2

A RAND NOTE

A Structural Approach to the Photonic Processor

Deborah Jackson

RAND

The research described in this report was supported by RAND as part of its program of public service.

Published 1991 by RAND
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138

A RAND NOTE

N-3399-RC

A Structural Approach to the Photonic Processor

Deborah Jackson

RAND

PREFACE

This Note is the product of an overview study on the field of photonics. Photonics is currently in the highest priority category of the DoD's critical technologies list because it is perceived as a potential solution to the bottleneck of serial processing (i.e., the *von Neumann bottleneck* in computer processors). These bottlenecks are occurring in critical military applications, such as automatic target recognition, clutter rejection in infrared search and track (IRST) applications, vision-assisted piloting tasks in unmanned robotic vehicles (e.g., Remotely Piloted Vehicles, Unmanned Aerial Vehicles, or Unmanned Underwater Vehicles), and imaging Identification Friend or Foe.

The study's intended purpose was to investigate how and where photonics could enhance current and future military systems. Funded by RAND in the summer of 1989, this study had two tasks. The goal of task 1 was to search the literature and define the types of applications in which photonic processing has demonstrated significant performance advantages over digital electronic processors. Task 2 was to identify how and where photonics might play a useful role in future military applications as the technology matures.

For the purposes of this work, it was necessary to devise a framework within which existing and potential photonics applications could be viewed. The initial insight came with the recognition that any problem involving the manipulation of an *image-formatted data base* inherently stresses the processing capabilities of serial processors. Parallel processors offer hope of circumventing this; the intrinsic parallelism of optics, in fact, implies that the potential exists for developing an optical system that is capable of extracting the spatial information content from these data bases. Consequently, this document proposes a procedure for evolving such a processor. While the basic methodology is adopted from that originally used to develop the digital electronic processor, both the architecture and design function of the photonic processor presented here differ from other current digital and optical processor designs.

This document will be of particular interest to those who need to extract information from image-formatted data bases in realtime (i.e., < 1 msec.) and who are currently limited by the von Neumann bottleneck in electronic processors. The author assumes that the reader is comfortable with basic concepts in geometrical optics and has a cursory knowledge of how computers are organized.

SUMMARY

Many of the processing limitations (or bottlenecks) occurring in military computation problems, automatic target recognition, clutter rejection in infrared search and track (IRST) applications, vision-assisted piloting tasks in unmanned robotic vehicles (e.g., RPV's, UAV's, or UUV's), and imaging IFF, among others, occur when trying to extract information from image-formatted data bases. These problems are usually attributed to the serial processing limitation, or *von Neumann bottleneck*. The photonic processor is perceived as a promising solution to the serial processor's limitations and is thus in the highest priority category of the DoD's critical technology list.

But developing a photonic processor has been slow. This Note argues that since the technologies of optics and electronics are so different, the most advantageous architecture for optical processors must necessarily differ from that for electronic processors. The eventual choice of architecture is very dependent on the applications class of problems for which the processor is designed. For image-formatted problems, the familiar but inappropriate digital electronic architecture is clearly a limiting factor. Essentially, there is a class of problems that the properly designed optical (image format) processor can solve with ease, and another class of problems that electronic (serial format) processors can solve with ease. These two different approaches to problem solving are complementary rather than competitive. The photonic processor should be designed to take advantage of the best properties of currently available electronic and optical technologies.

Although the architecture of the electronic processor is not suitable for image format problems, the methodology used to develop it can still be applied to evolve other types of architectures such as the photonic one described here. In this Note, the author proposes adopting the methodology previously used in developing the digital electronic processor to evolve an architecture that is customized to physical optical properties rather than one that is optimized for physical electronic properties.

This methodology includes the following steps: (1) Define the class of problems to be solved: those containing image-formatted data bases. (2) Define the *design functions* of the photonic processor as the extraction of spatial information content from image-formatted data bases. (3) Develop "optical circuits" or other block diagrams illustrating the applications. (4) Decompose optical circuits into basic optical operational functions. (5) Determine the physical design limits of the available technology base. (6) Conceptualize the hardware design for the photonic processor.

Having discussed the procedure for developing a photonic processor, which is design optimized for extracting information from image-formatted data bases, this document then explores what form an optical processor might take for several different navigation and tracking applications. More broadly, it lays out a methodology for exploring the current photonic technologies and correlating them to potential applications. The simplest application example, the map matching area correlator, is a demonstration of where the state-of-the-art technology has been for a long time. For more complex image manipulations (e.g., the dynamics of tracking applications), an image regeneration module and an image timing control module are critical. Developing these components would dramatically broaden the applications base. As an example, a class of vision-assisted tasks is discussed that would be possible to execute using optical transforms based on symmetry recognition, feature extraction using spatial transforms, and the spectral content available in optical images.

ACKNOWLEDGMENTS

Fruitful conversations with R. Hundley, J. Dewar, S. Bankes, N. Shapiro, K. Saunders, K. Horn, E. Bedrosian, G. Huth, E. MacCalla, J. MacCalla, and A. Brewer are gratefully acknowledged. In addition, I would like to thank A. Hearn and P. Davis for careful readings of the manuscript.

CONTENTS

PREFACE iii
SUMMARY v
ACKNOWLEDGMENTS vii
FIGURES AND TABLES xi
GLOSSARY xiii

Section

1. INTRODUCTION 1
2. SYSTEM IMPLICATIONS OF PHYSICAL PROPERTIES 4
Optical Versus Electronic Physical Design Trade-Offs 5
Architectural Implications 7
3. STRUCTURAL MODEL OF A PHOTONIC PROCESSOR 9
Digital Electronic Processor 10
Optical Processor 11
Electro-Optical (E-O) and Opto-Electronic (O-E) Interfaces 12
4. OPTICAL PRIMITIVES 15
Regenerators 16
Thresholder 16
Copy 17
Delay Loop 17
Fourier Transform 17
Spatial Filter 17
Spatial Shift 19
NOR, NAND, and XOR 19
Joint Transform Correlator 19
Spatial Inverter 19
Symbolic Substitution 19
Importance of the Timing Control Module 21
5. POTENTIAL APPLICATIONS 22
Map Matching Area Correlator 22
IRST Tracking and Clutter Rejection 24
Airborne Lookdown Tracking 24
Vision-Assisted Robot Tasks 26
6. CONCLUSIONS 28
Summary 29
Appendix: IMPLICATIONS OF FABRICATION REQUIREMENTS 31
REFERENCES 35

FIGURES

1. Selected Multispectral Scanner (MSS) Measurements Made Along One Scanline 2
2. Finite State Machines. 5
3. Structural Diagram of a Photonic Processor 10
4. Physical Optical Primitives. 15
5(a). Fourier Transform Optically Generated by a Single Lens Transformation 18
5(b). Spatial Filter Formed by a Pair of Fourier Transform Lenses with the Fourier Transform of the Image Features to Be Identified Interstitial Between the Two Lenses. 18
6. Optical Circuit of Symbolic Substitution 20
7(a). Aerial Viewing of Terrain Below Provides Input Image Frame for Map Matching Area Correlator. 23
7(b). Optical Circuit of a Map Matching Area Correlator. 23
8. Optical Circuit Used for IRST Tracking with Cluster Rejection Capabilities 24
9. Optical Circuit Used for Airborne Lookdown Tracking 25
10. Optical Circuit for Robotic Road-Following Control Loop 26

TABLES

1. State-of-the-Art (SOA) Switching Subcomponents. 7
2. Digital Optical Hardware Implementations 12
3. Photonic Applications and Associated Optical Operational Transforms. 30

GLOSSARY

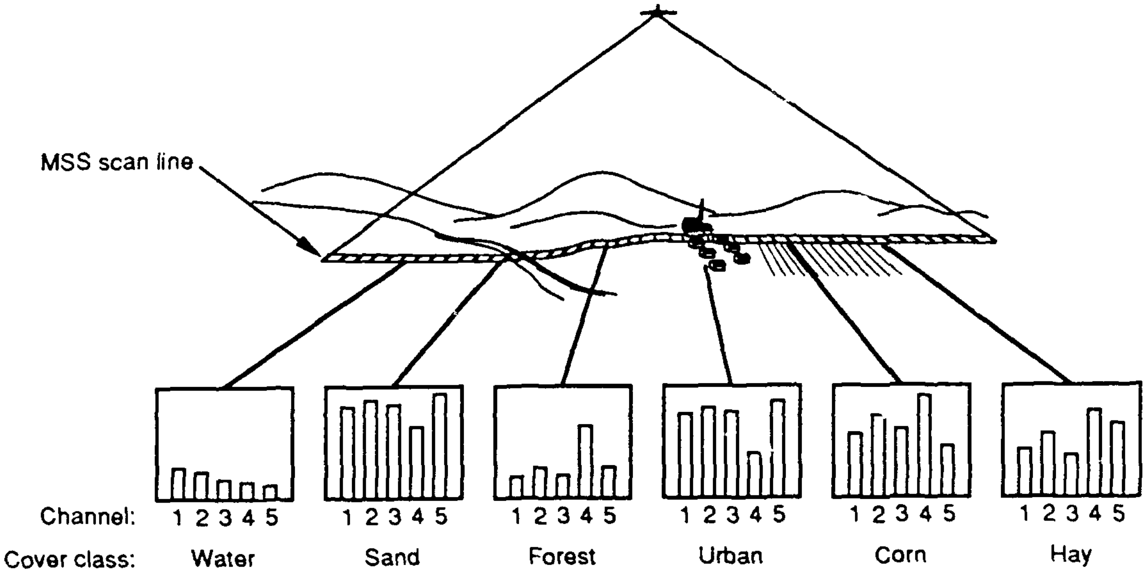
AlAs	Aluminum Arsenide, a semiconductor compound
dual-rail	Data format that is characteristically used for SEED arrays
Etalon	An optical device containing two extremely flat parallel mirrors. The transmission properties of etalons are useful in many optical applications.
GaAs	Galium Arsenide, a semiconductor compound
IFF	Identification Friend or Foe
IRST	Infrared search and track
MBE	Molecular Beam Epitaxy
MOCVD	Metal-Organic Chemical Vapor Deposition
MSS	Multi-spectral Scanner
OLE	Optical Logic Etalon
RPV	Remotely Piloted Vehicle
SAR	Synthetic Aperture Radar
SEED	Self Electro-Optic Effect Device
SELDA	Surface Emitting Laser Diode Array
SLM	Spatial Light Modulator
S-SEED	Symmetric Self Electro-Optic Effect Device
UAV	Unmanned Aerial Vehicle
UUV	Unmanned Underwater Vehicle
UV	Ultra-Violet
VHSIC	Very High Speed Integrated Circuits
VLSI	Very Large Scale Integrated

1. INTRODUCTION

In early 1990, photonics, the confluence of electronics and optics technologies to improve net processing efficiency, was advanced to the highest priority ranking on the DoD critical technologies list. Currently, photonics is considered a high-leverage technology because it is believed that photonic processors could potentially circumvent the serial processor limitation, or **von Neuman bottleneck,** which limits the throughput capacity of most electronic processors. Indeed, the *realtime* solutions to current military problems, such as high-accurate missile guidance, sensor fusion, automatic target recognition, automated guidance of remotely piloted vehicles, etc., are consistently crippled by information processing bottlenecks. Such bottlenecks are particularly endemic to image-formatted data bases.

