Using Planes of Optoelectronic Devices Interconnected by Free-Space Optics: Systems Issues and Application Opportunities

Haldun M. Ozaktas
Bilkent University
Department of Electrical Engineering
TR-06533 Bilkent, Ankara, Turkey

This work was originally prepared as an
AT&T Bell Laboratories Technical Memorandum
number BLO111680-940922-57TM dated September 22, 1994,
and released for publication on May 7, 1996.

May 1996

Abstract

The design of a computing machine takes place at several levels of abstraction ranging from materials and device engineering to system architecture to high level software. This system of levels of abstraction enables the design problem to be broken down into manageable subproblems, much as in a procedural programming language. On the other hand, it makes difficult the introduction of novel concepts and technologies such as optoelectronic device planes ("smart pixels"), which do not readily fit in the existing scheme of things. We try to develop an understanding of this system of levels of abstraction, why and how it resists the introduction of optical technology, and how one can modify it so as to successfully house optical technology. We argue that in the near future, optoelectronic technology can be successfully introduced if: i) changing technology or applications create a significant bottleneck in the existing system of levels of abstraction that can be removed by the introduction of optical technology (e.g. interconnections, memory access); ii) special purpose applications involving very few levels of abstraction can be identified (e.g. sensing, image processing); iii) it is possible to modify a few levels of abstraction above the level that optical technology is introduced, so that the optical technology is smoothly "grafted" to the existing system of levels of abstraction (e.g. modifying communications schemes or standards so as to match the capabilities of optical switching systems, employing parallel architectures to match the parallel flow of information generated by optical subsystems).

Hybrid integration of self-electrooptic effect devices on silicon integrated circuits is a promising optoelectronic device plane technology in which an array of optical "pinouts" is provided for the electronic circuits. Free-space optical interconnection technology allows these planes to be connected in a variety of ways. Among these, the multistage architecture is found to have many desirable features, but many alternatives remain unexplored.

Image processing is a significant application area in which optoelectronic technology can have a significant impact because the two-dimensional nature of the data, the global nature of important classes of operations, and the need for real-time pipelining and/or iterative processing naturally match physical architectures that can be efficiently implemented. Image processing operations are often classified by application. After reviewing these, we provide a classification based on the information flow required to realize these operations. Different degrees of information flow require different kinds of connections. By comparison with recently developed multiple active layer electronic technology, we argue that optoelectronic technology offers an advantage when a global pattern of connections is necessary.

We identify several tradeoffs and system level considerations which have not yet been fully clarified. One of these is the issue of trading speed for resolution; that is, using spatial multiplexing techniques to process larger images. Another is the issue of how to successfully assemble many device planes in parallel so as to be able to construct larger systems. We believe that it is important to break the limits imposed by a single optical aperture or a single chip.

Optoelectronic systems such as memory with parallel access, state machines, matrix processors, neural networks, etc. may represent more realistic challenges as compared to general purpose systems, for short term development and validation of optoelectronic device plane technologies. These systems do not involve too many levels of abstraction (which makes their conception possible), often involve regular patterns of information flow (which leads to simple physical architectures), and usually result in an interconnection bottlenecked system when implemented with purely electronic technologies. The major challenge with such a system is to either successfully interface it as a subsystem of a larger (possibly general purpose) system in a way that benefits from its high performance, or to find a special purpose application where it can directly exhibit its high performance.

We identify and define two directions for further research which we believe are worth investigating not only for their own sake, but also because they will provide a platform on which several system level issues can be clarified and the optoelectronic technology validated. The first is an optically interconnected multiprocessor computer. Although this is not a new idea, SEED on silicon technology makes it particularly attractive. The second is a multistage programmable iterative real-time image processor. The strength of this concept lies in the excellent mapping of computational requirements onto a physically convenient architecture.

Contents

1 Introduction

The integration of larger numbers of primitive computing elements (switches, transistors, gates, processors, etc.) to produce computers of greater processing power requires the use of interconnections with greater length/width ratios.[1]

As the length of an interconnection is increased, the time it takes for a signal to propagate to the other end also increases, at least as much as dictated by the speed of light. While this limitation holds for all types of interconnections, normally conducting electrical interconnections have much more severe limitations. The signal delay is a quadratic function of the length/width ratio beyond a certain length/width ratio, since the line becomes too lossy to allow pulse propagation. The energy per transmitted bit also increases with line length, even when repeaters are used [5, 6].

For these and other reasons[2] that have been extensively discussed in previous work, it has been suggested to use optical interconnections for implementing the longer connections in computing systems, especially when an electrical line to be used instead would have a high length/width ratio. (See [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and the references therein.)

Here we discuss various architectural and application oriented issues within the framework of a particular technology allowing hybrid integration of optical and electronic devices, namely flip-chip bonding of self-electrooptic effect devices (SEED's) on silicon [11]. Our discussions will mostly be of a sufficiently general nature to be applicable to other similar technologies in which optical inputs and outputs can be distributed essentially arbitrarily over electronic circuits manufactured under a mature platform (such as CMOS).

This is a working document so that the degree of completeness and refinement of the various sections is not uniform. Also, the various categories, classification schemes, and terminology introduced is not definite and subject to improvement.

2 Introducing optical technology into computing machinery

2.1 Levels of abstraction in problem solving

A lot of attention has been paid to determining at what level of the interconnection hierharchy optics should be employed (board to board? chip to chip? etc.). On the other hand, it seems that little attention has been paid to the issue of determining at what level of abstraction of the computational process optics should be introduced.[3]

To understand what this means, let us reflect on how one usually accomplishes a task by using a computer. Say that we wish to modify a high resolution image in some way so that it is more pleasing to the eye. It might be possible to come up with certain operations involving convolutions, matrix operations etc. that enable us to do this. These mathematical operations will have to be broken down into repetitive or recursive sequences of more elementary operations. In doing this, we will have developed an algorithm for solving the problem. The algorithm will be written down in the form of a high level programming language, which will be translated into assembly language, which will be run by a microprocessor, which is essentially a high level logic system, which is made up of lower level logic functions (such as shift, add, etc.), which are made of gates, which are made of transistors.

Given a certain number of transistors or logic gates, there is no reason to think that this way of doing image enhancement is optimal, but at least it is possible. Realization of the techniques associated with each level of abstraction can be posed as self-contained problems which can be solved by specialists, with some care being necessary to ensure successful interface to the levels immediately above and below. Some

[1] This can be avoided by resorting to architectures with local connections only, but for problems which intrinsically require global flow of information this merely amounts to breaking down the necessary long distance communication paths into a large number of short hops, which is not necesarrily optimal [9].

[2] E.g., the possibility of nonplanar interconnections, voltage isolation, very little or no frequency dependent crosstalk and distortion, no impedance matching problems even with multiple taps, etc.

[3] It will be evident that the two issues are not unrelated, the levels of abstraction to some degree corresponding to the levels in the packaging hierarchy.

degree of optimization within each level is usually performed, but it is often not possible to optimize over several levels. No central committee has ever decided on what these levels of abstraction are either; they are the outcome of historical developments.

Device physicists try to minimize the switching time and energy, and computer scientists try to minimize the consecutive number of steps required for the completion of a task. This suggests that the combined effort of both camps will result in an optimal machine, but closer examination reveals that things are not so simple. For instance, perhaps it is the case that for larger systems, performance saturates with increasing device speed, and devices beyond a certain speed offer no further increase in performance.

The controversy over the relative virtues of global and local computation cannot be resolved unless optimization over several levels of abstraction are performed (see [9] and references therein.) Globally connected systems allow fewer steps of computation but result in longer duration per step, whereas locally connected systems require a larger number of steps with shorter duration per step; these considerations being closely related to the choice of algorithm, architecture, and interconnection media. To find the optimum degree of globality or locality, one must optimize jointly over possible algorithms and physical realizations of the machine.

The difficulty of introducing optical technology despite its clear fundamental physical advantages can to some extent be explained in the light of the above discussion. If we had a theory of computing which allows joint optimization over all levels of asbtraction, we could throw in the possibility of optical interconnections and switching into the parameter space. Then, given a computing task, we would perform the optimization, which would not only clearly indicate whether and when we should use optics, but also the architectures and paradigms that must be used. Since we cannot do this, we instead try to show that, say, a globally connected interconnection network is faster if implemented optically. But what if a locally connected network, which can be implemented electrically, allows the same task to be done in overall less time by running a different algorithm? (We have argued that this is not the case in [9], but not definitively.)

2.2 Alternative systems of levels of abstraction for general purpose computing?

It is clear that a very fast, large, and low switching energy array of optical switches or "smart pixels" has tremendous computing potential. However, it is too difficult a task to start from this array and arrive at a general purpose system in a single leap. If we are interested in designing a general purpose computer, we must guide our efforts by some system of levels of abstraction. It is first necessary to show how certain elementary functional units (in the abstract sense) can be formed, and then how these can form higher level units and so on, until we arrive at some kind of high level "programming language" enabling the problem description to be formulated.[4]

We could try to come forward with a new system of levels of abstraction complete with the techniques necessary for realization of each level, and then build machines including optical components based on this system. Taking the array of optical switches as our starting point, and without being biased by the mainstream system of abstraction, we may try to work our way up to the level of problem description.

Some alternative systems of abstraction do already exist, such as cellular automata, connectionist systems, and most significantly parallel computing. There is some reason to think that these might house optical technology better, but unfortunately these "paradigms" (which differ from the mainstream in varying degrees) are not that well developed. For instance, the techniques for only the lowest levels of abstraction are developed for cellular automata; nobody has a high-level programming language which they can compile into some kind of "assembly language" which will run on some kind of cellular automata

[4] It should be clear that the burden of providing a higher level platform, which must rest on intermediate level platforms, belongs to whoever provides the computer. People will certainly be reluctant if they are presented a multi-stage array of programmable optical logic devices, no matter how fast or large, if we cannot show them where to plug the keyboard and monitor, and where to buy a C Language compiler. Without any registers, accumulators, microprocessors, assembly or C Languages, no user will want to program or configure their systems at such a low level.

hardware (which consists of several levels of abstraction down to the level of a single cell). The state of development of techniques for doing things with cellular automata is comparable to that of low level logic in the mainstream system, such as shift registers etc., and are not even at the level of a data flow architecture, let alone a microprocessor.[5]

In conclusion, it seems difficult to come forward with a general purpose optically interconnected computer based on such novel paradigms; the development of the mainstream system of abstraction having spanned at least a century.

2.3 General purpose systems versus special purpose systems

Unlike general purpose systems which can be programmed to do any task with reasonable efficiency,[6] special purpose systems are "hardwired" and can do only certain prescribed tasks. The more general purpose a computer is—that is, the greater the range of tasks it can do with reasonable efficiency—the greater the number of levels of abstraction it has. The more special purpose a computer is, the fewer (for instance, the abacus does not have many). Thus, special purpose systems provide a better opportunity for new technologies (such as optics) and underdeveloped paradigms (such as cellular automata).[7]

Midway between the extremes of special purpose and general purpose systems we can identify a class of systems which we may refer to as "quasi-general purpose" systems or "coprocessors." Such systems can perform a certain class of operations of general utility, such as math coprocessors or digital signal processing chips. Of course, the full picture is that there is a continuum of systems of varying degree of "general purposeness" between the two extremes of special purpose and general purpose systems.

