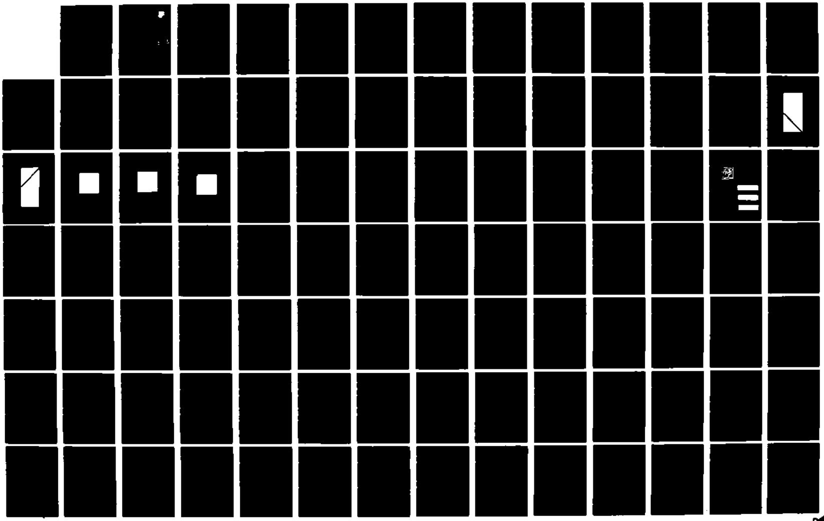
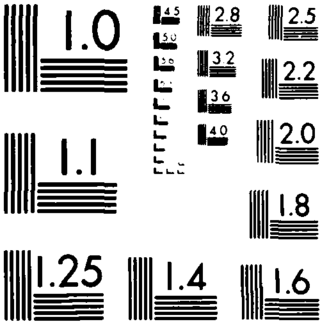
AD-A149 550 OPTICAL EIGENVECTOR(U) AERODYNE RESEARCH INC BILLERICA 1/2
MA J CAULFIELD OCT 84 ARI-RR-393 RADC-TR-84-208
F19628-82-C-0068

UNCLASSIFIED F/G 20/6 NL





MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963-A

12

RADC-TR-84-208
Final Technical Report
October 1984

OPTICAL EIGENVECTOR

Aerodyne Research, Inc.

J. Caulfield

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

This effort was funded totally by the Laboratory Directors' Fund

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

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.

RADC-TR-84-208 has been reviewed and is approved for publication.

APPROVED:

JOSEPH L. HORNER
Project Engineer

APPROVED: Harold Roth

HAROLD ROTH, Director
Solid State Sciences Division

FOR THE COMMANDER: John A. Ritz

JOHN A. RITZ
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 (ESO), Hanscom AFB MA 01731. 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.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
N/A

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
ARI-RR-393

5. MONITORING ORGANIZATION REPORT NUMBER(S)
RADC-TR-84-208

6a. NAME OF PERFORMING ORGANIZATION
Aerodyne Research, Inc.

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Rome Air Development Center (ESO)

6c. ADDRESS *(City, State and ZIP Code)*
45 Manning Road
The Research Center at Manning Park
Billerica MA 01821

7b. ADDRESS *(City, State and ZIP Code)*
Hanscom AFB MA 01731

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Rome Air Development Center

8b. OFFICE SYMBOL *(If applicable)*
ESO

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F19628-82-C-0068

8c. ADDRESS *(City, State and ZIP Code)*
Hanscom AFB MA 01731

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO
61101F	LDFP	04	C2

11. TITLE *(Include Security Classification)*
OPTICAL EIGENVECTOR

12. PERSONAL AUTHOR(S)
J. Caulfield

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM Mar82 TO Mar84

14. DATE OF REPORT *(Yr., Mo., Day)*
October 1984

15. PAGE COUNT
144

16. SUPPLEMENTARY NOTATION
This effort was funded totally by the Laboratory Directors' Fund

17. COSATI CODES

FIELD	GROUP	SUB. GR.
20	06	13

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Matrix
Optical computing
Eigenvalues
Signal processing

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

At the beginning of this contract, both we and the rest of the optical community imagined that simple analog optical computers could produce satisfactory solutions to eigenproblems. Early in this contract we improved optical computing conceptually and tested it experimentally. This demonstrated that high accuracy required digital optics. This led us to explore digital optical systolic array processors. Here we made sufficient progress to guarantee that the original contract goal (the use of optics for fast, accurate eigen solution) is now perfectly practical and to show that the hoped-for advantages in size, cost, and power consumption relative to equally fast electronic computers should be obtained.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Joseph L. Horner

22b. TELEPHONE NUMBER *(Include Area Code)*
617-861-5563

22c. OFFICE SYMBOL
RADC (ESO)

EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ABSTRACT

At the beginning of this contract both we and the rest of the optical community imagined that simple analog optical computers could produce satisfactory solutions to eigenproblems. Early in this contract we improved optical computing conceptually and tested it experimentally. This demonstrated that high accuracy required digital optics. This led us to explore digital optical systolic array processors. Here we made sufficient progress to guarantee that the original contract goal (the use of optics for fast, accurate eigen solution) is now perfectly practical and to show that the hoped-for advantages in size, cost, and power consumption relative to equally fast electronic computers should be obtained.

1. CONTRACT BACKGROUND

1.1 Background on Eigenproblems

The simplest eigenproblem can be stated in the form

$$A \vec{e}_i = \lambda_i \vec{e}_i \tag{1-1}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{11n} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ a_{n1} & a_{n2} & \cdots & a_{mn} \end{bmatrix} , \tag{1-2}$$

$$\vec{e}_i = \begin{bmatrix} (e_i)_1 \\ (e_i)_2 \\ \cdot \\ \cdot \\ \cdot \\ (e_i)_n \end{bmatrix} , \tag{1-3}$$

and λ_i is a scaler. The vector $\vec{e}_i$ is called an eigenvector and the scaler λ_i is called an eigenvalue. We will deal almost exclusively with the $m = n$ case.

We must make two observations about eigenproblems. First, they are a special case of a more general and powerful matrix analysis technique called singular value decomposition or SVD. We will deal with SVD as well in this report. Second, eigenproblems are of vital interest to the Air Force for several reasons. We note, as an example, how to use eigen solutions in antenna array processing.

In the selected example the antenna element weights (i.e., amplitude and/or phase adjustments) are to be found that steer a static multi-element antenna so as to maintain maximum signal-to-noise ratio (S/N) reception. As shown in Figure 1.1, each of m antenna elements receives (at a given time) a signal and a noise contribution, and these generally complex contributions form the m dimensional signal and noise vectors $\vec{s}$ and $\vec{n}$, respectively. Each antenna element is connected through a generally complex weight, and these weights form the vector $\vec{w}$. Note from Figure 1.1 that the weighted contributions from all antenna elements are summed to form the output (complex) signal whose S/N is to be maximized. This S/N may be expressed as shown in terms of the time-averaged squared moduli of the signal and noise parts of the output signal, and the resulting expression may then be reduced to the indicated eigenvalue equation. The matrix (in brackets) in this equation may be expressed as shown in terms of the time averaged outer products R_{nn} and R_{ss} of the noise and signal vectors respectively; these vectors are known from measurements on the unweighted output of each element. Thus the eivenvalue equation may be solved, and the eigenvector associated with the dominant (or maximum) eigenvalue will give the weights that maximize the S/N.

A computer simulation of the method that would be used by an optical systolic matrix vector system to solve the eigenproblem described above was carried out. This simulation required the specification of certain signal and noise statistics in accordance with practical expectations. Discussions with RADC experts familiar with such expectations led to the selection of a m = 128 antenna element problem with an average S/N on the order of unity at each element. A bimodal signal distribution across the antenna elements was selected so that $R_{ss} = \vec{s}'\vec{s}'^T + \vec{s}''\vec{s}''^T$, where the lognormal distributions $\vec{s}_k \sim (1/k) \exp\left[-(\ln k - \mu)^2/(2\sigma^2)\right]$ and $\vec{s}'_k = \vec{s}''_{129-k}$ were used with $64 \exp(\mu + \sigma^2/2)$, $64^2 = (\exp \sigma^2 - 1) \cdot \exp(2\mu + \sigma^2)$, $\vec{s}'_k = 1$, and $k = 1, 2, \ldots, 128$.

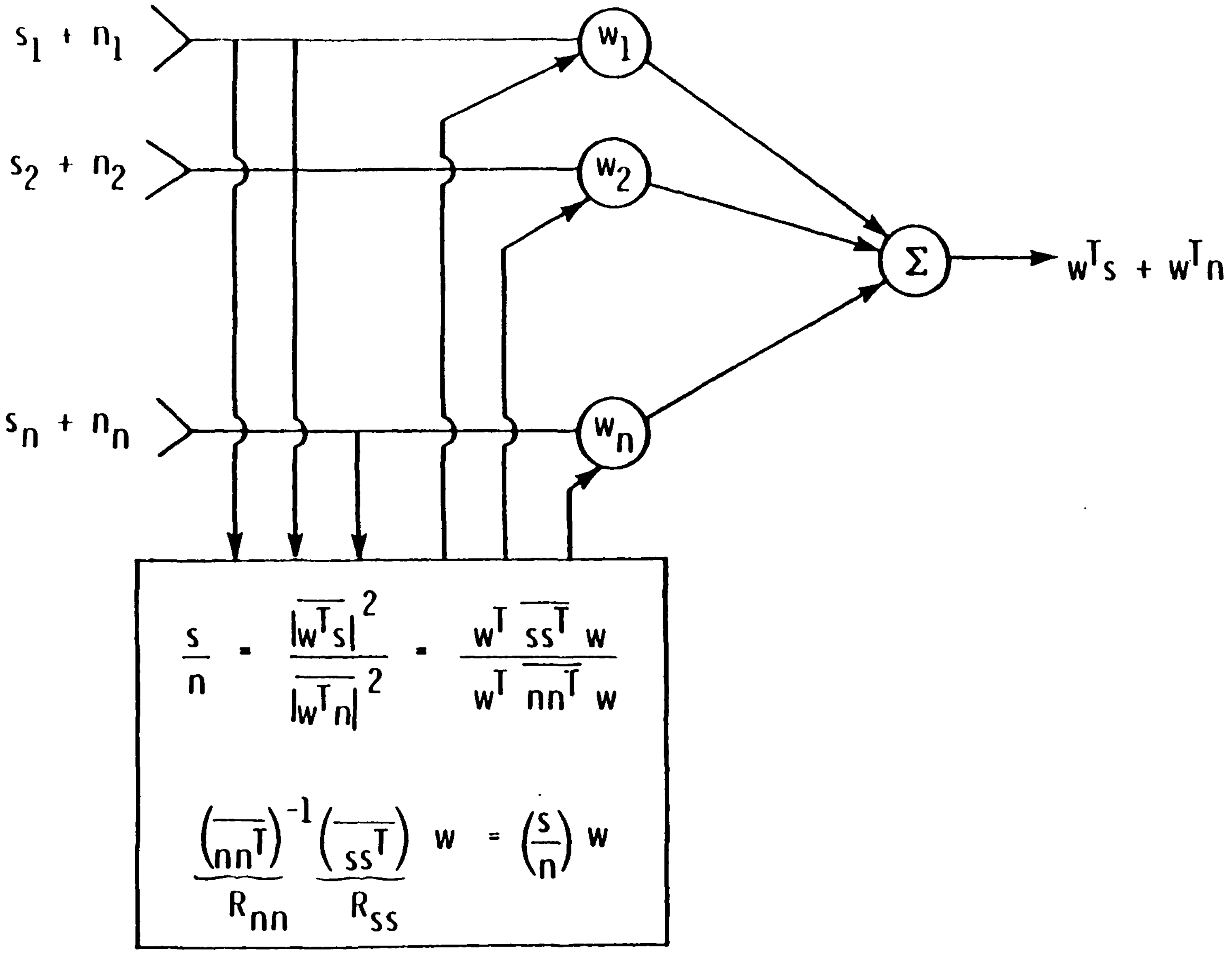

$s_1 + n_1$
$s_2 + n_2$
$s_n + n_n$
w_1
w_2
w_n
Σ
$w^T s + w^T n$
$\frac{s}{n} = \frac{\overline{|w^T s|}^2}{\overline{|w^T n|}^2} = \frac{w^T \, \overline{ss^T} \, w}{w^T \, \overline{nn^T} \, w}$
$\underbrace{\left(\overline{nn^T}\right)^{-1}}_{R_{nn}} \underbrace{\left(\overline{ss^T}\right)}_{R_{ss}} w = \left(\frac{s}{n}\right) w$

Figure 1.1. Schematic diagram of adaptive array antenna steering for maximum signal-to-noise ratio reception.

The matrix R_{nn} was selected to correspond to uncorrelated Gaussian noise so that $R_n = \vec{n}\vec{n}^T + \sigma_s^2 I$, where $\vec{n}_k = \vec{n} = 1$, $\sigma_s^2 = E(\vec{s}_k' - \vec{s}_k')$, and I is the identity matrix. Explicit matrix inversion was avoided in forming the eigenvalue equation matrix $M = R_{nn}^{-1} R_{ss}$ by using the expression[14]

$$R_{nn}^{-1} = D^{-1} - D^{-1} \vec{n}\vec{n}^T D^{-1}/(1 + \vec{n}^T D^{-1} \vec{n}) \qquad , \tag{1-4}$$

where $D = \sigma_s^2 I$. Note that M is a real, symmetric, positive definite matrix which will therefore have a set of real, positive eigenvalues. In general M and the weight eigenvector will be complex, but this case may be divided into separate real-part and imaginary-part eigenproblems of the form described above.

The computer simulation used the same power method that would be used by a typical optical systolic matrix vector system to obtain the eigenvector solution. The power method iterates matrix-vector multiplication operations, and the simulation determined that N = 35 such iterations were required to obtain the dominant eigenvalue and associated eigenvector of the 128 x 128 matrix M to a precision of 10^{-4}. The total time required to perform each matrix-vector multiplication iteration is, according to Table 1-1 and the discussion approximately $T_m = 23$ μs. At a 100 MHz clock rate the initial matrix input time is the time required to read in the 128(128 + 1)/2 symmetric matrix elements or approximately $T_\ell = 83$ μs, the final weight eigenvector output time for 128 vector elements is $T_r = 1.3$ μs, and the test for convergence time is $T_c =$ 0.01 μs. Thus the total OSAP system eigenvector solution time is approximately $T = T_\ell + N(T_m + T_c) + T_r = 0.89$ ms. Note that the basic block-floating-point computation rate is approximately $2n_2/T_m = 2\ (128)^2/(24$ μs) or about 1.4 GigaFLOPS (1.4 x 10^9 Floating Point Operation per Second).

The same eigenvector solution could be obtained by a state-of-the-art 5 MegaFLOPS all electronic board level array processor. In this case the matrix input and eigenvector output times would remain approximately the same, but the matrix vector multiplication for each iteration would require, in general, $2n^2 = 2(128)^2$ multiply and add operations. Since each operation would require 0.2 μs at the 5 MegaFLOPS rate, the total matrix vector multiplication time would be approximately $T_m = 6.6$ ms. Thus the total non-Optical Systolic Array Processor (OSAP) system eigenvector solution time would be approximately T = 35 (6.6 ms) = 230 ms, which is about 250 times longer than the Optical Systolic Array Processor (OSAP) system solution time estimated above. This comparison, which is displayed in Table 1-1, does not take into account the time required to calculate the matrix M given the vectors s and n. Assuming all electronic array processing at a 5 MegaFLOPS rate, this time would be on the order of 13 ms since the equivalent of roughly $4(128)_2$ multiply and add operations are involved in the calculation (which, as mentioned above, does not involve explicit matrix inversion). Thus even if the M matrix calculation time is included in the comparison, the OSAP system solution time is still much less than the non-Optical Systolic Array Processor (OSAP) system solution time. A separate Optical Systolic Array Processor (OSAP) system calculation of the M matrix could also be carried out, in

Table 1-1 - Optical Systolic Array Processor System Application Example Performance and Comparison

	OSAP System	All Electronic Board Level Array Processor
Symmetric matrix read-in time	83 μs	83 μs
General matrix-vector multiplication time	23 μs	6.6 ms
Dominant eigenvector readout time	1.3 μs	1.3 μs
Total eigenvector Solution time*	0.89 ms	230 ms
Typical arithmetic operation rate	1.4 GigaFLOPS	5.0 MegaFLOPS

*Includes read-in and read-out times and the time to execute the 35 iterations required to obtain the 129 eigenvector elements to a precision of 10^{-4}.

which case the Optical Systolic Array Processor (OSAP) system eigenvector solution time would be at least two orders of magnitude less than the non-Optical Systolic Array Processor (OSAP) system solution time. Hundreds of all electronic board-level array processors working in parallel might match the Optical Systolic Array Processor (OSAP) system computation speed, but only at considerable expense in size, power consumption, reliability, etc.

The specific adaptive antenna array processing example considered above clearly shows the potential of the Optical Systolic Array Processor (OSAP) system. In some applications (e.g., future millimeter wave adaptive arrays on tactical aircraft) an antenna array steering time for S/N maximization of less than 10 milliseconds may be required for arrays of more than 100 elements. The Optical Systolic Array Processor (OSAP) system would be of unique value as an enabling technology in such cases, and there is little doubt that an operational Optical Systolic Array Processor (OSAP) system would have a similar enabling role in a broad range of other applications.

1.2 Precontract Background Eigenproblem Algorithm

A group of optics workers from Aerodyne, Stanford, and Georgia Tech. published the first paper on optical solutions to eigenproblems (Appendix A). This paper led to this contract as well as to much research elsewhere on the same and related subject. The basic idea is extremely simple. We start with any vector $\vec{X}_o$. We can show that the set of n eigenvectors $\{\vec{e}_i\}$ forms a complete set, so we can write

$$\vec{X}_o = \vec{a}_1 e_1 + a_2 \vec{e}_2 + \ldots + a_n \vec{e}_n \quad . \tag{1-5}$$

Calling

$$\vec{X}_{m+1} = A \vec{X}_m \; (m = o, 1, 2, \ldots) \quad , \tag{1-6}$$

we have

$$\vec{X}_m = a_1 \lambda_1^m \vec{e}_1 + a_2 \lambda_2^m \vec{e}_2 + \dots + a_n \lambda_n^m \vec{e}_n \quad . \tag{1-7}$$

Clearly (except for the case of degenerate eigenvalues dealt with in Appendix A and elsewhere in this report) for large enough m we can approximate

$$\vec{X}_m \simeq a_k \lambda_k^m \vec{e}_k \quad , \tag{1-8}$$

where

$$(\lambda_k) > (\lambda_l) \tag{1-9}$$

for all $l \neq k$.

That is, by raising all of the eigenvalues to successively higher powers we reach a point at which one eigenvalue dominates. Hence this is called the power method.

Usually we set

$$\vec{e}_k^{\,T} \vec{e}_k = 1 \quad , \tag{1-10}$$

where the superscript T indicates transposition. Having $\vec{e}_k$, we find λ_k using Eq. (1-1).

1.3 Precontract Status of the Optical Matrix Processor

The optical processor we conceived of using was the Stanford processor (Appendix B). The reason was very simple: there was no alternative. The

Stanford processor was the beginning and the prototype, but it is clear in retrospect that its two major drawbacks were

1. Totally analog operation and hence very limited accuracy and

2. The necessity of using a two-dimensional spatial light modulator to allow changing of the matrix.

1.4 Precontract Goals and Approaches

The explicit overall goal of this contract was to use optical methods to solve eigenproblems rapidly and with "sufficient accuracy." Naturally we held closely to that goal.

The precontract approach was to implement the algorithm of Subsection 1.2 in the processor of Subsection 1.3. As we began to do this, we found that both approaches essentially guaranteed failure to meet the overall goal. Accordingly we set out to improve both the algorithm and the hardware. Both were accomplished, and we can now show that extremely useful optical eigenproblem solvers can be constructed showing advantages in

- Size,
- Weight, and
- Power consumption

over electronic processors having the same extremely high speed or, conversely, advantages in speed over electronic computers of the same size, weight, and power consumption.

2. REPORT APPROACH AND RATIONALE

A great deal of productive work was done under this contract. Therefore, telling the whole story as a continuous narrative runs the risk of hiding the coherence of the effort. Accordingly, we have chosen to relegate to appendices detailed discussions which were either published or prepared for publication under this contract. The text, therefore, serves as a comprehensible guide to and through these various individual efforts and concludes with an attempt to tie summarize of these efforts as they relate to the contract goal is ennunciated in Subsection 1.4.

3. PROBLEMS WITH THE PRECONTRACT ALGORITHM AND PROCESSOR APPROACHES

3.1 Algorithm Approach

Early in the contract we noted several major problems with the power method as described in Subsection 1.2. We summarize these briefly here. First, convergence might be very slow. Suppose the two highest eigenvalues are $\lambda(1+e)$ and λ, where $0 < e << 1$. Clearly convergence is not achieved until the iteration m in which

$$[\lambda(1+e)]^m >> [\lambda]^m \tag{3-1}$$

or

$$(1+e)^m >> 1 \quad . \tag{3-2}$$

We have

$$(1+e)^m \simeq 1+me \quad , \tag{3-3}$$

or

$$m >> 1/e \quad . \tag{3-4}$$

There is reason to believe that we may not need $m = 10^6$ or more. Even the speed of optics might not overcome this disadvantage so well as to make it superior to electronics. Second, the original approach (Appendix A) did not include a truly satisfactory way of finding eigenvectors beyond the first one. The general problem is called deflation - removing all previously-calculated eigenvector information from the problem.

3.2 Optical Processor Approach

The two drawbacks of the original Stanford processor for the goals of this contract were noted in Subsection 1.3. Here, we want to discuss in more detail why analog processing must be abandoned. For any processor we can argue that the solution obtained, while an inaccurate answer to the problem posed, is a perfectly accurate solution to another problem (an inaccurately-posed problem). Following this line of reasoning, mathematicians have been able to cast most linear algebra accuracy problems in the following manner. The average error in the answer, e(x), is related to the average error in the calculation itself, e(c), by

$$e(x) \simeq \text{cond}\ (A)\ e(c) \quad , \tag{3-5}$$

where

$$\text{cond}\ (A) = \text{"condition number"}$$

of the matrix. The condition number is the ratio of the largest to the smallest eigenvalue. In the case of antenna arrays with jammers in the field, the condition number of the matrix of interest can easily be 10^6. On the other hand the calculation error of a super analog optical processor might be $e(c) \approx 10^{-2}$. This suggests that the results of optical eigenproblem solvers might be essentially meaningless. This is why analog electronic computers have been largely abandoned in favor of digital electronic computers. This is also why we too soon abandoned analog computers in favor of digital ones. Optical digital computers will be slower and more expensive than optical analog computers, but we have no alternative when solving eigenproblems optically.

4. A DIGRESSION ON ANALOG OPTICAL COMPUTERS

While considering and rediscovering these concerns with optical analog computers, we developed a totally new way to use analog matrix processors. This new approach offers significant speed and convergence advantages over methods borrowed directly from the digital computer literature. We do not belabor these advantages here, because we have now abandoned analog methods for this problem. Our work in this area is shown in Appendices C and D.

5. ALGORITHM IMPROVEMENTS

Having noted the problems with the precontract algorithm in Subsection 3.1, we now describe our successful efforts to solve those problems. The convergence was accelerated greatly by going to another type power method. We explain it crudly here and in much more detail in Appendix E. The explanation here will be in terms of matrix-matrix multipliers but we show in Appendix E that much the same advantage can even be extended to matrix-vector multipliers.

A matrix can be expanded in terms of its eigenvectors. The eigenvectors themselves are orthonormal, i.e.,

$$\vec{e}_i^{\,T} \vec{e}_j = \delta_{ij} = \left\{ \begin{matrix} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{matrix} \right\} \quad . \tag{5-1}$$

The outer product of a single eigenvector is

$$\vec{e}_i \, \vec{e}_i^{\,T} = \begin{bmatrix} (\vec{e}_i)_1(\vec{e}_i)_1 & (\vec{e}_i)(\vec{e}_i)_2 \cdots (\vec{e}_i)\,(\vec{e}_i)_n \\ (\vec{e}_i)_2(\vec{e}_i)_1 & (\vec{e}_i)_2(\vec{e}_i)_2 \cdots (\vec{e}_i)_2\,(\vec{e}_i)_n \\ (\vec{e}_i)_n(\vec{e}_i)_1 & (\vec{e}_i)_n(\vec{e}_i)_2 \cdots (\vec{e}_i)_n\,(\vec{e}_i)_n \end{bmatrix} \tag{5-2}$$

We can write

$$A = \lambda_1 \, \vec{e}_1 \, \vec{e}_1^{\,T} + \lambda_2 \, \vec{e}_2 \, \vec{e}_2^{\,T} + \ldots + \lambda_n \, \vec{e}_n \, \vec{e}_n^{\,T} \tag{5-3}$$

We now evaluate A^2. It will contain "homogeneous" terms like

$$(\vec{e}_1 \vec{e}_1^{\,T}) (\vec{e}_1 \vec{e}_1^{\,T}) = \vec{e}_1 (\vec{e}_1^{\,T} \vec{e}_1) \vec{e}_1^{\,T} = \vec{e}_1 \vec{e}_1^{\,T} \tag{5-4}$$

and "heterogeneous" terms like

$$(\vec{e}_1 \vec{e}_1^{\,T}) (\vec{e}_2 \vec{e}_2^{\,T}) = \vec{e}_1 (\vec{e}_1^{\,T} \vec{e}_2) \vec{e}_2^{\,T} = \vec{e}_1 \cdot I \cdot \vec{e}_2^{\,T} = 0 \tag{5-5}$$

Thus

$$A^2 = \lambda_1^2 \vec{e}_1 \vec{e}_1^{\,T} + \lambda_2^2 \vec{e}_2 \vec{e}_2^{\,T} + \dots + \lambda_n^2 e_n e_n^{\,T} \quad . \tag{5-6}$$

Squaring again we get A^4, etc. After m squarings we obtain

$$A^{2^m} = \lambda_1^{2m} \vec{e}_1 \vec{e}_1^{\,T} + \lambda_2^{2^m} \vec{e}_2 \vec{e}_2^{\,T} + \dots + \lambda_\nu^{2^m} \vec{e}_n \vec{e}_n^{\,T} \quad . \tag{5-7}$$

Thus, for example, 10 squarings leads to raising the λ_k's to the power 1024! Thus the convergence has been improved tremendously. Of course once we conclude

$$A^{2^m} \simeq \lambda_k^{2m} \vec{e}_k \vec{e}_k^{\,T} \quad , \tag{5-8}$$

we can extract $\vec{e}_k$ by, for example, projecting along either the rows or the columns.

The other problems with the power method were also successfully attacked in Appendix E, but the required explanations are too tedious for the text.

In the process of working on this matrix squaring algorithms, we also made some important observations and innovations regarding Singular Value Decomposition (SVD). This work is shown in Appendix F and will be referred to in more detail later.

6. ACCURACY ISSUES

As noted in Subsection 3.2, the major non-algorithm issue in accomplishing the contract goals is accuracy. We have attacked the accuracy issue in several ways. We examine these complementary approaches below.

6.1 Matrix Reconditioning

This is an important but rather subtle method suggested in Appendices E and F. The basic idea is rather simple. There may be a way to replace A by an "approximate matrix" A' such that, for the given calculation accuracy, we get closer to the true answer by using the less-accurately-posed-but-better-conditioned matrix A than by using the more-accurate-but-worse-conditioned matrix A. As an "existence proof" we showed how to remove a singularity from A; converting an unsolvable problem into a solvable one! We show here only the basic ideas.

The SVD of A can be written

$$A = s_1 \vec{v}_1 \vec{v}_1^{\,T} + s_2 \vec{v}_2 \vec{v}_2^{\,T} + \ldots + s_n \vec{v}_n \vec{v}_n^{\,T} \quad , \tag{6-1}$$

where, by convention,

$$s_1 \geq s_2 \geq \ldots \geq s_n \quad , \tag{6-2}$$

The scalar s_k is called the k+h/ singular value and $\vec{v}_k$ is called the k+n/ singular vector. For symmetric A, Eqs. (5-3) and (6-2) are equivalent. One of many interesting properties of the SVD is that, in a meaningful and well-defined sense, the best $\ell \leq n$ outer product expansion of A is

$$A^{(\ell)} = s_1 \vec{v}_1 \vec{v}_2^{\,T} + s_2 \vec{v}_2 \vec{v}_2^{\,T} + \ldots + s_\ell \vec{v}_\ell \vec{v}_\ell^{\,T} \quad . \tag{6-3}$$

Furthermore, the "goodness of fit" is given by

$$G^{(\ell)} = (s_1^2 + s_2^2 + \ldots + S_e^2)/(s_1^2 + s_2^2 + \ldots + s_n^2) \quad . \tag{6-4}$$

It appears reasonable to choose ℓ such that

$$G^{(\ell)} \sim \varepsilon(c) \quad . \tag{6-5}$$

All of this is quite reasonable and, unfortunately, usually impractical. The reason is that the SVD is seldom given and is very difficult to calculate - usually much more difficult than the eigenproblem we set out to solve. Hence we set out to invent an Approximate Singular Value Decomposition (ASVD) method which is

- Very easy to calculate and
- Leads to an approximate matrix which is between the original matrix A and the optimum approximation $A^{(\ell)}$.

The ASVD is discussed in Appendix G.

6.2 Digital Optical Processing

Aerodyne did not invent optical digital processing, but it has made some advances in this area. The history of this field is described in Appendix H. The basic concept is tó encode a digital number by a string (in space, time, or both) of analog signals. By a judicious encoding we can achieve high overall accuracy without overtaxing the dynamic range of any analog channel.

Aerodyne's contribution to this effort in this contract was to develop an arithmetic well suited to optical digital computing. In optics, since we are using analog channels, there is no a priori reason to restrict ourselves to radix 2 numbers. With binary numbers only, one extra digit is needed to carry the sign information which converts a non-negative amplitude (e.g., 16 bits) into a real (positive or negative) number. In any other base there appears to have been no way to do this same thing. We have invented a new arithmetic which (a) solves the problem and (b) reduces to the known result for binary numbers. Details are given in Appendix I. Here we simply illustrate the result for decimal (radix 10) numbers in the range -99 to +99.

For a positive number, e.g., 5, we write

$$+\,5 = 505$$

where 5 is the sign digit which can be any of the following digits: 0, 2, 4, 6, 8.

For a negative number, e.g., -8, we first complement the magnitude (subtract it from 100) to obtain 92 and write

$$n = 592$$

where 5 = 1, 3, 5, 7, or 9. All 5 values are equally valid.

We now show side by side ordinary decimal operations and operations in the new arithmetic

$$\begin{array}{r} +\,8 \\ +\,3 \\ \hline +11 \end{array} \qquad\qquad \begin{array}{c} \begin{array}{r} 208 \\ +603 \\ \hline 811 \end{array} \\ \downarrow \\ +11 \end{array}$$

$$\begin{array}{r} -8 \\ \underline{+3} \\ -5 \end{array} \qquad\qquad \begin{array}{c} \begin{array}{r} 792 \\ \underline{\text{x}203} \\ 995 \end{array} \\ \downarrow \\ -5 \end{array}$$

$$\begin{array}{r} -\ 8 \\ \underline{\text{x}\ 3} \\ -24 \end{array} \qquad\qquad \begin{array}{c} \begin{array}{r} 592 \\ \underline{\text{x}003} \\ 1776 \end{array} \\ \downarrow \\ -24 \end{array}$$

6.3 Floating Point Operation

Simple fixed point arithmetic as conceived of in all other optical digital computers will probably be inadequate for many Air Force needs. Like their *electronic counterparts*, optical computers need floating point operations. Under this contract, Aerodyne devised the only two floating point systems yet proposed for optical computers. One method (Appendix J) computes magnitudes and exponents independently and accumulates magnitudes on exponent-determined detectors. The other method (Appendix K) uses a simultaneous spatial encoding for the same purpose.

7. HARDWARE CONSIDERATIONS

The interest in optical systolic array processing developed around the country in parallel with the work on this contract and, in fact, stimulated by the work on this contract (see Appendix H). On this contract "only" one new hardware approach was developed and a new way of using electronics in iterative linear algebra problems was described.

