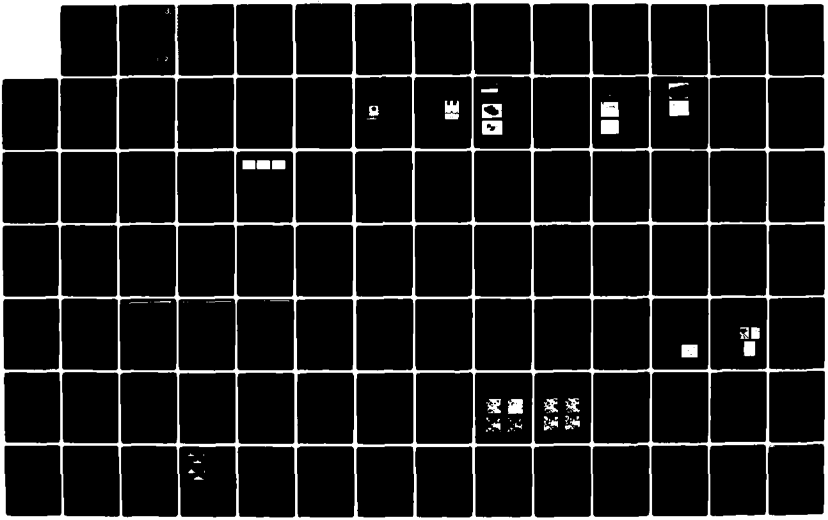
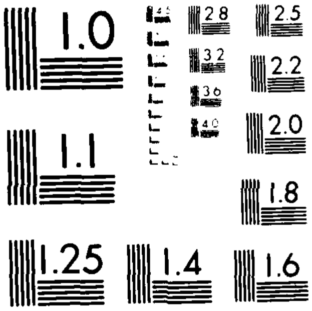
AD-A136 216 OPTICAL DATA PROCESSING FOR MISSILE GUIDANCE(U) 1/2
CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF ELECTRICAL
ENGINEERING D CASASENT 30 SEP 83 AFOSR-TR-83-1103
UNCLASSIFIED AFOSR-79-0091 F/G 17/7 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

AFOSR-TR- 83-1103

3

AD-A136 216

INTERIM REPORT

"OPTICAL DATA PROCESSING FOR MISSILE GUIDANCE"

Submitted by

David Casasent, Principal Investigator
Carnegie-Mellon University
Department of Electrical and Computer Engineering
Pittsburgh, Pennsylvania 15213

Submitted to

Air Force Office of Scientific Research
Bolling Air Force Base
Washington, D.C. 20332

ATTENTION: Lt. Col. Robert Carter, Building 410

Period Covered

30 September 1982 - 30 September 1983

DTIC
ELECTE
DEC 22 1983
S D
D

Approved for public release;
distribution unlimited.

83 12 19 194

Unclassified, 22 November 1983
SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFOSR-TR- 83-1103

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)
Optical Data Processing for Missile Guidance

5. TYPE OF REPORT & PERIOD COVERED
Interim Report
Sept. 1982 - Sept. 1983

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(*s*)
David Casasent

8. CONTRACT OR GRANT NUMBER(*s*)
AFOSR-79-0091

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Carnegie-Mellon University
Dept. of Electrical and Computer Engineering
Pittsburgh, PA 15213

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61102F
2305/B1

11. CONTROLLING OFFICE NAME AND ADDRESS
AFOSR/NE
Building 410
Bolling Air Force Base, D.C. 20332

12. REPORT DATE
30 Sept 83

13. NUMBER OF PAGES
120

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

15. SECURITY CLASS. (*of this report*)
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release;
distribution unlimited.

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

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Acousto-optics, chord transformation, correlator, direct matrix solutions, feature extraction, frequency-multiplexed systolic, matrix decomposition, matrix inversion efficiently, nonlinear local operators, PRIZ, singular value decomposition, sobel operator, synthetic discriminant functions, systolic array processors

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

Progress on real-time 1-D and 2-D spatial light modulators, optical systolic array processors, optical image processing, and distortion-invariant pattern recognition are reported. Our real-time spatial light modulator research concerns the Soviet PRIZ device and bulk acousto-optic transducers. Novel features detailed for the Soviet device include: uniform and directional spatial filtering and static image suppression or change detection. Systolic array processor research includes a new frequency-multiplexed architecture, realization of singular value decomposition and matrix decomposition algorithms. Optical image

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified; 22 November 1983
SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

BLOCK #20 CONTINUED:

processing and pattern recognition research received major attention. A new class of nonlinear local operators including the Sobel operator were described and demonstrated. New optical feature generation techniques for distortion-invariant pattern recognition were developed. These included the generalized optical chord transformation. Our primary distortion-invariant pattern recognition research addressed synthetic discriminant functions with attention to two new efficient calculation techniques. The performance of synthetic discriminant functions for multi-class distortion-invariant pattern recognition, and initial noise performance of this optical pattern recognition algorithm.

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes

Dist	Avail and/or Special
A|1	

ABSTRACT

Progress on real-time 1-D and 2-D spatial light modulators, optical systolic array processors, optical image processing, and distortion-invariant pattern recognition are reported. Our real-time spatial light modulator research concerns the Soviet PRIZ device and bulk acousto-optic transducers. Novel features detailed for the Soviet device include: uniform and directional spatial filtering and static image suppression or change detection. Systolic array processor research includes a new frequency-multiplexed architecture, realization of singular value decomposition and matrix decomposition algorithms. Optical image processing and pattern recognition research received major attention. A new class of nonlinear local operators including the Sobel operator were described and demonstrated. New optical feature generation techniques for distortion-invariant pattern recognition were developed. These included the generalized optical chord transformation. Our primary distortion-invariant pattern recognition research addressed synthetic discriminant functions with attention to two new efficient calculation techniques. The performance of synthetic discriminant functions for multi-class distortion-invariant pattern recognition, and initial noise performance of this optical pattern recognition algorithm.

TABLE OF CONTENTS

1. INTRODUCTION

During the past year (September 1982 - September 1983), our research in optical data processing for missile guidance has addressed many of the key issues and aspects of this technology. This research includes: various new devices and components, new system architectures, new high-speed general purpose optical data processing techniques and systems, tests on new image databases, basic studies of existing pattern recognition architectures, and new pattern recognition techniques, algorithms, and concepts. As in past years, we have been quite faithful in reporting our AFOSR sponsored research in various journals and conference publications. Copies of the more relevant papers we have published over the past year are included as the chapters of this report to provide complete documentation of each aspect of our work.

In Section 2, we provide a summary and overview of our research progress achieved during the past year. This work addresses six vital areas of optical data processing research: (1) real-time spatial light modulators (Sections 3 and 4); (2) systolic array processors (Sections 5-8); (3) image preprocessing (Section 9); (4) optical correlator analysis (Sections 10-11); (5) optical feature extraction (Section 12); and (6) synthetic discriminant functions (Sections 13-15). Topic (1) concerns the vital issue of real-time devices; topic (2) concerns the hottest topic in optical processing at present and a potentially quite general-purpose optical processor; topic (3) applies parallel optical techniques to achieve operations normally performed by digital image preprocessors; topic (4) provides a better understanding of conventional optical pattern recognition architectures with attention to the performance obtained from these systems; topics (5) and (6) address distortion-invariant optical pattern recognition techniques (feature extraction and correlation).

Details on the more salient results of our research are provided in Sections 3-15. In Section 16, we enumerate our AFOSR sponsored publications, the presentations given on this research at conferences and seminars during the past year, and the Master's and PhD students that this grant has supported.

During the past year, the principal investigator (PI) presented invited talks on our AFOSR sponsored research at the following conferences: SPIE International Conference (San Diego, August 1983), IOCC Conference (Boston, April 1983), and CLEO'83 Conference (Baltimore, May 1983). The PI also chaired conference sessions and seminars and served on the organizing committees for the following conferences and topics: SPIE (Robotics), CLEO (Lasers and Light Modulators), and IOCC (Optical Computing).

2. SUMMARY AND OVERVIEW

Sections 3-15 detail our recent research on seven different aspects of optical data processing. Brief highlights of each of these sections and topics follow below.

2.1 REAL-TIME SPATIAL LIGHT MODULATORS

Real-time spatial light modulators are the primary elements necessary for the realization of optical pattern recognition systems. Our research during the past year on two different types of these devices follow. The details of this work are provided in Sections 3 and 4 of this report.

2.1.1 Soviet PRIZ Light Modulator (Section 3): A unique opportunity arose for us to conduct a first-hand four month test and evaluation of a Soviet spatial light modulator (the PRIZ). Section 3 is the last paper on this topic. It summarizes our research, the performance of these devices, their applications, and their unique features (these include automatic edge enhancement, directional edge enhancement, and dynamic image subtraction). Further U.S. research is necessary on these devices to facilitate fabrication of equivalent U.S. components and to allow full understanding of the operation of these new devices.

2.1.2 Acousto-Optic Light Modulators (Section 4): The most commercially available, proven and reliable spatial light modulators are acousto-optic cells. A summary of these elements, the basic architectures employing them, and several new architectures and applications employing these components are reviewed in Section 4. These include systems to process long codes and systems less susceptible to various acousto-optic device shortcomings. These elements are, in general, most appropriate for signal rather than image processing and are quite attractive for optical systolic processor fabrication.

2.2 OPTICAL SYSTOLIC ARRAY PROCESSORS (SECTIONS 5-8)

One of the most general-purpose and flexible optical data processing techniques is the optical systolic array processor. These optical architectures perform matrix-vector and various linear algebraic operations optically with high-speed, parallel processing, and very high computational rates possible. Sections 5-8 detail our recent research in this area. This work is intended to develop an optical systolic processor for air-to-air missile guidance and control.

2.2.1 Frequency-Multiplexed Optical Systolic Processor (Section 5): A new optical systolic array architecture using acousto-optic cells was devised and described. This system employs frequency-multiplexing to represent 2-D data. We refer to it as a frequency-multiplexed acousto-optic systolic array processor. Specific attention was given to: a new matrix inversion and linear algebraic equation solution algorithm, a new technique for handling bipolar data and new matrix-matrix and matrix-matrix-matrix multiplication algorithms. Primary attention was also given to the efficient flow of data and operations on such a processor and to the demonstration that this one system could achieve all of the basic operations required in Kalman filtering. The method that we are pursuing for our air-to-air missile guidance problem employs a Kalman filter. We are working in conjunction with AFIT researchers in this area.

2.2.2 Direct Matrix Solutions (Section 6): Optical systolic array processors allow direct matrix decomposition to be efficiently performed. Optical matrix-vector processors cannot efficiently perform this operation. We were the first to address this topic and to note that the major computational load in a direct solution was the matrix decomposition itself. We thus considered techniques to achieve matrix decomposition on our frequency-multiplexed optical systolic array processor. We detailed how to realize the LU and Cholesky decomposition on our optical processor. This included a new parallel LU algorithm and full detailing of the flow of data and operations in the optical version of this algorithm. New attention to a parallel algorithm was required for efficient realization of this technique on an optical processor.

2.2.3 Optical Singular Value Decomposition (Section 7): Singular value decomposition is an attractive (and in many cases an essential) technique for solving matrix equations, inverting matrices and for singular matrix processing. We detailed one technique to achieve this operation on a general matrix-vector processor.

2.2.4 Guidelines (Section 8): Considerable interest and many publications exist on optical systolic processors. However, much of the published work is not properly directed. In this section, we note that attention should be given: to N^3 problems, to architectures that achieve flexible performance, to architectures that do not require parallel output 2-D detector arrays, to architectures that avoid optical system sources and to systems that can be easily fabricated and realized today.

2.3 OPTICAL IMAGE PROCESSING (SECTION 9)

We distinguish between optical image processing and optical pattern recognition. In image processing, the input to the system is an image and so is the output (usually an enhanced image). Our new research includes the first demonstration of an optical nonlinear local operator. Such operations are normally performed digitally and used for image preprocessing or image processing. We describe two optical architectures to achieve arbitrary nonlinear local operators of large size with the parallelism and speed of optical systems fully utilized. These architectures include correlators using multiple matched spatial filters and computer generated holograms. The nonlinear local operator we detailed and demonstrated was the Sobel operator. This edge-enhancement operator is most attractive and necessary in infrared and multisensor image preprocessing. Extensions of this new basic technique are possible and should be pursued.

2.4 OPTICAL CORRELATOR ANALYSIS (SECTION 10)

Optical correlators are well-known and often used. In our work, we addressed two potential error sources in optical correlators and especially in optical correlators in which coefficient-estimation is the purpose of the optical architecture. Those error sources considered include: finite space bandwidth product and global rather than zero-mean data correlation. The first issue is important when the statistical correlation features are optically estimated. We show from the standard deviation of the crosscorrelation coefficient that the error in the correlation estimate can be appreciable if the space bandwidth product of the image data is small. Major attention is given to the second possible error source, since it is not apparently well-known within the optics community. Conventional optical correlators automatically suppress dc spatial frequency data and thus correlate zero-mean data. This represents no problem if the location of the correlation peak rather than the correlation peak value is of major concern. When the correlation peak value itself is the parameter of concern, then one must carefully distinguish between local and global zero-mean data being used, if one is to achieve an accurate correlation peak estimate. Quantitative data on the magnitude of each of these effects is included.

2.5 BINARY IMAGE CORRELATIONS (SECTION 11)

As an extension of earlier AFOSR research, we experimentally considered the performance of correlators operating on binary images. Such architectures are attractive for digital and optical implementation, because of the reduced dynamic range that is needed. Using the peak-to-sidelobe ratio of the correlation output, we show in theory and confirm by experiments that a binary correlator (i.e. a correlator operating on binary or two-level image data) can provide better performance than a correlator operating on gray-scale imagery.

2.6 OPTICAL FEATURE EXTRACTION (SECTION 12)

In this year, we introduced a major new approach to distortion-invariant pattern recognition: the hybrid combination of optical feature generation, digital feature extraction and classification. In Section 12, we advance a new feature set (the generalized chord distribution), a new optical technique to produce these object features and the use of a digital classifier that maximizes the Fisher ratio. This digital classifier is used for feature extraction and for analysis of the optically generated feature outputs. We demonstrated this technique for distortion-invariant multi-class pattern recognition and obtained very promising results. We plan further work on this technique and other optically-generated features.

2.7 SHIFT-INVARIANT DISTORTION-INVARIANT PATTERN RECOGNITION (SECTIONS 13-15)

All feature extractors (optical or digital) cannot accommodate multiple objects or significant noise. Only correlators can achieve such multi-object or shift-invariant performance with high noise or clutter present. The major approach we have pursued for this purpose is the synthetic discriminant function concept. In this case, a synthetic discriminant function (capable of distortion-invariant pattern recognition) is synthesized off-line and then an optical matched spatial filter of it is formed and used in an optical or digital correlator.

2.7.1 Efficient Matrix Inversion (Section 13): Our off-line synthetic discriminant function synthesis techniques require processing and inversion of large matrices. In Section 13, we describe a new and most efficient technique we devised to achieve this with significantly reduced storage and computations. This is quite essential and necessary as the size of the synthetic discriminant function training set increases. The algorithm we developed is essentially an efficient matrix inversion technique with minimal data storage requirements. Experimental comparison on tank images was used to verify the superiority of this new off-line filter synthesis technique.

2.7.2 Synthetic Discriminant Function Performance (Section 14): We have obtained a large four-class database of ship images with 36 different distorted versions of each ship class available (taken at 10° intervals around the object from a zero-degree attack angle). In Section 14, we provide quantitative data on the first full test of our four different types of synthetic discriminant functions on this 144 image data base. The performance obtained is quite excellent with over 90% correct classification achieved for multi-class distortion-invariant pattern recognition.

2.7.3 Initial Noise Performance (Section 15): In Section 15, we review our synthetic discriminant function synthesis techniques and advance the first noise performance of these algorithms. The performance obtained in the presence of noise was excellent. These are perhaps the most impressive and comprehensive pattern recognition results (optical or digital) obtained for distortion-invariant multi-class pattern recognition.

3. APPLICATIONS OF THE PRIZ LIGHT MODULATOR

Reprinted from **Applied Optics**, Vol. 21, page 3846, November 1, 1982.

Applications of the Priz light modulator

David Casasent, Frank Caimi, M. P. Petrov, and A. V. Khomenko

The Priz light modulator suppresses input data at zero spatial frequency, can provide directional spatial filtering, and can perform dynamic image selection or change detection. In this paper, we summarize the Priz's performance and provide experimental confirmation of the above three image processing applications of this device.

I. Introduction

Optical signal processing and image pattern recognition applications require real-time and reusable devices on which the input data to be processed can be recorded for subsequent optical processing. These vital components in an optical processor are known as spatial light modulators (SLM). Although many candidate SLM devices exist,[1] we will restrict our present attention to only one such device, the Priz light modulator.[2-8] Priz is a Soviet acronym that translates as image transformer. This modulator was proposed by a group of researchers of A.F. Ioffe Physico-Technical Institute (FTI) of the Academy of Sciences of the U.S.S.R. It employs the same active element, i.e., the bismuth silicon oxide (BSO) type crystal, as the well-known Prom modulator.[9]

At the Electrical Engineering Department of Carnegie-Mellon University (CMU), five Prom and Priz units fabricated at the FTI laboratory headed by Petrov were tested and evaluated. One worker of FTI participated in the research program (Khomenko).

In Sec. II, we review the structure of the Priz light modulator and the motivation for its fabrication and highlight the spatial frequency response data obtained on the devices we evaluated at CMU. We then include (Sec. III) a summary of the dynamic and optical performance of the Priz. These data were obtained from experiments performed in both the Soviet Union and at CMU. We include several image and signal pattern recognition correlation examples of the use of the device in Sec. IV. Because of the transverse linear electrooptic effect used in the Priz to modulate read light, it exhibits three unique features that are of use in various image processing and pattern recognition applications. These include: suppression of dc and low spatial frequency data as well as directional filtering of input spatial frequencies oriented in selected angular directions (Sec. V) and a quite unique feature referred to as dynamic image selection in which the device responds only to *changes in the input image data* (Sec. IV). Our summary and concluding remarks follow in Sec. VII.

II. Operation of the Priz

The Prom light modulator[9] consists of a BSO crystal ~20 × 20 × 0.4 mm with Parylene insulating layers and transparent electrodes on the large faces. In operation the spatially modulated data to be processed are imaged or scanned onto the device in λ_W write light (350–450 nm). Photocarriers are generated in the photoconductive BSO, and a spatially modulated charge layer is produced within the BSO. When the device is illuminated with a uniform read light beam at λ_R (usually 633 nm), the λ_R light emerging from the device is polarization modulated spatially with an amplitude of modulation that varies spatially in accordance with the original λ_W input light or data pattern. This λ_R modulation occurs by the linear-longitudinal electrooptic or Pockels effect. The polarization modulation can be converted to amplitude modulation when a crossed analyzer is placed behind the modulator.

In the Priz light modulator,[2-8] a [110] or [111] cut BSO crystal is used rather than the [100] cut crystal used in the Prom. Other proprietary fabrication techniques are employed, but the issue of major importance is that with these different crystal cuts, the device now modulates λ_R light by the transverse rather than the longitudinal electrooptic effect. The spatially varying λ_W light distribution is still incident on the crystal's large faces collinear with the applied electric

---

M. P. Petrov and A. V. Khomenko are with A. F. Ioffe Physico-Technical Institute of the U.S.S.R. Academy of Sciences, 194021 Leningrad, U.S.S.R.; the other authors are with Carnegie-Mellon University, Department of Electrical Engineering, Pittsburgh, Pennsylvania 15213.

Received 22 April 1982.
0003-6935/82/213846-09$01.00/0.

field direction, and the spatially varying charge layer parallel to the crystal's large faces is still induced. However, the transverse component of this field is what is used to provide the spatial modulation of the λ_R light.

The Prom exhibits low diffraction efficiency η and a sharp $\eta \simeq 1/f^4$ decrease in usable output light intensity at high spatial frequencies f.[10] If we assume that the sine wave electric charge grating with amplitude σ_0 induced during image writing in the Prom is infinitely thin and that it exists at the crystal–dielectric interface, the phase modulation vs spatial frequency is described by[11]

$$\Delta\phi = \frac{2\pi\sigma_0}{U_{\lambda/2}f(\epsilon_d \coth 2\pi f d_d + \epsilon_c \coth 2\pi f d_c)}\,, \qquad (1)$$

where ϵ_d and ϵ_c are the relative dielectric constants of the dielectric layer and the BSO crystal, respectively, d_d and d_c are the thicknesses of these layers, and $U_{\lambda/2}$ is the halfwave voltage of the crystal. Equation (1) indicates that when f increases at $f \geq 1/2d_d$, $\coth 2\pi f d_d \simeq 1$ and $\coth 2\pi f d_c \simeq 1$, and thus $\Delta\phi$ decreases $\propto 1/f$. Since $\eta \propto (\Delta\phi)^2$, Eq. (1) predicts $\eta \propto 1/f^2$. However, in experiments a sharper ($\eta \propto 1/f^4$) dependence was observed. This was attributed to the fact that, in the process of image writing, a volume electric charge distribution is formed within the crystal volume rather than an infinitesimally thin one.[10] Using this new model, it has been shown that in the case when the charge is distributed throughout a layer of thickness d_a near the crystal–dielectric interface

$$\Delta\phi = \frac{\sigma_0[\cosh 2\pi f d_c - \cosh 2\pi f(d_c - d_a)]}{U_{\lambda/2}f^2 d_a(\epsilon_d \coth 2\pi f d_d + \epsilon_c \coth 2\pi f d_c)\cdot \sinh 2\pi f d_c}\,. \qquad (2)$$

Equation (2) predicts an $\eta \propto 1/f^4$ dependence at high f, which agrees with experimental results. Thus the sharp dependence of η on f for the Prom is attributable to the volume character of charge distribution. The most complete and detailed theoretical description of the Prom device can be found in Refs. 12 and 13. These papers discuss a model that includes the location of the charge layer within the BSO, the thickness of the different device layers, the wavelength of the light used. To overcome the disadvantages of the volume charge predicted by Eq. (2), it was suggested in Refs. 14 and 15 to use the transverse electrooptic effect for read light modulation rather than the longitudinal effect used in the Prom. The modulator that uses the transverse electrooptic effect is called the Priz. The corresponding calculation of the phase modulation for the case of the transverse effect shows that[6]

$$\Delta\phi = \frac{4\pi\sigma_0}{U_{\lambda/2}\epsilon_c f}\left\{1 - \frac{\epsilon_d[\cosh 2\pi f d_c + \cosh 2\pi f(d_c - d_a) - 1] + k_1}{2\pi f^2 d_a(\epsilon_d \tanh 2\pi f d_c + \epsilon_c \tanh 2\pi f d_d)\cosh 2\pi f d_c}\right\}\,, \qquad (3)$$

where $k_1 = \epsilon_c \tanh 2\pi f d_d \sinh 2\pi f d_a$.

From this formula, two characteristic features of the Priz device can be seen. First, $\Delta\phi(0) = 0$; i.e., the modulator suppresses the dc component. Second, at high f, $\Delta\phi \propto 1/f$ and $\eta \propto 1/f^2$, i.e., the Priz η vs f characteristic is superior to that of the Prom.

III. Performance of the Priz

In the MTF tests performed on the Priz at CMU, we operated the device at 2 frames/sec. In other experiments, we operated the device at 20 frames/sec. A faster frame rate should be possible, but no effort has yet been made to determine the device's maximum cycle time. However, results of several experiments are useful in providing some indication of the final device performance possible on the Priz.

In the experiments performed in FTI, the device was operated with write times as short at 7 nsec using a pulsed laser source with an intracavity electrooptic modulator. In this case, photocarriers were generated in a negligible time, but the output light pattern was not visible until 1 μsec later, and it peaked after 10 μsec. Carrier mobility and transit times thus appeared to limit the minimum write/read cycle times of the Priz to 10 μsec. In the more conventional operating mode, 1-msec exposure times are used, and the output pattern is then immediately visible.

Erase time is a second limitation on the device's speed. In all tests performed at CMU, a fixed 1-msec erase flash (from the standard erase unit provided with the U.S.A. Prom) was used. However, neither the Prom nor the Priz can be recycled immediately after erasure, and a delay time is necessary to allow redistribution and relaxation of excited carriers within the crystal. One millisecond of relaxation time and hence a total 1-msec erase time appear adequate. High-energy erase pulses cannot decrease this time, since they generate and dislodge other carriers within the bulk of the device, and longer relaxation times between the end of erasure and the start of a new write cycle then become necessary. Thus operation of the Priz at a 10^3-frame/sec (write/read/erase cycle) rate appears possible, but additional theory and experiments are necessary to confirm this. Moreover, an application for which the entire spatial input data changes every millisecond is necessary to merit such an effort together with attention to how one can introduce such a new 2-D λ_W spatial distribution to the device at these 10^3-frame/sec rates.

The storage times for the Priz are adequate for most applications [1-min storage in the dark and 10–20-sec storage under a high λ_R = 633-nm read light intensity (I_R = 2 mW/cm)]. These can be somewhat controlled by varying the thickness of the insulating layers. The lifetime of the Priz, like that of the Prom, appears to be excellent. Selected Priz units have been operated for over one million cycles at a 20-frame/sec rate with no noticeable change in performance. The only concern with the device's lifetime appears to be its operation with high I_R light levels. In this case, a device with high transmittance and electrodes with good conductivity is necessary. For such cases, InO_2 electrodes are used. The performance of such Priz units appears to be good (as several of them have operated successfully for several years).

The resolution of the Priz is mostly understood and has been experimentally verified by several techniques. Its diffraction efficiency η_0 at low spatial frequencies is ~1%, its 10-dB spatial frequency response $f_{0.1}$ is ~30 cycles/mm, and the spatial frequency $f_{0.01}$ at which $\eta = 0.01\eta_0$ is 100 cycles/mm. At CMU, we operated the device with input data having spatial frequencies as high as 80 cycles/mm. With higher quality optical systems and a different area detection technique, FTI researchers have measured diffraction efficiency on the device beyond 500 cycles/mm. Since the usable output light intensity at these high spatial frequencies is quite low, only in selected applications can such resolution actually be used. The Priz performs well with input write light exposures E_W of 50 $\mu J/cm^2$ or less with corresponding interharmonic distortions and hence response nonlinearities below 1% at these write light energies. The sensitivity of the device defined as the write light exposure E_W necessary to achieve $\eta = 1\%$ at 5 cycles/mm is 50 $\mu J/cm^2$. In our CMU data on these devices,[7,8] an available laser source not optimized to the thickness of the Priz used was employed. With this experimental setup, we obtained only $f_{0.5} = 20$ cycles/mm and $f_{0.1} = 30$-mm resolution. If the optimal read light wavelength were used, we expect superior results with $f_{0.5}$ in excess of 30 cycles/mm as obtained at FTI.

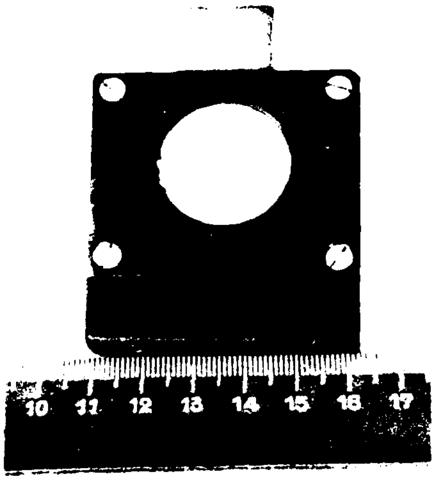


Fig. 1. Priz spatial light modulator.

Both the Priz and Prom modulators have high optical quality and allow use of large crystal sizes. The Priz units evaluated at CMU had a 15-mm diam active area with $\lambda/4$ optical quality. Larger units have been fabricated (up to 30-mm diam) with ~1λ optical flatness. The standard Priz units have also been fabricated, tested, and used with $\lambda/10$ optical quality. The BSO crystal in the Priz is nominally 400 μm thick, and the insulating layers are ~3 μm thick. No substrate is used in the Priz, or the Prom, as clamping effects change the dielectric constants of the materials and induce stresses in the device. One of the units tested at CMU is shown in Fig. 1.

In Table I, we summarize the salient Priz performance parameters. These data were obtained from diverse FTI and CMU tests and experiments. As with any BSO device, a wide range of performance is possible depending upon the thicknesses used for the different layers in the device. The parameters in Table I are all simultaneously obtainable, but they should be interpreted with the above consideration in *mind*. In all cases, the device should be operated at the intended write light exposures E_W and write light wavelength λ_W if optimum device performance is to be obtained. MTF data are frequently used to describe the spatial resolution of SLMs. Since the Priz has a dc response of zero, the MTF function $\tau(f)$ is not usable for such a device. Rather the diffraction efficiency η of such a device is the

Table I Priz Parameter and Performance Specifications

Parameter	Specification	Parameter	Specification
BSO crystal	400 μm thick	Address time	7 nsec (min)
Insulating layer	3 μm thick	Write/read cycle	1 msec (min)
Electrodes	Pt or InO_2	Erase time	1 μsec (min)
Active area		Erase cycle	1 msec (typical)
Typical	15-mm diam	Frame rate (typical)	20 frames/sec
Maximum	30-mm diam		
Optical quality		Dark Storage	1 *min* (typical)
Typical	$\lambda/4$	Storage with readout	20–30 sec (typical)
Special request	$\lambda/10$		
Write light		Diffraction effic. (η_0)	1%
Wavelength	350–500 nm	Resolution	
Exposure (typical)	50 $\mu J/cm^2$	at $\eta = 0.1\eta_0$	30 cycles/mm
Sensitivity (for $\eta = 1\%$ at 5 cycles/mm)	50 $\mu J/cm^2$	at $\eta = 0.01\eta_0$	> 100 cycles/mm
Read light		max. measured	> 500 cycles/mm
λ (typical)	633 nm	Spatial freq. response	$\eta \propto 1/f^2$
Intensity (typical)	2 mW/cm^2		

appropriate parameter. This is why all spatial frequency resolution data in Table I are given in terms of η vs f. These data were obtained with the Priz device operated dynamically in real time. In coherent optical processing, the amplitude transmittance $m(f)$ of the SLM vs frequency f is the parameter of interest. If an analyzer is used and the input pattern is the sine wave grating $I = I_0(1 + \sin 2\pi f x)$, the amplitude transmittance for the Priz can be written in the linear approximation as

$$t(x) = m(f)\cos 2\pi f x \qquad (4)$$

Equation (4) implies that the dc component of the Priz response is zero [since from Eq. (3) $\Delta\phi = 0$ at $f = 0$], and the readout pattern has a $\pi/2$ phase shift with respect to the recorded one. In this case, $m(f)$ is directly obtained from the $\eta(f)$ data provided by

$$|m| = 2\sqrt{\eta}. \qquad (5)$$

This expression follows from the Fourier transform analysis of Eq. (4).

In Fig. 2, we show the diffraction efficiency η of Priz and Prom units at comparable $E_W = 50\text{-}\mu\text{J/cm}^2$ exposures. From these data, we see the superior diffraction efficiency and resolution of the Priz. We also see that the response of the Priz decreases at higher spatial frequencies at a much lesser rate than that of the Prom. Where $m(f)$ vs f is plotted rather than $\eta(f)$, the differences would be even larger. We also note that the Priz exhibits a suppressed response at low spatial frequencies. This feature follows directly from the fact that, upon uniform illumination of the device with write light, a uniform longitudinal electric field is formed that has little or no transverse component. Since the Priz employs the transverse electrooptic effect, it will not modulate in response to such light. As a result η of the Priz device peaks at a spatial frequency of 4 cycles/mm (for the unit tested) rather than at dc.

IV. Use of the Priz in Pattern Recognition and Signal Processing Correlations

An attractive optical correlator for image pattern recognition is the joint transform correlator.[16] In this system, the reference object being sought is placed beside the real-time input scene in the input plane of a 2-D optical FT system. The objective is to determine if the reference object is present in the input scene and to determine its location. Such pattern recognition applications are appropriate for locating objects on an assembly line and locating areas and landmarks in satellite imagery as well as in missile guidance and many other applications. In the joint transform correlator, the Fourier transform of the input and reference data is formed on an intensity sensitive material (such as film, the Priz, or Prom). The Fourier transform of this joint FT pattern is then formed, and it can be shown[16] that it contains the correlation of the input and reference images. In Fig. 3 we show an example[3] of such a correlation performed on the Priz. The FT of the two input objects [Fig. 3(a)], identical images of lobsters, was recorded on a Priz placed in the FT plane of a lens, which was behind the joint input pattern of Fig. 3(a). The FT of the data recorded on the Priz is shown in Fig. 3(c) and its cross-sectional scan in Fig. 3(b). This full correlation plane pattern contains a central term that is the sum of the autocorrelations of each input object. The large spikes on the left and right in Fig. 3(b) are the correlation of the two input objects. Their presence indicates that the two input objects are similar, and the relative position of the peaks denotes the location of the reference object within the field of view of the input image.

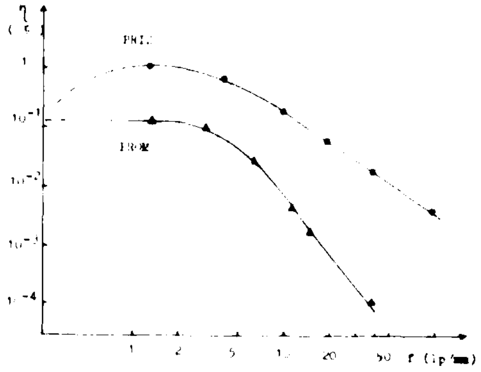


Fig. 2. Diffraction efficiency η vs spatial frequency f for the Prom and Priz spatial light modulators.

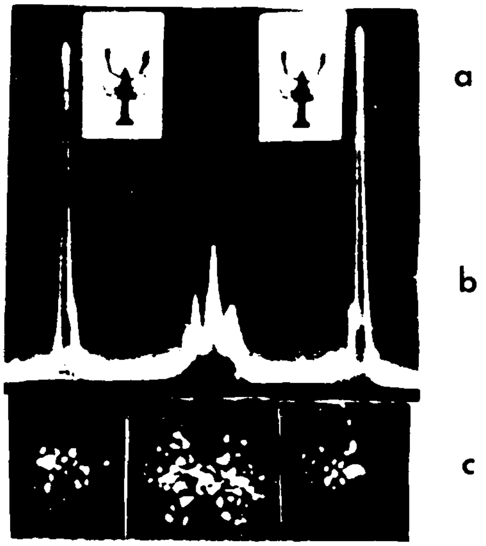


Fig. 3. Real-time image pattern recognition with the Priz used in the Fourier transform plane of a joint Fourier transform correlator: (a) input and reference images; (b) cross-sectional scan; and (c) an image of the output correlation plane pattern. The two peaks to the right and left in (b) and (c) represent the correlations of the two input objects.[3]

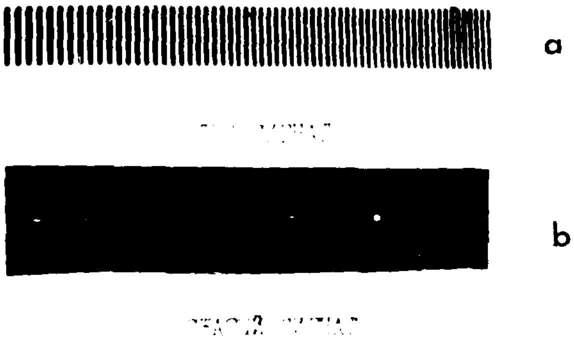


Fig. 4. Real time correlation or compression of a linear frequency modulated signal using the Priz as the input transducer: (a) linear frequency modulated input signal; (b) compressed output correlation signal.

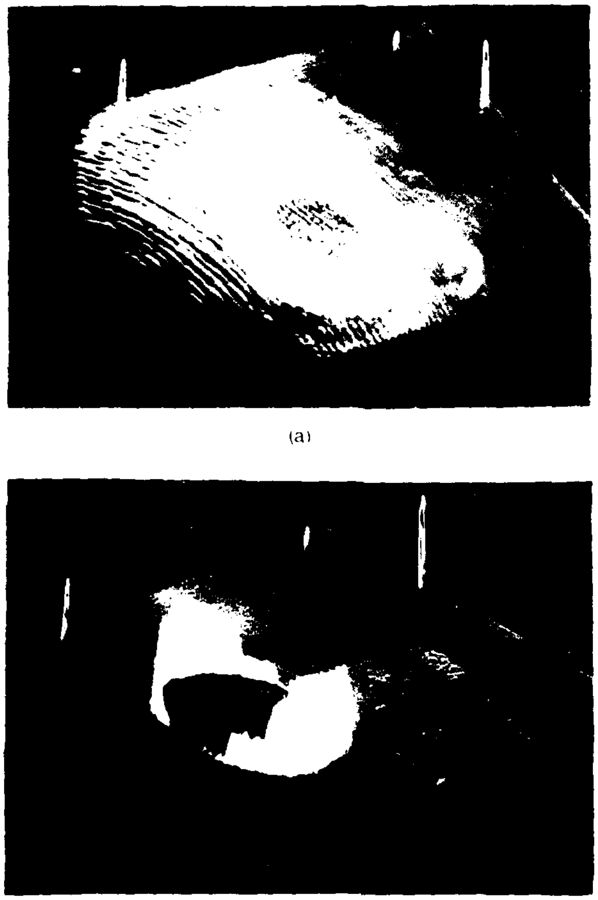
Signal correlations have also been performed on the Priz. An image of a linear frequency modulated (LFM) signal [Fig. 4(a)] was recorded on the Priz. The FT of this pattern formed with a lens whose focal length was matched to the frequency range of the LFM waveform yields the resultant compressed pulse or output correlation peak[2] shown in Fig. 4(b). The width of the experimentally obtained correlation peak was 1/100th of the width of the original LFM signal. This is in good agreement with the theoretical pulse compression factor of 120 for the LFM signal used. The difference between theory and experiment was due to taper in the input light beam and other effects.

V. Image Spatial Filtering Using the Priz

In Fig. 2 and Sec. III, we noted that because of the transverse electrooptic effect in the Priz, dc and low spatial frequency data are automatically suppressed. In Fig. 5(a), we show the reconstruction of a circular input object recorded on the Priz. As expected, only the edge contour of the object appears (due to the automatic dc spatial frequency suppression performed by the Priz). The amount of dc suppression depends on how closely the read light beam is incident to the normal to the crystal. In our experiments at CMU, we were able to obtain a dc suppression of 10 when the normal to Priz was aligned within 4° of the read beam. For a 10^4 suppression factor, 1° alignment is necessary. In the image processing experiments at CMU (Figs. 5 and 7), the read beam was incident at an angle of 1.7° to the normal to the crystal.

The Priz has anisotropic properties arising from those of the linear transverse electrooptic effect. In particular, it exhibits a distinct difference in response to circularly and linearly polarized read light. Figure 6 shows how diffraction efficiency to the Priz depends on orientation of the crystal's axes when read out with linearly and circularly polarized light. In the data of Figs. 5 and 7, the electric vector of the linearly polarized light was along the [112] axis of the crystal. In Fig. 6, the diffraction efficiency η as a function of the angle between the wave vector of the sine wave grating and the [110] crystal axis is plotted in polar coordinates.

The outer circle in Fig. 6 describes the device's response to circularly polarized input light. As seen, it is quite uniform, and thus operation with circularly polarized input light produces no directional preference for input spatial frequency. The reconstructed image in Fig. 5(a) verifies this response and is essentially how the circular outer curve in Fig. 6 was obtained. However, the response of the Priz to linearly polarized read light is quite different. In the two inner figure eight shaped curves in Fig. 6, we show the response for linearly polarized read light. When the device is exposed to linearly polarized input light, it exhibits a preferred response η for input spatial frequencies oriented in one direction, while greatly suppressing input spatial frequencies oriented in the orthogonal direction. The direction in which spatial frequencies are suppressed can be controlled by the polarity of the voltage applied to the modulator if the polarization of the read light is



(a)

(b)

Fig. 5. Real time image edge enhancement and directional spatial filtering using a [111] Priz. (a) edge enhanced reconstructed image; (b) directionally filtered reconstructed image.

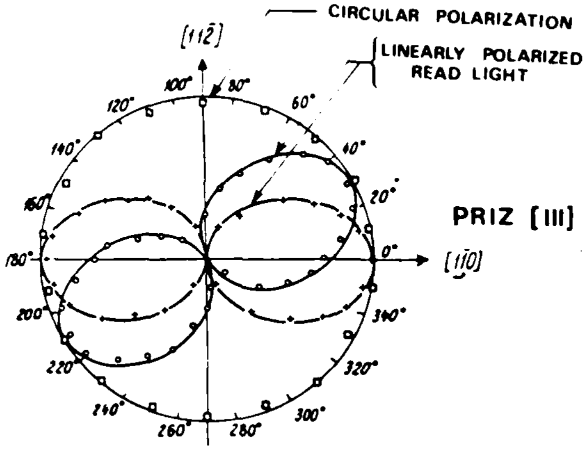


Fig. 6. Diffraction efficiency η for the Priz as a function of the angle between the wave vector of the recorded sine wave grating and the crystal's axes for input read light with circular and linear polarization. Curves 1 and 2 correspond to experimental points for linearly polarized light, and curve 3 corresponds to circularly polarized read light.

fixed. The two figure eight shaped plots in Fig. 6 were obtained with the same polarity of the read light but with different polarities of the applied voltage. Differences arising from changing the voltage polarity can be attributed to the optical activity of the BSO crystal.[17] Figure 5(b) shows the reconstructed image of a circular object recorded on the Priz and read with linearly polarized light. As seen, the spatial frequencies in one direction are suppressed as predicted by Fig. 6.

For a [110] cut Priz, similar plots of η vs the read wave vector's direction result. However, for circularly polarized read light, a saddle-shaped response rather than a circular one results. Similarly, a larger η (a factor of 2 larger than for the [111] cut Priz) results when the [110] cut device is operated with linearly polarized read light. Thus the [110] cut device is preferable for multichannel 1-D signal processing applications and others in which directional spatial filtering is desired. Conversely the [111] cut Priz is preferable for image processing where a uniform response is generally desired for all input spatial frequency directions.

To achieve a high degree of suppression of the dc component in the image (both with linearly and circularly polarized read light), the modulator should be placed between a high-quality polarizer and analyzer, which in the case of circularly polarized light can be achieved with a $\lambda/4$ wave plate and a linear polarizer. The dc suppression, directional spatial filtering, and edge enhancement features of the Priz are quite useful preprocessing operations for multisensor and IR pattern recognition. In Fig. 7, we show the original IR image [Fig. 7(a)], the image constructed from a [110] cut Priz with the read light polarized at 45° [Fig. 7(b)] and with vertically polarized read light [Fig. 7(c)]. The reconstructed image in Fig. 7(b) approximates an edge-enhanced version of the original image, whereas the reconstruction in Fig. 7(c) results in enhancement of vertical lines in the original image and suppression of horizontal spatial frequencies in the original input pattern.

VI. Dynamic Image Selection

In investigating the response of the Priz to spatially moving 2-D input patterns, it was found[1] that the device's response was a function of both the spatial frequency of the input data and the velocity with which the input data moved across the input field of view. A modified version of the Priz was used in these experiments. It had no insulating layers, so electrodes were evaporated directly on the crystal's surface.[18] The response of the device to an 0.5-mm wide input line was measured for different velocities (1–40 mm/sec) of the input object across the input plane, and it was found[4] that the response of the device peaked when the velocity was ~ 7 mm/sec. The response of the Priz is thus a function of both time and space (i.e., the spatial frequency of the input data and the rate at which it

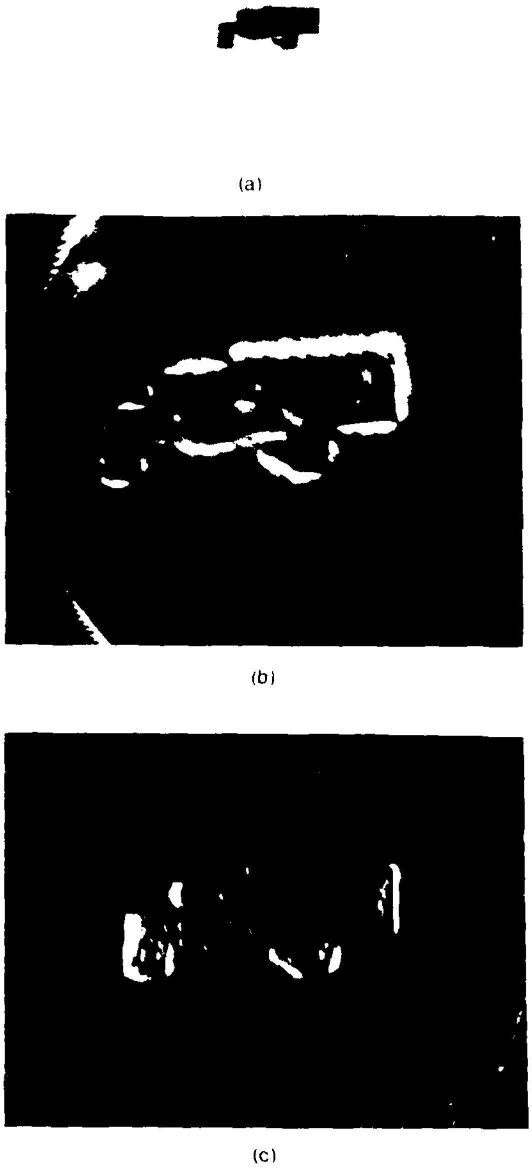


Fig. 7. Real-time image edge enhancement and directional spatial filtering of IR imagery on a [110] Priz: (a) original input image; (b) edge enhanced reconstructed image; (c) directionally filtered reconstructed image.

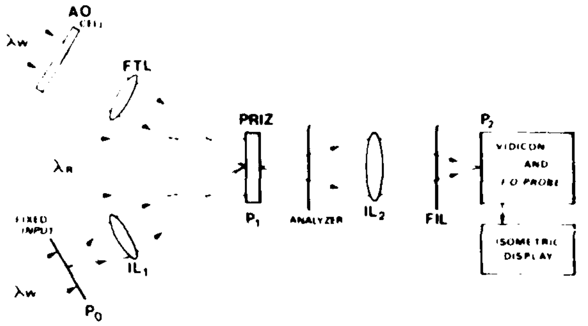


Fig. 8. Schematic diagram of the optical system used to demonstrate dynamic image selection or change detection on the Priz.

changes or moves with time). An initial attempt to describe the combined time and space response of the device has been reported.[5,18] Initial experiments[4,18] indicate that as the spatial frequency of the input data decreases, the input velocity for which the response η of the device peaks shifts to higher velocities.

To obtain this effect, the Priz is operated with a fixed voltage across its electrodes rather than with the applied voltage switched between positive and negative polarities. In the normal mode (polarization of the applied voltage switched), the device performs like the Prom. Only with a fixed voltage will it perform dynamic image selection. In this mode, the device responds only to changes in the input image. This operating mode is attractive for many applications such as change detection, and it also greatly simplifies the electronic support system necessary (since a fixed rather than a switching high-voltage supply can be used). Over a selected range of input temporal frequencies f_t (where this range varies as a function of the input spatial frequency f_x), the η vs f_t response is linear, and the device performs a time differentiation of the input data. This range of f_t is quite small, and moreover it varies with the intensity of the input write light. For these reasons, this Priz device features is best termed dynamic image selection (i.e., the device's output represents only the changing part of the input data) rather than temporal differentiation.[4] This effect can also be observed with a fixed input and with the write light beam pulsed on and off. In this case, whenever the write light changes (goes on or off), an output image of the input data appears and then decays with a time constant that is a function of the intensity of the write light. If the differential phase $\Delta\phi$ of the output light is measured, it is seen to be of opposite sign when the write light is switched from off to on compared to when it is switched from on to off.

