A Constant-Time Parallel Sorting Algorithm and Its Optical Implementation

High-speed electronic sorting networks are difficult to implement with VLSI technology because of the dense and global connectivity required. Optics eliminates this bottleneck by offering global interconnections, massive parallelism, and noninterfering communications. We present a parallel sorting algorithm and its efficient optical implementation using currently available optical hardware. The algorithm sorts n data elements in a few steps, independent of the number of elements to be sorted. Thus, it is a constant-time sorting algorithm, that is, O(1) time.

Ahmed Louri

James A. Hatch, Jr.

Jongwhoa Na

University of Arizona

Sorting is a basic, fundamental operation used for many symbolic, numeric, and artificial intelligence tasks. Some of the applications of sorting include the "togetherness" problem, that is, the problem of bringing together all identical items in a list or file. Another problem is matching, for example, when trying to find all matching entries of two files. If both files are first sorted, one pass through the data finds all matching entries.

A sort can also be used in database and knowledge base processing. Sorting algorithms can serve as a basis for performing many common and highly useful operations such as selection, projection, division, and join in the context of relational databases, and intersection, union, difference, and Cartesian products in the context of sets. Sorting simplifies searching. In addition to its widespread use in information processing, sorting is also important in communications where it serves as the basis for packet routing in networks.

Because of its importance, there has been a great deal of work on developing and analyzing sorting algorithms and architectures. In general, a sort on a string of n data elements can be done with $O(n \log_2 n)$ comparisons using the best serial algorithms.[1] Using a conventional sorting network and taking advantage of parallelism as much as possible permits sorting to be done in $O(\log_2 n^2)$ time steps. Optical architectures for sorting are worth developing for two reasons. Less conventional architectures may permit a sorting operation to be done in far fewer steps, and the fastest conventional sorting networks seem to require fairly dense, globally connected networks that are difficult to implement with conventional electronic technology alone. An example of the latter is a sorting network based on Batcher's bitonic sort,[2] which for a string of 2^k data elements requires $[k(k+1)]/2$ stages. Each stage consists of fast Fourier transform-like butterfly interconnections of various sizes.

Optical technology with its inherent parallelism, high spatial and temporal bandwidths, and noninterfering communications is well suited for implementing the sorting operation. In particular, optics is very attractive for sorting because it can process 2D data arrays in parallel. A 2D optical system has an extremely high number of individual communication channels, all operating in parallel. Besides this parallelism, optical interconnections permit the efficient and high-speed implementation of the global connection patterns required for sorting algorithms. Most importantly, optical systems permit the implementation of sorting with an execution time independent of the number of data elements to be sorted, that is, O(1) time. This is in contrast to electronic sorting systems where the execution time is some function of the number of data elements. In addition, optical sorting systems have

0272-1732/95/$04.00 © 1995 IEEE

minimum time skew and may communicate information at optical media bandwidths. Thus the sorting throughput can be quite large and is limited in practice by the response time of the optical active devices used. Finally, due to the widespread use of optical storage technology in modern high-performance computers and communication systems (asynchronous transfer mode switches), the data to be sorted is already in optical form. This may make optical sorting systems commonplace in future high-performance computers and communication systems.

Several optical sorting algorithms have been proposed in the past.[3] Stirk and Athale[4] proposed a parallel-pipelined sorting algorithm using optical compare-and-exchange modules that has a time complexity of $O(\log_2 n)$ steps. Researchers at IBM also proposed an optical enumeration sort[5] using an optical system based on phase addition and subtraction (interference) to perform analog algebraic operations. However, these coherent systems, as they are referred to, are difficult to construct since the alignment is very critical. Furthermore, these authors failed to suggest a way of physically reordering the data elements after their positions in the sorted output are determined. This restricts the system's usefulness to pointer-based computing systems.

Here, we explore the sorting problem, discussing the optical implementation of a highly parallel sorting algorithm that takes into account the unique properties of optics. Then, we propose an optical system capable of both determining the positions of the sorted data elements *and* physically reordering them in O(1) time steps. This is a significant performance improvement over state-of-the-art electronic sorting systems where the fastest sorting algorithm for n elements takes $O(\log n)$ but requires $O(n^2)$ processors. Our system uses photonics for highly parallel interconnections and optoelectronics, in the form of "smart pixels"[6,7] and deformable mirror devices (DMDs),[8-10] for processing. Thus, it exploits the advantages of both the optical and electrical domains. This is evidence that problems considered solved using conventional approaches need to be reconsidered so that the benefits of optics can be properly used to obtain new, faster solutions to old problems.

A constant-time parallel sorting algorithm

Since optics can process 2D arrays of data in one step, parallel algorithms of the divide-and-conquer classification[11] that are infeasible on electronic computers due to the large number of processors they require, become relevant. On occasion these algorithms are also the most natural way of describing a solution to the problem at hand. For example, to sort an array of n numbers and thus compute its rank (position in the sorted output), we compare every number to every other number. In the following, we describe a sorting algorithm that implements exactly this strategy in constant time, O(1), independent of the number of words being sorted. Later, we describe an optical implementation of each of the steps.