7.1 The RUBIC Cube

Invented under this contract in the course conversations with employees of the Naval Ocean Systems Center, the Rabid Unbiased Bipolar Incoherent Calculator (RUBIC) cube is a fully three-dimensional systolic matrix-matrix multiplier (Appendix L). The basic idea is to use two CCD shifting spatial light modulators (as made by Lincoln Labs. or as could be made by Hughes) to move two-dimensional data in such a way that the proper data are registered on proper detectors at the proper time. As the proper two-dimensional spatial light modulators were not available to us, we simulated the RUBIC cube with moving masks. That is, the problem was not that the needed components were too expensive or impossible. They were simply not available for sale or use. Indeed, working with Hughes, we showed that they could be built (Appendix M). Moving black and transparent masks allowed us to test the other hardware of a RUBIC cube. We tested squaring (the key operation in eigen solution as previously noted) for a 64 x 64 tridiagonal matrix of 1's along the diagonal and neighboring elements and 0's elsewhere. That is, the matrix is

$$A = \begin{array}{c}\begin{array}{cccccccc} & \multicolumn{6}{c}{\overbrace{\hphantom{1\quad 0\quad .\quad .\quad .\quad .}}^{32}} & \end{array}\\ \left[\begin{array}{cccccccc} 1 & 1 & 0 & . & . & . & . & 0 \\ 1 & 1 & 1 & 0 & . & . & . & 0 \\ 0 & 1 & 1 & 1 & . & . & . & 0 \\ . & . & . & . & & & & . \\ . & . & . & . & & & & . \\ . & . & . & . & & & & . \\ 0 & 0 & 0 & 0 & . & . & . & 1 \end{array}\right]\end{array} \left.\vphantom{\begin{array}{c}1\\1\\0\\.\\.\\.\\0\end{array}}\right\} 32 \qquad . \qquad (7\text{-}1)$$

These data are rearranged so that the left-to-right flow follows Figure 7.1 and the up-to-down flow follows Figure 7.2. The 1's are the clear (white) regions while the 0's are black. Figure 7.3 shows the two data sets immediately before they enter the region of the detector array and before the first light pulse. The first data pulse involves some overlap as indicated in Figure 7.4. The second pulse involves more overlap as shown in Figure 7.5. At the end of the second pulse, the (1,1) component of A^2 has been computed. The overall result should be

$$A^2 = \left[\begin{array}{ccccccccc} 2 & 2 & 1 & 0 & 0 & . & . & . & 0 \\ 2 & 3 & 2 & 1 & 0 & . & . & . & 0 \\ 1 & 2 & 3 & 2 & 1 & . & . & . & 0 \\ 0 & 1 & 2 & 3 & 2 & . & . & . & 0 \\ 0 & 0 & 1 & 2 & 3 & . & . & . & 0 \\ . & . & . & . & . & & & & 0 \\ . & . & . & . & . & & & & \\ . & . & . & . & . & & & & \\ 0 & 0 & 0 & 0 & 0 & . & . & . & 2 \end{array}\right] \qquad (7\text{-}2)$$

The question we examined is light source and detector variability effects. We found that we could not approach 1% overall uniformity in lighting without diffusing the light so badly as to be quite inefficient. Relief by photographic precompensation is clearly possible. We believe, however, that independent <u>a posteriori</u> gain control on each detector is the proper approach. For an NxN detector array at most N at a time must be read out; so sequential switches, N amplifiers, and N circulating memories can accomplish this. It follows, as well, that the same mechanism can correct for typical nonlinearities (a few percent) in detector arrays since the number of possible "true detector values" is quite

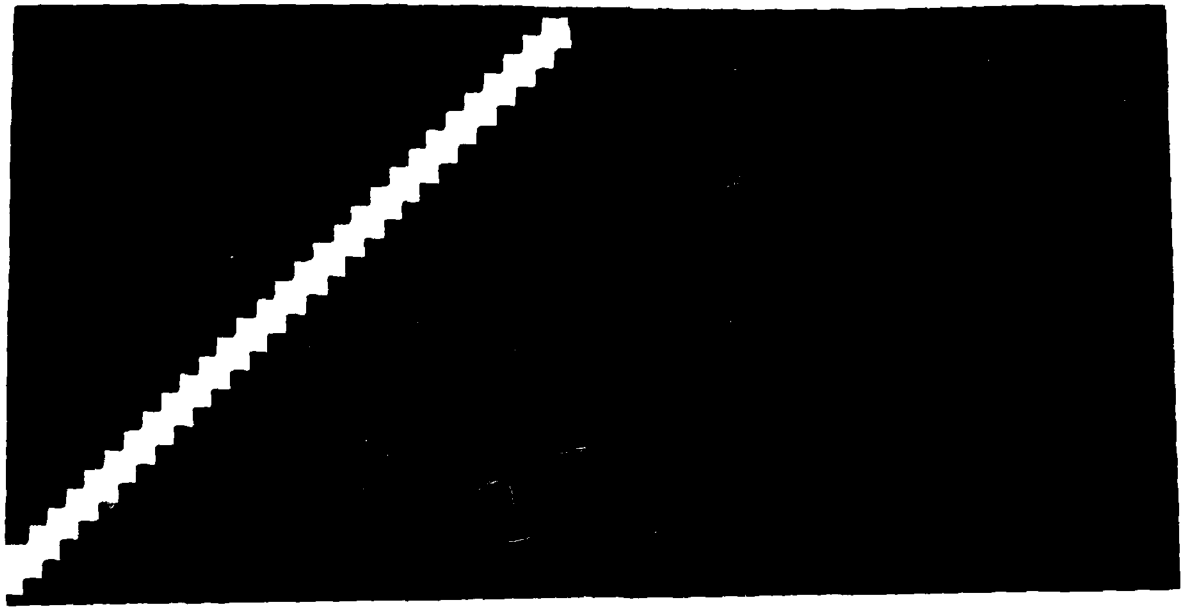


Figure 7.1. Left To Right Data Flow To Represent Matrix A Of Equation 7-1.

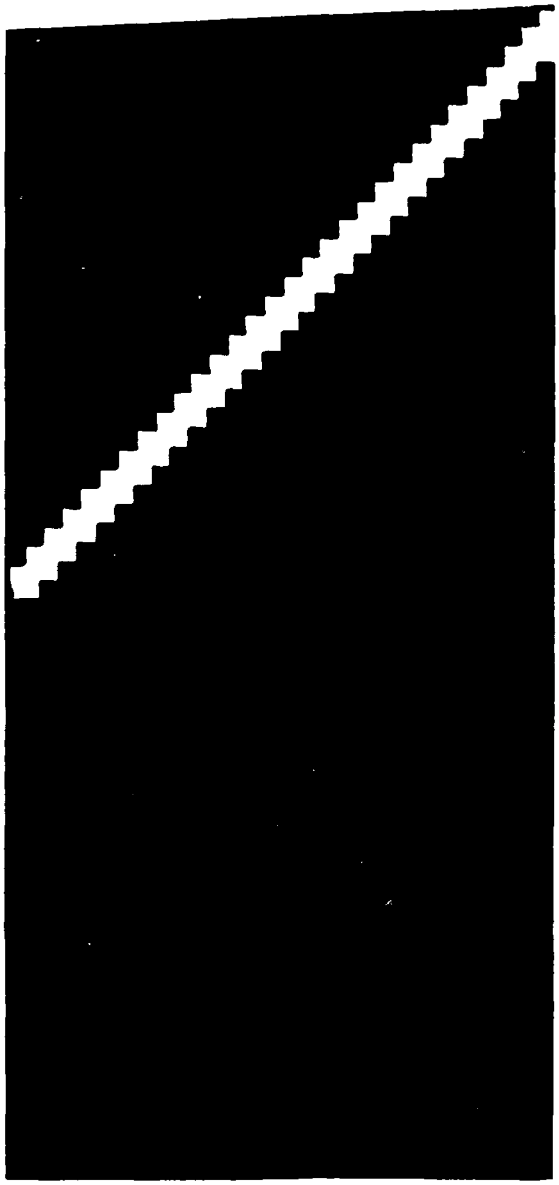


Figure 7.2. Up To Down Data Flow Representing Matrix A.

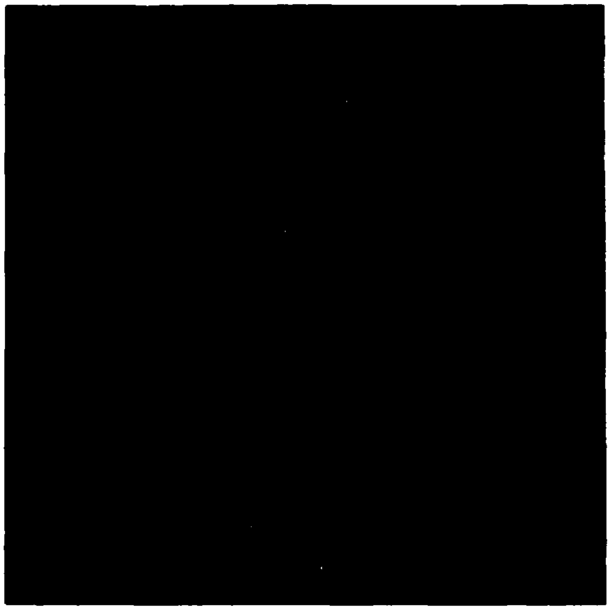


Figure 7.3. Immediately before the data enters the region of the detector array, the detectors see no signal as indicated here.

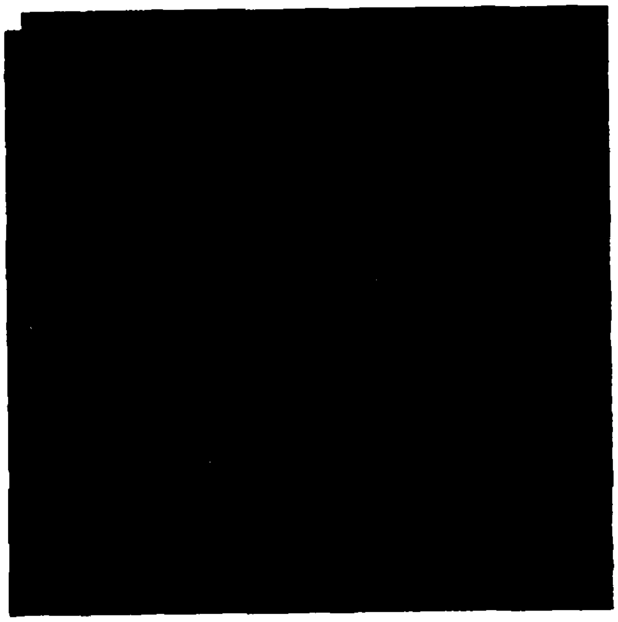


Figure 7.4. On the first data pulse a '1' is received from both data matrices in the (1,1) position on the detector array.

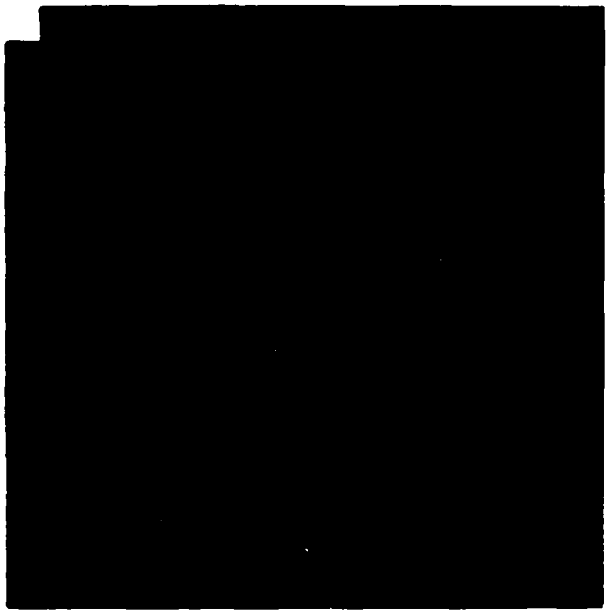


Figure 7.5. On the second pulse 4 detectors receive unit pulses.

small. We conclude that digital optical implementation of matrix squaring is quite feasible with components which have not but could be built. The speed, cost, size, and power advantages relative to current supercomputers make this appear quite worthwhile.

7.2 Iterative Algorithm Operation

Two basic types of algorithms can be devised for problems such as least squares, matrix inversion, and eigenproblems: iterative and direct. The power method we have chosen is an iterative method. Direct methods require a foreknown number of cycles but (unlike the iterative case), direct methods require full accuracy in each cycle. Thus it is not clear a priori which will work faster since the iterative schemes can use fast analog electronics (not of digital accuracy) in the loop. Thus iterative schemes require more cycles but the cycles can be faster.

Under this contract we worked out the feedback logic for iterative methods in general (Appendix N). In the power method one nonlinear step is required in each cycle: a renormalization to keep the result from growing either too large or too small. The Aerodyne approach, among other things, describes what may be called a "lagging renormalization" method which allows each digit to be renormalized and recycled in the same clock time in which it is generated.

8. CONCLUSION

This contract began with an inadequate algorithm to be implemented in an as-yet-unspecified manner on slow, inaccurate, analog optical hardware. It concluded with a vastly improved algorithm which can be implemented by well-defined methods on highly-accurate, digital optical hardware. The need now is no longer to determine what to do but to do what we have already learned how to do in principle. Significant advantages in speed, size, cost, and power consumption over electronics should result.

APPENDIX A

PRE-CONTRACT STATUS OF OPTICAL EIGEN PROBLEM ANALYSIS

Reprinted from Applied Optics, Vol. 20, page 2263, July 1, 1981
Copyright © 1981 by the Optical Society of America and reprinted by permission of the copyright owner

Eigenvector determination by noncoherent optical methods

H. J. Caulfield, David Dvore, J. W. Goodman, and William Rhodes

An iterative method for finding the eigenvectors and eigenvalues of a matrix via incoherent optical matrix-vector multiplication and simple electronic feedback is described.

I. Introduction

A variety of methods have been developed for doing certain simple matrix operations, e.g., multiplying a matrix by a vector, using optical methods.[1,2] These methods are of interest because they perform all or most of the required operations in parallel and thus potentially offer extremely high speed. More complicated matrix operations are as yet extremely difficult to carry out by optics. The finding of eigenvalues and eigenvectors of large matrices is quite difficult and slow by digital methods. Of course, the matrix of eigenvalues can be used to invert the matrix, so solving the eigenvalue problem is tantamount to doing matrix inversion. An iterative approach to matrix inversion has been attempted optically,[3] but it requires for convergence estimation of the largest eigenvalue. This is easily done by forming the square root of the squares of the elements of the matrix. We offer here a matrix inversion method of somewhat greater generality. In particular, we will find the eigenvectors and eigenvalues sequentially.

II. Method

The proposed optical approach utilizes an iterative method of computing eigenvalues and eigenvectors, known in linear algebra as the power method,[4,5] based on the orthogonality of the eigenvectors. This method works well if the matrix is of the real symmetric form assumed by the covariance matrix of a real vector. This guarantees real eigenvalues. We have no general test for its applicability to other cases. We suppose we have an $N \times N$ matrix M of rank N with a full set of eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_N$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_N$. We assume that the eigenvalues are not repeated and that they are numbered in order of decreasing magnitude. Thus

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_{N-1}| > |\lambda_N|. \tag{1}$$

As a starting point we choose some arbitrary input vector

$$\mathbf{V}_{(0)} = c_1\mathbf{e}_1 + \ldots + c_N\mathbf{e}_N. \tag{2}$$

Multiply V(0) by M yields

$$\mathbf{V}_{(1)} = c_1\lambda_1\mathbf{e}_1 + \ldots + c_N\lambda_N\mathbf{e}_N. \tag{3}$$

With successive such matrix multiplications, we obtain the general term

$$\mathbf{V}_{(n)} = M\mathbf{V}_{(n-1)} = c_1\lambda_1^n\mathbf{e}_1 + \ldots + c_N\lambda_N^n\mathbf{e}_N. \tag{4}$$

So long as the starting vector $\mathbf{V}_{(0)}$ contains some of eigenvector $\mathbf{e}_1$ (for which, recall, the corresponding eigenvalue is greatest in magnitude), the first term of Eq. (4) comes to domination after a sufficient number of iterations. Thus for n sufficiently large, we have

$$\mathbf{V}_{(n)} \simeq c_1\lambda_1^n\mathbf{e}_1. \tag{5}$$

Similarly, with an additional iteration, we have

$$\mathbf{V}_{(n+1)} \simeq c_1\lambda_1^{n+1}\mathbf{e}_1, \tag{6}$$

and, therefore,

$$\mathbf{V}_{(n+1)} \simeq \lambda_1\mathbf{V}_{(n)} \tag{7}$$

This relationship holds on a component by component basis, and thus the value of λ_1 can be solved for. The rate at which the process converges is determined by the ratio $|\lambda_1|/|\lambda_2|$.

If λ_1 is significantly larger or smaller than unity in magnitude, $V_{(n)}$ may become unacceptably large or small after a number of iterations, and in practice we must normalize at each iteration to keep the vectors of

William Rhodes is with Georgia Institute of Technology, Department of Electrical Engineering, Atlanta, Georgia 30332; J. W. Goodman is with Stanford University, Department of Electrical Engineering, Stanford, California 94305; the other authors are with Aerodyne Research, Ind., Bedford Research Park, Bedford, Massachusetts 01730.

Received 21 February 1981.
0003-6935/81/132263-03$00.50/0.

controlled size. Thus we might obtain an output after n iterations which we normalize to $\mathbf{U}_{(n)}$ so that $|\mathbf{U}_{(n)}| = 1$. Multiplying $\mathbf{U}_{(n)}$ by M produces an output $\mathbf{W}_{(n+1)}$. We normally would normalize $\mathbf{W}_{(n+1)}$ to $\mathbf{U}_{(n+1)}$, but if $(n + 1)$ is the terminal iteration, we can write

$$\mathbf{W}_{(n+1)} \simeq \lambda_1 \mathbf{U}_{(n)}. \tag{8}$$

To check for termination we compare the values of $\mathbf{W}_{(n)}$ and $\mathbf{W}_{(n+1)}$ either on a magnitude or a component-by-component basis. If the percentage change is acceptable, we terminate the iteration.

III. Implementation

It is clear that we need an optical matrix multiplier for speed with certain rapid electronic processing between matrix multiplications. Figure 1 shows the configuration in schematic terms. The optical matrix multiplier devised by Goodman *et al.* [2] seems to be ideally suited for this purpose. The feedback method of Psaltis *et al.*[3] is based on Goodman's method and appears to have all the necessary components to implement this scheme.

IV. Representation of Bipolar Quantities

Because we want to use nonnegative definite masks and incoherent light, the handling of negative quantities requires some encoding of the vectors and matrix to achieve monopolar operation. Let the matrix be

$$M = \begin{pmatrix} m_{11} & \cdot & m_{1N} \\ & & \\ m_{N1} & \cdot & m_{NN} \end{pmatrix} \tag{9}$$

and the kth input vector be $\mathbf{V}_k$. We write

$$M = M_+ - M_-, \tag{10}$$

where M_+ and M_- have nonnegative entries only, and the convention is adopted that

$$m_{mn}^+ = 0 \text{ if } m_{mn} \leq 0 \qquad m_{mn}^- = 0 \text{ if } m_{mn} \leq 0. \tag{11}$$

Similarly, let

$$\mathbf{V}_{(k)} = \mathbf{V}_{(k)}^+ - \mathbf{V}_{(k)}^-. \tag{12}$$

Then

$$\mathbf{V}_{(k+1)} = M\mathbf{V}_{(k)} \tag{13}$$

or

$$\begin{aligned} \mathbf{V}_{(k+1)}^+ - \mathbf{V}_{(k+1)}^- &= (M^+ - M^-)[\mathbf{V}_{(k+1)}^+ - \mathbf{V}_{(k+1)}^-] \\ &= [M^+\mathbf{V}_{(k+1)}^+ + M^-\mathbf{V}_{(K+1)}^-] \\ &\quad - [M^+\mathbf{V}_{(k+1)}^- + M^-\mathbf{V}_{(k+1)}^+]. \end{aligned} \tag{14}$$

Thus

$$\mathbf{V}_{(k+1)}^+ = M^+\mathbf{V}_{(k+1)}^+ + M^-\mathbf{V}_{(k+1)}^-, \tag{15}$$

$$\mathbf{V}_{(k+1)}^- = M^+\mathbf{V}_{(k+1)}^- + M^-\mathbf{V}_{(k+1)}^+. \tag{16}$$

We replace the vector $\mathbf{V}_{(k)}$ of N real components with r ew vectors

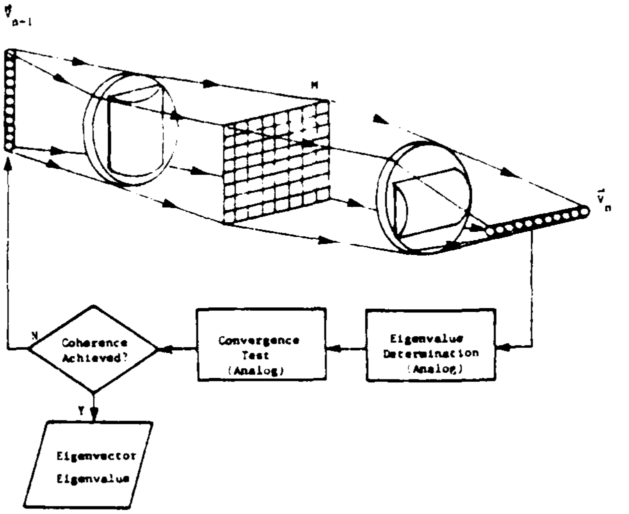


Fig. 1. Heart of the eigenvector analysis device is the optical matrix multiplier.[1,2] Analog circuitry provides the required feedback.

$$\mathbf{y}_{(k)} = \begin{bmatrix} \mathbf{V}_{(k)}^+ \\ \mathbf{V}_{(k)}^- \end{bmatrix}, \tag{17}$$

which contains $2N$ nonnegative components. We then operate on that vector by a new rank $2N$ matrix:

$$B = \begin{bmatrix} M^+ & M^- \\ M^- & M^+ \end{bmatrix} \tag{18}$$

to obtain

$$\mathbf{y}_{(k+1)} = B\mathbf{y}_{(k)} = \begin{bmatrix} V_{(k+1)}^+ \\ V_{(k+1)}^- \end{bmatrix}. \tag{19}$$

Note that neither $\mathbf{y}_{(k)}$ nor B has negative components, so incoherent optics is quite adequate to represent them both.

V. Finding New Eigenvectors and Eigenvalues

We suppose the first $K - 1$ eigenvalues and eigenvectors of M have been found to be λ_1, $\mathbf{e}_1$, λ_2, $\mathbf{e}_2$, . . . , λ_{K-1}, $\mathbf{e}_{k-1}$. We want want to find the kth eigenvalue and eigenvector. To do this we form a new matrix:

$$M_k = \prod_{n=1}^{k-1} (M - \lambda_n I), \tag{20}$$

where Π is the matrix product operator, and I is the unity matrix (which converts all vectors into themselves). We suppose M_k operates on an eigenvector $\mathbf{e}$ of M having an eigenvalue λ. Then

$$M_k\mathbf{e} = \left[\prod_{n=1}^{k-1} (\lambda - \lambda_n)\right]\mathbf{e} \tag{21}$$

Thus M_k and M have the same eigenvectors. Note, though, that for $\mathbf{e}_1$, . . . , $\mathbf{e}_{k-1}$, the M_k eigenvalues are zero. As our method tends to find that eigenvector with eigenvalue of highest absolute value, it will find an eigenvector $\mathbf{e}_k (\neq \mathbf{e}_1, \ldots, \mathbf{e}_{k-1})$. Call the M_k eigenvalue for $\mathbf{e}_k$ "Λ_k". Then we can find the corresponding M eigenvalue λ_k by solving the equation

$$\prod_{n=1}^{k-1} (\lambda_k - \lambda_n) = \Lambda_k \tag{22}$$

Thus we can find all the eigenvectors and eigenvalues of M sequentially

VI. Concluding Remarks

A hybrid electronic and incoherent optical approach for finding eigenvalues and eigenvectors of matrices has been proposed. The optical hybrid appears particularly attractive because of the extremely high speed with which the iterative matrix multiplications can be performed. Its most important potential application appears to be in problems in which the rank of the matrix is so large that standard digital methods are too slow. Accuracy required for complete implementation of the processing scheme depends on the ratio of the largest eigenvalue to the smallest (the condition number of the matrix). Specifically, to find λ_n, all larger eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{n-1}$ must be known to within an error of $<|\lambda_n|$. Variations on the method allow some relaxation in accuracy requirements.[6] The power method proposed here suffers from many of the range and accuracy problems common to optical processors. For higher accuracy we might use the eigenvectors determined optically as inputs for a few iterations of the digitally implemented method. In so doing we would utilize the optical processor for speed and the digital processor for numerical accuracy.

The optical matrix multiplier proposed[1,2] has the capability of handling multiple input vectors in parallel. This capability should be of advantage, if used properly, to allow for degenerate eigenvalues. If two eigenvalues are identical, unique eigenvectors are no longer defined Rather, any vector in a plane defined by two spanning vectors is an eigenvector. Of course, this is extendable to more than two eigenvalues. We conjecture (without proof) that by starting with N orthogonal vectors we can guarantee at least M eigenvectors for each M-degenerate eigenvalue. This is an automatic check for eigenvalue degeneracy as well as an automatic generator of spanning vectors for the corresponding eigenvectors By a Gramm-Schmidt process we can orthogonalize those vectors and reduce them to their minimum number M.

This work was performed under contrast F19628-80-C-0082, Rome Air Development Center, Deputy for Electronics Technology, Hanscom AFB, Mass. 01731.

References

1. J. W. Goodman, A. R. Dias, and L. M. Woody, Opt. Lett. 2, 1 (1978).
2. J. W. Goodman, A. R. Dias, L. M. Woody, and L. Erickson, Proc Soc. Photo-Opt. Instrum. Eng. 485 (1979)
3. D. Psaltis, D. Casasent, and M. Carlotto, Opt Lett 4, 348 (1979).
4. J. H. Wilkinson, *The Algebraic Eigenvalue Problem* (Clarendon, Oxford, 1965)
5. D. M. Young and R. T Gregory, *A Survey Of Numerical Mathematics II* (Addison-Wesley, Reading, Mass., 1973)
6. G. Strang, *Linear Algebra and Its Applications* (Academic, New York, 1976), p. 175.

APPENDIX B

PRE-CONTRACT STATUS OF THE OPTICAL MATRIX PROCESSOR

Fully parallel, high-speed incoherent optical method for performing discrete Fourier transforms

J. W. Goodman, A. R. Dias, and L. M. Woody

Department of Electrical Engineering, Stanford University, Stanford, California 94305

Received September 12, 1977

An incoherent optical data-processing method is described, which has the potential for performing discrete Fourier transforms of short length at rates far exceeding those afforded by both special-purpose digital hardware and representative coherent optical processors.

We report here on an incoherent optical method for performing discrete Fourier transforms (DFT's), which has the potential for an extremely high data-throughput rate. The DFT operation may be viewed as a process of multiplying an input vector **f** (consisting of N possibly complex-valued input samples) times an $N \times N$ matrix $\mathcal{H}$ [the n,mth element being $\exp(-j2\pi nm/N)$] to yield an output vector **g** (consisting of the N complex Fourier coefficients); thus we desire to perform

$$\mathbf{g} = \mathcal{H}\mathbf{f}. \qquad (1)$$

Two separate issues must be addressed in describing the method of interest here: (1) How do we perform the matrix product in a highly parallel and fast way? (2) How do we perform complex arithmetic using incoherent light, for which only nonnegative and real quantities (intensities) can be manipulated?

To address the first issue, suppose that the elements of **f** and $\mathcal{H}$ are nonnegative and real. Then the system shown in Fig. 1 can be used to perform the matrix-vector product. The elements of **f** are entered in parallel by controlling the intensities of N light-emitting diodes (LED's). Lenses L_1 and L_2 image the LED array horizontally onto the matrix mask M while spreading the light from any single LED vertically to fill an entire column of the matrix mask. Lens L_3 is a field lens. The matrix mask M consists of $N \times N$ subcells, each containing a transparent area proportional to one of the matrix elements. Lens L_4 is a cylindrical lenslet array, which is not essential to the operation of the system but which can be used to improve light efficiency. Lens combination L_5 collects all light from a given row and brings it to focus on one element of a vertical array of N photodetectors. Each photodetector measures the value of one component of the output vector **g**.

To permit the multiplication of a matrix $\mathcal{H}$ with complex elements times a vector **f** with complex elements, we decompose each of these quantities as follows[1,2]

$$\mathbf{f} = \mathbf{f}^{(0)} + \mathbf{f}^{(1)}\exp(j2\pi/3) + \mathbf{f}^{(2)}\exp(j4\pi/3),$$
$$\mathcal{H} = \mathcal{H}^{(0)} + \mathcal{H}^{(1)}\exp(j2\pi/3) + \mathcal{H}^{(2)}\exp(j4\pi/3), \qquad (2)$$

where $\mathbf{f}^{(0)}$, $\mathbf{f}^{(1)}$, and $\mathbf{f}^{(2)}$ each consist of N real and nonnegative elements, and $\mathcal{H}^{(0)}$, $\mathcal{H}^{(1)}$, and $\mathcal{H}^{(2)}$ consist of *$N \times N$ real and nonnegative elements.* If the output vector **g** is similarly decomposed, then we find that the overall matrix–vector product can be expressed as

$$\begin{bmatrix} \mathbf{g}^{(0)} \\ \mathbf{g}^{(1)} \\ \mathbf{g}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathcal{H}^{(0)} & \mathcal{H}^{(2)} & \mathcal{H}^{(1)} \\ \mathcal{H}^{(1)} & \mathcal{H}^{(0)} & \mathcal{H}^{(2)} \\ \mathcal{H}^{(2)} & \mathcal{H}^{(1)} & \mathcal{H}^{(0)} \end{bmatrix} \begin{bmatrix} \mathbf{f}^{(0)} \\ \mathbf{f}^{(1)} \\ \mathbf{f}^{(2)} \end{bmatrix}. \qquad (3)$$

Thus, complex operations can be performed at a price of a factor of 3 in the length of the input and output vectors.

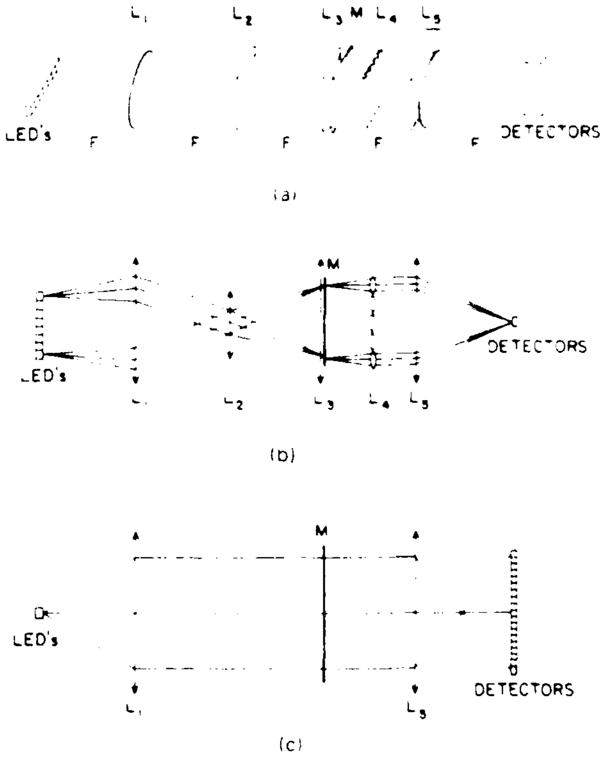


Fig. 1. Incoherent optical processor configuration: (a), pictorial view; (b), top view; (c), side view.

0146-9592/78/0100-0001$0.50/0

Simple electronic circuits for producing the components $\mathbf{f}^{(0)}$, $\mathbf{f}^{(1)}$, and $\mathbf{f}^{(2)}$ from $\mathbf{f}$ exist,[1] as do simple circuits for producing the real and imaginary parts of $\mathbf{g}$ from $\mathbf{g}^{(0)}$, $\mathbf{g}^{(1)}$, and $\mathbf{g}^{(2)}$.

Experiments have been carried out to verify the ability to perform complex arithmetic. The source was an unfiltered, linear-filament, clear-envelope, incandescent bulb. The 30 × 30 matrix mask used to perform a 10-point DFT is shown in Fig. 2. This mask is designed so that the three entire vectors $\mathbf{f}^{(0)}$, $\mathbf{f}^{(1)}$, and $\mathbf{f}^{(2)}$ are entered side by side, whereas the three output components $g_k{}^{(0)}$, $g_k{}^{(1)}$, and $g_k{}^{(2)}$ for the kth Fourier coefficient appear side by side. Thus the output display shows each DFT component as a triplet of real and nonnegative components.