When the Priz units were being tested at CMU, we found this Priz feature to be most attractive and thus assembled the system of Fig. 8 to demonstrate the use of the Priz in change detection. The system of Fig. 8 contains two input planes. Plane P_0 contained a fixed image, in our case a random pattern of uncorrelated noise and correlated noise of different correlation lengths and with different mean values. This fixed P_0

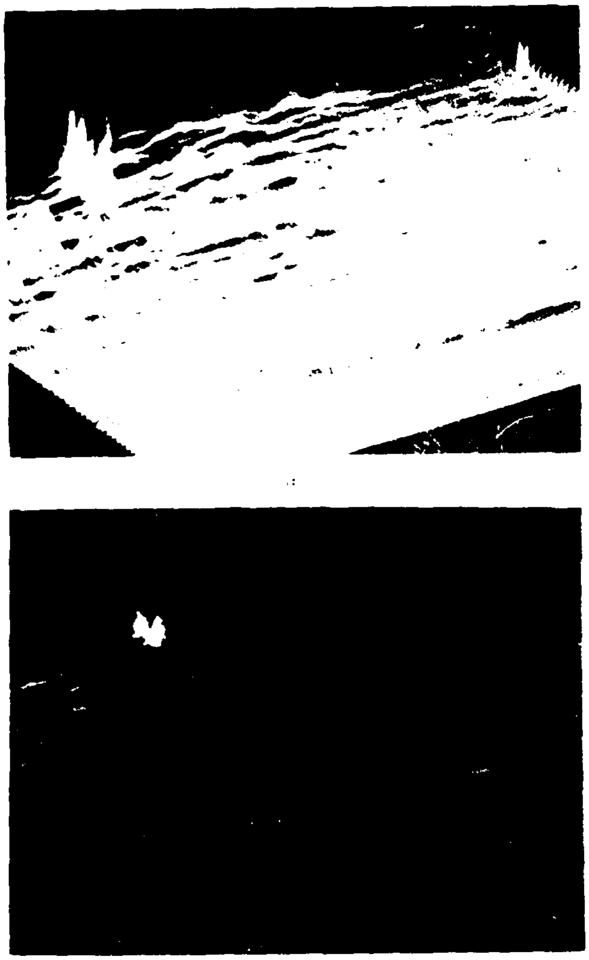


Fig. 9. Demonstration of dynamic image selection or change detection in real time on the Priz: (a) composite image of a moving target on a fixed noise background; (b) dynamic real-time selection of the moving target portion of the image in (a).

pattern was imaged onto the Priz at P_1 (by imaging lens IL) together with the FT of an acoustooptic) (AO) cell (using the FT lens FTL). The AO cell was operated in the scanning mode with a repeated LFM input signal. This caused a scanning spot to traverse the Priz (superimposed on the fixed noise pattern from P_0), thus simulating a moving object on a noise background. The velocity of the scanning spot was adjusted to be 2.8 mm/sec, and its size was ~1 pixel (40 μm).

Both the fixed and moving input patterns were imaged onto the Priz in λ_W = 476-nm light from an argon-ion laser. Readout was performed in λ_R = 633-nm light incident normal to the Priz as shown in the left side of Fig. 8. The pattern on the Priz was then imaged onto P using imaging lens IL through a crossed analyzer and a 633-nm filter (FIL). The output P pattern was detected on a vidicon, and the dynamic moving output was visible on an isometric display where it could be photographed. A fiber optic (FO) probe with a microscope and PMT was also placed at P to allow quantitative measurements of the output plane to be made.

In Fig. 9(a), we show the full output image at P (with the high-voltage Priz power supply operated in the normal pulsed mode). This output shows the fixed background noise pattern and the moving spot. (The moving object or spot is present in the back left of the figure.) When the high-voltage Priz power supply polarity was fixed, only the time varying portion of the input pattern appeared at the output. In this case, only the moving spot produced by the scanning AO cell was visible. In Fig. 9(b), we show the P output for one location of the scanning spot (corresponding to a simulated moving target in the constant noise background image). As can be seen, the Priz suppresses the fixed background noise quite well.

Table II Quantitative Dynamic Image Selection Data on the Priz

R_{in}	1	2.5	10	25
R_{out}	5	50	48	49

To quantify the amount of background suppression obtained, the intensity of the moving spot was varied and the ratio

$$R = \frac{\text{intensity of the moving spot}}{\text{average intensity of the fixed background}} \tag{6}$$

was measured at the input and output of the system (i.e., with and without dynamic image selection or change detection). The results are summarized in Table II. From these data, we note that once the intensity of the scanning spot has been increased so that $R_{in} = 2.5$ or greater, a constant $R_{out} \simeq 50$ ratio results with the intensity of the dynamic part of the output image being 50 times the average background level in the P_2 output. This occurred because when R_{in} was increased above 2.5, saturation of the Priz occurred, and thus no further changes in the effective R_{in} resulted. This is expected since, with the Priz in the FT plane of the AO cell, all the light from the cell was concentrated onto a single pixel on the Priz.

VII. Summary and Conclusion

A complete and unifying summary (at this present time) of a new light modulator, the Priz, has been presented. Many new experimental demonstrations recently obtained at CMU were included. The theory of operation of the Priz was reviewed and experimentally verified, and its similarity and differences from the Prom were noted (Sec. II). The first unifying summary of the performance parameters of the Priz was also advanced (Sec. III). A summary of many of the possible applications of the Priz was then presented. This included conventional optical pattern recognition and optical signal processing correlators (Sec. IV) plus three new image processing operations: dc suppression and directional filtering (Sec. V) plus dynamic image selection (Sec. VI). The first image processing operation had been experimentally demonstrated previously. Directional spatial filtering demonstrations had not been previously described. Our dynamic image selection experiments together with Ref. 18 represented the first examples of the use of the device for the selection of the dynamic part of an image from a constant fixed noise background (i.e., change detection).

Many aspects of the Priz device are well understood. However, further theoretical analyses and modeling together with further device fabrication and experimental testing are necessary to understand fully and describe many of the observed features of the device. Issues meriting further analysis include the nonlinearity associated with the transverse electrooptic effect, the combined time and space dependence of the resolution of the device, and a theoretical formulation of the dynamic image selection feature of the device with attention to the selection of device parameters to optimize and control this effect.

The promising performance parameters tabulated for the Priz light modulator and the experimental verification included of several of the novel features of this device indicate that a wealth of new research efforts and applications are still possible in the field of real time and reusable spatial light modulators.

References

1. D. Casasent, Proc. IEEE **65**, 143 (1979).
2. M. Petrov, at Conference on 4-D Optics, Mexico (1980).
3. M. P. Petrov *et al.*, Sov. Phys. Tech. Phys. **25**, 752 (1980).
4. M. P. Petrov *et al.*, Sov. Tech. Phys. Lett. **6**, 165 (1980).
5. M. P. Petrov and A. Khomenko, Opt. Commun. **37**, 253 (1981).
6. M. P. Petrov *et al.*, Zh. Tekh. Fiz. **51**, 1422 (1981) [Sov. Phys. Tech. Phys. **26**, 816 (1981)].
7. D. Casasent, F. Caimi, and A. Khomenko, Appl. Opt. **20**, 3090 (1981).
8. D. Casasent, F. Cimi, and A. Khomenko, Appl. Opt. **20**, 4215 (1981).
9. B. H. Horwitz and F. C. Corbett, Opt. Eng. **17**, 353 (1978).
10. M. Petrov *et al.*, Ferroelectrics **22**, 651 (1978).
11. W. Roach, IEEE Trans. Electron. Devices **ED-21**, 453 (1974).
12. Y. Owechko and A. Tanguay, Proc. Soc. Photo-Opt. Instrum. Eng. **202**, 110 (1979).
13. Y. Owechko and A. Tanguay, Proc. Soc. Photo-Opt. Instrum. Eng. **218**, 67 (19809).
14. M. Petrov *et al.*, Mikroelektronika Akad. Nauk SSSR **8**, 20 (1979).
15. A. Khomenko *et al.*, Sov. Tech. Phys. Lett. **5**, 133 (1979).
16. J. Rao, J. Opt. Soc. Am. **57**, 798 (1967).
17. M. Petrov and A. Khomenko, Fiz. Tverd. Tela (Leningrad) **23**, 1350 (1981) [Sov. Phys. Solid State **23**, 789 (1981)].
18. M. Petrov, in *Current Trends in Optics, Proceedings, ICO-12* (Taylor and Francis, London, 1981), pp. 161–172.

Two of the authors (D.C. and F.C.) thank the Air Force Office of Scientific Research (grant 79-0091) for supporting our time during this study and for supporting publication of this paper. The other authors (M.P. and A.K.) thank the U.S.S.R. Academy of Sciences for research support of their ongoing Prom and Priz light modulator studies.

4. ADVANCED ACOUSTO-OPTIC SIGNAL PROCESSORS

352-13

ADVANCED ACOUSTO-OPTIC SIGNAL PROCESSORS

David Casasent

Carnegie-Mellon University
Department of Electrical Engineering
Pittsburgh, Pennsylvania 15213

ABSTRACT

The basic acousto-optic signal processing architectures (spectrum analyzer, space-integrating, time-integrating and triple product processor) systems and algorithms such as the chirp-Z transform are reviewed. We then describe new acousto-optic data processing systems and applications that utilize these basic architectures and new ones. These include a matched spatial filter acousto-optic processor, two new hybrid time and space-integrating systems, a triple product processor and four new matrix-vector iterative feedback systems.

1. INTRODUCTION

Acousto-optic (AO) transducers are commercially available real-time spatial light modulators that can be used for many applications and system architectures. Such devices and systems have received considerable attention [1]. In Section 2, we review several of the more basic AO system architectures. In Section 3, we discuss a multi-channel AO processor [2] architecture in which multiple signal channels are accommodated by multiple LEDs or laser diodes (LDs) rather than by a multi-channel AO cell. One of the major attractive reasons for using AO processors is the large processing gain (PG) and time bandwidth product (TBWP) possible on such systems. In Section 4, we discuss PG, define various possible output correlation plane SNR measures and provide experimental verification of our remarks [3]. We also discuss a technique to facilitate generation of the long codes for which AO systems are useful [4,5]. In Section 5, we describe several new AO processors: a matched spatial filter (MSF) AO correlator that reduces the effects of non-uniform spatial AO cell response variations [3], two new hybrid time and space-integrating (TSI) architectures that simultaneously achieve the best features of a space-integrating and time-integrating processor [4], and a new application [5] of the triple product processor. In Section 6, we address a major new application area for AO devices: optical systolic array matrix-vector and iterative optical processors. We describe the basic optical matrix-vector feedback system and note the general-purpose nature of such a processor. We then discuss two deconvolution architectures using such a concept, an optical systolic matrix-vector multiplier and an optical systolic system for solving matrix-vector equations [6]. We then conclude with a new frequency-multiplexed version of these systems [7] and our summary and conclusions (Section 7).

2. BASIC ACOUSTO-OPTIC SIGNAL PROCESSING ARCHITECTURES

Various reviews exist of the possibilities of optical signal processing systems and architectures [1,8-10]. Thus, this present discussion of the same subject will be quite brief. First, we note that an AO cell illuminated with parallel laser light diffracts this light at angles proportional to the frequencies present in the input signal and with the amplitude of each diffracted wave proportional to the strength of each input signal component. Thus, a lens placed behind an AO cell forms the Fourier spectrum of the input signal data in its back focal plane. This spectrum analysis feature of AO devices is presently being used in many applications [1]. One particularly attractive application uses an AO cell with two transducers or two separate AO spectrum analyzers, each fed with a signal from a different antenna pointed in a different direction. From the location of peaks in the two resultant output Fourier transform (FT) planes, the frequency distribution of emitters present in the field-of-view of the antennas can be obtained. From the ratio of the amplitudes of the output peaks on detectors corresponding to the same frequency component, estimates of the direction of each emitter can be obtained. The resultant output information from such a system can thus be used to obtain a 2-D (frequency/direction-of-arrival, or f/DOA) display of the emitters present in the field-of-view of the antennas [11]. Such displays are most useful for many signal processing applications.

The space-integrating (SI) AO correlator or convolver system [12] of Figure 1 is a basic AO processor architecture. In this system, the signal $g(t)$ is fed to an AO cell at plane P_{1A}, whose transmittance is a function of time t and space x given by $g(t-x/v_s) = g(t-x')$ or $g(x'-t)$, where v_s is the velocity of sound in the AO cell. This pattern is then imaged onto a mask $h(x')$ at plane P_{1B}. The light distribution $g(t-x')h(x')$ or $g(x'-t)h(x')$ is then spatially integrated by lens L_3 and the time-history output from the photodetector at P_3 is the correlation

$$h \circledast g = \int g(x' - t)h(x')dx' = R(\tau) \qquad (1a)$$

or the convolution

$$h * g = \int g(t - x')h(x')dx' = C(\tau) \qquad (1b)$$

of the two signals g and h. In this architecture, the integration in (1) is performed over space and the convolution or correlation output is displayed with the shift variable τ being time t as in (1). We thus refer to this architecture as a space-integrating convolver or correlator.

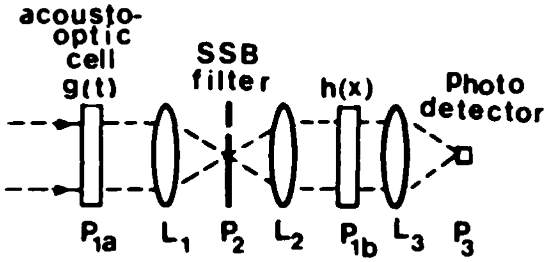


FIGURE 1 Schematic diagram of the basic space integrating acousto-optic convolver or correlator.

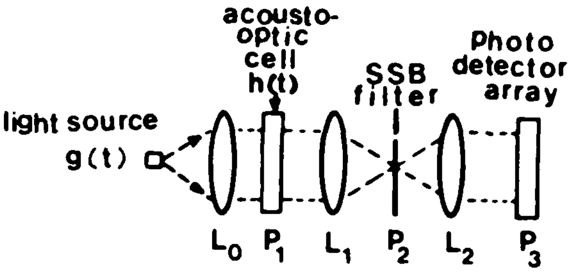


FIGURE 2 Schematic diagram of the basic time-integrating acousto-optic convolver or correlator.

The third basic AO processor architecture is the time-integrating (TI) convolver or correlator [13] of Figure 2. In this system, the signal g(t) is used to time-sequentially modulate the output from an LED, LD or point AO cell. This output uniformly illuminates an AO cell at P_1 fed with a signal h(t). With the transmittance of this AO cell described by h(x'-t) or h(t-x'), the light distribution leaving P_1 is the product of g(t) and the transmittance of the AO cell. The AO cell is then imaged onto a linear detector array at P_3, where time-integration of the product is performed directly on the detector. The resultant output at P_3 for this system is again either the correlation

$$g \circledast h = \int g(t)h(t - x')dt = R(\tau) \qquad (2a)$$

or the convolution

$$g * h = \int g(t)h(x' - t)dt = C(\tau). \qquad (2b)$$

Since the integration in (2) is performed in time, we refer to this as a time-integrating convolver or correlator and we note that the shift variable τ in the output correlation or convolution plane P_3 is the space variable x'.

The space-integrating system of Figure 1 enjoys a large range delay search, but can only operate on signals whose TBWP equals that of the AO cell (typically 1000 to 2000). Conversely, the TI system of Figure 2 can only search a range delay equal to the aperture time or transit time T_A of the AO cell (typically 1-50μsec), but can process a signal of long duration (>> T_A) and large TBWP and hence can provide a large PG and integration time T_I (limited by the dynamic range of the detector system used).

The fourth basic AO architecture is the triple product processor (TPP) system [14] (see Figure 7 in Section 5). In this system, a point light source uniformly illuminates one AO cell (oriented horizontally). The light distribution leaving this AO cell is compressed horizontally and expanded vertically to illuminate a second AO cell (oriented vertically). Both AO cells are then imaged onto an output plane P_3 where time-integration occurs on a 2-D detector array. If the signals to the point modulator and the two AO cells are denoted by u(t), v(t) and s(t), then the 2-D output at plane P_3 is the triple product

$$R(\tau_2,\tau_1) = \int_{T_I} u(t)v(t - \tau_1)s(t + \tau_2)dt, \qquad (3)$$

and hence the name TPP has been given to such an architecture. This system has considerable versatility, since its 2-D output plane pattern can be a folded spectrum, an ambiguity surface or (as we show in Section 5) a coarse/fine time delay display.

One particular algorithm that has provided each of these latter three AO architectures with an increased flexibility is the chirp-Z transform (CZT) algorithm [15]. We describe this algorithm with respect to the TI system of Figure 2. If the input signal g(t) is a signal f(t) multiplied by a chirp or LFM waveform and if the input signal h(t) to the AO cell is also a chirp and if the opposite sidebands of each chirp are filtered within the optical portion of the system, then the 1-D output pattern at P_3 can be shown to be the Fourier transform (FT) of the input signal f(t). This same technique can be applied to the TPP system to yield a 2-D FT output with coarse and fine frequency axes (i.e., a folded spectrum output display).

3. MULTI-CHANNEL ACOUSTO-OPTIC PROCESSING

There are various ways to achieve multi-channel and 2-D processing using 1-D AO devices. The use of crossed AO cells (as in the TPP system) is one technique. The use of AO cells with separate signal channels on the same cell is a second approach. In Figure 3, we show an alternate architecture [2] in which a single channel AO cell and a linear array of LEDs or LDs are used to achieve a multi-channel system. The input signal to the AO cell in this system of Figure 3 is g(t) and the transmittance of the cell is uniform along any vertical line. The input signals to the N LEDs are f_1 to f_N. The output pattern on the N 1-D detectors is then the N correlations $f_n \circledast g$. Such an architecture thus realizes N 1-D correlations of g(t) with the N input signals $f_n(t)$. If the CZT algorithm is used, the output pattern contains the N 1-D Fourier transforms $F_n(u)$ of the N input signals $f_n(t)$. Such an architecture is useful for imaging FT spectroscopy [2]. If the N input signals originate from N antennas pointed in N different directions, the output can also be an f/DOA display. This architecture is thus very attractive, since the same basic system can be used for multiple different applications (such as the ones mentioned above) simply by controlling the input electronic signals used.

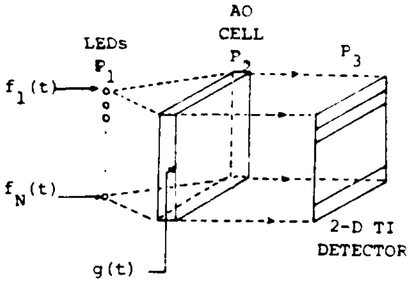


FIGURE 3 Simplified schematic diagram of a multi-channel acousto-optic general-purpose signal processor [2].

4. PROCESSING GAIN AND SNR CONCEPTS

One attractive feature of optical signal processors is their ability to accommodate signals of large duration and long T_I and large TBWP. The TI architectures are especially attractive for realizing such features. In such cases, a most attractive performance measure is PG. This is defined as the ratio SNR_O/SNR_i of SNR_O at the output of the processor to the SNR_i at the signal input to the system. In many cases, the signal code used is best described statistically, the signal is then a sample realization of a random process and we refer to the statistical correlation or the ensemble average (denoted by E) of two signals $s_1(t)$ and $s_2(t)$, i.e.

$$R(\tau) = E\{s_1(t)s_2(t + \tau)\}. \tag{4}$$

We have highlighted the PG and three different SNR_0 measures for optical systems in [3].

Let us first discuss the three different SNR_O measures possible. The classical SNR_O measure used is the ratio of the average peak intensity of the output correlation divided by the variance at the peak (assumed to lie at $\tau = t_d = 0$ for simplicity) averaged over many realizations of the process

$$SNR_1 = E^2\{C(0)\}/Var\{C(0)\}. \tag{5}$$

We refer to this as SNR_O at the peak. A second useful SNR_O measure is the peak-to-sidelobe ratio (PSR) or the value of the peak intensity divided by the variance of the correlation far from the peak

$$SNR_2 = \frac{E^2\{C(\tau)\}\big|_{\tau=0}}{Var\{C(\tau)\}\big|_{\tau>>0}} . \tag{6}$$

The SNR_O measure in (5) is difficult to measure experimentally since it requires evaluation at one point and thus requires many repetitions of the experiment. The SNR_O measure in (6) is easily obtained from one experiment. However, the dependence of this measurement on the size and location of the $\tau >> 0$ region used to estimate the variance in (6) can affect this SNR_O measure. Thus, a new SNR_O measure

$$SNR_3 = \frac{E^2\{C(0)\}}{(1/T_C)\int_{-T_C/2}^{T_C/2} Var\{C(\tau)\}d\tau} \tag{7}$$

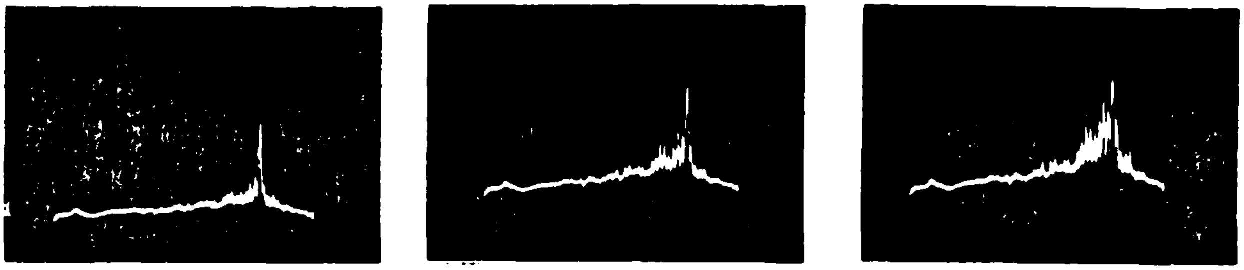
was defined, in which the variance is obtained for the full $T_C = T_A$ range of output delays being searched.

From an optical experimental standpoint, measurement of $SNR_2 = SNR_0$ is (6) is preferable. In [3], we showed that for the case of PRN signal with high noise (low SNR_i) that $SNR_1 = SNR_2$ and that for low noise (high SNR_i), $SNR_1 = \infty$ and SNR_2 = TBWP. In the practical low noise case, the simpler SNR_2 measure thus yields the same results as the more difficult SNR_1 measure of SNR_O. However, for the case of passive signals in which the recieved signal amplitude is continuous rather than discrete, the signal is well-modeled as Gaussian and for this case we showed [3] that the numerators in (5) and (6) will be identical and that for such Gaussian signals $SNR_1 = SNR_2$ - 3dB (if $SNR_i = \infty$, i.e. no noise). For the case of Gaussian signals and $SNR_i = 0$ (equal amounts of signal and noise), we find $SNR_2 = SNR_1$. We conclude by noting that SNR_1 is the more appropriate measure for communications (when the processor is in synchronization and when location or position of the correlation peak is known). Conversely, SNR_2 is more appropriate for parameter estimation (such as when the location or time delay of the correlation peak is desired).

In [5], we discussed how the bandwidth of the noise with respect to the bandwidth of the signal code affected PG and we noted that for narrowband noise PG = TBWP (of the signal) or $T_S B_S$ and for wideband noise PG = ITBWP (of the system) or $T_I B_S$. This distinction is important when the code used is repeated and when integration is performed over repeated versions of the code ($T_I >> T_S$). As we have shown [3,5] in such cases, the PG obtained will be less (i.e. PG = TBWP) if the noise is narrowband than if it is wideband (in this case PG = ITBWP >> TBWP). To demonstrate that PG = TBWP rather than ITBWP for the case of narrowband noise and to demonstrate the use of our SNR_2 measure of SNR_O, we [3] used a signal with a TBWP = 2047. We repeated this signal approximately 150 times and integrated the output on the TI system of Figure 2 for the full 150 cycles of the signal code. The TBWP = 2047 corresponds to a PG = 33dB and the ITBWP of approximately 300,000 corresponded to a PG = 55dB.

In Figure 4, we show the results of correlations of this signal with narrowband jammers of different strengths (different SNR_i levels). The results obtained verified that the PG of such a system is TBWP not ITBWP and it also verified that our SNR_2 measure gave adequate results that were more easily obtained in the laboratory. These experiments and our associated theory show the need for long codes and processors capable of coherent integration of codes of long duration and large TBWP. Many applications require such performance together with the need to search large range delays between the received and transmitted signals. In Section 5, we discuss several such processors and we also discuss ways to improve the uniformity of the correlation plane outputs shown in Figure 4.

A final important and practical aspect associated with the realization of a correlation of a code with a very large TBWP is the code generator itself. A very attractive technique for generation of a code with a large TBWP is to produce two codes $\mathcal{U}$ and $\mathcal{V}$ with N_u and N_v bits respectively and to repeat each code N_v and N_u times respectively in two separate signal generators. If we then form the bit-by-bit product of each code (Figure 5), we obtain a product code with a period equal to $N_sT_B = N_uN_vT_B$ or the product of the durations of each code (where T_B is the duration of one bit of the code).



(a) $SNR_i = +\infty$ (b) $SNR_i = -20dB$ (c) $SNR_i = -25dB$

FIGURE 4 Photographs of several correlation plane output patterns from the time-integrating correlator of Figure 2 demonstrating the processing gain and SNR measurement techniques. The signal used was a pseudorandom code with various narrowband noise jammers present [3].

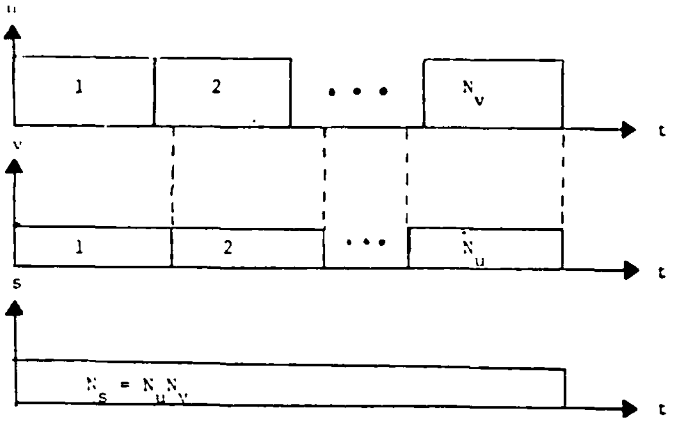


FIGURE 5 Simplified representation of the generation of a product code [4].

5. NEW ACOUSTO-OPTIC ARCHITECTURES AND APPLICATIONS

In this section, we briefly discuss four new AO architectures and applications. First, we recall the conventional MSF optical correlator [16] redrawn as in Figure 6. If we place an AO cell in the input plane and an MSF of the reference signal at plane P_2, and in the output use a parallel output shift and summation detector system (detailed below), we obtain a new hybrid time and space-integrating (TSI) architecture that we refer to as an AO MSF correlator [3]. Its operation, performance, architecture and features differ considerably from those of the prior AO systems. First recall that an MSF correlator is shift-invariant, i.e. the location of the output correlation peak corresponds to the location of the signal in the input plane. If an AO cell is placed at the input plane of an MSF correlator (as shown in Figure 6), then the output correlation peak will move across the output linear detector array as the input signal traverses the AO cell. At each detector element at each time instant, we have a partial correlation with the full PG of the TBWP of the AO cell. We detect one such correlation output pattern and the output pattern at the next time instant (the correlation peak will now lie on an adjacent detector). We delay these two detector outputs by one bit time, add them and repeat the process for N bit times. We then obtain the noncoherent summation of the N partial correlation outputs, each with a PG = TBWP of the AO cell. Thus, such a system yields a large PG. But also, since each correlation output is obtained for a different location of the signal in the AO cell, the final correlation output has the effects of spatial non-uniformities in the AO cell removed (or integrated over the aperture of the cell). We have demonstrated the use of such a system [3] for an AO cell with large dead areas in which negligible acoustic diffraction fields are present. The results were very attractive.

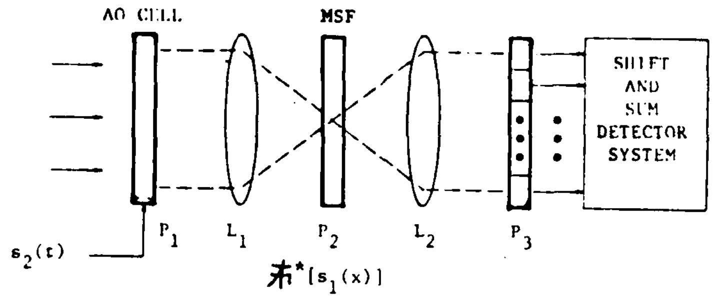


FIGURE 6 Schematic diagram of a matched spatial filter acousto-optic processor [3].

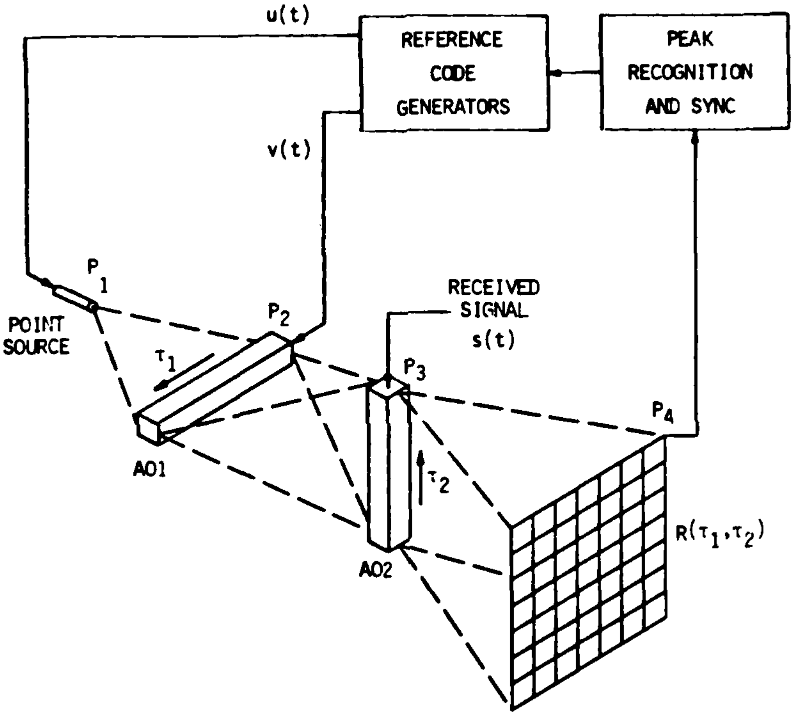


FIGURE 7 Schematic diagram of the triple product processor used for synchronization and demodulation of product codes [5].

Next, we note two other new TSI architectures. The first system uses an AO cell imaged onto a spatial mask with the FT of the resultant signal produced being multiplied by a second spatial mask on which the MSF of another signal was recorded. We have shown [4] that the output from such a system is the correlation of a large code equal to the product of the codes recorded on the two spatial masks. We have also modified this system to utilize a 2-D mask and a different vertical output detector system to achieve similar results with more flexibility in the code that can be used [4]. We have analyzed the noncoherent and coherent correlations obtained from such TSI architectures and verified that the desired envelope detection is obtained with the full PG possible [4].

As our fourth new AO application, we consider the TPP system for processing a repeated product code to obtain the full PG of the code and a large range delay search. In Figure 7, we show how this can be achieved. The received signal s(t) and the two reference product code signals $\hat{u}(t)$ and $\hat{v}(t)$ are fed to the elements of the TPP as shown. We have shown [5] that the output plane pattern for such a system contains axes that correspond to coarse and fine time delays and that the location of the correlation peak thus indicates the target's range to the full range resolution possible with the code. In addition, the PG of the system and hence the SNR of the output correlation peak corresponds to the TBWP of the full

352-13

product code. This can approach 10^6 in some cases.

6. ACOUSTO-OPTIC MATRIX-VECTOR PROCESSORS

In this section, we address a new class of AO processors that differ considerably from the dedicated architectures described in Sections 2 - 5. This new class of AO processor is very general-purpose in that all of the systems to be described perform matrix-vector operations. First, we introduce the basic matrix-vector multiplication concept together with the use of an iterative feedback optical processor [17]. Next, we consider the realization of a deconvolver with such an architecture using the AO systems of Figures 1 and 2 as the basic system elements [6]. This corresponds to a matrix-vector problem in which the matrix is Toeplitz. As our second AO matrix-vector architecture, we consider a systolic array optical processor using AO devices and its use in the multiplication of banded matrices and vectors of large dimension. We also consider its use in the solution of large matrix-vector problems when the matrix is banded. Finally, we conclude with a new iterative matrix-vector architecture using a frequency-multiplexed AO systolic array architecture and discuss its use for the solution of more complicated matrix problems.

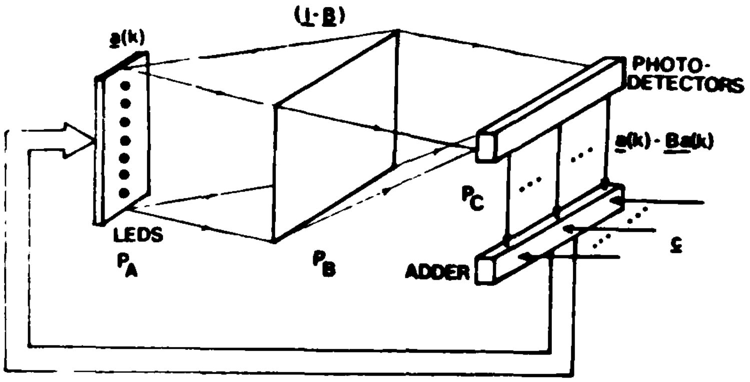


FIGURE 8 General optical matrix-vector feedback system (using a 2-D mask) [17].

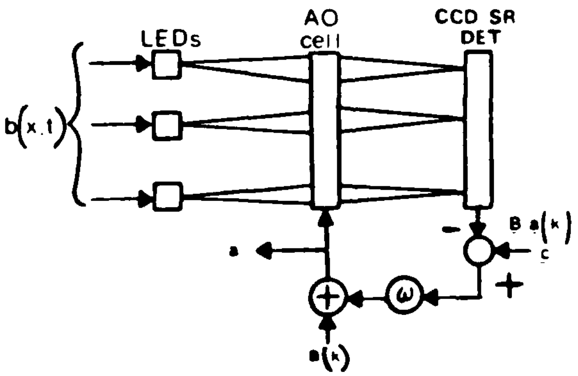


FIGURE 9 Acousto-optic systolic array iterative matrix-vector processor architecture [6].

In Figure 8, we show a general matrix-vector multiplication system (using a 2-D mask) as well as the iterative feedback architecture version of the same system. We restrict our attention to the P_A - P_B - P_C portion of the system for the moment. We image the outputs from the linear LED or LD input array vertically and expand the output from each horizontally to uniformly illuminate the rows of a 2-D mask at P_B. We then integrate the light leaving the separate columns of the P_B mask onto separate output detectors at P_C. If we describe the output from the LEDs as a vector $\underline{a}$ and the transmittance of the mask as a matrix $(\underline{I} - \underline{B})$, where $\underline{I}$ is the identity matrix, the detector outputs at P_C are seen to be the matrix-vector product $[\underline{I} - \underline{B}]\underline{a}$ and a parallel optical matrix-vector processor results. If we then add an external vector $\underline{c}$ to this matrix-vector product (and denote the vector output at iteration k by $\underline{a}(k)$) and then feed this vector sum back to the LED inputs as the input $\underline{a}(k+1)$ at iteration k + 1, the system is described by

$$(\underline{I} - \underline{B})\underline{a}(k) + \underline{c} = \underline{a}(k) - \underline{B}\,\underline{a}(k) + \underline{c} = \underline{a}(k+1). \quad (8)$$

In steady state, when $\underline{a}(k) = \underline{a}(k+1) = \underline{a}$, equation (8) reduces to $\underline{B}\,\underline{a} = \underline{c}$ and the output vector $\underline{a}$ is the solution

$$\underline{a} = \underline{B}^{-1}\underline{c} \quad (9)$$

of the matrix-vector equation $\underline{B}\,\underline{a} = \underline{c}$. Many very attractive applications forsuch an iterative optical matrix-vector processor have been described and demonstrated by us.

For our present purposes, we consider the realization of similar operations using AO devices that are more reliable and commercially available than 2-D spatial light modulators. We begin by considering realization of the matrix-vector product $\underline{B}\,\underline{a} = \underline{d}$, when $\underline{B}$ is tridiagonal, i.e.

$$\begin{bmatrix} b_{11} & & & & & \bigcirc \\ b_{21} & b_{22} & & & & \\ b_{31} & b_{32} & b_{33} & & & \\ & b_{42} & b_{43} & b_{44} & & \\ & & b_{53} & b_{54} & b_{55} & \\ & \bigcirc & & & & \ddots \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \\ \vdots \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \\ \vdots \end{bmatrix} . \quad (10)$$

Consider the system of Figure 9 used to realize the matrix-vector product in (10). In the system of Figure 9, the output from separate LEDs are imaged through separate regions of a AO cell and onto separate detector elements in the output plane. In [6], we showed that if the time-history of the three diagonals of $\underline{B}$ are fed to three input LEDs and the time-history of the vector $\underline{a}$ is fed to the AO cell and if the detector outputs are properly shifted and summed, then the time-sequential detector output is the desired matrix-vector product $\underline{B}\,\underline{a} = \underline{d}$ in (10). Addition of an external vector $\underline{c}$ and feedback of the detector outputs back to the AO cell inputs enables this system to solve matrix-vector equations such as $\underline{B}\,\underline{a} = \underline{c}$ for $\underline{a} = \underline{B}^{-1}\underline{c}$. This is indicated [6] in Figure 9 together with the use of a slightly different and more stable iterative algorithm than the one described in (8). This systolic array architecture of Figure 9 is a very practical and general-purpose AO matrix-vector processor.

Many AO processor architectures can be used as the basic element in the system of Figure 9. If the SI or TI convolver of Figures 1 or 2 were used, then the output vector would be the deconvolution of the received signal. This can be seen by writing the convolution as a matrix-vector multiplication where the matrix is Toeplitz as was done in [6]. Similarly, various difference equations can be solved on the system of Figure 9 by first writing the derivatives in the equation as finite differences as we described in [6].

Next, we consider a new extension of this AO systolic matrix-vector processor concept. First, we note that the bandwidth and TBWP of the AO cell are not fully used in the system of Figure 9 and that the detector shift and summation operations required are not attractive. In our new optical systolic matrix-vector AO processor of Figure 10 [7], we use a linear array of LEDs or LDs at the input and again image them through selected regions of a AO cell. However, now we form the FT of the product information leaving the AO cell and we detect the outputs on a linear output detector array with parallel readout. Into the LEDs, we feed a matrix properly arranged with its columns and rows multiplexed as functions of space and time. Into the AO cell, we feed another matrix with its columns and rows multiplexed as functions of time and frequency. As we have shown [7], the time-sequential output as a function of time and space from the output detectors can be obtained in parallel and that it corresponds to the matrix-matrix product. Note that in this system, no output detector shift and summation networks are required (as was the case in Figure 9). We also note that the time-history outputs from the detectors are available for direct feedback immediately into the LED inputs or the AO cell input for iterative operations. We have extended this basic concept to the solution of matrix-matrix equations, the multiplication of three matrices, matrix inversion, etc. [7] following the basic techniques used in our prior optical systolic array matrix-vector publications. We have also addressed the pipelining of data and operations in such systems. From the brief general remarks advanced above, we see that this new and most general-purpose type of optical matrix-vector systolic array processor architecture research has many diverse applications and that it represents a major new area of application research for AO devices.

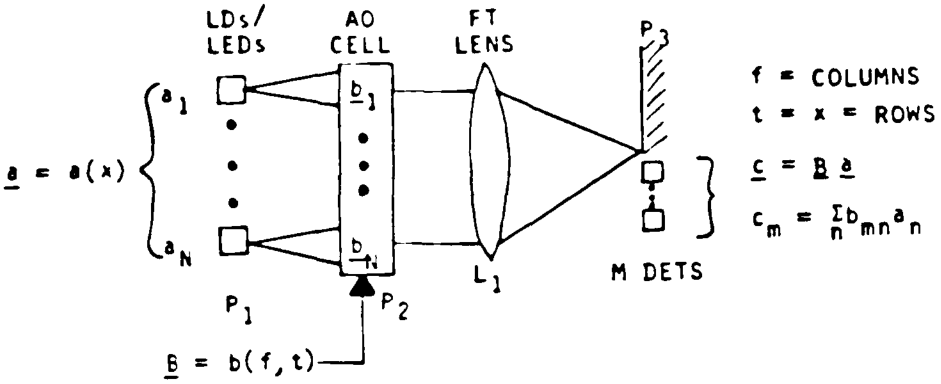


FIGURE 10 New frequency-multiplexed acousto-optic iterative matrix-vector systolic array processor architecture [7].

7. SUMMARY AND CONCLUSION

In these brief pages, we have reviewed the conventional AO processors that realize the Fourier transform, convolution and correlation operations. We have also described ten new AO processor architectures. These have used combined hybrid time and space-integrating concepts, 2-D output patterns, multi-channel architectures, and general-purpose matrix-vector systolic array concepts. From this brief summary of new and advanced AO architectures, it appears clear that these devices have a most attractive place in future signal processing applications and that they can serve as very general-purpose processors as well as being able to easily be configured into unique architectures for specific applications.

ACKNOWLEDGMENT

The support of various portions of this research by the National Science Foundation (Grant ECS-8114344), NASA Lewis (NAG 3-5) and contractors of USI are gratefully acknowledged. The author thanks Giora Silbershatz for the results included in Sections 4 and 5 and Mark Carlotto and James Jackson for some of the results included in Section 6.

REFERENCES

1. Proc. IEEE, Special Issue on "Acousto-Optic Bragg Diffraction", 69 (January 1981) and: SPIE, 154 (August 1978); SPIE, 180 (April 1979); SPIE, 241 (July 1980); and SPIE, 298 (August 1981).
2. D. Casasent and D. Psaltis, Applied Optics, 19, 2034 (1980).
3. D. Casasent, G. Silbershatz and B. Kumar, Applied Optics, 21, 2356 (1982).
4. G. Silbershatz and D. Casasent, IEEE, AES (submitted).
5. D. Casasent and G. Silbershatz, Applied Optics, 21, 2076 (1982).
6. D. Casasent, Applied Optics, 21, 1859 (1982).
7. D. Casasent, J. Jackson and C. Neuman, Applied Optics (submitted).
8. D. Casasent, EOSD, 39-46 (June 1981).
9. D. Casasent, EOSD, 41-47 (September 1981).
10. D. Casasent, IEEE, Communications Magazine, pp. 40-48 (September 1981).
11. R. Coppock and R. Croce, SPIE, 214, 124 (1979).
12. R. Sprague, Optical Engineering, 16, 467 (1977).
13. R. Sprague and C. Kolopoulos, Applied Optics, 15, 89 (1976).
14. P. Kellman, PhD Thesis, Stanford University (1979).
15. H. Whitehouse et al, SPIE, 154, 66 (1978).
16. A. Van der Lugt, IEEE, IT-10, 139 (1964).
17. M. Carlotto and D. Casasent, Applied Optics, 21, 147 (1982).

5. FREQUENCY-MULTIPLEXED AND PIPELINED ITERATIVE OPTICAL SYSTOLIC ARRAY PROCESSORS

Reprinted from **Applied Optics**, Vol. 22, page 115, January 1, 1983

Frequency-multiplexed and pipelined iterative optical systolic array processors

David Casasent, James Jackson, and Charles Neuman

Optical matrix processors using acoustooptic transducers are described with emphasis on new systolic array architectures using frequency multiplexing in addition to space and time multiplexing. A Kalman filtering application is considered as our case study from which the operations required on such a system can be defined. This also serves as a new and powerful application for iterative optical processors. The importance of pipelining the data flow and the ordering of the operations performed in a specific application of such a system are also noted. Several examples of how to effectively achieve this are included. A new technique for handling bipolar data on such architectures is also described.

I. Introduction

Optical processors that realize matrix–vector[1] and matrix–matrix[2] multiplications represent a most attractive and rather general-purpose optical processor concept. Iterative optical matrix–vector processors constitute an even more powerful version of these basic systems.[3] However, the practical realization of these systems requires real-time and reusable 2-D spatial light modulators.[4] Since electrically addressable 2-D spatial light modulators of adequate performance are not presently readily available, attention has recently shifted to systems using acoustooptic (AO) transducers that realize matrix–vector multiplications[5] and iterative optical matrix–vector operations.[6]

These recent architectures[5,6] are optical systolic array processors. However, these initial architectures have not yet fully utilized the frequency-bandwidth and time-bandwidth products possible in the AO devices.[7] In Sec. II we describe several basic frequency-multiplexed optical systolic array architectures that more fully utilize the signal capacity of the system and that require simpler output detection systems than did the original architectures proposed. In Sec. IV we include advanced AO systolic array optical processor architectures with attention to more complex operations and with attention to the throughput of the resultant system. A new basic optical systolic array processor architecture using a Fourier transform output plane with the simultaneous addition of multiple outputs on a linear detector array with parallel readout is presented.

Since different data processing problems require different operations, we select a specific case study and thus in Sec. III introduce and describe a new application (Kalman filtering) for implementation on an optical systolic array processor. This case study (Sec. III) is used to delineate the basic operations required on our system. It is important to properly pipeline the data flow and operations in a parallel processor to fully realize the parallel processing capability of such a system and to determine the basic processing elements or functions into which a specific problem should be formulated for proper implementation on such a system. In Sec. IV we address this issue with the operations determined from our Kalman filtering problem as our case study. The new optical systolic array processor architectures presented realize matrix–matrix multiplication, matrix–matrix–matrix multiplication, as well as matrix inversion. Previous iterative optical processors required various operations in the feedback loop. These are usually implemented digitally and can hence appreciably slow down the cycle time of the system. In particular, bipolar operations on these systems require the addition and subtraction of data from one or subsequent cycles and hence require A/D and D/A converters plus digital data storage and addition. In Sec. V we advance a new technique to accommodate bipolar data on such noncoherent optical processors.

The authors are with Carnegie-Mellon University, Department of Electrical Engineering, Pittsburgh, Pennsylvania 15213.

Received 30 August 1982.
0003-6935/83/010115-10$01.00/0.

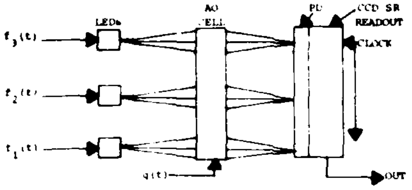


Fig. 1. Original optical systolic matrix vector multiplier system (after Refs. 5 and 6).

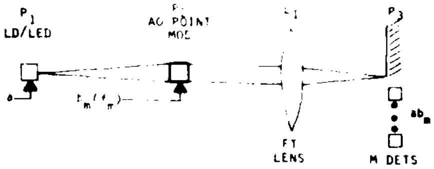


Fig. 2. Simplified frequency-multiplexed optical systolic matrix vector multiplier using a point acoustooptic modulator.

II. Basic Frequency-Multiplexed Optical Systolic Array Processors

By way of review we show the prior[5,6] optical systolic system in Fig. 1. This system achieves the multiplication of a banded matrix **B** of bandwidth three by a vector **a**; e.g., $\mathbf{B}\cdot\mathbf{a}=\mathbf{d}$, or

$$\begin{bmatrix} b_{11} & & & & & \\ b_{21} & b_{22} & & & & \\ b_{31} & b_{32} & b_{33} & & & \\ & b_{42} & b_{43} & b_{44} & & \\ & & b_{53} & b_{54} & b_{55} & \\ & & & & & \cdot \\ & & & & & & \cdot \\ & & & & & & & \cdot \\ & & & & & & & & \cdot \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \\ \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \\ \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}. \tag{1}$$

Without loss of generality, we have chosen the case in which the main and the two lower diagonals of the matrix contain the only nonzero elements of the banded matrix. To achieve this operation, the time-sequential input signals to the three LEDs (or laser diodes) and the acoustooptic cell are

$$\begin{aligned} f_1(t) &= \ldots \quad 0 \quad b_{42} \quad 0 \quad b_{31} \quad 0 \quad 0 \\ f_2(t) &= \ldots \quad b_{43} \quad 0 \quad b_{32} \quad 0 \quad b_{21} \quad 0 \\ f_3(t) &= \ldots \quad 0 \quad b_{33} \quad 0 \quad b_{22} \quad 0 \quad b_{11} \\ g(t) &= \ldots \quad 0 \quad a_3 \quad 0 \quad a_2 \quad 0 \quad a_1, \end{aligned} \tag{2}$$

where time increases from right to left and where the bit duration for each element of **B** and **a** is T_B. As was shown in Refs. 5 and 6, if the detector outputs are shifted down at a clock rate $1/T_B$ and if successive products are summed in the detector's CCD shift register readout channel, the time-history output from the detector system will be the desired matrix-vector product **d**.

