathcal{D} = \{a \in A : a \in A \} \). We are now ready to define the first four binary operators of our algebra.

**DEFINITION 2.2.**

Let \( A \) and \( B \) be images and \( E = A \cup B \).

1. The gray level sum of \( A \) and \( B \) is defined as \( A + B = \{(a) : a \in A, b \in B, a + b = 1 \} \).
2. The gray level product of \( A \) and \( B \) is defined as \( A \cdot B = \{(a) : a \in A, b \in B, a = b \} \).
3. The maximum of \( A \) and \( B \) is defined as \( A \lor B = \{(a) : a \in A, b \in B, a = b \} \).
4. The minimum of \( A \) and \( B \) is defined as \( A \land B = \{(a) : a \in A, b \in B, a = b \} \).

In contrast to the operations defined in Section 2, all the above operations are commutative and associative. In fact, \( A \cdot B = A, A \lor B = A, A \land B = A \).

**DEFINITION 3.3.**

Let \( N \) be a finite neighborhood of \( A \). The neighborhood sum, maximum and minimum of \( A \) and \( N \) are respectively defined as:

1. \( A \lor N = \{(a) : a \in A \cup N \} \)
2. \( A \land N = \{(a) : a \in A \cap N \} \)
3. \( A \land N = \{(a) : a \in A \cap N \} \)
4. \( A \land N = \{(a) : a \in A \cap N \} \).

The universal image algebra is now defined as the pair \( G = (J, T) \) where \( T = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \} \) 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 particularly evident when considering the last four neighborhood operations.

**4. APPLICATIONS**

Eight-neighbor logic operations are some of the most common operations used in image processing. These operations lend themselves particularly well to the type of array processor architecture portrayed in Figure 1. The HPP accomplishes eight-neighor operations by shifting over the entire array. We let \( H(a) \) 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 sub-neighborhoods. It will be convenient to label the corresponding subsets of \( H(a) \) by the counterclockwise numbering convention, illustrated by the following figures:

- \( H = \{(a) : a \in A \} \)
- \( H(a) = \{(a) : a \in A \} \)
- \( H(a) = \{(a) : a \in A \} \)
- \( H(a) = \{(a) : a \in A \} \)

In particular, \( H = H' \) corresponds to CLIP's neighborhood circuitry, while \( H = H' \) to the neighborhood arrangement of the HPP.
(ii) X = {(x,y,z) ∈ S | z = x} and
Y = {(x,y,z) ∈ S | z = y}

(iii) E(u,v) = \exp\left(-2\pi i (uXvY)\right),
where u,v ∈ K.

The Fourier neighborhood function of an
image is defined as the function F(\Lambda)
+ P(\Lambda), where F(u,v) = \Lambda(\Lambda x,y,v).

The Fourier transformed image, F(\Lambda), of \Lambda can
then be expressed by the simple formula
F(\Lambda) = \Theta^\Lambda(\Theta)\Lambda.

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 C.X. Ritter, "Applications of
Cellular Topology to Image Processing," Int. J.

2. E.E. Batcher, "Design of a Massively Parallel
Processor," IEEE Trans. Computers, 29(9),
1980, 830-840.

3. S. Chen and C.X. Ritter, "Image Processing
Architectures and Languages," Proc. IEEE Int'l
Conf. on Computer Design: VLSI in Comp. New
York, 1983, 723-728.

4. T.J. Fountain "CLIP4: A Progress Report,"
Languages and Architectures for Image

5. H. Hadwiger, Vorlesungen uber Inhalte,
Oberflaechen und Isoperimetrie, Springer
Verlag, Berlin, 1957.

6. J. Kirsch, "Computer Determination of the
Constituent Structure of Biological Images,"
Computer and Biomedical Research, 4(3), 1971,
373-376.

7. P.E. Miller, An Investigation of Boolean
Transformations, 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. V.K. Pratt, Digital Image Processing, Wiley

10. K. Preston, "Cellular Logic Computers for
Pattern Recognition," Computer, 16(1), 1983,
30-47.

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

Preprint.

13. J. Serra, Image Analysis and Mathematical

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

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

16. S.H. Unger "A Computer Oriented Toward Spatial