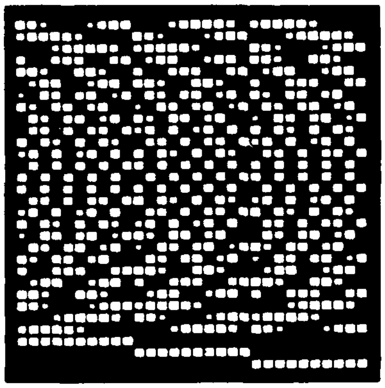


Fig. 2. Matrix mask for a 10-point DFT.

For this experiment the input functions were entered by hand as masks placed against the matrix mask, and output functions were detected on a 1024-element Reticon CCD detector array. Figure 3 shows both theoretical output distributions and experimentally obtained output distributions, the latter being photographed from an oscilloscope display. In parts (a) and (b), the function to be transformed consists of the sequence (1,0,0,0,0,0,0,0,0,0). The resulting DFT should be entirely real and of constant magnitude. As shown in these figures, the DFT components along the real axis are all nonzero and equal, whereas the components along 120° and 240° are all zero.

In parts (c) and (d), the input sequence was entirely real and constant. The DFT consists of a large, real zero-frequency component (on the far right), followed by triplets of equal strength for all other DFT components. Some thought shows that any DFT component with elements $g_k{}^{(0)}$, $g_k{}^{(1)}$, and $g_k{}^{(2)}$ exactly equal is equivalent to a zero result. Hence all DFT components, except the zero-frequency component, are zero.

Parts (e) and (f) show the results when the entire

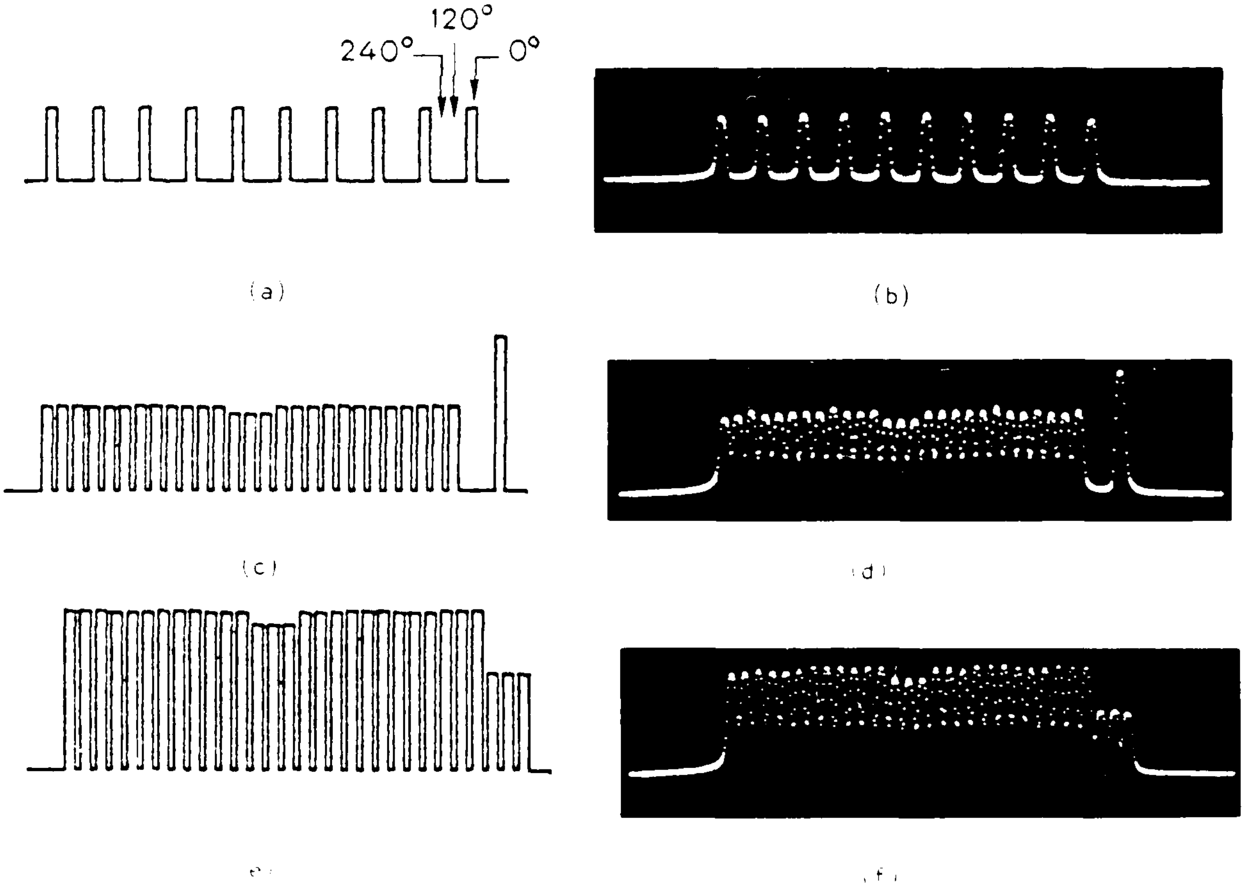


Fig. 3. Theoretical [(a), (c), (e)] and experimental [(b), (d), (f)] DFT results.

matrix mask is uniformly illuminated. In this case, some thought shows that the input is effectively a sequence containing all zeros. The output DFT shows triplets of equal strength, or a sequence of all zeros for the output.

A system composed of 96 high-speed LED's and 96 avalanche photodiodes would be capable of performing a 32-point DFT. Commercially available components have sufficient bandwidth, output power, and sensitivity to permit such a DFT to be performed every 10 nsec. The total throughput rate for such a processor is about 3×10^9 complex samples per second, whereas a corresponding number for special-purpose digital array processors is about 3×10^5 complex samples per second and a representative coherent optical processor[3] has a throughput of 3×10^7 real samples per second.

The chief significance of this processor is that the input data can be entered in parallel, and it is this fact that leads to its high throughput rate. Another system recently described[4,5] performs a similar matrix–vector product, but the data must be entered serially, and as a consequence the throughput rate is much lower. The processor described here is especially well suited for problems in which the elements of the input vector **f** are gathered by parallel sensors. Of course, matrices other than the DFT matrix can also be used if desired.

This work was supported by the Office of Naval Research.

References

1. J. W. Goodman and L. M. Woody, "Method for performing complex-valued linear operations on complex-valued data using incoherent light," Appl. Opt. **16**, 2611 (1977).
2. If one is sufficiently clever in eliminating unwanted terms at the output, real and imaginary components on biases can be used. However, the dynamic range of the system is reduced by such an approach.
3. We refer specifically to a system with an electron-beam addressed DKDP input light valve, which is capable of entering 10^6 data points 30 times per second. See D. Casasent, Proc. IEEE **65**, 143 (1977).
4. R. P. Bocker, Appl. Opt. **13**, 1670 (1974).
5. M. A. Monahan, K. Bromley, and R. P. Bocker, Proc. IEEE **65**, 121 (1977).

APPENDIX C

NEW ANALOG OPTICAL COMPUTER FOR ALGEBRAIC EQUATIONS

THIS APPENDIX HAS INTENTIONALLY BEEN LEFT BLANK.

APPENDIX D

APPLICATION OF THE OPTICAL PROCESSOR OF APP. C TO THE EIGEN PROBLEM

THIS APPENDIX HAS INTENTIONALLY BEEN LEFT BLANK.

APPENDIX E

ALGORITHM IMPROVEMENTS FOR THE EIGEN PROBLEM

Reprinted from Applied Optics, Vol. 22, page 2075, July 15, 1983

Algorithm improvements for optical eigenfunction computers

John Gruninger and H. J. Caulfield

Prior iterative approaches to optical eigenfunction solution have at least three major problems: slow convergence (sometimes); decreasing accuracy after the first solution; and imperfect parallel renormalization (leading to poor use of system dynamic range and hence poor accuracy). We introduce new approaches and algorithms to solve these problems. The new algorithms lead to a tight error bound on eigenvalues and an automatic handling of degenerate or near degenerate eigenvalues. Applications are discussed.

I. Introduction

There has been a recent increase in interest in using optics to perform certain simple algebraic operations[1-4] and to use those optical operators to perform iterative operations solving practical problems.[5-8] We are concerned here with the use of optical algebraic operations to solve eigenvector problems. Prior work[5,6] used optical vector–matrix multiplication to carry out a classical procedure called the power method. We will review the power method here, indicate the three major problems from which it suffers, and show how those problems can be solved.

Let us assume that we have a full rank symmetric $N \times N$ matrix A. We know that A has N real eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ and N eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N$ satisfying

$$\mathbf{e}_m \cdot \mathbf{e}_n = \delta_{mn}. \qquad (1)$$

Futhermore $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N$ span the allowable vector space. Thus an arbitrary vector $\mathbf{V}_0$ can be written

$$\mathbf{V}_0 = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 +, \ldots, + a_N\mathbf{e}_N, \qquad (2)$$

where $a_1, a_2, \ldots, a_N$ are scalers.

Let us write

$$\mathbf{V}_1 = A\mathbf{V}_0. \qquad (3)$$

Applying A to successive $\mathbf{V}_k$ values, we obtain

$$\mathbf{V}_2 = \mathrm{A}\mathbf{V}_1 = A^2\mathbf{V}_0,$$

$$\mathbf{V}_p = \mathrm{A}\mathbf{V}_{p-1} = A^p\mathbf{V}_0. \qquad (4)$$

Since

$$\mathrm{A}^p a_k\mathbf{e}_k = a_k\mathrm{A}^p\mathbf{e}_k = a_k\lambda_k{}^p\mathbf{e}_k, \qquad (5)$$

therefore,

$$\mathbf{V}_p = a_1\lambda_1{}^p\mathbf{e}_1 + a_2\lambda_2{}^p\mathbf{e}_2 + . \quad . + a_N\lambda_N{}^p\mathbf{e}_N \qquad (6)$$

If

$$|\lambda_l| > |\lambda_m| \qquad (7)$$

for $m \neq l$, the lth eigenvector will (above some number of iterations p) come to dominate $\mathbf{V}_p$, so for p sufficiently large

$$\mathbf{V}_p \simeq a_l\lambda_l{}^p\mathbf{e}_l \qquad (8)$$

Of course, we recognize this condition by the fact that

$$\mathbf{V}_p \propto \mathbf{V}_{p-1} \qquad (9)$$

Indeed

$$\mathbf{V}_p \simeq \lambda_l\mathbf{V}_{p-1} \qquad (10)$$

We can now discuss the problems with this method. First, the convergence can be very slow. If we require $P = 10^6$, even an optical processor is slow. The second problem relates to deflation, that is, finding the smaller $|\lambda_k|$ values and the corresponding $\mathbf{e}_k$ values. While there are many deflation methods, most lead to answers with decreasing accuracy. The primary problem is that most deflation methods assume a perfect accuracy of previously calculated results. Thus errors tend to accumulate, and very significant errors can occur for relatively low values of $|\lambda_k|$. It is sometimes true that we want only a few of the dominant eigenvectors, but it would be unwise to accept this limitation if it can be avoided. Third, we need a fully parallel way to deal with the normalization problem. Otherwise we lose the advantages of parallel optical computation. The renormalization referred to is a necessity forced on us by the fact that optics uses fixed point rather than floating point calculations. The vector components of

The authors are with Aerodyne Research, Inc., 45 Manning Road, Billerica, Massachusetts 01821.

Received 13 January 1983.
0003-6935/83/142075-06$01.00/0.

V_p may be either very large (if $|\lambda_k| > 1$) or very small (if $|\lambda_k| < 1$). Thus we renormalize after each iteration. To renormalize we must estimate the maximum component and set the input so that the maximum output value is large but not beyond the range of our optical computer. How do we estimate that component? How can we check for saturated components without looking at the components sequentially and thus slowing down operations? Besides these major problems there are also unanswered questions on how to handle degenerate solutions and how to estimate accuracy etc.

Having introduced the problems with prior approaches, we move to a discussion of possible solutions to those problems.

II. Convergence Problem

By reformulating the power method, we can introduce considerably more parallelism in each iteration and thus reduce the number of iterations dramatically. For example, a problem which would have required 10^6 iterations by the prior method will now require only 20 iterations. In general K iterations with the new power method is equivalent to 2^K iterations of the prior method. We achieve this by using the matrix squaring method.[9] We briefly explain the method as well as add our own observations concerning the advantages of the matrix squaring algorithm over the power method just described. The reader will note that matrix squaring is itself a power method, but it operates on the given matrix itself rather than operating on a vector while leaving the matrix unchanged.

The matrix squaring method for eigenvalue eigenvector analysis is based on the spectral representation of a symmetric matrix:

$$A = E\Lambda E^T, \qquad (11)$$

where Λ is a diagonal matrix containing the eigenvalues of A and E is an orthogonal matrix whose columns are the eigenvectors of A. That is, the kth column of E is the eigenvector $\mathbf{e}_k$ associated with the eigenvalue λ_k. The orthogonality of eigenvectors of symmetric matrices is expressed in matrix form as

$$EE^T = E^TE = I \qquad (12)$$

We use this property to express powers of A. Writing A^n as $(E\Lambda E^T)(E\Lambda E^T) \ldots (E\Lambda E^T)$ with n factors,

$$A^n = E\Lambda^n E^T \qquad (13)$$

Performing the matrix multiplications, A^n can be expressed as

$$A^n = \sum^N \lambda_i^n \mathbf{e}_i\mathbf{e}_i^T \qquad (14)$$

where N is the dimension of A.
For convenience we will assume that eigenvalues are ordered

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_N| \qquad (15)$$

For the case that λ_1 is a nondegenerate eigenvalue, for sufficiently large n,

$$A^n \approx \lambda_1^n \mathbf{e}_1\mathbf{e}_1^T \qquad (16)$$

Each column of A^n is proportional to the normalized eigenvector $\mathbf{e}_1$, each row is proportional to the transpose $\mathbf{e}_1^T$. To obtain the eigenvalue λ_1, any column of A^n can be multiplied by A. The power to which A must be raised depends on the dominance of λ_1. The convergence is of the order of $(\lambda_2/\lambda_1)^n$. If n is sufficiently large the rank of A^n is one, and each column can be normalized to $\mathbf{e}_1$. This operation forms a test to ensure that the eigenvalue is nondegenerate. Since the spectral decomposition of A contains contributions from all its eigenvectors, all the dominant eigenvectors are contained in A^n. Thus the rank of A^n is the degeneracy of the dominant eigenvalue. For a degenerate case, say degeneracy two, where two eigenvectors have the same eigenvalue, A^n can be approximated by

$$A^n \approx \lambda_1^n(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T), \qquad (17)$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are orthogonal but are associated with λ_1. Each column of A^n is a linear combination of $\mathbf{e}_1$ and $\mathbf{e}_2$ and hence is an eigenvector of A. However, on normalization, the columns of A^n will not be identical. The rank of A^n is equal to two, the degeneracy of λ_1. Any two linearly independent columns of A^n can be used to obtain two orthogonal eigenvectors of A. The clear advantage of the matrix squaring method is that all the degenerate eigenvectors of an eigenvalue can be obtained at once because no mechanism favors one over the others.

By actually forming A^n, a useful error bound for the magnitude of the dominant eigenvalue can be obtained. The bounds can be derived as follows. If we raise A to an even power, $n = 2m$, all the eigenvalues of A^n are positive. Its dominant eigenvalue λ_1^n will be smaller than its trace, which is equal to the sum of all its eigenvalues,

$$TrA = \sum^N A_{ii}^n = \sum^N \lambda_i^n.$$

Here N is the dimension of A. On the other hand, the dimension times the dominant eigenvalue is larger than the trace. Therefore,

$$\lambda_1^n < TrA^n < N\lambda_1^n \qquad (18)$$

Rearranging this and taking the nth root yield

$$\left(\frac{1}{N}\right)^{1/n}(TrA^n)^{1/n} < |\lambda_1| < (TrA^n)^{1/n} \qquad (19)$$

For n sufficiently large, $(1/N)^{1/n}$ approaches one to within the precision of the processor. A good estimate of $|\lambda_1|$ is the mean of the upper and lower bounds,

$$|\lambda_1| = \left[1 + \left(\frac{1}{N}\right)^{1/n}\right](TrA^n)^{1/n}/2, \qquad (20)$$

with error

$$\pm\delta = \left[1 - \left(\frac{1}{N}\right)^{1/n}\right](TrA^n)^{1/n}/2. \qquad (21)$$

It should be noted that the matrix squaring method raises A to an even power, and hence we are finding eigenvectors and eigenvalues of A^2 rather than A. The eigenvectors of A^2 will be identical to eigenvectors of A except in the case where A has two roots which satisfy

$\lambda_i = -\lambda_j$. In this case A^2 has a doubly degenerate eigenvalue λ_i^2. Only two particular linear combinations of the degenerate eigenvectors of A^2 will be eigenvectors of A. When a degeneracy or an apparent degeneracy occurs, a new eigenvalue eigenvector problem must be solved. We use the orthogonalized linear independent columns V_k, of A^n to form a new matrix G given by

$$G_{kl} = V_k^T A V_l \qquad (22)$$

The dimension of G is of the order of the apparent degeneracy. The eigenvectors of G yield the linear combinations of the V_k, which are eigenvectors of A. For a true degeneracy G is already diagonal.

If we accomplish matrix–matrix multiplication by sequential matrix–vector multiplications using the columns of the matrix as vectors, we require N matrix vector multiplications to accomplish one matrix squaring. If convergence requires M squarings, a total of MN matrix–vector cycles will be needed. Accomplishing the raising of A to the same power by the prior method would require 2^M matrix–vector multiplications. For slowly converging systems

$$2^M >>> 1, \qquad (23)$$

while M and N may be relatively small. For example, if $M = 20$ and $N = 50$, we would need 20 matrix–matrix multiplications by matrix squaring or 1000 matrix–vector multiplications, whereas 10^6 matrix–vector multiplications would be required by the prior power method. Clearly the convergence is improved dramatically by matrix squaring even if the hardware is restricted to vector–matrix multipliers.

III. Deflation

Deflation remains a vexing problem in that it tends to lead to decreasing accuracy in subsequent eigensolutions. This problem is magnified when only low precision is available. While we have arrived at no final solutions to the problem, we suggest two methods which may prove fruitful. The main problem is that one finds only approximate eigenvalues $\tilde{\lambda}_1$ and approximate eigenvectors $\tilde{\mathbf{e}}_1$ rather than the exact quantities. We seek methods which will be adaptable to the matrix squaring approach and for which the errors do not accumulate as successive eigenvalues and eigenvectors are found. The latter restriction is the most important for processing with low precision. Common approaches which can be incorporated into the matrix squaring method include deflation by subtraction and deflation by orthogonalization. Perhaps the most obvious technique is deflation by subtraction in which a new matrix to use for the power method is generated from A by subtracting $\tilde{\lambda}_1 \tilde{\mathbf{e}}_1 \tilde{\mathbf{e}}_1^T$ from A. This approach was first suggested by Hotelling.[10] However, in practice, errors in both the estimated eigenvalue and eigenvector can lead to numerical errors when the power method is applied to the deflated matrix to obtain λ_2.[1] For these reasons, the method should be used only in formal analysis.

The deflation by orthogonalization method addresses these difficulties by choosing a trial vector **V** for the power method, which is orthogonal to $\mathbf{e}_1$. However, since we only know $\mathbf{e}_1$ and λ_1 approximately, the orthogonalization is only approximate, and the true $\mathbf{e}_1$ component grows and may become dominant again.[11] A wise procedure is to reorthogonalize the current vector to previously found eigenvectors from time to time. A useful way to perform the orthogonalization in a trial vector **V** is to use the annihilation operation $(A - \lambda_1 I)$,

$$\mathbf{V}^1 = (A - \tilde{\lambda}_1 I)\mathbf{V}. \qquad (24)$$

Orthogonalizing in this way has the advantage of removing explicit error contributions due to errors in the eigenvector. Only the error in the eigenvalue estimate contributes to the growth of the unwanted component in the power method. This approach can be incorporated into the matrix times matrix approach by forming the product $A^m(A - \lambda_1 I)^k$, where we have multiplied the starting vector with A a total of m times and reorthogonalized k times.

Under the conditions of low precision the best procedure may be to reorthogonalize at each step. Then $k = m$, and the method is equivalent to finding the principal eigenvector of the matrix $A = A(A - \tilde{\lambda}_1 I)$. Error analysis shows that the λ_1 component contaminates the λ_2 as

$$\left(\frac{\lambda_1}{\lambda_2}\right)^m \left(\frac{\delta}{\lambda_1 - \lambda_2 + \delta}\right)^k, \qquad (25)$$

where δ is the error in our estimate of λ_1, i.e., $\delta = \lambda_1 - \tilde{\lambda}_1$. This procedure is safe, and the power method can be made to converge to each eigenvector in turn. The accuracy is limited by the accuracy of the previously estimated eigenvalues. The errors in estimates of eigenvalues must remain small compared with all the eigenvalues sought and to differences between eigenvalues sought. For the later eigenvectors the method becomes cumbersome, but as long as the magnitude of the eigenvalue sought is larger than the largest error in a previously estimated eigenvalue the method will converge.

A little known method for finding all the eigenvalues and eigenvectors involves double shifting.[12,13] It has the advantage that one starts fresh at each time, and thus no accumulation of errors results. It also is no more cumbersome as more eigenvalues are found. At no stage is the knowledge of eigenvalues to high precision required. It is based on forming a family of matrices

$$Q(\mu,B) = (A - \mu I)^2 - B^2 I \qquad (26)$$

for use with the power method. The eigenvectors of A are eigenvectors of Q. Q has eigenvalues

$$q_i = b_i^2 - B^2, \qquad (27)$$

where

$$b_i = \lambda_i - \mu \qquad (28)$$

The strategy is as follows. The b_i^2 are all positive. The smallest one is the one for which μ is closest to the eigenvalue λ_i. B^2 is chosen so that the most negati q_i, the one associated with the smallest b_i^2, is the dominant root. A sufficient condition is to choose B so that all the q_i are negative. Then the q_i associated with the smallest b_i^2 will be the most negative and hence the dominant root. The approach is to apply the matrix squaring method to the family of matrices $Q(\mu,B)$ until all the eigenvectors and eigenvalues of A are obtained. If the power method is first applied to A to obtain λ_1 and $\mathbf{e}_1$, a safe value of B is any number larger than $B > |\lambda_1| + |\mu|$. Here μ can be our best guess as to the next eigenvalue of interest. The convergence of the method to a solution depends on the two eigenvalues of A which are closest to μ. If λ_i is closest and λ_j is next closest, that is, if

$$(\lambda_i - \mu) < (\lambda_j - \mu) < (\lambda_k - \mu) \text{ for all } \begin{matrix} k \neq i \\ k \neq j, \end{matrix} \tag{29}$$

Q^m converges to $q_i^m \mathbf{e}_i \mathbf{e}_i^T$ as

$$(q_j/q_i)^m = \left[\frac{(\lambda_j - \mu)^2 - B^2}{(\lambda_i - \mu)^2 - B^2}\right]^m . \tag{30}$$

It is clear that only good choices for μ and B are required: no precise values are needed. However, the rate of convergence can be slowed by excessively large choices of B or a choice of μ for which $(\lambda_j - \mu) \approx (\lambda_i - \mu)$. The method is no more cumbersome for small roots than for large roots. The rate of convergence will be slower, however, if there are several small roots which are close together. Under those conditions q_j/q_i will be close to unity for any choice of μ. Precision will limit the dynamic range of eigenvalues that can be found. The magnitude of B must be such that q_j/q_i is less than one for convergence. Both deflation by orthogonalization and deflation by double shifting are attractive approaches for obtaining subsequent eigenvectors and eigenvalues of a matrix. Both are easily incorporated into the matrix squaring method.

IV. Role of Precision in Error Analysis

Important considerations in the application of the power method are the limits placed on the method by the precision of the computer.

These limits are based on the precision to which we can obtain the eigenvalue of largest magnitude. Deflation techniques based on orthogonality will not find eigenvectors for eigenvalues which are smaller than the er. ɹr in any proceeding eigenvalue. Assuming that errors arise only because of precision, the largest error will be associated with the principal eigenvalue. For example, if the precision is such that only s decimal figures are significant, the error associated with λ_1 is approximately $\lambda_1 \times 10^{-s}$.

Therefore, the smallest eigenvalue of A that can be found λ_{sm} satisfies

$$\frac{|\lambda_{sm}|}{|\lambda_1|} \geq 10^{-s} \tag{31}$$

This can be shown directly by substituting $\delta = \lambda_1 \times 10^{-s}$ into the convergence factor of Eq. (27), which must be less than unity. It is also not possible to distinguish between true degeneracies and near degeneracies if two or more eigenvalues differ by less than the error in the principal eigenvalue.

While the double-shift method does not require accurate values of previously obtained eigenvalues, there are direct effects of precision on the ability of the approach to resolve near degeneracies. If there are significant figures, the convergence factors must be $<1-10^{-s}$.

For the double-shift method

$$1 - 10^{-s} > \frac{(\lambda_j - \mu)^2 - B^2}{(\lambda_i - \mu)^2 - B^2} . \tag{32}$$

To insure that the most negative eigenvalue is the most dominant, B must have the same magnitude as λ_1. The best choice of μ is λ_i, and, therefore, the best possible convergence factor for the double-shift method is

$$1 - 10^{-s} > 1 - \frac{(\lambda_j - \lambda_i)^2}{\lambda_1^2} , \tag{33}$$

where we have substituted $\mu = \lambda_i$, $\lambda_1 = B$, and rearranged.

In this method eigenvalue pairs whose square difference satisfy $(\lambda_i - \lambda_j)^2 < \lambda_1^2 \times 10^{-s}$ will appear to be degenerate.

Another practical consideration is the power to which a matrix should be raised to obtain an eigenvalue estimate that is consistent with the number of significant figures of precision. An upper bound on the power to which a matrix can be raised to obtain meaningful results can be found by considering the bounds on the eigenvalue obtained from the trace. The error is given by

$$\delta = \left[\frac{1 - (1/N)^{1/P}}{2}\right] (Tr\mathrm{A}^P)^{1/P} \approx (Tr\mathrm{A}^P)^{1/P} 10^{-s}. \tag{34}$$

Dividing Eq. (34) by $(Tr\mathrm{A}^P)^{1/P}$ and solving for P, using the approximation $\ln(1 + x) \approx x$ for small x, yield $P \approx (10^s \ln N)/2$, where N is the dimension of the matrix. This assumes s is the number of significant decimal figures. P represents an upper bound to the power to which the matrix should be raised. For $s = 2$ and $N = 50$, we have $P \approx 185$.

V. Renormalization

The renormalization problem may become very important. The i,j term of A^2 is

$$(a^2)_{ij} = \sum_{k=1}^{N} a_{ik} a_{kj}. \tag{35}$$

A very conservative approach is to note that the maximum possible $(a^2)_{ij}$ is N times the square of the maximum a_{ij}. The trouble is that this approach is so conservative that it is likely to make very poor use of the available dynamic range of the optical processor and erode the accuracy of results in a system which already has limited accuracy. By doing each iteration twice (doubling an extremely short processing time), we can do much better. We use the ultraconservative but simple approach just described to normalize the inputs to estimate the maximum component of A^2 from A. With the estimated A^2 we do far less conservative renormalization and thus preserve accuracy

Thus we must search both the accurately calculated A and the crudely calculated A^2 for their maximum components. Remembering that in optical processors we work only with non-negative components which we can call $\delta_{i,j}$, we seek a parallel way to search for max-$(\delta_{i,j})$. The search need not occur on all N^2 components in parallel if (as often happens) the processor does not produce them that way. In a systolic processor, for example, as many as N components are available at any instant. We can find the maximum among them, compare with the prior maximum, and pass the larger value. In this way we can minimize memory requirements while achieving enough parallelism to avoid slowing down the process substantially.

A parallel search can be made by subtracting in parallel from all available component signals (δ_{ij}) a ramped signal

$$S(t) = S_0 t/\tau, \qquad (36)$$

where S_0 is the maximum allowable signal (a physical constraint) and τ is a preselected time constant. We then detect

$$d_{ij}(t) = \delta_{ij} - S(t) \qquad (37)$$

in parallel for all i,j. Each time a d_{ij} goes to zero its detector sends a unit signal to a counter. When the total count reaches $2N^2$, we note the time t_0. Then

$$\max(\delta_{ij}) = S_0 t_0/\tau. \qquad (38)$$

VI. Applications

Applications of eigenanalysis to direction finding, bandwidth compression (Karhunen-Loueve), pattern recognition, etc. are familiar. Here we want to point out that some nonobvious applications may prove quite useful as well.

Eigenvalue determination is one approach for finding roots of a polynomial:

$$P(x) = a_0 X^n = a_1 X^{n-1} + \quad + a_n = 0. \qquad (41)$$

It is convenient to write

$$P(X) = a_0(X^n + b_1 X^{n-1} + \quad + b_n), \qquad (42)$$

where, of course,

$$b_k = a_k/a_0. \qquad (43)$$

We can then write a matrix

$$C = \begin{bmatrix} 0 & 1 & 0 & \ldots 0 \\ 0 & 0 & 1 & 0 \\ & & . & \\ 0 & 0 & 0 & 1 \\ -b_n & -b_{n-1} & -b_{n-2} & -b_1 \end{bmatrix} \qquad (44)$$

so that the eigenvalues λ of C are the roots of $P(X)$. The eigenvalues must satisfy

$$\det(C - \lambda I) = 0, \qquad (45)$$

but

$$\det(C - \lambda I) = (-1)^n P(\lambda)/a_0 \qquad (46)$$

The form of C is easiest to see for a low-order polynomial. Thus for $n = 4$,

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -b_4 & -b_3 & -b_2 & -b_1 \end{bmatrix}. \qquad (47)$$

In this case

$$\det(C - \lambda I) = \det \begin{bmatrix} X & 1 & 0 & 0 \\ 0 & \lambda & 1 & 0 \\ 0 & 0 & \lambda & 1 \\ -b_4 & -b_3 & -b_2 & -b_1 \end{bmatrix} = P(\lambda)/a_0. \qquad (48)$$

By our method we can easily find the root nearest a chosen value. Likewise multiple roots are readily detected.

Of course, if we can solve $P(X) = 0$, we can solve

$$P_1(X) = P_2(X), \qquad (49)$$

since

$$P_N(X) = P_1(X) - P_2(X) \qquad (50)$$

must have a zero when $P_1(X) = P_2(X)$. More generally, to solve

$$P_1(X) = P_2(X) = \ldots = P_N(X) = 0, \qquad (51)$$

we form the new polynomial

$$Q(X) = \sum_{i=1}^{N} [P_i(X)]^2 \qquad (52)$$

Clearly $Q(X)$ can be zero only if each of the $P_i(X)$ is zero.

VII. Summary

Prior proposals for optical computation of eigenpairs have encountered major problems relating to slow convergence of the iterative algorithm, lower accuracy on less dominant eigenpairs, and low accuracy from poor renormalization. This paper discusses some methods reducing these problems considerably, although it cannot be said to have finally and definitively solved them. The convergence speed is increased dramatically by the matrix squaring approach. The deflation accuracy may be improved by the matrix reformulation methods discussed. Excellent use of the available dynamic range can be assured for a factor of 2 decrease in overall speed using the technique described.

Problems related to degeneracies and numerical accuracy have also been attacked here. In particular we have been able to show that matrix squaring handles degeneracies easily and automatically and that tight simple error bounds can be determined.

What we have dealt with are algorithm related problems. Implementation problems are also numerous, but they are beyond the scope of this paper. There are also rather fundamental problems relating to the numerical accuracy of the final answers. We believe that these problems can be solved so as to make optical eigenfunction solution practical and attractive.

This work was performed under U.S. Air Force contract F19628-82-C-0068.