We have examined various alternate arrangements of the matrix input data **B** to the linear LED arrays to avoid the need for a shift register summation readout system. Such an approach is attractive, since quite drastically improved dynamic range will result if the original architecture of Fig. 1 is modified and if proper format control of the input data is used.

The system in Fig. 1 uses a 1-D input array of LEDs or laser diodes and a 1-D AO transducer. The philosophy for realization of an optical matrix-vector product on such a system was to represent the 2-D matrix **B** by the time and space variables of the LED array, i.e., we represent the elements b_{mn} of **B** by $b(x,t)$. This time and space multiplexing of the input data is quite attractive since the LED input data must be provided in parallel from some source. As noted above, this system uses the AO cell only to shift the input vector data **a**. It does not use the large frequency-bandwidth and time-bandwidth products possible in AO cells.

To alleviate these problems and to produce a more powerful optical systolic processor, we frequency multiplex the input data. In Fig. 2 we show a simple initial system to demonstrate the architecture and concept. We consider the multiplication of a vector with elements b_m by a constant a. In Fig. 2 we show a point AO modulator (an AO cell with only one spatial resolution element or equivalently an AO cell with the input light source imaged through only 1 pixel of the cell). As the input to the AO cell, we place each b_m on a corresponding frequency carrier f_m with intensity modulation proportional to b_m and sum these M terms to form *one simultaneous input signal* $b_m(f_m)$ to the cell at 1-bit time T_B. Lens L_1 now forms (at the output plane P_3) the Fourier transform of the light distribution leaving the point modulator. For simplicity we ignore the bandpass frequency response of the AO cell, and we show the system for the case of an AO cell bandwidth $M\Delta f$ extending from Δf to $M\Delta f$, where Δf is the frequency resolution of the system (corresponding to the used aperture of the AO cell, i.e., the bit time $T_B = 1/\Delta f$). At P_3, we place M detector elements in a linear array at spatial locations corresponding to these M frequency components. At the instant that the input signal $b_m(f_m)$ is present in the point modulator, we pulse on the LED or laser diode with an output light intensity a. This light intensity multiplies the signal in the AO cell and M waves leave the cell (at angles proportional to f_m). These M wave fronts are focused onto the M output detectors where we obtain the M products ab_m in parallel.

Let us next consider the use of the system in Fig. 2 to realize the general matrix-vector product $\mathbf{Ba}=\mathbf{c}$ or

$$\begin{bmatrix} b_{11} & \ldots & b_{1N} \\ \cdot & & \cdot \\ \cdot & b_{mn} & \cdot \\ \cdot & & \cdot \\ b_{M1} & \ldots & b_{MN} \end{bmatrix} \begin{bmatrix} a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix} = \begin{bmatrix} c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_M \end{bmatrix}. \tag{3}$$

To realize this matrix-vector multiplication in the system of Fig. 1, we partition the matrix **B** into column vectors $\mathbf{b}_n$ and write Eq. (3) as

$$[\mathbf{b}_1 \ldots \mathbf{b}_n \ldots \mathbf{b}_N]\mathbf{a} = \mathbf{c}. \tag{4}$$

To realize $\mathbf{Ba}=\mathbf{c}$ in the form of Eq. (4) in the system of Fig. 2, we time multiplex the elements a_n of **a** as sequential LED inputs, and we time multiplex each $\mathbf{b}_n$

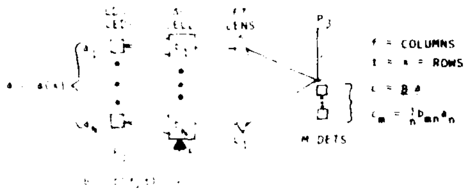


Fig. 3. Basic space-, time-, and frequency-multiplexed optical systolic matrix-vector processor system.

and frequency multiplex the M elements of each $\mathbf{b}_n$ as the AO cell inputs. Specifically, $\mathbf{b}_1$ is the AO cell input at time T_B (with the elements of $\mathbf{b}_1$ frequency-multiplexed, $\mathbf{b}_2$ is the AO cell input at time $2T_B$, etc.). We choose to frequency multiplex the components of each column vector $\mathbf{b}_n$ in Eq. (4) since this arrangement allows us to multiply the first element a_1 of $\mathbf{a}$ by all the required elements of $\mathbf{B}$ (e.g., $\mathbf{b}_1$) at time T_B. At time $2T_B$, we multiply a_2 by all the necessary elements $\mathbf{b}_2$ of $\mathbf{B}$, etc.

Several other combinations of input data sequences and different time- and frequency-multiplexing arrangements are also possible. Frequency multiplexing of the input LED or laser diode (or replacing it by a second point AO light modulator) are two of many different possibilities. We selected the choice noted above for matrix-vector multiplication, since all M components of $c_m = \sum_n b_{mn} a_n$ are formed at the proper M output detector element. Each $b_{mn} a_n$ product is formed at a different (nT_B) time interval, and thus we can sum the proper $b_{mn} a_n$ products by time integration directly on each of the M output detectors. This architecture does not require shifting the contents of the output detectors and summing the proper shifted contents of these detector elements (as was needed in earlier systems).[5,6] In the system of Fig. 2, after NT_B time intervals, the outputs from the M detectors at plane P_3 are the proper elements c_m of the matrix-vector product $\mathbf{Ba} = \mathbf{c}$.

The arrangement in Fig. 2 utilizes the bandwidth of the AO cell but neither its full potential time aperture nor its time-bandwidth product. In Fig. 3 we show a preferable topology for many cases. In this architecture, the input vector $\mathbf{a}$ is space-multiplexed rather than time-multiplexed and the matrix $\mathbf{B}$ is frequency- and time-multiplexed as before. In this system, after NT_B time intervals, the contents of the AO cell are as shown in Fig. 3. At this time the LEDs are pulsed on with the outputs shown in Fig. 3. This NT_B time delay has enabled the AO cell to convert the time- and frequency-multiplexed $b(f,t)$ representation of the matrix $\mathbf{B}$ into the indicated frequency- and space-multiplexed arrangement shown. When the LEDs are pulsed on as indicated, the full matrix-vector product in Eq. (3) or Eq. (4) is generated in one T_B time period in parallel on the linear output detector array.

These architectures (Figs. 2 and 3) are attractive and preferable to the original architecture of Fig. 1 because of the increased flexibility they provide and because of the increased number of computations they perform in any time period (this is necessary for such systems to become competitive with digital systolic array processors). The system of Fig. 2 performs one scalar-vector product every T_B and automatically adds the proper N such products on the correct detectors. It requires NT_B of time to produce the product of an $M \times N$ matrix by an N-dimensional vector. The system of Fig. 3 performs M vector inner products of the mth row of $\mathbf{B}$ and the column vector $\mathbf{a}$ in parallel and displays each of the M elements of the matrix-vector product $\mathbf{Ba}$ on M spatially separated output detectors in one T_B of time (after an initial NT_B delay time).

It is also possible to frequency multiplex each of the LED outputs in Fig. 3 (and thus input the matrix to the LEDs rather than the AO cell) or to combine frequency multiplexing of both the LEDs and the AO cell. In this discussion we do not consider all possible combinations of space, time, and frequency multiplexing of the different elements of the system. The present arrangement in Fig. 3 is the preferred architecture for realizing matrix-vector products (in Sec. IV we show that this system can also perform many other operations). We note that the roles of the matrix and the vector are reversed in this system (compared with the one in Fig. 1) with the matrix represented by frequency and time multiplexing being fed to the AO cell rather than the input LED array. In Sec. IV we consider iterative matrix algorithms as well as the pipelining and data flow of the system. In such applications we will find it preferable to reverse the matrix coding used to facilitate data flow and operational pipelining. For now, our initial concern is to convey the concept of frequency multiplexing and the new general optical systolic array Fourier transform architecture of Fig. 3.

If all N components of $\mathbf{a}$ are available in parallel, the system of Fig. 3 is preferable. If this is not the case, the system of Fig. 2 is preferable. We note that, although the system in Fig. 3 produces the full matrix-vector product in parallel, a time delay of NT_B is required before this output is obtained (this is necessary to properly format the contents of the AO cell, using it as a time delay element). In Sec. IV we discuss how proper pipelining of data and operations can avoid this delay. We do not consider the system of Fig. 2 further since its potential is quite limited.

The locations of the detector elements in these frequency-multiplexed architectures are easily calculated. For the RF frequency f_m associated with row m of the matrix $\mathbf{A}$, the spatial frequency of the signal in the cell is $u_m = f_m/v_s$, where v_s is the velocity of sound in the AO cell. The corresponding location of this f_m frequency component in the output P_3 plane is a distance $d_m \simeq u_m \lambda f_L/n = f_m \lambda f_L/n v_s$ below the central point of the output Fourier transform plane (where λ is the wavelength of light, n is the index of refraction, and f_L is the focal length of the Fourier transform lens).

To fully utilize the architecture of Fig. 3, we consider the realization of more complex basic operations beyond the simple matrix-vector multiplication considered thus far (e.g., matrix-matrix products, matrix-matrix-matrix products, and matrix inversions). We believe that the reduction and description of a specific data processing problem to such more complex fundamental operations represent the best utilization of such a parallel architecture. To detail such operations, a specific application is necessary. In Sec. III we describe a new and most powerful general matrix processing problem for such systems (Kalman filtering). A discussion of the realization of several of the matrix operations required in Kalman filtering using the architecture of Fig. 3 follows in Sec. IV, where we also address the associated pipelining and flow of data and operations in such a parallel processor.

III. Kalman Filtering on an Iterative Optical Systolic Array Processor

Our earlier optical matrix-vector research on optimal control[8,9] used an iterative optical processor with a 2-D spatial light modulator mask. In such applications, the refresh time (30 msec, typically) of 2-D light modulators restricted the algorithms we could consider to those which required very few changes of the 2-D mask. Our original optimal control problem involved the solution of the algebraic matrix Ricatti equation to calculate the feedback gains and optimal controls to be applied. Our proposed solution[8,9] involved a double-nested iterative algorithm with an inner and outer iterative loop. On each outer iterative loop, the mask (the 2-D matrix) had to be altered. Our new architecture in Fig. 3 removes the restriction of considering only algorithms that require a minimum number of changes of the matrix mask. Moreover, our prior optimal control applications were restricted to the case when we assumed that all states in our system model were observable and that the noise statistics were known in advance. In practice, this is not the case. Rather, we must estimate many of the states in the control system (and we must also update our estimates of the noise statistics after each new input measurement). A solution to these problems is the discrete-time Kalman filter. This is an optimal filter in the minimum mean-square sense.

In Table I we summarize (without derivation) the basic discrete-time Kalman filtering equations.[10] All control parameters used are defined in Table II. Equations (1a) and (1b) in Table I model the system as a linear vector difference equation. In our discrete-time system model, we assume that the system noise vector **w** and the noise vector **v** associated with measurement k are Gaussian distributed and of zero mean, as in Eqs. (1c) and (1d), and that they are uncorrelated as defined in Eq. (1e). The initial state estimate $\bar{\mathbf{x}}_0$ and the initial error covariance matrix $\mathbf{M}_0$ in Eq. (1f) complete our system model. The basic operations required in Kalman filtering and state estimation involve an iterative algorithm (with the iteration index k denoting different time samples). The actual sampling times kT_s are equally spaced. We assume for now that the noise statistics (**Q**,**R**) are known *a priori*. The system model (**Φ**,**Γ**,**H**) is assumed to be known. Just prior to time kT_s, we know the state $\mathbf{x}_k$ of the system and the error covariance $\mathbf{M}_k$ from the time update Eqs. (1j) and (1k). We also know (or can precompute if the noise statistics are known in advance) the matrices $\mathbf{P}_k$ and $\mathbf{K}_k$ in Eqs. (1g) and (1h). At time kT_s, we make a measurement $\mathbf{z}_k$. We then calculate the new state estimate $\hat{\mathbf{x}}_k$ from $\mathbf{z}_k$ as in Eq. (1i). This estimate is the expected value of $\mathbf{x}_k$ given all the measurements from $\{\mathbf{z}_k\}$ to $\mathbf{z}_k$. We then calculate the extrapolated state estimate $\bar{\mathbf{x}}_{k+1}$ as in Eq. (1j). This is based on the system model and our $\hat{\mathbf{x}}_k$ estimate. If $\mathbf{Q}_k$ and $\mathbf{R}_k$ are known as assumed, $\mathbf{P}_k$, $\mathbf{K}_k$, and $\mathbf{M}_{k+1}$ can be precomputed in advance from Eqs. (1g), (1h), and (1k).

Table I. Kalman Filtering Algorithm

DESCRIPTION	DEFINING EQUATIONS
System Model	(1a) $\underline{x}_{k+1} = \underline{\Phi}_k \underline{x}_k + \underline{\Gamma}_k \underline{w}_k$
	(1b) $\underline{z}_k = \underline{H}_k \underline{x}_k + \underline{v}_k$
	(1c) $\underline{w}_k \sim N(\bar{\underline{w}}_k, \underline{Q}_k)$
	(1d) $\underline{v}_k \sim N(0, \underline{R}_k)$
	(1e) $E[\underline{w}_k \underline{v}_j^T] = 0$ for all k and j
Initial Conditions	(1f) $\bar{\underline{x}}_0$ and $\underline{M}_0$
<u>Measurement Updates</u>	
Error Covariance	(1g) $\underline{P}_k = (\underline{M}_k^{-1} + \underline{H}_k^T \underline{R}_k^{-1} \underline{H}_k)^{-1}$ $= \underline{M}_k - \underline{M}_k \underline{H}_k^T (\underline{H}_k \underline{M}_k \underline{H}_k^T + \underline{R}_k)^{-1} \underline{H}_k \underline{M}_k$
Kalman Gain Matrix	(1h) $\underline{K}_k = \underline{P}_k \underline{H}_k^T \underline{R}_k^{-1}$
State Estimate	(1i) $\hat{\underline{x}}_k = \bar{\underline{x}}_k + \underline{K}_k (\underline{z}_k - \underline{H}_k \bar{\underline{x}}_k)$
<u>Time Updates</u>	
Extrapolated State Estimate	(1j) $\bar{\underline{x}}_{k+1} = \underline{\Phi}_k \hat{\underline{x}}_k + \underline{\Gamma}_k \bar{\underline{w}}_k$
Extrapolated Error Covariance	(1k) $\underline{M}_{k+1} = \underline{\Phi}_k \underline{P}_k \underline{\Phi}_k^T + \underline{\Gamma}_k \underline{Q}_k \underline{\Gamma}_k^T$

Table II. Control Parameter Notation

SYMBOL	DIMENSION	DEFINITION
k	Scalar	Sample time
$\underline{x}_k$	N x 1	System state vector
$\underline{\Phi}_k$	N x N	System transition matrix
$\underline{w}_k$	N x 1	System noise vector: Gaussian distributed with mean $\bar{\underline{w}}_k$ and covariance $\underline{Q}_k$
$\underline{\Gamma}_k$	N x N	Input noise distribution matrix
$\underline{z}_k$	N x 1	Measurement vector
$\underline{H}_k$	N x N	Measurement distribution matrix
$\underline{v}_k$	N x 1	Measurement noise vector: Gaussian distributed with mean zero and covariance $\underline{R}_k$
$\underline{P}_k$	N x N	Error covariance matrix
$\underline{K}_k$	N x N	Kalman gain matrix
$\hat{\underline{x}}_k$	N x 1	State estimate after measurement $\underline{z}_k$
$\bar{\underline{x}}_{k+1}$	N x 1	Extrapolated state estimate
$\underline{M}_{k+1}$	N x N	Extrapolated error covariance matrix

In this initial example the only required calculations are the state estimate $\hat{\mathbf{x}}_k$ in Eq. (1i) and the extrapolated state estimate $\bar{\mathbf{x}}_{k+1}$ in Eq. (1j). In this case, Kalman filtering requires only matrix-vector multiplications and vector additions and subtractions. Thus, the simple system of Fig. 1 or Fig. 2 suffices. We address how vector additions and subtractions can be accommodated in our system in Sec. V. For now we return to our Kalman filtering discussion and we consider the more practical case when our *a priori* noise statistics $(\mathbf{Q}_k, \mathbf{R}_k)$ are inaccurate or not known (together with their time history) in advance. In this case, after the measurement $\mathbf{z}_k$, we calculate extrapolated noise statistics $\mathbf{Q}_k$ and $\mathbf{R}_k$ from our prior estimates. We then calculate $\mathbf{P}_k$ and $\mathbf{K}_k$ in Eqs. (1g) and (1h) and finally $\hat{\mathbf{x}}_k$ in Eq. (1i). Next we extrapolate the state estimate and the covariance estimate to the next sampling time using Eqs. (1j) and (1k).

In this more practical case, Eqs. (1g)-(1k) must be calculated for each new input measurement. The operations in Eqs. (1g), (1h), and (1k) require the new operations of matrix-matrix-matrix multiplication and matrix inversion in addition to matrix summations. We note that the covariance matrices $\mathbf{M}_k$ and $\mathbf{P}_k$ are symmetric and specific algorithms can be used to simplify the computations. To maintain the generality of our processor, we will not exploit this matrix feature at present. Instead, we use the matrix operations that emerge from this Kalman filtering application to define more general and complex matrix operations required in a typical problem. We now consider how to realize these operations on our frequency-multiplexed systolic array processor with attention to an efficient pipelining and flow of data and operations.

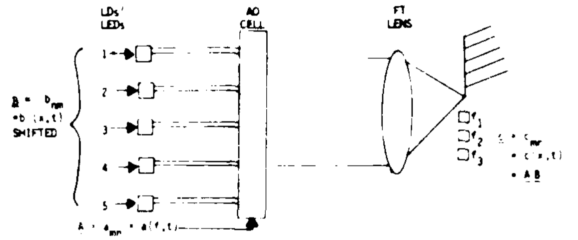


Fig. 4. Basic space-, time-, and frequency-multiplexed optical systolic matrix matrix multiplier.

Table III. Time History (with Space and Frequency Multiplexing) of the Contents of the Components in the System in Fig. 4 for Matrix–Matrix Multiplications

T_1	T_2	T_3	T_4	T_5	COMMENTS
a_{11}	a_{12}	a_{13}	-	-	AO CELL INPUTS AT FREQUENCIES f_1, f_2, f_3
a_{21}	a_{22}	a_{23}	-	-	
a_{31}	a_{32}	a_{33}	-	-	
-	-	-	-	b_{13}	INPUTS TO THE 5 LDs/LEDs
-	-	-	b_{12}	b_{23}	
-	-	b_{11}	b_{22}	b_{33}	
-	-	b_{21}	b_{32}	-	
-	-	b_{31}	-	-	
-	-	c_{11}	c_{12}	c_{13}	OUTPUTS FROM THE B DETECTORS
-	-	c_{21}	c_{22}	c_{23}	
-	-	c_{31}	c_{32}	c_{33}	

IV. Pipelining and Flow of Data and Operations

In this section we consider how the advanced operations of matrix-matrix and matrix-matrix-matrix multiplication, matrix inversion, and matrix addition can be realized on our basic optical systolic processor. To concentrate our attention on the architecture that achieves the best pipelining and flow of data and operations, we suppress the details of the required timing and control circuits and the required data storage and buffering. In all case studies to be presented, we assume square matrices $(M = N)$ with no loss of generality. To enable the data and operational flow to be more clearly shown, we consider only the case of matrices of order $N = M = 3$. Our system architectures can be directly extended to the case of matrices of larger order.

We first consider the basic space-, time-, and frequency-multiplexed optical systolic matrix–matrix multiplication system of Fig. 4 used to realize the matrix-matrix product $\mathbf{AB} = \mathbf{C}$. This architecture uses the basic system of Fig. 3 with the matrix $\mathbf{B}$ fed to the linear input LED array as $b_{nm} = b(x,t)$; i.e., using space and time multiplexing. The matrix $\mathbf{A}$ is fed to the AO cell as $a_{mn} = a(f,t)$, i.e., using frequency and time multiplexing as before. The matrix–matrix product in this system is thus realized as

$$\underline{A}\,\underline{B} = \underline{C} = \begin{array}{cc} & \xrightarrow[\qquad\qquad]{\text{SPACE/TIME}} n \\ \text{freq}\downarrow m & \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \end{array} \begin{array}{cc} & \xrightarrow[\qquad\qquad]{\text{TIME}} m \\ \text{space}\downarrow n & \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \end{array} = \begin{array}{cc} & \xrightarrow[\qquad\qquad]{\text{TIME}} m \\ \text{space}\downarrow m & \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \end{array}, \quad (5)$$

where the space, time, and frequency multiplexing are shown explicitly. To achieve the proper output, we recall that the AO cell converts time to space, that its transmittance shifts in space as time increases, and that the Fourier transform lens converts frequency to space in the output transform plane.

To simplify description of the system in Fig. 4, we show in Table III the time history of the space-multiplexed inputs to the LEDs and the frequency-multiplexed inputs to the AO cell as well as the space and time outputs from the detector array. Note that time proceeds from left to right in this table and that the basic space and time arrangements of the matrix input $\mathbf{B}$ to the LEDs are as before. Referring to Table III we see that, at time T_3, all elements of the first column of

A multiply b_{11}, all elements of the second column of **A** multiply b_{21}, and all elements of the third column of **A** multiply b_{31}. The proper terms in these three vector inner products are formed and summed on the correct detectors by the indicated choice of space, time, and frequency multiplexing. As before, there is a time delay of $2T_B$ before we can commence accessing the elements of **C**. The philosophy used in this space, time, and frequency multiplexing is the same as before. We form the products of the three rows of **A** and the first column of **B** to give the first column of **C**. We then note that the three rows of **A** are needed for subsequent multiplications by the remaining second and third columns of **B**. We thus input these later columns of **B** into the input LED array at successive times with their spatial locations shifted by 1-bit location (to account for the propagation of the **A** data in the AO cell).

Thus, at time T_4 we use the contents of the AO cell (the three rows of **A** with frequency-multiplexed columns) to multiply the second column of **B** yielding outputs that are the second column of **C** at time T_4. At time T_5 we similarly produce the third column of **C** as our output. Thus, in $5T_B$ this system performs nine vector inner products (three at a time in parallel) after a $2T_B$ initial delay. We gate the output detector elements off during this $2T_B$ initial time delay. As shown in Table III, the output from this system is the desired matrix-matrix product **AB** = **C** with elements:

$$c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}. \tag{6}$$

In general, the multiplication of two matrices of dimensions $(M \times N)$ and $(N \times M)$ in this system requires M frequencies, M detectors, $(2N - 1)$ LEDs, and $(2N - 1)T_B$ of time [of which $(N - 1)T_B$ is dead or delay time].

From this brief extension of our original system in Fig. 3 to the realization of more complex operations such as matrix-matrix multiplication as in Fig. 4, we have the intuitive feeling that, unless the data flow and operations are properly pipelined, this architecture will not achieve its full potential. We now consider the case of forming a matrix-matrix-matrix product (e.g., **ABD** = **E**) on this system as required in the Kalman filter formulation in Sec. III. This proves to be an excellent example of how proper pipelining of the data and operations can achieve a reduced overall delay time and efficient data flow. The basic matrix-matrix multiplier in Fig. 4 is again used to realize this. The operation and data flow we use involve first calculating **AB** = **C** and then feeding back **C** to the AO cell with **D** now applied to the LED inputs. This system then forms **CD** = **ABD** = **E**.

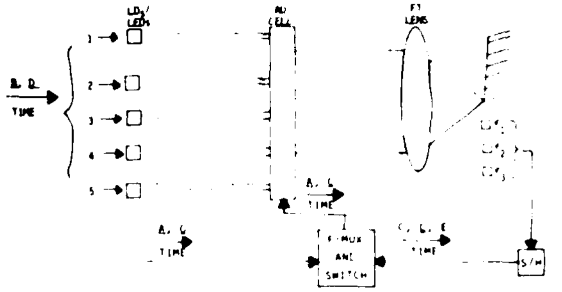


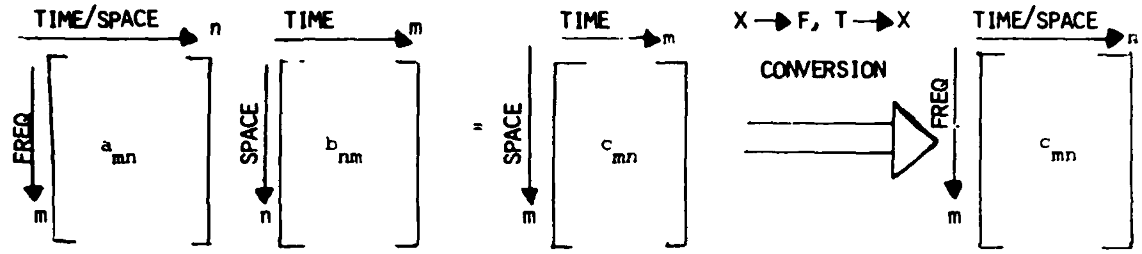
Fig. 5. Basic space-, time-, and frequency-multiplexed optical systolic matrix-matrix-matrix multiplier system.

Table IV. Time History (with Space and Frequency Multiplexing) for the Contents of the Components in the System in Fig. 5 for Matrix-Matrix-Matrix Multiplications

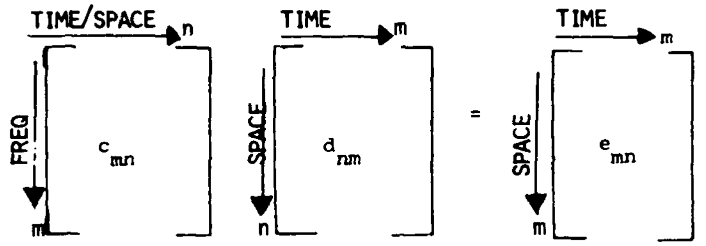
T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	COMMENTS
a_{11}	a_{12}	a_{13}	c_{11}	c_{12}	c_{13}	-	-	AO CELLS INPUTS AT FREQUENCIES f_1, f_2, f_3 T_4 - T_6 FROM DETECTORS
a_{21}	a_{22}	a_{23}	c_{21}	c_{22}	c_{23}	-	-	
a_{31}	a_{32}	a_{33}	c_{31}	c_{32}	c_{33}	-	-	
-	-	-	-	b_{13}	-	-	d_{13}	INPUTS TO THE FIVE LEDs/LDs
-	-	-	b_{12}	b_{23}	-	d_{12}	d_{23}	
-	-	b_{11}	b_{22}	b_{33}	d_{11}	d_{22}	d_{33}	
-	-	b_{21}	b_{32}	-	d_{21}	d_{32}	-	
-	-	b_{31}	-	-	d_{31}	-	-	
-	-	c_{11}	c_{12}	c_{13}	e_{11}	e_{12}	e_{13}	DETECTOR OUTPUTS T_3 - T_5 OUTPUTS ARE FED BACK TO AO
-	-	c_{21}	c_{22}	c_{23}	e_{21}	e_{22}	e_{23}	
-	-	c_{31}	c_{32}	c_{33}	e_{31}	e_{32}	e_{33}	

A simplified diagram of this architecture is shown in Fig. 5. This architecture includes the basic system of Fig. 4 with the addition of feedback through a sample and hold (S/H) unit to a frequency multiplexing and switch box. This unit insures that the inputs to the AO cell are the matrix **A** during the first three time periods and then the matrix **C** during the next three time periods. In Table IV we show the time-history flow (as functions of frequency and space) of the data inputs to the LEDs and the AO cell plus the detector outputs. By $T_3 = 3T_B$, the **A** matrix is present in the AO cell and now the first column of **B** is entered into the lower three LEDs. At this time, T_3, the detector outputs are valid, and at times from T_3 to T_5 the detector outputs are the associated columns of **C** (as in Fig. 4 and Table III). These detector outputs are delayed by one clock period T_B (by the S/H unit) and fed back immediately into the AO cell at times from T_4 to T_6. At T_5 the system has completed the first matrix multiplication **AB** = **C** and the LEDs are now free to accept the columns of **D** as inputs (during T_6 to T_8). This is the precise time at which they are needed. From T_6 to T_8, the system performs the final **CD** = **E** matrix multiplication and the elements e_{mn} of **E** = **ABD** appear at the detector outputs from T_6 to T_8. We note that, in this feedback architecture, no additional delay or dead time is accumulated after the initial $2T_B$ loading of the AO Cell and that new data can be fed to the AO cell at T_7 and the LEDs at T_9 if necessary in specific problems.

The two matrix-matrix cycles performed on the system are summarized:



(7a)



(7b)

where we again show the space, time, and frequency multiplexing used. The first **AB** = **C** matrix–matrix multiplication proceeds as in Eq. (5). The **C** output is then converted to frequency and time/space by the frequency-multiplexing unit in Fig. 5 as shown in Eq. (7a). The second **CD** = **E** matrix multiplication in Eq. (7b) proceeds exactly the same as in Eqs. (5) and (7a). For the $N = M = 3$ problem, the system performs the eighteen vector inner products, three at a time in parallel, in a total time of $8T_B$. In general, for $(M \times N)$ matrices, the system requires M frequencies and detectors, $(2N - 1)$ LEDs (equal to the time-bandwidth product of the AO cell), and $(3N - 1)T_B$ of processing time. The total time delay or dead time of the processor has not been increased over the initial $(N - 1)T_B$ time delay. After this initial delay, all operations of the system are properly pipelined with no additional delays.

Rather than considering further pipelining beyond the matrix-matrix-matrix multiplication operation in Fig. 5 and Table IV, we now consider the realization of a matrix inversion on the basic optical system in Fig. 4. Our present purpose is to determine the basic architecture of a space-, time-, and frequency-multiplexed optical systolic processor. In computing the error covariance matrix update and the Kalman gain matrix update in Eqs. (1g) and (1h), matrix inversion is required. This operation is computationally intensive and is thus a most attractive one to realize on our parallel optical system. To achieve this, we use a modified Richardson algorithm[11] that we previously described.[3] Its realization on the present system is quite different, however. Let us consider solving **C** = **HB** for **B** = $\mathbf{H}^{-1}\mathbf{C}$ without explicitly computing the matrix inverse $\mathbf{H}^{-1}$. This is attractive because in some cases **H** is ill-conditioned and hence cannot be inverted. To achieve this, we use the iterative algorithm[3]

$$\mathbf{B}_{k+1} = [\mathbf{I} - \omega\mathbf{H}]\mathbf{B}_k + \omega\mathbf{C}, \tag{8}$$

where ω is the acceleration parameter that is chosen to speed convergence of the iterative algorithm. When the results $\mathbf{B}_k = \mathbf{B}_{k+1}$ on two successive iterations k and $k + 1$ are equal, Eq. (8) reduces to the solution

$$\mathbf{B} = \mathbf{H}^{-1}\mathbf{C}. \tag{9}$$

To realize this algorithm in our system, we first rewrite Eq. (8) as

$$\mathbf{B}_{k+1}/\omega = [\mathbf{I}/\omega - \mathbf{H}]\mathbf{B}_k + \mathbf{C}. \tag{10}$$

We then recognize that **H** is known and fixed and that we can easily compute $\mathbf{I}/\omega$ by a simple scaling of its elements (this flexibility in a variable acceleration parameter ω on each iteration is attractive and necessary for increased generality of the problems for which we intend this processor to be used). We thus write $[\mathbf{I}/\omega - \mathbf{H}]$ as a matrix **A** which we assume to be known and fixed (or easily recomputed for different ω if necessary). Our iterative algorithm then becomes

$$\mathbf{B}_{k+1} = \omega[\mathbf{A}\mathbf{B}_k + \mathbf{C}] \tag{11}$$

Our final general purpose space-, time-, and frequency-multiplexed optical systolic processor architecture is then as shown in Fig. 6. This system is similar to that of Fig. 5 with the addition of an input unit that computes $[\mathbf{I}/\omega - \mathbf{H}] = \mathbf{A}$ and a resistive adder and amplifier with gain ω at the output of the detectors.

In this architecture, the optical system performs the matrix–matrix multiplication $\mathbf{AB}_k$, the matrix **C** is added one row at a time to the rows of the $\mathbf{AB}_k$ output (in a resistor adder) to form $[\mathbf{AB}_k + \mathbf{C}]$. This matrix summation is then multiplied by ω to realize the right-hand side of Eq. (11) or the next $\mathbf{B}_{k+1}$ iterative input to the AO cell. In Table V we show the time histories (as functions of space and frequency) of the inputs to the

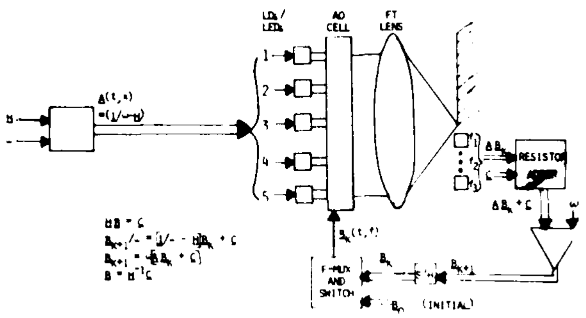


Fig. 6. General optical systolic iterative optical processor architecture (matrix inversion case study detailed).

AO cell and the LEDs, the outputs from the detectors, and the inputs and outputs from the adder. For notational simplicity in Table V, we denote the elements of the initial $k = 0$ matrix $\mathbf{B}_0$ by b_{mn}, the elements of $\mathbf{B}_1$ ($\mathbf{B}_k$ for iteration $k = 1$) by b'_{mn}, and the elements of $\mathbf{B}_2$ (for $k = 2$) by b''_{mn}. As we see from Table V, the initial b_{mn} inputs to the AO cell originate from our initial estimate as described in Refs. 3, 8, and 9 and thereafter all future AO cell inputs for all successive iterations come from the S/H output circuit. Thus, in this final architecture of Fig. 6, flow and pipelining of operations and data are ideal as there is no dead or delay time after the initial $(N - 1)T_B$ AO cell delay.

Several alternate architectures were considered for this system. These included feeding back the output to the LEDs rather than the AO cell (this required variable delays of the detector outputs with the delay time depending on the order N of the problem being solved). In the system shown in Fig. 6, a fixed $1T_B$ delay (achieved with the S/H circuit) is adequate for all problems. Thus, the feedback architecture indicated was chosen as preferable. Realization of the addition of the matrix $\mathbf{C}$ in the optical system by including additional input LEDs and additional time was also considered. Since this resulted in a slower system with poorer pipelining of data and operations and a general degradation of the modular structure of the processor's architecture, we chose to achieve matrix addition by the external resistor adder following the detector as shown in Fig. 6. On inspection of Table V we see that, to allow feedback of $\mathbf{B}_{k+1}$ to the AO cell rather than the LEDs (this is desirable as noted above), we have used $\mathbf{A}$ and $\mathbf{B}_k$ as the inputs to the LEDs and the AO cell, respectively. This differs from the prior systems (Figs. 4 and 5) in which $\mathbf{AB}$ was formed with $\mathbf{A}$ fed to the AO cell. As detailed in Table V, we achieve the proper matrix

Table V. Data and Operational Flow and Pipelining of the System in Fig. 6 for Matrix Inversion

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	COMMENTS
b_{11}	b_{21}	b_{31}	b'_{11}	b'_{21}	b'_{31}	b''_{11}	b''_{21}	AO CELL INPUTS AT FREQUENCIES f_1, f_2, f_3. T_4- T_6 AND T_7- T_9 ARE FROM DETECTORS.
b_{12}	b_{22}	b_{31}	b'_{12}	b'_{22}	b'_{32}	b''_{12}	b''_{22}	
b_{13}	b_{23}	b_{33}	b'_{13}	b'_{23}	b'_{33}	b''_{13}	b''_{23}	
-	-	-	-	a_{31}	-	-	a_{31}	INPUTS TO THE FIVE LEDs
-	-	-	a_{21}	a_{32}	-	a_{21}	a_{32}	
-	-	a_{11}	a_{22}	a_{33}	a_{11}	a_{22}	a_{33}	
-	-	a_{12}	a_{23}	-	a_{12}	a_{23}	-	
-	-	a_{13}	-	-	a_{13}	-	-	
-	-	d_{11}	d_{21}	d_{31}	d'_{11}	d'_{21}	d'_{31}	DETECTOR OUTPUTS $\underline{D} = \underline{A}\,\underline{B}$ $\underline{D}' = \underline{A}\,\underline{B}'$
-	-	d_{12}	d_{22}	d_{32}	d'_{12}	d'_{22}	d'_{32}	
-	-	d_{13}	d_{23}	d_{33}	d'_{13}	d'_{23}	d'_{33}	
-	-	c_{11}	c_{21}	c_{31}	c_{11}	c_{21}	c_{31}	MATRIX $\underline{C}$ INPUTS TO THE ADDER
-	-	c_{12}	c_{22}	c_{32}	c_{12}	c_{22}	c_{32}	
-	-	c_{13}	c_{23}	c_{33}	c_{13}	c_{23}	c_{33}	
-	-	b'_{11}	b'_{21}	b'_{31}	b''_{11}	b''_{21}	b''_{31}	$\underline{B}' = \underline{B}_{K+1}$ AND $\underline{B}'' = \underline{B}_{K+2}$ INPUTS TO THE S/H
-	-	b'_{12}	b_{22}	b'_{32}	b''_{12}	b''_{22}	b''_{32}	
-	-	b'_{13}	b'_{23}	b'_{33}	b''_{13}	b''_{23}	b''_{33}	
-	-	-	b'_{11}	b'_{21}	b'_{31}	b''_{11}	b''_{21}	FREQ - MUXED INPUTS $\equiv$ AO CELL INPUTS FROM T_4 ETC.
-	-	-	b'_{12}	b'_{22}	b'_{32}	b''_{12}	b''_{22}	
-	-	-	b'_{13}	b'_{23}	b'_{33}	b''_{13}	b''_{23}	

multiplication and feedback by space multiplexing of the rows of **A** and time multiplexing of its columns at the LED inputs and by frequency multiplexing the rows of **B** and time multiplexing its columns. This differs from the space, time, and frequency multiplexing used in Figs. 4 and 5. The **AB** = **D** product thus appears on the output detectors one row at a time. Further analysis shows that this architecture of Fig. 6 can realize all the prior matrix operations (matrix-vector multiplication, the solution of linear matrix-vector equations, matrix-matrix multiplication, matrix-matrix matrix multiplication, vector addition and subtraction, matrix addition and subtraction, matrix inversion, and the solution of linear matrix-matrix equations). We thus choose this as our basic architecture for the class of problems we have thus far considered. In later publications we will address the fabrication details of such a system, the partitioning of large matrix problems into smaller ones and the full details of implementation of Kalman filtering and other linear and nonlinear algebraic matrix applications on such a processor. We now address the final topic of this paper: operating on bipolar-valued matrix and vector data and the implementation of matrix subtraction.

V. Bipolar-Valued Data Handling

Many techniques have been suggested for accommodating matrices and vectors with bipolar- and complex-valued elements on an optical processor. We consider only bipolar data representation here (the extension to complex-valued data follows directly as before[3]). In all prior approaches to processing bipolar-valued data, the matrix was placed on a bias (this increased the dynamic range requirements of the detector and the matrix). In our prior bipolar-valued data algorithm[3] we operated the system twice, once with positive-valued input data and once with negative-valued input data. In postdetection electronics, we then form the difference of these two matrix-vector outputs. This achieved the necessary bipolar matrix-vector multiplication. However, this required considerable time (A-D and D-A conversion and data storage, plus a digital addition in the feedback loop). In contrast, our present architecture (Fig. 6) requires no A-D and D-A conversion and performs the necessary matrix addition in an analog system (this is possible to sufficient accuracy to be compatible with the accuracy of the rest of the processor). Our present initial concern is pipelining of data and operations and the speed of the system. In later publications we will address the accuracy of the resultant processor. To maintain a high system throughput, an alternate bipolar-valued data algorithm is thus desirable. We now describe our new solution to this problem.

We consider first the multiplication of two bipolar scalars a and b. We denote their positive and negative parts by (a^+,a^-) and (b^+,b^-), respectively. Each of these components is non-negative. We recall that if $a^+ > 0$, then $a^- = 0$, and if $a^- > 0$, then $a^+ = 0$ (i.e., half of the elements will always be zero). The required bipolar multiplication is

$$\begin{aligned} ab &= (a^+ - a^-)(b^+ - b^-) \\ &= (a^+b^+ + a^-b^-) - (a^-b^+ + a^+b^-) \qquad (12) \end{aligned}$$

From this we see that the positive and negative parts of the product ab are

$$(ab)^+ = (a^+b^+ + a^-b^-), \qquad (13a)$$

$$(ab)^- = (a^-b^+ + a^+b^-), \qquad (13b)$$

and the bipolar output product is

$$ab = (ab)^+ - (ab)^-, \qquad (14)$$

whereas before only one of the two terms in Eq. (14) is nonzero. We now extend this technique to the case of matrix-matrix multiplication **AB** = **C** where the matrix elements are bipolar-valued. We arrange each element a_{mn} of the matrix as a (2×2) submatrix and each element of **B** as a two-element column vector. For $N = M = 2$,

$$\underline{A}\,\underline{B} = \left[\begin{array}{cc:cc} a_{11}^{+} & a_{11}^{-} & a_{12}^{+} & a_{12}^{-} \\ a_{11}^{-} & a_{11}^{+} & a_{12}^{-} & a_{12}^{+} \\ \hdashline a_{21}^{+} & a_{21}^{-} & a_{22}^{+} & a_{22}^{-} \\ a_{21}^{-} & a_{21}^{+} & a_{22}^{-} & a_{22}^{+} \end{array}\right] \left[\begin{array}{c:c} b_{11}^{+} & b_{12}^{+} \\ b_{11}^{-} & b_{12}^{-} \\ \hdashline b_{21}^{+} & b_{22}^{-} \\ b_{21}^{-} & b_{22}^{+} \end{array}\right] = \left[\begin{array}{c:c} c_{11}^{+} & c_{12}^{+} \\ c_{11}^{-} & c_{12}^{-} \\ \hdashline c_{21}^{+} & c_{22}^{+} \\ c_{21}^{-} & c_{22}^{-} \end{array}\right] = \underline{C}. \qquad (15)$$

Each element c_{mn} of the output product matrix **C** is likewise represented as a two-element column vector as shown. As before, only one of the two elements $(c_{mn})^+$, $(c_{mn})^-$ will be nonzero, and all input and output elements will be positive or zero. We have detailed this new algorithm for the case of matrix-matrix multiplication in Eq. (15), since this is the basic operation performed on our new systolic processor shown in Fig. 6.

This algorithm leads itself to direct pipelining and incorporation into all our architectures since the matrix output is fed back to the AO cell, and this output **C** is in the required form of the **B** matrix as shown in Eq. (15). This increases the size of the processor [$(3N - 1)$ LEDs are now required rather than $(2N - 1)$ as before and $2M$ frequencies and $2M$ detectors are now required]. All previous bipolar-valued data algorithms require a similar increase in the number of point sources, frequencies, and detectors, however. Our new algorithm is superior to our prior one because it is directly incorporated into the flow and pipelining of data and operations (without the requirement to operate on output data from successive cycles) and because it reduces the dynamic range requirements of the detector and the matrices.

The arrangement in Eq. (15) can also be used to perform a true matrix subtraction without the need to handle negative numbers. To show this we consider forming

$$\mathbf{A} - \mathbf{BC} = \mathbf{D}. \tag{16}$$

Using our prior $(\ )^+$ and $(\ )^-$ notations, we write Eq. (16) as

$$\begin{aligned}\mathbf{D} = \mathbf{D}^+ - \mathbf{D}^- &= (\mathbf{A}^+ - \mathbf{A}^-) - [(\mathbf{BC})^+ - (\mathbf{BC})^-] \\ &= [\mathbf{A}^+ + (\mathbf{BC})^-] - [\mathbf{A}^- + (\mathbf{BC})^+],\end{aligned} \tag{17}$$

from which we see that the positive and negative parts of the bipolar output matrix $\mathbf{D}$ are obtained as

$$\mathbf{D}^+ = \mathbf{A}^+ + (\mathbf{BC})^-, \tag{18a}$$

$$\mathbf{D}^- = \mathbf{A}^- + (\mathbf{BC})^+. \tag{18b}$$

Realization of Eq. (16) in the form of Eq. (17) follows directly from the matrix partitioning used in Eq. (15). This new bipolar and matrix subtraction algorithm now completes our list of operations required for Kalman filtering. A quite similar list of operations exists for many other matrix problems.

VI. Summary and Conclusion

In this paper we have advanced and described a new iterative matrix–matrix application (Kalman filtering) for implementation on an iterative optical systolic processor. We have analyzed the operations required to realize the Kalman filter and found that they include matrix–matrix and matrix–matrix–matrix multiplication, matrix inversion, and matrix addition and subtraction. We have proposed several new space-, time-, and frequency-multiplexed iterative optical systolic array processor architectures. We addressed the flow and pipelining of data and operations on these architectures (with attention to the operations required for the Kalman filter). The final system in Fig. 6 represents a general-purpose basic optical systolic array module capable of all the required operations. We have also described a new algorithm for bipolar data and matrix subtraction. The resultant systolic architecture does not require CCD shift register readout and summation of the output detector array, and it more fully utilizes the frequency–bandwidth and time–bandwidth products of the AO cell. Our new bipolar data algorithm avoids the need for A–D and D–A conversion and digital subtraction in the feedback loop. It thus provides a fast system with continuously pipelined data and operations.

The applications for iterative optical processors are quite considerable. The applications we have addressed to date include: adaptive phased array radar and beam forming,[12] optimal control,[8,9] deconvolution and inverse filtering,[6] and Kalman filtering. Also documented[13] is the use of an iterative optical processor for general operations such as solution of simultaneous algebraic equations and difference or differential equations, least mean-square problems, computation of eigenvalues and eigenvectors, and nonlinear matrix problems. The use of iterative optical processors implemented with the new frequency-multiplexed systems described in this paper represents a most powerful and general-purpose data processing architecture.

References

1. J. Goodman, A. Dias, and L. Woody, Opt. Lett. **2**, 1 (1978).
2. A. Dias, in "Optical Information Processing for Aerospace Applications," NASA Conf. Publ. 2207 (NTIS, Springfield, Va., 1981).
3. M. Carlotto and D. Casasent, Appl. Opt. **21**, 147 (1982).
4. D. Casasent, Proc. IEEE **65**, 143 (1977)
5. H. Caulfield *et al.*, Opt. Commun. **40**, 86 (1981).
6. D. Casasent, Appl. Opt. **21**, 1859 (1982).
7. A. Warner *et al.*, J. Appl. Phys. **43**, 4489 (1972).
8. D. Casasent *et al.*, Proc. Soc. Photo-Opt. Instrum. Eng. **295**, 176 (1981).
9. C. Neuman *et al.*, Proc. Electro-Opt. Syst. Des. Conf. 311, (1981).
10. A. Bryson and Y. C. Ho, *Applied Optimal Control* (Blaisdell, Waltham, Mass., 1969), Chap. 12.
11. D. Young, *Iterative Solution of Large Linear Systems* (Academic, New York, 1971), pp. 94 and 361–365.
12. D. Casasent and M. Carlotto, Opt. Eng. **21**, 814 (Sept. 1982).
13. D. Casasent and C. Neuman, in "Proceedings, Optical Data Processing for Aerospace Applications," NASA Conf. Publ. 2207 (NTIS, Springfield, Va., 1981).

The support of this research by NASA Lewis (grant NAG 3-5), partial support by the Air Force Office of Scientific Research (grant AFOSR 79-0091), and by contractors of Unicorn Systems, Inc., is gratefully acknowledged.