An image-formatted data base is defined as a data base where, besides the information contained in each pixel, there is also information imparted by the spatial relationship among the data in the pixels. Thus in image data, variations in grey scale are used to define edges and corners. To extract the spatially imparted information, it is often necessary to compare the $N \times N$ pixels in the input image with the $N \times N$ pixels in a model image; this process takes N^4 comparison calculations. As the demand for higher resolution imagery increases and N gets larger, it becomes increasingly more difficult to make the image comparisons in *realtime*. Currently, digital electronic processor designs are optimized for numerical processing, which is an intrinsically serial operation. It is this serial nature that causes the limitations; the photonic processor, which can be designed with a more parallel architecture, has potential for circumventing this bottleneck. It is therefore anticipated that the intrinsic parallelism of optics will enable the photonic processor to solve problems in realtime that were previously considered unsolvable or only marginally solvable.

One application, where photonic processors offer a potentially unique advantage over electronic processors, is making realtime comparisons of images simultaneously recorded at several different wavelengths and extracting useful information from their spectral content. Lillesand et al. (1979) have demonstrated this concept by showing that the *relative amplitudes* of serial images, which are simultaneously recorded at five different wavelengths, will produce "spectral fingerprints" of the objects below. As shown in Fig. 1, pixel-by-pixel comparisons of the relative intensity amplitudes uniquely identify terrain features such as water, sand, forest, and roads. **Spectral fusion** is the ability to optically correlate this



Note: Channels cover the following spectral bands: 1-Blue; 2-Green; 3-Red; 4-reflected IR; and 5-Thermal IR (Hamilton et al. 1987).

Fig. 1—Selected Multispectral Scanner (MSS) Measurements Made Along One Scanline

spectral information at different wavelengths and extract the details of interest in realtime for hand-off to the electronic processor.

Spectral fusion could be used as the first pass at *object identification*, for example. This is expected to work extremely well in a controlled environment, which is defined as an environment where all existing objects have been "spectrally fingerprinted" and each "fingerprint" has an associated object label in the electronic data base. Since the "spectral fingerprint" uniquely defines an object in a controlled environment, this approach to object identification could be quicker than the standard method of trying to recognize an object from its shape. (Although the spectral fingerprint is unique, one might still want to verify the "fingerprint" identification by further querying for a specific feature detail that is known to be associated with the identified object.)

Even in an uncontrolled environment, spectral cues are useful in ruling out incorrect choices and eliminating confusion. To date, primitive spectral identification schemes have been successfully used by IR missiles to counter decoy flares that have been kicked out by the target aircraft. By comparing the relative brightness of IR and UV spectral *emissions*, one can uniquely differentiate decoy flares from the exhaust plume of a target aircraft. Clearly, with the proliferation of hardware decoys (cardboard tanks, missile launchers, etc.), the ability to differentiate real versus fake threats becomes more desirable. *Spectral fusion* has the potential for responding to this need. This is just one example of the potential applications that photonic processors open up.

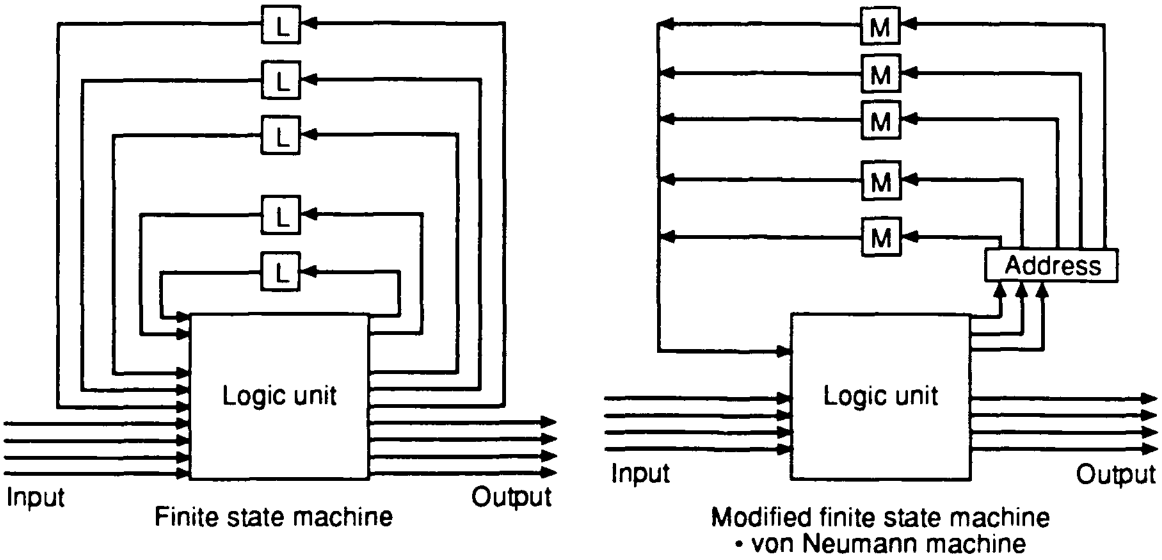
To help understand the relative importance of current research efforts, this Note defines a structural model for opto-electronic processors within which most of the currently implemented devices fit. The same model is then used as a framework for exploring potential applications. It suggests that it is important to evolve a processor whose design function is to extract the spatial content of information from image-formatted data bases. To arrive at the structural model. one must first understand something about the differences in physical properties between optics and electronics. Section 2 describes these differences and discusses their implications for system or architectural design. Sections 3 and 4, respectively, define a structural model for a photonic processor and describe some primitive optical transform modules. This lays the groundwork for a discussion in Sec. 5 of several applications that appear amenable to photonic processing. Specifically, the relevant "optical circuits" and their relationship to the opto-electronic interface and the electro-optical interface are discussed. The concluding remarks in Sec. 6 summarize the methodology used in developing the structural model of a processor optimized to solve *image-format* problems. This same approach would be used in refining the processor further as it evolved. Finally, the appendix discusses fabrication requirements and what they imply about implementation costs.

2. SYSTEM IMPLICATIONS OF PHYSICAL PROPERTIES

Many aspects of today's computer architectures and algorithms are designed to overcome the intrinsic limitations that the planar all-electronic format imposes on a processor. The intrinsically serial nature of the modern digital electronic processor is a critical example of these earlier limitations. Its ancestor, the **finite state machine**, was implemented with a parallel architecture. This early processor consisted of storage elements, a combinatorial logic unit, inputs, outputs, and various interconnections as shown on the left in Fig. 2 (adapted from Huang, 1984). With each cycle, the storage elements in this processor are simultaneously updated in parallel as indicated in the figure. This particular processor design has the disadvantage that its memory is limited by the physically allowable interconnect density between the logic unit and the storage elements.

Its successor, the standard **von Neumann** architecture currently in use, is a direct result of a past design decision to get around this limit. The resulting trade-off reduced the interconnect density in exchange for taking longer times to access the processor storage elements. A **von Neumann processor**, illustrated on the right side of Fig. 2, contains memory elements, a combinatorial logic unit, inputs, outputs, and an address module and associated interconnects. The logic unit can serially access the individual memory elements by requesting the contents of a selected address from the address module. Subsequently, the contents of the selected address are input into the logic unit, one at a time. With this architecture, the electronic processor memory has grown to large sizes without requiring the high interconnect density needed for a finite state machine. Operations on large data bases are simply accomplished by sequentially addressing each element of the data base. As the size of the memory grows, however, one also finds that processors typically slow down and bottleneck at program instruction loops that handle large data bases.

This early trade-off of longer processing times in exchange for reduced interconnect density clearly limits our current ability to improve processor efficiency; hence the recent interest in simpler variations of the **finite state machine,** such as parallel pipeline processors. Other parallel architectures, such as the the hypercube and the connection machine, are of interest because they offer more flexibility than the parallel pipeline processor in the crosswise transfer of information. Some of these parallel processors have been demonstrated on simple problems but encounter serious difficulties when applied to more complex problems because the formalisms for programming these systems have yet to be worked out. (As discussed later in this Note, the formalism is a set of rules that imposes



Note: The earliest processors were finite state machines. The modified finite state machine has a von Neumann architecture where the memory is location addressable (Huang 1984).

Fig. 2—Finite State Machines

enough order on the processor to make it useful in solving problems.) The optically based photonic processor has generated interest because it too has the potential for removing the bottleneck.

OPTICAL VERSUS ELECTRONIC PHYSICAL DESIGN TRADE-OFFS

Historically, photonics has been sold as the panacea for all of the limitations encountered by electronic interconnects, but always in the context of a von Neumann architecture, rather than that of a finite state machine. It is typically argued that since optical signals can be routed through each other without distorting the information content (i.e., with no crosstalk), they are an ideal replacement for electronic interconnections, which have limited packing densities. But in attempting to make a one-for-one replacement of electronic interconnects for optical interconnects, the advantages of optics are typically balanced by a comparable set of disadvantages, which must be traded-off in the actual design of a selected system architecture. Some of the trade-offs are identified in the following paragraphs.

High Bandwidth Versus Low Switching Power

It has often been argued that the inherently larger carrier frequency of optics (10^{14}Hz) meant that intrinsically higher bandwidth (i.e., faster) devices would be available for optical processing. Although optical devices have the potential for attaining much higher bandwidths, it has been empirically determined that the achievable bandwidth scales linearly with the switching power. Thus, when placed in the context of a full-scale system architecture, it is often impractical to realize bandwidths as high as 10^{12} Hz because the net system power dissipation would be too high.

The alternative is to use devices with slower switching speeds and to take advantage of the massive parallelism of optics to parallel process. Table 1 lists the optical devices that have high potential as building block candidates in photonic applications. It is interesting to note that these devices are much slower than their state-of-the-art electronic counterparts. Their potential for being parallel processed in large arrays (e.g., 128 × 128 or 1000 × 1000) is, however, much higher than similar electronic array processors. One thus has the potential for achieving an enhanced throughput capacity with an optical finite state machine as compared with an electronic von Neumann processor.

Low Crosstalk Versus Low Switching Power

Optical interconnects are less prone to crosstalk because photons do not interact directly with each other. Instead, they require a coupling medium that provides a coupling path through second- or third-order interactions. Electron-electron interactions, on the other hand, are a first-order effect. As a result, the price that electronic systems pay for superior switching characteristics is that they are heavily prone to crosstalk interference. On the other hand, optical devices have traditionally required higher switching powers than comparable electronic devices performing the same function.

Recently, bandgap-engineered materials have been used to significantly reduce the disparity in switching powers between optical and electronic devices. The corollary is not true; bandgap-engineered materials have not led to a reduction in the first-order electron-electron coupling interaction, which causes the crosstalk in electronic systems. This means that low-power optical devices are beginning to realize a performance advantage over electronics in certain applications. Consequently, optics offers a combination of low crosstalk and low switching powers, again making the von Neumann trade-offs unnecessary.