The "programming" of a general purpose system can take place at various levels. A microprocessor is a custom designed chip which can be programmed at a fairly high level (assembly language). On the other hand, a system to do the same task can be programmed at the much lower hardware level, for instance, by customizing a gate array. Both the microprocessor and the gate array can be viewed as general purpose systems which can be programmed to perform special purposes; the difference is in the nature of the programming and the level at which it takes place.

Under the light of what has been discussed until now, it is no wonder most successful optical systems to date have been special purpose systems. Such systems can be designed to perform a certain task standing alone, or they might be designed as a self-complete component of a larger computing machine. There is nothing complicated with the former. As a very simple example, an array of optical devices might be used as an image amplifier in a medical imaging system. The intricacies involved in the latter case will be discussed later.

[5] It has been shown how to simulate conventional logic operations in cellular automata, so that one can in principle do anything with a cellular automaton that one can do with conventional logic. However, this is a meaningless approach if the cellular automata is implemented using logic gates in the first place, or simulated on a workstation. But things may change if cellular automata are implemented by virtue of some atomic scale physical phenomena etc.

[6] It does not take much to be able to do any task, if one allows for gross inefficiency.

[7] It is quite conceivable for a limited number of, say, image processing researchers to start from a description of the capabilities of an array of optical devices and devise algorithms and methods for performing tasks they are interested in. (Many researchers were interested when the systolic computation paradigm was introduced for VLSI systems, unveiling a new class of solvable but unsolved problems.) The key issue seems to be that it should be possible for a single group or working unit to be able to obtain fruitful results by themselves, since this will give them the incentive to attack the problem. On the other hand, the effort towards the general purpose system would require a much bigger effort, requiring strategic commitment by a larger institution.

Researchers and engineers make carreers out of solving the problems associated with a certain level of abstraction in the mainstream system. They will not be willing to change their focus easily, since within the present system, the people working at the lower level are providing them with the technology to realize their stuff, and the people at the higher level want the stuff to realize whatever they are doing at their own level. No one will benefit from change unless everybody changes at once. This is a particularly severe kind of "chicken-and-egg problem," since it is not regenerative (that is, it will not by itself change for the better once given a sufficient but small initial momentum). On the other hand, there are always people willing to work on special purpose systems, which due to their limited number of levels of abstraction can be handled self-completely by a single person or group. Thus, successful exposition of the capabilities of optical technologies to the image processing and computer science communities may be rewarding.

An issue which perturbs these considerations is the fact that in areas of academia where theoretical achievements are valued, the interest of researchers may be independent of whether they can interface with upper or lower levels of abstraction.

2.4 Introducing optics into general purpose systems

Although we have seen that special purpose systems provide a conceptually simpler opportunity for optical technologies, we wish to explore how optics may be introduced into general purpose computing systems as well. Since we have seen that it is very difficult to come forward with an all-together novel system of levels of abstraction which would house optical technology in an efficient way, it is clear that general purpose computers will be mostly based on the mainstream system of levels of abstraction (which we might be able to modify to a limited degree).

First, let us consider modifying only the least abstract level of problem solving, the level of physical devices, wires, etc. In this approach, we start with the mainstream architectural and packaging paradigm and see whether it is possible to make a "better" machine by using optical components (interconnections and/or switches) instead of some of the electrical ones. Examples of this approach might be the introduction of optical backplanes or chip-to-chip modules instead of their electrical counterparts, while leaving the architectural conception and logical structure of the machine intact. This would change the job of the device physicist[8] and the person who designs the physical packaging, but would not affect people working at higher levels of abstraction, including those contemplating the logical and systemic architecture of the machine, as well as those providing the software.

This approach is appealing in that we do not have to worry about the development of new architectural concepts. However, there is no reason why the existing concepts should be particularly congenial to optical technology. In fact, they have historically developed to benefit from the strengths and accomodate the weaknesses of electrical technology, which are in some senses complementary to those of optics, so that this approach may not bring out the best of optical components. (VLSI architectures which try to minimize the length and number of chip to chip interconnections provide a good example.) Nevertheless, this may still be a valid and promising approach because it seems that replacing the longer wires with optical links does indeed result in a net advantage, even in existing systems.

If instead of the above simple approach, we wish to modify higher and higher levels of abstraction with the hope of better utilizing the particular optical technology at hand, we must face and overcome certain difficulties. For instance, we may attempt to replace a complete electronic combinatorial or sequential logic unit with an optical one which provides the same functionality, but in a "better" way. The interior structure and levels of abstraction of the optical unit may be entirely different, but it must interface with the system of levels of abstraction of the machine in which it is embedded at a certain level.

At relatively high levels of abstraction we might contemplate an optical microprocessor or digital signal processing coprocessor. At yet higher levels the physical and logical architecture of the machine will be altered significantly to suit the strength of the optical technology. For instance, we might contemplate an optically interconnected parallel random access machine where the processor locations and algorithms are designed so as to match precisely the type of connection patterns that can be provided by optics.

Modifying the system at higher and higher levels of abstraction so as to better suit the optical technology becomes an increasingly difficult task as we move upwards because of the need to maintain continuity between the different levels. If the central processing unit of a machine works on 32 bit wide words, its replacement must also work with 32 bit words. (It must be "plug compatible.") As another example, if we are to replace the existing electronic memory with an optical one, the input-output characteristics of the new memory must match those of what is being replaced. Notice that this requirement may sometimes resist improvements. A new optical memory, which provides much faster parallel access, may offer no system improvement, since the system in which it is embedded may not be able to utilize it. This makes it difficult to justify the optical technology, since the potential increase in performance offered by the optics cannot be utilized in this case, while its usually greater price will have to be paid.[9]

If we cannot succeed in getting a successful interface at one level, we might have to move up to a

[8] More precisely, it would create jobs for some device physicists while eliminating jobs for the others.

[9] On the other hand, perhaps an optical processor can be used to perform, say, some kind of parallel search on the data read from the optical memory, a feat which would be very expensive or slow with an electronic memory. The question now is whether the next higher level of the system can beneficially use the results of this fast parallel search. The answer would probably be yes if the search query as well as the result consist of small amounts of data.

higher level and try our chance at that level. By modifying this higher level (which may or may not involve optical components), we might be able to exploit the higher parallelism or bandwidth offered by optics. If not, we might have to move another level up, until the the intrinsic advantages of the optical technology seep through to the surface and translate directly into a user level performance advantage (such as getting the job done in less time).

This discussion should also clarify what is meant by doing something "better." Doing some intermediate level operation cheaper, faster, larger, etc. by introducing certain modifications at that level do not automatically result in user level improvements. It may be necessary to make further modifications at higher levels, until the fastness, cheapness, etc. can seep through to the user level (which is the highest level).

Modifying the system at a certain level of abstraction might correspond to introducing an optical subsystem (such as an optical logic unit) into the machine. but this need not be so. Remember that replacing all of the electronic switches and wires with optical ones does not alter the system of levels of abstraction at all, although we now have a computer consisting entirely of optical components. Alternatively, we may modify the architecture of the machine drastically, without introducing any optics at all. Despite the fact that the interconnection and packaging hierarchy often mirrors the levels of abstraction, the two concepts are distinct and must not be confused.

Before drawing some conclusions, we discuss a few examples to make the content of the last few paragraphs more concrete.

High-bandwidth "transparent" photonic switching Research in guided-wave wideband switching networks has resulted in rather impressive switches whose various strengths and weaknesses are not exactly matched to the requirements of existing multiplexed switching networks, so that it seems they may find less application than originally hoped for. The weakness of these switches is that they have a limited number of spatial channels. Their strength is that they can route very high-bandwidth signals transparently. Efficient use of this bandwidth cannot be made if bitwise multiplexing is employed, since these systems cannot switch at a rate as high as their transmission bandwidth. However, if we make the higher level modification of employing large-size block multiplexing instead of bitwise multiplexing, we can walk around this disadvantage. Now we must face the issue of whether the use of large-size block multiplexing is compatible with the next higher level of abstraction (which might be that of communication protocols and transmission standards). If not, we may try to push forward by suggesting modifications to the protocols and standards. If we do not arrive at a clear advantage within a few levels, we might have to give up.

Two-dimensional digital optical image processing Optical technology will probably make it possible to construct image processing subsystems which can perform two-dimensional signal processing operations (such as the discrete Fourier transform (DFT), convolutions, etc.) in parallel at a very fast rate. Given the fact that digital electronic hardware is extremely strained to perform such operations of even moderate complexity, it initially seems that digital optical signal processing coprocessors would have much to offer. However, it is not immediately clear how such an optical coprocessor can be interfaced to the rest of the system. Setting up the two-dimensional data serially from conventional electronic memory may largely nullify the potential advantages of such a system.[10] The limitation is actually that of the electronic processing system as a whole, which cannot handle larger amounts of data in parallel, not of the optical coprocessor. But the bottom line is that it may not be possible to improve overall performance by simply replacing the coprocessor, because the rest of the system is not good enough to take advantage of the increased capacity and speed. (One should not exclude the possibility that in some cases the replacement might indeed prove beneficial, despite the bottleneck involved in serial transfer at the interface, or there may be no interface problem because the data is already in optical form (coming from an optical memory or natural image). Nevertheless, it is likely that in most cases the capabilities of the optical subsystem will be largely underutilized due to this interface problem.)

[10] This has been referred to as the "fire hose problem."

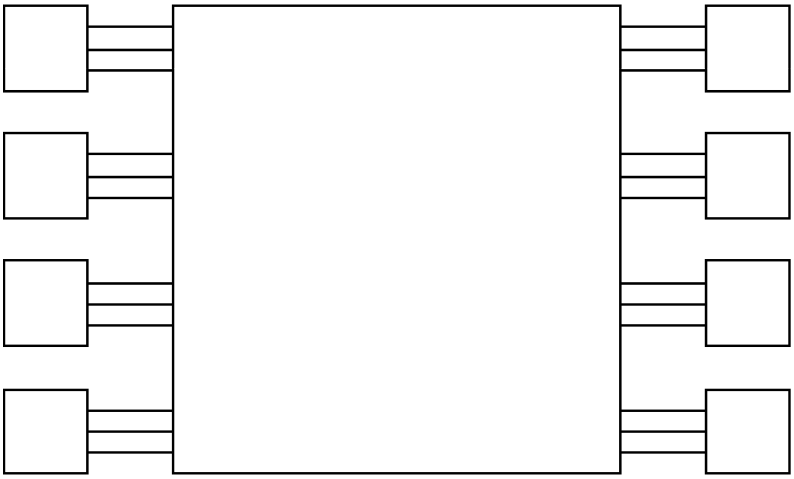

Figure 1: The large block in the middle represents global operations on the whole image. The smaller blocks on the left and right represent local operations on parts of the image.