6. LU AND CHOLESKY DECOMPOSITION ON AN OPTICAL SYSTOLIC ARRAY PROCESSOR

Volume 46, number 5,6 OPTICS COMMUNICATIONS 15 July 1983

LU AND CHOLESKY DECOMPOSITION ON AN OPTICAL SYSTOLIC ARRAY PROCESSOR

David CASASENT and Anjan GHOSH

Carnegie-Mellon University, Department of Electrical Engineering, Pittsburgh, PA 15213, USA

Received 16 February 1983

Direct rather than indirect solutions to matrix-vector equations on an optical systolic array processor are considered. A frequency-multiplexed optical systolic array processor for matrix-decomposition is described. The data flow and ordering of operations for *LU* decomposition or gaussian elimination and $\mathcal{L}\mathcal{L}^{T}$ or Cholesky decomposition on this system are detailed using an algorithm that utilizes the parallel processing ability of the optical systolic array processor. The time required for this optical algorithm is found to be much less than for the digital equivalent. The data flow in the optical system is seen to be most excellent.

1. Introduction

Optical matrix-vector processors [1,2] are very general-purpose systems appropriate for many applications. The new optical systolic array architectures [3–5] using acousto-optic (AO) devices are even more attractive because both the vector and matrix are easily updated in real-time. However, such processors require attention to the pipelining and flow of data and operations [5]. A primary application for such systems is the solution of matrix–vector equations of the form $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ (or similar matrix–matrix and nonlinear matrix equations) [1]. Thusfar, only indirect or iterative algorithms have been suggested for the solution of such problems on optical processors. In this paper, we advance a direct solution using LU matrix-decomposition (or gaussian elimination) [6] and also propose a parallel method for Cholesky decomposition [6].

In section 2, we discuss such solutions and formulate a parallel algorithm for LU matrix-decomposition that is very attractive for an optical realization. We also note that when direct techniques are used, it is preferable to realize the matrix-decomposition on an optical system and to utilize a digital processor for the solution of the simplified resultant matrix–vector problem. In section 3, we describe one method of realizing LU matrix-decomposition on a new [5] frequency-multiplexed optical systolic array matrix–matrix processor. In our solution, we also note that considerable attention must be paid to the pipelining and flow of data and operations in any systolic array processor. In section 4, we discuss a simple method for extending the process of LU decomposition to Cholesky $\mathcal{L}\mathcal{L}^{T}$ decomposition on our optical processor.

2. LU matrix-decomposition

A very popular direct solution to $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ for $\boldsymbol{x}$ is to decompose $\mathbf{A}$ into the product of a lower $\mathbf{L}$ and an upper $\mathbf{U}$ triangular matrix. The equation to be solved then becomes $\mathbf{LU}\boldsymbol{x} = \boldsymbol{b}$. One can solve this equation by first solving $\mathbf{L}\boldsymbol{y} = \boldsymbol{b}$ for $\boldsymbol{y}$ and then $\mathbf{U}\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$. Alternatively, one can compute $\mathbf{L}^{-1}$ and $\mathbf{L}^{-1}\boldsymbol{b} = \boldsymbol{b}'$ and solve $\mathbf{U}\boldsymbol{x} = \boldsymbol{b}'$. Since $\mathbf{L}$ and $\mathbf{U}$ are triangular matrices, the solutions by back substitution are easily achieved in dedicated digital hardware. The computational load associated with the LU decomposition is much larger than the solution of the simplified triangular equation that results [6]. Thus, the use of an optical systolic array processor for matrix-decomposition appears to be a new and most attractive application.

We now consider an LU matrix-decomposition algorithm that is most suitable for implementation on a parallel optical systolic array processor. For an $N \times N$ matrix $\mathbf{A}$, we require $N-1$ steps. In step 1, we form $\mathbf{M}_1\mathbf{A} = \mathbf{A}_1$ (where the first element is the only non-zero element in the first column of $\mathbf{A}_1$). In step 2, we form

$\mathbf{M}_2\mathbf{A}_1 = \mathbf{A}_2$ (where $\mathbf{A}_2$ is such that the first element is the only non-zero element in the first column and the first two elements are the only non-zero elements in the second column). We continue this procedure for $N-1$ steps until we obtain $\mathbf{M}_{N-1}\mathbf{A}_{N-2} = \mathbf{A}_{N-1}$ which is an upper triangular matrix $\mathbf{U}$. Each matrix $\mathbf{M}_n$ is an elementary lower triangular matrix of the general form of an identity matrix with non-zero elements below the diagonal only in the nth column,

$$\underline{M}_n = \begin{bmatrix} 1 & & & & \bigcirc \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{n+1,n} & \ddots & \\ & & -m_{n+1,n} & & \ddots \\ \bigcirc & & \vdots & \bigcirc & \\ & & -m_{N,n} & & 1 \end{bmatrix} \qquad (1)$$

where the non-zero elements of column n of $\boldsymbol{m}_n$ satisfy

$$m_{k,n} = a_{k,n}^{(n-1)}/a_{n,n}^{(n-1)}\,, \qquad (2)$$

for $n+1 \leqslant k \leqslant N$. By the symbol $a_{k,n}^{(n-1)}$ we denote element (k, n) of $\mathbf{A}_{n-1}$ at step $n-1$.

The product $\mathbf{M}_{n-1} \ldots \mathbf{M}_1 = \mathbf{M}$ is also a lower triangular matrix. To solve $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, we thus form the upper-triangular matrix $\mathbf{MA} = \mathbf{U}$ and the vector $\mathbf{M}\boldsymbol{b} = \boldsymbol{b}'$. We note that $\mathbf{M}^{-1} = \mathbf{L}$ is lower-triangular and that $\mathbf{A} = \mathbf{LU}$. Thus, $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ can be written as $\mathbf{M}^{-1}\mathbf{U}\boldsymbol{x} = \boldsymbol{b}$ or as

$$\mathbf{U}\boldsymbol{x} = \mathbf{M}\boldsymbol{b}\,. \qquad (3)$$

In our proposed LU decomposition of $\mathbf{A}$ to solve $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, we will compute only $\mathbf{U}$ and $\mathbf{M}\boldsymbol{b}$ in (3). These calculations will be performed optically and the simplified problem in (3) can then be easily solved digitally by back substitution. Such direct techniques are often more attractive than indirect or iterative matrix inversion algorithms when the same matrix is used many times (e.g. in the implicit solutions of partial differential equations as described in [7]). They are also attractive since the number of steps required $(N-1)$ is fixed and known. Conversely, the number of iterations required in indirect solutions is not easily estimated in advance. We assume that $\mathbf{A}$ is either strictly diagonally dominant or positive definite so that there is no need for pivoting (interchanging rows to insure $a_{n,n}^{(n-1)} > a_{k,n}^{(n-1)}$ for $n+1 \leqslant k \leqslant N$) [6].

3. Optical systolic array implementation

To optically implement the LU decomposition, we consider the frequency-multiplexed optical matrix–matrix systolic array processor [5] of fig. 1. In this architecture, M LEDs are imaged through M regions of an acousto-optic (AO) cell and the Fourier transform of the light distribution leaving the cell is detected on an output linear detector array. If the matrix $\mathbf{B}$ is fed to the LEDs with the matrix elements b_{mn} encoded in space x and time t as $b(x, t)$ and if the elements a_{mn} of the matrix $\mathbf{A}$ are encoded in frequency f and time t as $a(f, t)$, then the detected output $\mathbf{C}$ is a matrix with elements $c_{mn} = c(x, t)$. This matrix is the matrix–matrix product $\mathbf{C} = \mathbf{AB}$. If $b_{nm} = b(t, x)$ and $a_{mn} =$

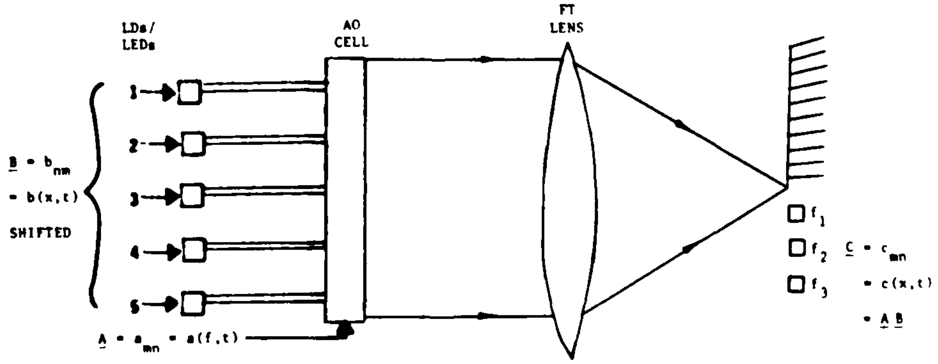


Fig. 1. Schematic diagram of a frequency-multiplexed optical matrix-matrix systolic array processor.

$a(t, f)$, then $c_{mn} = c(t, x)$ and $\mathbf{C} = \mathbf{BA}$ is produced. The operation of this system is detailed in [5]. We denote separate time slots on this system in units of a bit time T_B as $T_1 = T_B$, $T_2 = 2T_B$, etc. For $N \times N$ matrices, we require $(2N-1)$ LEDs. At each T_B, N LEDs are used. They are fed with successive rows or columns of $\mathbf{B}$. The N LEDs used are shifted up by one at successive T_B times. For example, for $b_{nm} = b(x, t)$, the bottom N LEDs are fed with the first row of $\mathbf{B}$ at T_B. LEDs 2 through $N+2$ are fed with the second row of $\mathbf{B}$ at $2T_B$, etc. This is necessary to allow the input data to properly track the matrix information present in the AO cell as it moves through the cell.

To implement our LU decomposition algorithm described in section 2 on the system of fig. 1, three operations are required at each of the $N-1$ steps. At step n, we:

(1) calculate $(1/a_{nn}^{(n-1)})$,

(2) calculate the terms $m_{kn} = [-1/a_{nn}^{(n-1)}]a_{kn}^{(n-1)}$ in (1) and (2) for $n+1 \leq k \leq N$,

(3) calculate $\mathbf{M}_n\mathbf{A}_{n-1} = \mathbf{A}_n$ and $\mathbf{M}_n\boldsymbol{b}_{n-1} = \boldsymbol{b}_n$.

After $N-1$ such steps, we have our desired $\mathbf{M}_{N-1}\mathbf{A}_{N-2} = \mathbf{MA} = \mathbf{U}$ upper triangular matrix and the $\mathbf{M}\boldsymbol{b}$ vector required in (3).

We perform steps (1) and (2) in simple analog electronics (fig. 2) and perform step (3) on the system of fig. 1. At successive T_B times, the circuit of fig. 2 produces successive rows of $\mathbf{M}_n$. We denote row m of $\mathbf{M}_n$ at step n by $\boldsymbol{m}_m^{(n)}$. Since each row has one element that is 1 and only one other non-zero element, a simple MOS switching gate array can select which two LEDs are on at each T_B and feed the 1 and m_{kn} data to these two LEDs. To form $\mathbf{M}_n\mathbf{A}_{n-1}$, we thus feed successive rows of $\mathbf{M}_n$ to the LEDs at successive times T_B. We frequency-multiplex each row of $\mathbf{A}_{n-1}$ (we denote the kth row by $\boldsymbol{a}_k^{(n-1)}$) and feed successive rows to the AO cell at successive times T_B. After NT_B of time, the full $\mathbf{A}_n$ matrix is present in the lower NT_B time slots of the AO cell. The lower N LEDs are now fed with $\boldsymbol{m}_1^{(n)}$ and the first row $\boldsymbol{a}_1^{(n)}$ of $\mathbf{A}_n$ is formed on the detectors. At successive nT_B times, successive rows of $\mathbf{A}_n$ are produced. We compute $\mathbf{M}_n\boldsymbol{b}_{n-1} = \boldsymbol{b}_n$ in step (3) in parallel with $\mathbf{A}_n$ by adding an additional $(N+1)$th frequency to the cell and encoding elements $b_k^{(n-1)}$ of $\boldsymbol{b}_{n-1}$ on this frequency at successive T_B times.

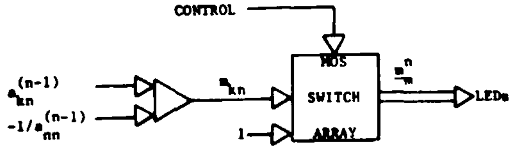


Fig. 2. Analog circuitry to compute the mth row m_m^n of $\mathbf{M}_n$ at step n.

Table 1
Detailed data flow for the realization of LU matrix-decomposition of a 3 × 3 matrix on the optical systolic array processor of fig. 1.

TIME ($T_n = nT_B$)	T_1	T_2	T_3	T_4	T_5	T_6	T_7
LED INPUTS (FROM FIG.2 CIRCUIT)			$\underline{m}_1^1$	$\underline{m}_2^1$	$\underline{m}_3^1$	$\underline{m}_2^1$	$\underline{m}_3^1$
AO CELL INPUTS FREQ f_1-f_3	$\underline{a}_1^0$	$\underline{a}_2^0$	$\underline{a}_3^0$	○	$\underline{a}_2^1$	$\underline{a}_3^1$	
AO CELL INPUTS FREQ f_4	b_1^0	b_2^0	b_3^0	○	b_2^1	b_3^1	
DETECTOR OUTPUTS			$\underline{a}_1^1$ b_1^1	$\underline{a}_2^1$ b_2^1	$\underline{a}_3^1$ b_3^1	$\underline{a}_2^2$ b_2^2	$\underline{a}_3^2$ b_3^2
STORE			Col.1 of $\underline{A}_1$ Elem.1 of $\underline{b}_1$			Col.2 of $\underline{A}_2$ Elem. a_{33}^2; b_2^2 and b_3^2	

The nth column of the final $\mathbf{U}$ matrix has been calculated at step n and at step $n+1$ we do not alter the first n columns and rows of $\mathbf{A}_n$ or the first elements of $\boldsymbol{b}_n$. Thus, at each step, we store the appropriate new column of $\mathbf{A}_n$ and the corresponding new element of $\boldsymbol{b}_n$ and we operate with matrices $\mathbf{M}_n$ and $\mathbf{A}_{n-1}$ reduced in order by one on each successive step. In table 1, we show the pipelining and flow of data and operations in the system of fig. 1 for the case of a 3 × 3 matrix. This table shows the inputs to the LEDs and the AO cell as well as the detector outputs and the data stored at successive times $T_n = nT_B$. As before, we denote row m of $\mathbf{M}_n$ and $\mathbf{A}_n$ by $\boldsymbol{m}_m^{(n)}$ and $\boldsymbol{a}_m^{(n)}$ and the element m of $\boldsymbol{b}_n$ by $b_m^{(n)}$ (note that $\mathbf{A}_0 = \mathbf{A}$ and $\boldsymbol{b}_0 = \boldsymbol{b}$).

For an $N \times N$ matrix, we require $2N-1$ LEDs, $N+1$ frequencies and an AO cell of length $(2N-1)T_B$. Processing the first column of $\mathbf{A}$ requires $(2N-1)T_B$ of time, processing the second column of $\mathbf{A}_1$ requires $(N-1)T_B$, for the third column $(N-2)T_B$, etc. Ignoring the initial $(N-1)T_B$ set-up time, the total time for an optical LU decomposition is

$$[(N)+(N-1)+(N-2)+\ldots+2]\,T_B = [(N^2+N-2)/2]\,T_B\,. \qquad (4)$$

For large N, approximately $N^3/3$ multiplications are required in the conventional serial digital LU decomposition approach. If we assume that a multiplication time and our bit time T_B are comparable, then the digital system requires approximately a factor of NT_B longer time than does the optical system. This occurs because the optical system performs N vector inner products in parallel during each T_B time. Memory access times, data management and bookkeeping can increase the time required digitally (especially if N is large). As shown, data flow in our proposed optical realization of this LU algorithm is quite ideal.

4. Cholesky decomposition and its optical implementation

When a matrix **A** is symmetric and positive-definite, it can be decomposed into the product of a lower-triangular matrix $\mathcal{L}$ and its transpose $\mathcal{L}^T$, i.e.

$$\mathbf{A} = \mathcal{L}\mathcal{L}^T\,. \qquad (5)$$

This is the Cholesky decomposition [6] and $\mathcal{L}$ is the square-root of the matrix **A** [8]. This decomposition is extremely popular and has many applications in science and engineering because, in many physical problems, symmetric (hermitian) and positive definite matrices arise. We now describe a new and simple parallel approach for Cholesky decomposition and discuss its implementation on our systolic optical processor. The three steps in the algorithm are:

(1) Perform the LU decomposition on the systolic optical processor as described above to determine **U**. This requires $[(N^2+N-2)/2]\,T_B$ of time.

(2) From **U**, compute the diagonal matrix

$$\mathbf{P} = \text{diagonal}\left(\frac{1}{\sqrt{u_{11}}}, \frac{1}{\sqrt{u_{22}}}, \ldots, \frac{1}{\sqrt{u_{NN}}}\right).$$

(3) On the systolic optical processor we then form the matrix-matrix product **PU**. This requires $(2N-1)T_B$ of time [5]. This is now our desired upper-triangular matrix

$$\mathcal{L}^T = \mathbf{PU}\,, \qquad (6)$$

and the Cholesky decomposition is uniquely determined.

Our optical implementation of Cholesky decomposition requires only $[(N^2+5N-4)/2]\,T_B$ of time. For large matrices of order $N \times N$, the conventional Cholesky decomposition on digital computers take approximately $N^3/6$ multiplications. Thus, the digital computation requires a time approximately a factor of $NT_B/3$ longer than does the optical computation.

Acknowledgment

The support of this research by the Air Force Office of Scientific Research (Grant AFOSR 79-0091-Amendment D) and NASA Lewis Research Center (NAG 3-5) is gratefully acknowledged as are many fruitful discussions on matrix algorithms with Professor C.P. Neuman.

References

[1] M.A. Monahan, K. Bromley and R.P. Bocker, Proc. IEEE, 65 (1977) 121.
[2] D. Casasent and C. Neuman, Proc. NASA Langley Conf. on Optical information processing, NASA Conference Publication 2207 (NTIS) (August 1981).
[3] H.J. Caulfield, M.J. Foster and S. Horvitz, Optics Comm. 40 (1981) 86.
[4] D. Casasent, Appl. Optics 21 (1982) 1859.
[5] D. Casasent, J. Jackson and C.P. Neuman, Appl. Optics 22 (1983) 115.
[6] G.W. Stewart, Introduction to matrix computations (Academic Press, New York, 1973).
[7] D. Casasent and A. Ghosh, Proc. SPIE 388 (1983).
[8] T. Kailath, Linear systems (Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980).

7. SINGULAR VALUE DECOMPOSITION USING ITERATIVE OPTICAL PROCESSORS

Singular value decomposition using iterative optical processors

B. V. K. Vijaya Kumar

Carnegie-Mellon University, Department of Electrical Engineering, Pittsburgh, Pennsylvania 15213.

Received 23 December 1982.
0003-6935/83/070962-02$01.00/0.

Recently, there has been increasing interest[1,2] in extending the repertoire of operations achievable through optical data processing. A major class of processors proposed[3] toward this end is that of iterative optical processors (IOP). The generic IOP has two data paths: a forward path, where an input vector **x** is multiplied by a matrix **M** to produce the output vector **y** and a feedback path which uses **y** to produce the input to the forward path for the next iteration. Implementations of IOP include fiber based systems[4] and more recently[5] systems based on the use of acoustooptic light modulators along with systolic processing concepts. Because the differences among these implementations are not conceptual, we will treat these as equivalent in this Letter. Applications of IOP include the optimal weight vector determination for adaptive radar arrays,[4] matrix inversion for control problems,[6] and eigenvalue/eigenvector estimation.[7,8] In this Letter we suggest a way of using the IOP for estimating the singular values and the corresponding singular vectors of a matrix.

The singular value decomposition (SVD) of a matrix **M** of size $m \times n$ (with $m \geq n$) is given by

$$\mathbf{M} = \mathbf{V}\mathbf{D}\mathbf{U}^T, \tag{1}$$

where **D** is an $m \times n$ matrix containing zeros everywhere except along the first r diagonal elements where r is the rank of the matrix **M**. Let us arrange the nonzero entries along the diagonal of this matrix in descending order as $\sigma_1, \sigma_2, \ldots, \sigma_r$. The matrices **U** and **V** are unitary and are of size $n \times n$ and $m \times m$, respectively, and the superscript T denotes the transpose operation. SVD was originally conceived as an accurate method of estimating the rank of a singular matrix. Andrews and Patterson[9] have shown its use for image compression applications. SVD can be considered as the deterministic counterpart of the statistical Karhunen-Loeve (K-L) transform in the sense explained below. The K-L transform is based on the computation of a covariance matrix from an ensemble of images and then estimating the dominant eigenvectors of this matrix. These dominant eigenvectors can then be shown to represent the set of images optimally, but a single image may not be represented well. Thus the K-L transform has the disadvantages that it needs an ensemble of images and that it may not represent a particular image very well. The SVD works only on a particular image and is geared toward representing that image in the sense of minimum square error. Even though the usefulness of SVD for image processing was demonstrated long ago,[9,10] it has not gained popular acceptance because of the heavy computational burden it imposes. If optical processing can be used to perform SVD, this computation can be carried at a rapid rate.

Because of the heavy computational burden of performing complete SVD, it is desirable to know *a priori* if SVD is useful for data compression with a particular image. This can be answered by estimating the singular values of a matrix and then comparing the magnitudes of the few dominant singular values. Sahasrabudhe and Vaidya[11] suggest a relation between these dominant singular values and the observed correlation coefficient for a Markov type image. Their method has the disadvantages that the relation at best is qualitative only and that this relation is true only for Markov type images. The method suggested in this Letter is applicable to general matrices and provides explicit estimates of the dominant singular values.

Because of the unitary nature of matrices **U** and **V**, we can get the following two results from Eq. (1):

$$\begin{aligned}(\mathbf{M}\mathbf{M}^T)\mathbf{V} &= (\mathbf{V}\mathbf{D}\mathbf{U}^T)(\mathbf{U}\mathbf{D}^T\mathbf{V}^T)\mathbf{V} \\ &= \mathbf{V}\mathbf{D}(\mathbf{U}^T\mathbf{U})\mathbf{D}^T(\mathbf{V}^T\mathbf{V}) \\ &= \mathbf{V}(\mathbf{D}\mathbf{D}^T),\end{aligned} \tag{2}$$

$$\begin{aligned}(\mathbf{M}^T\mathbf{M})\mathbf{U} &= (\mathbf{U}\mathbf{D}^T\mathbf{V}^T)(\mathbf{V}\mathbf{D}\mathbf{U}^T)\mathbf{U} \\ &= \mathbf{U}\mathbf{D}^T(\mathbf{V}^T\mathbf{V})\mathbf{D}(\mathbf{U}^T\mathbf{U}) \\ &= \mathbf{U}(\mathbf{D}^T\mathbf{D}).\end{aligned} \tag{3}$$

From the above results we can easily conclude the following. The n columns $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ of matrix **U** are the eigenvectors of matrix $\mathbf{M}^T\mathbf{M}$ with the corresponding eigenvalues given by the diagonal elements of $\mathbf{D}^T\mathbf{D}$ of size $n \times n$. Similarly the m columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ of **V** are the eigenvectors of $(\mathbf{M}\mathbf{M}^T)$ with the corresponding eigenvalues being the diagonal elements of $\mathbf{D}\mathbf{D}^T$. Since **D** is a matrix with diagonal nonzero elements $\sigma_1, \sigma_2, \ldots, \sigma_r$, the diagonal matrices $\mathbf{D}\mathbf{D}^T$ and $\mathbf{D}^T\mathbf{D}$ contain the nonzero elements $\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2$. Thus the singular values σ_i of the matrix **M** can be optically obtained by using the power method to determine the eigenvalues of the matrices $\mathbf{M}\mathbf{M}^T$ or $\mathbf{M}^T\mathbf{M}$ and taking the square root of these values. We can also obtain the dominant left singular vectors by determining the dominant eigenvectors of $\mathbf{M}\mathbf{M}^T$ and the dominant right singular vectors **U** by determining the dominant eigenvectors of $\mathbf{M}^T\mathbf{M}$. More details regarding the application of power method for determining the eigenvalues and eigenvectors can be found elsewhere.[7,8] This approach has the disadvantage that the mask representing $\mathbf{M}\mathbf{M}^T$ requires a dynamic range that is twice (in decibels) of a mask representing **M**. As pointed out by Stewart,[12] this situation can easily lead to very erroneous results in the presence of finite accuracy.

A possible approach to avoiding the dynamic range problems of the mask is to use two masks, one to represent **M** and another to denote $\mathbf{M}^T$. These two masks can be used in two different optical vector/matrix multipliers connected in a loop. The functions of these two matrix/vector multipliers can be better understood by looking at the following results easily obtained from Eq. (1):

$$\mathbf{M}\mathbf{U} = \mathbf{V}\mathbf{D}, \tag{4}$$

$$\mathbf{M}^T\mathbf{V} = \mathbf{U}\mathbf{D}^T. \tag{5}$$

The above exact equations suggest the following version of power method to estimate the singular values and the corresponding singular vectors.

Step 1: Choose an initial vector $\mathbf{u}_0$.

Step 2: Determine the matrix-vector product $\mathbf{M}\mathbf{u}_0$. Norm of this vector is the initial estimate for the first singular value. Normalized (Norm = 1) version of this vector is our new estimate for **v**, namely, this is $\mathbf{v}_k$.

Step 3: Find $\mathbf{v}_k$ from $\mathbf{u}_k$ according to $\mathbf{v}_k = \text{Norm}[\mathbf{M}\mathbf{u}_k]$, and $\mathbf{u}_{k+1}$ is found from $\mathbf{v}_k$ as per $\mathbf{u}_{k+1} = \text{Norm}[\mathbf{M}^T\mathbf{v}_0]$, where Norm[·] denotes the normalization such that the resulting vector is of unit length. The norm of the vectors before normalization is an estimate of the singular values of the matrix.

Step 4: Repeat step 3 until convergence is obtained in the estimated singular values. The convergence behavior of such an approach is discussed elsewhere.[13]

As can be easily seen from the above algorithm, the optical

Reprinted from **Applied Optics**, Vol. 22, page 962, April 1, 1983

realization of this would involve a top half of the loop which optically multiplies $\mathbf{u}_k$ by $\mathbf{M}$ to produce $\mathbf{v}_k$ and a bottom half which will multiply the $\mathbf{v}_k$ generated in the top half, by matrix $\mathbf{M}^T$ to produce the vector $\mathbf{u}_{k+1}$ as the input for the next cycle at the top half. Such an arrangement increases the complexity of the hardware by a factor of 2, but it enables us to obtain more accurate estimates of the singular value and the singular vectors.

The author would like to acknowledge the support of the Air Force Office of Scientific Research under grant AFOSR79-0091.

References

1. H. J. Caulfield, R. Haimes, and D. Casasent, Opt. Eng. **19**, 152 (1980).
2. D. Psaltis and D. Casasent, Opt. Eng. **19**, 193 (1980).
3. D. Psaltis, D. Casasent, and M. Carlotto, Opt. Lett. **4**, 348 (1979).
4. M. Carlotto and D. Casasent, Appl. Opt. **21**, 147 (1982).
5. D. Casasent, Appl. Opt. **21**, 1859 (1982).
6. D. Casasent and C. Neuman, "Operations Achievable on an Iterative Optical Processor," in Applications of Optical Processing to Aerospace Needs, NASA, 1981.
7. H. J. Caulfield D. Dvore, J. W. Goodman, and W. Rhodes, Appl. Opt. **20**, 2263 (1981).
8. B. V. K. Vijaya Kumar and D. Casasent, Appl. Opt. **20**, 3707 (1981).
9. H. C. Andrews and C. L. Patterson, IEEE Trans. Comput. **C-25**, 72 (1976).
10. S. C. Sahasrabudhe and A. D. Kulkarni, Comput. Graph. Imag. Proc. **9**, 203 (1979).
11. S. C. Sahasrabudhe and P. M. Vaidya, IEEE Trans. Acoust. Speech Signal Process. **ASSP-27**, 434 (1979).
12. G. Stewart, *Introduction to Matrix Computations* (Academic, New York, 1973).
13. S. Shlien, IEEE Trans. Pattern Anal. Mach. Intelli. **PAMI-4**, 671 (1982).

8. GUIDELINES FOR EFFICIENT USE OF OPTICAL SYSTOLIC ARRAY PROCESSORS

IOCC '83
IEEE Cat No
83CH1880-4
SPIE Vol 422

GUIDELINES FOR EFFICIENT USE OF OPTICAL SYSTOLIC ARRAY PROCESSORS

David Casasent

Carnegie-Mellon University
Department of Electrical Engineering
Pittsburgh, Pennsylvania 15213

ABSTRACT

The design, error analysis, component accuracy required, computational capacity, data flow and pipelining, plus the algorithm and application all seriously impact the use of optical systolic array processors. This paper provides initial remarks, results, examples and solutions for each of these issues.

1. INTRODUCTION

Optical matrix-vector processors [1-4] represent general-purpose optical processors. Optical computing has long desired and needed such systems. The iterative optical processor [5-6] includes feedback of the output back to the input of such systems through an analog (or digital) feedback circuit. This architecture thus relieves the problem of what to do with the output data generated by such systems and moreover it provides many additional operations of increased flexibility and complexity. These architectures all require the use of real-time and reusable 2-D spatial light modulators (SLMs) for the matrix data. Such devices have yet to become widely available. New engagement mode optical matrix-vector processors [7] may become very attractive if more advanced versions of 2-D SLMs can be produced.

A more attractive and realizable optical linear algebra processor can be achieved using bulk acousto-optic (AO) devices. Several versions of such systems [8-10] have been described. These architectures are referred to as optical systolic array processors. The most attractive, general and powerful system of this type is a frequency-multiplexed AO matrix-matrix systolic array processor [10]. This architecture is briefly reviewed in Section 2. Remarks on its speed versus that of the system using a 2-D SLM and associated remarks on input data requirements are noted in Section 3. This is felt to be the most attractive optical systolic array processor since it fully utilizes both the bandwidth, aperture time and space bandwidth product of an AO cell, since the basic operation it performs is a matrix-matrix rather than a matrix-vector operation, and since it lends itself quite nicely to pipelining of data and operations.

The accuracy achievable in any optical or analog processor will be limited. Thus, the effect of component errors on such a processor is a major issue. Such analyses of conventional optical linear algebra processors are very limited [11,12]. Our own results have addressed the use of such processors in the solution of a total system problem. In this area, we have found that for adaptive phased array radar and optimal control that the accuracy of the adaptive weights in the closed-loop poles calculated on the conventional optical systolic processors are sufficiently adequate. However, other applications require precision beyond 1%. In such cases, optical convolution to realize digital multiplication [13] can be incorporated into optical systolic processors as recently described in Ref.14, residue arithmetic can also be used in systolic processors as we recently described in Ref.15 or similar techniques can be employed to reduce the dynamic range requirements and improve the accuracy of optical matrix processors as required. Space does not permit us to detail these issues here.

Next, we consider the computational rate for optical systolic processors. The computing capacity of an optical linear algebra processor must significantly exceed that of VHSIC, digital systolic processors, GaAs and other technologies. This issue is not often detailed. In Section 3, we discuss the performance possible on the frequency-multiplexed optical systolic array processor and we note that the use of matrix-matrix rather than matrix-vector optical systolic processors represent an obvious method by which to achieve increased computing capacity. However, optical matrix-matrix processors which output the entire matrix in parallel are not necessarily realistic because of the data handling and data flow problems associated with feeding an entire matrix to a system in parallel or extracting an entire matrix from it in parallel. Thus, data flow, pipelining of operations, data management and storage requirements, avoiding A/D conversion, etc. are often more vital aspects of optical systolic processors. These aspects of computing capacity are discussed in Section 5.

In Section 4, we discuss algorithms appropriate for realization on such processors. The list is not complete, but attention should be given to the fact that these optical systems should work in concert with digital processors and not emulate digital linear algebra and digital systolic array

CH1880-4/83/0000/0209$01.00 © 1983 IEEE

algorithms. In Section 5, we use the solution of a matrix equation (or matrix inversion) as an example of a most vital issue for any systolic array processor: data flow and pipelining of operations. In Section 6, we advance initial remarks on the applications for which optical systolic array processors should be directed. These include large dimensionality problems and problems involving computationally complex matrix calculations. We also note that specific case studies are essential to detail the design, pipelining and performance of a given optical systolic array processor.

2. FREQUENCY-MULTIPLEXED AO SYSTOLIC ARRAY PROCESSOR

The frequency-multiplexed AO systolic array processor is shown schematically in Figure 1. It uses 2N-1 LEDs, each imaged through a different region of an AO cell, with the Fourier transform of the data leaving the AO cell recorded on a linear array of N detectors. To use the system for a matrix-matrix multiplication, we consider the calculation of $\underline{A}\,\underline{B} = \underline{C}$, where all matrices are N x N. We frequency encode the elements of each row of $\underline{B}$ and feed one row of $\underline{B}$ to the AO cell at each bit time T_B. After NT_B, the full $\underline{B}$ matrix is present in the lower NT_B time aperature of the AO cell. We then pulse on the bottom N LEDs with the first row $\underline{a}_1$ of $\underline{A}$. The detector outputs are then the first row of $\underline{C}$. This is achieved since the product of $\underline{a}_1$ and all elements of the first column $\underline{b}_1$ of $\underline{B}$ leave the AO cell at the same frequency f_1 and hence add on the same detector in the output Fourier transform plane. Similar remarks hold for the other elements of the first row of $\underline{C}$.

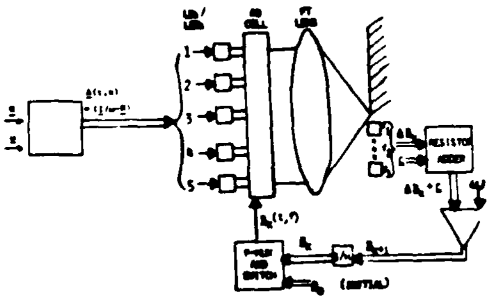


FIGURE 1: General optical systolic iterative optical processor architecture (matrix inversion case study detailed).

At each T_B, the system calculates N vector inner products (on N element vectors) in parallel. Frequency-multiplexing of the columns of $\underline{A}$ into the AO cell is also possible (and necessary) depending upon the operations required [10]. The general philosophy of this processor is to time and space multiplex the matrix data fed to the LEDs and to time (or space) and frequency-multiplex the matrix data fed to the AO cell. Such an architecture can fully use the bandwidth and time bandwidth product of the AO cell. Because one row (or column) of the output matrix is produced in parallel at one time, these outputs can be directly fed back to the AO cell, thus data flow and pipelining is nearly ideal in such a processor. We detail the pipelining of data and operations on this system in Section 5. The basic operation in this processor is a matrix-matrix multiplication rather than a matrix-vector multiplication. Thus, this architecture performs N times more computations per bit time T_B than do other AO systolic processors.

3. COMPUTING CAPACITY

The use of an AO cell rather than a 2-D SLM does not cause any loss of computing power for general matrix problems. In fact, such systolic processors provide increased computing capacity. However, frequency-multiplexing must be employed to achieve this. Consider a TeO_2 AO cell with a typical aperture time of 40μsec and a time bandwidth product of 10^3. With the use of frequency-multiplexing, we can enter an entire 30 x 30 matrix into the cell in 20μsec. Then, every $T_B = 20/30 =$ 66nsec the system performs 30 vector inner products on 30 point vectors in parallel and displays the 30 outputs on the linear detector array. This corresponds to 1000 operations per 66μs or 6 x 10^{10} multiplications and additions per second, for a typical multiplication time of 0.1 nsec. Thus such an architecture is very competitive with digital processors, even if its capacity is reduced by a factor of 100.

The optical matrix-vector processors using 2-D SLMs can perform a comparable number of computations. Thus, *no loss of computational rate results* when AO systems are used. When the matrix data in the 2-D SLM version of the system must be changed, 30msec is typically required. This can cause a significant loss in effective computations and data flow, since a considerable dead time of 30msec results every time the matrix must be changed, plus the output data (and the input data for the matrix) must be buffered during this time. Thus, AO systolic processors have more computing capacity and flexibility, than do matrix processors using 2-D SLMs. In addition, the dynamic range of an AO cell and its uniformity is typically superior to such parameters for 2-D SLMs.

To avoid A/D conversion of the output detector data and to provide proper data flow, the conventional iterative algorithms must often be modified. To demonstrate this, we recall the most popular iterative feedback algorithm (the Richardson algorithm) rewritten to solve the matrix-matrix equation $\underline{H}\,\underline{B} = \underline{C}$ for $\underline{B} = \underline{H}^{-1}\underline{C}$, i.e.

$$\underline{B}(k+1) = \underline{B}(k) + \omega[\underline{C} - \underline{H}\,\underline{B}(k)], \qquad (1)$$

where k denotes the iteration index and ω is the acceleration parameter used to speed convergence (ω is the reciprocal of the maximum eigenvalue of $\underline{H}$ which we approximate by the Euclidean norm of $\underline{H}$). We rewrite (1) as

$$[\underline{B}(k+1)]/\omega = [\underline{I}/\omega - \underline{H}]\underline{B}(k) + \underline{C}.$$

We realize that $[\underline{I}/\omega - \underline{H}]$ is known and easily calculated, since $\underline{I}/\omega$ affects only the diagonal elements of $\underline{H}$ and the effect is the same for all diagonal elements. Denoting $[\underline{I}/\omega - \underline{H}]$ by a new matrix $\underline{A}$, we rewrite our iterative algorithm as

$$\underline{B}(k+1) = \omega[\underline{A}\,\underline{B}(k) + \underline{C}]. \qquad (2)$$

The system of Figure 1 represents a rather general-purpose iterative optical matrix-matrix systolic processor. We now discuss the feedback portion of this system with attention to the iterative matrix solution or matrix inversion algorithm of (2). The LED inputs are the rows of $\underline{A}$. They are easily calculated from $\underline{H}$. The preprocessing box feeding the LEDs requires only one operational amplifier, one resistor adder and an N element MOS switch. At the detector output, one row of $\underline{A}\,\underline{B}$ appears in parallel at a time. Thus, we can add the matrix $\underline{C}$ to this matrix-matrix product [as required in the right hand side of (2)] one row at a time. This is achieved by a simple N element resistor adder. Multiplication by the acceleration parameter is likewise trivial in electronics. Thus, as shown, the entire right hand side of (2) can be calculated on-line without the need for A/D conversion, digital storage or any appreciable feedback delay. Such combinations of analog electronic adders and multipliers properly combined with optical matrix-matrix multiplipliers represent very excellent and powerful systolic processors with high computational rates and excellent data flow. Before detailing the pipelining and data flow, we consider parallel algorithms and appropriate operations for realization on such systems.

4. APPROPRIATE PARALLEL ALGORITHMS

Much of the recent research on optical systolic array processors has considered the use of these processors to realize conventional linear algebra algorithms and various operations. Many of these operations can already be easily performed in digital VHSIC and digital systolic processors. A typical example of this issue arises in the solution of a linear system of equations by direct (matrix-decomposition) rather than indirect (iterative) methods. The key issue we noted [16] is that performing the matrix-decomposition is the most computationally intensive operation. In Ref. 16, we noted that the solution of the resultant lower triangular system of equations is trivial in dedicated digital systolic processors by back or forward substitution algorithms.

In Refs.16-18, we detailed how QR and LU and Cholesky decomposition is possible on optical systolic array processors. In cases when an all optical solution is preferable (e.g. for data flow reasons), an optical solution of the resultant lower (or upper) triangular system of equations is possible as we detailed in [19]. The production of Hessenberg and tri-diagonal matrices is also possible as we have noted [17]. Such solutions lead directly to eigenvalue calculation applications of these processors and to the possible use of other optical systolic array processors [8-9] for solving tri-diagonal matrix equations. When Toeplitz matrices result, an optical deconvolver [9] is appropriate. The implicit and explicit solutions of partial differential equations [9,16] also merits similar remarks. Attention to parallel algorithms (for example the Householder algorithm) and algorithms that pipeline quite nicely (e.g. new descriptions of the solution of lower or upper triangular systems of equations[19]) are necessary and essential for the practical and efficient use of this new class of processor. Least square solutions and other iterative algorithms that are more appropriate than our general Richardson algorithm (when the matrix has special structure) exists as we have detailed [20].

To detail the preferable procedure for the direct solution of a matrix-vector equation, we considered the QR matrix-decomposition solution of $\underline{B}\,\underline{x} = \underline{a}$ for $\underline{x}$. The matrix $\underline{B}$ is generally decomposed into an orthogonal $\underline{Q}$ and upper triangular $\underline{R}$ matrix. We then obtain $\underline{Q}\,\underline{R}\,\underline{x} = \underline{a}$. Since $\underline{Q}^T = \underline{Q}^{-1}$, we can solve $\underline{R}\,\underline{x} = \underline{Q}^T\underline{a} = \underline{a}'$, where $\underline{Q}^{-1} = \underline{Q}^T$ is easily calculated from $\underline{Q}$. The solution of $\underline{R}\,\underline{x} = \underline{a}'$ is easy in digital hardware by back substitution, since this corresponds to a lower triangular system of equations. However, to best utilize such an algorithm on our optical system, we should never explicitly calculate $\underline{Q}$ and $\underline{Q}T$. Rather, we calculate $\underline{R}$ and $\underline{a}'$ directly by successive matrix-matrix and matrix-vector multiplications of the matrix $\underline{B}$ and the vector $\underline{a}$ by Householder decomposition matrices on our system of Figure 1. This yields $\underline{a}'$ directly, thus simplifying the entire problem. This also represents a preferable use of optical systolic processors in concert with digital systems (the conventional approach would be to optically emulate calculation of $\underline{R}$ and $\underline{Q}$ from $\underline{B}$). Other new parallel and revised linear algebra algorithms are possible for many other applications. They are typical of the more appropriate uses for such optical processors. We note that such a philosophy should be pursued in all future optical systolic array processor algorithms. Matrix-decomposition algorithms are a specific example of algorithms which require a new matrix mask at each iteration. If realized on a system with a 2-D SLM mask, 30msec would be required for each change of the matrix or on each iteration. Thus, this type of algorithm is typical of those that are appropriate for realization on a systolic processor, rather than a 2-D SLM matrix-vector system. These are also typical of the types of algorithms this new type of optical processor should address. We note in passing that no application has yet appeared requiring the real-time eigenvalue/eigenvector matrix calculations possible on such systems.

5. PIPELINING OF DATA AND OPERATIONS

Any systolic processor must pay close attention to the flow of data and operations. In Ref. 10, this issue was first noted. In Refs.10,16-19, we provided many examples of the data flow and pipelining associated with the realization of various linear algebra operations on such processors. This issue is vital for many reasons. First, once data enters the AO cell, it must be completely processed, i.e. all calculations requiring it must be performed before the data reaches the end of the cell (typically in 40-50μsec). Furthermore, as soon as the parallel output data appears at the detector, something must be done with it. In our algorithms and in our architectural studies of diverse linear algebra algorithms, we have given particular attention to such issues. As a most appropriate study of this issue, we show in Table 1 the flow of data into the LEDs and the AO cell, and the flow of data from the detectors back into the AO cell for realization of the solution in (2) to the matrix-matrix equation $\underline{H}\,\underline{B} = \underline{C}$ for $\underline{B} = \underline{H}^{-1}\underline{C}$ on the system of Figure 1 using the general feedback architecture shown in this figure. Each column of Table 1 shows the data contents of the corresponding system elements at times $T_1 = T_B$, $T_2 = 2T_B$, $T_3 = 3T_B$ etc. As shown, data flow and pipelining of operations is essentially ideal in this system. For our detailed 3 x 3 matrix example, we see that as soon as one row of the $\underline{B}(k+1)$ output is calculated, it is available for feedback directly into the AO cell. We also note that at the specific time that this detector output occurs, the bottom T_B of the time aperture of the AO cell is free and available for new input data.

TABLE 1: Data and operational flow and pipelining in the system of Figure 1 for matrix inversion.

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	COMMENTS
a_{11}	a_{21}	a_{31}	b'_{11}	b'_{21}	b'_{31}	b''_{11}	b''_{21}	AO CELL INPUTS AT FREQUENCIES f_1, f_2, f_3. T_4- T_6 AND T_7- T_9 ARE FROM DETECTORS.
a_{12}	a_{22}	a_{32}	b'_{12}	b'_{22}	b'_{32}	b''_{12}	b''_{22}	
a_{13}	a_{23}	a_{33}	b'_{13}	b'_{23}	b'_{33}	b''_{13}	b''_{23}	
-	-	-	-	a_{31}	-	-	a_{31}	INPUTS TO THE FIVE LEDs
-	-	-	a_{21}	a_{32}	-	a_{21}	a_{32}	
-	-	a_{11}	a_{22}	a_{33}	a_{11}	a_{22}	a_{33}	
-	-	a_{12}	a_{23}	-	a_{12}	a_{23}	-	
-	-	a_{13}	-	-	a_{13}	-	-	
-	-	d_{11}	d_{21}	d_{31}	d'_{11}	d'_{21}	d'_{31}	DETECTOR OUTPUTS $\underline{D} = \underline{A}\,\underline{B}$ $\underline{D}' = \underline{A}\,\underline{B}'$
-	-	d_{12}	d_{22}	d_{32}	d'_{12}	d'_{22}	d'_{32}	
-	-	d_{13}	d_{23}	d_{33}	d'_{13}	d'_{23}	d'_{33}	
-	-	c_{11}	c_{21}	c_{31}	c_{11}	c_{21}	c_{31}	MATRIX $\underline{C}$ INPUTS TO THE ADDER
-	-	c_{12}	c_{22}	c_{32}	c_{12}	c_{22}	c_{32}	
-	-	c_{13}	c_{23}	c_{33}	c_{13}	c_{23}	c_{33}	
-	-	b'_{11}	b'_{21}	b'_{31}	b''_{11}	b''_{21}	b''_{31}	$\underline{B}' = \underline{B}_{k+1}$ AND $\underline{B}'' = \underline{B}_{k+2}$ INPUTS TO THE S/H
-	-	b'_{12}	b'_{22}	b'_{32}	b''_{12}	b''_{22}	b''_{32}	
-	-	b'_{13}	b'_{23}	b'_{33}	b''_{13}	b''_{23}	b''_{33}	
-	-	-	b'_{11}	b'_{21}	b'_{31}	b''_{11}	b''_{21}	FEED - BACK INPUTS $\equiv$ AO CELL INPUTS FROM T_4 ETC.
-	-	-	b'_{12}	b'_{22}	b'_{32}	b''_{12}	b''_{22}	
-	-	-	b'_{13}	b'_{23}	b'_{33}	b''_{13}	b''_{23}	

6. APPLICATIONS

As our final topic, we note the fact that if optical linear algebra processors are to become competitive with digital technology, these processors must consider five key items:

1. The architectures for such systems should provide full processing capability. The use of matrix-matrix multipliers rather than matrix-vector operations is a typical example.
2. The data flow and pipelining of operations in such architectures must be addressed. Such designs will clearly show the superior data management, bookkeeping, and avoidance of A/D conversion that is possible in optical systolic processors. Table 1 is an example of the efficient pipelining possible on such systems with the proper choice of the algorithm used.
3. These processors should be applied to problems where the accuracy of the final actual result (not necessarily the matrix-vector or matrix-matrix product) is the performance measure used. Our adaptive phased array radar and closed-loop optimal control examples are viable specific cases where such a philosophy is most appropriate.
4. The problems addressed on such processors should be of sufficiently large size (i.e. the order N of the matrix should be large).
5. the calculations required in the solution of the problem should be proportional to N^3 (rather than N^2 or N). We have found matrix-decomposition and Kalman filtering to be examples that fully utilize the possibilities of such optical processors and that represent applications with high computational loads [10,16-19].

Applications satisfying such constraints exist in many areas and represent the class of problems for which such optical systolic array processors are applicable and most appropriate.

ACKNOWLEDGMENT

The support of this research by the NASA Lewis Research Center (Grant NAG 3-5) and its partial support by the Air Force Office of Scientific Research (Grant 79-0091) is hereby acknowledged, together with many helpful meetings and discussions with Anjan Ghosh and Professor C. P. Neuman of the Electrical Engineering Department at Carnegie-Mellon University.