Free-Space Optical Image/Signal Propagation Versus Electronic Interconnects

One of the attractive features of free-space optics is its massive parallelism. The penalty incurred with current free-space optical technology is that the choice of optical paths must be defined by the projective transformations allowable by geometric optical devices (mirrors, prisms, lenses, etc.). On the other hand, signal propagation in an electronic circuit

Table 1

State-of-the-Art (SOA) Switching Subcomponents

Device	Switching Time
HEMT (SOA electronic)	5 psec
OLE	30 psec
SEED (SOA optical)	131 psec
SLM	1 msec –1 μsec

is easily achieved through point to point wire connections which can be of arbitrary length and topology. Though at first glance this appears to be a drawback for optical systems, one will see in the next section that this reduced flexibility may actually prove useful in defining parallel architectures.

ARCHITECTURAL IMPLICATIONS

The difference between these two technologies (e.g., in susceptibility to crosstalk and in physical interconnection capabilities) debunks the widely held notion of one-for-one replacement of electronic subcomponents with their optical counterparts. It suggests that the architecture of an optical processor must necessarily differ from that of an electronic processor in order to best take advantage of their different physical properties.

Developing a whole new processor architecture is an expensive endeavor because it requires the development of a new formalism within our information processing concepts. The formalism is a set of rules that imposes enough order on the processor to make it useful in solving problems. In the new parallel electronic architectures, such as the connection machine or the hypercube, the options available for the sidewise transfer of information within a database rapidly overwhelm the abilities of the programmer. In contrast, processors based on free-space optical transforms are governed by the geometric laws of optics. Although the laws of optics restrict the ways in which one can move information about in a data base, it is still possible to achieve sidewise transfers of information by invoking certain optical transforms. Therefore, even though the regularity of optics may limit one from forming random interconnects, its governing laws also represent the potential beginnings of a formalism for manipulating a data set. With this realization, two other important questions still need to be answered:

(1) Can the body of knowledge that currently exists for free-space optical transforms simplify the effort needed to develop a useful formalism for the photonic processor?

Saleh (1989) has evaluated the broad classes of nonlinear transformations currently available for optical image processing. In his discussion, he emphasizes the decomposability of certain transformations into serial (cascaded) and parallel (multiplexed) combinations of elementary operations. The work thus lays the foundation for defining a set of primitive optical transformations.

(2) What are the unique advantages offered by the photonic processor that are not accessible to electronic processors?

Optical transforms are exploited whenever they can solve problems that are either unresolvable or cannot be resolved fast enough by standard digital electronic methods. Typically, transforms involve extracting the spatial information content embedded in an image-formatted data base (see the discussion on spatial filters in Sec. 4). Furthermore, the parallel nature of image transforms means that scaling up in resolution does not lengthen the timing loop. Likewise, it is much more awkward and wasteful to use optical transforms for numerical processing because numerical problems are much more serial in nature. Electing to solve a problem in *image format* rather than *serial format* is loosely analogous to rotating out of cartesian coordinates into spherical coordinates to solve selected classes of problems (e.g., orbital mechanics). That is, the new coordinate system can simplify the setup and subsequent solution of the problem.

The rest of this Note addresses the first question by describing the metholology for developing a generalized optical processor using available optical transforms. The second question stated above is, in fact, a design rule or requirement that must be met by photonic hardware if it is to be practical. Hardware that is not capable of passing this test may still have practical potential, however, if its performance can be improved through research and development.

3. STRUCTURAL MODEL OF A PHOTONIC PROCESSOR

Although it is generally believed that the fundamentals of how to build a computer processor are well understood, problems arise in designing a photonic processor because our architectural prototype, the digital electronic processor, has been optimized to take advantage of *physical electronics* and work around its unique set of problems. Because optics offers a different set of advantages and problems, it requires a different architecture that is designed around the advantages and drawbacks of physical optics. A photonic processor should incorporate the best of both worlds by utilizing the optics and the electronics where each is most powerful and effective.

To develop a photonic processor, one would use the digital electronic processor as a prototype, not for its architecture, but for the methodology that was used to develop it. Specifically, the digital electronic processor was evolved by first defining and understanding the general class of problems that the processor was expected to solve. Defining the applications base imposes certain performance requirements on the processor; these must then be correlated with available technology. One must remove any existing technology shortfall through research and development or by redefining the class of problems that one anticipates solving.

Having defined the problem of interest as extraction of spatial information content from image-formatted data bases, the next step is to closely examine potential applications to determine what specific **design functions** the photonic processor has to fulfill. The discussion that follows proposes a modification of the currently accepted generic computer architecture. It is within the framework of this modified architecture that potential image-format applications will be developeɑ. For the purposes of this discussion, assume two parallel, multilevel processors, one optical and one electronic, as shown in Fig. 3 (generic electronic processor in figure adapted from Tanenbaum, 1984). Assume also that the optical-electronic interface between the two processors (represented by the large question mark) is application dependent. A **finite state machine** architecture is chosen for the optical side of the processor because the higher memory limits of an optical processor allow more room for exploiting this architecture. The division of labor within the processor should have optics doing what it does best and electronics doing what it does best. Hence, applications where information is imparted both by the content of an individual pixel as well as its spatial relationship to adjacent pixels (e.g., matrix transformations, pattern recognition, etc.) are

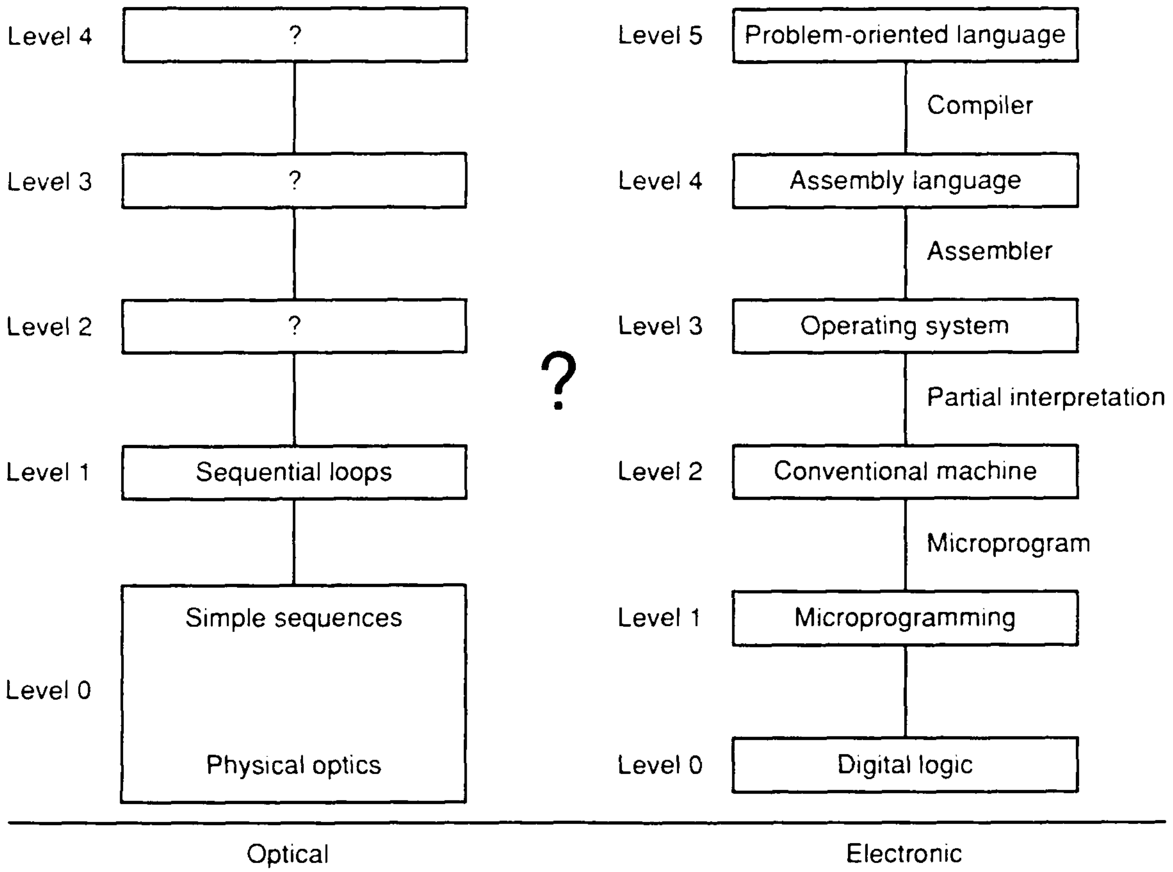


Fig. 3—Structural Diagram of a Photonic Processor

best executed optically. Problems that do not require that the data base entries preserve a spatial relationship and are amenable to solution via a time-sequenced series of events find optimal solution in the electronic processor. Examples include numerical processing, management functions within the central processing unit, and execution of the program code. The following discussion describes both the relative organization of the two sides of the processor and the ideal nature of the interface between them.

DIGITAL ELECTRONIC PROCESSOR

The evolution of the digital electronic processor is quite mature, entailing multiple levels of organization. Each level is built upon the one below it and is thus of increasing complexity.

Level 0. The digital electronic processor is composed of logic gates that can recognize and execute a few basic instructions. At this level of the tier, there is no concept of a program.

Level 1. By stringing the **level 0** instructions together in sequence, one introduces the concept of a *program*. Thus there is a small set of programs (less than 20) defined at **level 1** that makes up the *microprogramming instruction set*, which in turn becomes the programming language at the next level.

Level 2. Microprograms written at **level 2** using the *microprogramming instruction set* are *interpreted* at **level 1** so that they can be executed by the logic gates at **level 0**.

Level 3. At this level of the organization, more than just additional instruction sets are added. Other features, such as a different memory organization and the ability to run more than one program in parallel are introduced to add flexibility and more efficient system management to the processor.

Level 4. Language improvements above **level 3** are intended to make the language more user friendly or more application specific. In most conventional machines the language at **level 4** is assembly language.

Level 5. The language at this level is the next iteration in the process to evolve a more user friendly language.

OPTICAL PROCESSOR

A *niche* processor is defined as a processor that is designed to solve one specific problem and that, by design, lacks the flexibility to solve variations on this problem. In current state-of-the-art optical signal processing, the so-called *niche* processors are all represented at **level 0** in Fig. 3. These include applications such as optical synthetic aperture radar (Cutrona et al., 1966), acousto-optic synthetic aperture radar (Psaltis et al.,1987), optical correlators (VanderLugt, 1964), and optical crossbar switches (Sawchuck et al., 1987). Only a few of these niche processors, such as correlators and Symmetric Self Electro-Optic Effect Device (S-SEED) devices, have interfaces that support modularization, thus making them useful building blocks for an optical processor.