Does this mean there is no possibility of employing optical technology short of contemplating an all optical system from scratch, which we argued was a very difficult task? Not necessarily. Some modification in the higher level design of the system may enable a successful interface. The strength of optics in this case is that it can provide the interconnections necessary for the global flow of information, which electronics cannot. The strength of digital electronics is that it can provide complicated operations in a small space, which optics cannot. Thus, successful partitioning (or "factorization") of the overall problem to match the strengths of both technologies may lead to an architecture of the kind depicted in figure 1. The big block may represent an operation requiring regular global interconnections (implemented optically), whereas the smaller blocks represent digital electronic processing. The many smaller blocks on the left work on parts of the data independently in parallel and feed the optical subsystem in parallel, so that there is no serial bottleneck. After performing the necessary operations, the optical subsystem distributes the large array of data to the several digital processors on the right for subsequent processing.

In conclusion, we see that beneficial use of an optical subsystem may require integral redesign of the system architecture at one or more levels above that at which the optical subsystem was introduced.

Parallel memory access Recent developments promise high speed parallel random access of huge amounts of data from silicon as well as optical memories. The former involves optical devices integrated with silicon, whereas the latter involves transmissive or reflective readout from optical storage media.

The considerations here are similar to those in the preceding example. Although the large archival storage capacity of optical memory can be utilized without difficulty, its potential for fast parallel readout may not, unless system architectures are designed in a way that make this possible. Once again it seems that parallel architectures in the spirit of that suggested in the previous example might be useful. Alternatively, some form of optical processing may be used to distill the large amounts of data read from the optical memory, returning a small amount of information that can be handled by the electronics at the higher level. This might be useful in database applications.

Photonic digital (logical) switching Digital photonic switches are essentially optical/optoelectronic logic circuits, often based on multi-stage switching network architectures, which enable a given pattern of connections to be established between a large number of incoming and outgoing communications lines [20].

Since the large number of incoming and outgoing lines coming from distinct locations are simply bundled together to form the "fire hose," the interface problem discussed in the image processing example does not arise in this case. This makes optical technology particularly suited to this application.

2.5 Some conclusions

In conclusion, optical technology can be beneficially introduced into a general purpose computer if we can come up with modifications to a system involving the use of optical components in such a way that the intrinsic advantages of the optical technology result in user level improvements.

A particularly transparent special case is to replace an electronic subsystem with an optical one of identical functionality, such that the user level figure of merit is improved by the change in subsystem operating parameters (cost, speed, number of channels, etc.). The internal structure of the optical subsystem may be completely different from the electronic one.

The simplest special case of the above is to replace certain electrical switches or wires by optical ones without otherwise modifying the system. Examples are optical backplanes, fixed free-space interconnections between circuit boards, etc. (In this case, no change is made to the existing system of levels of abstraction. A backplane or free-space interconnection system is not a subsystem in the sense of the previous paragraph.) This may not be the best way to utilize optics though, so that dissapointments in this approach should not be used to judge the potential of optics.

However, it is very difficult to do something useful with an optical technology (despite its large speed and parallelism), if it cannot be interfaced to a certain level of abstraction of an established architecture in a seemless manner. Otherwise the user level problem description is too many levels away to make optical device arrays or the like useful for general purpose applications.

On the other hand, special purpose applications where the product can be provided in a form which can be directly used without requiring any low level "software" development by the user or third parties are clearly promising. Examples might be integrated optical detection planes for image preprocessing, dedicated image processing functions, "smart" optical sensors, etc.

We will not aim at providing a comprehensive elaboration of the general philosophy outlined above, but will rather concentrate on a certain optoelectronic hybrid integration technology, that of flip-chip bonding of SEED's on silicon. Of course, similar considerations apply for other technologies exhibiting similar characteristics.

3 Optoelectronic integration technology

In this section we outline the essentials of a particular technology (or set of technologies) which we believe will serve as a useful platform to discuss the opportunities for optics as well as to implement them in the short term. We also discuss certain architectural and system level issues and alternatives.

3.1 Hybrid integration of SEED devices with silicon

Initially, self-electrooptic effect devices (SEED's) were constructed as bistable circuits which could be used as latches and logic gates. This was true of the older two-terminal devices as well as the newer three-terminal devices, including the symmetric SEED arrays (S-SEED's). Later, FET-SEED technology [12] was developed as the interest in cointegration of transistors and optical devices increased. This interest was driven by the needs both to reduce optical input power (through transistor gain) and increase logical complexity (through transistor logic). A certain degree of non-trivial functionality, such as Boolean logic, is also possible between the pixels of S-SEED arrays, but not comparable to what can be done with transistors [13]. The monolithic FET-SEED technology may ultimately provide high-performance and low cost optoelectronic circuits. In the short term however, it seems wise to capitalize on the very advanced state of silicon technology by hybrid flip-chip bonding of SEED's on conventional silicon chips[11] [11]. This offers great flexibility in that we can design any kind of electronic circuits around the optical

[11] The relative importance of the monolithic and hybrid technologies depends on the extent that systems want very high logical complexity (e.g. number of transistors) or very high device performance (e.g. clock frequency).

devices. The technological possibility of being able to combine substantial amounts of optical inputs and outputs, with respectable performance, with essentially any complexity of silicon raises entirely new system possibilities. Indeed, it is one of the purposes of this paper to explore what these new systems might be.

The optical devices (the detectors and modulators) will probably will laid out on a regular array for simplicity and solder bonded to regularly spaced pads provided on the silicon. There are a number of different ways of looking at such a system. First, we can look at it as a conventional silicon circuit with optical input and output ports provided on a regular array. While shorter connections are established electrically in the usual way, longer connections may be provided via the nearest optical modulator and detectors.

An alternative view is that of looking at the system as an array of "smart pixels", each of which exhibit considerable functionality made possible by the electronics below. The electronics associated with each pixel may constitute isolated electrical islands, but we may also have pixel to pixel connectivity in the silicon if desired. If we consider the defining quality of smart pixels as their regular periodic structure, then we may consider them as "gate arrays" of silicon circuits with optical ports. The term smart pixel is artificial and exists for historical reasons, deriving from the evolution of the idea from optical logic arrays ("dumb pixels"). What is a smart pixel array and what is just an optically interconnected chip is perhaps best decided by the application. (It is also interesting to contrast these systems with spatially continuous (unpixellated) spatial light modulators such as those based on liquid crystals etc.)

Currently it seems that an optical input-output can be placed within every $50\,\mu$m $\times$ $50\,\mu$m square, set by requirements of the soldering pads, with the size of the optical devices themselves being about $10\,\mu$m. A possible 2cm chip might then contain 400×400 "pixels." The density of pixels attained seems close to that predicted as optimal by several system level studies [7, 10]. These figures also seem to be consistent with the number, separation, and spot size of the optical beams that can be generated and manipulated with the required quality.

Yield considerations limit the size of the chip and thus the total number of pixels. Currently about a thousand to ten thousand pixels seems possible.[12] These devices can be expected to switch in less than a nanosecond, and the energy per optical event can probably be made small enough (less than a picojoule) that it can be ignored in comparison to the energy dissipated in the silicon. If operated at the contemplated fast rates, it will be necessary to mount these chips flat on a heat sink [14].

One interesting prospect for this (or FET-SEED) technology is that of patterning holographic optical elements directly on top of the optical devices, by introducing a few additional steps to the planar process [15]. A few layers would be needed to emulate a thick hologram with single diffraction order. One can imagine these integral optical elements being used in conjunction with a single mirror above the plane of the devices.

A number of issues such as single rail versus dual rail encoding, transmissive versus reflective devices, light sources versus modulators, and associated beam combination are discussed in [20]. It seems that there are benefits to working with dual rail encoding, reflective devices, and modulators as opposed to their alternatives. Dual rail encoding allows insensitivity to parameters that may drift across the system, reflective devices are apparently easier to fabricate and allow for heat removal to take place from the other side, and modulators are preffered for a number of device based reasons.

We will refer to a silicon chip furnished with optical inputs and outputs as an *optoelectronic device plane.*

[12] One or two orders of magnitude more would be desired for image processing applications. However, the high speed of the devices compared with video rates might allow partitioning the image much as in the case of conventional electronic processing, or perhaps multiplexing sets of pixels together. (It is also interesting to speculate as to whether optics might somehow allow us to operate with technologies in which not every device has to work. If it could, we could break free of the size and scale limitations that absolute perfection impose. One might, for example, customize the optical interconnection pattern so as to bypass the defective units (or perhaps even customize at some higher software level). Such an idea is conceivable with arrays of identical, isolated smart pixels since parallel optical testing of the finite isolated pixels would be possible. Such approaches could allow very large systems not limited by yield or testing.)

3.2 Free-space optical interconnections

The various issues pertaining to free-space optical interconnections have been discussed in detail in [16, 17, 18, 19, 20], and will be only briefly touched upon here.

Space-variant or nearly space-variant connection patterns, or connections patterns exhibiting a certain degree of regularity or symmetry such as the Banyan, crossover, and perfect shuffle can be implemented with relative ease and high space-bandwidth product utilization efficiency. Other space-variant and irregular patterns of connections can be implemented directly with multifacet approaches, but in a space-bandwidth inefficient way. Arbitrary connection patterns can be obtained in a efficient way by employing a $\sim \log N$ stage architecture, but at additional complexity. Most desirable is to contemplate the higher level architecture of the system such that it relies on only a regular or space-invariant pattern of global connections. Many excellent examples of such architectures exist [20].

The manipulation of optical beams among the device planes introduces stringent requirements on the quality of the optics. Obtaining uniform distortion free spot arrays with a single lens over large extents is difficult; 64×64 arrays have been successfully utilized in systems [21], but it seems that larger arrays of even 100×100 pose a considerable challenge.

3.3 Heat removal considerations

Heat removal is one of the most significant issues for large three-dimensional systems. Various considerations suggest arraying all device arrays on a single heat sinking plane and providing optical interconnections on the other side of the plane [14, 23].

3.4 Physical architectures for optoelectronic systems

We now discuss architectural issues at a slightly higher level. We use the term "physical architectures" to distinguish our discussion from a discussion of logical architectures, which is at a yet higher level. The two concepts are nevertheless not unrelated; for instance, a multistage switching architecture maps naturally onto a multistage cascade of device planes.

It is simpler to imagine we are working with transmissive single rail systems with light sources, but we maintain the understanding that these can be converted into dual rail reflective systems with modulators without much difficulty.

Processing systems can in principle be designed by employing a continuum approach. In this approach we tailor the refractive index, nonlinear properties etc. of an optical medium such that it performs the desired operations on an optical signal. Since the design of such systems would be difficult, most practical design methodologies favor discrete thinking: wires and beams instead of waves, switches instead of continuous nonlinear media. In order to have a designable, manufacturable system, we discretize (and perhaps quantize) all optical signals, and segregate the nonlinear operations (realized as point operations) from the linear operations (realized through propagation and filtering). Despite such discretization, with a little imagination we can view the signals as "propagating" in some discrete space with certain rules of its own.

Computer architectures based on the optoelectronic device plane and free-space optical interconnection technologies described in previous subsections, can be classified based on several properties:

1. Whether they consist of a single device plane or many.

   A single device plane may receive its inputs in optical form and delivers its outputs in optical form, serving as some kind of optically controlled spatial light modulator. The inputs and/or outputs may be intrinsically optical (e.g. optical memory, natural image), or interfaced with other system units with optical inputs/outputs. Alternatively, the input or output may be read in or out electronically, in which case the device plane also serves as a transducer. In any event, all but minor processing is done electronically. Depending on the application, the optics preceding and following the device plane may be relatively sophisticated, perhaps performing various linear global operations.