References

1. R. A. Heinz, J. O. Artman, and S. H. Lee, Appl. Opt. **9**, 2161 (1970).
2. M. A. Monahan, R. P. Bocker, K. Bromley and A. Louie, "Incoherent Electro-Optical Processing with CCD's," at International Optical Computing Conference Digest (IEEE Catalog 75 CH0941-5C) (Apr. 1975).
3. J. W. Goodman, A. R. Dias, and L. M. Woody, Opt. Lett. **2**, 1 (1978).
4. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz, Opt. Commun. **40**, 86 (1981).
5. H. J. Caulfield, D. Dvore, J. Goodman, and W. T Rhodes, Appl. Opt. **20**, 2283 (1981).
6. B. V. K. Vijaya Kumar and D. Casasent, Appl. Opt. **20**, 3707 (1981).
7. D. Psaltis, D. Casasent, and M. Carlotto, Opt. Lett. **4**, 348 (1979).
8. J. W. Goodman and M. S. Song, Appl. Opt. **21**, 502 (1982)
9. J. H. Wilkinson *The Algebraic Eigenvalue Problem* (Clarendon, Oxford, 1965).
10. H. Hotelling, J. Educ. Psychol **24**, 417, 498 (1933).
11. A. Ralston, *First Course in Numerical Analysis* (McGraw-Hill, New York, 1965), p. 486.
12. A. Gamba, Am. J. Phys. **49**, 187 (1981).
13. A. S. Householder, *Principles of Numerical Analysis* (McGraw-Hill, New York, 1953), p. 156.

APPENDIX F

GENERALIZATION OF THE EIGEN PROBLEM TO SINGULAR VALUE DECOMPOSITION BY OPTICAL MEANS

(Accepted For Publication In Applied Optics)

OPTICAL SINGULAR VALUE DECOMPOSITION FOR THE $A\vec{x} = \vec{b}$ PROBLEM

John Gruninger and H.J. Caulfield

Center for Optical & Photographic Sciences
Aerodyne Research, Inc.
45 Manning Road, Billerica, MA 01821

Abstract

Optical approaches to solving the $A\vec{x} = \vec{b}$ problem have suffered from four difficulties: (1) an inability to handle the problem for nonsquare A, (2) the necessity of insuring convergence for nonsingular A, (3) the inability to handle a singular A, and (4) inaccuracies due to an ill conditioned A. We show that these problems can all be solved or mitigated by singular value decomposition (SVD). An accurate approach to optical SVD is shown.

Introduction

Optical computing has drawn much attention in terms of both architecture[1-5] and algorithms[6-9] in the last few years. This paper aims at a thorough discussion of optical singular value decomposition (SVD): a topic recently treated by Kumar.[10] We will show why SVD is not only particularly well suited for optical computation but also particularly useful as part of optical computing's repetoire. Our emphasis will be on a particular type of problem represented as

$$A\vec{x} = \vec{b},$$

where A is a known m x n (m rows, n columns) matrix, $\vec{x}$ as an n dimensional unknown vector, and $\vec{b}$ is an m dimensional known vector. Our task is to find $\vec{x}$. When m=n, this the familiar case of n linear equations with n unknowns. It is solvable in principle if A is nonsingular. When m > n, this is the equally familiar problem of optimum curve fitting (usually using a least squares criterion).SVD has numerious other applications in image processing, antenna field calculation and pattern recognition which have been discussed elsewhere.

The $A\vec{x} = \vec{b}$ problem is arguably the most important and most commmon problem in computing. A large fraction of all of the computer time in the world is used in solving large linear programming problems. Linear programming solutions occur in two parts: the solution of large $A\vec{x} = \vec{b}$ problems is the most time consuming part, the other part is some bookkeeping called the simplex algorithm. The authors have heard expert mathematicians argue that the least squares problem is the most important problem in mathematics in terms of its impact on the world. Such a claim could be supported by applications ranging from statistics to phased array antennas. Control theorists and many others are fond of posing sets of differential in equations in the $A\vec{x} = \vec{b}$ format. The number of applications there is quite large.

Prior optical approaches to solving $A\vec{x} = \vec{b}$ run into a variety of problems. First, they are limited to the m=n case and thus omit many important cases. Second, each of the iterative methods has a convergence condition which can be guaranteed only by going through a precalculation which either confirms the convergence or transforms the problem to assure convergence. Third, the result of our calculations may be in very serious error if A is ill conditioned. This problem is compounded by the inaccuracy of optical computers relative to their electronic counterparts. All of these problems combine to make optical solution of the $A\vec{x} = \vec{b}$ problem less make optical soution of the $Ax = b$ problem less attractive than electronic solution for many problems even though optics has well known advantages in m and n size, speed, computer size, and power consumption.

In the balance of this paper we will argue that SVD alleviates all of those problems for $A\vec{x} = \vec{b}$ solution. Specifically: (1) it allows $m \neq n$ and gives the least squares solution in that case; (2) it can be made to converge even when A is singular; and (3) it can offer us a way to find good but inexact solutions even when A is ill conditioned.

We consider here solving the least squares problem for non symmetric non square matrices A. In particular we will be concerned about matrices which may be less than full rank, and which may be ill conditioned. That is, if the dimensions of A are $m \times n$ with $m > n$, then we include for consideration matrices which have rank $k < n$ and have pseudo rank $\ell < k$. A natural approach to such problems is through the singular value decomposition of A. A can be expressed as

$$A = W \Lambda V^T$$

where W is a m x m orthogonal matrix and V is an n x n orthogonal matrix. Λ is a m x n matrix whose only non zero elements are the "diagonals", λ_{ii}, for $i = 1,k$ where k is the rank of A. The singular values λ_i are assumed to be in descending order $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_k$. We have dropped the second, redundant index. Performing the implied matrix multiplications yields

$$A = \sum_{i}^{k} \lambda_i \vec{w}_i \vec{v}_i^T \quad . \tag{2}$$

We use the lower case letters $\vec{w}$ and $\vec{v}$ to indicate column vectors of W and V respectively. The subscript i indicates the column number. If the matrix A has a pseudo rank of $\ell < k$, then the singular values $\lambda_{\ell+1}$ to λ_k will be very small. The Eckart Young Theorem[11] suggests that the last k-ℓ outer products can be deleted from the sum. That is A can be written as

$$A = A^\ell + \Delta A^\ell \tag{3}$$

where

$$A^\ell = \sum_{i=1}^{\ell} \lambda_i \vec{w}_i \vec{v}_i^T$$

and

$$\Delta A^{\ell} = \sum_{i=\ell+1}^{k} \lambda_i \vec{w}_i \vec{v}_i^T$$

Eckart and Young showed that A^{ℓ} is the best rank ℓ approximation to A in the Frobenius norm. The norm of the error term

$$\|\Delta A\|_F^2 = \sum_{i=\ell+1}^{k} \lambda_i^2$$

is given by the square root of the sum of the squares of the neglected singular values. If the elements of the matrix A were obtained experimentally or if they are stored in a computer with low precision, such that the stored version differs from the "true" version by δA then carrying more singular values than that number, ℓ, for which $\|\Delta A^{\ell}\| = \|\delta A\|$ is useless.

For numerical stability we replace A with A^{ℓ}. In matrix form we write

$$A^{\ell} = W^{\ell} \Lambda^{\ell} (V^{\ell})^T \tag{4}$$

where W^{ℓ} is the m x ℓ matrix whose columns are the first ℓ columns of W, V^{ℓ} is the n x ℓ matrix whose columns are the first ℓ columns of V and Λ^{ℓ} is the ℓ x ℓ matrix of singular values $\lambda_i, i=1,\ell$.

The least squares problem $A\vec{x} = \vec{b}$ is transformed into a new one by multiplying on the left by W^T and using the fact that $W^TW=1$.

$$W^T A\vec{x} = \Lambda V^T\vec{x} = W^T\vec{b}: \tag{5}$$

Defining $\vec{y} = V^T\vec{x}$ and $\vec{g} = W^T\vec{b}$ the least squares problem is

$$\Lambda\vec{y} = \vec{g} \tag{6}$$

The components of $\vec{y}$ are given by

$$\begin{aligned} y_i &= \frac{g_i}{\lambda_i} \qquad i = 1,k \\ y_i &= 0 \qquad i = k+1,n \quad . \end{aligned} \tag{7}$$

The solution vector $\vec{x}$ is obtained from $V\vec{y}$. The norm of $\vec{x}$ is a measure of the stability of the least squares solution. It is obtained from the square root of

$$\|\vec{x}\|^2 = \|\vec{y}\|^2 = \sum_{i}^{k} \left(\frac{g_i}{\lambda_i}\right)^2 \tag{8}$$

The square of the norm of the residuals is given by

$$R^2 = \|A\vec{x} - \vec{b}\|^2 = \sum_{i=k+1}^{m} g_i^2 \tag{9}$$

In the event that A is ill conditioned some of the columns of A are nearly linearly dependent, and some of the singular values will be small. Contributions from the small singular values lead to erratic changes in $\vec{x}$ and in a dramatic increase in its norm. Defining a pseudo rank of ℓ less than k we obtain solutions $\vec{x}^{\ell}$ for the least squares problem $A^{\ell}\ \vec{x}^{\ell} = b$ defining $\vec{g}^{\ell}$ as $(\vec{W}^{\ell})^T$ b we obtain

$$\begin{aligned} y_i &= \frac{g_i^{\ell}}{\lambda_i} \qquad & i = 1,\ell \\ y_i &= 0 \qquad & i = \ell+1,n \end{aligned} \tag{10}$$

The solution vector $\vec{x}^{\ell}$ is obtained from $V^{\ell}\vec{y}$. The square of the norm of $\vec{x}^{\ell}$is

$$\|\vec{x}^{\ell}\|^2 = \sum_{i=1}^{\ell} \left(\frac{g_i^{\ell}}{\lambda_i}\right)^2 \tag{11}$$

and the square of the norm of the residual is

$$\|R^{\ell}\|^2 = \|A^{\ell}\vec{x}^{\ell} - \vec{b}\|^2$$

the pseudo rank ℓ is chosen so that the norms of the solution vector $\|\vec{x}^{\ell}\|$, the residual $\|R^{\ell}\|$ and the error matrix $\|\Delta A^{\ell}\|$ are exceptably small. More details of this aspect of least squares problems can be found in Lawson and Hanson.[12]

When the pseudo rank ℓ is much less than n, a method for finding only the first ℓ singular values and the residual matrices W^{ℓ} and V^{ℓ} is desired. We propose obtaining this partial singular value decomposition of A by use of a power method. An iterative scheme can be based on the following pair of equations.

$$A \vec{v}_i = \lambda_i \vec{w}_i \tag{13}$$

and

$$A^T \vec{w}_i = \lambda_i \vec{v}_i \tag{14}$$

which are obtained from Eq. (2). Starting with an initial guess at $\vec{v}_1$ namely $\vec{v}_1^{\,o}$ and an initial $\vec{w}_1^{\,o}$, and an estimate of the singular value, λ_1, can be obtained from Eq. (13). $\vec{w}_1^{\,o}$ in turn can be used in Eq. (14) to find an improved $\vec{v}_1$. We use superscripts to indicate iteration numbers.

After J iterations we have

$$\lambda_1 \vec{v}_1^{\,J} = A^T \vec{w}_1^{\,J-1} \tag{15}$$

$$\lambda_1 \vec{w}_1^J = A \vec{v}_1^J \qquad . \qquad (16)$$

The procedure can be stopped when $\| \vec{v}_1^J - \vec{v}^{J-1} \|$ and $\| \vec{w}_1^J - \vec{w}_1^{J-1} \|$ are sufficiently small. This procedure will yield the dominant singular value λ_1 and singular vectors $\vec{w}_1$ and $\vec{v}_1$. Applying the procedure to the deflated matrix

$$\tilde{A} = A - \lambda_1 \vec{w}_1 \vec{v}_1^T \qquad (17)$$

will yield λ_2, w_2 and v_2 and so on. This approach has been recently proposed by Shlien[13] and by Kumar.[10] An alternate approach is suggested here. If Eq. (15) and (16) are substituted into one another one obtains

$$\lambda^2 \vec{v}^J = (A^T A)\vec{v}^{J-1} = S \vec{v}^{J-1} \qquad (18)$$

and

$$\lambda^2 \vec{w}^J = (AA^T) \vec{w}^{J-1} = M \vec{w}^{J-1} \qquad (19)$$

This approach is equivalent to finding the principal eigenvectors of the n x n and m x m positive semidefinite matrices $S = A^T A$ and $M = AA^T$, respectively. The right singular vectors, $\vec{v}_i$ of A are eigenvectors of S while the left singular vectors, $\vec{w}_i$ are eigenvectors of M. The non-zero eigenvalues of S and M are equal to the square of the corresponding singular value, λ_i^2.

It is not necessary to find the eigenvectors of both S and M. A simple approach is to find the eigenvectors to the matrix of smallest dimension, namely S. Several approaches to the use of the power method for eigenvectors of symmetric matrices have appeared in the literature.[6,14,15] Once the first eigenvector $\vec{v}_1$ of S is found, $\vec{w}_1$ can be obtained from Eq. (13). The matrix A can be deflated by the combined use of Eq. (17) and Eq. (13).

$$\tilde{A} = A - A\,\vec{v}_1\,\vec{v}_1^{\,T} \tag{20}$$

The positive semi definite matrix $\tilde{A}^T\tilde{A}$ can be formed and the procedure repeated to find $\vec{v}_2$, λ_2 and $\vec{w}_2$ and so on.

One concern in using a power method for singular value decomposition is the loss in dynamic range that occurs when A^TA or AA^T is formed. As Eqs. (18) and (19) we derived from (15) and (16), the formation of these square matrices results from any formulation of the power method. The best one can do is to initially equilibriate the columns of A and normalize the approximate (singular vectors) eigenvectors at each iteration. Equilibration is the process of finding that diagonal matrix D which will scale the columns of A so that they have unit length. We let

$$A \rightarrow AD \qquad \vec{x} \rightarrow D^{-1}\,\vec{x}$$

and solve

$$(AD)\ (D^{-1}\ \vec{x}) = \vec{b}$$

We assume that A was previously equilibrated in the above discussion. The key to numerical stability in the power method is not in the formation of the square matrices S and M. The key is that deflation be performed on A in order to find additional singular values. One should not attempt to deflate S or M. The success of our proposed method as well that the methods of Shlien[13] and Kumar[10] is based on this deflation.

The difficulties that we address in terms of dynamic range can be illustrated by the following example matrix.

$$A = \begin{pmatrix} 1 & 1 \\ 0 & \varepsilon \\ \varepsilon & 0 \end{pmatrix} \tag{21}$$

where ε is within the dynamic range of the computer and ε^2 is not. The norm of its columns is $(1 + \varepsilon^2)^{1/2} = 1$, so the matrix is equilibrated. The matrix A has rank 2 but is ill conditioned. The matrices S and M that will be

$$S = \begin{pmatrix} 1 + \varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \tag{22}$$

$$M = \begin{pmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon^2 & 0 \\ \varepsilon & 0 & \varepsilon^2 \end{pmatrix} = \begin{pmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & 0 & 0 \\ \varepsilon & 0 & 0 \end{pmatrix} \qquad (23)$$

generated in the computer will be the rank 1 matrices on the right in Eq. (22) and (23) respectively. The key point here is information about $\vec{v}_1$ and $\vec{w}_1$ are still retained in S and M while information about $\vec{v}_2$ and $\vec{w}_2$ are lost. Numerical instability will occur when attempting to deflate S or M to find subsequent eigenvectors. Numerical stability is maintained only if A is deflated through Eq. (20). That is the power method will find

$$\vec{v}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

and deflation of A will yield

$$\bar{A} = \frac{\varepsilon}{2} \begin{pmatrix} 0 & 0 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}$$

$\vec{v}_2$ $(1/\sqrt{2})$ $[1, -1]^T$ will be found by applying the power method to $\bar{A}^T\bar{A}$. The key to the successful use of the power method for singular values is the use of the deflation of A. Attempts to deflate S or M will yield matrices which contain only noise. The principal eigenvectors to the matrices S and M can be obtained from the power method however. The use of a power method requires the formation of at least one of these matrices.

We summarize the proposed procedure for singular values decomposition as follows

(i) Equilibrate A, call it $\bar{A}_1$.

(ii) Form $S_i = \bar{A}_i^T \bar{A}_i$, scale if necessary.

(iii) Find the principal eigenvector $\vec{v}_i$ of S_i.

(iv) Calculate A $\vec{v}_i$.

(v) Find λ_i by normalizing A $\vec{v}_i$

if $\lambda_i = 0$ stop.

(vi) W_i is the resulting normalized A $\vec{v}_i$.

(vii) Form $\bar{A}_{i+1} = \bar{A}_i - A \vec{v}_i \vec{v}_i^T$,

scale if necessary.

(viii) Go to ii.

This procedure will terminate after obtaining the ℓth singular value λ_ℓ and singular vector $\vec{v}_\ell$. The least square problem is then solved using Eqs. (10), (11) and (12).

References

1. J.W. Goodman, A.R. Dias, and L.M. Woody, Optics Lett. 2, 1 (1978).

2. H.J. Caulfield, W.T. Rhodes, M.J. Foster, and S. Hovitz, Opt. Common. 40, 86 (1981).

3. D. Psaltis, D. Casasent, and M. Carlotto, Optics Lett. 4, 348 (1979).

4. D. Casasent, J. Jackson, and C. Neumann, Appl. Opt. 22, 115 (1983).

5. R.P. Bocker, H.J. Caulfield, and K. Bromley, Appl. Opt. 22, (1983).

6. H.J. Caulfield, D. Dvore, J.W. Goodman, and W.H. Rhodes, Appl. Optics 20, 2263 (1980).

7. M. Carlotto and D. Casasent, Appl. Opt. 21, 147 (1982).

8. J.W. Goodman and M.S. Song, Appl. Optics 21, 502 (1982).

9. W.K. Cheng and H.J. Caulfield, Opt. Common. 43, 251 (1982).

10. B.V.K.V. Kumar, Appl. Opt. 23, (1983).

11. C. Eckart and G. Young, Psychrometrika 1, 211 (1936).

12. Charles L. Lawson and Richard J. Hanson, "Soluins Least Squares Problems," Prentice Hall, New Jersey (1974).

13. Seymor Shlien, IEEE Trans. PAMI 4, 671 (1982).

14. B.V.K.V. Kumar and D. Casasent, Applied Optics 21, 3707 (1982).

15. J. Gruninger and H.J. Caulfield, Applied Optics 23 (1983).

Acknowledgments

This work was supported under Contract No. F19628-82-C-0068 from Rome Air Development Center, Hanscom AFB, MA 01731.

APPENDIX G

APPROXIMATE SINGULAR VALUE DECOMPOSITION

(This work was presented at the 1983 OSA meeting as noted using the attached viewgraphs. The write up for publication is still being pursued. We will submit the paper for publication in Applied Optics).

MONDAY, OCTOBER 17, 1983 REGENCY A, 8:30 A.M.

ALEXANDER A. SAWCHUK, *Presider*

Image Processing I

Contributed Papers

MF1. Two-Dimensional Optical Fourier Transformation by Time-Integration Methods. WILLIAM T. RHODES, KEITH D. RUEHLE, AND ROBERT E. STROUD, *School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.*—Time-integration optical processing methods in the past have been applied to the spectrum analysis of one-dimensional signal waveforms. However, they can also be used to evaluate the spatial-frequency content of two-dimensional images. The key idea is a mapping of object points into associated two-dimensional fringe patterns. A system using a modulated laser and two orthogonally scanned mirrors has been used to produce a time-integration (complex) spatial-frequency spectrum of a test pattern. Helpful analogies between system operation and incoherent holography can be drawn. (13 min.)

MF2. Cosinusoidal Transforms in White Light.* SHEN-GE WANG AND NICHOLAS GEORGE, *The Institute of Optics, University of Rochester, Rochester, New York 14627.*—Theory and techniques of white-light interferometry are being studied in order to develop new methods of optical pattern recognition. In white-light illumination or with rough input objects, the conventional diffraction-pattern sampling system is not applicable without the use of an incoherent-to-coherent converter. As an alternative approach, we report on a diffraction-limited transform system that can be used with spectrally broad, spatially incoherent illumination. The transform obtained is a spatial, two-dimensional cosine transform of the input plus a bias term. The optical system consists of a double-imaging interferometer with a beam splitter and two right-angle prisms followed by an achromatic optical-transform lens pair. The design of the interferometer and the achromatic optical transform are detailed and contrasted with earlier versions. Excellent diffraction-limited performance is obtained for the entire visible spectrum in an all-glass system. A photodiode array is placed in the optical transform plane in order to interface the system to a digital computer. The bias term of the cosinusoidal transform is subtracted electronically. Noise limitations are described. Cosine transforms have been obtained for a variety of rough objects, and recent experiments are described. (13 min.)

* This research was supported by the U.S. Air Force Office of Scientific Research and the U.S. Army Research Office.

MF3. Matrix and Image Decomposition by Projection Encoding. JOHN GR[illegible] SANGER AND H. J. CAULFIELD, *Aerodyne Research Inc., 45 Manning Road, Billerica, Massachusetts 01821.*—The image decomposition problem is related to projection encoding. In projection encoding a two-dimensional array of data is collapsed or projected onto two or more one-dimensional arrays. The primary operation of reconstruction is the backprojection of the one-dimensional arrays into two-dimensional space. If the matrix is column collapsed and row collapsed, the backprojection corresponds to the vector outer product of the one-dimensional arrays. This backprojection yields a rank-one approximation to the original matrix. Repeating this process on successive residuals leads to a decomposition of the image into a sum of rank-one matrices. The projection method can be considered as an approximation to singular value decomposition. Lower bounds to singular values are obtained. We show that the method is computationally simple and that its extension to three-dimensional images is straightforward. (13 min.)

MF4. Image Processing in Signal-Dependent Noise Using Local Statistics and the Generalized Homomorphic Transformation. H. H. ARSENAULT AND M. LEVESQUE, *Department of Physics, LROL, Université Laval, Ste-Foy, Quebec G1K 7P4, Canada.*—In an image with signal-dependent noise, a general point transformation can transform the image into a space where the noise is independent of the signal. Local statistics methods of image processing appropriate to additive noise then may be applied to the image before transforming back to the original space. Experimental results for film grain noise and for multiplicative noise are shown and compared with results without the homomorphic transformation. This method, suitably implemented, is about 10 times faster than the global method using the fast-Fourier transform. (13 min.)

MF5. Minimum Bias Pupil Design for Bipolar Incoherent Spatial Filtering. JOSEPH N. MAIT AND W. T. RHODES, *School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.*—Incoherent spatial filtering offers a number of advantages over coherent spatial filtering, most notably its insensitivity to dust and system imperfections. However, incoherent systems are linear in intensity, not in complex-wave amplitude, and the spatial impulse response, or point-spread function, is nonnegative and real. The removal of bias by means, for example, of image subtraction, allows for the synthesis of bipolar point-spread functions and has been demonstrated with multichannel hybrid electro-optical systems.[1] Bias reduction prior to detection can increase contrast and decrease noise in the final processed image and may be achieved through pupil mask design. Bias reduction through pupil design may be presented as a problem in constrained minimization, thus allowing standard optimization techniques to be used to find minimum bias pupils. (13 min.)

[1] W. T. Rhodes, Opt. Eng. 19, 323–330 (1980).

APPROXIMATE SVD

JOHN H. GRUNINGER &

H. J. CAULFIELD

AERODYNE RESEARCH, INC.
45 MANNING ROAD
BILLERICA, MA. 01821

OCTOBER, 1983

SVD MOTIVATION

- CONDITIONING, SINGULARITY, AND ALL THAT
- BEATING THE SYSTEM
  - ECKHART-YOUNG
  - GRUNINGER-CAULFIELD

MOTIVATION FOR ASVD

- REAL SVD IS VERY HARD
- ONLY NEED ASVD ANYWAY

SVD/ASVD

- $A = W \Lambda V$ (SVD)
- $A \approx \overline{W} \; L \; \overline{V}$ (ASVD)
- FOR $A = A^T$

  $W = V^T$

ASVD (COLUMNS ONLY)

- WRITE $G = C^T A$

  $\nwarrow [c_1 | c_2 | \ldots | c_K]$ $K << R$

- $S = GG^T = C^T AA^T C$ $K \times K$

- $E = [E_1 | E_2 | \ldots | E_K]$ EIGENVECTORS OF S

  $\lambda_1, \lambda_2, \ldots, \lambda_K$ EIGENVECTORS OF S

  ↑

  EIGEN VALUES

  SO

  $$E^T S E = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_K \end{bmatrix} \equiv \underline{\underline{S}}$$

- CAN SHOW

  $$S_I \leq \lambda_I{}^2$$

  SINGULAR
  VALUE

ASVD II

- $\bar{W} = CE$
- JUSTIFICATION

$$\text{SVD} \begin{cases} A = W\Lambda V \\ AA^T = W\Lambda^2 W^T \\ W^T AA^T W = W^T W \Lambda^2 W^T W = \Lambda^2 \quad \text{DIAGONAL} \end{cases}$$

$$\text{ASVD} \begin{cases} \bar{W}^T AA^T \bar{W} \\ = E^T C^T \underbrace{AA^T} CE \\ = E^T SE \end{cases}$$

$$= \underline{\underline{S}} \quad \text{DIAGONAL}$$

REMAINING PROBLEM:

FIND C

- PHILOSOPHY -

- FIND AVERAGE COLUMN

  $\longrightarrow c_1$

- ELIMINATE FROM $A \rightarrow A_1$
- FIND LARGEST COLUMN OF A_1
- ELIMINATE FROM $A_1 \rightarrow A_2$
- CYCLE

PSEUDO MODIFIED GRAM-SCHMIDT

- STOP WHEN SOME CRITERION IS MET.

ASVD DETAILS

- "EQUILIBRATE"

  $A \rightarrow AD$ D DIAGONAL

  $A = [A_1 | A_2 | \ldots | A_R]$

  $|A_K|^2 = 1$

- FIND AVERAGE COLUMN
  - CALL IT C_1

- USE PSEUDO MODIFIED GRAM-SCHMIDT

  $$A_J^{(1)} = A_J - \underbrace{(C_1 A_J)}_{\text{OPTICAL STEP}} C_1 \longrightarrow A_1$$

- FIND LARGEST COLUMN IN $A_1 \longrightarrow C_2$
  - APPLY PMGS $\rightarrow A_2$

- CYCLE

ASVD, III

- $|C_K|^2$ IS A MEASURE OF THE ERROR IN PSEUDORANK K-1 APPROXIMATION.

- POSSIBLE STOPPING CRITERION:

$$|C_K|^2 < \epsilon$$

- FOR NORMALIZED R X R MATRIX

  TRACE = R, BUT

  TRACE = SUM OF SINGULAR VALUES SQUARED

  → ANOTHER CRITERION.

EXAMPLE

- $1832 \overset{47}{\begin{bmatrix} \quad \\ \quad \end{bmatrix}} = AD$ SUSPECTED OF BEING POORLY CONDITIONED.

- SVD OF A

$$\lambda_1^2 = 41.89$$
$$\lambda_2^2 = 4.03$$
$$\lambda_3^2 = 0.35$$
$$\lambda_4^2 = 0.32$$

$$\sum_{K=1}^{47} \lambda_K^{\cdot 2} = 47$$

COMPARISON

K	RMS ERROR USING PR = κ FOR RANDOM VECTOR		$\left\|C_{K+1}\right\|_{MAX}$	
1	0.33		0.56	
2	0.15		0.53	
3	0.12		0.35	
4	0.09	10% AFTER κ = 4	0.26	
5			0.20	
6			0.11	10% AFTER κ = 6

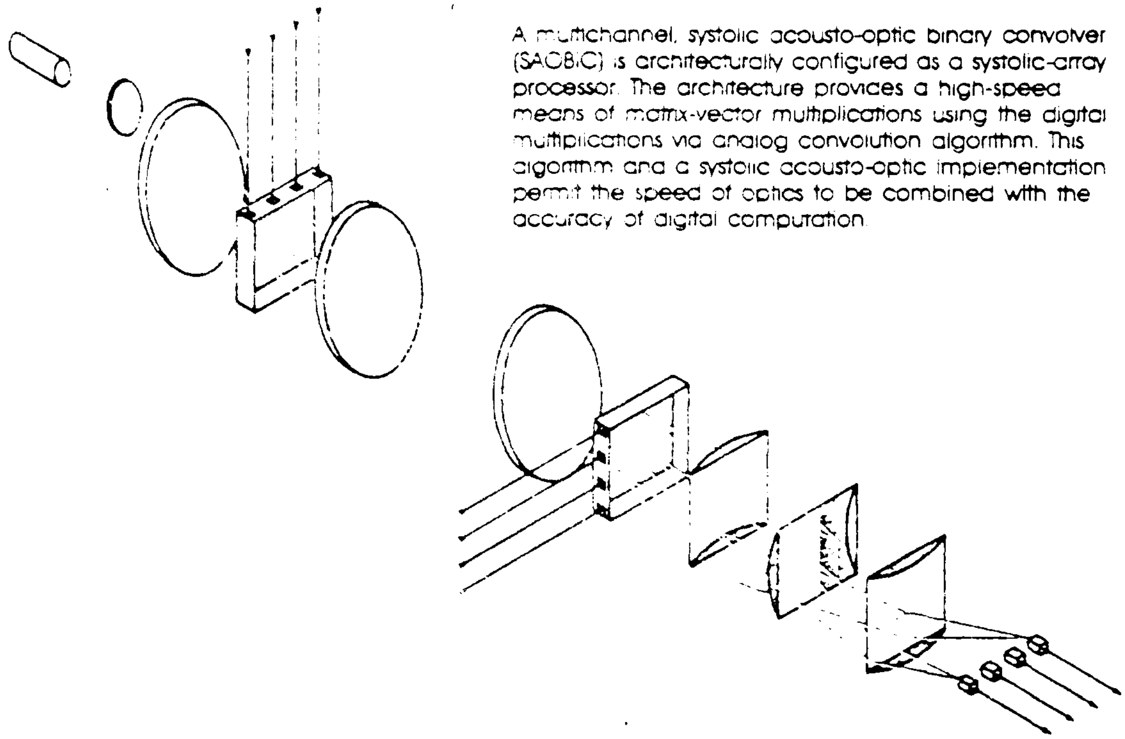
APPENDIX H

A BRIEF OVERVIEW OF THE FIELD OF OPTICAL DIGITAL PROCESSING SPAWNED BY THIS CONTRACT

OPTICAL COMPUTING: THE COMING REVOLUTION IN OPTICAL SIGNAL PROCESSING

Development is progressing toward a new generation of optical computational devices that may provide for ultra-high-speed matrix algebra and for the density of interconnections needed in optical supercomputers.

By H. John Caulfield, John A. Neff, and
William T. Rhodes



A multichannel, systolic acousto-optic binary convolver (SAOBiC) is architecturally configured as a systolic-array processor. The architecture provides a high-speed means of matrix-vector multiplications using the digital multiplications via analog convolution algorithm. This algorithm and a systolic acousto-optic implementation permit the speed of optics to be combined with the accuracy of digital computation.

Optical signal processing has its roots in the nineteenth-century work of Lord Rayleigh, Huygens, Abbe, Lippmann and others, and its greatest promise in the twenty-first century. Here in the late twentieth century a small number of optical processing systems (spectrum analyzers, synthetic-aperture radar processors, ambiguity function generators, etc.) have already supplanted their electronic counterparts, and others may succeed soon (pattern recognition, direction finding, etc.). The advantages of optics over electronics in these systems include some combination of lower cost, reduced size, lower power consumption, higher speed, and potentially enhanced reliability.

Although it is not yet realistic to plan for a general-purpose optical computer, it is possible to think seriously about many general optical-array processors, as suggested by Fig. 1, that can be used as adjuncts to digital computers for performing specific algebraic computations at very high speeds. Designs are currently under consideration for ultra-high-speed optical processors to evaluate polynomials, matrix-vector products, matrix-matrix products, and solutions of sets of linear equations.

This article reviews the developments of the last several decades that led to this position, describes briefly some important areas of current research and development, and lists several areas of expected major future development.

Philosophy and recent developments

Operations performed by optical systems are described by simple mathematics: convolution, multiplication, integration, etc. It requires only a minor change in outlook to convert from mathematics as a description of optics to mathematics as the goal of the optics. Such a viewpoint was taken by Cutrona at the University of Michigan as early as 1965 when he described the application of optical systems to the evaluation of general superposition integrals and to the multiplication of a vector by a matrix. Indeed, many of the early researchers in optical signal processing systems—Gabor, [illegible] Vander Lugt, S[illegible] Goodman—recognized the potential of optical systems for performing a variety of mathematical operations. These researchers, and many after them, concentrated [illegible] analog [illegible] eral [illegible] transformations [illegible] that [illegible] er or [illegible] to [illegible]

During the past several years attention has turned to a different application of optics to mathematical operations, in this case operations that are numerical, sometimes discrete, and often algebraic in nature. Indeed, the redirection of attention has been so vigorous that many view it as a small revolution in optics: optical signal processing is beginning to encompass what many feel is aptly described as optical *computing*, where the term is fully intended to imply close comparison with the operations performed by scientific digital computers. The optical-array processor, mentioned earlier, forms the basis for this revolution. (The term *optical computing* has been used occasionally for nearly two decades now in connection with analog optical processors, but a major fraction of the optical signal-processing community has never felt comfortable with it because of the implied comparison with general-purpose digital computers. That situation is poised for change.)