Table 2 lists a number of recent hardware systems that have been built to demonstrate that digital optical processors can solve practical problems. Several of these (Huang, 1989; Guilfoyle et al., 1988; Ramanan et al., 1990; and Murdocca et al., 1988) are programmable and would be considered **level 1** processors. The question marks at **level 2** and above imply that optical processors have not climbed the ladder to include formal languages. *Nevertheless, the limited class of problems (i.e., image format) that they can solve tend to be complex by digital electronic standards.*

Table 2

Digital Optical Hardware Implementations

AT&T—Alan Huang	Crossover network NOR-OR logic; lo-fan in, lo-fan-out
Westinghouse—Dick Magerian	Adaptive array processor AND-OR logic; hi-fan-in, hi-fan-out
Opticomp—Peter Guilfoyle	Globally interconnected processor AND-OR-INVERT; hi-fan-in, lo-fan-out
University of Colorado—Harry Jordan	Transmission line crossbar switch Long latency, fiber-based memory
Herriot Watt University	SLM with e-beam write input

At this point, the main significance of these hardware implementations is that they have demonstrated a very good snapshot of where the engineering advantages and problems exist in optically based processors. Among the recurring themes, designers have leveraged off of the high fan-in and high fan-out available with optics, exploited long latency times (often as a distributed memory in the form of fiber or free-space delay lines), and/or utilized the precision time control available from optics. In addition, the ability to optically threshold, the ability to regenerate the signal, and the ability to cascade from one device to another are common to all of these processor designs.

One can think of the physical optical devices that have been used to develop niche processors in optics as being prospective components for a photonic processor. They are analogous to the physical electronic devices (transistors, resistors, inductors, capacitors, etc.) that are used as building blocks in digital electronics.

ELECTRO-OPTICAL (E-O) AND OPTO-ELECTRONIC (O-E) INTERFACES

The choice of interface is very important in determining the processor efficiency. In photonic systems that have been reduced to practice, the interface takes one of two forms:

- Localized to a spatial plane and a single point in time, or
- Distributed either in time or to points lying outside of a single spatial plane.

In the local interface, each point in the arriving image frame maps into each point of the interface simultaneously. Distributed interfaces, on the other hand, often use fibers to form random interconnects at a variety of points in space, the total set of which defines many different planes in space (Goutzoulis et al., 1988; Ramanan et al., 1990). The SAR processor (Cutrona et al., 1966) represents a variation on this. It uses optics to perform a Fourier transform on the radar return pulse, but the spatial information in the return image is distributed in time. To be useful, it must still be unscrambled after the *image-formatted* data are converted at the O-E interface. Generally, a distributed interface forces one into niche applications.

The electronic side of the local interface can take one of three forms: (1) a single serial data port, (2) a bus data port, or (3) a CCD readout. Currently, the most common local interface in use is the serial data port.

Reciprocity at the Interfaces

In both the electronic and optical technology bases, converting a single bit of information from electronic to optical or from optical to electronic is easy. However, because the electronic format is intrinsically serial and the optical format is intrinsically parallel, there is no reciprocity between the O-E and the E-O conversion. Pixels arriving at the interface for E-O conversion are converted immediately because they are retrieved one at a time. The O-E conversion, on the other hand, will be prone to bottlenecking because a full frame of $N \times N$ pixels arrives at the interface simultaneously. Clearly, as high resolution becomes more important in images (i.e., as full frame images go from 256 × 256 to 1000 × 1000, etc.) the need for optical image processing will become paramount because the serial aspects of digital electronics cannot compensate for the information bottleneck.

Rules for Transiting the Interface

Because of this lack of reciprocity, the first rule of thumb in using a photonic processor is to **minimize the O-E conversion**. Many architects of photonic systems violate this rule by constantly converting from optical to electronic and back while still in image format. Although this procedure may be useful for simulating certain processes, it will always have difficulty competing with a pure digital electronic processor.

The second rule of thumb is to **use transforms to reduce the image-formatted data base to serial format** before the O-E conversion. If this is not possible, then at least reduce the size of the data base to be converted by picking out either the regions or information of interest.

The third rule of thumb is to **minimize the number of times** images are subjected to **E-O conversions.** Even though this is not technically a bottleneck, these conversions can be time consuming because an N x N pixel image requires N^2 cycles to complete the conversion. In the examples given in Sec. 5, the information passed in the E-O conversion is limited to control functions received from the electronic processor to reprogram the optical modules for the next data set.

4. OPTICAL PRIMITIVES

Before looking at potential photonic applications, some knowledge of the tools available for their implementation is in order. This section defines a basic set of optical transforms (symbolically represented in Fig. 4), which will be used as building blocks in illustrating and exploring potential applications in Sec. 5. These transforms can be thought of as "black boxes" that accept image inputs from the left and output them to the right after performing the designated transform. Two conventions of interpretation are assumed: (1) A symbol placed inside a box implies that the operation or transform is simultaneously carried out on each element of an $N \times N$ image-formatted data base. (2) The box also implies that after the image is transformed, it is regenerated.

Except for the delay loop module, all of the described primitive modules have been reduced to practice in an array (*image-formatted*) form. This set of primitives is chosen only from those that were clearly cascadable. They have been selected only to illustrate some

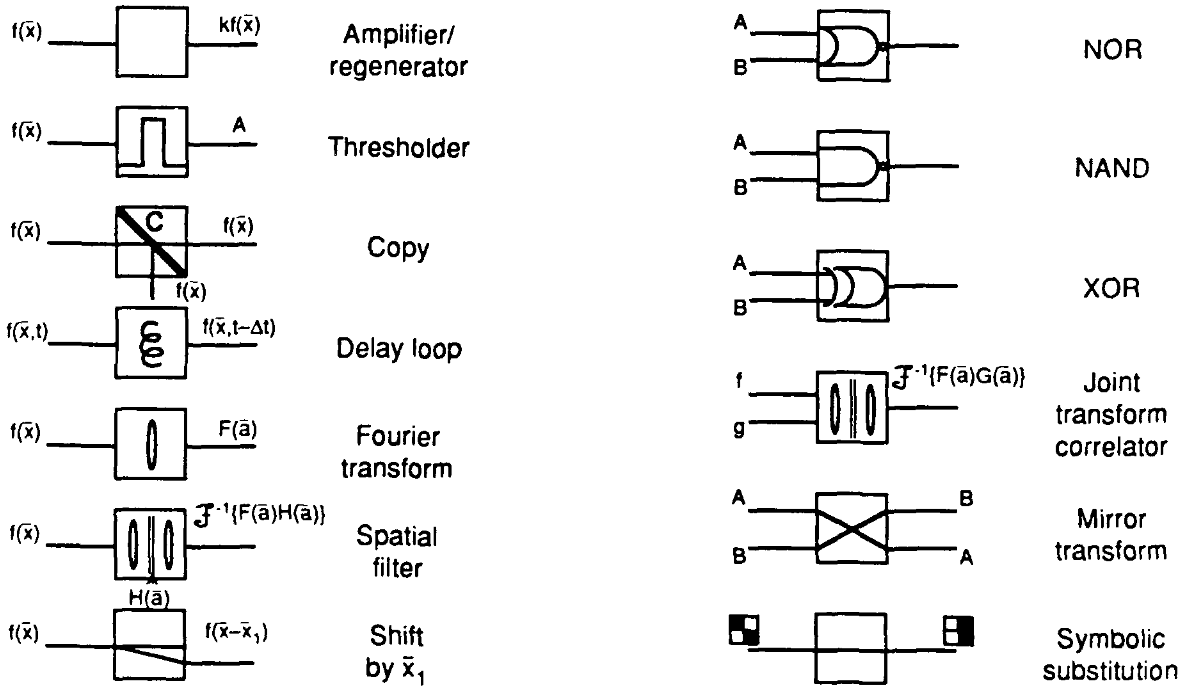


Fig. 4—Physical Optical Primitives

basic concepts about how one moves towards solving more complex image-format problems. They have not been rigorously derived and demonstrated to be a complete set; in fact, some of these primitives might not be included in a later set that was more rigorously defined.

There are two modules that are very important to achieving the capability to perform more complex operations. Specifically, the ability to program any random sequence of optical transforms on an image input requires the capability to regenerate an image and the capability to delay or control the time of arrival of an image frame at an image transform module. The ability to regenerate the image from one module to the next *without the loss of information content in the image* is clearly important if one wishes to solve more complex *image-format* problems. As mentioned earlier in Sec. 2, practical regeneration modules have already been realized. In contrast, the ability to control the arrival time of the image data at a preselected image transform module is not as far along.

Although the delay loop or some other means of controlling the time at which one introduces an image frame into a processing module is realized only for very short delay times, it is emphasized that its use here is symbolic of a generic timing control module. In the classical digital electronic processor, the functional analog of the optical delay loop is the memory. Optical memories can be one-, two-, or three-dimensional. The one that makes the most sense for image-formatted data bases is three-dimensional holographic memories, which preserve the spatial information content of the image pixels by permitting a simultaneous full-frame write-in or read-out of the data. Although optical holographic memories are currently available, none utilize realtime methods of performing the read and write functions. As a result, we have chosen not to include a memory module in the set of primitives in Fig. 4 and have chosen instead to represent the time delay function using a series of delay loops and regenerators. This simple form of a two-dimensional memory module will be used in our discussion examples.

REGENERATORS

A regenerator accepts an optical input in *image format*, and regenerates this image with an increased optical amplitude k, where k is greater than 1. The regenerator output can be at the same wavelength as the input or at a different wavelength.

THRESHOLDER

Thresholding can be done by measuring either amplitude or phase. The amplitude thresholding module accepts an analog input and outputs intensity peaks at each pixel whose intensity exceeds a certain threshold value and whose intensity nulls otherwise. In this discussion, it is assumed that the thresholder is followed immediately by a *dual-rail converter* that converts the image pixels to a *dual-rail format* that is compatible with SEED devices.

COPY

The copy transform is a 50 percent beamsplitter. Each output leg of the beamsplitter is followed by a regenerating module that resets the two outputs to the initial intensity of the input function.

DELAY LOOP

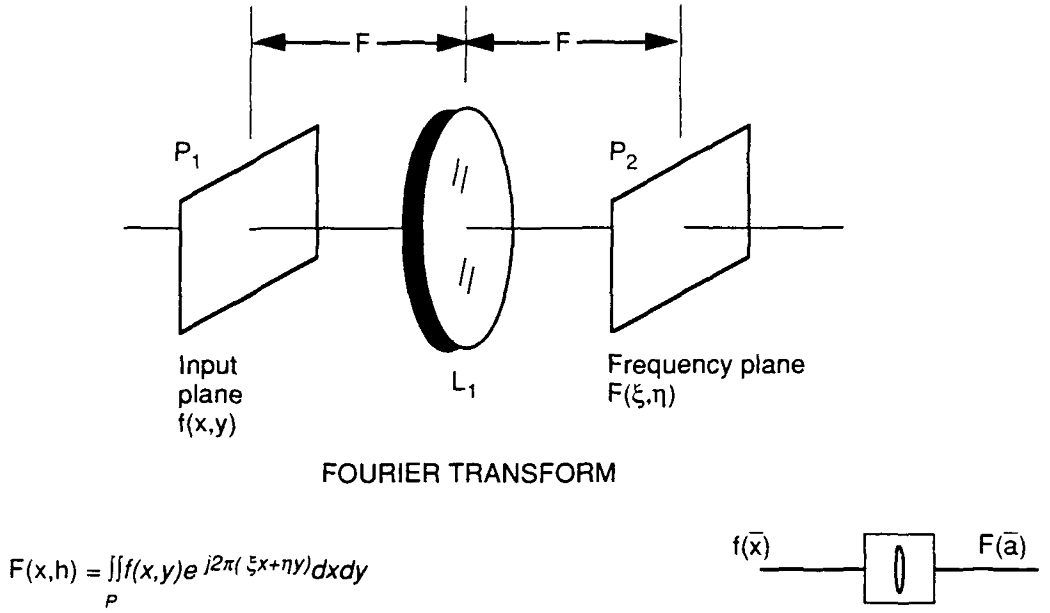
The delay loop introduces additional optical pathlengths, which allow one to control the arrival time of two optical images at a gate element. This discussion assumes delay loops of fixed time intervals that are compatible with currently available modulator technologies. Although delay loops have been demonstrated with the SEED devices, they have yet to be demonstrated with Spatial Light Modulators (SLMs) and Optical Logic Etalons (OLEs). If the delay loops were designed to be variable, the types of image-format problems that could be solved with the photonic processor would increase significantly.