Systems containing multiple device planes with free-space optical interconnections between them offer various architectural options, which are discussed next.

2. Optically interconnected array of device planes versus multistage cascade of device planes.

   i.) It is possible to simply array a number of device planes in some two or three dimensional geometry and provide optical interconnections among them. This presumes that some form of circuit is going to be partitioned between the device planes and that the chip to chip connections are going to be made optically.

   ii.) On the other hand, it is possible to cascade device planes in a sequence such that each has connections only with the immediately preceding and following planes. The pattern of optical interconnections between consecutive stages would most likely be regular patterns, since these are more efficiently implemented.

   This classification is somewhat skewed, since the first option is much more general, and includes the second as a very special case. Many other special cases corresponding to, say, systolic, cellular, or other paradigms could also be singled out. Other special groupings or connection topologies for the several device arrays can also be considered.

   The privileged status of the multistage architecture can be at least partly attributed to historical reasons. Conventional optical systems (e.g. telescopes) have the general form of planar or quasi-planar optical elements cascaded along an optical axis. This general form can be seen in analog processing systems, and also in generic digital optical computing schemes in which one or more device planes are connected in cascade with various optical elements, all aligned along an axis, with the output of the system tied back to its input in a form resembling a finite state machine.

   It may be beneficial to break this bias and explore other possibilities. The simple linear sequential arrangement is the simplest connection topology possible; a vast number of alternatives remain to be explored. Notice that here we are not talking about the connection topology between the pixels of two planes (which we presume will be one of the regular patterns discussed previously), but of the connection topology between the device planes. There seems much to investigate in terms of which topologies may be useful and at the same time allow a clever optical arrangement of the many bundles of connections among the many device planes. One approach would be to implement a processor on each device plane, and connect these processors to each other according to some multiprocessor interconnection network. The large number of space-invariant/regular channels among the connected processors will allow fast transfer of large amounts of data from processor to processor.

   It is worth noting that in any case, the use of reflective devices and heat removal considerations will most probably force all device planes to be arrayed on a single surface, even if logically we tend to imagine the multistage architecture in the form of a sequential stack of parallel planes. (This is actually true of the most sophisticated optical digital systems yet built [21, 22].)

3. Unidirectional or bidirectional information transfer.

   Multistage architectures are usually contemplated as a feedforward pipeline in which information flows always in the same direction through the sequence of planes. Alternatively, it is possible to allow connections in both directions, allowing greater flexibility in design.

   In both cases, it is possible to connect the last plane in the sequence to the first, so that the information can loop or iterate through the system (in the first case) or we have a cyclic system (in the second case). (A simple example of this is an optical "bus" connecting boards on a backplane.)

It is important to be aware of fully electrical quasi three-dimensional integration technologies. Multichip modules with the order of fifty interconnection layers allow a considerable amount of wiring space. More impressive is solder-bump bonding of the order of tens of active electronic chips [24]. The density of the chip to chip "vias" (spaced $\sim 0.2\,\mathrm{mm}$ apart) is comparable to that of the density of optical "pads" possible with the optical technology outlined above. Thus the number of out-of-plane "pinouts" offered by both technologies are similar. However, only local (right across) interconnections are possible with the electrical technology, whereas global connections are possible with the free-space optical technology. Furthermore, these electrical connections may have large capacitance or inductance and thus limited speed, compared to the optical connections that should be able to operate at high speed and bandwidths.

The device plane concept may be generalized to allow for such multilayer structures, be they multilayer wiring substrates or multiple active planes as described above. Optical devices would be mounted on the outermost layers of such stacks. The resulting compound device planes would be able to provide more silicon circuitry per optical pixel, should that prove desirable.

3.5 How much silicon per optical pixel?

One of the several general issues and problems that might be treated with respect to a particular application is the problem of adjusting the optical pixel density with respect to the electronic circuit density, assuming that we want to implement a particular pattern of connections, perhaps with a specified line length distribution.[13] In other words, how do we break a circuit into smart pixels? This problem is essentially the same as that treated in [5, 7, 10], so that one can arrive at specific conclusions for a particular situation by modifying the treatment in these references.

3.6 Analog versus digital

We distinguish between discreteness in space, time, and amplitude. In conventional optical processing, continuous spatial light modulators allow manipulation of spatially continuous images. The technology under consideration inherently introduces spatial sampling, even if the objects we are working with are continuous signals or images. Circuit based computations (e.g. logic) are inherently spatially discrete anyway. Although perhaps not intrinsically necessary, the optical technology under consideration will most likely impose temporal discreteness as well (clocked operation with short pulses).

What remains for us to decide is whether to work with signals that are discrete or continuous in amplitude (i.e. digital or analog). Both approaches are compatible with the technology.

One of the major factors hindering the widespread use of optical processing despite its enormous temporal and spatial bandwidth has been its low amplitude accuracy. If the new technology is going to overcome this weakness, digital operation is probably to be preferred. The high speed of the devices will allow quantized optical amplitudes to be bit serialized, without sacrificing any spatial parallelism.

3.7 Prospects of the technology

It will be worth restating some of our previous conclusions for the technology under consideration. The options for utilizing this technology seem to be as follows:

1. To use it for special purpose applications, if these can be identified.

2. To use SEED on silicon technology simply as a way of wiring up circuits designed under the conventional paradigm (low level modification). This is a special case of the next item.

3. To identify a component or subsystem of a conventional electronic computer which constitutes a bottleneck, in the sense that replacing this component or subsystem with one that is faster (or cheaper or can handle larger amounts of data etc.), will result in the overall computer to be faster (or cheaper or able to handle larger amounts of data).[14]

[13] The problem is colored by the fact that if we use a regular global pattern of optical interconnections, arbitrary irregular connections will have to involve local electrical "fine routing."

[14] One might propose that every existing computer has a limiting component or subsystem, which is true. Nevertheless, situations in which replacement of a component or subsystem would result in substantial overall improvement may not be commonplace, since the historical evolution of design concepts and technology has a tendency to balance the various components and subsystems in a way that no component is "overqualified" for the purpose it serves.

If such a component or subsystem is found, it would be beneficial to replace it with an optical one which exhibits improvements in the relevant characteristics. For instance, if the clock rate or power dissipation of a computing system is determined by the longest wires, and we can reduce the delay or dissipation along these wires by replacing them with optical channels (as suggested in the previous item), we can obtain a net improvement at the level of the overall computer. As another example, assuming that the speed of a computing system is solely determined by the memory access delay or the time it takes a coprocessor to invert a matrix, replacing these subsystems with optical ones may result in direct benefit.

4. If the conditions for successful application of the previous item do not exist, either because the optical technology is not directly compatible with the higher levels, or because its intrinsic advantages are buried at that level, modify the system architecture a few levels upwards with the hope of achieving the goal of the previous item (e.g. as in the example on page 7).

We have seen that the strength of the electronics is its flexibility in providing arbitrary operations and its weakness its inability to provide global connections. The strength of optics is precisely the weakness of the electronics. Thus, the beneficial use of the outlined optical technology is mostly dependent on whether a particular application can be efficiently decomposed (or “factorized”) into stages of globally connected operations.

4 General purpose systems

As we have discussed, replacement of electrical components by optical ones in a mainstream system architecture is a possible approach with a certain promise. The silicon chips with optical inputs and outputs can be physically arranged in many different ways. Several considerations suggest arraying the chips on a planar substrate which also serves as a heat sink [14]. Lenslet arrays or holograms immediately above the detectors and reflective modulators, together with an overhanging mirror or further holographic elements may provide the optical imaging. Less aggressively, the optical pinouts can be used to provide board to board connections between boards stacked on a conventional rack.

On the other hand, the development and utilization of new general purpose architectures is more difficult, despite the fact that various possibilities have been suggested or come to mind, such as pipeline processors based on cascades of optically interconnected arrays, parallel machines based on parallel random access machine (PRAM) models, etc. Optically interconnected parallel computers will be discussed further in section 10.1.

Apart from this, fully general purpose systems will not be further discussed here, although some of our discussions below might be relevant for such systems as well. We will first discuss special purpose systems, with emphasis on image processing. We will then discuss what we call “quasi-general purpose systems” (or “coprocessors” or “generic subsystems”), such as logic units, memory, matrix processors etc.

5 Special purpose systems: information processing

Special purpose systems by definition are hardwired to do specifically one or a few jobs. An image plane processor which differentiates the input image will be considered a special purpose system, whereas a processor which can multiply an image with any given matrix will be considered a quasi-general purpose system.

Special purpose systems can be classified as those which manipulate pure information, and those which interact with the physical environment. The former category is the subject of this section (in which we discuss special purpose systems by application area, with the major emphasis being on image processing), and the next section (in which we classify the necessary operations by their information flow requirements). The latter category is the subject of section 7. Quasi-general purpose systems will be discussed in section 8.

5.1 Image processing

The use of conventional optical systems for processing images has been widely suggested and to some extent realized. Such systems have usualy suffered from a lack of quality spatial light modulators and insufficient dynamic range. Both problems are alleviated with the digital technology under consideration.

Image processing applications are often classified as image enhancement, filtering, restoration, analysis, reconstruction, compression, etc. We first give a short overview of these, referring the reader to [26, 27, 28] for details. Afterwards, we move on to categorize the elementary operations used in these applications in terms of the optical architectures they might require, which is a more useful classification for our purpose.

The following list, which is strongly based on [27], should not be taken to be exhaustive but rather as representative of the mainstream of image processing techniques, development, and research. (Certain generalizations such as multispectral, color, or adaptive processing are not explicitly considered, but represent possible extensions.)

Image enhancement Operations used for image enhancement can be classified as:

1. *Point operations*: contrast stretching, clipping, thresholding, negation, level slicing, bit extraction, range compression, histogram modification.

2. *Spatial operations*: noise smoothing with linear or nonlinear (e.g. median) filtering, unsharp masking, low-pass, bandpass, high-pass filtering, zooming.

3. *Transform domain operations*: linear and nonlinear filtering in the transform domain.

Image filtering, restoration, estimation, and prediction The methods used in this area are:

1. *Linear filtering*: inverse filtering, Wiener filtering, interpolation, least squares and singular value decomposition methods, iterative methods, recursive (Kalman) filtering.

2. *Statistical and information theoretic methods*: Bayesian methods, maximum entropy methods.

3. *Geometric methods*: coordinate transformations and geometric mappings.

Image analysis and computer vision The procedures in this area can be grouped as follows:

1. *Feature extraction*: extraction of spatial features (amplitude and histogram), transform features, edges and boundaries, shape features, moments, texture.

2. *Pattern recognition*: matched filtering and detection.

3. *Segmentation:* template matching, thresholding, boundary detection, morphology, clustering, texture-matching.

4. *Classification*: clustering methods, statistical methods, decision trees, similarity methods.

Image reconstruction from projections This area involves operations such as the Radon transform and its inverse, back projection, as well as standard operations such as convolution, Fourier transforms, filtering, and (matrix) algebra.