In retrospect, the beginning of modern optical-array processors was the invention of what is now often called the Stanford optical matrix-vector multiplier (OMVM). This device, illustrated in Fig. 2, has a capability of multiplying a 100-component vector by a 100 × 100 matrix in roughly 20 ns. Components of the input vector **x** are input via a linear array of LEDs or laser diodes. The light from each source is spread out horizontally by cylindrical lenses, optical fibers, or planar lightguides to illuminate a two-dimensional (2-D) mask that represents the matrix **A**. Light from the mask, which has been reduced in intensity by local variations in the mask transmittance function, is collected column by column and directed to discrete horizontally arrayed detectors. The outputs from these detectors represent the components of output vector **y**, where **y** is given by the matrix-vector product **y** = **Ax**:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & & & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

Since light intensity, which is always nonnegative, is used to represent the various mathematical quantities, special coding techniques must be employed if both positive and negative (or complex) numbers are to be accommodated.

As originally conceived, the Stanford OMVM suffers from several potentially serious limitations

- Accuracy is limited by the accuracy with which the source intensities can be controlled and the output intensities read;
- Dynamic range is source and/or detector limited;
- Rapid updating of the matrix A requires the use of a high-quality 2-D read-write transparency—a spatial light modulator (SLM)—whose optical transmittance pattern can be changed rapidly. Unfortunately, such a device does not yet exist with all the desired characteristics, although candidate devices are being improved rapidly.

Despite these drawbacks, the Stanford development brought about an important swing within the optical signal-processing community from a preoccupation with coherent, Fourier-transform-based processors to incoherent, geometrical optics-based processors. It is interesting to note that this change in direction was initiated by Prof. Joseph W. Goodman, whose book on Fourier optics [illegible] coherent optics so firmly in many minds.

The speed [illegible] of the optical hardware [illegible] researchers with a perplexing [illegible] processor could operate at speeds [illegible] the ability to input and output [illegible] which often required digitization for [illegible] with surrounding electronic systems. One approach to circumventing this problem is to use the OMVM for *iterative* algorithms, where the processor output is directed in analog form back to the input. A variety of iterative processing [illegible] were developed by Casasent, [illegible] and Rhodes. One example is [illegible] inversion of matrix equation [illegible] for vector x, given vector [illegible] iterative algorithm [illegible] counterpart.

[illegible]

[illegible]

[illegible] the [illegible] particu- [illegible] at [illegible] Center and [illegible] Research Laboratory [illegible] convinced the [illegible] three [illegible]

[illegible] racy in the same way electronics does—by going digital. This led to the first suggestion by Psaltis of a means to achieve digital optics. Third, the newly emerging field of systolic-array processing should be amenable to optical implementation. This latter suggestion led to work, primarily by Caulfield and Rhodes, on an optical systolic-array processor, described below. Soon, both published and unpublished work by Tamura, Casasent, and others advanced this area greatly.

Systolic-array processing, developed principally by H. T. Kung at Carnegie-Mellon University and S. Y. Kung at the University of Southern California, is an algorithmic and architectural approach to overcoming limitations of VLSI electronics in implementing high-speed signal-processing operations. Systolic processors are characterized by regular arrays of identical (or nearly identical) processing cells (facilitating design and

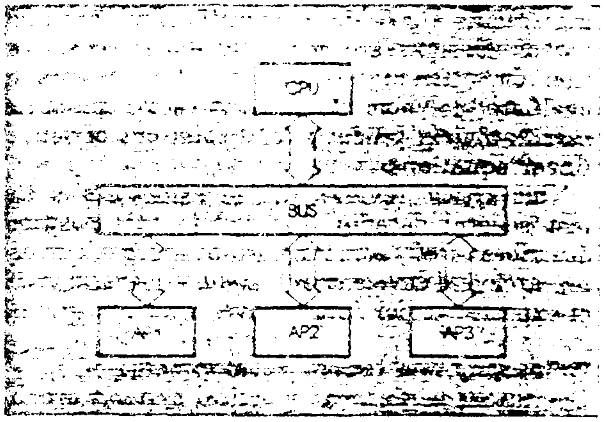


FIGURE 1. Array processors (AP) shown hosted by a general-purpose central processing unit (CPU). Two-way communications are via a data bus.

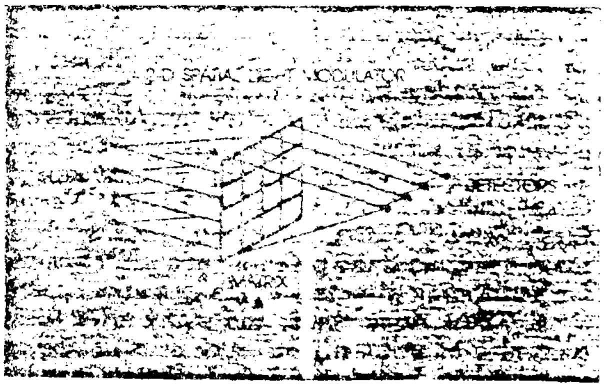


FIGURE 2. [illegible]

fabrication, local spatial interconnections between cells, reduced signal propagation delay times, and regular data flows eliminating synchronization problems.

Although the motivating factors are different, systolic processing algorithmic and architectural concepts are also applicable to optical implementation. This is primarily because of the regular data-flow characteristics of optical devices like acousto-optic cells and CCD detector arrays, and because of the ease of implementing regular interconnect patterns optically.

An example of an optical systolic matrix-vector multiplier is shown in Fig. 3. The processor consists of an input LED or laser diode array, collimating lenses for each LED, an acousto-optic cell, a Schlieren imaging system, and a linear array of integrating detectors. The pedagogical example of Fig. 3 is set up for the multiplication of a 2-component vector by a 2 × 2 matrix.

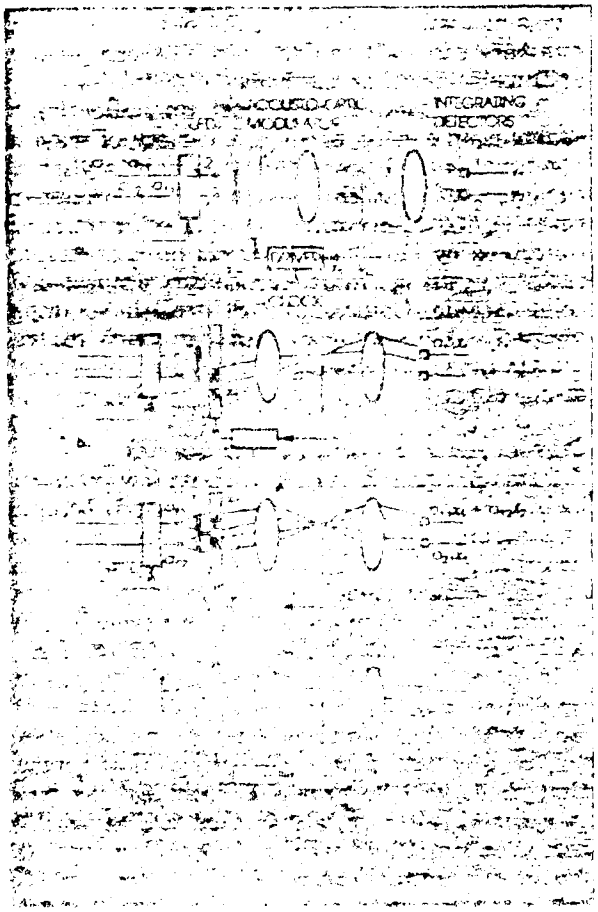


FIGURE [illegible]

The first input to the acousto-optic cell, vector component x_1, produces a short diffraction grating, with diffraction efficiency proportional to x_1, that moves across the cell. When that grating segment is in front of LED 1, as shown in Fig. 3(b), the LED is pulsed with light energy proportional to matrix coefficient a_{11}, and Integrating Detector 1 is illuminated with light energy in proportion to the product $a_{11}x_1$. The next critical moment occurs when the x_1 grating segment is in front of LED 2 and a second grating segment, with diffraction efficiency in proportion to vector component x_2, has moved in front of LED 1, as shown in Fig. 3(c). At that moment LED 1 is pulsed with light energy in proportion to a_{12} and LED 2, with light energy in proportion to a_{21}. The integrated output of Detector 1 is now proportional to $a_{11}x_1 + a_{12}x_2$, which is the output vector component y_1; the integrated output of Detector 2 is $a_{21}x_1$. The final critical moment in the computation, shown in Fig. 3(d), occurs after grating segment x_2 has moved in front of LED 2. A final pulse from that LED in proportion to a_{22} yields at the output of Detector 2 a voltage in proportion to $a_{21}x_1 + a_{22}x_2$, the second component y_2 of the output vector.

Much like the Stanford OMVM, the systolic optical processor described has a dynamic range and accuracy determined by the sources, modulator, and detectors. Output accuracy is limited to eight to ten bits. A realistic processing capability for such a system would be the multiplication of a 100-component vector by a 100 × 100 matrix in approximately 10 µs. This is much slower than the Stanford processor speed; however, unlike the latter, the systolic system does not require a 2-D SLM, and the matrix can be changed with each operation.

Shortly after the development of the optical systolic matrix-vector multiplier, two important advances took place—the invention of optical matrix-matrix multipliers (see box: "Matrix-Matrix Multipliers") by Dias; by Athale, Stilwell, and Collins; by Bocker, Bromley, and Caulfield; and by Casasent—and the achievement by Guilfoyle; by Athale, Collins, and Stilwell; and by Bocker, of digital accuracy with optical algebraic processors.

One method for obtaining high digital accuracy using optical processors is to implement digital multiplication by convolution. This method was first brought to the attention of the optical signal-processing community by Speiser and Whitehouse and first implemented by Psaltis et al. The method is explained with the aid of Fig. 4, where base-2 multiplication of the decimal numbers 39 and 15 is performed to obtain the decimal result 585. In Fig. 4(a) the multiplication is performed in normal fashion. Lines 1 and 2 contain the input binary numbers to be multiplied, Line 3 contains a mixed-binary representation of the output, and Line 4 contains the output in full binary. In the mixed-binary representation, each digit represents the multiplier of a power of 2; however, unlike full binary, the value of the digit is not restricted to be 0 or 1. One means for converting from mixed binary to full binary is shown in Fig. 4(b). Each digit of the mixed-binary representation (Line 3) is expressed in full binary form, and these binary numbers, appropriately shifted, are added using a standard base-2 adder. The resultant binary number, 1001001001, is the decimal product 585 expressed in base 2.

Binary multiplication via convolution is possible because the intermediate mixed-binary representation can be calculated as the discrete convolution (or serial product) of the binary input sequences. This is illustrated in Fig. 4(c). Convolution of binary sequences is easily accomplished by acousto-optic convolvers. Since only 1's and 0's need to be represented in the acousto-optic cells, cells can be operated at peak diffraction efficiency without concern for nonlinear response. Furthermore, the output detector is only required to have sufficient dynamic range to distinguish between a small number of light levels. For five-bit inputs, as in the example considered, the output levels will range, when quantized, from zero to five. In general, N-bit inputs require that N levels be distinguishable at the output. Negative numbers can be handled using 2's complement arithmetic or other methods.

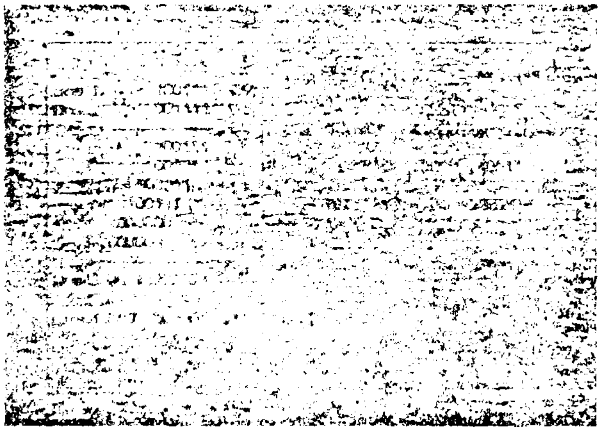


FIGURE 4 [illegible]

The above method for digital multiplication by convolution can be used in a variety of ways in algebraic optical processors to obtain higher accuracy, albeit at the cost of lower processing rates. A digital-accuracy matrix-vector processor conceived by Guilfoyle achieves high processing rates by using multitransducer acousto-optic cells. Athale, Collins, and Stilwell have implemented digital-accuracy outer-product matrix-matrix multipliers using a single pair of acousto-optic cells.

THE NEED FOR HIGH ACCURACY

Computers calculate by the same elementary operations (addition, subtraction, multiplication, division) that humans use. The result of each calculation has associated with it an uncertainty or error. Depending on the number, order, and nature of the required calculations, these errors can be multiplied greatly.

This is why 32- and even 64-bit accuracy computations are sometimes done even when a 6-bit answer will suffice. This is also why analog solutions (electronic or optical) to algebraic problems must often be avoided.

In electronics, analog computers are used for high-speed, easily implemented operations, but digital computers are used for algebra. Not surprisingly, optical computation makes the same division of tasks.

Current research and new directions

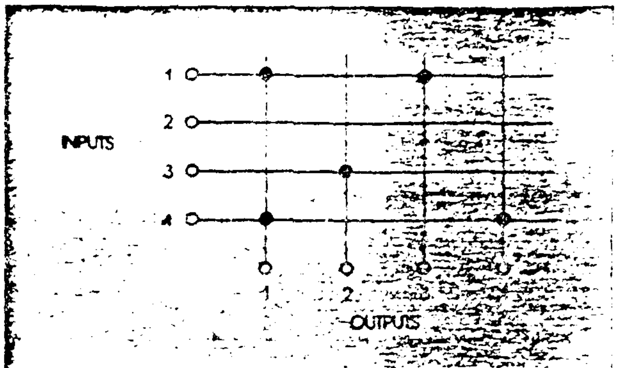
Efforts undertaken during the next few years will be in two directions. First, optical matrix computer systems based on the concepts we have been describing will be built, tested, improved, and applied to new areas. Second, new types of nonmatrix optical computers will be developed. We will touch on both of these directions briefly.

In optical matrix computers the two thrusts are implementation and extension. To date, very little implementation has taken place. Doing this will require both time and money; it now appears that these will be provided. Practical issues of component selection, electronics, and system integration must and will be faced. However, the design of practical optical matrix processors still [illegible]. Complex operations must be planned, e.g., Kalman filtering. Kalman filtering is a means for obtaining the statistically best estimate of the current and future state of a process governed by a known differential equation and measured in a fixed way with known measurement statistics. Because a single "cycle" of a Kalman filtering operation involves many matrix calculations, real-time Kalman filtering must be restricted to relatively small problems. Performing the matrix operations (triple multiplications, inversions, etc.) optically may permit the handling of large problems in real time. Casasent has started this effort, and several others are working on it. Either floating-point operations or on-the-fly scale adjustment is needed. Caulfield has shown that both are possible, but his solutions are probably more existence proofs than final answers. New algorithms are needed to extend the range of applications and, possibly, to speed up calculations. To date, all important algorithms have been iterative. Noniterative, fully parallel solution of linear equations is possible in analog optical processors. Can similar things be done for digital optical processors?

Nonmatrix optical processors are developing independently and rapidly. Perhaps the most widely pursued of these is the use of optics to make arbitrary interconnections among electronic (Goodman) or electro-optic (Lohmann, Lee, Collins, Goodman, Sawchuck, Strand, etc.) systems. Sawchuck, Strand, and their coworkers have implemented a variety of space-variant and space-invariant interconnect patterns using computer holograms to generate the patterns and spatial light modulators to feed the information back into the system. Their system (like those due to Lohmann, Lee, Collins, etc.) closes on itself for feedback. Clearly, however, this is not the only configuration. Feedforward configurations lead to a variety of optical artificial-intelligence systems.

The continuing demand for higher throughput rates will drive future research toward higher speeds and greater parallelism. In these large systems, or supercomputers, of the future, a major problem in achieving high throughput rates will be how to facilitate generalized communications among the large number of processing units. In a general-purpose computer, the full advantage of parallelism will only be realized if each processing unit has direct communication with every other unit, thus permitting each to handle a part of the action on a continuing basis.

The highest level of communications, or interconnect as it is called, entails a generalized crossbar network involving N^2 interconnects available for N processors (N units communicating with N units), as shown in Fig. 5. Such a



Suppose we have M input signals that we wish to connect to N output receivers. Each input can be connected to 0, 1, 2, ... N outputs. This concept is best represented as a crossbar arrangement, like that shown here for $M = N = 4$. In this diagram we have used a dark spot ● to indicate connections made. Thus input 1 is connected to Outputs 1 and 3, etc. If we regard the input as a vector

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

and the output as a vector

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

then we can write

$$\vec{y} = \mathbf{A}\vec{x},$$

where **A** is a binary matrix. In the case just illustrated,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The rule for obtaining **A** is simple. Form a matrix **B** with a 1 where every connection in the crossbar plot occurs. In the example,

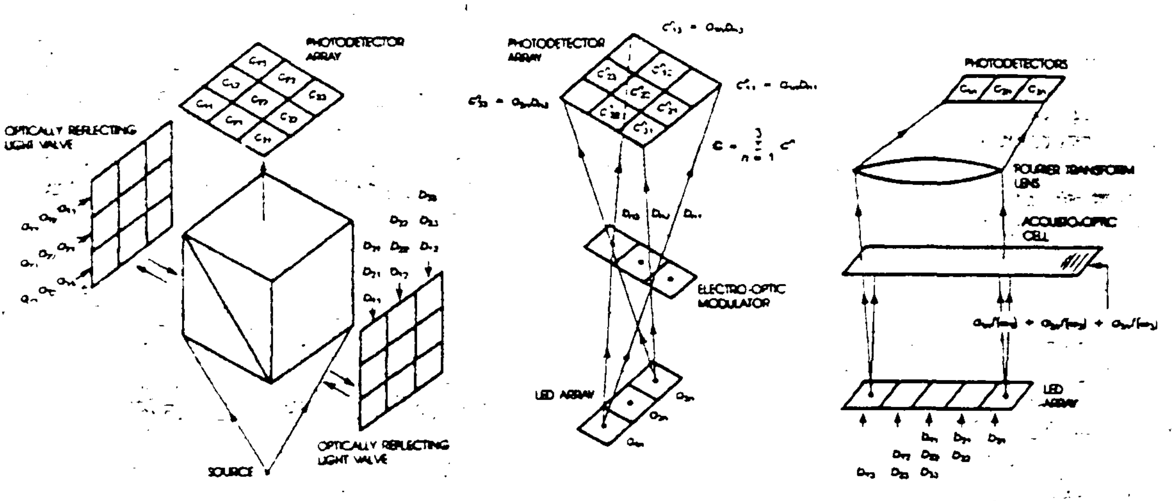
$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Then

$$\mathbf{A} = \mathbf{B}^T,$$

where T indicated transportation (row/column exchange).

FIGURE 5. Generalized crossbar network



(a) RUBIC CUBE

(b) OUTER PRODUCT

(c) FREQUENCY MULTIPLEXED MATRIX-VECTOR

MATRIX-MATRIX MULTIPLIERS. (a) *RUBIC cube.* This is a systolic architecture whose major components are a pulsed noncoherent light source, a spatial light modulator for each of the two input matrices, a 2-D photodector array for reading out the output matrix, and a polarizing beam splitter. The two light modulators synchronously march the matrix information across the optical aperture, where the proper terms superimpose to produce each element of the output matrix.

(b) *Outer product processor.* If one desires to relax the dimensionality requirements of the input devices, the matrix-matrix problem may be formulated in terms of outer products, rather than the customary inner products. For the multiplication of two $N \times N$ matrices **A** and **B**, the ouptut may be expressed as

$$\mathbf{C} = \sum_{n=1}^{N} \mathbf{C}^n = \begin{matrix} a_{1n}b_{n1} & \cdots & a_{1n}\,b_{nN} \\ \vdots & & \\ a_{Nn}b_{n1} & \cdots & a_{Nn}b_{nN}. \end{matrix}$$

Each matrix term may be taken as the outer product between the nth column vector of **A** and the nth row vector of **B**. This may be done optically as shown in the figure employing two crossed arrays. The summation of the individual matrices may be realized via a 2-D time integrating photodector array.

(c) *Frequency multiplexed.* This is a systolic architecture that uses a linear LED array, an acousto-optic cell, a Fourier transform lens, and a linear photodetector array. Input matrix **B** is fed in the space- and time-multiplexed fashion shown (rows of **B** spatially multiplexed and columns time multiplexed), while the matrix **A** is multiplexed in frequency and space, using the acousto-optic cell. Each row element of matrix **A** is placed on a separate frequency carrier, such that after multiplication with the appropriate **B** elements via acousto-optic modulation, the resulting output term is deflected by the transform lens to a particular photodetector element, depending on the carrier frequency. This architecture may be viewed as a matrix-vector multiplier in which frequency multiplexing is used to expand the vector to a matrix.

network becomes very expensive when implemented electronically for large N, but the inherent parallelism of optics holds great potential for inexpensive and high-speed crossbar switching.

The generalized crossbar can be expressed analytically in terms of a vector-matrix multiplication, so optical algebra forms the basis of solving the interconnect problem. For example, consider the Stanford OMVM described previously. Let $\bar{x}$ and $\bar{y}$ be the vectors of the crossbar inputs and outputs, respectively, and let A represent the interconnect switch settings. That is, $a_{ij} = 1$ if, and only if, the *i*th output is connected to the *j*th input. Otherwise, $a_{ij} = 0$. The OMVM with these a_{ij}'s automatically makes the desired connections optically. Note, too, that numerical accuracy is not an issue for this application.

The Stanford processor is, of course, nonprogrammable, therefore, it can only be used in a system with a pre-established set of interconnects. If one were to replace the matrix filter with a real-time device such as a 2-D spatial light modulator, then a switchable, generalized crossbar becomes a possibility; likewise, the binary matrix mask could be replaced with a hologram. Going one step further, one begins to envision generalized crossbars with picosecond switching speeds via real-time four-wave mixing or an optically addressed bistable array. Such a capability would bring us into a realm of computer communications beyond the wildest dreams of electronic interconnection architects.

A more structured optical arrangement is the fiberoptic lattice filter (Tur, Goodman, etc.). When the computational problem has sufficient symmetry, a full matrix approach may be an inelegant and expensive approach. The lattice filter work represents an exploration of simpler systems for simpler problems. A very common problem in algebra is the evaluation of polynomials. If an analog optical polynomial evaluator could be built, it would be possible to find the roots of polynomials in a totally new way: scan the independent variable(s) and see where the roots occur. This leads to a solution of another long-standing optical problem as well. The quotient $1/a$ is simply the root of the function $(1/x) = a$, which can be evaluated efficiently in polynomial form. Work along this line is being carried out (Verber, Caulfield, Ludman, Stilwell, etc.). Since holographic memory technology allows ready content-addressable access to vast amounts of data, a truth-table lookup processor appears both feasible and appealing. This approach is now being studied closely (Gaylord, etc.).

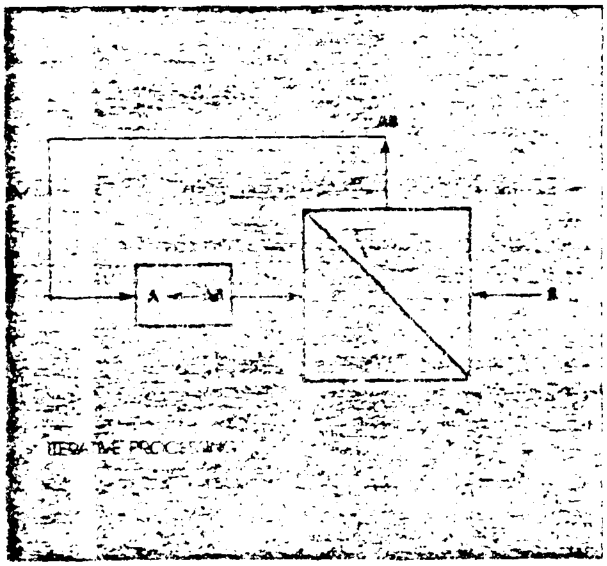


FIGURE 6. Architecture for performing iterative processing with the [illegible] using [illegible]

Finally, all of these optical computers are in need of improved or specialized components. A major DARPA-sponsored effort to improve spatial light modulators is just beginning. This seems likely to lead to improved throughput rates by providing a 2-D medium capable of 1000 × 1000 individually addressable modulator elements, a cycle rate (READ/WRITE time cycle) of 1 kHz, a dynamic range of 30 dB, and less than 3% spatial nonuniformity. Other needs include source and detector arrays that are compatible in resolution, intensity, and dynamic range with these spatial light modulators and that possess individually addressable elements.

Conclusions and outlook

Upon considering the broad area of optical algebra, including parallel algorithms, architectures, devices, and their associated materials, a large spectrum of interesting and important research areas comes to "light." As the national interest in the computational sciences begins to shift toward the supercomputers envisioned for the 1990s, it will be vitally important for the optics community to pursue those research areas for which optics holds the greatest appeal, such as large-scale matrix-matrix or matrix-tensor operations and processor inter- and intracommunications. We must also allow ourselves to look past the research discussed above and into the use of optics to perform real-time circuit reconfiguration. For example, light could be used to modify the index of refraction within waveguides in such a manner as to change channel layouts and beam-control elements on a circuit module, thereby adding much-needed flexibility to optical computing. These new directions are mentioned to convey to the reader something of the excitement of a field that is not only maturing, but also expanding.

Acknowledgment

Many of the ideas presented in this paper were topics of discussion at a May 1983 workshop, "Optical Techniques for Multi-Sensor-Array Data Processing," sponsored by the Army Research Office and the Air Force Office of Scientific Research.

Further reading

Rather than provide a complete list of specific references, which would lengthen the article considerably, the authors direct the interested reader to the following general sources. Much recent research on optical computing architectures is reported in *Applied Optics* issues of the past two years. In addition, the reader is referred to proceedings of conferences on the subject: *Advances in Optical Information Processing*, G. M. Morris, ed. (*Proc. SPIE 388*, 1982); *10th International Computing Conference* (IEEE, 1983, Catalog No. 83CH1880-4); *Real Time Signal Processing VI*, K. Bromley, ed. (*Proc. SPIE 431*, to be published late 1983 or early 1984); *Optical Engineering*, Jan. 1984. For papers reviewing the general area of analog optical signal processing, see the following: *Proc. IEEE 69*, 1 (Jan. 1981), special issue on acousto-optic signal processing; *Proc. IEEE 65*, 1 (Jan. 1977), special issue on optical computing; *Proc. IEEE 62*, 10 (Oct. 1974), invited paper by A. B. Vander Lugt.

H. JOHN CAULFIELD is Principal Research Scientist at Aerodyne Research, Inc., 45 Manning Rd., Billerica, MA 01821; JOHN A. NEFF is Program Manager, Defense Science Office, DARPA, Washington, DC 22209; WILLIAM T. RHODES is Professor of Electrical Engineering at Georgia Institute of Technology, Atlanta, GA 30332.

APPENDIX I

NEW FORM OF NUMBER REPRESENTATION SUITABLE FOR OPTICAL IMPLEMENTATION

(Submitted to Applied Optics)

EFFICIENT REAL NUMBER REPRESENTATION WITH ARBITRARY RADIX

H.J. Caulfield, D.S. Dvore, and J.H. Gruninger
Aerodyne Research, Inc.
45 Manning Drive
Billerica, MA 01821

ABSTRACT

Because most optical digital computers use only nonnegative quantities, it is of great interest to find an efficient way to represent real numbers. For radix 2 (binary) numbers the twos complement method requires only one extra digit beyond that needed for non negative numbers. We introduce here an arbitrary radix generalization.

BACKGROUND

Optical computers (1-10) have become extremely popular because of their speed, low power consumption, and relatively low volume and weight. Digital number representation is as necessary for accuracy in optics as it is in electronics. In ootical digital computers the optimum radix choice is by no means clear and may even be computer or problem dependent. For radix 2 (binary) representation, the twos complement method (11) is an optimally-efficient way to represent real numbers in that an N-bit real number can be represented with only N + 1 digits. Obviously no more efficient representation can be devised. We have been unable to locate in the literature a scheme for representing N digit radix R (> 2) numbers with only N + 1 digits. This work represents our attempt at the needed generalization.

EXPOSITION APPROACH

Our exposition will proceed in two stages aimed at making the method understandable. We avoid theorum and lemma proving in favor of simplicity and clarity. The method works only with even radix. First, we will illustrate this method with examples from the familiar radix 10 numbers. Second, we will offer an explanation which is radix independent.

NOTATION (Radix 10)

We suppose that the numbers of interest are of absolute value less than 10^N, where N is a preselected integer such as 4. For N = 4, the numbers lie between -9999 and 9999. Thus only N digits are needed to represent the absolute value. To this we add a single sign digit. The sign digit for a positive number will be 0, 2, 4, 6, or 8. The sign digit for a negative number will be 1, 3, 5, 7, or 9. For negative numbers we complement the absolute value, i.e., subtract it from 10^N. For convenience of notation, we give this new method the name "parity sign" and the normal representation "arithmetic". Table 1 shows some sample arithmetic and parity sign representations of the same number.

ADDITION EXAMPLES

Let us add +0012 to +0008. We know that the answer is +0020. In parity sign we might have

$$\begin{array}{r} 20012 \\ +\ 80008 \\ \hline 100020 \end{array} \qquad (1)$$

Table 1.

The same radix 10 numbers represented in both arithmetic and parity sign notation. For each number there is one and only one arithmetic representation but five equally-valid parity sign representation.

Arithmetic Representation	Acceptable Parity Sign Representation
+0012	00012
+0012	80012
+0012	20012
-0012	19988
-0012	99988
-9092	70908
+0008	40008

The last five digits are 00020 which is one of the parity sign representations of +0020. Now let us add +0008 to -0012. We might write

$$\begin{array}{r} 40008 \\ +\ 99988 \\ \hline 139996 \end{array} \qquad (2)$$

The last five digits are 39996 which is one of the parity sign representations of -0004.

MULTIPLICATION EXAMPLES

Let us multiply +0012 by +0004. We might write

$$\begin{array}{r} 00012 \\ \times\ 40004 \\ \hline 00048 \\ 00000 \\ 00000 \\ 00000 \\ 00048 \\ \hline 000480048 \end{array} \qquad (3)$$

The last five digits are 80048 which is one of the parity sign representations of +0048.

Now let us multiply -0012 by +0004. We might write

$$\begin{array}{r} 99988 \\ \times\ 00004 \\ \hline 399952 \end{array} \qquad (4)$$

The last five digits are 99952 which is one of the parity sign representations of -0048.

EXPLANATION

We are used to graphing the arithmetic representation of a number versus itself (i.e. plotting $f(x) = x$ in arithmetic notation). Figure 1 shows such a plot for the domain $|x| < 10^5$. If we restrict $|x|$ to that domain, we can plot a multivalued representation $m(x)$ vs. x as shown in Figure 2. If we now restrict ourselves to $m(x) > 0$, we can still represent any number in $|x| \leq 10^5$. the negative x's will have an odd fifth digit. Even numbers will have an even fifth digit. Furthermore

$$m(x + y) = m(x) + m(y), \tag{5}$$

where we mean by $m(x)$ all of the values of $m(x)$, etc. Likewise

$$m(xy) = m(x)m(y) \tag{6}$$

OTHER EXAMPLES

For the special case of radix 2 we obtain a signed twos complement. Thus +0011 plus -1010 (+3 -10 in radix 10) is in parity sign

$$\begin{array}{r} 00011 \\ +\ 10101 \\ \hline 11000 \end{array} \tag{7}$$

which is the parity sign representation of -0111 (-7 in radix 10). Thus in the binary case the parity sign digit is no longer multiple.

CONCLUSION

The parity sign representation is easy to use and easy to understand. It includes the traditional binary signed twos complement method as a special case while extending the one-digit-sign-indication efficiency advantage to arbitrary radix. Finally, one must be careful to prevent "overflow" - attempted calculation of numbers greater than the maximum the system is designed to handle. When overflow occurs, the numerical part of the result (in our example, the last four digits) is correct but the amount of overflow is undetermined.