REFERENCES

1. A. Edison and M. Nobel, Optical Analog Matrix Processors, AD646060 (November 1966).
2. P. Mengert et al, U.S. Patent 3525856 (October 6, 1966).
3. M. Monahan et al, Proc. IEEE, 65, 121 (1977).
4. J. Goodman et al, Optics Letters, 2, 1 (1978).
5. D. Psaltis et al, Optics Letters, 4, 348 (1979).
6. M. Carlotto and D. Casasent, Applied Optics, 21, 147 (1982).
7. H. J. Caulfield et al, SPIE, 388 (Jan.1983).
8. H. J. Caulfield et al, Optics Communication, 40, 86 (1981).
9. D. Casasent, Applied Optics, 21, 1859 (1982).
10. D. Casasent et al, Applied Optics, 22, 115 (1983).
11. D. Casasent et al, Proc. EOSD, 311 (1981).
12. J. W. Goodman and M. S. Song, Applied Optics, 21, 502 (February 1982).
13. D. Psaltis et al, SPIE, 232, 151 (1980).
14. R. Athale et al, Applied Optics, 22, 368 (1983).
15. J. Jackson and D. Casasent, Applied Optics (Submitted).
16. D. Casasent and Anjan Ghosh, SPIE, 388, (January 1983).

17. D. Casasent and Anjan Ghosh, Applied Optics (Submitted).
18. D. Casasent and Anjan Ghosh, Optics Communication (Submitted).
19. Anjan Ghosh and D. Casasent, Applied Optics (Letter) (Submitted).
20. D. Casasent and C. Neuman, NASA Conference, Publication 2207 (NTIS), August 1981.

9. NONLINEAR LOCAL IMAGE PREPROCESSING USING COHERENT OPTICAL TECHNIQUES

Reprinted from **Applied Optics**, Vol. 22, page 808, March 15, 1983

Nonlinear local image preprocessing using coherent optical techniques

David Casasent and Jiabi Chen

Two coherent optical systems are described that can realize local nonlinear preprocessing operators such as the Sobel edge-enhancement function in parallel on a 2-D input image. By local, we refer to the size of the image region rather than a nonstationary process. Realization of such operators using a *multiple-exposure* matched spatial filter and a computer-generated hologram is discussed. Experimental results using these techniques for 3 × 3 and 5 × 5 Sobel operators, respectively, are presented. Our techniques can be extended to larger window sizes and other edge-enhancement operators. These new operators are achieved by novel coherent systems using complex arithmetic with magnitude evaluation of the output pattern.

I. Introduction

Essentially all image processing applications and systems utilize local nonlinear preprocessing operators such as median filters and various types of edge-enhancement operators.[1-5] The computational load for realizing these nonlinear local operators is quite high, and although many of them can be realized to various degrees of real time in pipelined digital electronics,[5] it is worthwhile to consider the realization of such local operators on optical processors. This is especially true when such operators can be combined with synthesis of an optical matched spatial filter and when optical processors are being considered for other portions of the processing. Conventional optical spatial filtering systems can achieve low-pass high-pass bandpass and similar linear and global spatial filtering operations. The results of global operations differ from local ones (e.g., the mean of an entire image can differ considerably from the mean of each local 3 × 3 image region). Nonlinear preprocessing operators provide improved SNR performance[1,6,7] over simpler linear edge-enhancement operations such as high-pass filtering. Such local nonlinear preprocessing operations are thus quite different from the conventional linear and global optical spatial filtering functions. Their study and realization on an optical processor represent a new class of most useful operations that we consider. We restrict attention to nonlinear edge-enhancement operators. Our concern is not with image distortions produced by nonlinear operators and which edge-enhancement operator is best (for a comparison, see Refs. 6, 7, and 1, pp. 497–99), but rather we are concerned with the design and demonstration of several optical systems to realize some nonlinear local operators.

In Sec. II, we review several possible nonlinear local operators and the features of each. We also provide a brief summary and motivation for the different kinds of edge-enhancement operators that have been found useful for different applications. All these involve some form of derivative operation. Our discussion emphasizes the Sobel edge-enhancement operator, although the techniques we propose can directly be extended to many nonlinear local operators. We also briefly discuss nonlinear operators of larger window sizes (5 × 5) and note that many types of operators are possible if larger window functions are available. In Sec. III, we describe our new technique for realizing local nonlinear image processing operations optically. In Sec. IV, we detail the optical realization of a 3 × 3 Sobel operator using multiple-exposure matched spatial filter techniques and include experimental verification of our work. In Sec. V, we consider use of a computer-generated hologram to realize such operations, and we demonstrate this technique for a 5 × 5 Sobel operator.

II. Local Operators

In the processing of multisensor, IR, and other types of imagery, the edges in the image are well known to be useful features for target detection and classification. Many types of edge detection and edge-enhancement operations have been proposed, but little theoretical analysis has been presented to address the types of edge-enhancement needed for various pattern recognition applications. We briefly summarize some of the available literature on this topic we have found to be useful. This provides useful motivation for direction

---

The authors are with Carnegie-Mellon University, Department of Electrical Engineering, Pittsburgh, Pennsylvania 15213.

Received 22 September 1982.
0003-6935/83/060808-07$01.00/0.

into the types of edge-enhancement operations that are needed. Any such general treatment requires careful attention to (1) the forms assumed for the spectra of the signal and the noise (Is the noise white? Is the same spectra assumed for both the signal and the noise?); (2) the pixel correlation measure η for the image ($\eta \simeq 1$ for a highly correlated image, $\eta \ll 1$ for an uncorrelated image; η is closely related to the correlation length and correlation coefficient of the image data); and (3) the performance measures used (e.g., peak-to-sidelobe ratio of the correlation output or SNR at the correlation peak; the second measure is appropriate for image registration applications). We consider the applications of pattern recognition and image registration, and thus we use the performance measures of the peak-to-sidelobe ratio (PSR) and the SNR of the output correlation plane. We note that these are significantly different from other measures[1,6,7] of performance previously considered for edge-enhancement techniques (e.g., sharpness of the edge, noise in the edge-enhanced image).

Let us now recall some prior results on the edge-enhancement preprocessing required for our applications. In Ref. 8, it is shown that if the image spectrum is white ($\eta \ll 1$), maximization of PSR requires no preprocessing. However, if the image is highly correlated ($\eta \simeq 1$), maximization of PSR requires the first derivative of each image to be formed prior to correlation. These results were obtained assuming no noise in the input image. If noise is present, it has been shown[9,10] that maximization of SNR requires a derivative operation involving the sum of the original image, its first derivative, and a mixed derivative term. If $\eta \simeq 1$ for the images (and no noise is present), one can show[11] that maximization of PSR can be achieved by forming the second derivative of the reference image rather than the first derivative of each image separately. This follows[12,13] since the correlation of the first derivatives of two images with $\eta \simeq 1$ is the second derivative of the correlation of the original two images and since the order of the operations can be reversed (because they are commutative). A more detailed investigation[12,13] shows that all above results are only appropriate for imagery for which the noise and the signal have the same statistics. We have also shown[12,13] that for highly correlated images ($\eta \simeq 1$), the necessary weighting to maximize SNR (as well as PSR) is well approximated by the |Grad| operator (followed by histogram reshaping). Several of these results are also present in Ref. 14 in a somewhat different form.

Our present purpose is not to derive or discuss the optimum edge-enhancement preprocessing operator required but rather to note the nature of the preprocessing required in different pattern recognition and image registration applications. From the prior brief remarks, we have seen that the problem being addressed (pattern recognition or image registration, target detection or target location, etc.) and the nature of the image (correlated or uncorrelated pixels, $\eta \simeq 1$ or $\eta \ll 1$) affect the optimal preprocessing required. We have also seen that all of the preprocessing noted in Refs. 8-14 involves some form of derivative operator (first, second, mixed, Grad, etc.), and thus attention to this general class of nonlinear operators merits attention with particular emphasis on the |Grad| operator.

Let us next address how such operations are conventionally performed. Digital edge-detection techniques generally involve the convolution of local linear operators (3 × 3 window sizes are common) with the input image $f(x,y)$. These linear operators all provide various approximations to first or second derivatives of the input image with various window masks. The simple 2 × 2 masks to realize $\partial f/\partial x$ and $\partial f/\partial y$ are

$$\frac{\partial}{\partial x} = \begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}, \quad \frac{\partial}{\partial y} = \begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix}. \tag{1}$$

A much better approximation to the directional derivatives is possible with the 3 × 3 operators

$$\frac{\partial}{\partial x} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, \quad \frac{\partial}{\partial y} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}. \tag{2}$$

Similar operators (e.g., compass gradient masks[3]) are possible for realization of the derivatives in different diagonal directions. Convolution masks that approximate the Laplacian are also possible. These operators consist of various approximations to the second derivative. For example, the second derivative can be estimated as the differences of the first differences. For image pixel (m,n), this approximation yields (for the second derivative in x)

$$\partial^2 f/\partial x^2 = f(m-1,n) - 2f(m,n) + f(m+1,n). \tag{3}$$

We can realize Eq. (3) by convolving the image $f(m,n)$ with a 3 × 3 mask function such as

$$\frac{\partial^2}{\partial x^2} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4}$$

One can realize other types of Laplacian operator by convolution with other masks such as

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{5a}$$

or

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \tag{5b}$$

In Eq. (5a), the 3 × 3 operator shown is obtained by summing individual operators such as the one in Eq. (4). In Eq. (5b), the Laplacian is formed by differencing along diagonals rather than rows and columns. Many other types of Laplacian operator can be produced; e.g., Eqs. (5a) and (5b) can be combined, or one can average the difference along three columns or three rows. In each case, these 3 × 3 operators are applied to each pixel in the input image, and the central pixel in the corresponding 3 × 3 region of the input image is replaced by the product of the mask weights shown in Eqs. (1)–(5) and the corresponding pixel values in the *input* image.

These linear edge-enhancement techniques are simple to realize in a pipelined processor. However, they amplify high spatial frequencies (i.e., noise) as well as the edges in the image. Preferable techniques employ nonlinear operators in which nonlinear combinations of the results of the basic linear operators are produced. Figure 1 shows a block diagram[6] for an edge-detection algorithm for the magnitude gradient operator. The output image from the local operator is thresholded to yield the final edge-enhanced output pattern. To describe these operators and how they are realized, a specific numbering procedure for the pixel values a_i in the input image surrounding the general input point (m,n) whose pixel value is $f(m,n)$ has been adopted:

$$\begin{array}{|c|c|c|} \hline a_0 & a_1 & a_2 \\ \hline a_7 & f(m,n) & a_3 \\ \hline a_6 & a_5 & a_4 \\ \hline \end{array} \, . \qquad (6)$$

We consider the |Grad| operator (noted earlier to be useful for preprocessing) as realized by a Sobel local operator.[1] For this operator, the output image value $g(m,n)$ at pixel (m,n) is

$$g(m,n) = (X^2 + Y^2)^{1/2}, \qquad (7)$$

where the local operators are

$$\begin{aligned} X &= (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6), \\ Y &= (a_0 + 2a_1 + a_2) - (a_6 + 2a_5 + a_4). \end{aligned} \qquad (8)$$

Recall that the a_i values in Eqs. (8) correspond to the intensities of the pixels surrounding the general input image point (m,n) and that the operation described by Eqs. (7) and (8) is applied at each input image pixel. Thus Eqs. (7) and (8) describe the final image pixel value for pixel (m,n) as the nonlinear combination of its surrounding neighbors. [Note that the original values of image pixel (m,n) is not used in computing the corresponding output pixel value and that the sum of the weights in the Sobel masks is 0.] We can thus describe this realization of |Grad| by the two linear local mask operators

$$X = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \qquad Y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \qquad (9)$$

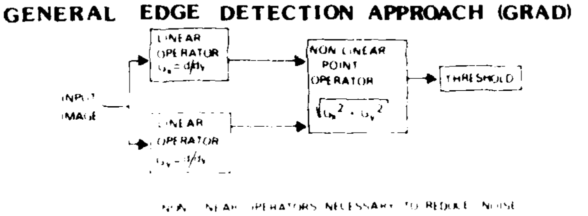


Fig. 1. Block diagram of the Sobel operator as a linear operator function and a nonlinear point operator (after Ref. 6).

Inspection of Eq. (9) and comparison with Eq. (2) show that the linear operators used in the Sobel masks correspond to the partial derivatives in x and y averaged over the image pixels in three horizontal and vertical rows and columns, with the central row and column weighted most heavily. In Ref. 15 it was shown that the averaging of difference estimates over several rows and columns of the image pixels with a decreased weighting for pixels further from the mask center provided improved performance. The Sobel operator described by Eqs. (7) and (8) includes both of these desirable features and is thus a most attractive edge-enhancement operator. It was thus chosen for our optical experiments.

To best describe the operation performed by the Sobel, we write the output function as

$$g = [(\langle \partial f/\partial x \rangle_y)^2 + (\langle \partial f/\partial y \rangle_x)^2]^{1/2}, \qquad (10)$$

where $\langle\ \rangle_x$ and $\langle\ \rangle_y$ denote averages over x and y, respectively. This formulation emphasizes that each local *operator in* Eqs. (9) *performs* the first difference approximation to the partial derivative of f in x and y averaged over y and x, *respectively*. A detailed derivation of Eq. (10) follows directly.[16] We detail only the Sobel nonlinear operator, but the technique is directly applicable to realization of many other operators such as the Roberts, Prewitt, Kirsch, and others by simply changing the mask weights in Eq. (9). An alternate formulation is needed to describe edge-enhancement operators such as compass gradients.

In the optical realization of such nonlinear preprocessing operators, we can easily use larger mask windows (e.g., 5 × 5). For such cases, the flexibility possible increases significantly as many choices are possible for the various mask elements. In one quite simple 5 × 5 Sobel-type operator, Eq. (9) would become

$$X = \begin{bmatrix} -1 & -1 & 0 & +1 & +1 \\ -1 & -1 & 0 & +1 & +1 \\ -2 & -2 & 0 & +2 & +2 \\ -1 & -1 & 0 & +1 & +1 \\ -1 & -1 & 0 & +1 & +1 \end{bmatrix}, \qquad Y = \begin{bmatrix} +1 & +1 & +2 & +1 & +1 \\ +1 & +1 & +2 & +1 & +1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -2 & -1 & -1 \\ -1 & -1 & -2 & -1 & -1 \end{bmatrix}. \qquad (11)$$

Many other 5 × 5 edge-enhancement operators can be formulated. We will refer to the 5 × 5 nonlinear local operator defined by Eqs. (7) and (11) as a 5 × 5 Sobel. We consider its optical realization in Sec. V.

As described in Ref. 15, averaging over larger neighborhoods (five rather than three elements) greatly improves the performance of the operator when the input SNR is low. Such operators produce wider edges, but this is not of major concern in the pattern recognition cases we consider, and edge thinning is also a possibility if necessary. As noted in Ref. 15, different weights (besides 0, 1, and 2) for the elements of the masks can also significantly improve performance of the operator. Many other operations besides the 5 × 5 Sobel in Eqs. (11) can thus be achieved by judicious choice of the mask elements. One possible operator is the realization of two successive Sobels by one pair of 5 × 5 masks. Another possibility is the reduction of a 5 × 5 input image region to a 3 × 3 region (in which each pixel value is the result of a 3 × 3 Sobel applied to the corresponding 3 × 3 region of the original input image) and the subsequent reduction of these 3 × 3 image regions to one pixel value that is the Sobel of the 3 × 3 set of pixel values obtained from the first Sobel. This sequence of two Sobels is one form of the second derivative of the original image. Such operators should be realizable in extended 5 × 5 masks. Other local masks can be used to achieve mixed derivative operations. With the use of 5 × 5 masks, realization of Laplacian and similar operations is also directly possible. In general, use of window operators of higher order are more easily realizable on the optical systems we consider, and they should be able to achieve the types of nonlinear local operators noted in Refs. 8–14 as desirable.

III. Nonlinear Local Optical Operators

Optical systems can perform the linear spatial differentiation operation of various techniques.[17] The one of interest to us is performed on an optical matched spatial filter correlator. The system's output is the convolution of the input image $f(x,y)$ and the reference function h or the impulse response of the system. (The complex Fourier transform H of h is recorded on the matched spatial filter.) When h is two delta functions separated by d, the system's output is

$$\text{output} = f \ast h = f(x,y) \ast [\delta(x + d,y) - \delta(x,y)] \simeq df/dx, \quad (12)$$

or the first-difference linear two-point approximation to the 1-D spatial differentiation of the input image $f(x,y)$. This is still a linear operation and to realize the Sobel or other such nonlinear operators, we must develop new optical techniques.

We begin by rewriting the Sobel output function in Eq. (7) as

$$g(m,n) = (X^2 + Y^2)^{1/2} = [|X + jY|^2]^{1/2} = |X + jY|. \quad (13)$$

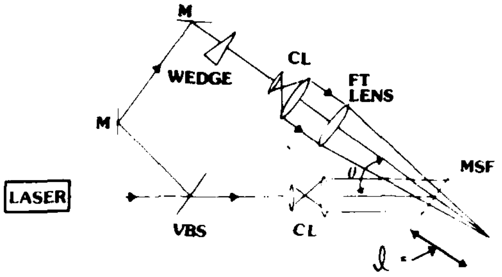


Fig. 2. Multiple-exposure Sobel filter synthesis system: M, mirror; VBS, variable beam splitter; CL, collimation lens system; FT lens, Fourier transform lens; MSF, matched spatial filter.

This complex arithmetic description is most useful. Should a digital processor capable of direct complex arithmetic in one pass be developed and implemented, one can realize the Sobel operator as in Eq. (13). This is not available in present digital systems; however, an optical system to realize Eq. (13) is possible. The light amplitude distribution at the output of this optical system will be given by Eq. (13). After detection, the square of Eq. (13) will be produced in an optical system. This represents no significant difference since the square-root operation in Eq. (13) is often omitted in the digital realization of such operators with no adverse effects.[6] This issue is not essential for preprocessing or for the realization of a nonlinear filter. We thus consider the realization of an optical system whose output amplitude at each point is

$$\begin{aligned} X + jY = {} & [(a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6)] \\ & + j[(a_0 + 2a_1 + a_2) - (a_6 + 2a_5 + a_4)]. \end{aligned} \quad (14)$$

where a_i are the pixel values of the input image in Eq. (6). We describe Eq. (14) as in Eq. (12) by the convolution of the input function $f(x,y)$ with a sum of delta functions at eight spatial locations with complex-valued weights for each delta function:

$$g(x,y) = f(x,y) \ast \sum_{m,n} k_{m,n}\delta(x - md,y - nd) \quad (15a)$$

$$\begin{aligned} = {} & f(x,y) \ast [(1 + j)\delta(x - d,y - d) + 2\delta(x - d,y) \\ & + (1 - j)\delta(x - d,y + d) + 2j\delta(x,y - d) \\ & - 2j\delta(x,y + d) - (1 - j)\delta(x + d,y - d) \\ & - 2\delta(x + d,y) - (1 + j)\delta(x + d,y + d)]. \end{aligned} \quad (15b)$$

where d is the spacing between pixels in the input image and where the weights and locations of each delta function are obtained from Eq. (14).

Our proposed technique for the optical realization of nonlinear local preprocessing operators such as the Sobel is thus to employ an optical correlator with a matched spatial filter whose impulse response is a weighted sum of delta functions. The necessary weights are complex-valued. However, we can achieve these by holographic techniques and appropriate phase shifting as we detail in Sec. IV. We can also achieve the required impulse response by use of a computer-generated hologram as we detail in Sec. V.

IV. Realization Using Multiple-Matched Spatial Filters

The desired impulse response in Eq. (15) could be achieved by forming the holographic matched spatial filter (MSF) of an input function containing delta functions (apertures) at the correct locations and of the correct radii (to adjust the intensity) and with the necessary phase factors achieved by placing $\lambda/4$ or $\lambda/2$ plates behind the appropriate apertures. We chose to form the desired MSF by eight multiple exposures on one plate (because the necessary light levels and exposure times were easier to achieve). The MSF synthesis system used is shown in Fig. 2. It is a conventional

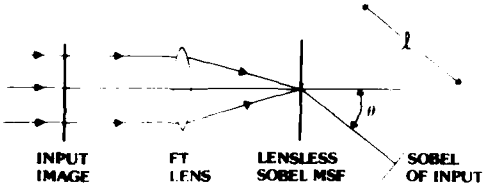


Fig. 3. System to produce the Sobel edge-enhanced image from a multiple-exposure matched spatial filter.

holographic MSF system with the addition of the wedge and the FT lens. The FT lens allows a lensless MSF to be recorded (the MSF plus an FT lens with focal length $f_L = l = 250$ mm) as we have described and used previously.[18] We ignore this FT lens for our present discussion.

The MSF recorded at each exposure is the interference of two plane waves and thus produces a sine wave grating whose impulse response is a delta function. The amplitude of each delta function is set by varying the exposure. This does not alter the modulation of the grating being recorded, since the bias level is also changed for each exposure; however, the amount of light diffracted into the output correlation plane does vary proportionally to the exposure as is desired. The location of each delta function was controlled by translating the FT lens in its plane (and thereby varying the spatial frequency of the recorded grating). We produced the necessary complex-valued strength for each delta function by shifting the wedge in 1-D in its plane. After synthesis of this Sobel MSF in Fig. 2, we place it in Fig. 3 and in the output plane obtain the Sobel of the input image.

We now detail these systems for the case of a multiple MSF to realize the 3×3 Sobel of Eq. (14). All lenses in Fig. 2 had $f_L = 760$ mm. In the Sobel correlator (Fig. 3), the first FT lens used $f_L = 500$ mm, and the focal length of the lensless MSF was $l = f_L = 250$ mm. This correlator reconstruction system exhibited a 2:1 demagnification. The highest input spatial frequency we set the system to handle was $u_m = 20$ cycles/mm. The spatial frequency plane in Fig. 3 must thus have an aperture of $2\lambda f_L u_m = 12$ mm. We used 15-mm diameters for all beams in Fig. 2 and for the corresponding diameter of the MSF. To select the distance d between the delta function impulse responses (and similarly the amount d by which to translate the FT lens between exposures), we determine that the smallest edges of interest in the 8- × 8-mm^2 input image were of 0.2-mm width. In the output plane of Fig. 3, they would be 0.1 mm. We chose a spacing $d = 12.5\ \mu$m between delta functions and a corresponding 12.5-μm FT lens shift. This produced edges in the output of 25-μm width. This will produce sufficient detail to provide a good edge-enhanced output. Note that in this optical Sobel system, one can easily choose the filter to select the minimum input edge size we wish to consider.

The eight exposures in Eqs. (15) were arranged so that successive exposures had a phase difference of $\pi/4$. We now compute the amount by which the wedge in Fig. 2 must be translated between exposures to provide a $\pi/4$ or $\lambda/8$ phase shift. The wedge angle was 4 sec of arc, and thus shifting it caused minimal displacements in the input beam and thus did not cause misalignment of the system. To measure the angle α of the wedge, we measured the fringe spacing d_1 produced by interference from the front and back surfaces and found $d_1 = (\lambda/2)/\alpha n = 10$ mm, where n is the index of refraction of the wedge. If the wedge is shifted vertically by a distance d_u, the beam traveling through it sees a path difference $\alpha d_u (n - 1)$ from its prior value. Substituting for α and equating this path difference to $\lambda/8$, we find that a shift $d_u = 7.5$ mm provides the desired phase factor.

The above procedure was followed, and eight multiple exposures of the MSF plate were made. After development, we inspected the impulse response of the lensless MSF and found it to be as required. We then placed this MSF in the lensless correlator of Fig. 3. A converging beam system (with the input behind the first FT lens) was not possible since such a system was not used during synthesis of the MSF. The additional phase term introduced in the converging beam system would thus not be canceled by this MSF. If the angle of the input plane wave in Fig. 2 were altered, the Sobel version of the input image would appear on-axis in Fig. 3 rather than at an angle $\theta = 15°$ to the optical axis as shown. The input image size used was 8×8 mm^2, although larger image sizes could be accommodated in the system used. An input text image and the corresponding Sobel edge-enhanced |Grad| output image obtained on this system are shown in Fig. 4. The results

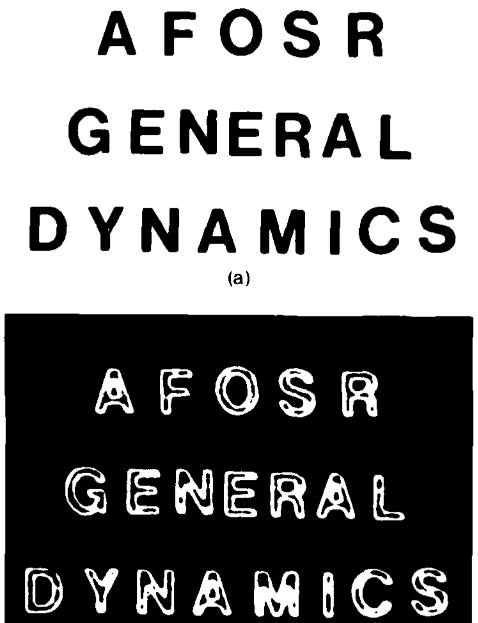


Fig. 4. Input image (a) and Sobel edge-enhanced image (b) using the multiple-exposure matched spatial filter system.

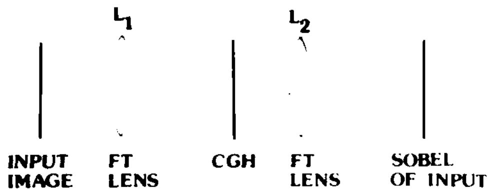


Fig. 5. System to produce the Sobel edge enhanced image using computer-generated holograms.

appear as expected. All input lines are $>2d$ in width and thus have output image edges of width $2d$.

V. Realization Using a Computer-Generated Hologram

It is possible to produce the necessary FT plane filter function for the Sobel (or other) edge-enhancement operators by use of computer-generated holograms (CGH).[19] The general form for the desired impulse response for a Sobel operator is

$$\sum_m \sum_n k_{mn}\delta(x - md,y - nd). \tag{16}$$

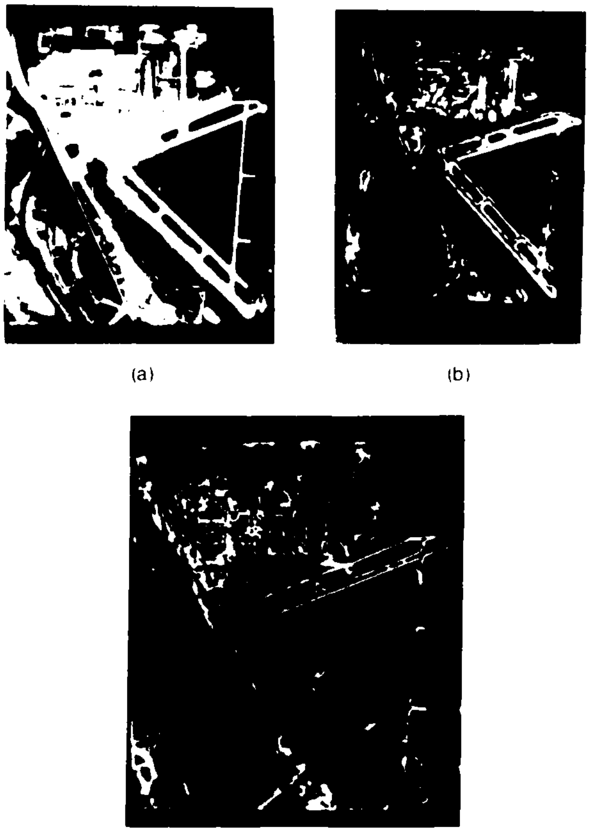
To realize such an impulse response with a CGH, we Fourier transform[16] sample it, and for each sampling cell in the CGH we record an amplitude A and a phase ϕ. In (16), we ignore (for simplicity of the expressions presented) the fact that this CGH is an off-axis hologram. The transmittance of the FT plane CGH is thus of the general form

$$A(x,y)\exp[j\phi(x,y)], \tag{17}$$

where (x,y) in (17) refers to the coordinates of the FT plane where the CGH will be placed. This CGH filter technique is quite appropriate for Sobel operators with large neighborhoods (i.e., 5×5), since 25 multiple exposures are quite difficult to achieve with the holographic MSF system of Fig. 2. As discussed earlier, such filters are much more flexible and are of significant importance when the input has a low SNR.

We chose to demonstrate this technique for the 5×5 Sobel in Eqs. (11). We now describe the design and fabrication of the CGH. The CGH contains $L \times W$ cells, each of size $dx \times dy$ and each containing a transparent aperture of size $l \times w$. We fix $w = dx/8$ and control the aperture transmittance of each cell by varying l. To reduce noise and produce a good image reconstruction with a CGH, we require $dx = dy \ll \lambda f_L/b$, where b is the size of the output plane (assumed to be square for simplicity). In our case, the CGH reconstruction or its impulse response is a pattern of delta functions. The maximum dimension of this output pattern is $b = 4d$. For our CGH, we thus used $d = 11$ μm and $dx = dy = 0.03$ mm. The size of the Calcomp plotter pen, the 40× reduction available, and the resolution of the FT plane film allowed us to fabricate a CGH with only 250×250 cells in 7.5×7.5 mm^2. This 7.5-mm FT plane width will only pass input spatial frequencies below 5 cycles/mm (for the $f_L = 500$-mm FT lens). Thus input image detail smaller than 0.2 mm will be lost in the reconstructed output from this filter. A different CGH synthesis facility would provide less image resolution restrictions.

The CGH corresponding to Eqs. (11) was computed, plotted, and photoreduced as described above. It was then placed in the FT plane of the system of Fig. 5 with the focal length of the first lens being 500 mm and that of the second lens being 115 mm. The input and output images obtained for an aerial input photograph are shown in Figs. 6(a) and (b). For comparison, the output from the 3×3 multiple-exposure MSF Sobel applied to this same image is included in Fig. 6(c). The higher resolution of the 3×3 Sobel is apparent as are the broader lines present in the 5×5 Sobel output. For a 5×5 Sobel and an input with edges of $1d$ width, we expect the edges in the output to be $3d$ wide (compared with $2d$ for the 3×3 Sobel). Our data verify this since all edges in the output from the 5×5 Sobel [Fig. 6(b)] are wider than those in Fig. 6(c) (obtained using a 3×3 Sobel). The edges that are not present in Fig. 6(b) were (in general) lost by the limited 5-cycle/mm maximum bandpass of the CGH.


(a)

(b)

(c)

Fig. 6. Input image (a) and edge enhanced output images (b) using computer generated holograms (5 × 5 Sobel) and (c) a multiple matched filter (3 × 3 Sobel).

The available aerial image used [Fig. 6(a)] contains lines (i.e., two edges) rather than one edge separating different textured regions. Thus, in analyzing the output images shown, we must recall that the edge-enhanced version of a line consists of two edges of width $2d$ or $3d$ (this also depends on the sharpness of the edges) separated by the width of the line. The spacing d chosen and the spatial frequency bandwidth and resolution of the Sobel filters also affect the observed outputs. A detailed analysis of the input image used and the filter parameters chosen confirms that the results are as expected. To demonstrate and quantify the SNR of the output images and the SNR improvement obtained using different 3×3 and 5×5 operators, a textured image with edges (not lines) separating regions of different variances should be used. This typifies future work to be performed on such processors.

VI. Summary and Conclusion

As image sizes increase, the nonlinear edge-enhancement preprocessing required becomes quite expensive to perform in real time by digital or electronic methods. The new techniques we have developed result in a coherent optical system capable of nonlinear local operators. We derived and detailed this technique for the case of the Sobel operator. However, it can easily be extended to many nonlinear local operators (Prewitt, Roberts, as well as the Laplacian and higher-order difference operators). We have described and demonstrated two techniques for realization of such functions optically. The multiple MSF technique requires a specially fabricated positioning system for good results, but such a system is quite easily produced if needed. We demonstrated good 3×3 Sobel edge-enhanced output data on our initial system with results (linewidths and SNR) in agreement with those expected by theory. We also described and demonstrated a 5×5 Sobel using CGHs. The results were quite attractive and again agreed with theory. A particularly attractive aspect of the optical realization of nonlinear local operators is that larger window sizes and higher-order operators can be used with no loss in processing speed. It is also possible to combine MSF, Sobel, and lensless filters into one plane. Such a system would realize the compactness of a lensless MSF correlator and the preprocessing provided by optical Sobel operators. To obtain the full benefit from any such system, one should first determine the optimal nonlinear preprocessing operator required. The production of this operator with the desired window size and order can then be achieved by parallel lensless nonlinear optical preprocessors such as we have described.

We thank the Air Force Office of Scientific Research (grant AFOSR 79-0091) for support of D.C. and for support of our optical data processing program of which this is one project aspect. We also thank B.V.K. Vijaya Kumar for many fruitful discussions.

Jiabi Chen is a visiting scholar from Huazhong Institute of Technology, Wuhan, China.

References

1. W. Pratt, *Digital Image Processing* (Wiley, New York, 1977).
2. L. Davis, Comput. Graphics Image Process. **4,** 248 (Sept. 1975).
3. G. Robinson, Comput. Graphics Image Process. **6,** 492 (1977).
4. R. Duda and P. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
5. G. Robinson and J. Reis, "A Real-Time Edge Processing Unit," in *Proceedings, IEEE Workshop on Picture Data Description and Management*, Chicago, Apr. 1977, (IEEE, New York, 1977), pp. 155-164.
6. I. Abdou and W. Pratt, Proc. IEEE **67,** 753 (1979).
7. J. Fram and E. Deutsch, Proc. IEEE **C-24,** 616 (June 1975).
8. A. Arcese, P. H. Mengert, and E. W. Trombini, IEEE Trans. Inf. Theory **IT-16,** 534 (1970).
9. C. D. McGillem and M. Svedlow, IEEE Trans. Geosci. Electron. **GE-14,** 44 (1976).
10. C. D. McGillem and M. Svedlow, IEEE Trans. Geosci. Electron. **GE-15,** 257 (1977).
11. W. K. Pratt, IEEE Trans. Aerosp. Electron. Syst. **AES-10,** 353 (1974).
12. Y. Barniv and D. Casasent, Proc. Soc. Photo-Opt. Instrum. Eng. **292,** 25 (1981).
13. Y. Barniv, H. Mostafavi, and D. Casasent, Proc. Soc. Photo-Opt. Instrum. Eng. **238,** 156 (1980).
14. J. Prewitt, "Object Enhancement and Extraction," in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds. (Academic, New York, 1970).
15. I. Abdou, Report 830, U. Southern California, Image Processing Institute (1973).
16. J. Chen and D. Casasent, at LIA Conference, International Congress on Applications of Lasers and Electro-Optics, Boston (Laser Institute of America, September 1982).
17. S. K. Yao and S. H. Lee, J. Opt. Soc. Am. **61,** 474 (1971).
18. F. Caimi *et al.*, Appl. Opt. **19,** 2653 (1980).
19. W. H. Lee, "Computer-Generated Holograms: Techniques and Applications," in *Progress in Optics, Vol. 16* (North-Holland, New York, 1978).

10. ERRORS IN OPTICAL COMPUTATION OF CORRELATION COEFFICIENTS

Reprinted from **Applied Optics**, Vol. 22, page 209, January 15, 1983

Errors in optical computation of correlation coefficients

B. V. K. Vijaya Kumar

Carnegie-Mellon University, Department of Electrical
Engineering, Pittsburgh, Pennsylvania 15213.
Received 10 September 1982.
0003-6935/83/020209-03$01.00/0.

A variety of methods have been proposed for solving pattern recognition problems using optical methods.[1] Some of these involve generating a synthetic discriminant function that is a weighted combination of the functions available as training data. The weights for the combination are obtained either from the principal component of the data set[2] or by imposing different deterministic constraints that ensure the separation of the classes.[3] Caulfield[4] has shown that these problems can be unified as solving the following matrix-vector equation:

$$\mathbf{R}\mathbf{w} = \mathbf{c}, \tag{1}$$

where **R** is the matrix of correlation coefficients among the members in the training set, **w** is the vector of desired weights, and **c** is the vector of constraints.

Before we can solve Eq. (1) for weight vector **w**, we have to determine the correlation coefficient matrix **R** from the given training data. Various elements of **R** can be determined optically[5] by forming cross correlations of all pairs among the training data set and then noting the peak values. Through this Letter we would like to point out two sources of error in the computation of **R**. Errors in **R** lead to inaccurate weights which, in turn, can result in a discriminant function that does not satisfy the constraints.

A well-known[6] source of error in the estimation of any statistical parameter is the finite length of the observation. If the input functions $x(t)$ and $y(t)$ are of length T and bandwidth B, we can easily show[6] that the standard deviation of the estimated cross-correlation coefficient ρ_{xy} is

$$\sigma_1 = \rho_{xy}[(1 + \rho_{xy}^{-2})/(2BT_{\text{total}})]^{1/2}, \tag{2}$$

where T_{total} denotes the total length of the signals used to estimate ρ_{xy}. The correlation coefficient ρ_{xy} is related to the cross-correlation function by normalizing constants that are related to the energies in the signals $x(t)$ and $y(t)$. We note from Eq. (2) that σ_1 is inversely proportional to $(BT_{\text{total}})^{1/2}$, where (BT_{total}) denotes the total time-bandwidth product of the signal. This type of error can be reduced by increasing the length of observation. In the case of images, the space-

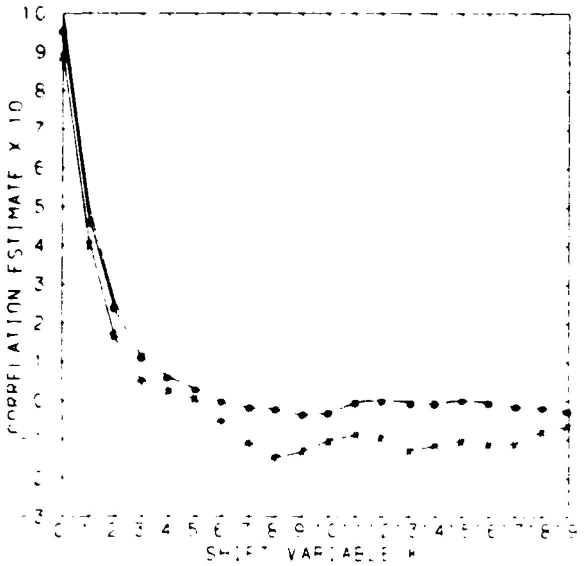


Fig. 1. Correlation functions for $\rho_0 = 0.5$: solid line, ideal correlation; line with Xs, observed correlation for $T = 30$; line with Os, observed correlation for $T = 100$.

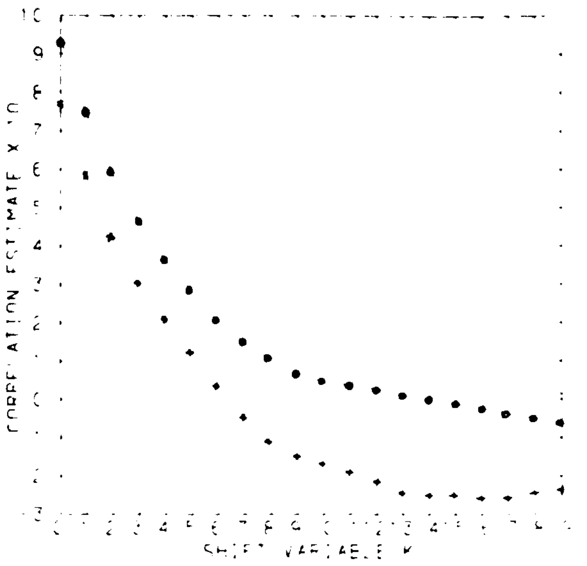


Fig. 2. Correlation functions for $\rho_0 = 0.8$: solid line, ideal correlation; line with Xs, observed correlation for $T = 30$; line with Os, observed correlation for $T = 100$.

bandwidth product easily exceeds 1000 and σ_1 can be considered insignificant. But with increasing use of 1-D devices such as acoustooptic cells, even for image processing,[7] care should be taken to see that this type of error is not significant.

The second source of error in estimation seems to be less well known among the researchers in optical pattern recognition. The rest of this Letter will be devoted to a discussion of this error source. When two functions $x(t)$ and $y(t)$ are correlated using matched spatial filter (MSF) techniques, the nonlinear properties of the recording medium in the MSF plane force the average values of $x(t)$ and $y(t)$ to go to zero.[8] This property of optical correlators to force the mean to zero does not pose a problem in the conventional applications of cross correlators where the primary objective is to locate the peak in correlation plane and the value of this peak is not important. Then the effective correlation is between $x'(t)$ and $y'(t)$ where

$$x'(t) = x(t) - \langle x(t) \rangle, \qquad y'(t) = y(t) - \langle y(t) \rangle, \tag{3}$$

and where $\langle\ \rangle$ denotes the average over a duration of T. Even in digital estimation of ρ_{xy}, we end up using only $x'(t)$ and $y'(t)$, because the coefficients in the Fourier transform corresponding to zero frequencies are usually suppressed.

Let $\rho_{xy}(\tau)$ denote the cross-correlation coefficient of $x(t)$ and $y(t + \tau)$. Then $\rho_{xy}'(0)$, obtained from the mean removed signals $x'(t)$ and $y'(t)$, is related to $\rho_{xy}(\tau)$ as[9]

$$\rho_{xy}'(0) = \rho_{xy}(0) - (2/T) \int_0^T [1 - (\tau/T)]\rho_{xy}(\tau)d\tau. \tag{4}$$

Note from Eq. (4) that we are underestimating $\rho_{xy}(0)$ because of the use of mean removed signals $x'(t)$ and $y'(t)$. To better understand the magnitude of this error, let us assume $\rho_{xy}(\tau)$ to be an exponential function

$$\rho_{xy}(\tau) = \rho_0|\tau| = \exp[(-a|\tau|)], \tag{5}$$

where ρ_0 is inversely related to the bandwidth of the signal and a is a measure of the signal bandwidth. For this correlation function we can show that

$$\rho_{xy}'(0) = 1 - (2/aT) + (2/a^2T^2) - (2/a^2T^2)\exp(-aT) \tag{6}$$

For large (aT) values, the above relation can be approximated as

$$\rho_{xy}'(0) = 1 - (2/aT) \tag{7}$$

Thus the removal of mean from a signal $x(t)$ can cause a significant bias in the estimated value of $\rho_{xy}(0)$. This can be a significant problem when we use low time-bandwidth devices to process long duration signals or large images. Since these devices can accommodate only a limited time-bandwidth product at any instant, mean removal can cause a significant bias in $\rho_{xy}(0)$ estimated from each segment.

To illustrate the above problems, we generated digital sequences of lengths $T = 30$ and $T = 100$ with ρ_0 values of 0.5 and 0.8. Ideal autocorrelation functions according to Eq. (5) and the autocorrelation functions estimated from the mean removed signals are shown in Figs. 1 and 2. The results in these figures were obtained by averaging over fifty independent sequences. Thus the random error in Eq. (2) resulting from the finite signal length should be negligibly small. But the bias errors due to the mean removal do not average out and can be clearly seen in Figs. 1 and 2. In Fig. 1, ρ is 0.8 (small TBW) and biases are large, whereas in Fig. 2, ρ is 0.5 (large TBW) and biases are small. In fact, the expected values for $\rho_{xy}'(0)$ according to Eq. (7) are 0.972 and 0.911 for $T = 100$, whereas we observe 0.964 and 0.937 experimentally. For $T = 30$, theoretically expected values are 0.904 and 0.703, whereas the experimentally obtained values are 0.900 and 0.775. As can be seen, the experimental results are close to the theoretically expected values.

We note from our results that the correlation coefficients obtained from optical MSF correlators should be accepted only if the time-bandwidth products of the signals are large. Otherwise, the biases in them should be corrected by adding $(2/aT)$ where (aT) is the time-bandwidth product. Often we do not know the *a priori* value of a and this must also be estimated. This causes variance in estimated correlation coefficients and one has to take this into account.

References

1. D. Casasent, Proc. IEEE **67**, 813 (1979).
2. B. V. K. Vijaya Kumar, D. Casasent, and H. Murakami, Opt. Eng. **21**, 43 (1982).
3. H. J. Caulfield and W. T. Maloney, Appl. Opt. **8**, 2354 (1969).
4. H. J. Caulfield, Appl. Opt. **19**, 3877 (1980).
5. C. F. Hester and D. Casasent, Appl. Opt. **19**, 1758 (1980).
6. J. S. Bendat and A. G. Piersol, *Engineering Applications of Correlation and Spectral Analysis* (Wiley, New York, 1980), p. 78.
7. D. Psaltis, J. Opt. Soc. Am. **71**, 198 (1981).
8. B. V. K. Vijaya Kumar and D. Casasent, Opt. Commun. **34**, 4 (1980).
9. E. L. O'Neill and A. Walther, J. Opt. Soc. Am. **67**, 1125 (1977).

11. CORRELATION OF BINARIZED IMAGES

Correlation of Binarized Images

Correlator technique has been used in pattern recognition applications. We introduce a correlation performance measure, peak to background ratio (PBR), defined as the ratio of the peak intensity to the average background variation in a single correlation output. Using this performance measure, we analyze the binary correlator and show that the binarization of the images can result in better performance. We consider a real-time implementation of the binary correlator, using VLSI technology. The analytical results are supported by computer simulations.

I. INTRODUCTION

Matched filter or correlator receiver was originally [1] derived as the optimum linear time-invariant filter to de-

Manuscript received February 26, 1981

tect a known signal in the presence of additive white noise. The performance criterion used in this situation was the ratio of the average value of the correlation output at a particular time instant to the standard deviation of the output at that time instant. This ratio is commonly known as the signal-to-noise ratio (SNR) and has been the optimality criterion in many communication receiver designs.

Recently, correlation operation has found widespread application in image pattern recognition because of advances in coherent optical methods [2] as well as digital methods [3]. In such image correlations, we are interested not only in detecting the presence of a "target" in a noisy "scene," but also in estimating the location of the target in the scene. For this purpose, it is desirable to keep the correlation peak (at the true location) as high as possible while reducing the sidelobe levels all around this peak. As a result, the ratio of average correlation peak to the standard deviation of output correlation "far" from the peak, commonly denoted as peak-to-sidelobe ratio (PSR), is used [4] as a performance measure to evaluate the image correlations. PSR has been widely used to evaluate image correlations in the presence of image distortions such as coordinate transforms [4, 5], frequency plane blurring [6], and input transducer nonlinearity in coherent optical correlators [7].

The arbitrary nature of sidelobe determination in PSR calculations is evident in the selection of position "far" from the peak. In fact, a recent paper [8] talks about the PSR at three different locations: exact registration, near misregistration, and gross misregistration. We feel that a good measure of the correlator performance should utilize the variance at all points in the correlation plane rather than at any single point. This will avoid the possibility of getting a high PSR simply because we happen to choose a correlation plane location where variance was small. In this paper, we utilize a somewhat similar measure known as peak-to-background ratio (PBR). PBR is defined as the ratio of the average correlation peak to the average of standard deviations at all points in the output. One can generalize the PBR by appropriately weighting the standard deviations in the correlation output before averaging them. For example, standard deviations of points close to registration may be given more importance compared with points of gross misregistration.

We analyze the performance of the binary correlator in this paper using PBR. The conventional correlator performs poorly in comparison to some *sequential similarity detection algorithms (SSDA)* [9] in terms of computational efficiency. Such computational considerations have led to renewed interest in the use of correlators where the input signals are hard limited (to two amplitude levels) prior to correlation. Svedlow et al. [10] report the experimental use of such binary correlators, Vijaya Kumar [11] analyzes the performance using PSR, Boland et al. [12] suggest a real-time digital implementation, while Cole [13] proposes an incoherent optical processor for binary correlation. In Section II, we show that the binarization does not appreciably degrade the PBR of a correlator. Our theoretical results are in good agreement with the prior work [11] utilizing PSR as the criterion. We present simulation results in Section III to support the results of our analysis. The main advantage of a binary correlator is that it can be realized without multiplication. It is thus ideally suited for direct digital logic design and we consider an implementation of this using VLSI technology [14] in Section IV.