Image data compression This currently very active field may be broken down as follows:

1. *Pixel coding*: temporal pixelwise pulse-code modulation, run-length coding.

2. *Spatial predictive coding*: delta modulation, differential pulse-code modulation

3. *Transform coding*: zonal coding, threshold coding.

4. *Interframe (temporal) coding*: adaptive spatial predictive coding, motion compensation.

Image detection and display Image detectors and displays constitute the interface of image processing systems with the real world. Useful processing may take place during detection or prior to display.

There has been some interest in developing smart image sensors inspired by biological examples. Animal eyes as those of humans have some local preprocessing, serving the purposes of mean luminance adaptation, contrast enhancement around edges etc. [29].

Smart sensor arrays that look at an image and draw some reduced data conclusions, such as for purposes of alignment, inspection, vibration monitoring, movement detection, autofocus, ranging, limited robot vision, or other simple task can be envisioned. As will be discussed in a later section, it is not clear that such local operations require the global connectivity of optics, but since the input is in optical form anyway, SEED on silicon technology may prove to be useful.

A simple, yet interesting example is the use of a smart pixel array to find a linear or nonlinear combination of a number of images, or the "distance" between two images in some vector space. Yet another might be, say, a medical viewing application where we enhance the image by using a smart pixel array. This would essentially be an electronic processor, but with optical input and output.

This application category may include optical memory readout and write systems as well.

Generalizations The above methods and operations can be generalized or refined in a number of ways, such as joint processing of multiple channels and color images, iterative, recursive, and adaptive processing, etc. Such extensions are not explicitly discussed but should be possible in most cases thanks to the flexibility afforded by the silicon.

For instance, iterative processing can be implemented by allowing several loops through the optical system, whereas adaptive processing would be possible by allowing each smart pixel to operate with different parameters which are updated as required by the algorithm.

5.2 Communications switching

One of the simplest switching problems is that in which we realize a one-to-one mapping (a permutation) of N lines onto another set of N lines in the desired manner. More generally one can desire the ability to provide connections that are not one-to-one. The optical technology emphasized here is more suited to logical switching, rather than transparent relational switching.[15] In most cases the desired functionality can be achieved by employing a multistage interconnection network. Details can be found in [20].

5.3 Sorting

Sorting N items with respect to a certain key is similar in nature to the permutation problem, and is also discussed in [20]. The sorting problem is important in that it is a basic operation frequently used by other algorithms.

6 Classification by information flow requirements

Image processing tasks are accomplished by performing various operations on the images. Above, the various operations were classified by application. Here, they are classified by optical architecture. We will identify several prototype operations and associated architectures which constitute equivalence classes in the sense that the difficulty and success associated with all members of a class are, or are expected to be, similar.

While considering these operations, we directly associate the array of optical devices with pixels of the image, although a certain amount of multiplexing or spatial division might be required if the sizes of these do not match. (Other alternatives include piping in space-time chunks of the image sideways; that is, aligning the temporal dimension of the image with one of the spatial dimensions of the hardware.)

[15] In such switches light is passively or actively routed but does not take part in any nonlinear interaction or switching itself.

We are also assuming that the number of optical pixels is of the same order of, or differs at most by a constant factor from, the number of pixels in the array to be processed. Thus, we exclude approaches where the number of optical pixels required are proportional to, for instance, the square of the number of pixels to be processed. This excludes multi-facet approaches [4] [19] to providing free-space optical interconnections.

It is also worth noting that in some cases the input image is transformed into an output image, but in other cases a certain amount of data reduction may take place so that some kind of feature vector, rather than an image, is produced as the output.

6.1 Type I: point operations

Point operations are those in which the value of an output pixel depends only on the value of the corresponding input pixel.

These were encountered in image enhancement and coding (individual pixel coding) and might also include algebraic and logical operations on a small number of images, such as image addition and subtraction[16], logical AND etc., and also amplitude feature extraction, temporal coding of arrays of data with no interpixel coding, and image detection and display.

Prototype operation: Nonlinear point operation on an image.

Prototype architecture: Single device plane with optical input and output. The silicon circuitry associated with each pixel is isolated from the others and performs the nonlinear operation, which may be space-varying, time-varying, adaptive, etc.

If several operations of this kind are to be performed consecutively, they might either be performed *in place*, or by pipelining the array of information through several consecutive device planes which are connected optically. In the latter case, the optics would simply image each pixel on any given plane onto the corresponding pixel on the next plane, creating an isolated pipeline for each pixel. Since the isolated pipeline for each pixel can be bunched or curled up within a single pixel, we conclude that optical interconnections are superfluous in this case. Even if we run out of silicon to do this on a single device plane, multi-active-layer electronic technologies can be used to construct the pipeline without any optics, since no cross connections are involved and the electrical interconnections would be short.

In evaluating the usefulness of optical technology in performing this type of operation, one should distinguish between two cases. That in which the data is available and/or required in electronic form and must be serially transferred to/from a two dimensional array, and that in which the data is available and/or required in two-dimensional optical form. In the first case, there does not seem to be a point in using any optics, as the task can be performed with an electronic array just as well. However, if the input and output of the system is to be in the form of a two-dimensional array of optical signals, then it might be useful to employ a smart pixel technology.

Thus, the question is whether and when the input and output would be in the form of a two-dimensional array of optical signals. If the rest of the system is a collection of units performing similar points operations, then the input and output will most likely not be such an array of optical signals, since the system would most likely be implemented electronically based on a similar argument as made above. However, if the rest of the system is performing global operations for which we will see optics is useful, then the data coming from and going to these stages will be in optical form, so that a smart pixel technology would be compatible with the rest of the system. Alternatively, the source of information may be intrinsically optical, such as in the case of a natural image or optical memory readout.

6.2 Type II: local operations

Local operations are those in which the value of an output pixel depends only on the value of the input pixels that lie within a reasonably small neighborhood of the original pixel.

Among these are most spatial operations used in image enhancement, most (but not all) of which can be expressed in the form of convolution or space-variant linear superposition with a relatively small

[16] Operations on two or more images might be performed by introducign them consecutively, or by interlacing them.

window function (smoothing, median filtering, low-pass, high-pass, band-pass filtering, differentiating, interpolation). Finite impulse response Wiener filters extending over small windows and recursive (Kalman) filtering are some of the filtering and restoration operations falling into this class, as well as various image analysis methods such as histogram feature extraction, edge detection, boundary extraction, thresholding, most morphological operations, and texture feature extraction. We can also mention matched filter detection of small objects, spatial predictive coding, pyramid image coding (by coding the difference in consecutively low pass filtered versions), and image detection and display as operations requiring local operations.

Prototype operation: Nonlinear function over small window.

Prototype architecture: Single device plane with optical input and output. The silicon circuitry associated with each pixel is connected to those within a small neighborhood and performs the specified nonlinear operation.[17]

Our arguments for point operations are applicable to this class of operations as well. More silicon is needed to perform the more complicated operations, as well as to provide the interconnections which will carry information from/to nearby pixels. Nevertheless, provided the neigborhood (window) size remains small, the basic argument remains valid.

As we consider operations which require information from greater and greater neighborhoods, this category merges into the next (global operations). Eventually, after a certain point it becomes useful to employ optical interconnections. We do not treat this case of intermediate connectivity separately, but subsume it in the following subsection.

Even in the case of natural light input, such as a smart image sensor, it is not clear that the SEED on silicon technology has anything special to offer. Since the processing can be done electronically, the only use of optics is for detection. Thus we must ask ourselves whether a SEED detector array is better than, say, a CCD array. Perhaps a CCD array smartened with some silicon circuitry might be preferable to the technology considered here.

6.3 Type III: global operations

Global operations are those in which the value of an output pixel may depend on the values of the input pixels across the whole image. We will classify such operations into three subtypes. For type A systems, we require it to be possible to decompose the operation such that it can be realized in a regular multistage system with a small constant number of stages. For type B, it should be possible to decompose the operation such that it can be realized in a regular multistage system with the order of $\log N$ stages.[18] Type C operations are defined as those in which the values of the output pixels may depend on the values of the input pixels across the whole image, and realization with a regular multi-stage system requires more than the order of $\log N$ stages. The extreme case would involve N^2 independent parameters to specify the relation between the input and output pixels, as in a matrix-vector multiplier or crossbar. (If there are only N degrees of freedom per stage, this would imply at least N stages.)

Orthogonal image transforms such as the discrete Fourier, cosine, Hadamard, Haar transforms etc., generalized linear filtering involving transform domain operations, convolutions and some linear superpositions with large windows belong to this class (mostly type B). Most methods of linear filtering (in particular inverse and Wiener filtering), restoration, prediction and estimation based on stochastic models (type B or C), with the exception of recursive filtering, but including most statistical and geometric methods, smoothing by splines and global interpolation (type C) also fall into this category.

Space-invariant filtering operations can usually be implemented on a type B system by employing Fourier transforms (by employing a digital “4f” processor). If we wish to implement space-variant filtering

[17] Area modulation is another (apparently inferior) approach to realizing convolutions with small windows [13].

[18] Here N is the number of pixels. Such regular multistage architectures with $\log N$ stages are well known as permutation switching networks [20]. The fact that these networks can be viewed as a distributed physical embodiment of the prominent divide-and-conquer paradigm of the theory of algorithms suggests that they should be useful for other problems also requiring global flow of information.

([27, page 292]), it may not be possible to realize this in $\sim \log N$ stages and a type C system might be necessary.

Also in this category are operations such as transform feature extraction (type B), geometrical and moment based shape feature extraction (type B or C), matched filter detection of large objects (type B), and image segmentation and classification (type B or C). Operations used for image reconstruction from projections are intrinsically global (type B or C), as well as transform domain coding, sorting, and permuting (type B). Certain classes of communications switches which require greater flexibility than provided by a permutation network may require type C architectures.

A particularly interesting possibility is the use of digital optical global transforms for transform domain image coding. In electronic implementations the image is split into blocks of moderate size since large DFT's are difficult to handle [25, 27]. Since larger blocks should yield better compression, it seems that optical technology should be particularly beneficial.[19] However, the 16×16 blocks that are currently used already seem reasonably close to the best that can be achieved with arbitrarily large blocks [27, figure 11.16], because the coding efficiency saturates as a function of block size. Thus, unless more advanced global coding schemes are developed, it seems that is not possible to make use of the global transform capability made possible by optics.

Prototype operation A: Nonlinear function depending on the value of a number of input pixels globally spread according to a restricted pattern.

Prototype architecture A: Small number of device planes connected consecutively with regular connection patterns such as the perfect shuffle, Banyan, crossover etc.

Prototype operation B: Discrete Fourier transform or one-to-one permutation.

Prototype architecture B: Same as in A but with $\sim \log N$ stages.

Prototype operation C: General space-variant linear superpositions (matrix operations), crossbar switching.

Prototype architecture C: A generic architecture is difficult to identify if we require that the space-bandwidth product is of the same order of, or differs at most by a constant factor from, the number of pixels in the array to be processed. (It does seem that an N stage architecture can do the job, since it has enough degrees of freedom, but this must be clarified.) Alternatively, a single stage is enough if we are willing to employ an optical system with space-bandwidth product much larger than the number of pixels, by using matrix vector product type architectures (also known as multifacet architectures). A discussion of these tradeoffs may be found in [17, 19].