The simplest way to prevent overflow is to test input numbers. We suggest the following, quite-conservative test for radix r amplitudes which must be less than r^{2N}. We write

$$r = 2s$$

since r is even. We ignore the sign digit and require

(1) For multiplication both numbers be less than r^N, so the first s most significant digits must be zero and

(2) For addition both numbers be less than $s \cdot r^{2N-1}$ and therefore the most significant digit must be s-1 or less.

ACKNOWLEDGEMENT

This work was performed under contract to Rome Air Development Center, Hanscom Air Force Base, Contract No. F19628-82-C-0068.

REFERENCES

1. J.W. Goodman, A.R. Dias, and L.M. Woody, Optics Lett. 2, 1 (1978).

2. D. Psaltis, D. Casasent, and M. Carlotto, Optics Lett. 4, 348 (1979).

3. H.J. Caulfield, W.T. Rhodes, M.J. Foster, and S. Horvitz, Opt. Common. 40, 86 (1981).

4. R.A. Athale and W.C. Collins, Appl. Opt. 21, 2089 (1982).

5. D. Casasent, J. Jackson, and C. Neumann, Appl. Opt. 22, 115 (1983).

6. R.P. Bocker, H.J. Caulfield, and K. Bromley, Appl. Opt. 22, (1983).

7. R.P. Bocker, Appl. Opt. 16, 2401 (1983).

8. R.A. Athale, W.C. Collins, and P.D. Stilwell, Appl. Opt. 22, 365 (1983).

9. P.S. Guilfoyle, Opt. Eng. 23, 20 (1984).

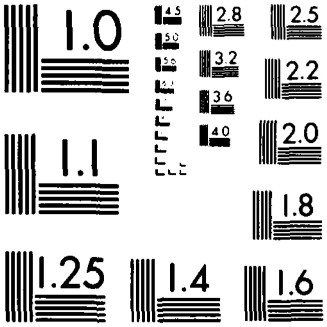
10. R.P. Bocker, Opt. Eng. 23, 26 (1984).

11. Most texts in numerical analysis cover this. For example: D.M. Young and R.T. Gregory, A Survey of Numerical Mathematics, Vol. 1, Addison-Wesley, Reading, MA (1972), pp 30-35.

AD-A149 550 OPTICAL EIGENVECTOR(U) AERODYNE RESEARCH INC BILLERICA 2/2
MA J CAULFIELD OCT 84 ARI-RR-393 RADC-TR-84-208
F19628-82-C-0068

UNCLASSIFIED F/G 20/6 NL

END
FILMED
DTIC



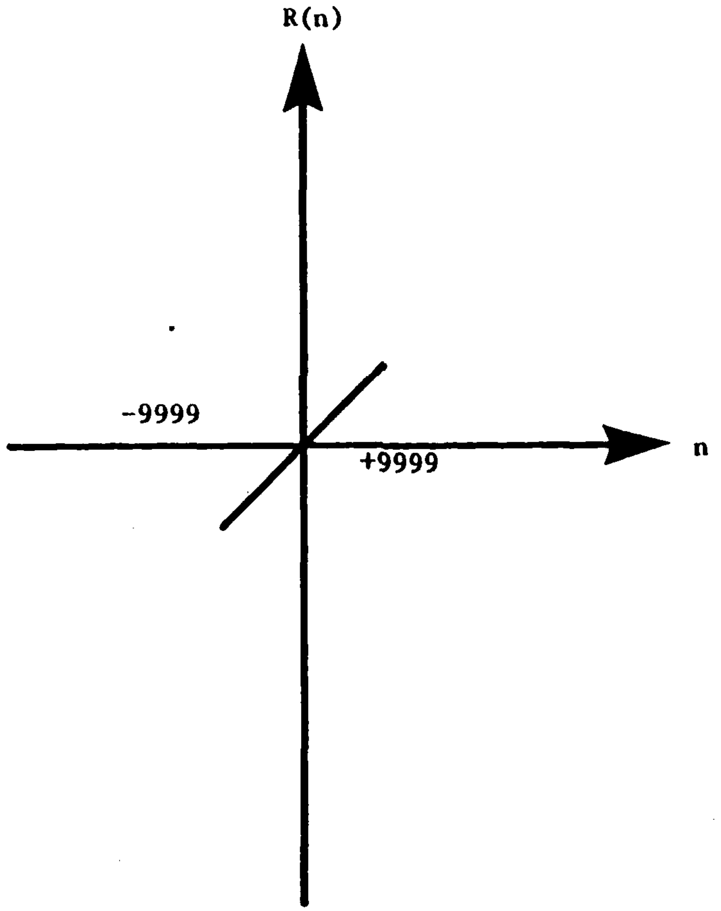
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

FIGURE CAPTIONS

Figure 1: A representation R(n) of numbers n satisfying $-9999 \leq n \leq 9999$.

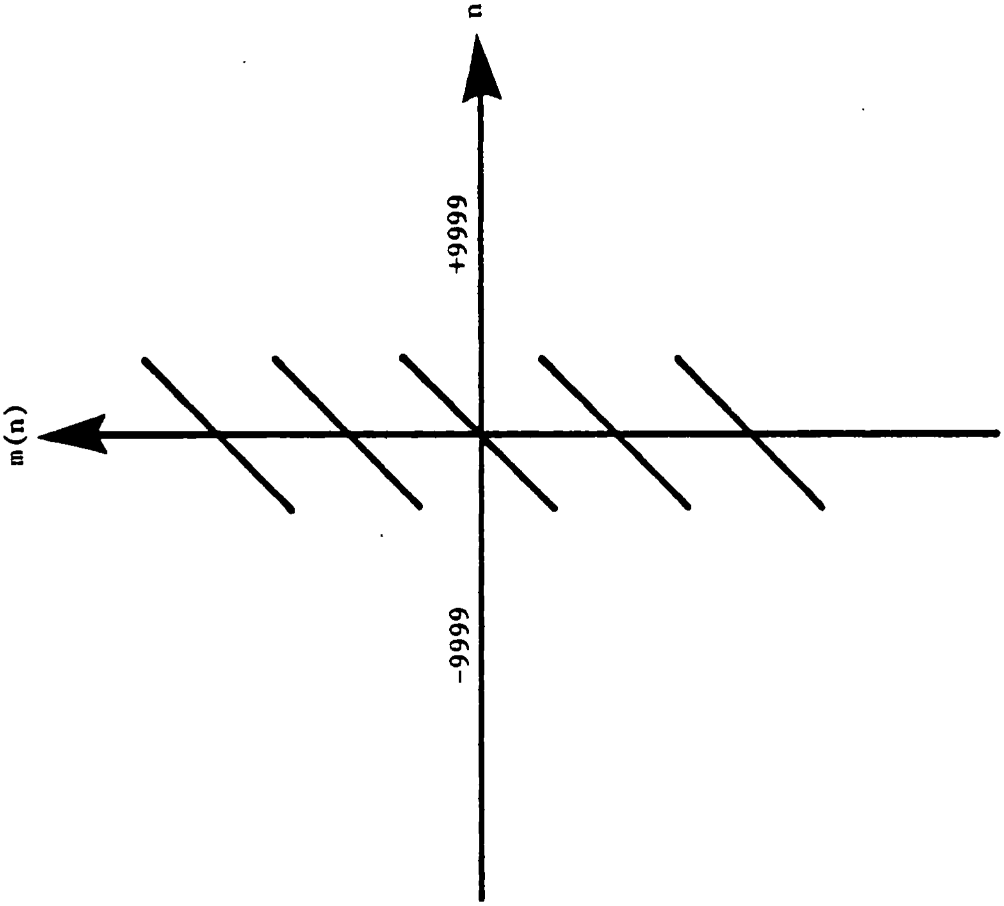
Figure 2: A multivalued representation m(n). All values of m(n) for a given n are equally valid.

84-152



Caulfield Fig 1

84-153



Caulfield Fig 2

APPENDIX J

FLOATING POINT OPTICS
FOR MATRIX VECTOR MULTIPLIERS

Floating point optical matrix calculations

H. J. Caulfield
Aerodyne Research, Inc.
45 Manning Road
Billerica, Massachusetts 01821

Abstract. The recent explosion of interest and activity in optical numerical processing has occurred despite the fact that calculations had to be carried out with integer or fixed point arithmetic. We show here that floating point optical matrix-vector multiplication is feasible.

Keywords: optical computing; matrix calculations; algebra; floating point systems.

Optical Engineering 22(6), 765-766 (November/December 1983).

CONTENTS

1. Background on floating point algebra
2. Background on optical vector-matrix multipliers
3. Dual representation approach
4. Unresolved problems
5. Personal conclusion
6. Acknowledgment
7. References

1. BACKGROUND ON FLOATING POINT ALGEBRA

A wide variety of new architectures and algorithms for optical matrix operations have been introduced recently.[1-6] Without exception these have used fixed point arithmetics. The sustained interest in these systems arises from the high capacity, high speed, and low power consumption of these optical computers and from the fact that their fixed point calculations can be very accurate.[7-9] Of course, the range of applications could be expanded tremendously if floating point calculations could be performed.

In floating point notation (base b) every number is written

$$n = 0.i_1 i_2 i_3 \cdots i_M \times b^e , \tag{1}$$

where $i_1, i_2, \ldots, i_M$ are integers between 0 and $b - 1$, M is a preset integer, $i_1 \neq 0$, and e is an integer. We call

$$m = 0.i_1 i_2 \cdots i_M \tag{2}$$

the mantissa and e the exponent. With two numbers of the form

$$n_1 = m_1 b^{e_1} \tag{3}$$

and

$$n_2 = m_2 b^{e_2} , \tag{4}$$

we have

$$n_1 n_2 = m_1 m_2 b^{e_1 + e_2} . \tag{5}$$

Neither the mantissa multiplication $(m_1 m_2)$ nor the exponent addition $(e_1 + e_2)$ is difficult to achieve optically. What is necessary but far more difficult optically is adding n_1 and n_2. Obviously,

$$n_1 + n_2 = m_1 b^{e_1} + m_2 b^{e_2} = m_3 b^{e_3} . \tag{6}$$

In a computer one finds the larger of e_1 and e_2. Without loss of generality, we assume $e_1 > e_2$.

Clearly,

$$n_1 + n_2 = m_1 b^{e_1} + m_2 b^{e_2} b^{e_1} b^{-e_1} = (m_1 + m_2 b^{e_2 - e_2}) b^{e_1} . \tag{7}$$

Therefore,

$$e_3 = e_1 , \tag{8}$$

and m_3 is calculated by rounding off $m_1 + m_2\, b^{e_2 - e_1}$ after M places. We see no obvious way to do those steps optically, so we have adopted a new but largely equivalent approach.

2. BACKGROUND ON OPTICAL VECTOR-MATRIX MULTIPLIERS

The prototypical modern vector-matrix multiplier is that of Goodman et al.[1] More recently systolic and engagement versions have been introduced to simplify hardware and speed up the operations.[2-6] All of these start with a linear array of N discrete incoherent light sources representing the input vector components and produce a linear array of N discrete points of light (each of which is detected on a discrete detector) to give the N components of the product vector. The Goodman processor calculates the full matrix "instantly," while the systolic approaches require time integration over N pulses to arrive at the final answer. The floating point need is present in both

Short Communication SC-6108 received June 22, 1983; accepted for publication July 11, 1983; received by Managing Editor July 18, 1983

3. DUAL REPRESENTATION APPROACH

The approach proposed here is limited to the systolic processors. The key idea is to use different means for representing input and output numbers optically The input vector components are represented by encoding both their mantissas and exponents as source brightnesses in separate and parallel processors. One processor does nothing but multiply mantissas. A similar but separate processor adds the exponents. We assume

$$-e_m \leq e_1, e_2 \leq e_m \tag{9}$$

and encode e as

$$f = e + e_m ; \tag{10}$$

clearly,

$$0 \leq f \leq 2\, e_m \tag{11}$$

The signal

$$f_3 = f_1 + f_2 = 2\, e_m + (f_1 + f_2) \tag{12}$$

is used to drive an optical light deflector to one of $(2e_m + 1)$ possible positions (one for each possible value of $f_1 + f_2$) spatially normal to the line of output vector component points. A two-dimensional detector array [N vector components by $(2e_m + 1)$ exponents] receives the deflected light and integrates over the required N pulses. Then for each vector component we call the highest nonzero value of an exponent e_0. We then take the time-integrated mantissa products on that detector, add to that 1/b of the products for the next lowest e value, etc., until we have added all of the products. That weighted sum we call s_0. We then write

$$n_1 + n_2 = S_0 b^{e_0} . \tag{13}$$

While this looks in form like Eq. (1), the condition in Eq. (1) that $i_1 \neq 0$ may not hold. Thus, we may not have $e_0 = e_3$ Nevertheless, this is a floating point operation with all of the accuracy advantages thereof

4. UNRESOLVED PROBLEMS

Two major problems with this technique remain unsolved. First, simple electro-optical light deflectors are very fast but do not give many resolvable spots (limiting e_m), while mechanical or acousto-optic light deflectors give many resolvable spots but may slow up the system too much. Thus, the choice of deflector is critical and difficult. Second, because one spatial dimension is used for the exponent, it is by no means clear if this technique is extendable to the modern optical matrix-matrix multipliers[4,5] which already require a two-dimensional detector array.

5. PERSONAL CONCLUSION

It has been my experience that an "existence proof" (such as I have offered here for optical floating point algebra) invariably produces almost immediate improvements by others. I trust and hope this will happen here.

6. ACKNOWLEDGMENT

This work was supported under Contract No. F19628-82-C-0068 from Rome Air Development Center, Hanscom AFB, MA 01731.

7. REFERENCES

1. J. W. Goodman, A. R. Dias, and L. M. Woody, Opt. Lett. 2, 1(1978).
2. H. J. Caulfield, W T. Rhodes, M. J. Foster, and S. Horvitz, Opt. Commun. 40, 86(1981).
3. M. Carlotto and D. Casasent, Appl. Opt. 21, 147(1982).
4. R. A. Athale and W C. Collins, Appl. Opt. 21, 2089(1982).
5. R. P Bocker, H. J. Caulfield, and K. Bromley, Appl. Opt. 22, 804(1983).
6. D. Casasent, J. Jackson, and C. Neuman, Appl. Opt. 22, 115(1983).
7. R. A. Athale, W. C. Collins, and P. D. Stillwell, Appl. Opt. 22, 368(1983).
8. R. P. Bocker, S. R. Clayton, and K. Bromley, Appl. Opt. 22, 000(1983).
9. P. Guilfoyle, Personal communication (1982).

APPENDIX K

FLOATING POINT OPTICAL COMPUTATION FOR ALL MATRIX OPERATIONS

Spatial encoding for optical floating point computation

H. John Caulfield

Following the lead of electronic computers, optical computers must adopt floating point calculation to allow for both high accuracy and high dynamic range. Given here is a method for using spatial encoding for that purpose.

I. Introduction

High numerical accuracy is required for most algebraic calculations. Long ago this forced electronic computer designers to adopt digital rather than analog number representations and to introduce floating point calculations. Recently optical computer designers have devised a number of ways of using digital number representations.[1-4] Thus the remaining step is floating point calculation. Although a preliminary step toward floating point optical computing has been taken,[5] no universally applicable method is known.

II. Basic Concepts

The basic idea of floating point operation is to represent a positive number by

$$n = m \times b^{e}, \tag{1}$$

where m = a positive number within a well-defined range.
b = a fixed positive integer called the base or radix, and
e = a real (positive or negative) integer called the exponent.

(Negative and even complex numbers are easily represented also, but this would be an unnecessary digression here.) In an electronic digital computer one normally keeps $b > m \geq 1$. For optical computing we may relax that requirement slightly.

Now consider two numbers:

$$n_1 = m_1 \times b^{e_1}, \tag{2}$$

$$n_2 = m_2 \times b^{e_1} \tag{3}$$

The two operations optical computers perform are addition and multiplication. Multiplication is the easier task. We have

$$n_1 \times n_2 = (m_1 \times m_2) \times b^{e_1+e_2} \tag{4}$$

We already know how to multiply m_1 by m_2. We need to add e_1 and e_2 at the same time. Adding two integers of moderate size can be done either electronically or optically. The only problem appears to be that of bringing $m_1 \times m_2$ back within the desired range before subsequent calculations. This, it appears, will be a recurrent problem in optical floating point operations. If

$$b > m_1, m_2 \geq 1, \tag{5}$$

then

$$b^2 > m_1 \times m_2 \geq 1. \tag{6}$$

If we have time to test $m_1 \times m_2$ we can either use it (if $m_1 \times m_2 < b$) or divide it by b (if $b^2 > m_1 \times m_2 > b$) and replace $e_1 + e_2$ by'$e_1 + e_2 + e_1$.

Adding n_1 to n_2 is more difficult. In an electronic computer we calculate

$$n_1 + n_2 = m_1 b^{e_1} + m_2 b^{e_2}. \tag{7}$$

If we determine $e_1 \geq e_2$, then

$$n_1 + n_2 = (m_1 + m_2 b^{e_2-e_1}) b^{e_1}. \tag{8}$$

Since

$$b^{e_2-e_1} \leq 1 \tag{9}$$

(for $e_1 \geq e_2$), this means attenuating m_2 before adding it to m_1. Finally, in an electronic computer, we round off $m_1 + m_2 b^{e_2-e_1}$ to the desired number of bits. Thus there is a nonlinear decision step which is difficult to implement optically.

These difficulties are compounded by the fact that all optical matrix algebra computers involve accumulating products, e.g.,

$$n (n_1 \times n_2) + (n_3 \times n_4) + (n_5 \times n_6) \tag{10}$$

The author is with Aerodyne Research, Inc., 45 Manning Road, Billerica, Massachusetts 01821.

Received 15 October 1983.
0003-6935/84/020239-03$02.00/0.

That is, multiple products must be added. What follows is a solution (indeed several solutions) to this problem.

III. Vector-Matrix Multipliers

Here we use optics to calculate

$$A\mathbf{x} = \mathbf{y} \tag{11}$$

or

$$y_i = \sum_{j=1}^{N} a_{ij} x_j. \tag{12}$$

We assume that all a_{ij} and x_j values are furnished in floating point form.

We must begin with a naive and totally fallacious solution. We could introduce an attenuator with transmission $b^{e_1+e_2}$ before the detector of $m_1 m_2$. This places the whole burden on the dynamic range and repeatability of the multiplier. That is, it offers no improvement over fixed point operation.

We conclude that we need a separate detector for each $e_1 + e_2$ value. Then, in final readout, the accumulated values in each $e_1 + e_2$ bin are first thresholded to eliminate pure noise and then added with appropriate weights to give the final result.

Let us illustrate with the following $b = 10$ example:

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 10 \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} 2 \\ 10 \end{bmatrix}. \tag{13}$$

$$\mathbf{y} = A\mathbf{x} = \begin{bmatrix} 1 \times 2 + 2 \times 10 \\ 2 \times 2 + 10 \times 10 \end{bmatrix}.$$

We write

$$A = \begin{bmatrix} 1 \times 10^0 & 2 \times 10^0 \\ 2 \times 10^0 & 1 \times 10^1 \end{bmatrix}, \tag{14}$$

$$\mathbf{x} = \begin{bmatrix} 2 \times 10^0 \\ 1 \times 10^1 \end{bmatrix}. \tag{15}$$

Then

$$\mathbf{y} = \begin{bmatrix} 1 \times 2 \times 10^{0+0} + 2 \times 1 \times 10^{0+1} \\ 2 \times 2 \times 10^{0+0} + 1 \times 1 \times 10^{1+1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \tag{16}$$

We suppose that there are at least three detectors for each y_i corresponding to 10^0, 10^1, and 10^2 products. We then operate electronically according to the decision tree of Fig. 1. Clearly we can achieve as many exponent sums as we have detectors.

In optical vector-matrix multipliers each y_1 is detected by a single detector. To allow floating point calculation, we need to replace each single detector with multiple detectors—one for each possible exponent sum. One way to do this is to use a light deflector for each y_i driven to deflect the mantissa product onto the appropriate detector. This is the method of Ref. 5.

In integrated optical matrix-vector multipliers the deflectors might be built in since the common base material (lithium niobate) makes a good acoustooptic deflector.[5]

In bulk optics, convenience demands many deflectors on a single substrate. This is now practical. Another way of doing this is encoding the exponent sum (computed electrically) as a frequency of modulation on the input light and frequency analyzing the output with an acoustooptic frequency analyzer.

Unfortunately these methods fail for matrix-matrix multipliers which already use 2-D detector arrays.

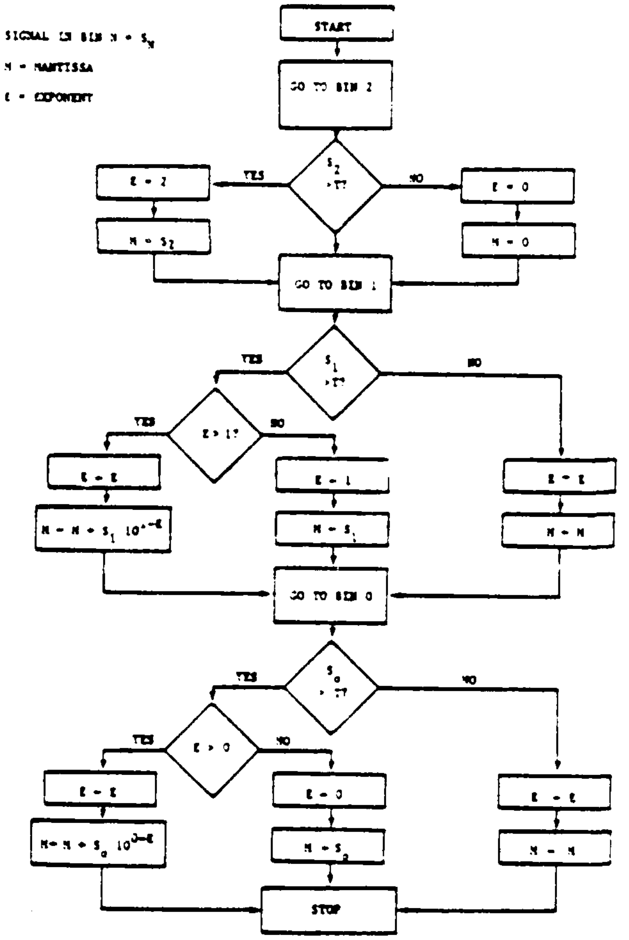


Fig. 1. Logic for assigning mantissa and exponent to the number accumulated in three bins (detectors).

IV. Matrix-Matrix Multipliers

As just indicated, deflectors seem impractical for matrix-matrix multipliers, so alternatives must be considered.

One alternative is to use frequency encoding of exponent sums as suggested before but analyze the bins electrically. This is no real solution in the sense that it still uses only one detector for all the frequency bins to be searched. Indeed we must accept multiple detectors as a fundamental price to be paid for floating point operation.

Spatial encoding exponents seem to be a rational approach to the problem. We show below how we might encode the matrix-vector problem used as an earlier example:

$$A = \frac{1}{\sqrt{3}} \begin{array}{cc} & \text{Exponent} \\ \left[\begin{matrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}\right] & \begin{matrix} 0 \\ 1 \\ 2 \\ 0 \\ 1 \\ 2 \end{matrix} \end{array}, \tag{17}$$

$$\mathbf{X} = \frac{1}{\sqrt{3}} \begin{array}{cc} \left[\begin{matrix} 2 & 0 & 0 \\ 2 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{matrix}\right] & \\ \begin{matrix} 0 & 1 & 2 \end{matrix} & \text{Exponent} \end{array}. \tag{18}$$

Note that each number is represented by a 3 × 3 array of numbers. In A the number is repeated three times horizontally. In X the number is repeated three times vertically.

Multiplying, we have

$$\mathbf{y} = \begin{bmatrix} a_{11}X_1 + a_{12}X_2 \\ a_{21}x_1 + a_{22}X_2 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 2 & 0 & 0 \\ 2 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 2 & 2 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \\ \\ \frac{1}{3}\begin{pmatrix} 2 & 2 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 2 & 0 & 0 \\ 2 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \\ \begin{pmatrix} 4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{pmatrix} 2 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \\ \begin{pmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{bmatrix}. \tag{19}$$

Each position in the output 3 × 3 detector array corresponds to a unique exponent sum. Assigning exponent values as shown below

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 0 + 2 = 2,$$
$$1 + 0 = 1, \quad 1 + 1 = 2, \quad 1 + 2 = 3,$$
$$2 + 0 = 2, \quad 2 + 1 = 3, \quad 2 + 2 = 4,$$

in the final result leads to

$$\mathbf{y} = \begin{bmatrix} \begin{pmatrix} 2 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \\ \begin{pmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 2 \times 10^0 + 2 \times 10^1 \\ 4 \times 10^0 + 1 \times 10^2 \end{bmatrix}$$

$$= \begin{bmatrix} 22 \\ 104 \end{bmatrix} \tag{20}$$

as required. Clearly this extends readily to the matrix–matrix case.

V. Conclusions

Optical floating point calculations are readily achievable by spatial encoding. Like all the other improvements in number representation for optical computing (capability of representing real numbers, complex numbers, and digital numbers), the price that is paid is a loss in the throughput rate at which numbers are processed. As high throughput is one of the supposed advantages of optical computing, designers must exercise care in system design. Finally, we should note that on-the-fly scale adjustment can achieve many of the effects of floating point operation with no penalty in throughput but some penalty in complication.[7] Thus multiple solutions to the dynamic range problem are now available.

This work was sponsored under contract F19628-82-C-0068 from Rome Air Development Center, Hanscom AFB, Mass.

References

1. D. Psaltis, D. Casasent, D. Neft, and M. Carlotto, Proc. Soc. Photo-Opt. Instrum. Eng. 232, 151 (1980).
2. P. Guilfoyle, to be published in Opt. Eng. 23, (1984).
3. R. A. Athale, W. C. Collins, and P. D. Stilwell, Appl. Opt. 22, 368 (1983).
4. R. P. Bocker, S. R. Clayton, and K. Bromley, Appl. Opt. 22, 3149 (1983).
5. H. J. Caulfield, to be published in Opt. Eng. 22 (1983).
6. An insight offered by C. Verber; private communication (1983).
7. H. J. Caulfield and J. Gruninger, Opt. Lett. 8, 398 (1983).

APPENDIX L

THE MATRIX-MATRIX MULTIPLIER
DEVELOPED PARTIALLY UNDER THIS CONTRACT

Reprinted from **Applied Optics**, Vol. 22, page 504, March 15, 1983

Rapid unbiased bipolar incoherent calculator cube

R. P. Bocker, H. J. Caulfield, and K. Bromley

Presented in this paper is one of several possible electrooptical engagement array architectures for performing matrix–matrix multiplication using incoherent light. Essential components of this new signal-processing device include two dynamic light valves operating in a reflection mode, a 2-D photodetector array, and a single polarizing beam splitter.

I. Introduction

In this paper we present a new concept for performing the mathematical operation of matrix–matrix multiplication using electrooptical technology. This concept is based on the pioneering work of Kung[1] for performing matrix–matrix multiplication using an all-electronic systolic-array architecture. A novel feature of the electrooptical approach is that it is not limited to 2-D architectures as is the case when employing silicon technology in an electronic implementation. Before describing the electrooptical approach, let's briefly review prior work for performing both matrix–vector and matrix–matrix multiplication using optical techniques.

II. Background

The use of optical correlation techniques involving coherent light for performing matrix–matrix and matrix–vector multiplication has been extensively studied mathematically[2] and experimentally demonstrated for matrices of order 2.[3] This technique has the undesirable feature that, as the matrix order increases, the number of unwanted circular distributions of light appearing in the output plane of the processor rapidly escalates, thus reducing the light available at those positions corresponding to product matrix element information. In addition to this technique, there have been a number of other techniques investigated using incoherent light for performing matrix–vector multiplication. For example, preliminary studies in this area describe the computation of 1-D discrete Fourier transforms,[4] sine, cosine, and Walsh-Hadamard transforms as well as a variety of linear filtering operations.[5] The technical feasibility of this particular approach was demonstrated for matrices of order 32 using an optical device earlier developed[6] for performing correlation and convolution operations with incoherent light. In the original version of this optical correlator, a single light-emitting diode, photographic film transparency, mechanical scanning mirror, and a vidicon detector were employed. More recently,[7,8] the scanning mirror and vidicon detector were replaced by a solid-state area-array charge-coupled device, thus greatly reducing the size of the processor. Matrix–vector multiply operations involving matrices of order 128 are presently performed using this approach.

A second technique for computing matrix–vector products using incoherent light involves the use of a linear array of light-emitting diodes, an optical transparency, and a linear array of photodetectors.[9] This architecture has the advantage that the data vector information may be entered in parallel, thus allowing for higher throughput rates. The feasibility of this approach has been demonstrated for matrices of order 10. Combining this architecture with a 1-D adder in a feedback loop gives rise to an iterative electrooptical processor.[10] With this capability, it is possible to perform other higher-level matrix operations such as the solution of simultaneous algebraic equations, least-squares approximate solution of linear systems, matrix inversion, and eigensystem determination,[11,12] just to mention a few.

Most recently, much attention has been focused on implementing parallel-processing architectures for performing a variety of matrix operations using exclusively electronic components. Most noteworthy is the work of Kung on systolic-array architectures.[1,13,14] Combining VLSI/VHSIC technology with systolic-array processing techniques should give rise to increased signal-processing capabilities by at least a factor of 100.[15] Already a 2-D systolic-array testbed has been designed and fabricated for validating many of the proposed architectures and algorithms envisioned.[16] A

H. J. Caulfield is with Aerodyne Research Inc., 45 Manning Road, Billerica, Massachusetts 01821; the other authors are with U.S. Naval Ocean Systems Center, Signal Processing Technology Branch, San Diego, California [illegible]
Received [illegible] September [illegible]

similar all-electronics parallel approach has been proposed[17] using an engagement-array architecture. As it turns out, these new systolic/engagement types of architecture are not restricted to solely electronic implementations. For example, an acoustooptic approach using incoherent light for performing matrix–vector multiplication employing the systolic/engagement-array architecture has recently been described.[18] This acoustooptic processor uses a linear array of light-emitting diodes for inputting the matrix information, an acoustooptic traveling-wave modulator for inputting the vector information, and a linear-array charge-coupled device for computing the desired output vector information. This approach has the advantage that the input vector and matrix information may be entered in real time.

III. Preliminaries

To illustrate the concept of matrix–matrix multiplication using an optical engagement-array architecture, consider the case when the matrices involved have real-positive elements only and are of order 3. That is,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}, \qquad (1)$$

or, equivalently,

$$AB = C, \qquad (2)$$

where A and B are known input matrices, and C is the desired output matrix. Each element of matrix C is obtained by the equation

$$c_{ik} = \sum_{j=1}^{3} a_{ij} b_{jk} \qquad i,k = 1,2,3. \qquad (3)$$

The techniques presented here certainly apply to matrices of order >3. Order 3 matrices were chosen merely to illustrate easily the concepts involved. Shown in Fig. 1 is a 2-D array of photodetectors initially containing *zero* charge at each detector site, two optical transparencies encoded with the matrix A and B information, with each transparency capable of translating in front of the photodetector array as shown, and an incoherent light source providing a spatially uniform collimated light beam comprised of a time sequence of equal intensity pulses. Light propagation is from left to right. As seen in this figure, each optical transparency is partitioned into an array of rectangular-shaped resolution cells, some containing the matrix A and B information, the remaining being optically opaque. Those cells containing matrix information each have an intensity transmittance proportional to the magnitude of the corresponding matrix element located at that cell as depicted in Fig. 1. At any one instant in time, only a 3 × 3 array of resolution cells in each transparency is illuminated by a single light pulse of short time duration. The resulting spatially modulated light beam impinges on the photodetector array, whence photoelectric charge is generated and accumulated.

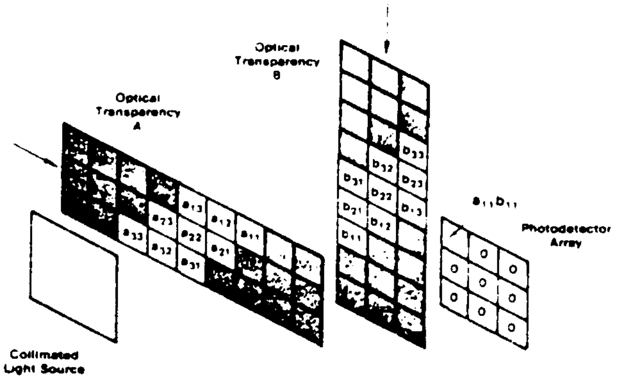


Fig. 1. Optical engagement matrix–matrix multiplier using sliding optical transparencies. (Initial State.)