II. BINARY CORRELATOR PERFORMANCE

Let $\{x_i\}$ and $\{y_i\}$, $i = 1, 2, \ldots, N$ denote the target and scene, respectively. One-dimensional notation is used only for simplicity and extension to the two-dimensional case of images is straightforward. In general, y_i is related to x_i by

$$y_i = x_{i-i_0} + n_i \qquad (1)$$

where i_0 denotes the shift between the two signals and n_i is a sequence of zero mean uncorrelated random variables. Without any loss of generality, we assume i_0 to be zero in the rest of this paper. Then the correlation output is given as

$$C(j) = (1/N) \sum_{i=1}^{N} x(i)y(i+j), \qquad j = 1, 2, \ldots, N \qquad (2)$$

where we have implicitly assumed that $y(i)$ is periodically repeated for values of arguments beyond N. This is precisely what happens when correlations are performed digitally using discreet Fourier transforms [15]. For this correlation output $C(j)$, we define the PBR γ as follows:

$$\gamma = N|C(0)|^2 / \left(\sum_{j=1}^{N} |C(j)|^2 \right) \qquad (3)$$

is the ratio of the correlation peak intensity (when the reference and input are perfectly matched) to the average correlation output over the entire range (not just at $j = 0$ or $j \to 0$). $C(j)$ is in general very small as j increases. Thus we note from (3) that PBR increases as the length N of the signals increases. This supports the experimental observation [16] that the PBR of a correlator is directly proportional to the space bandwidth product of the input data.

We evaluate the performance of the binary correlator in this section. The input signals $\{x_i\}$ and $\{y_i\}$ are subjected to hard limiting before correlation:

$$\hat{x}_i = \begin{cases} a & \text{if } x_i \geq 0 \\ -a & \text{if } x_i < 0 \end{cases} \qquad (4)$$

where the caret denotes binarized variables. We have implicitly assumed above that the input signals $\{x_i\}$ and $\{y_i\}$ are both of zero mean. Such an assumption is justified due to one of the following two reasons. In coherent optical processors [16], the matched filter plane medium usually exhibits a dark central region, thus removing any average value present in the input signals. Average values in images are deliberately removed [3] in many image correlations because it results in an output bias level, thus degrading the output correlation PBR.

The binary correlation peak value in the absence of noise is given by

$$\hat{C}(0) = (1/N)[\sum_{i=1}^{N} (\hat{x}_i)^2] = a^2 \tag{5}$$

which can be seen to be the maximum possible peak value for a given input dynamic range $[-a, +a]$. In digital correlators, correlation peak values may not be important, but in optical correlators, system noises such as detector noise, reflections, speckle noise, and scatter indicate the need for a large correlation peak [11]. Thus binary correlation reduces the effects of system noise in optical correlators.

PBR of binary correlation can be written as

$$\hat{\gamma} = N(a^4)/\sum_{j=1}^{N} [(1/N)\sum_{i=1}^{N} \hat{x}_i\hat{x}_{i+j}]^2 \tag{6}$$

where we have assumed no noise in the input. We consider the effect of noise at a later stage in this section.

It is difficult to evaluate the denominator of (6) unless the input signal sequence $\{x_i\}$ is known. The exact nature of $\{x_i\}$ is rarely known a priori, but we will assume that $\{x_i\}$ is a sample realization from a zero mean Gaussian process with a second-order statistics $R_x[j]$. With such an assumption, we have

$$\begin{aligned} E\{\hat{x}_i\hat{x}_{i+j}\} &= a^2\{\Pr[x_i x_{i+j} > 0] - \Pr[x_i x_{i+j} \le 0]\} \\ &= a^2\{(2/\pi)\sin^{-1}[R_x(j)/R_x(0)]\} \end{aligned} \tag{7}$$

where $E\{\cdot\}$ denotes ensemble average and the last equality is derived elsewhere [17]. The quantity within the parenthesis in the denominator of (6) can be seen to be a sample estimate of the ensemble average in (7). For sufficiently large N, the error in the sample estimate is expected to be small and we can approximate $\hat{\gamma}$ as

$$\begin{aligned} \hat{\gamma} &\cong N a^4/\sum_{j=1}^{N} [E(\hat{x}_i\hat{x}_{i+j})]^2 \\ &= Na^4/\sum_{j=1}^{N} \{(2/\pi)a^2 \sin^{-1}[R_x(j)/R_x(0)]\}^2 \\ &= N(\pi/2)^2/[\sum_{j=1}^{N} \{\sin^{-1}[R_x(j)/R_x(0)]\}^2]. \end{aligned} \tag{8}$$

One can show that for any $0 \le \beta \le 1$,

$$\beta \le \sin^{-1}\beta \le (\pi/2)\beta \tag{9}$$

and using this inequality in (8), we can bound $\hat{\gamma}$ as

$$\begin{aligned} N[R_x^2(0)/\sum_{j=1}^{N} R_x^2(j)] \le \hat{\gamma} \le N[\pi/2]^2 \\ \cdot [R_x^2(0)/\sum_{j=1}^{N} R_x^2(j)]. \end{aligned} \tag{10}$$

We can calculate the expected PBR for any input signal by substituting its autocorrelation function $R_x(j)$ in (8) and evaluating it. To understand the effects of binarization on PBR, we estimate the PBR γ of unbinarized correlation from (3) using a similar approach as

$$\gamma \cong N[R_x^2(0)/\sum_{j=1}^{N} R_x^2(j)]. \tag{11}$$

Comparing (10) and (11), we note that PBR $\hat{\gamma}$ for the binary correlation is greater than the PBR γ for the unbinarized correlation. We also can see that this improvement is at the most a factor of $(\pi/2)^2$ or 3.8 dB. The analysis of the binary correlator has so far assumed no noise at the input. To analyze the effect of noise, we replace the signal $\{x_i\}$ by $\{x_i\} + \{n_i\}$, where $\{n_i\}$ is a sample realization from a zero mean, Gaussian random process with an autocorrelation function $R_n(j)$. $\{x_i\}$ and $\{n_i\}$ are also assumed to be jointly Gaussian and uncorrelated.

Proceeding as before, the correlation peak value for the noisy case is given by

$$\begin{aligned} \hat{C}(0) &= (1/N)\sum_{i=1}^{N} \hat{x}_i\widehat{(x_i + n_i)} \\ &\cong (2/\pi)a^2 \sin^{-1}\{[R_x(0)]/ \\ &\quad \sqrt{R_x(0)[R_x(0) + R_n(0)]}\} \\ &= (2/\pi)a^2 \sin^{-1}\sqrt{R_x(0)/[R_x(0) + R_n(0)]}. \end{aligned} \tag{12}$$

Comparing (12) with (5), we note that the correlation peak value decreases as the amount of input noise $R_n(0)$ increases. The PBR $\hat{\gamma}$ in the presence of the noise can be derived as

$$\begin{aligned} \hat{\gamma} &= [(1/N)\sum_{i=1}^{N} \hat{x}_i\widehat{(x_i + n_i)}]^2 \\ &\quad \cdot \{(1/N)\sum_{j=1}^{N} [(1/N)\sum_{i=1}^{N} \hat{x}_i\widehat{(x_{i+j} + n_{i+j})}]^2\}^{-1} \\ &= [(2/\pi)a^2 \sin^{-1}\sqrt{R_x(0)/\{R_x(0) + R_n(0)\}}]^2 \\ &\quad \cdot \{(1/N)(2a^2/\pi)^2 \\ &\quad \cdot \sum_{j=1}^{N} [\sin^{-1}\sqrt{R_x^2(j)/\{R_x^2(0) + R_x(0)R_n(0)\}}]^2\}^{-1} \\ &= N[\sin^{-1}\sqrt{R_x(0)/\{R_x(0) + R_n(0)\}}]^2 \\ &\quad \cdot \{\sum_{j=1}^{N} [\sin^{-1}\sqrt{R_x^2(j)/\{R_x^2(0) + R_x(0)R_n(0)\}}]^2\}^{-1}. \end{aligned} \tag{13}$$

Using the inequality of (9) in (13), we can show that

$$(2/\pi)^2\,\gamma \le \hat{\gamma} \le (\pi/2)^2\,\gamma \tag{14}$$

where γ is the PBR of unbinarized correlation as in (11).

The inequality in (14) shows that the ratio $(\hat{\gamma}/\gamma)$ varies from -3.8 dB to $+3.8$ dB; thus depending on the input noise level, we will find situations where the binary correlator will outperform the unbinarized correlator. As we have shown in (8), $\hat{\gamma}$ is closer to the upperbound for the case of low noise. It has been shown elsewhere [11] using peak to sidelobe ratio (PSR) that binarization results in a loss of 3.8 dB in the case of very noisy inputs. Using an exponential model for $R_s(j)$, Kumar and Casasent [11] examine the PSR for binarized correlation as a function of signal bandwidth.

In this section, we have shown that binary correlators perform at least as well as unbinarized correlators. But their main advantage is in the ease of implementation which is considered in Section IV. Before we examine the simulation results, a few words of caution are in order. We considered the performance only from the viewpoint of PBR, but as the input noise level increases, there is a greater probability that the correlation has a maximum at $j \neq 0$. The effect of input binarization on this false peak probability is of importance in location estimation problems. Correlation is often used to separate two classes of targets. Binarization effects on class discrimination should also be considered. When we use the definition in (3) for PBR, we are implicitly assuming that the correlation has a peak at $j = 0$. This is true only for low noise situations and thus our conclusions should not be used for high noise input scenarios.

III. SIMULATION RESULTS

In this section, we support the conclusions of Section II with the help of a very limited image data set shown in Fig. 1. These are all digitized images of the same area

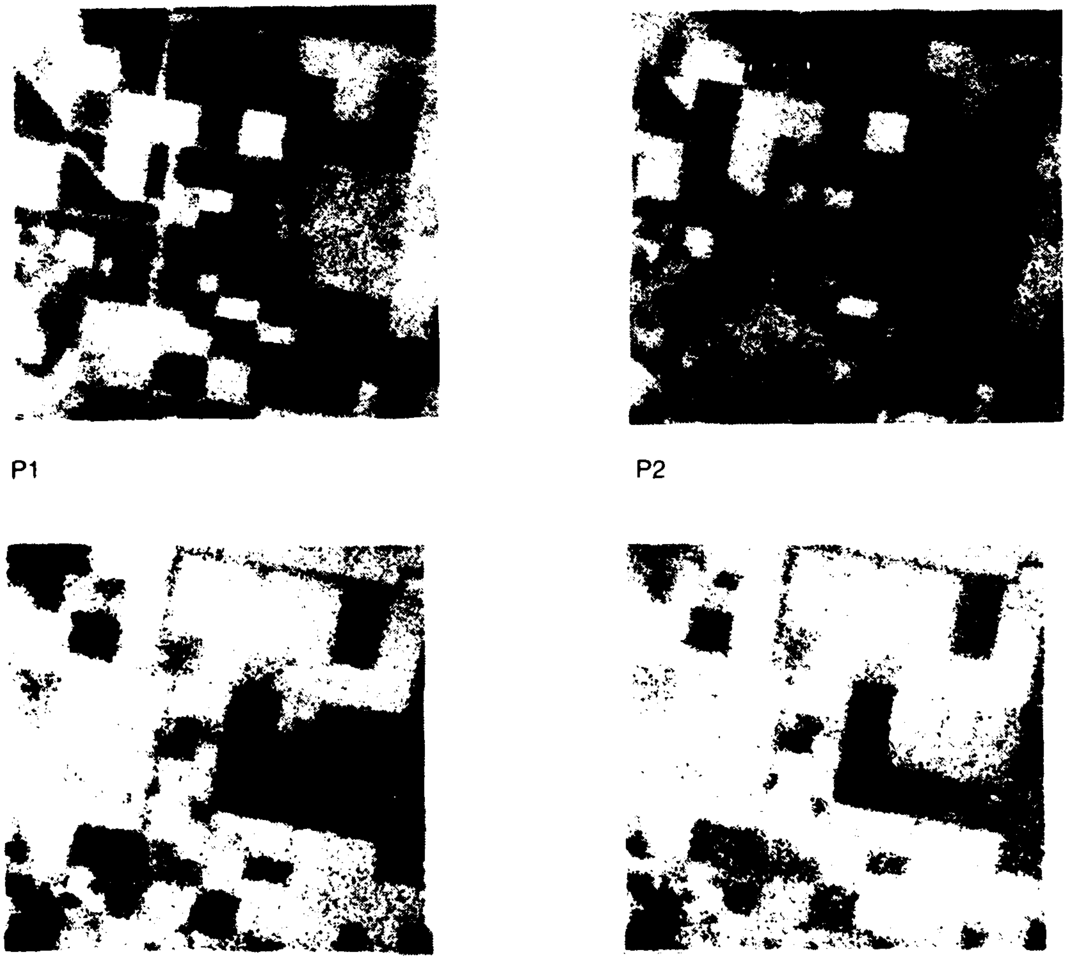


Fig. 1. Multispectral pictures of south of Fresno, California. Spectral bands of each picture are as follows: P_1 0.8-1.1 μ, P_2 0.7-0.8 μ, P_3 0.6-0.7 μ, P_4 0.5-0.6 μ.

0018-9251/83/0300-0325 $00.75

south of Fresno, California, but taken with different sensors. The spectral bands in which these images were taken are indicated in Fig. 1. These images P_1, P_2, P_3, and P_4 of size 128 × 128 represent the same scene and thus all cross correlations (as well as autocorrelations) should peak at $j = 0$ in the correlation plane. As was discussed in Section II, the mean of the image carries no relevant information (especially for multispectral data) and is thus removed *before any processing. For our* multispectral data, the gain factor of each multispectral sensor is not only different, but also unreliable, and as a result, we had to rescale the gray levels in all four images so that the energies in all of them are the same. Such preprocessing is necessary to use the model in (1).

After the above normalization on the four images, cross correlations were performed among them. The resulting correlation peaks and PBRs are listed in Table I. From Fig. 1, we note that P_1 and P_2 are similar and P_3 and P_4 are similar. This behavior is indicated by larger cross-correlation peaks and PBRs for $P_1 * P_2$ than for $P_1 * P_3$ and $P_1 * P_4$. We use $*$ to denote correlation operation.

TABLE I
Performance of Unbinarized Correlation

	P_1	P_2	P_3	P_4
A. Normalized Intensities at Cross-Correlation Origins				
P_1	1.0000	0.6175	0.0622	0.0609
P_2	0.6175	1.0000	0.0023	0.0026
P_3	0.0622	0.0023	1.0000	0.8357
P_4	0.0609	0.0026	0.8357	1.0000
B. Peak to Background Ratios (PBRs)				
P_1	63.163	53.286	8.2671	9.9870
P_2	53.286	93.628	11.060	12.533
P_3	8.2671	11.060	44.912	51.848
P_4	9.9870	12.533	51.848	76.739

*The four images P_1 to P_4 are now binarized to pro*duce the hard-limited images S_1 to S_4 as shown in Fig. 2. The median in each image was chosen as the threshold

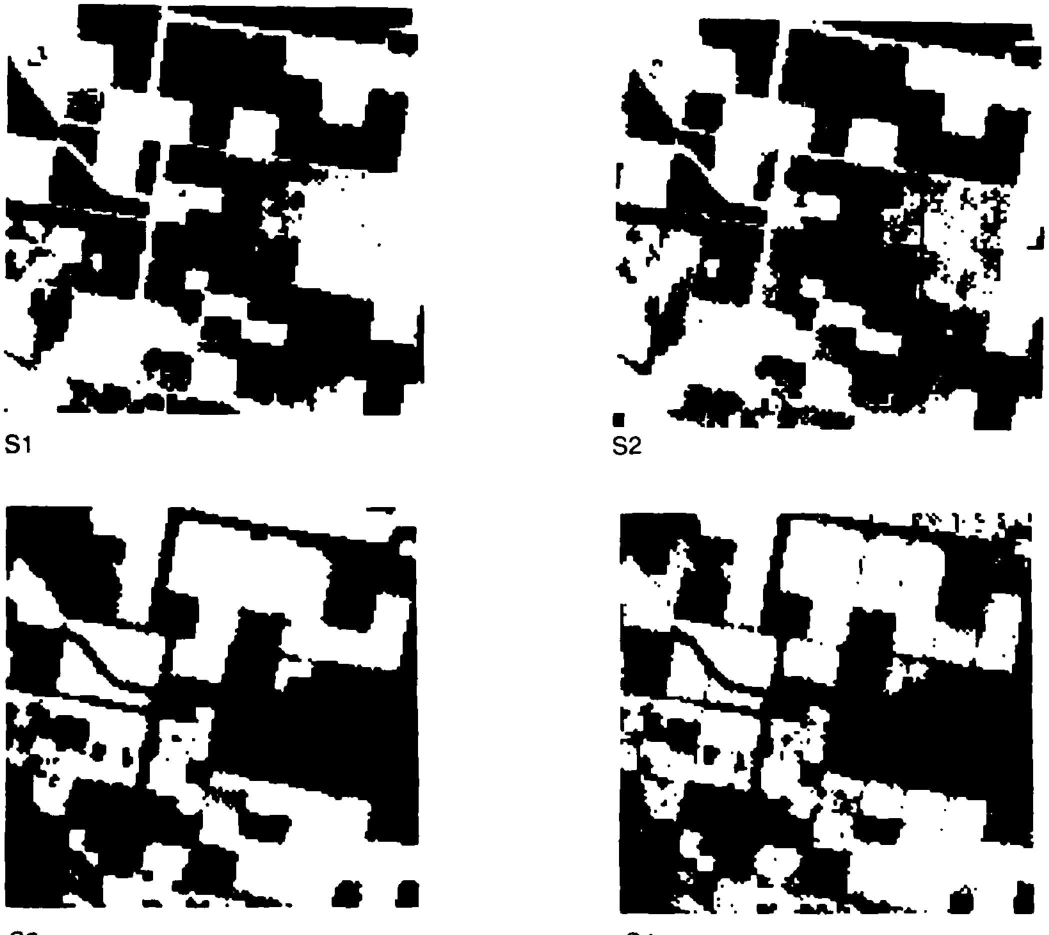


Fig. 2. Binarized versions of the multispectral pictures shown in Fig. 1.

rather than the mean to satisfy our assumption that the average value of $\{\hat{x}_i\}$ is zero. Even though the gray levels as shown in Fig. 2 represent completely dark (0) and completely white (+1) areas, we treat them as denoting −1 and +1 in our actual processing. Cross correlations are performed on these binary images and the resulting parameters are shown in Table II. Once again S_1 correlates better with S_2 than with S_3 or S_4. Comparing the diagonal elements of Table IB and Table IIB, we see that unbinarized autocorrelations yield PBRs of 63.2, 93.6, 44.9, and 76.7, while binarized autocorrelations yield PBRs of 100.3, 117.5, 139.6, and 154.0, respectively. Thus we see that for all four images, binarization improves the correlation PBR by factors of 1.25 to 2.0. These improvement factors are well within the range of 1.0 *to* $(\pi/2)^2 = 2.5$ *predicted by* (10). The PBRs in all cases are estimated using the definition in (3) on the observed correlation output. The PBRs for binarized cross correlations are consistently higher than the PBRs for unbinarized correlations. Improvement factors of almost 5.0 (beyond the 2.5 suggested by (10)) are observed in the cross correlations of S_1 and S_4 and P_1 and P_4. This is because P_1 and P_4 differ drastically (through random contrast reversals) and the differences cannot be modeled simply as additive noise. Binarizing removes the unreliable amplitude information while retaining only the major edge information and thus results in better PBR [18].

TABLE II
Performance of Binarized Correlation

	S_1	S_2	S_3	S_4
A. Normalized Intensities at Cross-Correlation Origins				
S_1	1.0000	0.5806	0.2171	0.2101
S_2	0.5806	1.0000	0.2157	0.1911
S_3	0.2171	0.2157	1.0000	0.6733
S_4	0.2101	0.1911	0.6733	1.0000
B. Peak-to-Background Ratios (PBRs)				
S_1	100.27	64.595	44.880	48.693
S_2	64.595	117.48	49.385	49.507
S_3	44.879	49.385	139.56	100.51
S_4	48.693	49.507	100.51	154.00

We now observe the effects of additive noise in the input on the observed autocorrelations. In Table III, we show the peaks and PBRs obtained as different amounts of additive white noise is introduced in the autocorrelation $P_1 * P_1$. The ratio *N/S* indicates the white noise power to the image (signal) power. The peak values and PBRs obtained in the binarized autocorrelation $S_1 * S_1$ for various N/S values are shown in Table IV. For the reasons discussed in Section II, only small amounts of noise are considered here. From Table III, we note that PBR of $P_1 * P_1$ is relatively unaffected by small amounts of noise. On the other hand, the PBR of binarized correlation decreases as noise increases; but for the range of noise levels shown, binarized correlation still gives better PBR compared with unbinarized correlation. Binarized correlation peak is also seen to be more sensitive to noise in the input signal. Thus we see that binary correlator still outperforms unbinarized correlation in the presence of small amounts of noise.

TABLE III
Effect of Input White Noise on Unbinarized Correlations

N/S	Peak	PBR
0.00	1.0000	63.163
0.02	0.9805	63.052
0.04	0.9621	62.548
0.06	0.9439	62.759
0.08	0.9267	63.265
0.10	0.9095	62.149
0.12	0.8957	62.505
0.14	0.8809	61.826
0.16	0.8644	61.450
0.18	0.8539	62.333

Notes: Input is the intensity normalized $\{P_1 + \text{noise}\}$. Reference is the intensity normalized $\{P_1\}$.

TABLE IV
Effect of Input White Noise on Binarized Correlation

N/S	Peak	PBR
0.00	1.0000	100.27
0.02	0.9129	93.301
0.04	0.8707	89.097
0.06	0.8402	86.798
0.08	0.8175	86.640
0.10	0.7848	83.453
0.12	0.7680	83.297
0.14	0.7472	81.270
0.16	0.7233	80.562
0.18	0.7308	81.689

Notes: Input is the binarized $\{P_1 + \text{noise}\}$. Reference is the binarized $\{P_1\}$.

IV. DISCUSSION

In this paper we have shown that binary correlation gives better peak-to-background ratios even in the presence of small amounts of noise. We supported these theoretical arguments with the help of simulation on a set of multispectral images.

The binary correlator is attractive mainly from implementational considerations. By using the VLSI technology [14], the binary correlator can be implemented in a reasonable number of chips. The $+a$ and $-a$ levels of the binarized signal in (4) correspond to logical 1 and 0 in a digital implementation. The multiplication laws are then as given in Table V. We observe from this table that

TABLE V
Multiplication Laws for Binary Inputs

X	Y	$X\,Y$	$L[X]$	$L[Y]$	$L[X\,Y]$
a	a	a^2	1	1	1
a	a	$-a^2$	1	0	0
a	a	$-a^2$	0	1	0
$-a$	$-a$	a^2	0	0	1

Note: $L[\;]$ refers to the corresponding logic level.

0018-9251/83/0300-0327 $00.75 © 1983 IEEE

a binary multiplication can be easily implemented by a simple exclusive-OR circuit. In order to obtain the binary correlation $\hat{C}(j)$, we must add $(\hat{x}_i\hat{y}_{i+j})$ for all values of i for every j. For a sequence of length N, let M_j represent the number of the times the product $(\hat{x}_i\hat{y}_{i+j})$ equals $+a^2$ (or +1 in digital circuits). Then $\hat{C}(j)$ can be written as

$$\begin{aligned}\hat{C}(j) &= a^2M_j - a^2(N - M_j)\\ &= a^2(2M_j - N). \qquad (15)\end{aligned}$$

Thus $\hat{C}(j)$ can be easily obtained by just counting the number of times the product $(\hat{x}_i\hat{y}_{i+j})$ equals +1 among all possible i for a fixed j. This can be implemented as in Fig. 3. In Fig. 3, the reference sequence $\{\hat{x}_i\}$ is stored in the top array of shift registers and then by serially shifting $\{\hat{y}_i\}$ through the second array, the serial output $\{M_j\}$ is obtained. $\hat{C}(j)$ can be obtained from M_j using (15) in a postprocessor.

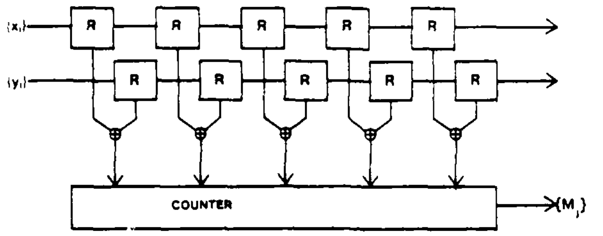


Fig. 3. Digital binary correlator circuit. R is a 1-bit shift register, + is an exclusive-OR circuit, and M_j is the counter input.

HIROYASU MURAKAMI
Research and Development Section
Yanagicho Works, Toshiba Corporation
70-Yanagicho, Kawasaki, Japan

B.V.K. VIJAYA KUMAR
Electrical Engineering Department
Carnegie-Mellon University
Pittsburgh, PA 15213

REFERENCES

[1] Whalen, A.D. (1971)
Detection of Signals in Noise
New York: Academic Press, 1971.

[2] Van Der Lugt, A. (1964)
Signal detection by complex spatial filtering.
IEEE Transactions on Information Theory, 1964, *IT-10*, 139-145.

[3] Pratt, W.K. (1974)
Correlation techniques of image registration.
IEEE Transactions on Aerospace and Electronic Systems, 1974, *AES-10*, 353-358.

[4] Mostafavi, H., and Smith, F.W. (1978)
Image correlation with geometric distortion. Part I: Acquisition performance.
IEEE Transactions on Aerospace and Electronic Systems, 1978, *AES-14*, 487-493.

[5] Mostafavi, H. (1979)
Optimal window functions for image correlations in the presence of geometric distortions.
IEEE Transactions on Acoustics, Speech and Signal Processing, 1979, *ASSP-27*, 163-169.

[6] Vijaya Kumar, B.V.K., and Casasent, D. (1980)
Space-blur bandwidth product in correlator performance evaluation.
Journal of the Optical Society of America, 1980, *70*, 103-110.

[7] Vijaya Kumar, B.V.K., and Casasent, D. (1980)
Nonlinear *t-E* curve effects in an optical correlator.
Optics Communications, 1980, *34*, 4-6.

[8] Munteanu, C. (1981)
Evaluation of the sequential similarity detection algorithm applied to binary images.
Pattern Recognition, 1981, *13*, 167-175.

[9] Barnea, D.I., and Silverman, H.F. (1972)
A class of algorithms for digital registration.
IEEE Transactions on Computers, 1972, *C-21*, 179-186.

[10] Svedlow, M., McGillem, C.D., and Anuta, P.E. (1978)
Image registration: Similarity measure and preprocessing.
IEEE Transactions on Aerospace and Electronic Systems, 1978, *AES-14*, 141-152.

[11] Vijaya Kumar, B.V.K., and Casasent, D. (1981)
Binarization effects in a correlator with noisy input.
Applied Optics, 1981, *20*, 1433-1437.

[12] Boland, J.S., Pinson, L.J., Peters, E.G., Kane, G.R., and Malcom, W.W. (1979)
Design of a correlator for real-time video comparison.
IEEE Transactions on Aerospace and Electronic Systems, 1979, *AES-15*, 11-19.

[13] Cole, T.W. (1980)
Incoherent optical 1-bit cross-correlators for radio antenna arrays.
Applied Optics, 1980, *19*, 2169-2173.

[14] Mead, C., and Conway, L. (1980)
Introduction to VLSI Systems
Reading, Mass.: Addison-Wesley, 1980.

[15] Oppenheim, A.V., and Schafer, R.W. (1975)
Digital Signal Processing.
Englewood Cliffs, N.J.: Prentice-Hall, 1975.

[16] Casasent, D., and Furman, A. (1977)
Sources of correlation degradation.
Applied Optics, 1977, *16*, 1652-1661.

[17] Papoulis, A. (1965)
Probability, Random Variables and Stochastic Processes
New York: McGraw-Hill, 1965.

[18] Barniv, Y. (1981)
Multi-Sensor Image Registration.
Ph.D. dissertation, Carnegie-Mellon University, Pittsburgh, Pa., 1981.

0018-9251/83/0300-0328 $00.75 © 1983 IEEE

12. GENERALIZED CHORD TRANSFORMATION FOR DISTORTION-INVARIANT OPTICAL PATTERN RECOGNITION

Reprinted from Applied Optics, Vol. 22, page 2087, July 15, 1983

Generalized chord transformation for distortion-invariant optical pattern recognition

David Casasent and Wen-Thong Chang

An optical processor that realizes a generalized chord transformation is described. The wedge-ring detector samples of an autocorrelation are shown to be the histograms of the chord distributions. This dimensionality reduced set of features is used as the feature vector inputs for a Fisher linear classifier to determine the class of the input object independent of geometrical distortions. Initial discussions on the use of different classifiers, the polarity of the classifier's output, and selection of the image training set are also advanced.

I. Introduction

Obtaining object classification in the face of geometrical distortions in the input object (due to scale, rotation, aspect variations, etc.) is a major pattern recognition problem that has received extensive attention. Various optical systems of increasing flexibility and potential have been suggested as partial solutions to this problem.[1] These include weighted matched spatial filters,[2] space-variant optical pattern recognition systems,[3] optical correlators using generalized matched spatial filters,[4] and synthetic discriminant functions.[5] Conventional digital techniques for distortion-invariant pattern recognition involve extracting features from regions of the input image, determination of a linear discriminant function or linear combination of these features (from image training sets of the different object classes), and using the measured features for an input object in a classifier to determine the class of the input object. A recent optical system[6] follows this approach by extracting the moments of the input object optically and then using these as the features to be fed to a digital nonlinear least-mean-square estimator to perform object classification.

In this paper, we describe a new optical architecture that realizes a new operation, a generalized chord transformation (Sec. II). This constitutes our observation space. We utilize a wedge-ring detector[7] to extract features from this output for reduced dimensionality and for distortion-invariant pattern recognition (Sec. III). These measured features are then fed to a Fisher classifier[8] (Sec. IV) from which the class of the input object is determined even if geometrical distortions are present in the input object. Initial simulation results (Sec. V) are most impressive. Extensions to multiclass problems and the use of alternate classifiers are then advanced together with new techniques to select the training set image data (Sec. VI).

II. Optical Realization of a Generalized Chord Transformation

The chord transformation[9] is defined for a binary solid object described only by its boundary (i.e., Fig. 1). For each pair of points on the boundary, we construct a chord and note its length r and angle θ. We then describe the object (boundary) by the distribution $h(r,\theta)$ of all chords. This conversion of a binary solid object $f(x,y)$ into $h(r,\theta)$ is a chord transformation.

To develop a generalized chord transformation and its optical realization, we first consider a simplified technique (suitable for optical realization) to produce the chord transformation. We consider an image and denote points in the image by (x,y) and the boundary of the image by those points $b(x,y) = 1$. A chord then exists between two points if

$$g(x,y,r,\theta) = b(x,y)b(x + r\cos\theta,y + r\sin\theta) = 1, \tag{1}$$

i.e., if the two points are on the boundary, $g = 1$, and a chord defined by r and θ exists. The distribution $h(r,\theta)$ of chords in the image can then be described from

$$\begin{aligned} h'(r,\theta) &= \iint g(x,y,r,\theta)dxdy \\ &= \iint b(x,y)b(x + r\cos\theta,y + r\sin\theta)dxdy \end{aligned} \tag{2}$$

Substituting $(\xi,\eta) = (r\cos\theta,r\sin\theta)$, Eq. (2) becomes

$$\begin{aligned} h(\xi,\eta) &= \iint b(x,y)b(x + \xi,y + \eta)dxdy \\ &= b(x,y) \star b(x,y), \end{aligned} \tag{3}$$

The authors are with Carnegie-Mellon University, Department of Electrical Engineering, Pittsburgh, Pennsylvania 15213.

Received 19 February 1983.
0003-6935/83/142087-08$01.00/0

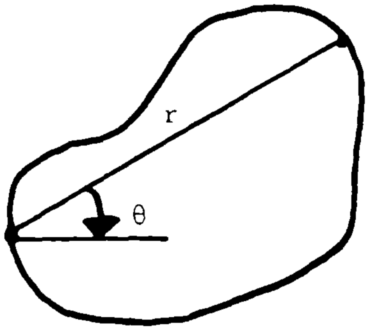


Fig. 1. Chord transformation.

which we recognize as the autocorrelation of the boundary $b(x,y)$.

The calculation of all (r,θ) values for all boundary points is very time-consuming. However, from Eq. (3), we see that the $h(\xi,\eta)$ distribution can also be obtained by simply autocorrelating the boundary of the object. (This fact was also noted in Ref. 10 and elsewhere.) Since optical processors easily realize the correlation operation, an optical realization of the chord transform appears quite attractive. However, optical systems can realize the autocorrelation of the full object (not just its boundary) with no increase in system complexity. We denote the autocorrelation of the full image $f(x,y)$ rather than just its boundary $b(x,y)$ by

$$h_G(\xi,\eta) = \iint f(x,y)f(x+\xi,y+\eta)dxdy = f \circledast f. \tag{4}$$

Since this autocorrelation is not restricted to the boundary of the object nor is it restricted to binary objects, we denote the result as a generalized chord transformation.

One can optically realize the autocorrelation of an image by several techniques. Nichols[11] used a two-stage Fourier transform in which the autocorrelation was obtained from the Fourier transform of the power spectrum. If two 2-D spatial light modulators are used, the autocorrelation can be realized entirely optically in parallel on one system. A joint transform correlator[12] using mirrors or point holographic elements to replicate the input pattern is yet another possibility. With a nonlinear crystal positioned in the joint Fourier transform plane, these systems can be further simplified. Thus the optical realization of the autocorrelation of an input image (as required to produce a chord transformation) is easily achieved by many techniques.

III. Feature Space

Much research[10,11,13,14] has been performed on different manipulations and displays of the chord transformation output distributions to achieve distortion-invariant pattern recognition. In Ref. 10, the distributions of the lengths and angles of the chords of an object were considered as features. However, only the peak value of the distribution was used, and the classification of only several simple geometrical shapes was considered. In Ref. 11, the length and angle distribution of the chords of different characters were digitally computed from the optical autocorrelation, and differences in both distributions were noted to exist for different letters of the alphabet. However, no classification algorithm was suggested. In Ref. 13, the sum and difference of the angles between the chords and the boundary of the object were calculated. The authors refer to the distribution of these angular sums and differences as a 3-D histogram. In Ref. 13, such a display was considered for ship classification. In Ref. 14, the distribution of the lengths of the chords was digitally computed for various noisy circular objects and map outlines.

We found much potential shape information and pattern recognition potential in the chord transformation. However, the computational complexity required in prior techniques, the use of only boundary objects in prior work, and the lack of attention to the classification algorithm to use motivated our present work. We first consider a new technique to extract the $h(r,\theta)$ radial and angular chord distributions from an optically generated autocorrelation of the input image. As noted earlier, our $h_G(r,\theta)$ will be the distribution of a generalized chord transformation. Hereafter, we do not specifically distinguish between h and h_G.

As a preferable approach, we consider a simple and direct technique for feature extraction from the generalized chord transformation observation space with attention to dimensionality reduction and obtaining distortion-invariant features. Dimensionality reduction is essential, since otherwise $h(\xi,\eta)$ can contain 10^6 output samples (for a $10^3 \times 10^3$-pixel image), and the corresponding object would have to be modeled as a vector $\mathbf{f}$ with dimensionality $D = 10^6$. Processing such vector data would be computationally intense. An attractive, easily realized, and intuitively pleasing choice for a feature space results if the $h(\xi,\eta)$ optical autocorrelation output plane pattern is sampled with a wedge-ring detector. This detector[7] consists of 32 wedge-shaped detector elements in one-half of a 2.54-cm (1-in.) circular detector and 32 annular or ring-shaped detector elements in the other half of the circular detector area. All 64 outputs are available in parallel on 64 separate leads. Figure 2 shows the general schematic of our hybrid processor. We discuss the features produced below, and in Sec. IV we discuss the classifier.

Denoting our generalized chord transformation observation space by $h(\xi,\eta) = h(r\cos\theta, r\sin\theta)$, where $\xi = r\cos\theta$ and $\eta = r\sin\theta$, the wedge-sampled outputs are

$$h(\theta) = \int h(r\cos\theta, r\sin\theta)rdr, \tag{5a}$$

and the ring-sampled detector outputs are

$$h(r) = \int h(r\cos\theta, r\sin\theta)rd\theta. \tag{5b}$$

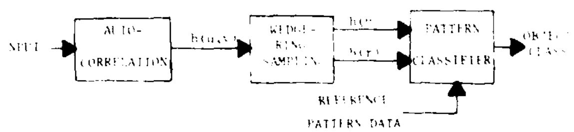


Fig. 2. Simplified block diagram of a hybrid optical/digital distortion invariant pattern recognition system using a generalized chord transformation.

Inspection of Eq. (5a) shows that the $h(\theta)$ features are independent of r (or chord length) and are thus also effectively invariant to scale changes in the input image. Similarly, from Eq. (5b), we see that the $h(r)$ output features are independent of θ (or the orientation of the chord distribution) and are thus also effectively invariant to rotational differences in the input image.

Thus, as our feature space, we use the 64 wedge-ring detector-sampled outputs of the generalized chord observation space. This constitutes a 64-sample representation of the $h(r,\theta)$ chord distribution. Such a representation of an image $f(x,y)$ is equivalent (in conventional pattern recognition terminology[8]) to the representing of an image as a vector $\mathbf{f}$ of dimensionality $D = 64$, with the elements of $\mathbf{f}$ being the 64 $h(\theta)$ and $h(r)$ wedge-ring detector-sampled values. This new optical architecture thus produces the desired $h(r,\theta)$ chord distribution in parallel with a greatly reduced computational load (compared to a digital version of the equivalent system).

IV. Classifier

As the second major issue in this paper, we consider the classifier to be used with such a feature space. In conventional pattern recognition and object classification techniques, the features (the elements of the vector $\mathbf{f}$) are fed to a classifier in which the class of the input object is determined. This is achieved by projecting the measured vector $\mathbf{f}$ onto a linear discriminant function vector $\mathbf{w}$ and comparing the scalar output obtained to a threshold. In our general architecture of Fig. 2, we use a parallel optical system to produce our observation space (the autocorrelation) of the input object), a wedge-ring detector to provide the dimensionality reduced feature space, and a digital postprocessor for classification. This is a new quite general and very attractive concept and architecture for advanced pattern recognition applications. It employs the parallelism of the optical system for feature extraction and the flexibility and numerical computation features of a digital processor for classification.

We now discuss the classifier we used. We consider the use of a linear discriminant function (LDF) that maximizes the Fisher ratio to determine the linear combination of output features to use for object classification. Our analysis and formulation follow that in Refs. 8, 15, and others. Caulfield *et al.*[15] also used the Fisher criteria, but they used Fourier transform coefficients as the image features and used the Fisher LDF to synthesize a matched spatial filter. Leger *et al.*[16,17] used Fukunaga-Koontz techniques[16] and least-mean-square algorithms[17] to synthesize a set of optical filter functions. Our approach differs significantly in concept and implementation, since we use the optical system to produce the features and a digital postprocessor to perform the classification. In approaches such as those in Refs. 16 and 17, the location of the output correlation peak indicates the class of the input object rather than the location of the object (as in a conventional correlator). In such systems, a new filter function must be produced for each new class of object to be recognized or for each new type of classifier used. Our system is more general and flexible; since its outputs can directly be used in any classifier for any object class, we choose to use the Fisher criteria because of their intuitive appeal.

In many classifiers, image training sets (images of each class of object with, for example, different geometrical distortions) are used to determine the LDF. We consider a two-class pattern recognition problem with the image training set being $\{\mathbf{f}_{1i}\}$ and $\{\mathbf{f}_{2i}\}$ for classes 1 and 2, respectively. We denote the number of training set images in each class by N_1 and N_2, respectively. For each image, we calculate its feature vectors $\mathbf{f}_{1i}$ and $\mathbf{f}_{2i}$. We also calculate the vector means $\mathbf{m}_1$ and $\mathbf{m}_2$ for the two classes of training set data, where for class 1

$$\mathbf{m}_1 = (1/N_1) \sum_i \mathbf{f}_{1i} \tag{6}$$

We also calculate the scatter matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ for each class, where for class 1

$$\mathbf{S}_1 = \sum_i (\mathbf{f}_{1i} - \mathbf{m}_1)(\mathbf{f}_{1i} - \mathbf{m}_1)^T. \tag{7}$$

In Eqs. (6) and (7), the summation is over the $i = N_1$ or N_2 images in the corresponding training set.

The purpose of a linear classifier is to determine (from the training set data) an LDF or vector $\mathbf{w}$ that is a linear combination of the observed features. After this has been done, the system is tested. In such a case, a linear combination of the observed features for an unknown input image $\mathbf{f}$ is calculated and compared to a threshold level T. This is equivalent to calculating the vector inner product $\mathbf{w}^T\mathbf{f}$ or $\mathbf{f}^T\mathbf{w}$ or the projection of $\mathbf{f}$ onto $\mathbf{w}$ (all vectors are column vectors, the superscript T denotes the transpose or a row vector) and comparison of $\mathbf{w}^T\mathbf{f}$ to the threshold T. The LDF vector $\mathbf{w}$ and the threshold T are selected from the training set data (as we detail later). If $\mathbf{w}^T\mathbf{f} \gtrless T$, we classify the unknown input object as a member of class one or two, respectively.

To determine $\mathbf{w}$, we must first select the performance measure to be optimized. In a Fisher classifier, the parameter maximized is the Fisher ratio

$$J = \frac{|m_1 - m_2|^2}{S_1^2 + S_2^2}. \tag{8}$$

In Eq. (8), m_1 and m_2 are the means of the projections of the training set data in classes one and two on $\mathbf{w}$, i.e., for class 1

$$m_1 = (1/N_1) \sum_i \mathbf{w}^T\mathbf{f}_{1i} = \mathbf{w}^T\mathbf{m}_1, \tag{9}$$

and S_1 and S_2 are the scatter of the projections for the training set data in classes one and two, respectively; i.e., for class 1

$$\begin{aligned} S_1^2 &= \sum (\text{Projections}_{1i} - m_1)^2 \\ &= \sum (\mathbf{w}^T\mathbf{f}_{1i} - \mathbf{w}^T\mathbf{m}_1)^2 = \mathbf{w}^T\mathbf{S}_1\mathbf{w} \end{aligned} \tag{10}$$

The Fisher ratio J in Eq. (8) is an intuitively pleasing performance parameter. In a hyperspace data representation, each input image is a vector described by a given position in this hyperspace. The numerator in

Eq. (8) is the difference in the means of the projections for the two classes of training data, and the denominator is the sum of the scatter of the projections. Thus the numerator in J is a measure of the separation of the classes, and the denominator is a measure of the clustering within each class. The reader should recall that the parameters in Eqs. (6)–(10) refer to the training set data for the two classes.

To determine the LDF **w** that maximizes J in Eq. (8), we must describe J in terms of **w**. Using Eqs. (6) and (9), we can write the numerator in Eq. (8) as

$$(m_1 - m_2)^2 = (\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2)^2 = \mathbf{w}^T\mathrm{S}_B\mathbf{w}, \tag{11}$$

where the between-class scatter matrix S_B for the two-class discrimination problem is the vector outer product:

$$\mathrm{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T. \tag{12}$$

We describe the denominator in Eq. (8) using Eqs. (7) and (10) for the two-class problem as

$$S_1^2 + S_2^2 = \mathbf{w}^T\mathrm{S}_1\mathbf{w} + \mathbf{w}^T\mathrm{S}_2\mathbf{w} = \mathbf{w}^T\mathrm{S}_W\mathbf{w}, \tag{13}$$

where the within-class scatter matrix S_W is

$$\mathrm{S}_W = \mathrm{S}_1 + \mathrm{S}_2. \tag{14}$$

Substituting Eqs. (11) and (13) into Eq. (8), we find that the Fisher ratio can be described in terms of the LDF **w** as

$$J(\mathbf{w}) = \mathbf{w}^T\mathrm{S}_B\mathbf{w}/\mathbf{w}^T\mathrm{S}_W\mathbf{w}. \tag{15}$$

Differentiating Eq. (15) with respect to $\boldsymbol{\omega}$ and setting the result equal to zero, we find that **w** is the solution *of the general eigenvalue problem*

$$\mathrm{S}_B\mathbf{w} = \lambda\mathrm{S}_W\mathbf{w}. \tag{16a}$$

If S_W is nonsingular, we can write Eq. (16a) as

$$\mathrm{S}_W^{-1}\mathrm{S}_B\mathbf{w} = \lambda\mathbf{w}. \tag{16b}$$

The unique solution to Eq. (16b) is our LDF. It is given by

$$\mathbf{w} = \mathrm{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2). \tag{17}$$

Thus to determine the LDF in Eq. (17) we must calculate the vector means $\mathbf{m}_1$ and $\mathbf{m}_2$ plus the within-class scatter matrix S_W for our training set data. Substituting these into Eq. (17) then yields the optimal LDF **w** that maximizes the Fisher ratio $J(\mathbf{w})$ in Eq. (8). As our threshold, we use the average of the means of the projections for our training set data, i.e.,

$$T = (m_1 + m_2)/2. \tag{18}$$

V. Initial Experimental Verification

A. Available Data Base

To demonstrate the use of this Fisher-discriminant classifier with a wedge-ring detector-sampled generalized chord transformation observation space, we used an available IR ship image data base. This data base contained two classes of ships with 36 images in each class (one image every 10° with a 90° depression angle). Gray-scale and binary images were used in all experiments, and the hull of the ship was removed (retaining only the superstructure). The number of features (wedge-ring detector samples) or the dimensionality D of the feature space used affects the number of training set images $N_1 + N_2 = N$ that are necessary. The cases to consider are $D > N$, $D = N$, and $D < N$. If $D > N$, many solutions are possible, and we simply pick one. In $D > N$, we find that S_W *is singular* (its rank is less than D), and thus an LDF **w** to maximize the Fisher ratio does not exist (since S_W^{-1} does not exist), and instead many solutions exist. If $D = N$, a unique solution exists and is easily obtained by solving a matrix–vector problem. If $D < N$, no direct solution exists, and thus we approximate one by least-mean-square techniques. Since maximizing the Fisher ratio is known[8] to be equivalent to a least-mean-square solution, use of such an approach automatically requires $N > D$. We thus find the soution **w** that minimizes the least-square error.

B. Dimensionality Reduction

The consequence of the $N > D$ requirement (which is the practical case of concern in pattern recognition) is that if our available image data base is of finite size (this is obviously the practical situation), we must thus reduce D before a Fisher or similar classifier can be designed and before significant test data are possible. To quantify these remarks, we consider our present problem and available data. If we retain all $D = 64$ wedge-ring detector features, we require $N \geq 65$ training set images. For a two-class problem, we only have available 36 images per class or 72 images total. Thus, if all 64 wedge-ring detector values are retained as features, we would have to use 65 of our available 72 images for training. This would leave us with only 7 of the 72 images for use as new images during testing. The results of such experiments (training on 65 of 72 images) would not be impressive. Thus we next consider further dimensionality reduction of our feature space.

To perform more impressive tests and to reduce the size of the training set and the amount of calculations required, the number of features used must be reduced. Such dimensionality reduction is commonly employed (for different reasons, i.e., to reduce the computations required). From an analysis of the wedge-ring detector outputs for our full set of 72 images, we found that the outer 8 ring readings were approximately zero. We thus ignored these features in all cases and reduced the maximum number of features to $D = 56$. For a feature vector of dimension 56, we would still require 29 images in each class for training. To further reduce the dimensionality of our feature space, we calculated the means m_{1d} and m_{2d} and the variances σ_{1d}^2 and σ_{2d}^2 of each wedge and ring detector output ($d = 1, \ldots, 56$) for various two-class training sets of data. We then calculated $R = (m_{1d} - m_{2d})/(\sigma_{1d}^2 + \sigma_{2d}^2)$ for each case for all 56 detectors. As our features, we retained the 18 detector outputs with the largest mean difference to variance ratio R for the various training sets. Our simplified technique for reduction of the dimensionality

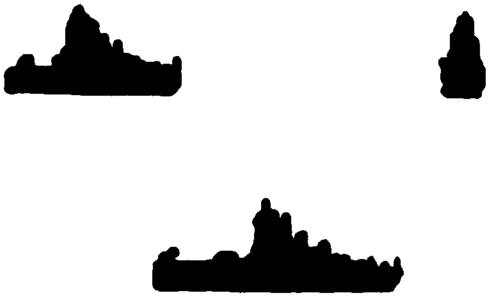


Fig. 3. Representative images for the class 1 ship imagery: (top right) 0, (top left) 45, (bottom) 90°.