FOURIER TRANSFORM

A Fourier transform is generated when one inputs an image, f(x,y), at the plane located a focal length from a transform lens (see Fig. 5(a)). The Fourier transform of that image, F(ξ,η), is then produced after ray propagation through the lens to the plane located at a focal lengths distance on the other side of the lens.

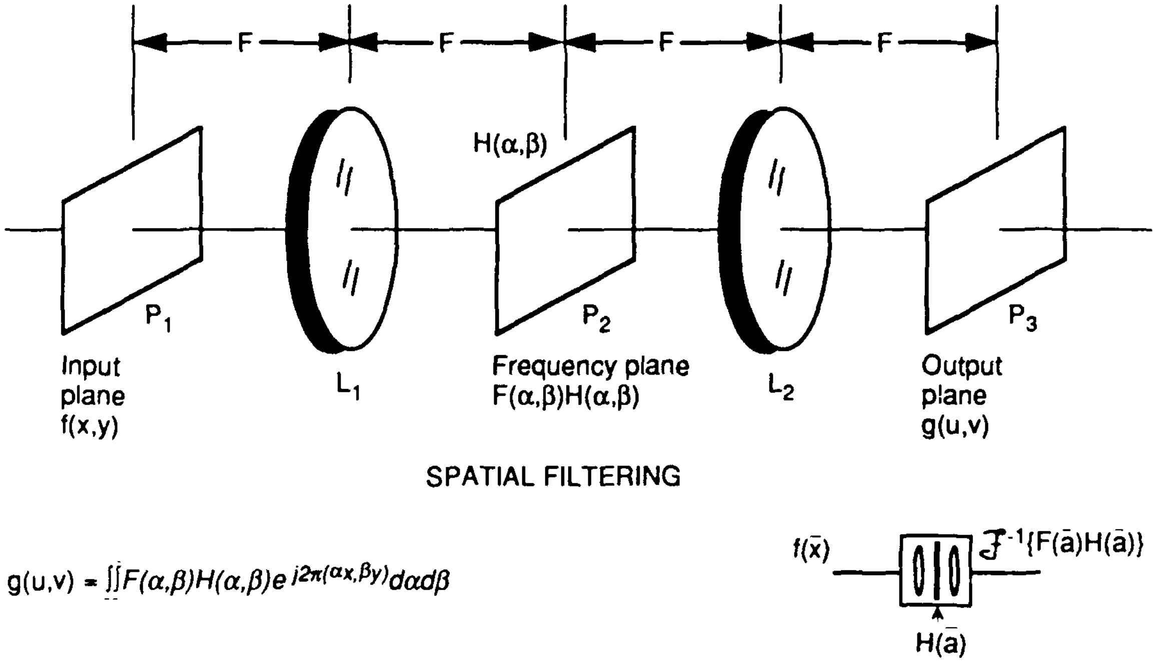
SPATIAL FILTER

A spatial filter is illustrated in Fig. 5(b). Basically one places a "filter" in the form of a spatial light modulator that contains the Fourier transform, H(α,β), of the pattern one wishes to recognize interstitial between a pair of lenses. Each lens then Fourier transforms the incoming optical input, F(x,y), as it transits from left to right. When there is a filter between the lenses, the second lens Fourier transforms, calculating the correlation function of the product of the input transform, F(α,β), and filter transform H(α,β). The output produces an intensity peak everywhere that they correlate.



$$F(x,h) = \iint_P f(x,y)e^{j2\pi(\xi x+\eta y)}dxdy$$

Fig. 5(a)— Fourier Transform Optically Generated by a Single Lens Transformation



$$g(u,v) = \iint F(\alpha,\beta)H(\alpha,\beta)e^{j2\pi(\alpha x,\beta y)}d\alpha d\beta$$

Fig. 5(b)— Spatial Filter Formed by a Pair of Fourier Transform Lenses with the Fourier Transform of the Image Features to Be Identified Interstitial Between the Two Lenses.

SPATIAL SHIFT

This transform evenly translates all elements of an image to the left or right, up or down by a specified number of pixels. For image transforms, this is typically accomplished using mirrors tilted to give the proper translation. However, acousto-optic modulators also can produce the same results.

NOR, NAND, AND XOR

These Boolean logic modules are the first set of dual-image input transforms to be described. Essentially, two input frames arrive and overlap at the transform module. Then each *dual-rail* pair of SEEDs outputs according to the following status substitution rules:

A	B	NOR	A	B	NAND	A	B	XOR
0	0	1	0	0	1	0	0	0
1	0	0	1	0	1	1	0	1
0	1	0	0	1	1	0	1	1
1	1	0	1	1	0	1	1	0

The NOR transform module is a direct manifestation of stringing together two SEEDs in parallel across a voltage source (Lentine, 1988). The NAND and XOR modules can then be constructed out of the NOR.

JOINT TRANSFORM CORRELATOR

A joint transform correlator is a dual-input transform with two image inputs (Hamilton et al., 1987). The transform outputs correlation peaks everywhere that the two images match.

SPATIAL INVERTER

The spatial inverter or mirror transform exchanges the contents of pixels that are spatially opposite across either the vertical or horizontal image axis. This technique has been used extensively by Murdocca et al. (1988) to reconfigure interconnects in a high bandwidth, large array crossover network. In *image-format* applications, this transformation can be very powerful in identifying symmetries, an ability that is often important in vision-assisted tasks.

SYMBOLIC SUBSTITUTION

Symbolic substitution is an image transformation that allows one to locate a specified spatial configuration of information among the image pixels. This configuration can be recognized simultaneously everywhere that it occurs in the image. After the recognition step, one can then replace the identified pixels with a *different* spatial configuration. As shown in Fig. 6,this is a complex transform composed of several basic modules strung together. In the example given, the substitution rule is indicated below the circuit diagram. The dark squares are considered to be optically on and the light squares are optically off. After four copies of the full input image frame are made, each copy is shifted up, down, right, and left by one unit, respectively. When the shifted images are summed at a threshold module, the intensity threshold is exceeded only at the positions in the image where the input rule pattern is found. The output rule pattern is subsequently produced by combinations of down-left, up-left, down-right, and up-right shifts. These are then summed to produce a final image with appropriate substitutions. With the development of a formal language, this transform has the potential for being very powerful since it introduces a method for moving informatior from one portion of the image to another and represents a general formalism for performing logic operations using symbolic representations within image data bases. In fact, when it was first introduced by Brenner et al. (1986), it was used to demonstrate two concepts: (1) how to perform binary addition within an image and (2) how to achieve the bit carry.

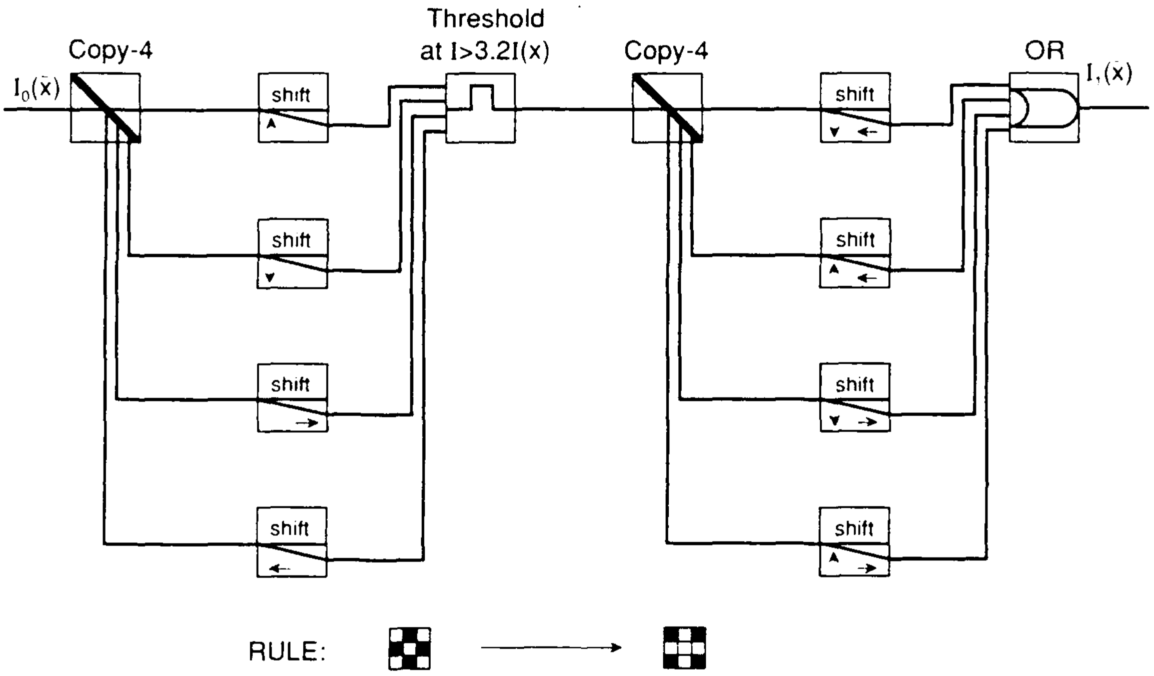


Fig. 6—Optical Circuit of Symbolic Substitution

IMPORTANCE OF THE TIMING CONTROL MODULE

In this discussion, the notion of time control has been principally that of introducing an image frame to a module for processing some fixed time delay, Δt, after it would have arrived in the absence of a delay loop. Given the current modulator technology, time delays on the order of several microseconds are necessary to accommodate the slower switching speed of the spatial light modulator. In the current state-of-the-art, image delays of roughly 10 nanoseconds have been demonstrated. These delay times are adequate for demonstrating some of the applications concepts discussed here only if the faster SEED or OLE devices were adapted to perform the spatial filtering and correlation transforms.

Alternatively, if the image could be regenerated an infinite number of times without losing the information of interest, the requirement for bootstrapping to a delay loop of sufficient length could be satisfied through regeneration. Wilkinson (1963) has already proven that a 1×1 or serial data base using a binary coding scheme shows convergence when it is regenerated. This means that it can be regenerated an infinite number of times without loss of information. In the case of the $N \times N$ optical image regenerators, one must determine what conditions are needed to provide a similar convergence on an image or whether there is a need to limit the number of permitted regenerations so that the spatial information content does not degrade. At any rate, acquiring the ability to control the arrival time of an image frame by a fixed time delay (or some multiple thereof) has the effect of broadening the applications base from niche applications such as the map matching area correlator to more dynamic problems such as tracking and road following.