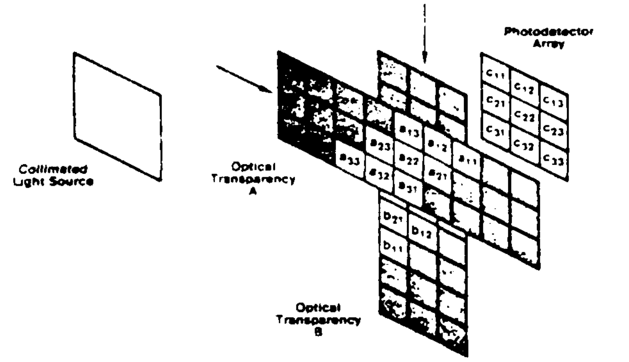


Fig. 2. Optical engagement matrix–matrix multiplier using sliding optical transparencies. (Final State.)

Initially the optical transparencies are so positioned that the first light pulse passing through the system passes through those 3 × 3 arrays containing only the a_{11} and b_{11} element information, respectively. The result is that only the photodetector in the upper-left corner of the detector array receives light. The amount of photoelectric charge generated at that particular detector is proportional to the product of a_{11} and b_{11}. Next, optical transparency *A* is shifted horizontally to the right one resolution cell width, and transparency *B* is shifted vertically downward one resolution cell height. At this point, the light source generates a second pulse of light identical to the first. Now the upper-left three photodetectors in the array each generate quantities of photoelectric charge proportional to the product of the transmittances of those resolution cells directly in front of each detector. This process continues in this manner until the optical transparencies have physically translated past the detector array as shown in Fig. 2. On closer examination, we find that at each photodetector element site there is a quantity of photoelectric charge which has accumulated that is proportional to each matrix element comprising the desired matrix C. This then represents a simple version of the engagement-array architecture for performing matrix–matrix multiplication using two optical transparencies which physically translate across the face of a fixed photodetector array.

IV. Proposed Electrooptical Configuration

The architecture just described for performing matrix–matrix multiplication using an optical engagement-array approach was primarily examined for the purpose of illustrating the basic concepts involved. Unfortunately, this architecture lacks the capability of updating or changing the input matrices A and B in a real-time manner. This is principally because most optical transparencies are made of photographic film. Of course, one way around this difficulty is through the use of light valves whose optical properties can be changed in real time by electronic means. That is, if we simply replace the translating optical transparencies by stationary light valves whose transmission characteristics can be changed and updated, matrix–matrix multiplication can be performed without the need for translating components.

A compact architecture based on these ideas is illustrated in Fig. 3. The basic components required for this system concept include a polarized incoherent collimated light source with the same properties as before, a polarizing beam splitter, two light valves operating in a reflection mode, and a 2-D array of photodetectors also with the same properties as before. Collimating and imaging optics may be required but are not shown here. The use of optical lens elements would certainly have to be employed when diffraction effects could not be ignored. The matrix A and B information are clocked into their respective light valves shown in Fig. 4. The transferring of the matrix data within the light valves using this architecture is analogous in all respects to the physical translating of the optical transparencies as previously described. Again, the desired matrix C information is generated within the photodetector array, where it may be clocked out at a later time.

The reason for using a polarizing beam splitter in this architecture is to eliminate light from propagating directly from the light source to the photodetector array without first reflecting from each of the two light valves. Of course, for this to be true, the incoherent light source must be polarized as noted earlier. If the light valves were to behave, for example, as reflecting mirrors, one type of polarizing beam-splitter arrangement which could be employed is shown in Fig. 5. The polarizing beam splitter would be of the Glan prism variety.[19] In addition, an input linear polarizer and two quarterwave plates would also be required. It is noted that the exact electrooptical configuration used for performing the matrix-matrix multiply operation will be highly dependent on the nature of the particular light valves employed. Light-emitting diodes or laser diodes appear most attractive as the incoherent light source. The photodetector array could be an array of photodiodes or possibly a photoactivated charge-coupled device.

For the architecture described herein, it has been assumed for the sake of simplicity that the elements of the matrices A, B, and C were real and positive only. The issue of performing matrix operations involving matrices and vectors whose elements are bipolar or even complex using incoherent light has previously been addressed.[4,9,10] These techniques, therefore, could

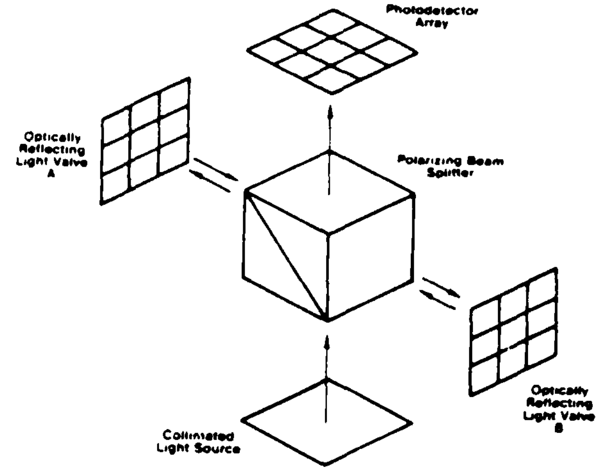

Photodetector Array
Optically Reflecting Light Valve A
Polarizing Beam Splitter
Optically Reflecting Light Valve B
Collimated Light Source

Fig. 3. Key components of a solid-state optical engagement array matrix–matrix multiplier.

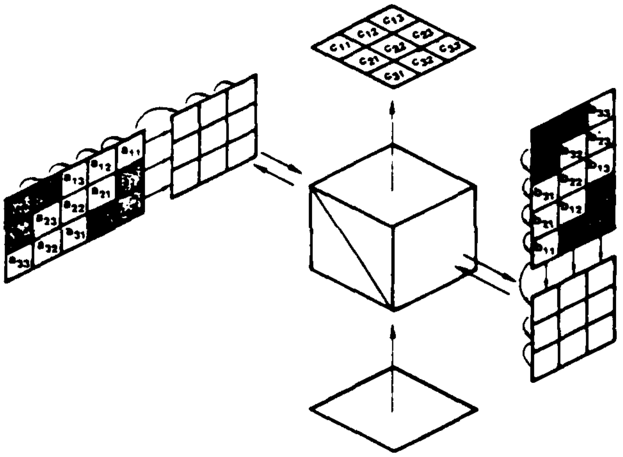


Fig. 4. Data handling in the optical engagement array processor.

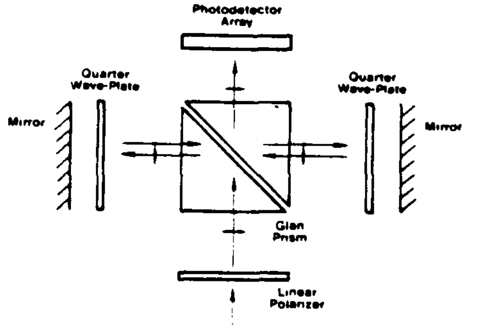


Fig. 5. Polarizing beam splitter with support optics.

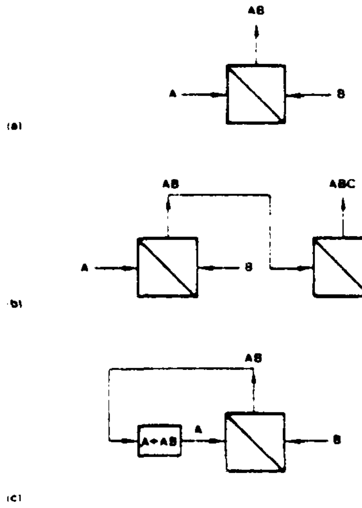


Fig. 6 Architectures for performing (a) basic matrix–matrix multiplication AB, (b) the matrix operation ABC, (c) iterative processing using feedback.

easily be extended to include this architecture as well. Since the mathematical operation of matrix–matrix multiplication is so fundamental to a number of higher-order matrix operations, this basic architecture could serve as a modular building block for these higher-order operations. The basic matrix–matrix multiply operation using the processing cube structure presented in this paper is symbolically represented by the diagram in Fig. 6(a). Again, A and B are the input matrices, and AB is the desired output matrix. If it was important to perform the multiplication of three matrices, that is,

$$ABC = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}, \quad (4)$$

two processing cubes could be connected in a serial manner as depicted in Fig. 6(b). It should be noted here that the AB data must be both spatially and temporally reformatted between the two cubes employed for this algorithm to work. The product of three matrices would be useful for image-processing type applications. One example would be computing the 2-D discrete Fourier transform of an array of pictorial information. Matrix B would contain sampled values of the image, matrices A and C would contain the discrete Fourier transform kernel information, and matrix ABC would yield the desired discrete Fourier transform. The architecture depicted in Fig. 6(c) would be useful in those areas (for example, using iterative processing requiring feedback. The expression A ← AB as seen in Fig. 6c means A is replaced by the matrix product of A and B. As previously mentioned, the solutions of simultaneous equations, matrix inversion, and eigensystem determination are representative of higher-order operations which can be performed using iterative processing.

V. Summary

This paper has presented the basic concept of a rapid unbiased bipolar incoherent calculator cube (RUBIC cube) for performing matrix-matrix multiplication using an optical engagement-array architecture. Future work will address the implementation of this architecture.

References

1. C. Mead and L. Conway, *Introduction to VLSI Systems* (Addison-Wesley, Reading, Mass., 1980), pp. 271–292.
2. R. A. Heinz, J. O. Artman, and S. H. Lee, Appl. Opt. **9**, 2161 (1970).
3. D. P. Jablonowski, R. A. Heinz, and J. O. Artman, Appl. Opt. **11**, 174 (1972).
4. R. P. Bocker, Appl. Opt. **13**, 1670 (1974).
5. R. P. Bocker, "Optical Matrix–Vector Multiplication and Two-Channel Processing With Photodichroic Crystals," Ph.D. dissertation, University of Arizona, Tucson (1975) (U Microfilms 75-26 925).
6. K. Bromley, Opt. Acta **21**, 35 (1974).
7. M. A. Monahan, R. P. Bocker, K. Bromley, and A. Louie, in *Digest of the International Optical Computing Conference Digest*, IEEE Catalog 75-CH0941-5C (IEEE, New York, 1975).
8. M. A. Monahan, K. Bromley, and R. P. Bocker, Proc. IEEE **65**, 121 (1977).
9. J. W. Goodman, A. R. Dias, and L. M. Woody, Opt. Lett. **2**, 1 (1978).
10. D. Psaltis, D. Casasent, and M. Carlotto, Opt. Lett. **4**, 348 (1979).
11. H. J. Caulfield, D. Dvore, J. W. Goodman, and W. T. Rhodes, Appl. Opt. **20**, 2263 (1981).
12. M. Carlotto and D. Casasent, Appl. Opt. **21**, 147 (1982).
13. H. T. Kung, Proc. Soc. Photo-Opt. Instrum. Eng. **241**, 76 (1980).
14. H. T. Kung, Computer **15**, 37 (1982).
15. J. J. Symanski, Proc. Soc. Photo-Opt. Instrum. Eng. **298**, 27 (1981).
16. J. J. Symanski, Proc. Soc. Photo-Opt. Instrum. Eng. **341**, 2 (1982).
17. J. M. Speiser and H. J. Whitehouse, Proc. Soc. Photo-Opt. Instrum. Eng. **298**, 2 (1981).
18. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz, Opt. Commun. **40**, 86 (1981).
19. G. R. Fowles, *Introduction to Modern Optics* (Holt, Rinehart & Winston, New York, 1968), pp. 182–182.

APPENDIX M

SPATIAL LIGHT MODULATOR DESIGN FOR THE MATRIX-MATRIX MULTIPLIER OF APPENDIX L

Optical Information Processing for Aerospace Applications II

Compiled by Robert L. Stermer
Langley Research Center

Proceedings of a NASA conference
held at Langley Research Center
Hampton, Virginia
August 30-31, 1983

NASA
National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1984

THE APPLICATIONS OF SILICON LIQUID CRYSTAL LIGHT VALVES TO OPTICAL DATA PROCESSING: A REVIEW

U. Efron and B. H. Soffer
Hughes Research Laboratories
Malibu, CA 93065

and

H. J. Caulfield
Aerodyne Research, Inc.
The Research Center at Manning Park
Billerica, MA 01821

ABSTRACT

The applications of the photo-activated, the CCD-addressed, and the variable-grating mode liquid crystal light valves (LCLVs) to optical data processing are described. These applications include image correlation, level slicing, spectral analysis and correlation, bi-spectral image division, and matrix-matrix multiplication.

INTRODUCTION

Coherent optical data processing (CODP) (ref. 1) offers many potential advantages in image processing as well as in the processing of wide bandwidth electrical signals which are amenable to two-dimensional (2-D) form. One of the main limitations of this technology has been the lack of a fast, high-resolution, real-time spatial light modulator (SLM) (refs. 2, 3). These devices impose, on a coherent optical beam, a 2-D image that is derived from either an incoherent optical source (photoactivated SLM) or directly from a properly formatted electrical input signal (electronically addressed SLM). While the first of these tasks can be accomplished with the photoactivated hybrid field-effect mode (HYFEM) liquid crystal light valve (LCLV) (ref. 4), the second can be implemented by the use of the charge-coupled device (CCD)-addressed LCLV (ref. 5).

The first generation, CdS-based photoactivated device is already in production at Hughes. A second-generation, fast-response silicon photoconductor-based device is currently under development at Hughes Research Laboratories. These types of devices operate in conjunction with an optical input source, such as a CRT or a laser scanner to provide a real-time coherent output image (ref. 6). The novel, photoactivated silicon LCLV (Figure 1) with its high-broadband input sensitivity may also be used for direct imaging of the scene and subsequent image processing (e.g., for robotics).

In CODP applications (such as radar signal processing or real-time matched filters), it is desirable to convert the electrical input directly to an optical output image without the intermediate step of first converting to an input image via a CRT. To realize this function, we have designed and developed a novel type of CODP inputting device that uses a CCD array to serially load and store a full frame of analog electrical information which is subsequently transferred in parallel to a liquid crystal (LC) layer (Figure 2). The elimination of the CRT (or equivalent process) from the ODP system greatly simplifies the system; in particular, it eliminates several of the drawbacks associated with it, such as geometrical distortions, stability, and jitter. This device can be used with both coherent and incoherent readout sources, extending in spectral range from the near ultraviolet to the near infrared.

In the following section, some applications of both the silicon photoactivated LCLV and the electronically addressed CCD-LCLV to ODP will be described. These applications include image correlation and level slicing, spectral analysis and correlation, bi-spectral image division, and matrix-matrix multiplication.

OPTICAL PROCESSING APPLICATIONS OF THE SILICON LIGHT VALVES

Image Correlation and Level Slicing

Optical data processing is applicable in two main categories of data processing: the processing of wideband serial signals, and in 2-D or image processing. The photoactivated device is most effectively used in image processing applications, while the CCD-addressed spatial light modulator can be used in both of these categories.

One example of image processing is that of correlating an image with a reference pattern, as shown in Figure 3. Here the images analyzed, A(t) (in video form), and the reference image, B(t), are correlated using a joint-transform technique (ref. 7). The two CCD-SLMs are used as the electro-optic transducers to generate real-time coherent optical images in which amplitudes are superimposed in the Fourier plane. The intensity at the input to the photoactivated device contains, among other terms, the multiplied amplitudes of the two Fourier-transformed images. The photoactivated LCLV is then used to retransform the multiplication image, resulting in the correlation required.

An important application of the photoactivated silicon LCLV is direct-scene imagery followed by coherent processing. This function is required, e.g., in robot vision systems. Here, one can utilize the two important features of the silicon device: (1) its broadband sensitivity (400 to 1,100 nm, with typically 50 μW/cm^2 at 540 nm); and (2) its fast time response, permitting fast scenery changes to be processed. In the configuration shown in Figure 4, the input scene is imaged and converted to coherent modulation using the Si-LCLV, and is subsequently correlated with a matched pattern using the CCD-LCLV as a programmable matched filter.

The use of the silicon photoactivated device for such direct image processing further permits the dual-frequency mode of the liquid crystal activation to be applied (ref. 8). This may result in cutting the response time from the current 16 ms to 1 to 2 ms.

Another powerful application of optical processing is with the use of a special photoactivated device: the variable grating mode (VGM) SLM (ref. 9). The device is based on the formation of grating-type regions in the LC, the spatial frequency of which is determined by the voltage drop across the LC. Since a very high-impedance photoconductor is required for this light valve, the silicon-MOS configuration is a potential candidate.

A useful application of the device is intensity-to-spatial frequency conversion, shown in Figure 5. Here, the device is used to level slice an input image (shown in three levels: I_1, I_2, I_2). Filtering at the frequency plane with $F = F_2$ (corresponding to $I = I_2$) results in the generation of the $I = I_2$ level of the input image at the output plane.

Large Time-Bandwidth Spectrum Analyzer

We have demonstrated a real-time rf spectrum analyzer with an extraordinarily high resolution and time-bandwidth product using the LCLV, with resolution $<10^2$ Hz. The scheme of the apparatus is shown in Figure 6. The rf signals were amplified and displayed in raster fashion on a CRT. The signals were obviously asynchronous with the raster scan of $\omega_s = 20 \times 10^3$ sec^{-1} and a frame time of 7×10^{-2} sec. The incoherent optical display was focused on the photoconductive input of the LCLV which acted as a coherent-to-incoherent transformer as the output of the LCLV was illuminated with a coherent HeNe laser. This transformation permitted an optical Fourier transformation to be performed. It is well known that the Fourier transform of a raster pattern in time is a raster pattern in frequency, as shown in Figure 6. Low-frequency, Morse-coded tone-modulated rf signals from oil field transmitters displayed the simple textbook A.M. spectral pattern of a carrier and two pulsating sidebands. More complex modulations were also evident in the display. The theoretical resolution is given by the ratio of ω_s to the number of lines, which with $N = 1.4 \times 10^3$ lines is 14 Hz. Because of the falloff in resolution of the LCLV and associated optics, the resolution achieved was somewhat less (80 Hz). An obvious improvement of this system will be the replacement of the CRT-imaging lens with a CCD-addressed LCLV.

In this case, the ultimate, 1,000 array CCD-LCLV would provide 10^6 point resolution over 100 MHz bandwidth at (real-time) frame rates of 100 Hz. Comparable performance, taking into account size and power requirements, will not be achievable by even the most advanced digital technology currently in development (i.e., VHSIC).

A Real-Time Spectrum Analyzer/Correlator

Another important application is real-time spectrum analysis of a given scene. A silicon light valve-based system that can perform this operation is shown in Figure 7. The operation of this system is described below.

The radiation from the scene to be analyzed, I(W), is split by the beam splitter in a Michelson interferometer configuration. Two mirrors, a standard one and a staircase one, are used. The interference pattern at the output of the interferometer (i.e., at the input to the LCLV) is the (spatial) Fourier transform of the input spectrum. This is analogous to a conventional Fourier transform spectrometer (FTS) (ref. 10), in that each of the staircase steps represents one mirror location in a moving mirror spectrometer. The subsequent spatial Fourier transform of the output of the light valve results in the spectral analysis of the input beam at the imaging array. Figure 7 shows the operation of the spectral correlator. The readout laser beam is spatially modulated by the Fourier transformed reference spectrum using the CCD light valve. This modulated beam is then used as a readout light for the photoactivated light valve. At the input of the photoactivated light valve, the spatial interferogram of the input beam is present. The emerging output beam consists of a multiplication of the input and the reference, Fourier-transformed spectra presented by the CCD-LCLV. The subsequent inverse Fourier transformation carried out by the lens results in the appearance of correlation and convolution terms of the two spectra at the imaging array. This system, which is based on the FTS principle, benefits from two important advantages of the FTS system, namely, the multiplexing, or the Felgett's advantage in signal-to-noise ratio, and the throughput, or the Jaquinot's advantage.

An attractive feature of this system is that it can be used for pattern recognition purposes with a flip of a mirror. In this way, the pattern of the incoming beam, rather than its spectral content, can now be analyzed and correlated with a suitable reference image presented by the CCD light valve, as in Figure 4. The system can thus perform both spectral and pattern correlations of the scene.

The spectral range of this system is limited by the photoactivated light valve since it must be sensitive in the spectral range used. The existing silicon light valve enables us to use the 400-nm to 1,200-nm range. Since the detection of longer wavelengths may require cooling of the light valve, the LC will be the limiting component for such a longer wavelength light modulator. It is estimated that operation up to 3 μm can be achieved using LC operating at low temperatures. Possible photoconductor candidates for such IR light valves are Ge, InAs, InSb, or extrinsic silicon, depending on the cutoff wavelength required.

The spectral resolution largely depends on the manufacturing of the staircase mirror. One could conceive more than 10,000 elements of resolution. It should be pointed out that for the photoactivated and CCD-addressed light valves, a resolution on the order of 10^6 elements is possible.

One obvious limitation for the application above is the intensity of the input beam, or the radiation level from the scene analyzed. Using the silicon light valve, a rough estimate for the input illumination level required is 100 $\mu W/cm^2$ in the visible spectral region. Projected performance of such a correlator for two spectral regions is presented in Table 1. Finally, it should be pointed out that other, possibly more efficient methods of self-interference of the incoming analyzed beam have been previously suggested (ref. 11).

A particularly important type of signal processing in which the CCD-LCLV may be used is radar signal processing. This field encompasses ambiguity-function generation and synthetic aperture radar (SAR) processing.

An ambiguity-function generation system using two LCLVs was previously described (ref. 12). The replacement of the photoactivated LCLV by a CCD-addressed LCLV will significantly improve the system, elimating the CRT and the acousto-optic units required.

The Bi-Spectral Imaging/Image Division System

Another potential application of the Si-LCLV for combined spectral and scene analysis is the Bi-Spectral Imaging/Image Division System. The purpose of this system is to obtain the (logarithmic) image of the intensity ratio of the scene at two wavelengths in the 400-nm to 1100-nm spectral range. This operation results in the enhancement of specific textures in the scene. Thus, it has applications in texture recognition such as the remote Earth-features identification system currently under development by NASA (ref. 13). The schematics of the Si-LCLV-based system are shown in Figure 8. The operation is as follows. The scene imaged by the input optics is split into two channels which are each wavelength filtered in the two spectral regions (λ_1, λ_2) required (400 nm < λ_1, λ_2 < 1100 nm). Then the filtered images are spatially modulated by logarithmic halftone screens with different spatial frequency for each channel, λ_1-F_1 and λ_2-F_2. A variable attenuation compensator placed at one of the channels acts to compensate for intensity imbalance between the two channels. The two images, each modulated by a different spatial carrier, are then recombined at the input to the silicon liquid crystal light valve. Thus, each of the two images at the two different wavelengths is "tagged" with a different spatial frequency modulation. The photoactivated silicon liquid crystal light valve acts as a sensitive, broadband, incoherent-to-coherent image converter. A spatial Fourier transform is then performed on the data readout by the laser beam. The diffractions of the two wavelength images will now appear separately in the Fourier plane, due to the different spatial carriers for each of those images. Spatial filters corresponding to each of the two halftone screens are placed at the appropriate locations in the Fourier plane. This results in the formation of logarithmic intensity images following a retransforming lens (ref. 14). A 180° phase retardation plate placed at one of the filter locations will result in one of the logarithmic images (λ_2) having a reversed phase with respect to the other. Thus, the amplitude of this image formed at the video detector plane will be proportional to

$$A_{out} = A_1(x,y) + A_2(x,y) \propto \log I_1(x,y) - \log I_2(x,y) = \log \left[I_2/I_2\right]$$

where $I_1(x,y)$ and $I_2(x,y)$ are the intensities of the input images at λ_1 and λ_2, respectively. The image amplitude following reconstruction at the vidicon input will be proportional to log $[I(\lambda_2)/I(\lambda_1)]$, i.e., to the (logarithmic) ratio of the images at λ_1 and λ_2. Due to the high sensitivity of the silicon photoconductor in the silicon light valve configuration (about 40 $\mu W/cm^2$), the imaging system is expected to have sufficient sensitivity for direct imaging of Sun-illuminated scenes.

It should be noted that the same physical region of the light valve is utilized in both channels. This is done in order to minimize non-uniformities in the ratio image obtained by the wavefront substraction. Thus, non-uniformities associated with amplitude or phase defects originating in the light valve will be automatically substrated. The "penalty", however, is the need to use two different spatial frequencies, reducing the bandwidth available for image information.

The Spectral Range of the bi-spectral imaging/image division system is limited by the silicon LCLV (400 nm to 1100 nm). As indicated above, it may be possible to extend the spectral range of the silicon device into the 3- to 5-μm region.

The Dynamic Range of this system is limited by the Si-LCLV, which is typically 100:1. An important advantage of this optical processing system is that the output ratio is presented by a coherent light. This enables a straightforward use of optical post-processing (e.g., ratio image correlation).

The Spatial Resolution of this system depends on the spatial frequencies employed, as well as on the Si-LCLV performance. Taking Fo = 25 cycles/mm at 30%

modulation as the current performance of the Si-LCLV, and using the two carrier frequencies, as: Fo/4 and 3Fo/4, it is found that over 500 pixels of resolution are available using the 43-mm aperture device, with ΔF = Fo/2.

Application of the CCD-LCLV to Systolic Array Processing

Optical numerical processing offers a unique application of CCD-addressed LCLVs. For high speed, an optical numerical processor must utilize spatial parallelism. A two-dimensional data array offers great parallelism but can entail significant addressing problem. If, however, data could be entered a line at a tim and be made to march across the LCLV at the chosen clock rate, a single N × 1 CCD line could address a full N × N data array. The use of moving electronic data in a plane for such numerical operations was popularized as "systolic array processing" by Kung (ref. 15). The first extension of systolic array processing to the optical domain used one-dimensional transducers (acousto-optic delay lines and CCD detectors) in direct analogy with VLSI transducers (ref. 16). Recently, Bocker et al. (ref. 17) proposed the use of optics for systolic array processing in three dimensions, which electronics cannot do. Their Rapid Unbiased Incoherent Calculato cube (or RUBIC cube) uses two electronically addressed spatial light modulators to move components of matrices A and B across the spatial light modulator at certain clock rates. One possible configuration is shown in Figure 9. Because two pixels are needed for real-number representation, we can multiply the two (N/2) × (N/2) matrices together with the RUBIC cube in (N-1) clock periods. The cube's ability t multiply very large matrices very rapidly with low power consumption should make th RUBIC cube very important. To use the CCD-addressed LCLV for the RUBIC cube, one must use an external buffer memory which will feed the CCD-LCLV with one line/colum displacement in each frame. Alternatively, it may be possible to modify the structure of the CCD-LCLV to incorporate an internal buffer memory. This will enable the line/column clocking operation required. This possibility, although not a simple t sk, may also be desirable for other applications of the CCD-SLM.

REFERENCES

1. D. Casasent: Coherent Optical Pattern Recognition, Proc. IEEE 67, 813 (1979).

2. D. Casasent: Spatial Light Modulators, Proc. IEEE 65, 143 (1977).

3. D. Casasent: Performance Evaluation of Spatial Light Modulators, Appl. Optics 18, 2445 (1979).

4. J. Grinberg et al.: A New Real-Time Noncoherent-to-Coherent Light Image Converter. Opt. Eng. 14, 217 (1975).

5. M. J. Little et al.: CCD-Addressed Liquid Crystal Light Valve, Soc. Info. Display 1982 Tech. Digest, 250 (1982).

6. U. Efron, P. O. Braatz, M. J. Little, R. N. Schwartz, and J. Grinberg: Silicon Liquid Crystal Light Valves: Status and Issues, Opt. Eng. 22, 682 (1983).

7. J. E. Rau: Detection of Differences in Real Distributions, J. Opt. Soc. Am. 56, 1490 (1966).

8. C. S. Bak et al.: Fast Decay in a Twisted Nematic Induced by Frequency Switching, J. Appl. Phys. 46, 1 (1975).

9. B. H. Soffer et al.: Variable Grating Mode Liquid Crystal Device for Optical Processing, Proc. SPIE, Devices and Systems for Optical Signal Processing 218, 81 (1980).

10. P. R. Griffith: Chemical Infrared Fourier Transform Spectroscopy (J. Wiley and Sons, 1975), Chapter 1.

11. H. J. Caulfield: Holographic Spectroscopy, in Advances in Holography, Vol. 2, N. H. Farhat, ed. (M. Dekker, Inc., N.Y., 1976), p. 151.

12. W. P. Bleha et al.: Applications of the Liquid Crystal Light Valve to Real-Time Optical Data Processing, Opt. Eng. 17, 371 (1978).

13. R. Gale Wilson and W. Eugene Sivertson, Jr.: Earth Feature Identification and Tracking Technology Development, Proc. SPIE, Smart Sensors 178, 185 (1979).

14. S. R. Dashiel and A. A. Sawchuk: Nonlinear Optical Processing: Analysis and Synthesis, Appl. Opt. 16, 1009 (1977).

15. H. T. Kung: Special-Purpose Devices for Signal and Image Processing: an Opportunity in Very Large Scale Integration (VLSI), Proc. SPIE 241, 76 (1980).

16. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz: Optical Implementation of Systolic Array Processing, Opt. Commun. 40, 86 (December 1981).

17. R. P. Bocker, H. J. Caulfield, and K. Bromley: Rapid Unbiased Bipolar Incoherent Calculator Cube, Appl. Opt. 22, 804 (1983).

TABLE 1.- PROJECTED SPECIFICATIONS OF THE Si-LCLV-BASED FOURIER TRANSFORM SPECTROPHOTOMETER/CORRELATOR

1. VISIBLE RANGE: $400 \text{ nm} < \lambda < 1200 \text{ nm}$

   BANDWIDTH: $\Delta f = 16{,}700 \text{ cm}^{-1}$

   NO. OF RESOLUTION ELEMENTS: $N = 100 \times 100$

   SPECTRAL RESOLUTION: $\delta f = 1.67 \text{ cm}^{-1}$

   MAXIMUM "STROKE": $\delta D_{MAX} = 1/\delta f = 0.6 \text{ cm}$

   "ROUGH" STEPS: $\delta D_X = 0.6 \text{ cm}/100 = 60 \ \mu\text{m}$

   "FINE" STEPS: $\delta D_Y = 60 \ \mu\text{m}/100 = 0.6 \ \mu\text{m}$

2. 1.5-μm TO 4.5-μm REGION

   BANDWIDTH: $\Delta f = 4440 \text{ cm}^{-1}$

   NO. OF RESOLUTION ELEMENTS: $N = 100 \times 100$

   SPECTRAL RESOLUTION: $\delta f = 0.44 \text{ cm}^{-1}$

   MAXIMUM "STROKE": $\delta D_{MAX} = 1/0.44 = 2.27 \text{ cm}$

   "ROUGH" STEPS: $\delta D_X = 227 \ \mu\text{m}$

   "FINE" STEPS: $\delta D_Y = 2.27 \ \mu\text{m}$

STEPS DIMENSION (BOTH CASES) $\approx 0.5 \text{ mm} \times 0.5 \text{ mm}$ FOR 50-mm APERTURE

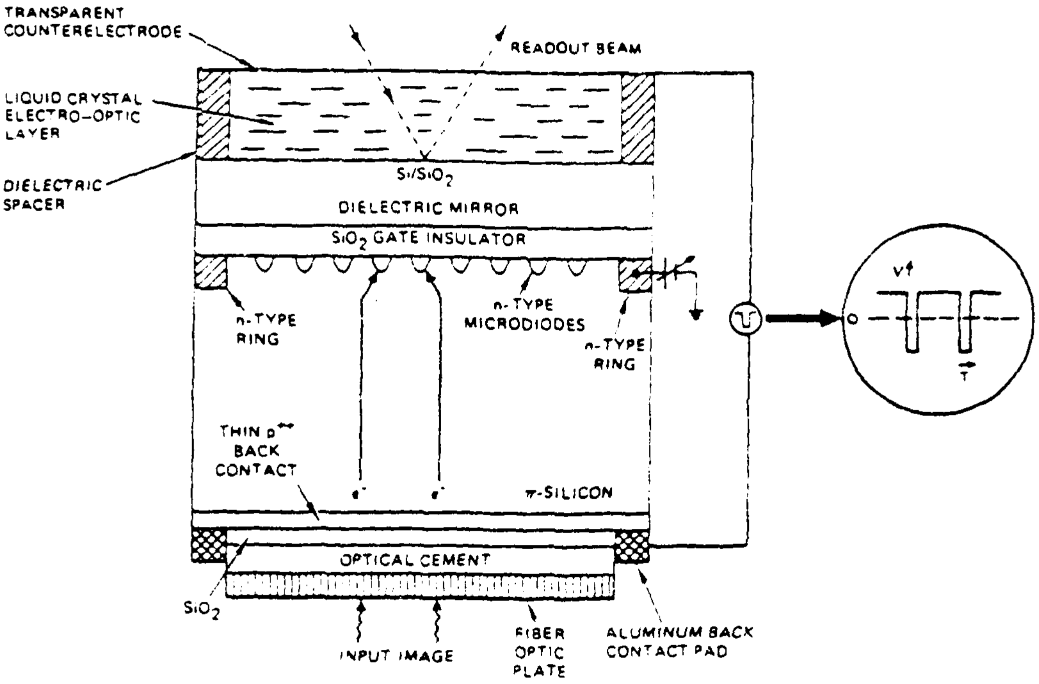


Figure 1.- A cross section of the photoactivated silicon liquid crystal light valve.