It is almost certain that by using optics it is possible to supercede the capabilities of any existing or in fact non-existing electrical architecture [5, 6] for type A and type B operations. The same conclusion should hold for type C operations; however these are difficult to implement with any technology, so that it may be difficult to actually realize a system that can handle a respectable number of pixels.

6.4 Temporal locality and interframe processing

The above categorization, as well as the section on image processing applications, mostly concentrates on the processing of single image frames at a time. Given the speed of the devices and the possibility of pipelining, it is presumed that such individual frames can be processed at a fast rate. But until now we have mostly ignored operations which work on pixel values from several frames at once. Temporally local operations would involve pixels ranging over a small number of consecutive frames. An example in which temporal operations are of interest is MPEG motion picture coding [27], in which consecutive frames are compared in order to detect movement.

Such operations can be realized by providing sufficient memory in the silicon. In general it would most likely be difficult (and perhaps not so useful) to handle more consecutive frames than of the order of ten or so, depending on the application. In any event, we will presume that temporal operations are handled in the silicon, so that this extension need not be considered separately.

[19] Analog transforms have been suggested for the same purpose [30], but this approach would be difficult (if at all possible), due to the insufficient dynamic range of analog optical systems.

7 Special purpose systems: sensing and display

In this section we briefly discuss for completeness those special purpose systems which interact with the physical environment. Depending on whether they are influenced by or influence the environment, such systems can be referred to as sensors or displays. Sensing and display applications can be crudely (and nonexhaustively) classified as follows:

Metrology Measurement of physical quantities including temperature, pressure, liquid level, stress and strain, chemical concentration, position, motion or velocity, acceleration, electric or magnetic field, light and intensity, etc.

Industrial process control and manufacturing Alignment, positioning, velocity tracking, part inspection, stress analysis, vibration analysis, sample classification and identification, lithography, laser machining, etc.

Scientific research Identification and measurement of samples, monitoring processes, various forms of microscopy, etc.

Environmental and geophysical sensing Atmospheric monitoring, remote sensing with optical radar, etc.

Biomedicine Identification or processing of a specimen or sample for diagnosis, noninvasive imaging, laser surgery, etc.

Information processing Printers, scanners, monitors, cameras, optical memory systems, etc.

Most of the systems that can be employed for the above applications can be unified under the following generic model. In its most general form, this model consists of an array of light sources (or modulators) and an array of detectors. (In the simplest case, the arrays may have only a single element.) In some cases, the sources or detectors may not exist as devices, their roles being replaced by natural light sources, the human eye, or a piece of light sensitive paper, etc. We distinguish between three possibilities:

1. We may directly detect the light from the sources, considering any interventions as degradations, in which case our purpose is to measure the source, or some physical quantity altering or controlling the source (or sometimes some physical quantity altering the detector). Measurement of self-luminous sources, spectroscopy, and scintillation counting are a few examples.

2. The light coming from the source may be altered before it reaches the detector, in which case we measure the medium altering the light, or some other physical quantity altering the optical properties of the medium. In this case we are not interested in measuring the light, which is used as a means to probe the medium we are interested in measuring. The difference from the preceding item is that the light is affected after it is generated, not before. Optical radar, scattering measurements, most fiber optic sensors, and even optical memory can be listed as some examples.

3. We purposefully control the light so as to impose a certain pattern of light on a medium, or so as to impose a certain patterned physical effect on a medium which is induced by the light. Laser shows, lithography, laser surgery, and laser printing are examples.

Existing technologies that provide a basis of comparison for proposals based on SEED on silicon technology include CCD cameras followed by digital processing, CCD or other detector arrays with active electronics in their pixels ("smart sensors"), the imaging endoscope, and fiber optic sensors. In most of such technologies, the number of degrees of freedom of the source is small, although the number of detectors may be large. Usually, the source is a plane wave or unit resolution probe spot. The potential of SEED on silicon technology is that it can offer a similarly large degree of sophistication on the source side as it can on the detector side.

It is possible to imagine an interactive optical probe, which illuminates the area under investigation with a patterned source and picks up the reflected information on a detector array, allowing a larger amount of information to be collected, improving the performance of the sensor. In most current systems, the same objective is sought by scanning a single spot. Under certain situations, the use of arrays of modulators and detectors might improve accuracy and/or efficiency.

Low power patterning and printing, machining and surgical applications may likewise benefit from the flexible illumination possible with an array, instead of a scanned beam. With such a system it might be possible to do precision low power surgery (perhaps of the skin or eye), spatially controlled optically induced chemical reactions, micro tooling, cutting, or welding, etching, and lithography. Scanning array sensors and displays may in combination form the basis of a fast digital photocopying system.

8 Quasi-general purpose processors or coprocessors

As we have discussed before, it is very difficult to come up with a novel system of levels of abstraction that would lead the way to optimally designed optically interconnected or switched general purpose computing systems. However, it may be possible to introduce optical components within a subsystem provided we can solve the problem of how to interface this subsystem to the main system.

The terms "quasi-general purpose processor" and "coprocessor" are both somewhat unsatisfactory. Also, the boundary between special purpose processors and quasi-general purpose processors is not a sharp one. For instance, we can perform arbitrary convolution type filtering operations by employing a multistage architecture to take the discrete Fourier transform (DFT) of an image, multiplying it with the filter function, and then taking its inverse DFT. We have included convolution among image processing operations and thus classified it as special purpose, whereas non-convolution type linear operations are included in this section, classified as quasi-general purpose.

8.1 Memory with parallel access

Memory is an important subunit of any computing system, perhaps with the exception of very simple systems. Parallel access of large words or other chunks of data from electronic memories or optical disks may be useful for various purposes, such as database machines, pattern recognition systems, etc. It may be useful to feature error correction and decryption operations as an integral part of such access systems.

The considerations here are similar to those of section 6.2. If an electronic memory is going to be used, there may not be much point in using optical interconnections since parallel electrical transfer of information between several device planes seems possible [24]. However, if we are dealing with a very large electronic memory, the use of optical readout may prove to be the quickest way of getting the desired page of information to the desired processor. (We can imagine parallel optical readout from a large array of electronic memory chips followed by a switched optical concentrator.) Or, if an optical disk is used, the data will be in optical form anyway so that an optical write or read system would be considered. Also, if the data read (or to be written) is going to be processed in a way that requires global connections, optical write or read might be considered even with an electronic memory. (As an example, assume that we require the rapid delivery of a certain amount of information from the memory to a processor chip, but extraction of this information requires a relatively simple operation to be performed on a relatively large portion of the memory.)

8.2 Combinatorial logic units and finite state machines

These are systems which take a binary vector as input and deliver another binary vector (of not necessarily the same length) as output. If the output depends on past outputs, we are dealing with a finite state machine rather than purely combinatorial logic. A natural extension is to consider non-binary quantized amplitudes, in which case the system exhibits similarities to those used for image processing or matrix algebra (discussed below).

Murdocca has already shown how to employ a selectively masked multistage interconnection network with crossover connections between SEED logic gates to implement logic functions using a programmable logic array (PLA) approach [20]. Nevertheless, there seems to be much that has not yet been investigated in this area. The efficiency of multistage networks in providing arbitrary permutations is known. If we use two state logic gates instead of the two-by-two switches found in a permutation network, we get a logic fabric which can be configured to provide a wide range of logic functions. Both the analysis and design problems for such systems are open: What is the functionality that can be achieved in $\log N$ or so stages? How can a desired logic function be mapped on this regular structure?[20]

8.3 Matrix-vector multipliers and matrix processors

Matrix-vector multipliers implement the discretized version of the general linear operation

$$g(x) = \int h(x, x') f(x')\, dx',$$

expressed as

$$g_i = \sum h_{ij} f_j.$$

In the most general case the number of independent coefficients of h_{ij} is equal to the square of the space-bandwidth product of the input and output. If these coefficients are represented on a single plane in the form of a matrix, the resulting system will have a space-bandwidth product equal to the square of the space-bandwidth product of the input signal. This will limit the space-bandwidth product of the signals that can be processed. Multi-stage or systolic like approaches may help, especially if we are dealing with a restricted class of matrices.

More general matrix processors offering other functions including matrix inversion would most likely find useful applications in numerical analysis and processing. This is an area worth further investigation.

8.4 Neural networks

Several optical device planes with feedback and bidirectional information flow might be able to efficiently house neural networks. Neural networks and other connectionist architectures are claimed to be particularly useful for solving certain types of problems and intrinsically require high connectivity so that they seem well matched to the optical technology under consideration.

In most proposed optical neural network architectures, the required optical space-bandwidth product is dictated by the number of independent weights. Rather than trying to provide full connectivity with a weight associated with each connection (equivalent to a matrix-vector product), it is possible to employ some efficiently implementable interconnection pattern and a smaller number of weights. The associated optical architecture could be type A, type B, or type C, depending on the flexibility desired.

8.5 Cellular automata

Classical cellular automata or its generalizations may provide a new paradigm suitable for optics. However, two- or three-dimensional automata which can be realized with local connections only may not benefit from optics since they can also be implemented with multi-active-layer electronics technology [24]. It would be of benefit to generalize to non-locally connected systems to benefit from the strengths of the optical technology.

The author is unaware of a sufficiently complete theory that allows useful functions to tbe mapped onto such systems, making it difficult to speculate on the potential of this approach.

[20] The number of degrees of freedom of a system with $\log N$ stages is limited and thus so is the number of different functions that can be realized. A type C architecture, in the form of a crossbar, may be able to provide greater flexibility.

8.6 Subsystems with few inputs and outputs

In general one would expect a function with a small number of inputs and outputs to be implemented with a small number of components. However, in certain special cases it might be possible to come up with an efficient implementation involving redundant replication, outer product generation etc. of the data, followed by some fairly simple or regular processing in parallel, followed by reduction to the desired answer.

Such a subsystem would more easily fit into existing computing systems without requiring major architectural changes at the higher levels, since the parallelism of optics is exploited in an entirely transparent way, and does not lead to any interface problems.

One example is the optical method of correlation where the input function is replicated in shifted versions, multiplied by the replicated mask, and then integrated. Digital optical implementation should be possible by performing the spreads and integrations in $\log N$ stages. Searching a large database is another example. Assume that we have a certain number of subject terms and we wish to retrieve the entries containing this subject term. The input and output are small, but the search must take place through a large space. Yet another example is matrix-vector multiplication with a fixed matrix. It should be possible to increase the number of indivual examples, but more important is to find a general class of such problems which have some central significance.

When an optical solution of this kind is found, we must immediately inquire whether more efficient electronic implementations exist, since the small number of inputs and outputs suggest that it may possible to implement the desired system with a small number of components. For instance, systolic convolution or correlation on a linear array can be performed in the order of N time. If the input to the subsystem is arriving in serial manner, it will take this long to read it in anyway so that the optical method will not present any advantages. If, however, the input vector is available in parallel, there is a chance that the optical method might offer some advantages such as lower cost or greater speed.