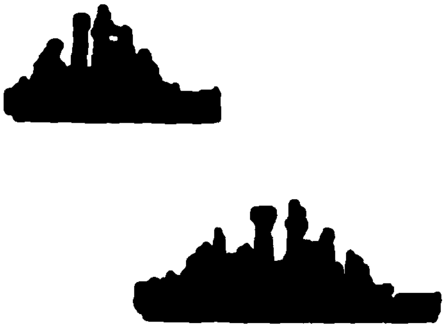


Fig. 4. Representative images for the class 2 ship imagery: (top right) 0, (top left) 45, (bottom 90°.

of the feature space assumes independent features. A more precise technique[18] using the covariance of the features could be employed (to account for dependent relationships between features). However, our results were adequate, and thus such techniques were not employed.

c. Image Training Sets

In Figs. 3 and 4, we show three of the images from the set of 36 images for each of the two classes of ships used. We number the 36 images in each class consecutively from the bow to the stern, with the class one images denoted by 1–36 and the class two images by 37–72. Images 1 and 37 are the bow views, images 9, 27, 46, and 64 are the broadside views, etc. We selected five training sets of data (Table I). The same aspect views were used for each ship class for compatibility, and thus only the ship image numbers for class 1 are listed in Table I. Training set 1 contains all the imagery. Training set 2 contains half of the imagery (images every 20°). Three sets (sets 3–5) of 12 training set images per class were chosen In set 3, the training images were clustered within ±50° of broadside on the front and back of the ship. In set 4, the 12 images were clustered every 10° from 40° to 150° on one side of the ship only, and in set 5 they were evenly spaced every 30°.

Table I. Image Training Sets Used

Image training set number	Number of training set images used per class	Training set images used for class 1 (class 2 analogous)	Training set remarks
1	36	1,2, . . . ,35,36	All images
2	18	1,3,5, . . . ,35	Images every 20°
3	12	4,6,8, . . . ,12,14; 22,24, . . . ,30,32	Images every 20° (within ±50° of broadside)
4	12	4,5, . . . ,14,15	Images every 10° (one side only, near broadside)
5	12	1,4, . . . ,13,16; 19,22, . . . ,31,34	Images every 30°

Table II. Simulated Test Data (Binary Imagery)

Experiment number	Fisher ratio	Number of errors	Probability of correct classification (%)
1	114	0	100
2	126	0	100
3	1180	8	88.9
4	2322	4	94.4
5	158	0	100

Simulation Results

In Table II, we list our experimental results obtained for binary ship image data with the hull of the ship removed. (This was done to remove nonreliable gray-scale image data, and hence the binary version of such imagery was used in our present tests also.) In all cases, the 18 wedge-ring detector features previously noted (Sec. V.B) were used. The five experiment numbers listed correspond to the tests performed using the corresponding five image training sets noted in Table I. In each case we calculated $\mathbf{m}_1$, $\mathbf{m}_2$, and S_W^{-1} for the indicated training set data and from these computed the LDF $\mathbf{w}$ in Eq. (17) and the threshold T in Eq. (18).

From experiment 1, we note that the entire data base was used for training and that no errors occurred. This indicates that the full data base is linearly separable using as few as 18 features. It is thus realistic to assume that some subset (less than all 36 images per class) of the full image data base might provide similar performance to that obtained using the full data base (i.e., 100% probability of correct recognition). One cannot theoretically prove this, however, but test data seem to verify this intuitive remark. In experiment 2, we use 18 images per image class and find that the computed LDF is again capable of correctly classifying all 72 images in both classes (half of which it had not previously seen) with no errors.

Experiments 3–5 consider the use of a much smaller training set (12 images per class or one-third of the available imagery) and the testing of the computed LDF against all 72 images. In experiment 3, 12 images per object class were used (clustered around the broadside views of the target, from both sides of the ship). Good results, 8 errors out of 72 images or 88.9% correct recognition, were obtained. As might be expected, most of the 8 errors occurred for test images close to the bow and stern views. Since such images have fewer pixels on the target and since these images were not included in the test data, these errors are to be expected. In a ship recognition scenario with such imagery, one would not expect to perform classification on such bow and stern views of a ship.

However, to pursue further the issue of training set selection and the performance of this feature-extraction space, we considered other training sets of 12 images per class (from our data base of 36 images per class). In experiment 4, 12 images taken from only one side of the ship were used during training. The motivation for this choice was that the views from both sides of the ship were rather similar, and thus we could possibly contain all information on the ship class by training on images from one side of the ship only. Our results (4 rather than 8 errors out of 72 images, 48 of which were never seen in the training of the classifier) verified this conjecture. As might have been expected, the majority of errors encounteed again occurred for images near the bow and stern. These images were not present in the training set, and thus such errors are again expected. As our third set of twelve training set images per class, we used set 5 (12 images, evenly spaced every 30° about the full ship). Our performance (no errors for all 72 test set images) is the best possible (matching that obtained in experiments 1 and 2 using 36 and 18 training set images, respectively, per class).

In all cases, no more than 8 errors occurred (above 88.9% probability of correct recognition). The third major issue of concern in this paper is further analysis of these results to obtain insight into why such excellent performance results and insight into how selection of the training set used affects the results obtained.

Before leaving our experimental data section, we present (Table III) our ship data test results obtained using gray-scale imagery (for the same experiments and training sets used for binary imagery). These results are quite similar to those obtained using binary imagery (Table II) with a reversal in the results for tests 3 and 4 being the most obvious differences. (We attribute this to additional reliable gray-scale image data present in the superstructure and deck line of the ship.) These experimental results are noteworthy because they represent the generalized chord transform operation that we noted (Sec. II) that our optical system could directly and easily perform.

VI. Discussion

Our optical processor to realize a chord-distribution sampled feature space has proven most attractive as has our wedge-ring detector-sampled chord feature space. Our technique to reduce further the dimensionality of our feature space has likewise been shown to be useful and adequate. Further research is needed to optimize our image training set selection technique. Our three training sets of data (for our 12 image per class training sets) were chosen to investigate and demonstrate certain features useful for training set selection. We will amplify on our general remarks to follow in a future publication when we consider the effects of noise on the performance of our system.

Table III. Simulated Test Data (Gray-Scale Imagery)

Experiment number	Number of errors	Probability of correct classification (%)	Training set remarks
1	0	100	All images
2	1	98.6	Images every 20°
3	3	95.2	Images every 20° (within ±50° of broadside)
4	6	91.6	Images every 10° (one side only, near broadside)
5	0	100	Images every 30°

For our present discussion, we will consider how the Fisher ratio value J (obtained from training set data) can be used to influence our image training set selection criteria.

We first note that maximization of J is a least-square problem, and thus the J parameter (Table II) provides a measure of the accuracy of the least-squares fit solution. Specifically, J is inversely proportional to the least-squares error. Thus a low J value (Table II) indicates a large least-squares error in our solution. However the major point we advance at this junction is that a large J (and hence a small least-mean-square error) does not necessarily reflect that better classification performance will result. Rather, a smaller J for the training set indicates that the training set is more divergent (large mean square error) or, equivalently, that it is more representative of the full image data base.

Thus the philosophy we presently discuss advocates selection of a training set with a lower J value. This premise is based upon the fact (confirmed by our initial experiments) that such an image training set provides a better statistical representation of the test data. As we noted earlier, such measures for training set selection are needed, both to provide more meaningful test data and to provide better discrimination performance.

To quantify these remarks, let us consider the Fisher ratio J (column 2 of Table II) calculated for the various training sets. Considering the first three entries, we note that J increases as the number of training set images used increases (114 for all 72 images, 126 for 36 images, and 1180 for 24 images). This reflects the obvious fact that it is easier to separate 24 or 36 images than 72. The large increase (126–1180) in going from 36 to 24 images is also due to the 12 images chosen per class (all were within ±50° of broadside and thus cluster well). Similar remarks apply to the large J (2322) obtained for the case when the 12 training set images per class were chosen from the same side of the ship.

Recall that a large J merely reflects a small least-squares error that indicates that the training set separates well between classes and clusters well within each class. However, a large J value does not imply good performance on test images. In fact, the opposite effect occurs. (An LDF formed from a training set with a large J yields worse performance than does one with a low J.) This is obvious and provides insight into a more formal rather than an ad hoc technique for training set selection that we now advance.

Once a training set is selected, our algorithm produces an LDF with the largest J possible (for the given training set). The value of J obtained indicates the divergence of the training set. Thus an LDF with a low J indicates that the training set used is more statistically representative (larger divergence) of the imagery than is an LDF with a larger J. Once we have chosen the number of training set images we desire to use (e.g., 12 per class), we can use J as a measure of which training set and which LDF is best. Specifically, we choose the training set and LDF with the lowest J.

Comparing J for the three training sets of 12 images per class (experiments 3–5), we find a low J (158) when the 12 training images per class were evenly distributed (every 30°) around the entire ship rather than clustered on one side of the ship (experiment 4, $J = 2322$) or near broadside on both sides of the ship (experiment 3, $J = 1180$). Intuition also tells us that training set 5 should be more statistically representative of the image classes. (The J value reflects and quantifies this.) Our experimental data confirm this. [The LDF produced from training set 5 yields fewer errors (zero) than did the LDFs in cases 3 and 4.] This technique for training set selection is directly applicable to other feature spaces and other classifiers. We will elaborate further on this issue and other methods of training set selection that explain the classification percentages in experiments 3 and 4 in a future publication.

Care must still be exercised in selecting the number of training set images N when D features are used. Specifically, to use a least-mean-square solution, we require $N > D$. However, the N training set images must be totally independent. This is difficult to achieve in practice, and hence we do not choose $N = D + 1$ but rather utilize several additional training set images. When D was large (e.g., 56 features), we found the sum of the covariance matrices for both classes to be very unstable with digital truncation error alone often being sufficient to affect the rank of the matrix and permit its inversion. In general, operating with a smaller dimensionality feature space is also preferable for computational reasons, but this is not the major issue in an optical processor.

We have considered other simpler observation spaces such as the magnitude of the Fourier transform of the input object. The wedge-ring detector was originally intended to be used in Fourier plane sampling. Since the information content of the power spectrum and the autocorrelation are similar (the latter is the Fourier transform of the former), it might appear that there is no difference in which plane we sample. However, the physical significance of the two representations (wedge-ring detector samples of the Fourier transform and the autocorrelation) is quite different. We make no effort to decide which is best for pattern recognition. We note that the sampling of the autocorrelation is equivalent to the chord histogram. We also note that, by blocking dc in forming the autocorrelation, the conventional chord distribution is produced (corresponding to the boundary of the object). More light level and a less narrow autocorrelation (with more structure) result from our generalized chord transform. Hence this appears preferable and to be a more attractive observation space for an optical processor.

Our architecture (in which the features of the input object are optically computed) is quite general and attractive. It utilizes the high-speed and parallel-processing features of an optical system for feature extraction, and it allows these optically computed features to be fed to any output classifier. Discussions of other optical pattern recognition systems using the Fisher, Fukunaga-Koontz, least-mean-square, and other classifiers were advanced in Sec. IV. However, these systems[15–17] used such classifiers to synthesize the filter used. Thus they all require different filters to be made for every different pattern recognition application and new filters to be made if different classifiers are used (as is required in certain applications). Our architecture is more general, since the optical system need not change if different pattern recognition problems are being addressed or if different classifiers are to be used.

Another important aspect of our approach can be seen by considering the application of the Fisher criteria to synthesis of a matched spatial filter using a Fourier coefficient feature space. In such an approach,[15] the LDF **w** is a linear sum of the Fourier coefficients of the training set data, and thus a matched spatial filter of this 2-D LDF can be fabricated and placed at the Fourier transform plane of a frequency plane correlator. The output correlation plane in this case would consist of peaks of light of amplitude proportional to m_1 or m_2 (the means of the projections for the data in classes 1 and 2, respectively). If the data in the input were in class 1, the output would have an amplitude proportional to m_1. If the data were of class 2, the output would be proportional to m_2 and, with the location of each peak of light, proportional to the location of each object in the input field of view. However, it has been shown in Ref. 8 that maximizing the Fisher ratio forces the output projection values for the two classes of data to be ± 1, respectively. Although the least-squares solution will not give projections of exactly ± 1, it will (with high probability) give positive and negative means for the two classes. Thus the intensity output from such an optical correlator will not allow one to distinguish the sign or polarity of the output projections and hence will not permit target classification. Heterodyne detection of the correlation plane output is thus necessary if a Fisher criterion is used for synthesis of the matched spatial filter in a correlator. Only with such methods can the shift-invariant advantages of such systems (over our feature-extraction method) be realized. In our use of the Fisher criteria (in which the classification is performed in a digital postprocessor), different projection polarities are easily handled and are in fact preferable. Our algorithm and general architecture can also be directly extended to multiclass pattern recognition applications (involving more than two classes of objects) more easily than can other optical architectures (that require a different optical system to be used when different numbers of classes of data are involved).

VII. Summary

In this paper, we have described a new optical system that computes the histogram of the generalized chord transformation of the input object. Our wedge-ring detector-sampled feature space provides a reduced dimensionality vector representation of the input object that has the fundamental data necessary for scale and rotational invariance. As our classifier, we chose the linear discriminant function that maximizes the Fisher ratio. Our initial experiments showed that we could recognize and correctly classify the input image data with as few as 18 features and that using the proper training set of 12 images per class, we could achieve 100% correct recognition of the 72 images available in our data base. Extensions of this approach to multiclass pattern recognition were noted together with the use of other output classifiers and the initial ideas for techniques to select the optimal training set of data for any such system.

The support of this research by a grant from the Air Force Office of Scientific Research (grant AFOSR 79-0091) is gratefully acknowledged.

References

1. H. J. Caulfield, R. Haimes, and D. Casasent, Opt. Eng. **19,** 152 (1980).
2. D. Casasent and A. Furman, Appl. Opt. **16,** 1662 (1977).
3. D. Casasent and D. Psaltis, Proc. IEEE **65,** 77 (1977).
4. H. J. Caulfield and R. Haimes, Appl. Opt. **19,** 181 (1980).
5. C. F. Hester and D. Casasent, Appl. Opt. **19,** 1758 (1980).
6. D. Casasent, J. Pauly, and D. Fetterly, Proc. Soc. Photo-Opt. Instrum. Eng. **302,** 126 (1981).
7. H. Kasdan and D. Meade, Proc. Electron. Opt. Syst. Des. 248 (1975).
8. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973); see also Y. T. Chien, *Interactive Pattern Recognition* (Dekker, New York, 1978).
9. G. Tenery, IEEE Trans. Mil. Electron. **ME-7,** 196 (1963).
10. D. J. H. Moore and D. J. Parker, Pattern Recognition **6,** 149 (1974).
11. D. G. Nichols, Opt. Commun. **43,** 168 (1982).
12. J. Rau, J. Opt. Soc. Am. **56,** 1490 (1966).
13. H. N. Kannapell, "Image Classification Based on a Three-Dimensional Angle Histogram," M. Eng. Thesis, U. Louisville, Louisville (1981).
14. S. P. Smith and A. K. Jain, Comput. Graphics Image Process. **20,** 1699 (1982).
15. H. J. Caulfield and M. H. Weinberg, Appl. Opt. **21,** 1699 (1982).
16. J. R. Leger and S. H. Lee, J. Opt. Soc. Am. **72,** 556 (1982).
17. Z. H. Gu, J. R. Leger, and S. H. Lee, J. Opt. Soc. Am. **72,** 787 (1982).
18. T. Marill and D. M. Green, IEEE Trans. Inf. Theory **IT-9,** 11 (1963).

13. EFFICIENT APPROACH TO DESIGNING LINEAR COMBINATION FILTERS

Reprinted from **Applied Optics**, Vol. 22, page 1445, May 15, 1983
Copyright © 1983 by the Optical Society of America and reprinted by permission of the copyright owner.

Efficient approach to designing linear combination filters

B. V. K. Vijaya Kumar

Various linear combinations of simple matched spatial filters have been proposed in the literature to improve the discrimination in multiclass pattern recognition. It has been shown that all such approaches based on deterministic constraints can be modeled as similar matrix/vector problems, the only differences arising in the individual constraint vectors. Since the design of any of these linear combination filters (LCF) can be posed as a common matrix/vector problem, efficient iterative methods can be used to determine the LCFs. *The application of one such method called the modified hyperplane (MHP) method for determining* the LCF is described and its convergence behavior is numerically investigated for a set of seven patterns. It is shown that the MHP method yields correct LCFs (with rms error <0.1%) in less than ten iterations.

I. Introduction

Matched spatial filters (MSF) have been extensively used in optics[1] to identify a known pattern in a noisy background. This simple approach has been shown to fail in the presence of common image distortions such as in-plane rotations and scale changes. Simple MSFs are also seen to be inadequate[2] for multiclass pattern recognition applications. Multiclass pattern recognition (MPR) *can be defined as the task of discriminating* a pattern from among many classes of patterns. A good example of MPR is the identification of a given letter in a language from its alphabet.

Linear combinations of MSFs matched to the individual patterns have been used to improve the discrimination in MPR.[2] Such an approach resulted not only in improved discrimination between patterns but also in increased tolerance to geometrical distortions in a given pattern class. Hester and Casasent[3] have shown through the use of the Gram-Schmidt procedure that a single spatial filter can be designed to produce identical cross-correlation values with all the patterns in the input set. For obvious reasons, this type of filter is known as an equal correlation peak (ECP) filter. Braunecker *et al.*[4] utilize multiple linear combination filters to produce compact binary descriptions of the input pattern. All these filters are purely deterministic in the sense that they depend only on the training set patterns. Principles of statistical pattern recognition were used by Caulfield and Haimes[5] to produce statistically optimal LCFs. The common nature of all these LCFs is well unified by Caulfield.[6]

In this paper we utilize the unifying notation of Caulfield[6] and our earlier work[7] which showed that these previous LCF approaches can be reduced to a common matrix/vector problem. We note that the size of this matrix/vector problem is directly proportional to the number of patterns in the input training set and that, since the accuracy and the usefulness of a LCF increases with the size of the input training set, we need to solve larger matrix/vector problems for better accuracy. The contribution of this paper is that this reformulation of the LCF design into a single matrix/vector solution can benefit from the vast amount of results available[8] for solving matrix/vector problems. As an illustration of this approach, we propose the use of the modified hyperplane (MHP) method to design the LCFs. The MHP was initially used by Ramakrishnam *et al.*[9] for image restoration and by Vijaya Kumar[10] for adaptive array null synthesis. In this paper we point out the usefulness of this technique in a LCF synthesis for MPR.

II. Linear Combination Filters

Consider the set of N input characters $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$. Each pattern $\mathbf{X}_i$ is converted from an image to a vector for convenience. The length of the vectors equals the total number of pixels in the patterns. A linear combination filter $\mathbf{Y}$ is defined as

$$\mathbf{Y} = a_1\mathbf{X}_1 + a_2\mathbf{X}_2 + \ldots + a_N\mathbf{X}_N, \tag{1}$$

where $a_1, a_2, \ldots, a_N$ are real values denoting the contribution of each input pattern in the LCF $\mathbf{Y}$. The LCF designs discussed below describe three techniques for determining the coefficients $a_1, a_2, \ldots, a_N$. These techniques differ only in the constraints imposed for

---

The author is with Carnegie-Mellon University, Department of Electrical Engineering, Pittsburgh, Pennsylvania 15213.

Received 22 January 1983.

0003-6935/83/101445-04$01.00/0.

determining these weights and the number of filters used. The N mutually orthogonal filters (MOF) $\mathbf{Y}_1,\mathbf{Y}_2,\ldots,\mathbf{Y}_N$ are designed such that each filter correlates with only one pattern and produces zero cross correlation with all other inputs, i.e.,

$$\mathbf{Y}_i \bullet \mathbf{X}_j = \delta_{ij}, \tag{2}$$

where the dot denotes the inner product of the two vectors and δ_{ij} denotes the Dirac delta function.[2]

The ECP filter approach suggested by Hester and Casasent[3] requires that a single LCF $\mathbf{Y}$ be determined that correlates equally well with all the patterns in the training set, i.e.,

$$\mathbf{Y} \bullet \mathbf{X}_i = 1 \qquad \text{for } i = 1,2,\ldots,N. \tag{3}$$

In the nonredundant filter (NRF) method[4] M LCFs $\mathbf{Y}_1,\mathbf{Y}_2,\ldots,\mathbf{Y}_M$ are chosen such that, when all these M filters are correlated with a given input $\mathbf{X}_i$, the resulting M cross-correlation values provide an M-tuple description of i, the input number. If M is greater than $\log_2 N$, we can see that the resulting description is a binary one. To illustrate this method, consider an input set of three patterns $\{\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3\}$ and two nonredundant filters $\mathbf{Y}_1$ and $\mathbf{Y}_2$. Since the binary descriptions of the numbers 1,2,3 are (0,1),(1,0),(1,1), respectively, $\mathbf{Y}_1$ and $\mathbf{Y}_2$ should satisfy

$$\begin{aligned} &\mathbf{Y}_1 \bullet \mathbf{X}_1 = 0; \quad \mathbf{Y}_2 \bullet \mathbf{X}_1 = 1;\\ &\mathbf{Y}_1 \bullet \mathbf{X}_2 = 1; \quad \mathbf{Y}_2 \bullet \mathbf{X}_2 = 0;\\ &\mathbf{Y}_1 \bullet \mathbf{X}_3 = 1; \quad \mathbf{Y}_2 \bullet \mathbf{X}_3 = 1. \end{aligned} \tag{4}$$

The LCF solutions to Eqs. (2)–(4) have been shown to be described[7] in terms of the solution $\mathbf{a}$ to

$$\mathbf{R}\mathbf{a}_i = \mathbf{u}_{mi}. \tag{5}$$

For the MOF, $\mathbf{u}_{mi} = \mathbf{u}_{1i}$ is an N-element vector with a 1 in the ith position and zeros everywhere else. $\mathbf{R}$ is the $N \times N$ correlation matrix of the input data $\{\mathbf{X}_1,\ldots,\mathbf{X}_N\}$. For the ECP, $\mathbf{u}_{mi} = \mathbf{u}$ is an N-element vector containing all ones. For the nonredundant filters, $\mathbf{u}_{mi} = \mathbf{u}_{2i}$ is a vector consisting of ones and zeros only, corresponding to the ith bits in the binary descriptions of numbers $1,2,\ldots,N$. In our example $N = 3$, $M = 2$, and $\mathbf{u}_{21}$ would be given as $[0\ 1\ 1]^T$, where the superscript T denotes transpose. Similarly $\mathbf{u}_{22}$ is given as $[1\ 0\ 1]^T$. A nonbinary description such as ternary can also be used to reduce the number M of the filters but at the expense of increased dynamic range requirements in the output plane of the correlators.

From this we see that all three techniques reduce to the same matrix/vector problem in Eq. (5). Only the constraint vectors $\mathbf{u}_{1i}$, $\mathbf{u}$, and $\mathbf{u}_{2i}$ differ in the three cases. This leads to the possibility of solving all three problems with only a few changes. For good filters we require a large input data set, thus increasing the value of N. Since the size of the matrix $\mathbf{R}$ is $N \times N$, we need to consider efficient methods of solving for $\mathbf{a}$.

III. Modified Hyperplane (MHP) Method

The matrix/vector Eq. (5) can be solved by finding the inverse of the matrix $\mathbf{R}$. However, N is often so large that determining the inverse of $\mathbf{R}$ directly can be computationally cumbersome. Projection techniques[11] are usually suggested for solving such equations when N is large. In projection methods, the N constraints in Eq. (5) in terms of the unknown coefficients $a_1,a_2,\ldots,a_N$ are considered to represent N hyperplanes in an N-dimensional hyperspace. The N coordinates of the intersection of these N hyperplanes are the N-coefficient values.

The alternating orthogonal projection (AOP) method suggested by Youla[11] projects the initial guess (solution) vector orthogonally on all N hyperplanes, one after another. Ramakrishnam *et al.*[9] suggested a modification to this method to improve its convergence behavior. This modification involves rearranging the N hyperplanes of our equations so that the adjacent hyperplanes are orthogonal to each other. This modification was shown[10] to be preferable to AOP both in terms of the computational load per iteration and in terms of the number of iterations needed for convergence.

Let us consider Eq. (5) with $\mathbf{u}_{mi} = \mathbf{u}$ (a unit vector) for illustration purposes. The other two equations can be handled in a similar manner. Let us denote the N rows of the matrix $\mathbf{R}$ by the vectors $\mathbf{S}_1,\mathbf{S}_2,\ldots,\mathbf{S}_N$. Then the N hyperplanes in the N-dimensional space are represented by

$$\mathbf{S}_i{}^T\mathbf{a} = u(i), \tag{6}$$

where $u(i)$ is the ith element in the constraint vector $\mathbf{u}$. The two steps in the MHP are detailed in the next two subsections.

A. Hyperplane Rearrangement

The hyperplanes characterized by $\mathbf{S}_1,\mathbf{S}_2,\ldots,\mathbf{S}_N$ are rearranged such that the new hyperplanes $\mathbf{S}_1^\#,\mathbf{S}_2^\#,\ldots,\mathbf{S}_N^\#$ are orthogonal to their adjacent hyperplanes. The algorithm to perform this is

$$\mathbf{S}_1^\# = \mathbf{S}_1, \tag{7}$$

$$\mathbf{S}_n^\# = \mathbf{S}_n - [(\mathbf{S}_n{}^T\mathbf{S}_{n-1})/(\mathbf{S}_{n-1}{}^T\mathbf{S}_{n-1})]\mathbf{S}_{n-1}, \qquad n = 2,3,\ldots,N, \tag{8}$$

$$u^\#(1) = u(1), \tag{9}$$

$$u^\#(n) = u(n) - [(\mathbf{S}_n{}^T\mathbf{S}_{n-1})/(\mathbf{S}_{n-1}{}^T\mathbf{S}_{n-1})]u^\#(n-1), \qquad n = 2,3,\ldots,N. \tag{10}$$

Now the matrix equation to be solved is

$$\mathbf{R}^\#\mathbf{a} = \mathbf{u}^\#, \tag{11}$$

where the superscript # throughout indicates modified quantities.

B. Projection Algorithm

The projection step starts with an initial guess $\mathbf{a}_0$ for the solution vector. This point $\mathbf{a}_0$ in the N-dimensional space is then orthogonally projected onto the first hyperplane $\mathbf{S}_1^{\#T}\mathbf{a} = u\ (1)$ to yield the next solution vector estimate $\mathbf{a}_1$. In general the projection onto the nth hyperplane $\mathbf{S}_n^{\#T}\mathbf{a} = u\ (n)$ results in the nth estimate for the solution vector:

$$\mathbf{a}_n = \mathbf{a}_{n-1} - \{[\mathbf{a}_{n-1}{}^T\mathbf{S}'_n - u'(n)]/(\mathbf{S}'_n{}^T\mathbf{S}'_n)\}\mathbf{S}'_n, \quad n = 1,2,\ldots,N. \tag{12}$$

Once $\mathbf{a}_N$ is determined, it is projected back again onto the first hyperplane to start a second iteration. This process of going from $\mathbf{a}_0$ to $\mathbf{a}_N$ is known as one complete iteration. It has been observed[10] that this algorithm converges to the true solution in a small number of iterations.

IV. Numerical Results

As was discussed earlier, all three filters use the same matrix **R** for a given input training set. Once **R** is determined, the coefficients for the three LCFs can be obtained by using the MHP method in Eq. (5) with a common matrix **R**. Only the constraint vectors are different in the three cases.

As an illustrative example, we considered the input data set used by Hester and Casasent.[3] This data set consists of seven IR views of a tank. We applied the MHP technique to the 7×7 correlation matrix **R** provided by Hester and Casasent.[3] To obtain the ECP filter we used the constraint vector $[1\ 1\ 1\ 1\ 1\ 1\ 1]^T$. The starting guess for the solution vector was also a vector with all ones, i.e.,

$$\mathbf{a}_0 = [1\ 1\ 1\ 1\ 1\ 1\ 1]^T. \tag{13}$$

The recursions are carried out using the MHP algorithm outlined in Sec. III, and these recursions are stopped when the error,

$$\text{err} = \sqrt{\|\mathbf{a}_n - \mathbf{a}^*\|}, \tag{14}$$

decreases below a predetermined threshold, where $\| \bullet \|$ denotes the Euclidean norm of the vector and $\mathbf{a}^*$ denotes the true coefficient vector obtained by the direct solution of the corresponding equation. In general $\mathbf{a}^*$ will not be available *a priori* and we can use the difference between successive estimates of **a** to monitor convergence. The MHP can be deemed to be successful if the error decreases as the number of complete iterations increase.

The error in the coefficient vector obtained by the MHP method is listed in Table I for ECP, MOF, and NRF linear combination filters. Close inspection of Table I reveals that the solution vector obtained after ten complete iterations is quite accurate. After

Table I. The rms Errors Vs Iteration Number

Iteration no.	ECP	MOF3	NRF1
1	2.3709	2.6099	2.6458
2	1.1986	0.2114	1.4359
3	0.4705	0.0403	0.5615
4	0.1831	0.0147	0.2189
5	0.07157	0.00582	0.08557
6	0.02797	0.00228	0.03344
7	0.01093	0.00089	0.01307
8	0.00427	0.00035	0.00510
9	0.00167	0.00014	0.00199
10	0.00065	0.00005	0.00078
15	0.6×10^{-5}	0.5×10^{-6}	0.7×10^{-5}
20	0.7×10^{-7}	0.4×10^{-8}	0.9×10^{-7}
25	0.4×10^{-8}	0.2×10^{-9}	0.9×10^{-9}

Table II. The rms Errors Vs Iteration Number

Iteration no.	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$
1	1.2432	1.1507	1.214	1.2928	2.3709
2	0.0700	0.035	0.15626	0.26394	1.1986
3	0.6×10^{-4}	0.18×10^{-2}	0.67×10^{-2}	0.01592	0.4705
4	0.55×10^{-5}	0.16×10^{-3}	0.46×10^{-3}	0.97×10^{-3}	0.18314
5	0.45×10^{-7}	0.14×10^{-4}	0.47×10^{-4}	0.89×10^{-4}	0.07156
6		0.12×10^{-5}	0.43×10^{-5}	0.94×10^{-5}	0.02797
7		0.11×10^{-6}	0.37×10^{-6}	0.93×10^{-6}	0.0109
8		0.37×10^{-8}	0.38×10^{-7}	0.85×10^{-7}	0.0043
9		0.19×10^{-8}	0.11×10^{-7}	0.12×10^{-7}	0.0017
10			0.85×10^{-8}	0.53×10^{-8}	0.00065
11			0.11×10^{-8}	0.42×10^{-8}	0.00025
12				0.56×10^{-8}	0.00009
13				0.53×10^{-8}	0.38×10^{-4}
14				0.42×10^{-8}	0.15×10^{-4}
15					0.59×10^{-5}
20					0.69×10^{-7}
25					0.37×10^{-8}

twenty-five complete iterations, an error of only 0.4×10^{-8} is observed indicating an accurate solution using the MHP. If an accuracy of 0.1% suffices, we need to use only ten complete iterations for the design of an ECP filter.

In all our numerical experiments, we have observed that the rms error monotonically decreases to a small value as the number of iterations is increased until a certain stage beyond which the error either remains constant or oscillates between two small values. This can be easily shown to be due to the finite accuracy in the digital computer representation.

The number of iterations required to obtain a solution vector of a desired accuracy depends on the size of the matrix **R**. To illustrate the dependence of convergence on the size N of the input data set, we considered different input data sets $\{\mathbf{X}_1,\mathbf{X}_2,\ldots,\mathbf{X}_N\}$, where N can take on values from 3 to 7. The resulting correlation matrices are simply the principal minors of the 7×7 matrix **R**. The rms errors obtained after each iteration for various matrix or training set sizes are shown in Table II. From this table we see that more iterations are required to obtain the same accuracy as the size N of the input data set is increased.

Thus far we have discussed the use of the MHP for ECP filter determination and its convergence performance. A similar numerical analysis was carried out for the seven mutually orthogonal filters (MOF) as well as the three binary nonredundant filters (NRF) of Sec. II. The convergence behavior for MOF3 as well as for NRF1 is presented in Table I. MOF3 is obtained using the MHP method in a manner similar to that used for the ECP except that a constraint vector $[0\,0\,1\,0\,0\,0\,0]^T$ was used instead of a vector with all ones. Similarly, NRF1 was obtained by using the constraint vector $[1\,0\,1\,0\,1\,0\,1]^T$. We note from Table I that the rms error once again decreases as the number of iterations is increased. Less than ten complete iterations are needed to achieve an error <0.1%.

Strictly speaking, only the seven MOFs need to be computed. The ECP and NRF filters can then be expressed as simple known linear combinations of these seven MOFs. Such an approach is computationally inefficient when only one type of filter is of interest. For example, when an ECP filter is needed, the indirect approach using MOFs requires seven times more work compared with the direct approach that utilizes the constraint vector $[1\,1\,1\,1\,1\,1\,1]^T$.

One can easily expand the NRF synthesis to an arbitrary base as 8. With a base of 8, the NRF would yield the integer i as the output when the input is $\mathbf{X}_i$. Then the constraint vector becomes $[1\,2\,3\,4\,5\,6\,7]^T$. Even with such filters we did not need more than ten complete iterations for an accurate solution.

Thus, we have shown that for various filter designs we can use MHP for less than ten complete iterations when the input set contains seven patterns. We have also shown this to be a unified computational approach to the design of linear combination filters since only the constraint vectors need to be changed from one type of filter to another type.

V. Conclusions

For multiclass pattern recognition, several approaches utilize linear combination filters for improved discrimination. All these approaches are based on deterministic constraints and can be modeled as similar matrix/vector problems. Such an approach unifies the computational considerations because only the appropriate constraint vectors need to be specified for different filters. In this paper we have examined the convergence behavior of the modified hyperplane (MHP) method in the filter design. Good convergence properties were exhibited for the correlation matrices that arose in these cases.

References

1. A. VanderLugt, *IEEE Trans. Inf. Theory* **IT-10**, 139 (1964).
2. H. J. Caulfield and W. T. Maloney, Appl. Opt. **8**, 2354 (1969).
3. C. F. Hester and D. Casasent, Appl. Opt. **19**, 1758 (1980).
4. B. Braunecker, R. Hauck, and A. W. Lohmann, Appl. Opt. **18**, 2746 (1979).
5. H. J. Caulfield and R. Haimes, Appl. Opt. **19**, 181 (1980).
6. H. J. Caulfield, Appl. Opt. **19**, 3877 (1980).
7. D. Casasent, B. V. K. Vijaya Kumar, and V. Sharma, "Synthetic Discriminant Functions for 3 Dimensional Object Recognition," Proc. Soc. Photo-Opt. Instrum. Eng. **360** (1982).
8. L. A. Hageman and D. M. Young, *Applied Iterative Methods* (Academic, New York, 1981).
9. R. S. Ramakrishnam, S. K. Mullick, R. K. S. Rathore, and R. Subramanian, Appl. Opt. **18**, 464 (1979).
10. B. V. K. Vijaya Kumar, IEEE Trans. Antennas Propag. **AP-30**, 512 (1982).
11. D. C. Youla, IEEE Trans. Circuits Syst. **CS-25**, 694 (1978).

We would like to thank D. Casasent for his critical reading of the manuscript and AFOSR for support under grant AFOSR79-0091-Amendment D.

14. PERFORMANCE OF SYNTHETIC DISCRIMINANT FUNCTIONS FOR INFRARED SHIP CLASSIFICATION

IOCC'83
IEEE Cat No 83CH1880-2
SPIE Vol 422

PERFORMANCE OF SYNTHETIC DISCRIMINANT FUNCTIONS FOR INFRARED SHIP CLASSIFICATION

David Casasent and Vinod Sharma

Carnegie-Mellon University
Department of Electrical Engineering
Pittsburgh, Pennsylvania 15213

ABSTRACT

Our unified synthetic discriminant function (SDF) filter synthesis technique using the correlation matrix of the image training set is reviewed. Four different synthetic discriminant functions for intra-class recognition, inter-class discrimination and both intra and inter-class pattern recognition are considered. All techniques proposed are appropriate for object identification, location and classification in the presence of 3-D geometrical distortions in the input object. Initial results obtained on a set of four different classes of infrared ship imagery are presented. Excellent performance (over 90% correct classification) was achieved.

1. INTRODUCTION

Many different optical pattern recognition techniques to permit object recognition and classification in the face of geometrical distortions have been proposed. These include: space-variant processors[1-2], generalized matched filters[3-4], synthetic discriminant functions (SDFs)[5-7], phase-coded systems[8] and others. In this paper, we consider new SDF research. The SDF approach to 3-D distortion-invariant pattern recognition appears to be most attractive, since a matched spatial filter of the SDF can be produced and used in an optical frequency plane correlator. Thus, SDFs exhibit shift-invariance and hence are capable of recognizing multiple objects in the field of view and providing information on their location. Other techniques such as space-variant and phase-coded systems require the input object to be centered in the input plane. Since SDFs are used in correlators, they provide a processing gain and are thus capable of locating objects in the presence of severe noise and man-made clutter.

In Section 2, we describe SDFs and in Section 3 we discuss our new generalized technique[9] for SDF synthesis and how the same basic algebraic equation can be used to synthesize four different types of SDFs. This synthesis technique is attractive because it automatically determines the optimum basis function set. It is thus preferable to ad hoc techniques and to more restricted techniques[3-4] using a Fourier transform coefficient basis function set. In Section 4, we discuss the infrared ship image data base we used and in Section 5 we present our initial experimental results obtained for four different types of SDFs.

2. SDF BASIC SYNTHESIS

The general pattern recognition problem that we initially consider is the design of a filter function $h(x,y)$ that can recognize all objects $\{f_n(x,y)\}$ of one class of input data, independent of geometrical distortions present in the input image. As noted in Section 1, we plan to optically correlate all images $\{f_n\}$ with h and will design h such that all correlations are equal to a constant output value, i.e.

$$h \circledast f_n = c. \tag{1}$$

As our constant c, we select unity (with no loss of generalization). Because this technique is quite analogous to a hyperspace description of the input image data, we represent input images and the SDF h by vectors and write (1) as

$$\underline{h} \cdot \underline{f}_n = 1. \tag{2}$$

In conventional pattern recognition terminology, one would describe each f_n as a linear combination of a set of basis functions ϕ_j, i.e.

$$\underline{f}_n = \sum_{1} a_{nj}\underline{\phi}_j, \tag{3}$$

where in (3) the f_n and ϕ_j are 2-D functions of x and y (i.e. images). We write each ϕ_j in terms of a linear summation of the input images f_n as

$$\underline{\phi}_j = \sum_n d_{jn}\underline{f}_n. \tag{4}$$

Analogous to (3), we write our unknown filter function h as a linear combination of the same basis functions

$$\underline{h} = \sum_j b_j\underline{\phi}_j. \tag{5}$$

Substituting (4) into (5), we obtain

$$\begin{aligned}\underline{h} &= \sum_j b_j \sum_n d_{jn}\underline{f}_n \\ &= b_1\sum_n d_{1n}\underline{f}_n + b_2\sum_n d_{2n}\underline{f}_n + \dots\end{aligned} \tag{6}$$

Grouping terms in f_1 and f_2, etc. separately, we can rewrite (6) as

$$\underline{h} = e_1\underline{f}_1 + e_2\underline{f}_2 + \dots = \sum_n e_n\underline{f}_n. \tag{7}$$

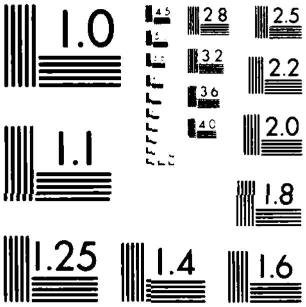
We have now described our filter function h as a linear combination of the input training set $\{f_n\}$ It remains only to determine the weighting coefficients e_n in (7) and then we have determined our

CH1880-4/83/0000/0193$00

AD-A136 216 OPTICAL DATA PROCESSING FOR MISSILE GUIDANCE(U)
CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF ELECTRICAL
ENGINEERING D CASASENT 30 SEP 83 AFOSR-TR-83-1103
UNCLASSIFIED AFOSR-79-0091 F/G 17/7 NL

2/2

END
DATE
FILMED
1 84
DTIC


1.0
4.5
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.1
1.8
1.25
1.4
1.6

SDF h. To achieve this, we recall our condition in (2) and substitute (7) into (2) to obtain

$$\underline{h}\cdot\underline{f}_n = 1 = (\sum_j e_j\underline{f}_j)\cdot\underline{f}_n = \sum_j e_j(\underline{f}_j\cdot\underline{f}_n). \qquad (8)$$

We recognize the last term in parentheses in (8) to be the correlation, or projection, of f_j onto f_n. This is element R_{jn} of the correlation matrix $\underline{R}$ of the training set data $\{f_n\}$ and (8) thus becomes

$$\sum_j e_j R_{jn} = 1. \qquad (9)$$

We write (9) in matrix-vector notation as

$$\underline{R}\,\underline{e} = \underline{u}, \qquad (10)$$

where $\underline{u}$ is the unit vector. The solution $\underline{e}$ to (10) and hence the information necessary to calculate the SDF $\underline{h}$ in (7) is

$$\underline{e} = \underline{R}^{-1}\underline{u}. \qquad (11)$$

Thus, to determine the e_n in (9) or the eigenvalue in (11), we form the correlation matrix $\underline{R}$ of the training set data $\{f_n\}$, invert this matrix and multiply it by the constant vector $\underline{u}$.

3. GENERALIZED SDF SYNTHESIS

We refer to the SDF described in Section 3 as an equal correlation peak (ECP) SDF, since it provides a constant equal correlation peak value for all input images $\{f_n\}$ of one class of data. We should note explicitly that the images in the training set $\{f_n\}$ can (and usually are) images of the same object taken with different geometrical distortions (e.g. different rotated or aspect views). This type of SDF is of use in intra-class pattern recognition (recognition of one object class, independent of geometrical distortions within this object class). This is similar to our initial SDF work[5], but uses a new SDF synthesis technique.

Next, we consider a mutual orthogonal function (MOF) SDF for inter-class pattern recognition. We consider a two-class pattern recognition problem with one image f_1 in class one and one image f_2 in class two. We desire two filters h_1 and h_2, such that

$$\underline{f}_n\cdot\underline{h}_m = \delta_{nm}, \qquad (12)$$

i.e.

$$\begin{aligned} &\underline{f}_1\cdot\underline{h}_1 = 1,\ \underline{f}_2\cdot\underline{h}_1 = 0 \\ &\underline{f}_1\cdot\underline{h}_2 = 0,\ \underline{f}_2\cdot\underline{h}_2 = 1. \end{aligned} \qquad (13)$$

Each filter will again be a linear combination of all training set images (f_1 and f_2), i.e.

$$h_1 = \Sigma a_n f_n,\ h_2 = \Sigma b_n f_n. \qquad (14)$$

To compute the a_n and the b_n coefficients in (14), we solve the matrix-vector equations

$$\underline{R}\,\underline{a} = \underline{u}_1,\ \underline{R}\,\underline{b} = \underline{u}_2, \qquad (15)$$

where $\underline{R}$ is the 2 x 2 correlation matrix, $\underline{u}_1^T = [1,0]$ and $\underline{u}_2^T = [0,1]$. This filter synthesis technique is similar to that described in[6], but is formulated in our new unified notation. This technique produces one filter per input object class. As such, it is useful for inter-class pattern recognition.

We now provide a generalization of this technique that allows both intra-class and inter-class discrimination. We formulate this new MOF SDF for the case of two object classes with N_1 and N_2 images respectively in each class. We desire two SDFs h_1 and h_2, such that

$$\begin{aligned} &\underline{h}_1\cdot\underline{f}_{n1} = 1,\ \underline{h}_1\cdot\underline{f}_{n2} = 0 \\ &\underline{h}_2\cdot\underline{f}_{n2} = 0,\ \underline{h}_2\cdot\underline{f}_{n2} = 1, \end{aligned} \qquad (16)$$

where $\underline{f}_{n1}$ and $\underline{f}_{n2}$ denote any member of class one or class two respectively. We describe these SDFs as in (14), where the summation is now over $N_1 + N_2$. There are now $N_1 + N_2$ weighting coefficients a_n and b_n. They are given by (15), where now $\underline{R}$ is an $(N_1 + N_2)\mathrm{x}(N_1 + N_2)$ matrix and $\underline{u}_1^T = [1...1,0...0]$ contains N_1 ones and N_2 zeroes and $\underline{u}_2^T = [0...0,1...1]$ contains N_1 zeroes and N_2 ones. We refer to this as a mutual orthogonal correlation peak (MOCP) SDF. It recognizes different object classes (inter-class discrimination) even in the face of geometrical distortions within each object class (intra-class recognition). We will discuss later how this technique can be extended to greater than two classes. We also note that for the two object class example, one filter suffices for both discrimination and recognition. We also note that the choice of the constants in (13) and (16) is arbitrary. In our digital simulations, we used $\underline{+}1$, rather than 1 and 0. An optical system would use 0 and 1 or 1 and 2, etc., since only intensity output detection is easily possible in optical processors.

Next, we consider the synthesis of a nonredundant SDF or a nonredundant filter (NRF). This is similar to those described by Braunecker et al[10] expressed in terms of our unified notation in (11) and (15). We also extend the original work in[10] + the case of a NRF for both intra-class recognition and inter-class discrimination. As an example, we consider a three-class pattern recognition problem. We denote the three classes of objects by $\{f_{N1}\}$, $\{f_{N2}\}$ and $\{f_{N3}\}$, where N_1, N_2 and N_3 training set images are present respectively in each object class. We consider one SDF h, such that

$$\underline{h}\cdot\underline{f}_{n1} = 1,\ \underline{h}\cdot\underline{f}_{n2} = 2,\ \underline{h}\cdot\underline{f}_{n3} = 3 \qquad (17)$$

where as before, $\underline{f}_{n1}$, $\underline{f}_{n2}$ and $\underline{f}_{n3}$ denote any member of object class one, two or three respectively. As seen, the value of the output correlation determines the object class. As before, the choice of the constants 1, 2 and 3 in (17) is arbitrary. We describe the SDF h as

$$\underline{h} = \Sigma a_n\underline{f}_n, \qquad (18)$$

where the summation is over all $N_1 + N_2 + N_3$ training set images. In matrix-vector terminology, the problem is described as the solution of $\underline{R}\,\underline{a} = \underline{u}_3$ for $\underline{a}$, where $\underline{u}_3 = [1...1,2....2,3...3]^T$ contains N_1 ones, N_2 twos and N_3 threes and where $\underline{R}$ is of order $N_1 + N_2 + N_3$. The solution for the weight $\underline{a}$ in (18) is just $\underline{a} = \underline{R}^{-1}\underline{u}_3$ as before.

As our final SDF, we consider a multi-class MOCP SDF. For this case, we consider a four-class problem with training set images $\{f_{N1}\}$, $\{f_{N2}\}$, $\{f_{N3}\}$ and $\{f_{N4}\}$ and the synthesis of two filters whose truth table is given in Table 1. From Table 1, we see that four unique pairs of correlation output values result and determine the input object class. The two filters are described by

$$\underline{h}_1 = \Sigma a_n \underline{f}_n, \; \underline{h}_2 = \Sigma b_n \underline{f}_n, \tag{19}$$

where the summations are over $N_1 + N_2 + N_3 + N_4$ (i.e. the full training set of images). The coefficients $\underline{a}$ and $\underline{b}$ are defined by the solution to the matrix-vector equation

$$\begin{bmatrix} a_1 & b_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ a_N & b_N \end{bmatrix} = \begin{bmatrix} \\ R \\ \\ \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ \cdot & \cdot \\ 0 & 1 \\ \cdot & \cdot \\ 1 & 0 \\ \cdot & \cdot \\ 1 & 1 \end{bmatrix}, \tag{20}$$

where the vector has N_1 pairs (0,0), N_2 pairs (0,1), etc. and where $\underline{R}$ is of order $N_1 + N_2 + N_3 + N_4$.