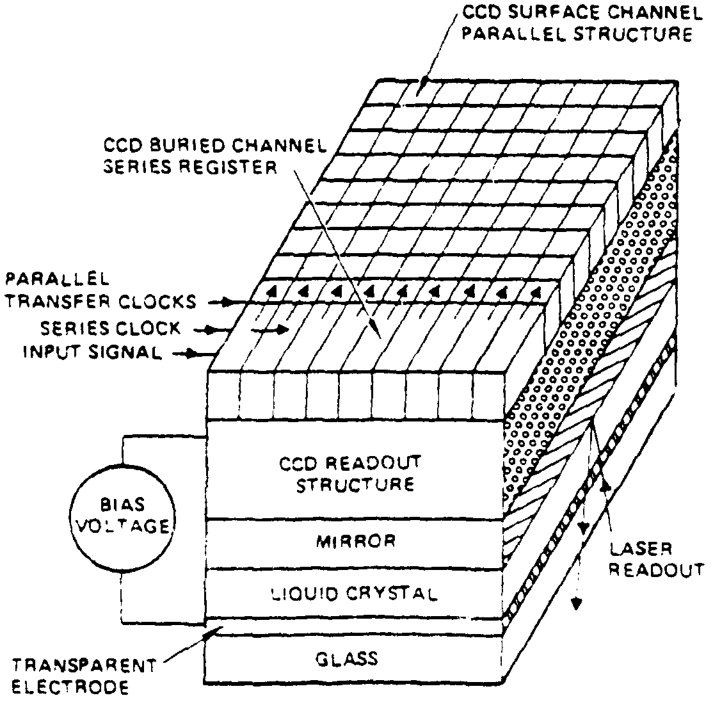


Figure 2.- Structure of the CCD-addressed liquid crystal light valve.

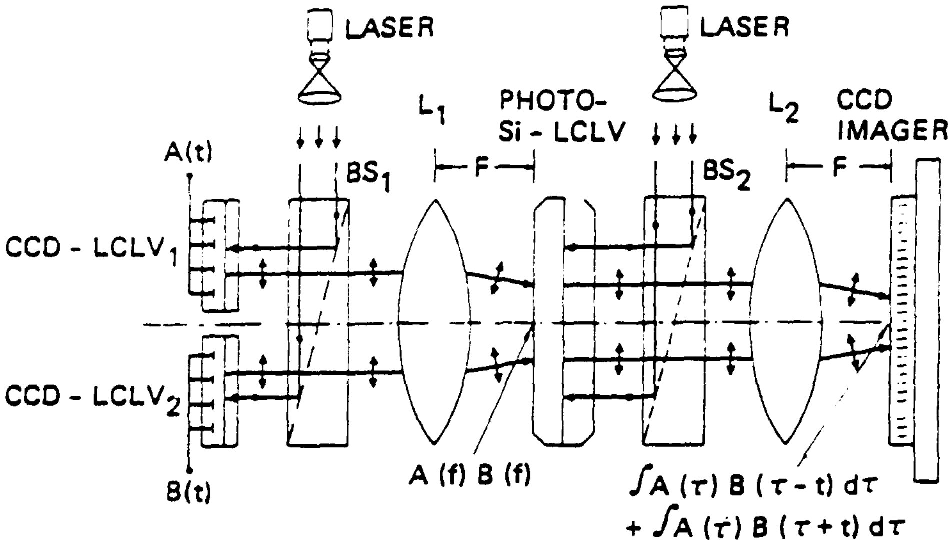


Figure 3.- A joint transform-based image correlation system using CCD-addressed and photoactivated devices.

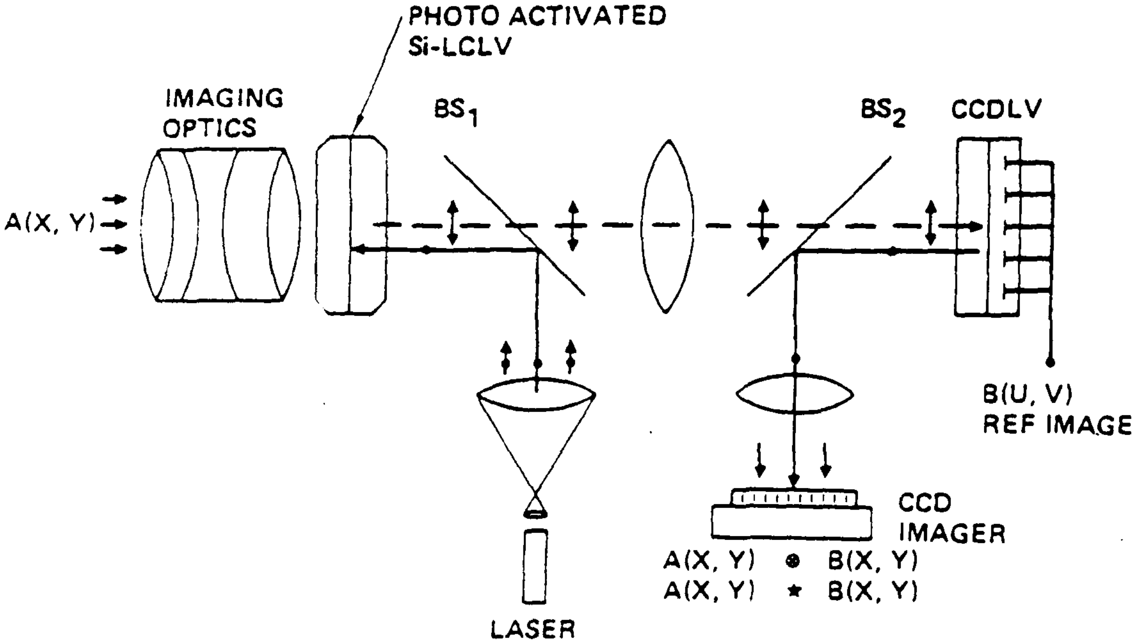


Figure 4.- An imaging/scene correlation system using the silicon liquid crystal light valves.

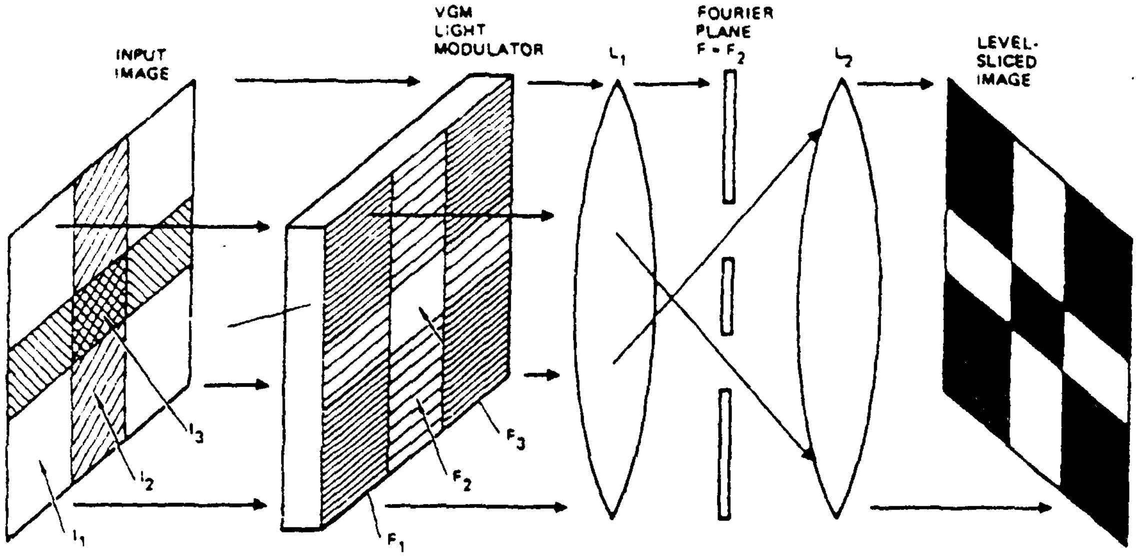


Figure 5.- Intensity level slicing of an image using the VGM modulator. The $I = I_2$ level is reproduced at the output.

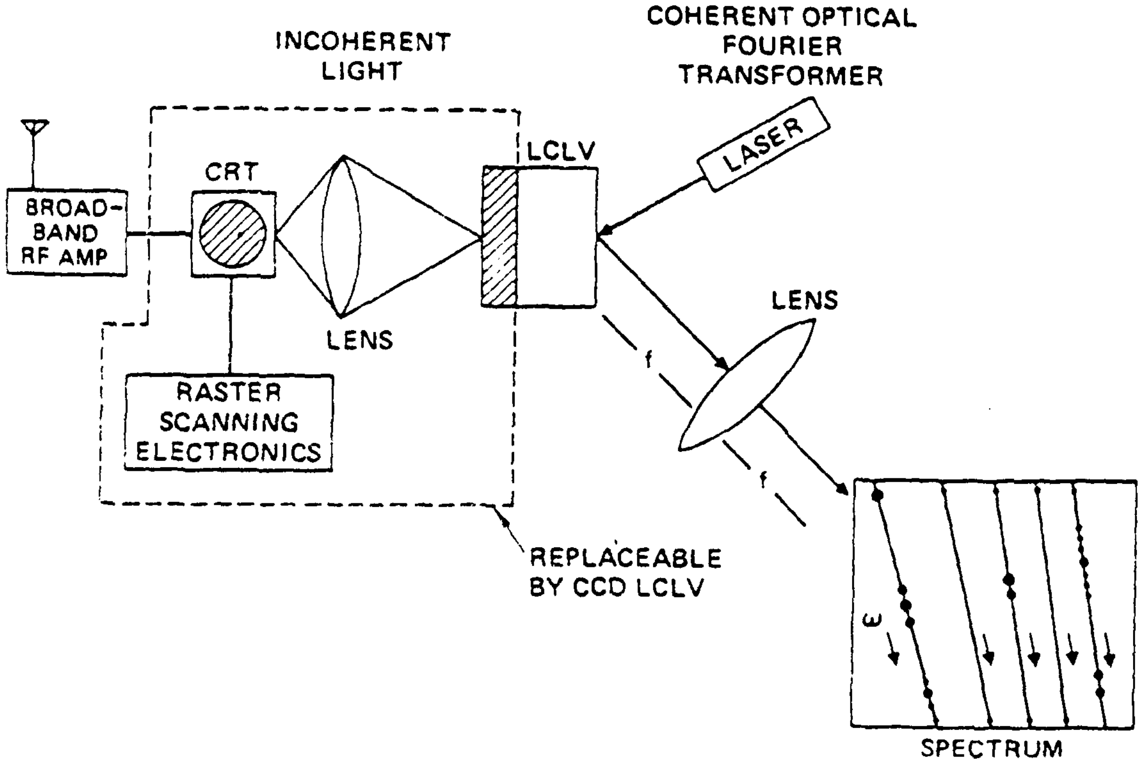


Figure 6.- Real-time, large time-bandwidth spectrum analysis.

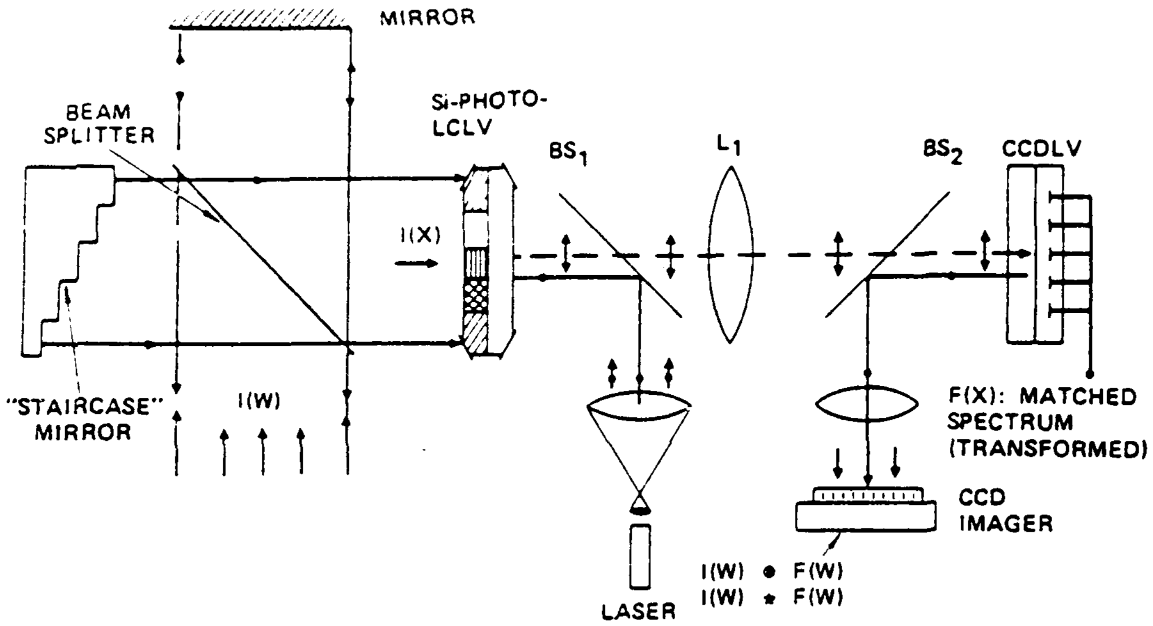


Figure 7.- A correlation system using the silicon LCLV.

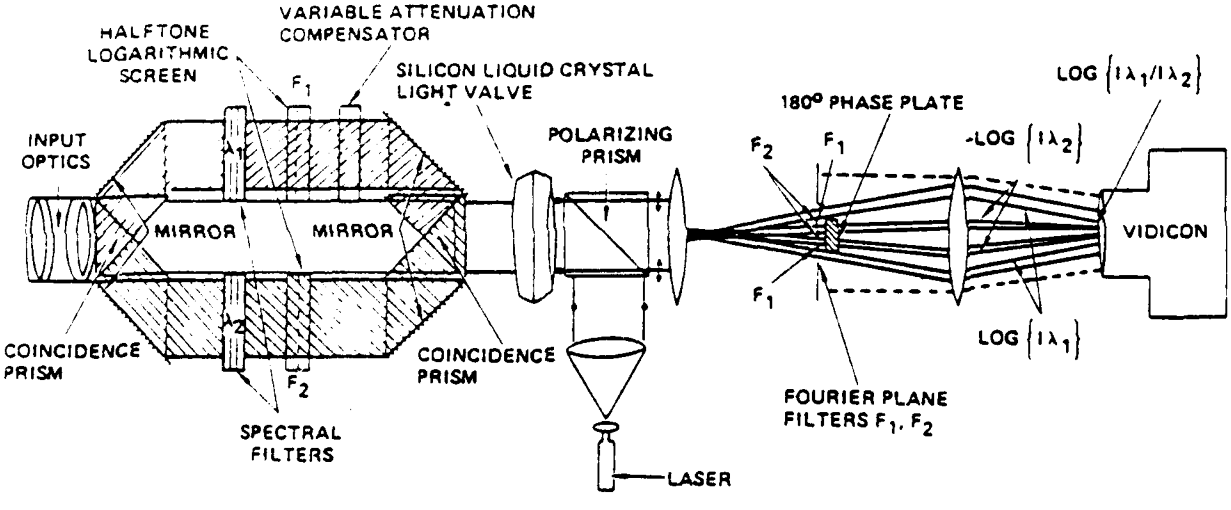


Figure 8.- A bi-spectral imaging/image division system based on the silicon LCLV.

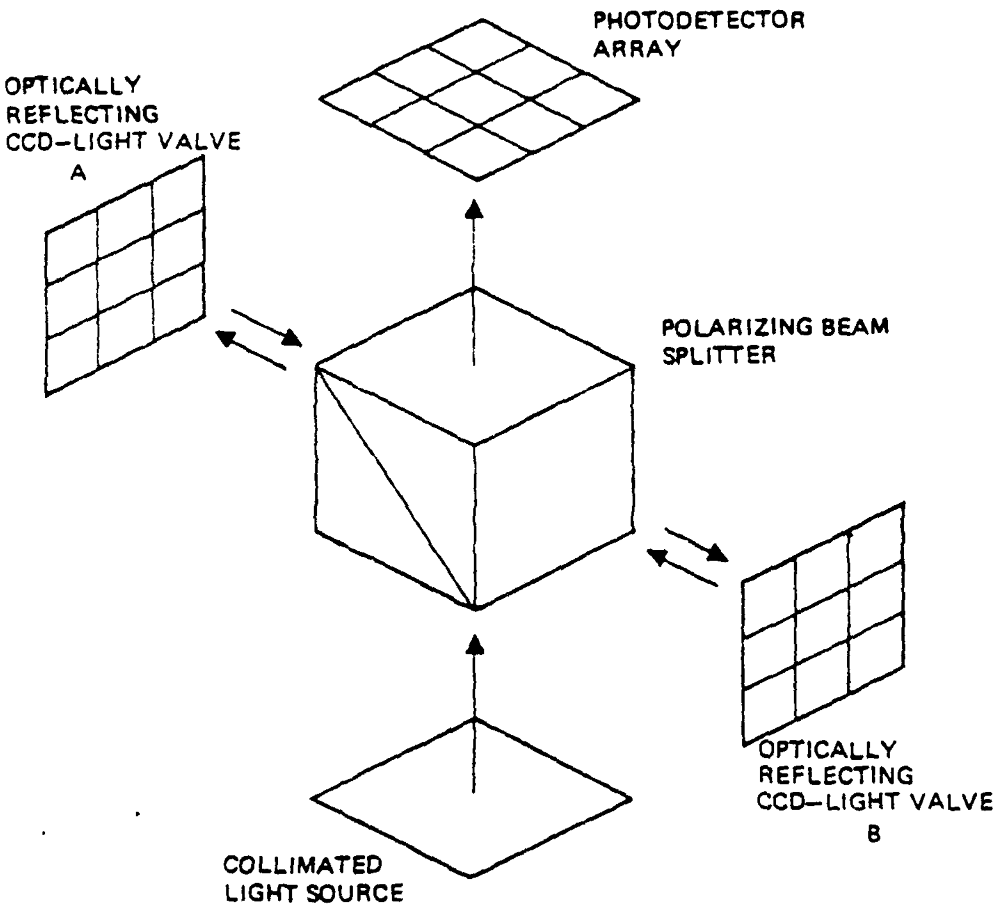

PHOTODETECTOR ARRAY
OPTICALLY REFLECTING CCD–LIGHT VALVE A
POLARIZING BEAM SPLITTER
OPTICALLY REFLECTING CCD–LIGHT VALVE B
COLLIMATED LIGHT SOURCE

Figure 9.- The application of the CCD-addressed LCLV's in a "RUBIC cube" system.

APPENDIX N

SPECIAL CONSIDERATIONS FOR OPERATING OPTICAL ITERATIVE CALCULATIONS

Reprinted from **Optics Letters**, Vol. 8, page 398, July, 1983

Feedback methods for optical systolic and engagement matrix processors

H. J. Caulfield and John Gruninger

Aerodyne Research, Inc., 45 Manning Road, The Research Center at Manning Park, Billerica, Massachusetts 01821

Received February 25, 1983; revised manuscript received April 18, 1983

The matching of the feedback circuitry to the optical systolic or engagement processor permits simple pipelining of stationary iterative algorithms as well as on-the-fly scale adjustment similar in effect to floating-point calculation.

Once a suitable stationary iterative algorithm is chosen, an optical systolic or engagement matrix algebra processor can be used to perform a useful operation, such as solution of N linear equations with N unknowns, singular value decomposition, and eigen problems. Most past work has been either on algorithms[1–5] or on processors.[6–9] Here we seek to complete the analysis by showing how the processor and feedback circuitry can be combined to achieve pipelined iterative systolic processing (i.e., a feedback data flow so matched to the processor input/output data flow that no slowdown of the processor is required). That is, the feedback electronics that implements the iterations must be matched to the matrix processor. We show that such electronics can also adjust the scale of the problem during each cycle in such a way as to assure optimum use of the dynamic range of the system. Thus optical processors can operate in a numerical mode that is neither integer (fixed point) nor floating point but much closer to the latter and much more useful than the former.

We consider only matrix–vector processors in detail, but extension to matrix–matrix processors is straightforward. In a systolic or engagement processor for the equation

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

the $\mathbf{x}$ components are input sequentially and (after a certain loading time of the pipeline) the $\mathbf{b}$ components are output sequentially. For a full unbanded optical engagement matrix–vector processor, the first $\mathbf{b}$ component is completed during the same clock period in which the last $\mathbf{x}$ component is entered. If there is to be no slowdown, the first component of the new iteration of $\mathbf{x}$ must be calculated during the same clock time. Thus the iteration must be of a form such as

$$x^{(K+1)} = f(x^{(K)}) \tag{2}$$

or

$$x^{(K+1)} = g(x^{(K)} \text{[illegible]}) \tag{3}$$

where [illegible] and [illegible] are fixed (stationary) functions. The superscripts represent iteration numbers, and the subscripts represent components. We use as our example the Jacobi iterative method for finding

$$\mathbf{x} = A^{-1}\mathbf{b}. \tag{4}$$

We write

$$A = L + D + U, \tag{5}$$

where L and U are lower and upper triangular matrices, respectively, and D is a diagonal matrix. Inserting Eq. (5) into Eq. (1) and rearranging leads to

$$\mathbf{x} = -D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}. \tag{6}$$

From this we obtain

$$\mathbf{x}^{(K+1)} = \mathbf{y}^{(K)} + \mathbf{c}, \tag{7}$$

where

$$y^{(K)} = B\mathbf{x}^{(K)},$$

$$B = -D^{-1}(L + U),$$

$$\mathbf{c} = D^{-1}\mathbf{b}.$$

Clearly, Eq. (7) has the form of Eq. (2). For many problems,

$$\lim \mathbf{x}^{(K)} = \mathbf{x}, \qquad K \to \infty \tag{8}$$

A necessary and sufficient condition is that the spectral radius of B be less than one.[10] Again, this is just an example for definiteness. Many other stationary iterative algorithms for this problem and for other problems, e.g., the eigen problem, exist.

An engagement processor produces the components of $\mathbf{x}^{(K+1)}$ in sequence. To obtain the ith component, we require the following: (1) a sequencer that puts the proper $y_i^{(K)}$ signal (the one just completed) into the adder, (2) a sequencer that puts the proper c_i into the adder, (3) an adder of $y_i^{(K)}$ and c_i, and (4) whatever amplifier may be needed to insert $x_i^{(K+1)}$ into the matrix multiplier. Figure 1 shows the system schematically. Note that only one adder and only one amplifier are needed.

One problem remains: scaling. Let us define $M^{(K)}$ as the magnitude of the largest component of $\mathbf{x}^{(K)}$. Let us also define M as the maximum value that a component of $\mathbf{x}$ can assume and still be represented. Usually the components of $\mathbf{x}$ are represented by transmissions, so

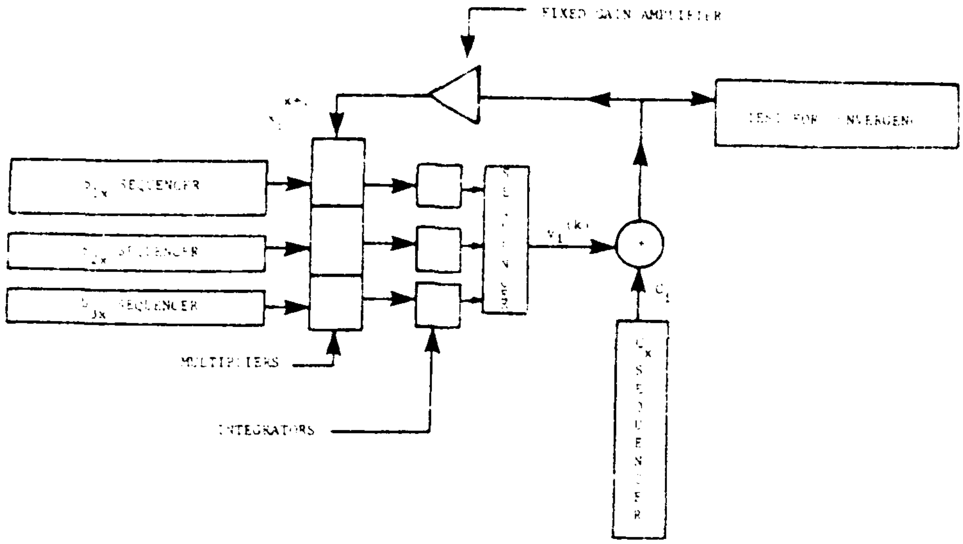


Fig. 1. Implementation of $\mathbf{x}^{(K)} = B\mathbf{x}^{(K-1)} + c$ with an optical engagement processor. The b_{ij}'s are the components of the B matrix.

$$M = 1. \tag{9}$$

If $M^{(K)} \ll M$, the system accuracy is poor. If $M^{(K)} > M$, the input saturates and accuracy is poor. Clearly, good accuracy requires that $M^{(K)} \approx M$. Unfortunately we do not know $M^{(K)}$ until after it is calculated. The best that we can do is to use $M^{(K-1)}$ to approximate $M^{(K)}$ and scale the input to aim at $M^{(K+1)} = \alpha M$, where $\alpha < 1$. We might choose $\alpha = 0.9$. We then set the amplifier to feed back not $x_i^{(K+1)}$ but $sx_i^{(K+1)}$, where s is a scale chosen to give $M^{(K+1)} \approx \alpha M$. Here we are using the fact that, if $A\mathbf{x} = \mathbf{b}$,

$$A(s\mathbf{x}) = s\mathbf{b}. \tag{10}$$

To find $\mathbf{x}$ from $s\mathbf{x}$ we will need to remember the last s used. It may happen that, despite the fact that $s < 1$, the next $M^{(K)}$ is equal to one (indicating probable saturation). In that case we apply in the next iteration a smaller scale. These improvements require more electronics than the simple system of Fig. 1, including some memory [as it uses iterations of the form of Eq. (3)]. In this regard the logic is similar to that of using Kalman filtering in control systems.

What we gain is adaptive scaling. If we can divide the output into N levels of spacing M/N, the full N:1 dynamic range is available at iteration K only if $M^{(K)} \approx M$. So far as M^{K} goes, the adaptive scaling makes this a floating-point calculation. All other components of $\mathbf{x}$ are not calculated with equal accuracy since only one scale is used for $\mathbf{x}$ and it is chosen to be optimum for the largest-magnitude component of $\mathbf{x}$.

The components of $\mathbf{x}$ now appear to be calculated in floating point. Unfortunately, this is not the case. Only the maximum component is calculated by a true floating-point method. While ensuring that the largest component remain close to but less than M, this approach can reduce other components to be much less than M. We seek a method that will scale all components individually. A generalization of the above scaling method will permit a floating-point calculation on all components of $\mathbf{x}$. We rewrite $A\mathbf{x} = \mathbf{b}$ as

$$AS^{-1}S\mathbf{x} = \mathbf{b}$$

and multiply on the left by S:

$$(SAS^{-1})S\mathbf{x} = S\mathbf{b}, \tag{11}$$

where S is a diagonal matrix that is used to scale $\mathbf{x}$. The diagonal element S_{ii} scales the ith component of $\mathbf{x}$. This matrix also scales $\mathbf{b}$. If all the diagonal elements of S are equal, that is, if S is the constant matrix sI, then this approach simplifies to the previous one. The matrix $G = SAS^{-1}$ is similar to A. It has the same eigenvalues, determinant, and condition number. The condition number gives an indication of the size of uncertainties generated in the solution vector $\mathbf{x}$ from uncertainties in the elements of A. The size of the uncertainties in $\mathbf{x}$ is bounded by the condition number times the size of the uncertainties in A. The transformation to G causes no change in the conditioning. If S is the constant matrix sI, it commutes with A, and Eq. (11) reduces to Eq. (10). The matrix S can be changed at each iteration to scale $\mathbf{x}$.

We define

$$\mathbf{v}^K = S^K\mathbf{x}^K \tag{12a}$$

as the scaled solution vector after the Kth iteration and

$$\mathbf{w}^{K+1} = S^K\mathbf{x}^{K+1} \tag{12b}$$

as unscaled output vector after the $k + 1$-th iteration.

Further defining

$$\hat{B}^K = S^K B(S^K)^{-1} \tag{12c}$$

and

$$\mathbf{c}^K = D^{-1}S^K\mathbf{b} \tag{12d}$$

leads to the generalization of Eqs. (6) and (7). For the Kth iteration the two-step process is to find $\mathbf{w}^{K+1}$ as

$$\mathbf{w}^{K+1} = \hat{B}^K \mathbf{v}^K + \mathbf{c}^K \tag{13}$$

and then scale $\mathbf{w}^{K+1}$ to find S^{K+1} and $\mathbf{v}^{K+1}$.

The matrix $\hat{B}^K$ is similai to B. It has the same eigenvalues as B. In particular, the spectral radius of $\hat{B}^K$ is equal to that of B. Therefore the convergence properties of the method using Eq. (13) are the same as those obtained from Eq. (6). In principle, the method allows $\hat{B}$ to be updated at each iteration. If N is the dimension of the problem, this update requires $2N^2$ operations. However, far fewer updates may be required. If $\mathbf{w}$ is not saturated or if none of its elements is small compared to M, then S^{K+1} can be taken to S^K, and no update of $\hat{B}$ is needed. The dynamic range of $\hat{B}$ changes from update to update and is different from $\hat{B}$ The elements of $\hat{B}$ are related to B by $\hat{B}_{ij} = S_{ii}B_{ij}S_{jj}^{-1}$ The columns of B are scaled by S^{-1}; the rows are scaled by S. The method permits apportionment of the dynamic range between the solution vector $\mathbf{v}$ and the matrix $\hat{B}$.

In conclusion, we have considered a generalized scheme that allows the scaling of each component of $\mathbf{x}$ in the solution of the $A\mathbf{x} = \mathbf{b}$ problem and a simplified version in which a single scale is used for all components. The proposed method does not alter the conditioning of the problem (the G matrix is similar to A), nor does it alter the convergence rate (the matrix $\hat{B}$ is similar to B). In the general form, however, a new $\hat{B}$ must be formed from time to time, a computational burden of $2N^2$ multiplications. If the simpler form is used, in which all scale factors are the same, then $G = A$, $\hat{B} = B$, and no additional computational burden is required.

It is clear that, to the extent that the dynamic range of both the given problem and the given hardware permits it, floating-point optical systolic and engagement processors are feasible.

This research was supported by U.S. Air Force contract F19628-82-C-0068.

References

1. D. Psaltis, D. Casasent, and M. Carlotto, Opt. Lett. **4**, 348 (1979).
2. H. J. Caulfield, D. Dvore, J. W. Goodman, and W. T. Rhodes, Appl. Opt. **20**, 2263 (1981).
3. M. Carlotto and D. Casasent, Appl. Opt. **21**, 147 (1982).
4. J. W. Goodman and M. S. Song, Appl. Opt. **21**, 502 (1982).
5. W. K. Cheng and H. J. Caulfield, Opt. Commun. **43**, 251 (1982).
6. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz, Opt. Commun. **40**, 86 (1981).
7. D. Casasent, Appl. Opt. **21**, 1859 (1982).
8. D. Casasent, J. Jackson, and C. Neuman, Appl. Opt. **22**, 115 (1983).
9. R. P. Bocker, H. J. Caulfield, and K. Bromley, Appl. Opt. **22** (1983).
10. R. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, N.J., 1962), p. 13.

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.

Printed by
United States Air Force
Hanscom AFB, Mass. 01731