9 Some system level issues

9.1 Trading speed for resolution

The optical technology under consideration, as well as several other similar technologies, presently cannot provide as large a number of pixels as one would like to work with. Although some applications might be found that match the available number of pixels and speed, for other applications requiring a greater number of pixels, such as image processing, methods of trading speed for parallelism should be explored.

The number of optical pixels that can be fitted onto a chip, as well as the number of optical beams that can be handled with small enough aberrations, both seem to be limited to about 100×100. On the other hand, typical images that one would like to work with might be 1000×1000. One option is to multiplex local groups of image pixels through a single optical pixel. If a video image at 100 frames per second has 10 bits per pixel, we would have to deal with 10^5 bits per second per optical pixel, which is quite manageable, and still leaves room for the possibility of iterative processing.

Of course, the silicon beneath the optical pixel will have to latch, store, and process 100 image pixels worth of information in conjunction with neighboring pixels. Channeling the information associated with several image pixels through a single optical pixel will also introduce constraints on the type of global connection patterns possible, an issue that has to be considered and analyzed explicitly for a given situation.

Another option, which must be seriously considered, is not to limit oneself to a single chip and single optical aperture. We return to this option next.

9.2 Increasing parallelism

The number of optical pixels that can be provided by a single lens with sufficiently small aberrations, and the amount of silicon circuitry that can be provided on a single device plane with reasonable yield, are limited. Currently it seems that it is possible to provide about a thousand pixels.[21]

This figure hardly approaches the advertised potential parallelism of optics. To increase the number of parallel channels, one can consider tiling several device panes ("chips") to obtain in effect larger device planes. Several optical components may likewise be used in parallel to increase the effective optical aperture. It is important to come forward with an optomechanical platform which allows extensibility in parallelism as well as pipelining. It is also necessary to devise schemes that allow global connections (e.g. a perfect shuffle or banyan) over the full effective aperture, and not merely the individual apertures (as the latter would correspond to merely replicating the single aperture system). This seems to be possible by virtue of the hierarchical nature of multistage networks such as the Banyan; apparently a Banyan twice the size of the available aperture can be emulated by four Banyans (each the size of the available aperture) arranged in the form of two stages with an array of switches in between.

A study of the capacity of single and multiple aperture lens systems which might be relevant for the above discussion is given in [31].

9.3 Trading globality for speed

In the above exposition we have classified certain operations, such as convolution or superposition with a large kernel, as global operations. It is important to realize that at least in some cases these operations can be broken down (or factored) into a sequence of local operations. The cumulative effect of consecutive local operations can allow the necessary global transfer of information to take place. An example is the use of small generating kernels for performing convolutions with large kernels [28, page 253]. In this technique, a convolution kernel is factorized into a large number of kernels, each of which is of small extent. This in effect allows us to eliminate the need for global connections, but requires a greater number of time steps for completion of the operation. This is similar to the use of a locally connected computer network to simulate a globally connected one by routing the information in several hops. Thus, an operation requiring global connections may either be implemented with optical interconnections, or it may be reformulated and implemented without optical interconnections, the latter case requiring a greater number of steps, and also perhaps significantly greater capacity per channel (since many global communication paths may have to share the local interconnections). The tradeoffs here are similar to those analyzed in [5, 9].

10 Directions for development

Based on the accumulated knowledge and understanding, we now present some directions for future development. Research along these lines should provide a platform for resolving many of the open issues discussed, and may demonstrate the power and usefulness of the optical technology under consideration.

10.1 Optically interconnected parallel computers

The importance of the microprocessor concept is that it provided a platform on which computer technology could develop the way it has. The ingredients of the microprocessor paradigm are the stored program concept, (very) large scale integration technology, and some secondary ideas such as the data-flow concept etc. Similar ingredients must be put together if we wish to come forward with a new platform with comparable impact.

Optoelectronic device planes combined with free-space interconnection technology corresponds to the electronic integration technology that made the microprocessor possible. The multi-stage architecture, for instance, might correspond to the data-flow concept in microprocessors. These will be made the basis of a proposal for a fairly general programmable image processing system in the next subsection. In this

[21] This is not a hard number, as yield considerations are very technology dependent, and the limits imposed by aberrations can to some extent be softened by virtue of the digital regeneration that takes place after every communication event.

subsection, we will combine these ingredients with certain parallel computing models, which replace the stored program concept for microprocessors.

We consider a full size silicon chip tiled into small regions, in each of which a processor resides. There should be at least one, and perhaps only one optical pixel per processor. The processors communicate with each other via these optical connections, provided by some kind of optical interconnection architecture. Various possibilities exist:

1. A dynamic multistage permutation network might be provided which provides arbitrary one-to-one connections between the processors. This would support a message passing model of parallel computation.

2. A fixed interconnection network is provided between the processors. If the pattern of connections is regular, it can be implemented with a multistage network. Especially interesting is to provide a perfect shuffle connection pattern, or the most significant stage of a Banyan, which can be implemented in a single stage.

3. Instead of a single array of processors, it is possible to consider several arrays (or a partition of a single large array) and create a pipeline.

4. Still further alternatives are possible if the memory elements are separated from the processors or there is a common shared memory.

Most simple and attractive among the above alternatives seems to be to provide a fixed perfect shuffle or similar connection pattern among the processors, whose optical implementation is particularly simple. This may seem to be restrictive since each processor has limited choice as to which other processors it can send information to. However, in this type of architecture, algorithms are developed such that the routing of information is interspersed with the processing more uniformly than in the case where a dynamic permutation network is provided. Some regeneration will probably have to take place in the intermediate stages of a multistage permutation network anyway, requiring active intermediate planes. Thus, it makes sense to consider systems where the processing power is distributed throughout the stages, since this will allow more efficient use of the silicon in the many active layers.

The important fact is that considerable effort has already been spent and will be spent on the development of algorithms for such parallel computing models. Combination of this effort with the technology under consideration may result in the realization of powerful parallel computers.

10.2 The two-dimensional data-flow architecture

A microprocessor is not designed by specifying its logic function and then using standard procedures to design and layout the logic. Several levels of abstraction are involved within the microprocessor itself. A major feature of most modern microprocessors is the data-flow architecture. This not only enables the desired functionality to be designed in a systematic way, but also leads to an area efficient implementation, well matched to the limitations of planar integration.

Three-dimensional integration technologies may inspire similar architectures. For instance, let us consider the three-dimensional integration technology discussed on page 11. One way of using this technology would be to simply lay out on it an electronic circuit designed in any conventional manner. A more efficient approach might be to use this technology to house a two-dimensional data-flow architecture with the data flowing in the third dimension perpendicular to the active device planes.

A two-dimensional data-flow architecture employing optical interconnections would provide vastly greater flexibility by allowing global connections. Depending on the application (logic, image processing, matrix algebra, etc.), several functions might be selected by appropriately setting control lines, much as we can select between functions such as add, shift, etc. in a microprocessor. Furthermore, several operations may be performed in pipeline fashion as the input propagates through several stages.

The potential for parallelism and pipelining must be exploited by mapping the desired operations and data properly on the architecture.[22] As an example, let us consider an object recognition system. Various consecutive image processing operations must be performed on the raw image. These might be preprocessing, edge detection, edge thinning, edge thresholding, segmentation, feature extraction, and classification [32]. We might assign each operation to one or more device planes, depending on the amount of silicon needed for each as dictated by their relative complexity. By allocating more silicon for those operations that take more time, we can balance the workload through the pipeline. The preprocessing and edge operations require local communication only, so that optical interconnections are not necessary between the initial stages. The latter set of operations (feature extraction, classification) may require global connections which can be provided optically. Since it is beneficial to use optics between some stages, we might use it to provide the local connections between the initial stages as well, depending on whether this simplifies construction. Notice that in this example the problem of parallelizing and pipelining the task is trivial due to the nature of the operations and the data. The problem of mapping the operations and data onto the physical architecture might not be so straightforward in other applications.

One of the most important aspects of the suggested approach is its programmability. The several stages would each be designed so as to provide a certain set of operations with several free parameters, corresponding to the instruction set of a specialized high level programming language, which can be used to realize a relatively general class of image processing algorithms.[23]

By designing the complete system end to end in the form of a single integral architecture, we have avoided the problem of interfacing. The input may be natural light which comes in two dimensional optical form, and the output is of greatly reduced information content anyway. If this image processing subsystem were to be part of a larger computer, the issue of reading the image into the subsystem would have to be explicitly addressed. In the event that the images to be processed are stored on optical disk, it may be possible to recover the encoded data and restore it in the form of an optical image without ever creating an interface bottleneck.

Pipelining may be of no utility if such a system is to be used for real time video processing at conventional frame rates. In this case, the high speed of the devices may offer another advantage; they would enable a large number of passes through the system before the next frame. This would allow the implementation of iterative algorithms, and also give plenty of time for the next frame to be set up on the input array, even if it is being serially read in from an electronic memory. In this case the setup time would not be considered a bottleneck because it would be less than the inherently large time needed for the iterative computation (which would presumably take much more time if implemented electronically).

It is interesting to contrast the approach of this section with that of [32] where the authors consider mapping a similar problem on a message passing hypercube multiprocessor.

11 Conclusions

It is clear that large arrays of very fast and low energy optical devices integrated with established electronic technology and interconnected with free-space optics has very large computational power in the raw sense, but realizing this potential may not be so easy. The difficulty stems from the fact that a whole system of paradigms and levels of abstraction has been constructed around the capabilities and limitations of purely electronic systems, and the dominance of this system of abstractions resists the introduction of a new technology with completely different capabilities and limitations. There does not seem to be much point in trying to build an optical microprocessor, and the user level improvements obtained by replacing the longer wires in conventional systems may be limited. On the other hand, starting with an array of smart pixels, we are too many levels of abstraction away from being able to write a program that plays chess.

[22] Pipelining would be beneficial if the frame rate is high; for instance, if we are processing videos off-line at a faster than real time rate. This issue will be brought up again in a later paragraph.

[23] Of course, there is no reason why this general approach could not be used for other applications, such as a matrix processor, or even a three-dimensional general purpose microprocessor.

Since the construction of a totally new system of paradigms and platforms is an exceedingly difficult task, it is necessary to find ways in which a manageable degree of modification of the existing system would allow net benefits at the user level. The burden of doing this lies with those who want to promote their new technology, but it may also be possible to identify already existing computational paradigms and concepts which were until now only academic exercises but can now be implemented with optoelectronic technology.

One of the main features of the mentioned system of paradigms and levels of abstraction is that its various components are more or less balanced in ability, in the sense that no part of the system is a bottleneck. (This is partly because more effort and resources are put into parts of the system that tend to create a bottleneck, and little effort and resources are put into those parts that are already too good compared to the rest of the system.) However, this state of affairs is dynamic; as applications change, new technologies evolve, and new ideas are introduced, it occurs that one part of the system appears as a bottleneck. Suddenly a flurry of activity begins to improve that part of the system, since any improvements in that part will automatically improve the overall machine.