If one is able to introduce variable time delays, the applications base broadens further. Finally, if one is able to access the image frames in a different order from the original sequence in which they arrived, the applications base broadens even further. Holographic memories, which are currently available in forms that are very analogous to both content-addressable memories (Dunning, 1987) and location-addressable memories (Paek, 1989), would permit flexible time control over the image information. Currently, realtime holographic memories are not available.

5. POTENTIAL APPLICATIONS

Before d^eveloping a processor, one needs to have detailed knowledge of its intended applications. After all, the applications ultimately define the performance specifications imposed on the processor. Knowledge of the applications also highlights which system tradeoffs are permissible. The purpose of this section is to describe some potential photonics applications. In the next few pages, *optical circuits,* which are built from the optical primitives defined in Sec. 4, are described. They demonstrate solutions for the following *image-format* problems:

- Map matching area correlators
- IRST tracking
- Airborne lookdown tracking
- Vision-assisted robot tasks

As previously stated, the assumption is that the process operations flow from left to right.

MAP MATCHING AREA CORRELATOR

This is an example of **feature extraction** using a spatial filter to perform airborne terrain matching as a means of assisting in-flight navigation (Fig.7(a)). In this instance, the Fourier transform of the terrain map, $H(\boldsymbol{a})$, is stored as a filter between the two transform lenses. The objective is to correlate the instantaneously observed image, $f(\boldsymbol{x})$, of the ground below the aircraft with its corresponding location on the map. The left side of Fig. 7(b) shows actual data of such a terrain map, $h(\boldsymbol{x})$. The Fourier transform, $H(\boldsymbol{a})$, of an $N \times N$ pixel map is stored on a spatial light modulator in the frequency plane. The highlighted inset is the instantaneously observed image of the ground below, $f(\boldsymbol{x})$, that contains $M \times M$ pixels (Fig. 7(a)). This is the input image of the spatial filter, which forms the product $F(\boldsymbol{a})H(\boldsymbol{a})$ in the frequency plane. The output of this correlation integral is shown on the right side of Fig. 7(b). Note the correlation peak at the position of the pattern match between the map and the input scene. Assuming that the output goes into a charge coupled device (CCD) interface followed by a thresholder, one has essentially reduced a $N^2 \times M^2$ step operation on a digital electronic processor to a pair of lens transforms through an interstitial filter. Although the total number of pixels read-out by the CCD array is still N^2, only the data of interest that have been extracted from the image-formatted data base are transformed to a serial format. Thereafter, they are conveyed across the opto-electronic interface to the digital electronic processor.

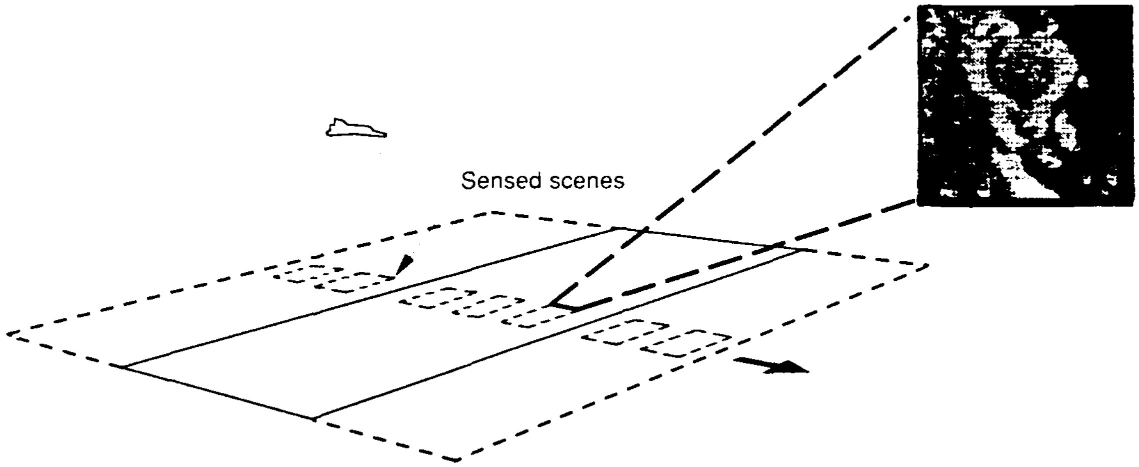
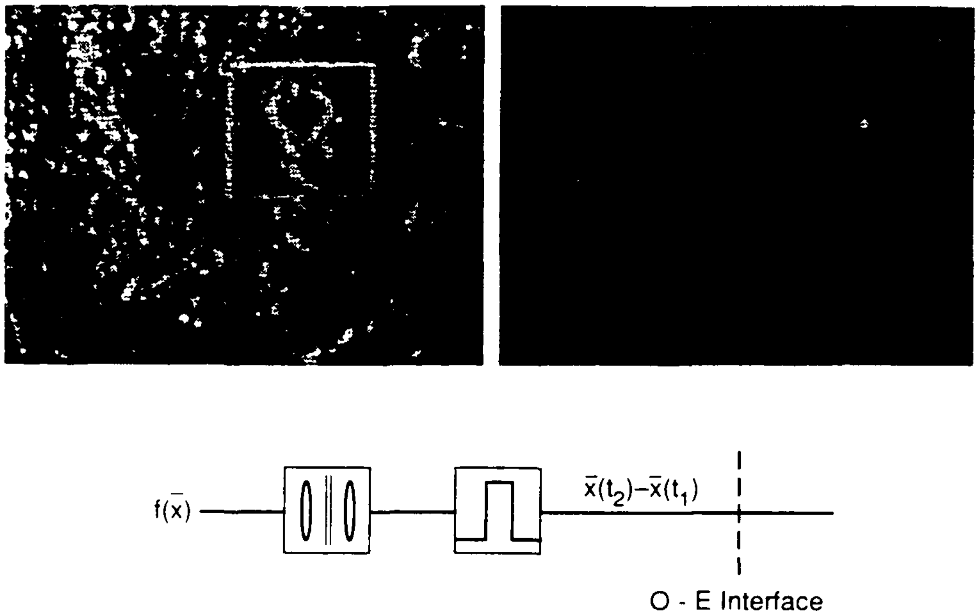


Fig. 7(a)—Aerial Viewing of Terrain Below Provides Input Image Frame for Map Matching Area Correlator (actual image data from Chao et al., 1989).



Note: The intensity peak in the upper right hand corner indicates an image match (actual image data from Chao et al., 1989).

Fig. 7(b)—Optical Circuit of a Map Matching Area Correlator

IRST TRACKING AND CLUTTER REJECTION

IRST tracking is an example of **dynamic extraction**, an operation that can be used to detect any moving objects within the field of view. In this application, the observed scenery is first binary then *dual-rail* encoded with a thresholding module (Fig. 8). The output of the thresholder is then copied and regenerated back to its initial intensity. One copy is sent through a single-frame delay line, while the other is sent directly to an XOR module, which compares the nth frame to the (n-1)th frame. At each position that the contents of the overlapping frames do not match, a Boolean one is outputted at that pixel position. This readily identifies any moving objects in the image. Again, the data have been simplified by the optical circuit from an *image format* to a set of coordinates that represent the information of interest. This is then processed through an A/D converter and made available as data points in a tracking update program that is running on the electronic processor.

AIRBORNE LOOKDOWN TRACKING

In the airborne lookdown tracking scenario, one is looking for changes in the scenery from frame to frame that signify the locations of moving objects. Unfortunately, the background scenery also changes with each frame because of the motion of the aircraft, so it is necessary to include a background correction for the relative motion of the aircraft before performing the frame comparison. This application combines **dynamic extraction** with **dynamic compensation**. The latter dynamically corrects for apparent background motion. Figure 9 depicts an optical circuit that can make this background correction before searching

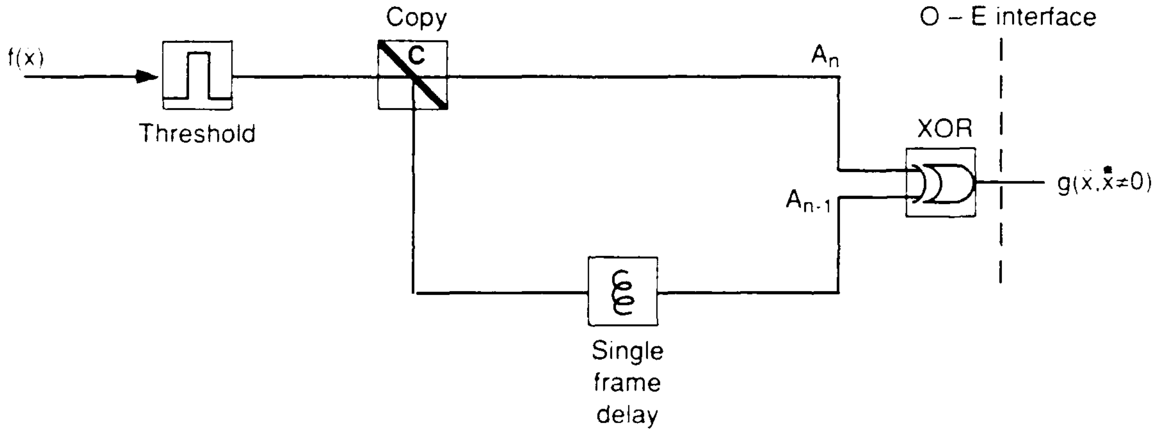


Fig. 8—Optical Circuit Used for IRST Tracking with Cluster Rejection Capabilities

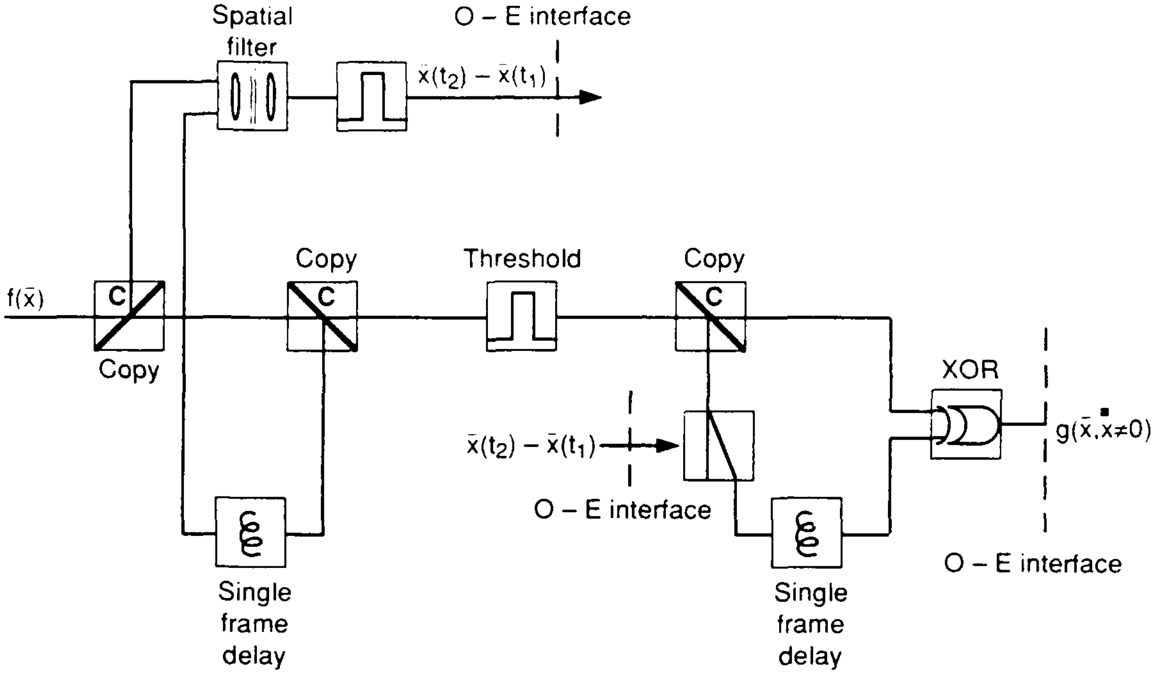


Fig. 9—Optical Circuit Used for Airborne Lookdown Tracking

for moving targets against the background clutter. The right half of the circuit uses a joint transform correlator (spatial filter) to determine how much the relative position between two successive frames has changed. Any position shifts are read-out at the O-E interface and input into the electronic processor. This information is then used to set the amount of spatial shift that is imparted to the image when it arrives at the shifting module. The shifted and delayed (n-1)th image frame is then compared with the nth image frame at the XOR module. This comparison yields an output at every position of the image where something has moved.