TABLE 1
Truth Table for the Four-Class
Multi-Class MOCP SDF

INPUT DATA	FILTER h_1 OUTPUT	FILTER h_2 OUTPUT
f_{N1}	0	0
f_{N2}	0	1
f_{N3}	1	0
f_{N4}	1	1

4. DATA BASE

To test the performance of the four SDFs described in Section 3, we used a ship image data base. It contained 36 images (0-350°, at 10° increments) of four different ships taken from a 90° depression angle. Representative examples of all four ships (broadside views only) are shown in Figure 1. Each image was 128 x 32 pixels. We numbered the class one images one 1-36, the class two images as 37-72, etc.

5. INITIAL SIMULATED RESULTS

In Table 2, we summarize the experimental results for our four SDFs calculated as described in Section 3.

For our ECP-1 SDF, we calculated the intraclass filter from only six of the 36 ship images in class one. When this SDF was correlated with all 36 images of class one, all output correlation peak intensities were above the 0.5 threshold value chosen. Similar results were obtained for the class two imagery. This data is contained in the ECP-2 SDF experiments in Table 2. These first two experiments demonstrated for the first time the fact that an ECP SDF can correctly recognize and classify other distorted versions of the input object not present in the training set.

For our MOCP SDF, we calculated one SDF for intra and inter-class pattern recognition, using the same six training set images per class. For this one SDF, 69 of the 72 test images gave correct peak correlation outputs with respect to the threshold of 0.5. This represents over 97% correct recognition. This is quite impressive since it includes both intra and inter-class image distortion and recognition as well as discrimination.

a) CLASS 1

b) CLASS 2

c) CLASS 3

d) CLASS 4

FIGURE 1 Representative Ship Imagery (Broadside View Only) of the Ship Image Data Base Used in our Experiments.

TABIF 2. SDF Performance Test Results

SDF TYPE	TRAINING SET	CLASS 1 ERRORS	CLASS 2 ERRORS	CLASS 3 ERRORS	CLASS 4 ERRORS
ECP-1	1,6,10,15, 20,25	0	-	-	-
ECP-2	38,45,50, 55,60,65	-	0	-	-
MOCP-1 ($\Delta P \gtrless 0.5$)	1,6,10,15, 20,25	1	2	-	-
NRF	(6 Images Per Class)	2	0	3	-
Multi-Class MOCP-1 (±1)	(6 Images Per Class) (10°-340°)	2 (95%)	0 (100%)	3 (92%)	1 (97%)
Multi-Class MOCP-2 (±1)	(6 Images Per Class) (10°-260°)	0 (100%)	0 (100%)	4 (97%)	5 (96%)

Our NRF SDF experimental results, for a three-class pattern recognition problem, using output correlation peak projection values of (0,1,2) are shown next. As noted, only 5 errors occurred in this case. When this NRF SDF technique was extended to the case of four object classes, a considerable increase in the number of image classification errors resulted.

Encouraged by the above excellent results, we considered the full four-class pattern recognition problem using two SDFs (our multi-class MOCP SDF). We fabricated one set of SDFs using 9 images per class for training and a second set of SDFs using only 6 images per class for the training. As seen, a total of 6 errors (and 9 errors) out of 144 images resulted. These correspond to over 93% correct recognition of all multi-class objects with intra-class geometrical distortions present.

6. SUMMARY AND CONCLUSION

These initial results are most encouraging. They demonstrate the potential of this technique to achieve intra-class pattern recognition and inter-class discrimination in the presence of severe (3-D aspect) geometrical distortions. The data shown clearly indicates the ability of this technique to perform most excellently on images not present in the image training set. Our future work will address training set selection and the performance of the system in the presence of noise. Our unified SDF synthesis technique greatly simplifies the off-line calculations required. The fact that one can calculate the SDF off-line from training set images and then use the same SDF for similar object recognition problems in the future is of great significance in the practical application of such a technique.

ACKNOWLEDGEMENTS

The support of the Air Force Office of Scientific Research (Grant AFOSR 79-0091, Amendment D) and the internal research and development funds of General Dynamics for this research is greatly appreciated.

REFERENCES

1. D. Casasent and D. Psaltis, Proc. IEEE, 65, 143 (1977).
2. D. Casasent and D. Psaltis, Chapter in Progress in Optics, E. Wolf Ed., Vol. XVI, Holland Pub. Co., New York, pp. 284-351 (1978).
3. H.J. Caulfield and R. Haines, Applied Optics, 181 (1980).
4. H.J. Caulfield, Applied Optics, 3877 (1980).
5. C. Hester and D. Casasent, Applied Optics, 19, 1758 (1980).
6. C. Hester and D. Casasent, SPIE, 292, 25 (1981).
7. C. Hester and D. Casasent, SPIE, 302, 108 (1981).
8. J. Leger and S.H. Lee, Applied Optics, 274 (1982).
9. D. Casasent et al, SPIE, 360 (1982).
10. B. Braunecker et al, Applied Optics, 2746 (1979).

15. SHIFT-INVARIANT AND DISTORTION-INVARIANT OBJECT RECOGNITION

442-05

SHIFT-INVARIANT AND DISTORTION-INVARIANT OBJECT RECOGNITION

David Casasent and Vinod Sharma

Carnegie-Mellon University
Department of Electrical and Computer Engineering
Pittsburgh, Pennsylvania 15213

ABSTRACT

The problem of recognizing multiple objects in a highly cluttered background in the face of geometrical object distortions is addressed in this paper. A correlation architecture using a matched spatial filter of a synthetic discriminant function is employed to achieve the required performance. Synthesis of the synthetic discriminant function is discussed as is the initial performance obtained in the face of noise. Initial remarks are advanced on various methods to select the training set of images to use in this algorithm.

1. INTRODUCTION

In Section 2, the advantages and features of a correlator for extracting a signal or object in the presence of noise is briefly reviewed. A correlator is essential for our particular application since we require shift-invariance (the ability to recognize multiple objects and obtain high performance in the face of severe noise or clutter). However, correlators are well-known to perform quite poorly when geomtrical distortions (scale, rotation, etc.) are present between the input and reference object. To overcome this limitation and still retain the other advantageous features of a correlator, we employ synthetic discriminant functions (SDFs) and we form the matched spatial filter (MSF) from the SDF. The SDF is a linear combination of a training set of images of the object (e.g. different scaled, rotated or aspect views). This concept was first advanced in [1-3]. Other MSFs referred to as generalized matched filters [4] also exist. These assume a Fourier coefficient basis function set and appear to be more restrictive than the present filters we discuss, although no extensive tests have been performed on them.

In Section 3, we describe the algorithm for synthesis of five different types of SDFs. The SDF is synthesized off-line from training set data and is then used in a real-time correlator against test imagery not present in the training set. We follow the new unified SDF synthesis algorithm we briefly described in [5] earlier. In Section 4, we discuss our data base used and we present initial test results on the performance of these SDFs. These present results include new data on the performance of such a system in the presence of noise. Our earlier initial and less extensive results [6] did not consider the performance of such a system in the presence of noise. In Section 5, we advance initial remarks on two new techniques for training set selection and the superior performance obtained with the use of an organized rather than an ad hoc training set selection method. The bulk of the test data included in Section 4 uses an ad hoc training set selection technique (to allow more errors than one could otherwise achieve) and the use of a "no decision" threshold region (to stabilize statistical fluctuations in the results obtained).

2. CORRELATORS FOR PATTERN RECOGNITION

Correlators are very well-known [7] to be quite capable of extracting signals or objects from a cluttered background environment. A MSF correlator is optimal for Gaussian white noise backgrounds, but performs most admirably in the face of structured noise as well. Whitening of the image and noise data is possible and can allow optimal performance, but such measures do not appear to be necessary in the present problems and applications being considered. Correlators are also well-known to be linear shift-invariant systems. As such, they allow multiple objects to be recognized within the input field of view.

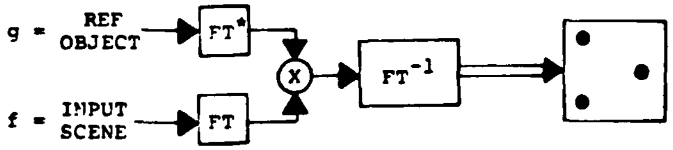


FIGURE 1. Block diagram of a Fourier Transform-Based Correlator Architecture.

442-05

The block diagram of the basic Fourier transform (FT) based correlator is shown in Figure 1. The system's output is

$$u(x,y) = \mathcal{F}^{-1}[G^{*}F] = g \circledast f \tag{1}$$

or the correlation of the two space functions g and f (G and F denote the Fourier transforms of the corresponding space functions). The processing gain (PG) of a correlator (the ratio of the output and input signal to noise ratio SNR_0/SNR_I) is related to the space bandwidth product (SBWP) of the MSF by

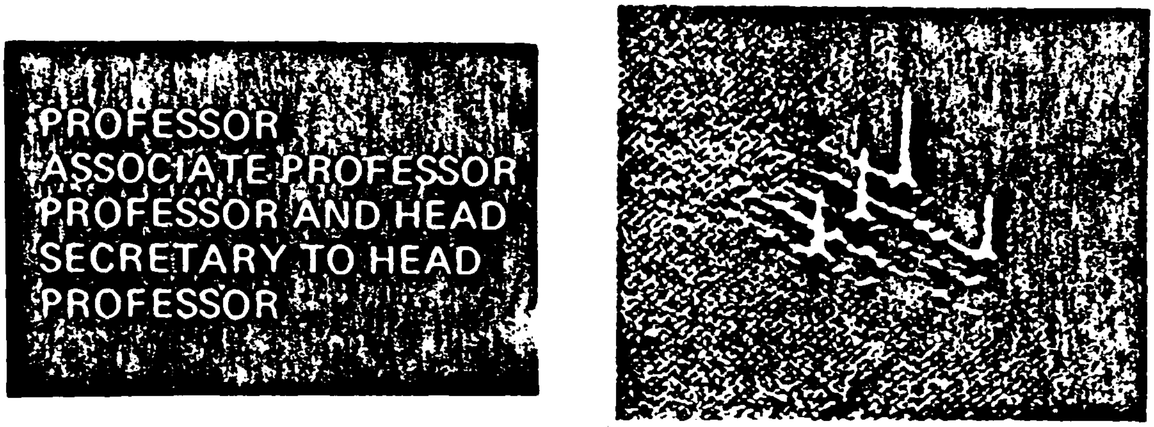
$$PG = (SBWP) = SNR_0/SNR_I. \tag{2}$$

This PG allows a correlator to extract objects from strong cluttered background. The shift-invariant, multiple-object recognition ability and the processing gain of a correlator are easily demonstrated by considering the general case of an input image f(x,y) that contains N occurrences of the reference object g at spatial locations (x_n,y_n) plus noise n(x,y), i.e.

$$f = \sum_n g(x-x_n,y-y_n) + n(x,y). \tag{3}$$

The output correlation for this case consists of N autocorrelations plus the cross correlation of the reference object and the noise, i.e.

$$OUT = f \circledast g = {}_n g \circledast g * \delta(x-x_n,y-y_n) + g \circledast n. \tag{4}$$



(a) Input Image

(b) Output Correlation of (a) and the word PROFESSOR.

FIGURE 2. Example of multiple-object shift-invariant correlation.

An example that vividly demonstrates this point is shown in Figure 2, where we show the input image f (a set of words with four occurrences of the word PROFESSOR) and the output correlation plane pattern obtained using a MSF of the word PROFESSOR. As seen, four correlation peaks are present and the location of each corresponds to the location of one of the different occurrences of the reference object (here the word PROFESSOR) in the input plane. All other input words in this input image are viewed as the noise n(x,y) in (3) and (4) for this case. As seen, the correlation output plane has a very low noise level. This is due to the large SBWP of the word PROFESSOR and the PG of a correlator.

3. SYNTHETIC DISCRIMINANT FUNCTION SYNTHESIS

The general concept of SDF synthesis is shown in the simple block diagram of Figure 3. An SDF is calculated off-line from training set images f_1, f_2, etc. and then on-line correlated with a real test image. The preprocessing box shown is generally not necessary as noted above. A hyperspace description of a SDF correlator is possible [2-3]. In such a formulation, each training set image is described by a vector point in hyperspace with the basis functions for this hyperspace being 2-D space functions. The conventional hyperspace pattern recognition concepts [8] of separating objects of different classes and clustering different object variations within a class still apply.

442 05

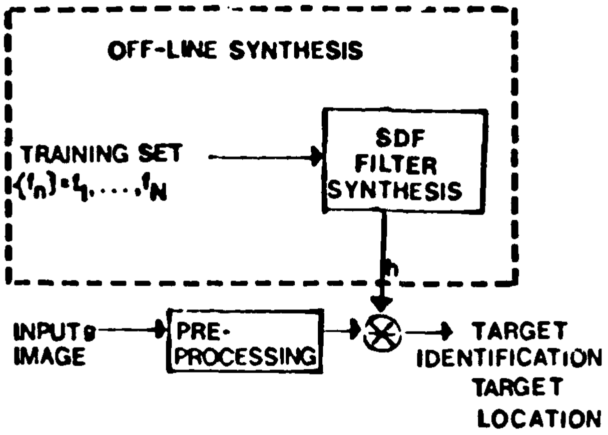


FIGURE 3. Block diagram to illustrate the synthetic discriminant function synthesis concept and its use in a correlator.

3.1 EQUAL CORRELATION PEAK SYNTHETIC DISCRIMINANT FUNCTION

Let us now describe SDF synthesis using our new general formulation technique [5]. We first consider an equal correlation peak (ECP) SDF. For this case, we assume that we wish to determine an MSF filter function h such that its correlation with any member of a set $\{f_n\}$ of objects of the class yields a constant output peak value such as one, i.e.

$$h \circledast f_n = 1. \tag{5}$$

To determine h to satisfy (5), we assume that h can be represented as a linear combination of the training set images f, i.e.

$$h = \sum_j e_j f_j. \tag{6}$$

For simplicity of notation, we denote all images and the MSF by vectors f or h. The training set images can be different scaled, rotated or aspect views of the object. We also represent the correlation operation by the simple vector dot or inner product, i.e. $h \circledast f = \underline{h} \cdot \underline{f}$. Substituting (6) into (5), grouping the dot product of f and defining the correlation matrix R to have elements r_{jn}, we obtain for (5)

$$h \cdot f_n = \sum_j (e_j f_j) \cdot f_n = \sum_j e_j (f_j \cdot f_n) = \sum_j e_j r_{jn} = 1 \tag{7}$$

Thus, our desired ECP filter is described by

$$\sum_j e_j r_{jn} = 1, \tag{8}$$

or in matrix-vector form by

$$\underline{\underline{R}}\,\underline{e} = \underline{u}, \tag{9}$$

where $\underline{u} = [1,1,\ldots,1]^T$ is the unit vector. The solution to (9) for the weighting coefficients $\underline{e}$ in (6) is

$$\underline{e} = \underline{\underline{R}}^{-1}\underline{u}, \tag{10}$$

where the e_j in (6) are the elements of the vector $\underline{e}$ in (10). Thus, to synthesize an ECP SDF, we form the correlation matrix of the training set of images, invert this matrix and multiply it by the unit vector $\underline{u}$. The resultant vector $\underline{e}$ defines our SDF filter function $\underline{h}$ in (6). To unify future notation, we write (9) as

$$\underline{\underline{R}}\,\underline{e} = [1,1,\ldots,1]^T. \tag{11}$$

3.2 MUTUAL ORTHOGONAL FUNCTION SDF (2-CLASS, ONE FILTER CASE)

We note that this ECP SDF MSF in (9), (10) or (11) is capable of intra-class recognition only. To extend this concept to the use of one filter for a two-class problem with inter-

412.05

class discrimination and intra-class recognition required, we consider two classes of objects $\{f_1\}$ and $\{f_2\}$, with N_1 and N_2 objects in the training set for each class respectively. The required SDF is now described by a linear weighted sum of all of the training set images, i.e.

$$h = \sum_{n=1}^{N_1+N_2} a_n f_n . \tag{12}$$

To determine the coefficients a_n in (12), we require the projections of any member f_{n1} of set 1 on h to be +1 and the projections of any member f_{n2} of set two of images to be -1 (any other suitable pair of constants may be used). By direct extension of the results in Section 3.1, this desired mutual orthogonal function (MOF) SDF is defined by the solution to the matrix-vector problem

$$\underline{R}\,\underline{a} = [1,\ldots,1,0,\ldots,0]^T = u_1 , \tag{13}$$

where $\underline{R}$ is now a $(N_1+N_2) \times (N_1+N_2)$ correlation matrix or is of size $2N \times 2N$ (when $N_1 = N_2 = N$). The solution and the SDF in (12) is defined by

$$\underline{a} = \underline{R}^{-1}\underline{u}_1 . \tag{14}$$

This single MOF SDF achieves intra and inter-class pattern recognition.

3.3 NONREDUNDANT FILTER SYNTHETIC DISCRIMINANT FUNCTIONS (ONE FILTER FOR MULTI-CLASSES)

This type of SDF is referred to as a nonredundant filter (NRF) SDF. It is a rather direct extension of the MOF SDF in (13). Here, we simply require the correlation output peak intensity to equal a different constant value (e.g., 1, 2, 3, etc.) for each object class. We consider a 3-class pattern recognition problem. The SDF is now described by

$$h = \sum_{n=1}^{N_1+N_2+N_3} a_n f_n , \tag{15}$$

where the summation is over all $N_1+N_2+N_3$ members of the three training sets for the three object classes. For this SDF, the output correlation peak value defines the input object class. The coefficients a_n that define this NRF SDF are the solution of the matrix-vector problem

$$\underline{R}\,\underline{a} = [1,\ldots,1,0,\ldots,2,3,\ldots,3]^T = u_2 . \tag{16}$$

That is,

$$\underline{a} = \underline{R}^{-1}\underline{u}_2 , \tag{17}$$

where now $\underline{R}$ is a $3N \times 3N$ correlation matrix (when $N_1 = N_2 = N_3 = N$).

3.4 MULTI-CLASS MOF SDF (ONE FILTER PER OBJECT CLASS)

In some image pattern recognition problems, the constraint placed on the MOF or the NRF SDFs in Sections 3.2 and 3.3 may be too stringent and hence performance on test data may be degraded. In this case, M separate SDFs can be produced, each of which gives a constant output value of 1 for a different object class and a 0 output for all other object classes. For a 3-class pattern recognition problem, we consider the synthesis of three SDF filters

$$h_1 = \sum_n a_n f_n ,\quad h_2 = \sum_n b_n f_n ,\ \text{and}\ h_3 = \sum_n c_n f_n , \tag{18}$$

where all summations in (18) are over all $N_1 + N_2 + N_3 = 3N$ image training sets of data.

The algorithm for synthesis of these 3 SDFs is similar to before, i.e.

$$\underline{R}\,\underline{a} = \underline{u}_1 = [1\cdots1,0\cdots0,0\cdots0]^T ,\quad \underline{a} = \underline{R}^{-1}\underline{u}_1 \Longrightarrow \underline{h}_1 , \tag{19a}$$

$$\underline{R}\,\underline{b} = \underline{u}_2 = [0\cdots0,1\cdots1,0\cdots0]^T ,\quad \underline{b} = \underline{R}^{-1}\underline{u}_2 \Longrightarrow \underline{h}_2 , \tag{19b}$$

$$\underline{R}\,\underline{c} = \underline{u}_3 = [0\cdots0,0\cdots0,1\cdots1]^T ,\quad \underline{c} = \underline{R}^{-1}\underline{u}_3 \Longrightarrow \underline{h}_3 . \tag{19c}$$

In this case, the correlation matrix is $3N \times 3N$ (for $N_1 = N_2 = N_3 = N$). By inspection of (19), we see that h_1 is selected to provide an output of 1 for the first N_1 training set images (i.e. class-one objects) and "0" for the other images (classes two and three objects).

4-2 05

Conversely, SDF h_2 is defined to produce a 1 output for the N_2 training set images in class two and a 0 output for the other classes. The class three SDF h_3 is described similarly.

3.5 MULTI-CLASS MOF SDF (N FILTERS FOR 2^N CLASS RECOGNITION PROBLEM)

For this SDF, we consider another multi-class type of SDF formulation that is between the single SDF cases considered in Sections 3.2 and 3.3 and the multiple filter example considered in Section 3.4. This particular application is most attractive for multi-class pattern recognition problems in which very many multiple classes of data must be considered. Here, we consider an M-class pattern recognition problem. We assume binary correlation peak threshold outputs and the use of N SDFs (where $2^N \geq M$). For a four-class problem, we thus consider the use of two SDFs, h_1 and h_2. Each input image is projected onto both SDFs and the two correlation outputs are used to form a 2-bit digital word which defines by the truth table in Table 1 below in which of the four input classes $\{f_1\}$ through $\{f_4\}$ the input object lies.

TABLE 1: Truth Table for Correlation Peak Output Decoding Using a Multi-Class Mutual Orthogonal Function Synthetic Discriminant Function with N Filters for a $2^N = M$ Class Pattern Recognition Problem. $M = 4$ and $N = 2$ Case Shown.

INPUTS	OUTPUTS	
	h_1	h_2
$\{f_1\}$	0	0
$\{f_2\}$	0	1
$\{f_3\}$	1	0
$\{f_4\}$	1	1

The algorithm for calculation of the two filters is

$$h_1 = \sum_n a_n f_n, \quad h_2 = \sum_n b_n f_n. \tag{20}$$

The solution for the vectors $\underline{a}$ and $\underline{b}$ that define these two filter functions follows by extension of our previous SDFs to be

$$\begin{bmatrix} \\ R \\ \\ \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} . \tag{21}$$

3.6 UNIFIED FORMULATION

By inspection of (11), (13), (16), (19) and (21), we find that the synthesis of all five SDFs can be described by the same basic matrix-vector equation $\underline{R}\,\underline{a} = \underline{u}_n$, where the size of the correlation matrix and the specific form of the vector $\underline{u}_n$ are different for each filter function and for each class of problem being addressed. This is especially attractive since all SDF filter synthesis is performed off-line. Thus, a simple digital matrix-vector routine can be used for all cases. Because this SDF algorithm relies on a training set, the same SDF filter will perform adequately until the structure of the object to be recognized changes significantly (due to new designs, etc.) or until rejection of new alternate objects is required (this depends on the specific details of each application). The unified formulation in which the solution of the same basic matrix-vector equation is required makes this generalized formulation of SDF synthesis most attractive [5] for many practical reasons and applications. In all cases, if $\underline{R}$ is not invertible, a generalized inverse is used.

4. INITIAL TEST AND PERFORMANCE RESULTS

The data base we used in our test and evaluation of these SDF algorithms consisted of four different types of ships. For each ship, 36 images were available at 10° intervals from a 0° depression angle. In Figure 4, the broadside views of each ship are shown. In Figure 5, the views at several different aspect angles are shown for one of the ships. From this data, one can clearly see the severe magnitude of the intra and inter-class recognition problem being considered. A human cannot correctly classify many of the different ship aspect views from such imagery. Quite simply, several of the ships look essentially identical from several different aspect views. Recognition of the ship class from the bow or stern views alone (e.g., Figure 5d) is clearly quite difficult, even for a human observer.

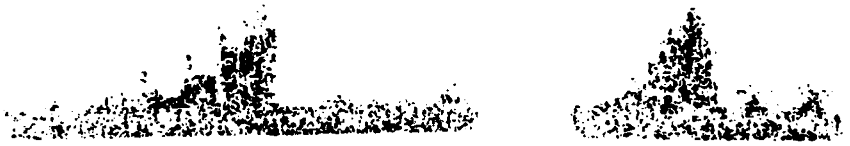
a) CLASS 1

c) CLASS 3

b) CLASS 2

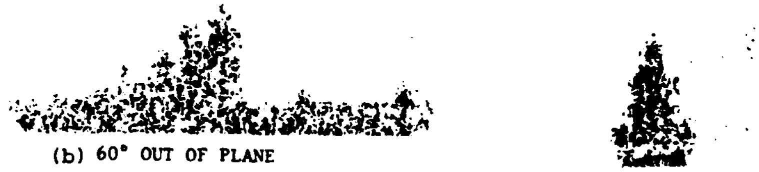
d) CLASS 4

FIGURE 4. Representative ship imagery (broadside view only) of the ship image data base used in our experiments.



(a) CLASS 1 - 90°

(c) 30°



(b) 60° OUT OF PLANE

(d) 0° ANGLE

FIGURE 5. Selected aspect views of one ship.

The results of our initial experiments on four of the different types of SDFs described in Section 3 are summarized in Table 2. As seen, the results obtained are quite excellent (over 95% correct recognition was obtained in all cases). The data in Table 2 was obtained with the hull present and was obtained for binarized imagery properly thresholded. It should be noted that in no test were more than 9 images per class used for training and that in most instances, only 6 images per class were used for training and for SDF synthesis.

In Table 3, new results obtained with different levels of input noise added to the test imagery and to the training set imagery are shown. To stabilize the results obtained and to decrease fluctuations in these results due to statistical variations, any correlation peak value that was within ±0.03 of the threshold value of 0.5 was classified as a "no decision" image. The number of no decisions obtained for this data are entered in parentheses, together with the number of errors obtained for each class of object. The hull of the ship was again retained in this data and only 6-9 images per class were used for training. As seen from the last column of Table 3, the number of correct recognitions was very constant (127 or 128 correct recognitions out of 144 test images) with only 5-9 errors (and over 93% correct recognition in all cases). The constancy of the number of correct recognitions as the input noise was increased (down to an $SNR_I = 6$, corresponding to an input noise with a variance = 0.04) and its invariance as different amounts of noise were added to the training set itself indicates the robustness of this SDF filter to noise. Since this is a correlator, we expect such performance as noted in Section 2. Tests in which noise was present in the background only and in both the background and the target were also conducted. Results very

442 05

similar to those obtained in Table 3 again resulted for this case.

TABLE 2. Initial Noise-Free Performance Obtained with 4 of the Different Types of SDFs.

TYPE	TRAINING SET	NUMBER OF ERRORS	PERCENT CORRECT RECOGNITION
ECP INTRA-CLASS	6 IMAGES CLASS 1	0 OUT OF 36	100%
2-CLASS OUT = ±1	6 IMAGES PER CLASS	3 OUT OF 72	96%
3-CLASS OUT = 1,2,3	6 IMAGES PER CLASS	5 OUT OF 108	95%
4-CLASS TWO FILTERS (1,1),(1,-1),(-1,1),(-1,-1)	9 IMAGES PER CLASS	7 OUT OF 144	95%

TABLE 3. Synthetic Discriminant Function Performance in the Presence of Noise.

SNR$_1$		NUMBER OF ERRORS (NO DECISION)				NUMBER ERRORS	NUMBER CORRECT
TRAIN	TEST	CLASS 1	CLASS 2	CLASS 3	CLASS 4		
∞	∞	0 (1)	0 (0)	4 (1)	1 (3)	5	134
∞	25	0 (5)	0 (1)	4 (1)	1 (4)	5	128
∞	10	4 (3)	0 (1)	4 (1)	1 (2)	9	128
∞	6	4 (3)	0 (1)	4 (1)	0 (3)	8	128
25	25	0 (4)	0 (5)	4 (2)	1 (1)	5	127
25	10	1 (3)	0 (5)	4 (3)	0 (1)	5	127

5. TRAINING SET SELECTION

The six training set images per class used in Table 2 were 50° apart. Those in Table 3 were 30° apart and from one side of the ship only. This ad hoc training set selection approach is quite compatible with the intuition associated with such a pattern recognition problem (i.e., ship images appear in general to contain little new information if viewed from one side rather than the other). However, in other cases, more automated training set selection techniques are required. Two such techniques are briefly described and initial test and performance data obtained using them are presented in this section.

The purpose of the training set for each object class is to represent each object type by the most valid statistical representation from the available imagery. The SDF algorithm itself provides the inter-class discrimination required. Thus, our concern in the automatic selection of a training set should address selecting a small set of images from the available image data base such that they represent a sufficiently valid statistical representation of each object. We have considered two ways to achieve this: a correlation-subspace and an eigenvector subspace selection technique.

In the first technique, we initially pick one image x_{i1} (e.g., the bow view of a ship). As the second image, we select that image from the remaining ones which gives the minimum correlation with the prior image. Such an image contains the most new information and is the best one to select for our training set. We continue this technique and at each successive step, we pick as our new image the one with the minimum projection on the sum of the previously chosen training set images. With proper image normalization and with the use of a modified Gramm-Schmidt technique, we can thus select the P best training set images to use. The value of the correlation between a new image and the sum of the prior training set images provides a measure of the new data added by a given additional training set image. This procedure is used for each object class separately.

In the second technique, we form the correlation matrix R for each object class and calculate its eigenvalues and eigenvectors. We select the dominant eigenvectors and calculate the projections of all images onto these dominant eigenvectors. Those images with the largest projections best represent the object class and are thus included in the training set. The eigenvalue is used to determine the number of images to be selected for each dominant eigenvector. The value of the projection of a new image on a new dominant eigenvector determines the additional information present if this image is included in the training set.

The results obtained with this technique are summarized in Table 4 and compared to those obtained earlier using our ad hoc training set selection technique. As seen, our ad hoc selection of 6 images gave poorer performance than did both of the new methods noted in this section. Essentially perfect performance (0 or 1 error out of 144 images) was obtained using our new training set selection methods. As the last entry in Table 4 shows, a successful single SDF for the 4-class problem was realized using such training set methods. The performance of these new SDFs in the presence of noise was found to be comparable to those obtained with the previous data. An average of only 4 or 5 errors was obtained (versus 8-10 errors for the cases noted in Tables 2 and 3).

TABLE 4. Comparison of the Performance of Synthetic Discriminant Functions Using Ad Hoc and Automated Techniques (Correlation Subspace and Eigenvector (e-v) Subspace) to Select the Image Training Sets.

SDF TYPE	TRAINING SET (IMAGES PER CLASS	NUMBER OF ERRORS				TOTAL NUMBER ERRORS	P_c
		CLASS 1	CLASS 2	CLASS 3	CLASS 4		
(1) MULTI-MOF (HULL)	ONE SIDE	1	0	5	4	10	93%
(2) MULTI-MOF (NO HULL)	ONE SIDE	0	0	0	9	9	94%
(3) MULTI-MOF (NO HULL)	e-v SUBSPACE	0	0	0	1	1	99.3%
(4) MULTI-MOF (NO HULL)	CORREL SUBSPACE	0	0	0	0	0	100%
(5) NPF (NO HULL)	CORREL SUBSPACE	0 (0)	6 (2)	0 (1)	0 (0)	6 (135)	94%

6. SUMMARY AND CONCLUSION

6. SUMMARY AND CONCLUSION

In this paper, we have described a new technique to recognize multiple classes of objects in the face of geometrical distortions. Our technique uses a correlator and thus allows multiple object recognition and excellent performance in the presence of structured noise and clutter. The latter features result from the use of a correlator and the shift-invariance and processing gain of such a pattern recognition architecture. The ability to recognize distorted object views and to discriminate between different object classes is achieved by our new matched spatial filter technique using a matched spatial filter of a synthetic discriminant function.

Five different synthetic discriminant functions have been described and a general unified synthesis procedure to form each off-line using a training set of images has been advanced. Initial simulation results showed excellent performance and showed the robustness of this algorithm in the presence of noise. New techniques to select the training set for such a system were also advanced and initial performance results were obtained and were compared to those obtained for other ad hoc training set selection techniques. Superior performance resulted when these new training set-based filters were employed.

ACKNOWLEDGEMENTS

The support of much of the initial phase of this research by the Air Force Office of Scientific Research on Grant 79-0091 and the recent support of this work by the Internal Research and Development Funds of General Dynamics-Pomona is gratefully acknowledged.

REFERENCES

1. C. Hester and D. Casasent, Applied Optics, 19, 1758-61, June 1980.
2. C. Hester and D. Casasent, SPIE, 292, 25-33, August 1981.
3. C. Hester and D. Casasent, SPIE, 302, 108-116, August 1981.
4. H. Caulfield and R. Haimes, Applied Optics, 19, 181 (1980).

442 05

5. D. Casasent, B. Kumar, V. Sharma, *SPIE*, *360*, 136-142, August 1982.
6. D. Casasent and V. Sharma, *Proc. IOCC'83*, Boston, Massachusetts, April 1983, IEEE Cat. No. CH1880-4/83, *SPIE Vol. 422*.
7. E.L. O'Neill, *IRE, Trans. Info. Theory*, *IT-2*, 56 (1956).
8. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley and Sons, New York (1973).

16. PUBLICATIONS AND PRESENTATIONS

16.1 PUBLICATIONS (AFOSR SUPPORTED, 1979-DATE)

Publications from 30 September 1979 - 30 September 1980 on work performed under AFOSR-79-0091 are listed in Section 16.1.1. Publications during 30 September 1980 - 30 September 1981 follow in Section 16.1.2, and publications in FY82 continue in Section 16.1.3. Our new publications from September 1982-September 1983 follow in Section 16.1.4. A list of presentations at conferences, companies, and seminars on our AFOSR research conducted during the prior year then follow.

16.1.1 PUBLISHED PAPERS UNDER AFOSR SUPPORT (30 SEPT. 1979 - 30 SEPT. 1980)

1. "Photo-DKDP Light Valve in Optical Data Processing", Applied Optics, 18, 3307-3314, October 1979 (Casasent, Luu).
2. "Coherent Optical Pattern Recognition", Nikkei Electronics, 150-181, October 1979 (in Japanese) (Casasent).
3. "Optical Data Processing for Advanced Missile Guidance Needs", AIAA, October 1979 (Casasent).
4. "Spread Spectrum Optical Signal Processors", Proc. EOSD, 333-342, October 1979 (Casasent, Psaltis).
5. "Space Blur Bandwidth Product in Correlator Performance Evaluation", JOSA, 70, 103-110, January 1980 (Kumar, Casasent).
6. "Optical Image Processing", EOSD, Tokyo, January 1980 (in Japanese) (Casasent).
7. "Optical Signal Processing", EOSD, Tokyo, January 1980 (in Japanese) (Casasent).
8. "Beyond Matched Filtering", Opt. Engr., 19, 152-156, March 1980 (Caulfield et al).
9. "Multivariant Technique for Multi-Class Pattern Recognition", Applied Optics, 19, 1758-1761, June 1980 (Psaltis, Casasent).
10. "Optical Fourier Transform Techniques for Advanced Fourier Spectroscopy", Applied Optics, 19, 2034-2037, June 1980 (Casasent, Psaltis).
11. "Nonlinear t-E Curve Effects in an Optical Correlator", Opt. Commun., 34, 4-6, July 1980 (Kumar, Casasent).

12. "Correlation of Images with Random Contrast Reversals", SPIE, 238, 156-165, July 1980 (Barniv, Mostafavi, Casasent).

13. "A Laser Diode Lensless MSF-HOE Correlator", Applied Optics, 19, 2653-2654, August 1980 (Caimi et al).

16.1.2 PUBLISHED PAPERS UNDER AFOSR SUPPORT (30 SEPT. 1980 - 30 SEPT. 1981)

14. "Hybrid Processor to Compute Invariant Moments for Pattern Recognition", Opt. Lett., 5, 395-397, September 1980 (Casasent, Psaltis).

15. "Optical Word Recognition, Case Study in Coherent Optical Pattern Recognition", Opt. Engr., 19, 716-721, September 1980 (Casasent et al).

16. "Lensless Matched Spatial Filter Correlator Experiments", Opt. Commun., 34, 311-315, September 1980 (M. Shen et al).

17. "HOE/Lensless Matched Spatial Filter Correlator Experiments", Opt. Commun., 34, 316-320, September 1980 (M. Shen et al).

18. "A Laser Diode/Lensless MSF Optical Pattern Recognition System", EOSD, 46-52, November 1980 (Casasent et al).

19. "Optical Pattern Recognition: Matched Spatial Filter Processors", EOSD, 33-39, November 1980 (Casasent).

20. "Optical Pattern Recognition: Beyond Matched Spatial Filtering", EOSD, 39-47, March 1981 (Casasent).

21. "Pattern Recognition: A Review", IEEE Spectrum, 28-33, March 1981 (Casasent).

22. "Processing Flexibility by Hybrid Optical/Digital Techniques", Proc. Workshop of Future Directions in Optical Data Processing, Texax Tech. Rept., 1 March 1981, 17-23 (Casasent, Kumar).

23. "Beyond Holographic Matched Filtering", Israel Journal of Technology, 18, 255-260, March 1981 (Casasent).

24. "Binarization Effects in a Correlator with Noisy Input Data", Applied Optics, 20, 1433-1438, April 1981 (Kumar, Casasent).

25. "Correlation of Images with Random Contrast Reversals", SPIE, 238, 156-165, July 1980 (Barniv, Mostafavi, Casasent).

26. "Image Quality Effects in Optical Correlators", SPIE, 310, 183-192, August 1981 (Casasent, Eiva, Kumar).

27. "Multisensor Image Registration: Experimental Verification", SPIE, 292, 160-171, August 1981 (Barniv, Casasent).

28. "Intra-Class IR Tank Pattern Recognition Using SDFs", SPIE, 292, 25-33, August 1981 (Hester, Casasent).

29. "Inter-Class Discrimination Using SDFs", SPIE, 302, 108-116, August 1981 (Hester, Casasent).

16.1.3 PUBLISHED PAPERS UNDER AFOSR SUPPORT (30 SEPT. 1981 - 30 SEPT. 1982)

30. "An Iterative Optical Processor: Selective Survey of Operations Achievable", Proceedings NASA Langley Conference on Optical Information Processing, Publication 2207, August 1981, 105-118 (Casasent, Neuman).

31. "A Review of Optical Signal Processing", IEEE Commun., 40-48, September 1981 (Casasent).

32. "Optical Signal Processing II: Applications, Systems and New Techniques", EOSD, 41-47, September 1981 (Casasent).

33. "The Soviet Priz Spatial Light Modulator", Applied Optics, 20, 3090-3092, September 1981 (Casasent, Caimi, Khomenko).

34. "A Laser Diode/HOE Pattern Recognition System", Acta Optica Sinica, 1, 401-410, September 1981 (Casasent et al).

35. "Eigenvector Determination by Iterative Optical Methods", Applied Optics, 20, 3707-3710, November 1981 (Kumar, Casasent).

36. "A New Soviet BSO Light Modulator for Optical Data Processing", Proc. EOSD, 297-303, November 1981 (Casasent, Caimi).

37. "A Correlator for Optimum Two-Class Discrimination", Proc. EOSD, 321-330, November 1981 (Casasent et al).

38. "Test and Evaluation of the Soviet Prom and Priz Spatial Light Modulators", Applied Optics, 20, 4215-4220, December 1981 (Casasent, Caimi, Khomenko).

39. "A Microprocessor-Based Fiber-Optic Iterative Optical Processor", Applied Optics, 21, 147-152, January 1982 (Carlotto, Casasent).

40. "Principal Component Imagery for Statistical Pattern Recognition Correlators", Opt. Engr., 21, 43-47, January/February 1982 (Kumar, Casasent).

41. "Adaptive Phased Array Radar Processing Using an Optical Matrix-Vector Processor", SPIE, 341, May 1982 (Casasent, Carlotto).

42. "New Research in Holographic Pattern Recognition", Proc. SPIE, 353, 6-11, August 1982 (Casasent).

43. "Synthetic Discriminant Functions for 3-D Object Recognition", Proc. SPIE, 360, 136-142, August 1982 (Casasent, Kumar, Sharma).

44. "Multidimensional Adaptive Radar Array Processing Using an Iterative Optical Matrix-Vector Processor", Opt. Engr., 21, 814-821, September 1982 (Casasent, Carlotto).

16.1.4 PUBLISHED PAPERS UNDER AFOSR SUPPORT (30 SEPT. 1982 - 30 SEPT. 1983)

45. "Advanced Acousto-Optic Signal Processors", Proc. SPIE, 352, 50-58, August 1982 (Casasent).

46. "A Fisher Discriminant Approach to Distortion-Invariant Pattern Recognition Using Autocorrelations", Lasers and Electro-Optics, 34, 18-23, September 1982 (Casasent, Chang).

47. "Realization of a Sobel Operator by Coherent Optical Techniques", Lasers and Electro-Optics, 34, 24-30, September 1982 (Chen, Casasent).

48. "Applications of the Priz Light Modulator", Applied Optics, 21, 3846-3854, November 1982 (Casasent, Caimi, Petrov, Khomenko).

49. "Frequency-Multiplexed and Pipelined Iterative Optical Systolic Array Processors", Applied Optics, 22, 115-124, January 1983 (Casasent, Jackson, Neuman).

50. "Optical Linear Algebra", SPIE, 388, January 1983 (Casasent, Ghosh).

51. "Nonlinear Local Image Preprocessing Using Coherent Optical Techniques", Applied Optics, 22, 808-814, March 1983 (Casasent, Chen).

52. "Developments in Acousto-Optic Signal Processings", Trends and Perspectives in Signal Processing", 3, 1-6, June 1983 (Casasent).

53. "LU and Cholesky Decomposition on an Optical Systolic Array Processor", Optics Communications, 46, 270-273, July 1983 (Casasent, Ghosh).

54. "Guidelines for Efficient Use of Optical Systolic Array Processors", IOCC Conference, Boston, Massachusetts, April 6-8, 1983, IEEE Cat. No. CH1880-4/83, SPIE Vol. 442, p. 209-213, (Casasent).

55. "Performance of Synthetic Discriminant Functions for Infrared Ship Classification", IOCC Conference, Boston, Massachusetts, April 6-8, 1983, IEEE Cat. No. CH1880-4/83, SPIE Vol. 422, p. 193-196 (Casasent, Sharma).

56. "Generalized Chord Transformation for Distortion-Invariant Optical Pattern Recognition", Applied Optics, 22, 2087-2094, March 1983 (Casasent, Chang).

57. "Recent Advances in Optical Signal Processing", CLEO Conference, May 17-20, 1983, Baltimore, Maryland (Casasent).

58. "Shift-Invariant and Distortion-Invariant Object Recognition", SPIE, 442, August 1983 (Casasent, Sharma).

16.2 SEMINARS, CONFERENCE, ETC. PRESENTATIONS OF AFOSR RESEARCH (1 SEPTEMBER 1982-1 SEPTEMBER 1983)

September 1982:

1. Night Vision and Electro Optics Laboratory, Ft. Belvoir, Virginia - "Optical Pattern Recognition".

2. Night Vision and Electro Optics Laboratory, Ft. Belvoir, Virginia - "Optical Signal Processing".

3. LIA Conference, Boston, Massachusetts - "Realization of a Sobel Operator by Coherent Optical Techniques".

4. LIA Conference, Boston, Massachusetts - "A Fisher Discriminant Approach to Distortion-Invariant Pattern Recognition Using Autocorrelations".

5. Ames Building, Washington, D.C. - "Optical Matrix-Vector and Optical Systolic Array Processors"

October 1982:

6. Carnegie-Mellon University, Pittsburgh, Pennsylvania, Sophomore Seminar - "Optical Data Processing at Carnegie-Mellon University".

7. TASC, McLean, Virginia - "General Purpose Optical Processors".

November 1982:

8. Hughes Corporation, Conoga Park, California - "Moment-Based Hybrid Optical/Digital Pattern Recognition".

9. Hughes Corporation, Conoga Park, California - "SDFs for Distortion-Invariant Recognition of Multiple Targets in Clutter".

December 1982:

10. Eaton Corporation, Milwaukee, Wisconsin - "Optical Processing Techniques for Industrial Inspection, Automation and Pattern Recognition".

January 1983:

11. SPIE, Los Angeles, California - "Optical Linear Algebra" (Anjan Ghosh).

April 1983:

12. IOCC Conference, Boston, Massachusetts - "Guidelines for Efficient Utilization of Optical Systolic Array Processors".

13. IEEE, Boston, Massachusetts - "Optical Data Processing for Target Recognition and Classification".

14. IOCC Conference, Boston, Massachusetts - "Performance of Synthetic Discriminant Functions for Infrared Ship Classification".

May 1983:

15. Carnegie-Mellon University, Pittsburgh, Pennsylvania, Freshman Seminar - "Optical Data Processing".

16. Westinghouse Corporation, Baltimore, Maryland - "Optical Systolic Processors for Adaptive Phased Array Radar".

17. Baltimore, Maryland - "Optical Data Processing Research at Carnegie-Mellon University".

18. CLEO Conference, Baltimore, Maryland - "Advanced Optical Data Processing Research".

19. Rosslyn, Virginia, "Advanced Object Pattern Recognition in Man-Made Clutter".

June 1983:

20. AGARD/NATO, Bonn, Germany - "Fundamentals of Optical Pattern Recognition".

21. AGARD/NATO, Bonn, Germany - "Advanced Optical Pattern Recognition Techniques".

22. AGARD/NATO, Bonn, Germany - "Optical Matrix-Vector and Systolic Array Processing".

23. Lebedev Institute, Moscow, USSR - "Optical Image, Signal and Information Processing".

24. Latvian Academy of Sciences, Riga, Latvia, USSR - "Optical Image and Data Processing".

25. Institute of Physics, Belyo, Minsk, USSR, "Optical Image and Information Processing".

26. Shuvalov Institute, Leningrad, USSR - "Optical Image, Signal and Data Processing".

27. Ioffe Institute, Leningrad, USSR - "Optical Image and Data Processing".

28. University of Kosova, Pristina, Yugoslavia - "Optical Image, Signal and Data Processing".

July 1983:

29. Instituto de Optiko, Madrid, Spain - "Optical Image and Data Processing".

August 1983:

30. SPIE, San Diego, California - "Direct and Indirect Optical Solutions to Linear Algebraic Equations: Error Source Modeling".

31. SPIE, San Diego, California - "Linear Algebra Techniques for Pattern Recognition: Feature Extraction Case Studies".

32. SPIE, San Diego, California - "Shift-Invariant and Distortion-Invariant Object Recognition".

16.3 THESES SUPPORTED BY AFOSR FUNDING (SEPTEMBER 1980 - SEPTEMBER 1983)

1. Hiroyasu Murakami, M.S. Dissertation, "Matched Filter Statistical Correlator" (February 1981).

2. Saulius Eiva, M.S. Dissertation, "Image Quality Effects in Optical Correlators" (May 1981).

3. Charles Hester, PhD Dissertation, "Synthetic Filters for Multi-Class Pattern Recognition" (May 1981).

4. Yair Barniv, PhD Dissertation, "Multi-Sensor Image Registration" (May 1981).

5. Mark Carlotto, PhD Dissertation, "Iterative Electro-Optic Matrix Processor" (May 1981).

6. Andrew Sexton, M.S. Dissertation, "Digital Analysis of Space-Variant Optical Processors" (July 1981).

7. Bernard Szymanski, M.S. Dissertation, "A Computer-Controlled Film Recorder for Optical Processing" (July 1983).

8. John Lycas, M.S. Dissertation, "An Optical Implementation of the Extended Kalman Filter" (September 1983).

9. Vinod Sharma, "Synthetic Discriminant Functions" (PhD Expected in December 1983).

10. Warren Allmond, "New Holographic Optical Elements and Architectures" (M.S. Expected in 1984).

11. Eugene Pochapsky, "Digital Preprocessing and Simulation for Optical Pattern Recognition" (M.S. Expected in 1984).

12. R. Lee Cheatham, "Optical Moment-Based Pattern Recognition" (PhD Expected in 1984).

13. Anjan Ghosh, "Linear Algebra System Performance and Analysis" (PhD Expected in 1984).
14. G. Kashipati, "Synthetic Discriminant Functions" (M.S. Expected in 1984).
15. Bruce Thomas, "Moments for Distortion Measurement" (M.S. Expected in 1984).
16. William Rozzi, "Synthetic Discriminant Functions" (M.S. Expected in 1984).
17. Wen-Thong Chang, "Optical Pattern Recognition" (PhD Expected in 1985).

16.4 PATENT DISCLOSURES (SEPTEMBER 1980 - SEPTEMBER 1983)

1. Multiple-Invariant Space-Variant Pattern Recognition System.
2. Pattern Recognition by Invariant Moments.
3. Synthetic Discriminant Functions for Multi-Class Pattern Recognition.
4. Equalization and Coherence Measure Correlator.
5. Multi-Variant Technique for Multi-Class Pattern Recognition.

ATE

LMED

8