A very important example that has been increasingly recognized in the past ten years is the interconnection bottleneck. Increasing use of memory, the ambition of processing large amounts of information such as with images and video, the advent of parallel computing, and purely geometrical and physical reasons are some of the factors that have contributed to the increasing importance of interconnections. The most widespread approach has been to replace the longer electrical interconnections with optical ones without otherwise modifying the logical architecture. Examples are optical backplanes, fixed free-space interconnections between circuit boards, etc. In this spirit, SEED on silicon technology can be used to help wire up electronic circuits designed in the conventional way, by providing a large number of pinouts and high performance long distance connections. Although this approach certainly has a certain promise, it is not the one that we believe will bring the greatest rewards.

A more radical approach is to replace an electronic subsystem with an optical one whose internal structure may be completely different from the electronic one it replaces. This is easier said than done, since the overall system that has been optimized with the low performance electronic subsystem in mind may not be able to reflect the superior performance of the optical subsystem to the user level. User level improvements would be observed only if that particular subsystem was already significantly bottlenecking the performance of the overall system, or if successful modification of the overall architecture can be made such that the optical subsystem is smoothly grafted to the overall system.

Special purpose applications in which only a few levels of abstraction are involved are excellent candidates for introducing optical technology in the short term since the architectural and systems issues that must be tackled are less severe than those associated with general purpose systems. The difficulty here is that most such applications do not require high performance, so that already existing technologies seem to suffice. “Smart image sensing” and image processing are two related special purpose applications which seem particularly promising. Optical switching networks seems to be another. There are certain characteristics that make these applications strong candidates: i) they involve large volumes of data (both spatially and temporally) and require global flows of information so that they strain the limits of existing systems; ii) the format of the data and the logical organization of the processing task map naturally into optical architectures that we know can be efficiently implemented.

We already know more or less what SEED on silicon smart pixel arrays will look like, but there are still many open alternatives for free-space optical interconnections. Strong reasons exist for working with reflective devices arrayed on a planar surface, as well as for limiting oneself to regular optical interconnection patterns among the pixels of each smart pixel array. However, there are many topologies with which the arrays themselves can be connected to each other. The (linear) multistage architecture is one promising alternative. Few others have been investigated.

Based on the understanding and knowledge accumulated, we propose two directions for further research. The first is an optically interconnected multiprocessor computer. This might consist of a large number ($\sim$ 1000) of processors on a large wafer or multichip module. The faulty processors may be eliminated by individual testing prior to mounting, or bypassed either by appropriate adjustment of the optical interconnection network (a redundant network should enable routing around faulty processors) or at the software level. Either a multistage permutation pattern, or a simple perfect shuffle pattern might be provided among the processors. The latter seems to allow a better distribution of the electronic circuits on the many device planes in the system. Many other variations are also possible.

Our second proposal is what we have called a two-dimensional data flow architecture for a programmable image processing machine. Most image processing applications involve a series of consecutive operations to be performed on an image. These operations may involve global flows of information throughout the extent of the image. (We have shown that operations involving only local flows of information can probably be more efficiently implemented electronically.) The operations to be performed on various parts of the image can be parallelized to a significant extent. For these reasons, the operations to be performed map nicely onto a multistage architecture, which is known to be attractive from the hardware point of view as well. By the term "programmable" we mean that the system can be used to realize a large class of image processing operations in succession with adjustable parameters.

Acknowledgements

It is a pleasure to acknowledge the benefit of many discussions with David A. B. Miller, Ashok Krishnamoorthy, and Joseph Ford of the Advanced Photonics Research Department, Bell Laboratories, Holmdel, New Jersey, U.S.A. I would especially like to thank David A. B. Miller for his suggestions and comments.

References

[1] J. W. Goodman, F. J. Leonberger, S.-Y. Kung, and R. Athale. "Optical interconnections for VLSI systems." *Proc. IEEE*, 72:850–866, 1984.

[2] D. A. B. Miller. "Optics for low-energy communication inside digital processors: quantum detectors, sources and modulators as efficient impedance converters." *Opt. Lett.*, 14:146–148, 1989.

[3] M. R. Feldman, S. C. Esener, C. C. Guest, and S. H. Lee. "Comparison between optical and electrical interconnects based on power and speed considerations." *Appl. Opt.*, 27:1742–1751, 1988.

[4] M. R. Feldman, C. C. Guest, T. J. Drabik, and S. C. Esener. "Comparison between optical and electrical interconnects for fine grain processor arrays based on interconnect density capabilities." *Appl. Opt.*, 28:3820–3829, 1989.

[5] H. M. Ozaktas. *A Physical Approach to Communication Limits in Computation*, Ph.D. thesis, Stanford University, Stanford, California, June 1991.

[6] H. M. Ozaktas and J. W. Goodman. "The limitations of interconnections in providing communication between an array of points." In *Frontiers of Computing Systems Research, Volume 2*, edited by S. K. Tewksbury, pages 61–130, Plenum Press, New York, 1991.

[7] H. M. Ozaktas and J. W. Goodman. "Elements of a hybrid interconnection theory," *Appl. Opt.*, 33:2968–2987, 1994.

[8] H. M. Ozaktas and J. W. Goodman. "Implications of interconnection theory for optical digital computing," *Appl. Opt.*, 31:5559–5567, 1992.

[9] H. M. Ozaktas and J. W. Goodman. "Comparison of local and global computation and its implications for the role of optical interconnections in future nanoelectronic systems," *Opt. Commun.*, 100:247–258, 1993.

[10] A. V. Krishnamoorthy, P. J. Marchand, F. E. Kiamilev, and S. C. Esener. "Grain-size considerations for optoelectronic multistage interconnection networks." *Appl. Opt.*, 31:5480–5507, 1992.

[11] K. W. Goossen, J. E. Cunningham, and W. Y. Jan. "GaAs 850 modulators solder-bonded to silicon." *IEEE Phot. Tech. Lett.*, 5:776–778, 1993.

[12] L. A. D'Asaro, L. M. F. Chirovsky, E. J. Laskowski, S. S. Pei, T. K. Woodward, A. L. Lentine, R. E. Leibenguth, M. W. Focht, J. M. Freund, G. G. Guth, and L. E. Smith. "Batch Fabrication and Operation of GaAs-Al$_x$Ga$_{1-x}$As Field-Effect Transistor-Self-Electrooptic Effect Device (FET-SEED) Smart Pixel Arrays." *IEEE J. Quan. Elec.*, 29:670–677, 1993.

[13] D. A. B. Miller. "Novel analog self-electrooptic-effect devices." *IEEE J. Quan. Elec.*, 29:678–698, 1993.

[14] H. M. Ozaktas. "Heat removal considerations for optical computing," Internal Report, Department of Electrical Engineering, Bilkent University, Ankara, August 1992.

[15] D. A. B. Miller. "Optical interconnection of devices on chips." United States Patent 4,711,997, Dec. 8, 1987.

[16] H. M. Ozaktas and J. W. Goodman. "Lower bound for the communication volume required for an optically interconnected array of points," *J. Opt. Soc. Amer. A*, 7:2100–2106, 1990.

[17] H. M. Ozaktas, Y. Amitai, and J. W. Goodman. "Comparison of system size for some optical interconnection architectures and the folded multi-facet architecture," *Opt. Commun.*, 82:225–228, 1991.

[18] H. M. Ozaktas, Y. Amitai, and J. W. Goodman. "A three dimensional optical interconnection architecture with minimal growth rate of system size," *Opt. Commun.*, 85:1–4, 1991, Errata in 88:569, 1992.

[19] H. M. Ozaktas and D. Mendlovic. "Multi-stage optical interconnection architectures with least possible growth of system size," *Opt. Lett.*, 18:296–298, 1993.

[20] H. S. Hinton. *Introduction to Photonic Switching Fabrics.* Plenum, New York, 1993.

[21] F. B. McCormick, F. A. P. Tooley, J. M. Sasian, J. L. Brubaker, A. L. Lentine, T. J. Cloonan, R. L. Morrison, S. L. Walker, and R. J. Crisci. "Parallel interconnection of two 64 × 32 symmetric self electro-optic effect device arrays." *Electron. Lett.*, 27:1869–1871, 1991.

[22] F. B. McCormick, T. J. Cloonan, A. L. Lentine, J. M. Sasian, R. L. Morrison, M. G. Beckman, S. L. Walker, M. J. Wojcik, S. J. Hinterlong, R. J. Crisci. R. A. Novotny, and H. S. Hinton. "Five-stage free-space optical switching network with field-effect transistor self-electro-optic-effect-device smart-pixel arrays." *Appl. Opt.*, 33:1601–1618, 1994.

[23] H. M. Ozaktas, H. Oksuzoglu, R. F. W. Pease, and J. W. Goodman. “Effect on scaling of heat removal requirements in three-dimensional systems.” *Int. J. Elec.*, 73:1227–1232, 1992.

[24] M. J. Little and J. Grinberg. “The 3-D computer: an integrated stack of WSI wafers.” In *Wafer-Scale Integration*, chapter 8. Kluwer Academic Publishers, New York, 1988.

[25] H. M. Ozaktas. “Optical preprocessing for real time image coding,” Internal Report, Department of Electrical Engineering, Bilkent University, Bilkent, Ankara, November 1991, revised August 1992.

[26] J. S. Lim. *Two-Dimensional Signal and Image Processing.* Prentice-Hall. Englewood Cliffs, New Jersey, 1990.

Elements of signals, systems, Fourier transforms, z-transforms, discrete Fourier transforms, finite and infinite impulse response filters, and spectral estimation for two dimensional signals. Also has a concise discussion of image processing basics (including a nice discussion of light, the human visual system, and visual phenomena), and an introduction to image enhancement, restoration, and coding. Very readable. Especially the later chapters on image processing are not overwhelming and quite educational.

[27] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

Rather broad treatment with considerable detail, both factually and mathematically. Alternative approaches, extensions, etc. are discussed or pointed out in addition to the basic and mainstream ideas. Sometimes it is difficult to single out the essential ideas which get lost in the clutter. At some points sufficient intuition or motivation is not provided. A good review of mathematical preliminaries is followed by discussion of human image perception, color theory, sampling, quantization, and unitary transforms. Notable is the chapter on stochastic models for image representation. The remaining chapters are on image enhancement, image filtering and restoration, image analysis and computer vision, image reconstruction from projections, and image data compression.

[28] W. K. Pratt. *Digital Image Processing*, second edition. John Wiley and Sons, New York, 1991.

Classic text including a solid exposition of the essentials. Second edition does not sufficiently reflect knowledge accumulated in the field since the first edition.

[29] R. B. Darling and W. T. Dietze. “Implementation of multiplicative lateral inhibition in GaAs sensory neural-network photodetector array.” *IEEE J. Quan. Elec.*, 29:645–654, 1993.

[30] A. Yoshida and J. H. Reif. “Optical computing techniques for image/video compression.” *Proc. IEEE.* 82:948–954, 1994.

[31] H. M. Ozaktas, H. Urey, and A. W. Lohmann. “Scaling of diffractive and refractive lenses for optical computing and interconnections,” *Appl. Opt.*, 33:3782–3789, 1994.

[32] Y. Moon, N. Bagherzadeh, and J. Sklansky. “Macropipelined multicomputer architecture for image analysis.” *J. Opt. Soc. Am. A*, 6:951–962, 1989.