This application is an example of how the information flow across the interface between the optical and electronic processors is expected to take place. In this example, the image is optically processed to produce a set of coordinates at the output. The coordinates are input into the electronic processor and used as feedforward control on the optical shift module. Introducing this feedforward allows the proper relative comparison of successive optical image frames to allow the extraction of positional information on any moving targets within the field of view. The final output of the optical circuit, any identified targets, goes to a tracking update program running on the electronic processor.

VISION-ASSISTED ROBOT TASKS

Robotics is another emerging technology area that has the potential for being symbiotic with photonics. Specifically, the development of autonomous robots depends on evolving systems concepts that permit the realtime interpretation of sensory inputs to accommodate decisions. An important class of robotic tasks are those that are aided by image inputs from the environment. To date, image information has often been retrieved for interpretation by teleoperators. We describe two examples of very simple vision-assisted tasks that might be automated, thereby allowing the teleoperator to concentrate on more complex tasks.

Robotic Road Following

This example describes how **symmetry recognition** can be used to orient and keep a robot on the road. This application employs a spatial inversion transform with a joint transform correlator to look for symmetries in the image. In the road-following example, the robotic definition of a road is "a highly defined image symmetry about the vertical axis of the robot's field of view." In Fig. 10, two copies of the robot's image input are copied and compared after one of the images has been spatially inverted about the vertical axis of the robot's vision. Correlation peaks are observed as long as the whole road is within the robot's

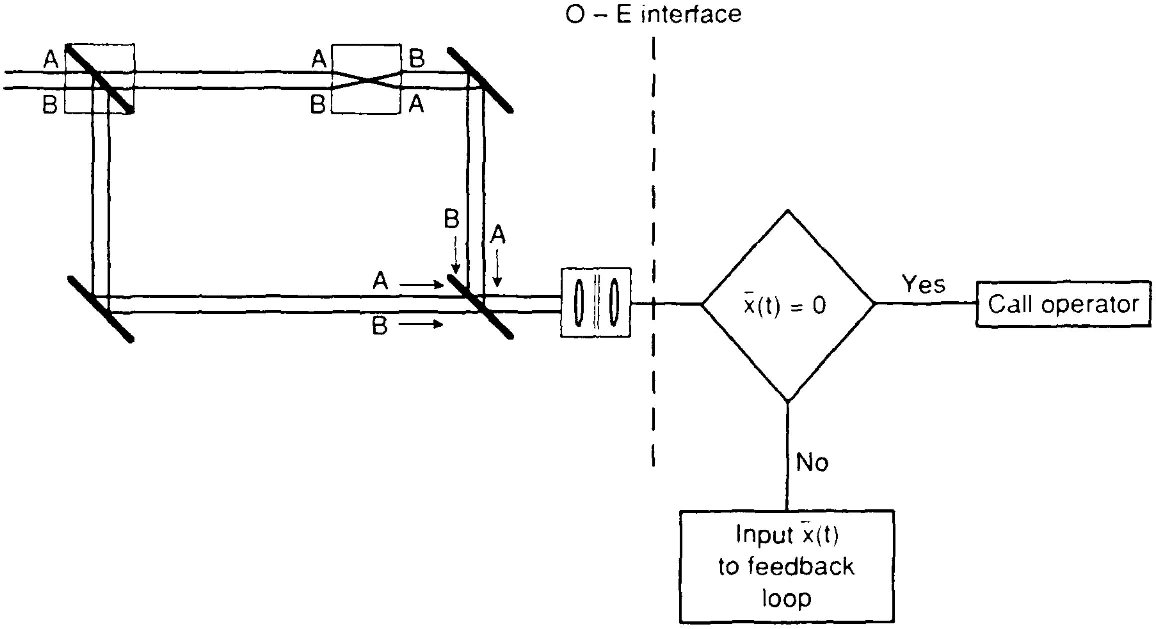


Fig. 10—Optical Circuit for Robotic Road-Following Control Loop

field of view. The electronic processor inputs the position of the correlation peaks into a feedback loop that is used to center the robot at the desired traveling position (i.e., at the center of the road or X meters to the left or right of center). If the symmetry is broken by the presence of an obstacle or any other drastic changes, the robot signals the operator for help as indicated by the decision box in the figure.

Implications of Spectral Fusion on Vision-Assisted Robot Tasks

As discussed in the Introduction, spectral cues are useful in uncontrolled environments for ruling out incorrect choices and eliminating confusion. In the robot road-following example, a symmetric fork in the road that straddles a symmetric concrete barrier creates a decision ambiguity that could lead to a mishap. Spectral fusion can be used to resolve the ambiguity by preselecting that part of the image that has the spectral signature of asphalt. This is accomplished by choosing to look only at symmetries that occur in that portion of the image that has the spectral signature of asphalt. This constraint on the symmetries of interest uniquely defines the road, even during periods of heavy traffic. Thus, one can use a series of symmetry transforms, rotation transforms, spatial filtering (of lines, curves, or angular features within an object), and spectral fusion to begin performing vision-assisted tasks in realtime.

6. CONCLUSIONS

Because of the fundamental differences between the two technologies, it is erroneous to assume (as some researchers have) that the systems engineering work that was done for the digital electronic processor transfers to the optically oriented photonic processor. Given that the digital electronic processor is to be used as the prototype for developing a photonic processor, one must take pains to adopt the *methodology* that was used in evolving the digital electronic processor rather than adopting the *architecture* that was ultimately developed for it. The methodology evolved for the digital electronic processor applies equally well for a photonic processor and is as follows:

- Identify the intended applications base
- Define the required **design function** of the processor
- Develop "circuits" based on available technology
- Identify a basic set of logic functions
- Determine physical design limits of technology base
- Conceptualize hardware design for processor

This study was initiated by completing the first two steps in this methodology. Originally, the intended applications base of the digital electronic processor was composed of classes of problems involving computable numbers and computable functions. Over time, the digital electronic processor has also been applied to more abstract problems such as the extraction of spatial information content from image-formatted data bases. However, the results were mixed because the digital electronic processor design is optimized to accept and operate on *serial-formatted data bases*. The class of problems involving image-formatted data bases is the applications base that the photonic processor, with its intrinsic capability for image parallelism, should be designed for. In addition, the author defines the intended design function of the photonic processor as the extraction of the spatial information content from image-formatted data bases. In contrast, the design function of the digital electronic processor is numerical processing.

The procedure for executing the third step is illustrated through explicit examples in the last two sections where a set of "optical circuits" for potential applications were constructed from a set of defined primitive modules. With this approach, the following basic optical operational functions were defined: **symbolic substitution, feature extraction, dynamic extraction, dynamic compensation,** and **symmetry recognition.** An important conclusion is that the dynamic operational functions will be difficult to realize without the introduction of a timing control module. Although the ability to compare two successively captured image frames on a realtime basis has traditionally been thought of as a "memory function," this analysis demonstrates that for the defined set of applications, the delay line implicit in the architecture of the **finite state machine** is an acceptable means of achieving the delay as long as the information content of the image is preserved when the delays are implemented.

As stated earlier, the set of defined primitives used in this discussion have by no means been proven to be a complete set. But with the addition of other optical primitive modules, one might devise "optical circuits" that demonstrate other useful optical transforms such as *spectral fusion, range extraction, the ability to rotate the axis of symmetry,* etc. Table 3 illustrates the fourth step in the methodology by correlating some applications with six of the seven operational functions that have been mentioned thus far. These six functions would form the basic logic functions employed at level zero of the optical side of the photonic processor. The objective in developing such a table is to maximize the number of applications included in the table while minimizing the number of operational functions needed to construct the applications.

The last two steps of the methodology are beyond the scope of this Note. Clearly, an important area for future research is understanding where processing limitations occur for selected applications such as robot vision-assisted tasks or sensor fusion (of laser radars, IRSTs, day-TV, low light-level TV, millimeter wave radar, etc.). This is an iterative process, in which any identified physical or technological limitations are used to modify the assumptions in earlier steps; however, the order in which the steps are implemented is not sacred. In this particular discussion, for example, a simple structural architecture was conceptualized as soon as the design function was defined in order to limit the physical optical devices considered for developing "optical circuits" to those compatible with the image-formatted data base. With the establishment of a complete set of operational functions, one is ready to develop a more detailed concept of the photonic processor hardware and how it works.

SUMMARY

In summary, the motivation for developing photonic processors is the desire to remove the *von Neumann bottleneck* as a limitation on processor throughput capabilities. In military applications, for example, the vast majority of these bottlenecks occur when extracting information from high-resolution image data bases that, in turn, are readily amenable to

Table 3

Photonic Applications and Associated Optical Operational Transforms

	Functions					
Applications	Feature Extraction	Spectral Finger-print	Dynamic Extraction	Dynamic Compensation	Range Extraction	Symmetry Recognition
Map matching area correlators	X					
Decoy rejection/ID		X				
Tracking			X			
Airborne lookdown tracking			X	X		
Automatic target recognition	X	X				
Missile threat warning		X	X		X	
Autonomous robot tasks						
—object ID	X	X				
—road following		X				X
—macro-navigation	X	X	X	X	X	X
—micro-navigation	X		X		X	X

manipulation by optical image transforms. Electing to solve a problem in *image format* rather than *serial format* is loosely analogous to rotating from cartesian coordinates into spherical coordinates to solve a selected class of problems (e.g., those in orbital mechanics) because the new coordinates simplify the setup and subsequent solution of the problem. Therefore, it is important to realize that there is a certain class of problems that the optical (*image format*) processors can solve with ease, and there exists another class of problems that electronic (*serial format*) processors can solve with ease. These two different approaches to problem solving and processor design are complementary rather than competitive. In fact, it is anticipated that the *image-format* processor will enable one to solve problems in realtime that were previously considered unsolvable or only marginally solvable.

The photonic processor described here could represent a practical solution to a specific class of processing problems, especially if its applications base is large enough to lower implementation costs through economies of scale. The introduction of a timing control module is seen as a critical development to expanding the applications base. In particular, a timing module would open up the possibility of correlating, in realtime, the spectral content of a single image generated at different wavelengths. This *spectral fusion* capability would introduce a significant enhancement over current image processing capabilities in applications such as automatic target recognition, object recognition and identification, and robotic vision-assisted tasks.

Appendix
IMPLICATIONS OF FABRICATION REQUIREMENTS

In addition to the one-time cost of developing a control formalism, there are also expenses associated with developing a fabrication infrastructure for a photonic-based technology. Although it was not possible to examine all of the cost implications of developing photonic processors within the framework of this study, we can nonetheless estimate the magnitude of the expenses because of the close parallel between the optical and digital electronic fabrication technology bases. This appendix briefly outlines these cost implications.

There are five basic sets of subcomponents that are used as building blocks for optical transform modules. These are (1) optical sources, (2) optical modulators, (3) optical detectors, (4) holographic memory elements, and (5) classical geometrical optics elements. Because we wish to take advantage of the high-density interconnects available with free-space optics, one desires planar arrays of sources, modulators, and detectors. Since these are all semiconductor-based technologies, current photolithographic techniques developed for very large scale integrated (VLSI) circuits and very high speed integrated circuits (VHSIC) processing technologies can be used to produce these arrays in compact form. These same photolithographic processing techniques will also become important in fabricating the geometric optical routing elements. In turn, they will be an important driver behind the manufacturing cost, especially if VHSIC clean room standards must be applied to this technology. For example, we estimate that the nonrecurring initial capitalization required to develop a facility to fabricate miniature "optical circuits" would be comparable to that of implementing a VHSIC facility. Moreover, the recurring cost of each process run used to fabricate these optical devices would be about $50,000. Again, there are only two ways to justify such an investment in technology: (1) identify an application that is so important that expense is not an issue and there is no other way to do it, or (2) develop an optical generalization of the processor's capabilities to widen its applications base. In this fashion, one can achieve economies of scale.

Finally, the availability of bandgap and refractive index materials engineering from current molecular beam epitaxy (MBE) and metal-organic chemical vapor deposition (MOCVD) techniques may prove useful in the development of realtime holographic memories.

OPTICAL SOURCES

The function of the optical sources is to provide electronic to optical (E-O) conversion of input information. The state-of-the-art semiconductor diode laser technology currently provides single-mode stable sources that turn on at acceptably low threshold currents (approximately 1 mAmp). Surface emitting laser diode arrays (SELDAs) have recently been made available by both AT&T and Bellcore. These devices emit at near infrared wavelengths between 0.9 µm and 0.82 µm.

OPTICAL MODULATORS

Recent developments in the array-sized modulator technologies provide several practical candidates for incorporation into a processor. The two-dimensional modulators are the Self Electro-Optic Effect Device (SEED), the spatial light modulator (SLM), and the Optical Logic Etalon (OLE) (Lentine, 1988; Tanguay, 1985; Gibbs, 1985). These semiconductor devices can be configured to accept a single *image-formatted* input and regenerate it. In other configurations, inputting two images simultaneously permits simultaneous Boolean logic on an $N \times N$ array of pixels (SEEDs) or the identification and location of correlation patterns between pairs of images. Although its switching time is faster than the SEED's or the SLM's, the OLE array is not easy to cascade; hence it is not used in any example applications given here. SLMs have a mature technology base because they play an important role in display technology. Having been around for a long time, they are extensively employed in photonic applications. The SEEDs, on the other hand, have been available commercially only within the past few years. Finally, acousto-optic modulators can be used to provide one-dimensional modulation capabilities.

OPTICAL DETECTORS

Optical detectors provide the optical to electronic (O-E) conversion function. Semiconductor detectors can easily accept optical inputs, which are then converted to an electronic readout at the output. These devices are readily available as single detectors, linear detector arrays, or two-dimensional CCD arrays. The techniques of fabricating and addressing large arrays of optical semiconductor detectors configured in CCD arrays is currently considered a very mature technology.

HOLOGRAPHIC MEMORY ELEMENTS

There are holographic memory elements available; however, in-depth research on the current status of this technology area was not done as part of this study. Suffice it to say that these elements do exist, and that configurations suggesting potential exploitation in photonic applications are currently realized. However, none of the holographic memories have realtime implementations; hence they will not be used as examples in our applications discussions.

GEOMETRICAL OPTICS

These elements, which perform the routing function in the processor, fall into two broad categories: (1) guided wave optics and (2) bulk optical elements for free-space propagation. Optical fibers and thin film waveguides are two examples of guided wave optics. Free-space propagation is generally achieved using classical geometrical optics elements, which are governed by linear optical transforms. These elements include mirrors, lenses, prisms, gratings, etc.

Currently, the size of optical processing setups is dominated by the dimensions of the routing optical elements. These optical elements and their attendant mechanical positioning and adjustment mounts tend to be too bulky for practical applications where compactness counts. Hence there is a need to miniaturize them by developing micro-lenses, micro-gratings, and other photolithographically generated free-space optical components. AT&T is currently integrating such devices on single blocks of substrate material. Computer-aided design (CAD) is used to lay out the optical setups, and materials fabrication techniques, such as molecular beam epitaxy (MBE), are used to engineer the selected refractive index of the materials. To eliminate the need for mechanical adjustments or tweaking on the finished setup, computer simulations of the optical wavefront propagations are used to fine-tune the surface topologies and remove standard optical aberrations. This means controlling the positioning of all optical surfaces to a fraction of a wavelength (0.1 micron), a dimension that is comparable to the minimum feature size in very high speed integrated circuits (VHSIC) technology.

REFERENCES

Bell, T. E., "Optical Computing: A Field in Flux," *IEEE Spectrum*, August 1986, pp. 35–57.

Brenner, Karl-Heinz, Alan Huang, and Norbert Streibl, "Digital Optical Computing with Symbolic Substitution," *Applied Optics* 25, 1986, pp. 3054–3060.

Chao, T. H., E. Ochoa, and R. Juday, "Optical Information Processing for NASA's Space Exploration," *Proceedings of SPIE,* Vol. 1151, *Optical Information Processing Systems and Architectures*, 1989, pp. 333–346.

Cutrona, L., et al., *Proc. IEEE,* 54, 1966, p. 1026.

Dunning, G. J., et al., "All-Optical Associative Memory with Shift Invariance Multiple-Image Recall," *Optics Letters,* 12, 1987, pp. 346–348.

Gibbs, H. M., *Optical Bistability: Controlling Light With Light*, Academic Press, New York, 1985.

Goutzoulis, A. P., et al., "Optical Processing with Residue LED/LD Lookup Tables," *Applied Optics*, 27, 1988, pp. 1674–1681.

Guilfoyle, P. S., and W. Jackson Wiley, "Combinatorial Logic Based Optical Computing Architectures," *Applied Optics,* 27, 1988, pp. 1661–1673.

Hamilton, M. C., and B. L. Powers, "Real-Time Joint Transform Correlator with Phase Conjugation for Distortion Correction," *Optics Letters,* 12, 1987, pp. 549–551.

Huang, Alan, "Architectural Considerations Involved in the Design of an Optical Digital Computer," *Proceedings of the IEEE,* 72, 1984, pp. 780–786.

Huang, A., "Optical Digital Computers," Proceedings of Supercomputing '89 in Reno, Nev., Nov. 13–17, 1989, ACM, New York, N.Y., pp. 446–449.

Jewell, J. L, et al., "GaAs-AlAs Monolithic Microresonator Arrays," *Applied Physics Letters,* 51(2), 1987, p. 94.

Lentine, A. L., et al., "Symmetric Self Electro-Optic Effect Device: Optical Set-Reset Latch," *Applied Physics Letters,* 52(17), 25 April 1988, pp. 1419–1421.

Lerner, E. J., "Computing with Photons," *Aerospace America*, October 1988, pp. 28–30.

Lillesand, T. M., and R. W. Kiefer, *Remote Sensing and Image Interpretation*, John Wiley and Sons, New York, 1979, pp. 442–486.

Mataloni, Raymond J., "Photonics in Electronic Warfare," *Journal of Electronic Defense*, December 1989, pp. 31–38.

Miller, D.A.B., "Quantum Wells for Optical Information Processing," *Optical Engineering,* 26(5), May 1987, pp. 368–372.

Miller, D.A B., et al., "The Quantum Well Self Electro-Optic Effect Device: Optoelectronic Bistability and Oscillation, and Self-Linearized Modulation," *IEEE Journal of Quantum Electronics,* QE-21, 1985, pp. 1462–1476.

Murdocca, M. J., et al., "Optical Design of Programmable Logic Arrays," *Applied Optics,* 27(9), 1 May 1988, pp. 1651–1660.

Paek, E. G., et al., "A Compact and Ultra-Fast Holographic Memory Using a Surface Emitting Micro-Laser Diode Array," *Optics Letters,* Vol. 15, pp. 341–343, 1990.

Palais, J. C., *Fiber Optics for Communications*, Prentice-Hall, Englewood Cliffs, N.J., 1984.

Prise, M. E., N. Streibl, and M. M. Downs, "Optical Considerations in the Design of Digital Optical Computers," *Optica and Quantum Electronics,* Vol. 20, 1988, pp. 49–77.

Psaltis, D., and M. Haney, "Acousto-Optic Synthetic Aperture Radar Processors," *Optical Signal Processing,* Academic Press, New York, 1987.

Ramanan, S. V., and H. F. Jordan, "Serial Array Shuffle-Exchange Architecture for Universal Permutation of Time-Slots," *Digital Optical Computing II,* R. Arrathoon (ed.), Los Angeles, 17–19 January 1990, SPIE, Vol. 1215, Paper 38.

Sahai, R., et al., "Multiple-Quantum-Well-Based Spatial Light Modulators for Electro-Optical Implementation of Neural Networks," to be published in the *SPIE O-E LASE Proceedings,* January 1990, Los Angeles, Vol. 1151, pp. 427–434.

Saleh, Bahaa E. A., "Architecture for Nonlinear Optical Image Processing," *SPIE,* Vol. 1151, *Optical Information Processing Systems and Architectures,* 1989, pp. 320–331.

Sawchuck, A. A., et al., "Optical Crossbar Networks," *IEEE Computer* 2-D, No. 6, 1987, p. 50.

Smith, P. W., "Applications of Optical Switching and Logic," *Philosophical Transactions of the Royal Society of London,* A 313, 1984, pp. 349–355.

Smith, P. W., "On the Role of Photonic Switching in Future Communications Systems," IEEE International Conference on Communications, June 7–10, 1987, pp. 1570–1574.

Tanenbaum, A. S., *Structured Computer Organization,* Prentice-Hall, Englewood Cliffs, N.J., 1984.

Tanguay, A. R., "Materials Requirements for Optical Processing and Computing Devices," *Optical Engineering,* Vol. 24(1), January/February 1985, pp. 2–18.

VanderLugt, A., "Signal Detection by Complex Spatial Filtering," *IEEE Transactions on Information Theory,* Vol. IT-10, 1964, pp. 139–145.

Wilkinson, J. H., *Rounding Errors in Algebraic Processes*, Prentice-Hall Series in Automatic Computation, Englewood Cliffs, N.J., 1963.