Before formalizing the algorithm, let us first discuss the conventions we use. We indicate row vectors by lowercase underlined letters, such as $\underline{x}$; an uppercase letter indicates a matrix. A subscript indicates the index of the vector or matrix. Thus, x_j indicates the jth element of the row vector $\underline{x}$, and $A_{i,j}$ indicates the element in the ith row and the jth column of matrix A. On occasion, the notation x_i indicates the ith element of column vector $\underline{x}^T$, where the i illustrates a correspondence to a matrix row. Finally, we refer to 2D data as matrices in the context of algorithms and data arrays in the context of optics. This adheres to both mathematical and optical conventions for representing 2D arrays.

As an example, consider sorting data vector $\underline{a}$ = [7 8 2 8 5]. Recall that sorting an n element vector requires comparing each element of the vector against every other element of the vector, implicitly or explicitly. This problem can also be formulated as broadcasting vector $\underline{a}$, and performing one comparison operation for every resulting pair of elements. As optics allows the processing of 2D data arrays in parallel, we vertically "spread" vector $\underline{a}$ such that we can take advantage of this property. This creates n copies of the original vector $\underline{a}$, and places them in a matrix, say A. In the case of our example,

$$A = \begin{bmatrix} 7 & 8 & 2 & 8 & 5 \\ 7 & 8 & 2 & 8 & 5 \\ 7 & 8 & 2 & 8 & 5 \\ 7 & 8 & 2 & 8 & 5 \\ 7 & 8 & 2 & 8 & 5 \end{bmatrix}$$

Hence, each element of the original vector occupies one column of the new matrix. Now, if we transpose this spread matrix, each element of the original vector occupies one row of the transposed matrix.

$$A^T = \begin{bmatrix} 7 & 7 & 7 & 7 & 7 \\ 8 & 8 & 8 & 8 & 8 \\ 2 & 2 & 2 & 2 & 2 \\ 8 & 8 & 8 & 8 & 8 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

The comparison operation between every pair of elements of $\underline{a}$ is then performed by taking the difference of the two matrices A and A^T, where the subtraction is represented as the addition of a negative quantity.

$$D = A + \left(-A^T\right) = \begin{bmatrix} 0 & 1 & -5 & 1 & -2 \\ -1 & 0 & -6 & 0 & -3 \\ 5 & 6 & 0 & 6 & 3 \\ -1 & 0 & -6 & 0 & -3 \\ 2 & 3 & -3 & 3 & 0 \end{bmatrix}$$

Note that in difference matrix D each column j represents the comparison information for element a_j with every other element of $\underline{a}$. Recall that if we are to sort the elements, the rank of an element should indicate its position in the sorted output. Even though elements a_j of the original vector might not be unique, that is, they could be multiple occurrences of the same value, each element must be assigned a unique rank from 1 to n.

The rest of the sorting problem then involves using the information in the difference matrix to arrive at a unique rank for each element in the original vector. Each negative number in column j of D implies the existence of element $a_i > a_j$. Then, if we are sorting the numbers in an ascending order, by our definition of rank, a negative number in column j of D contributes zero to a_j's rank. By a similar argument, each positive number in column j of D contributes one to the rank of a_j. Finally, all zeros in column j represent nonunique elements, that is, numbers with value equal to a_j. Each column necessarily contains one zero, the result of comparing a_j against itself. If we allow zeros to also contribute one to the rank of a_j, we arrive at the following:

$$R = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Here, $R_{i,j} = 0$ when $a_i > a_j$, meaning that a_i appears after a_j in the sorted output and should not contribute to a_j's rank. Similarly, $R_{i,j} = 1$ when $a_i \leq a_j$, indicating that a_i will appear in the sorted output before a_j and contribute to its rank. Summing each column j, we obtain the nonunique rank for the corresponding element a_j. Thus, in the case of our example, the ranks of the sorted elements are:

7 ranks 3,
8 ranks 5,
2 ranks 1,
8 ranks 5, and
5 ranks 2.

Note that the multiple instances of the number 8 are assigned the same rank. To resolve the nonunique ranks, we break ties by comparing the locations of the two elements being compared. So for every two nonunique numbers in vector $\underline{a}$, if one occurs before the other, we consider it to be the larger of the two. Hence, for every $\{i, j\}$, such that $i < j$ and $a_i = a_j$, we treat $\{a_i, a_j\}$ as though $\{a_i > a_j\}$.

To translate this into an operation on difference matrix D, we focus our attention on the zeros of D. As mentioned, the existence of zeros on the main diagonal is guaranteed since the condition $i = j$ represents the comparison of an element to itself. Thus, they are of no concern to us. However, zeros of D for the condition $i \neq j$ mean that the two different elements of $\underline{a}$ being compared are equal. Hence, we have identified the existence and location of nonunique numbers.

Each pair of nonunique numbers is repeated, the first in $\{i,j\}$ and the second in $\{j,i\}$. Since we need to modify the rank of only one occurrence, we select the earlier occurrence to have the smaller rank. By our revised rule, if $D_{i,j} = 0$ and $i < j$, then $D_{i,j}$ should not contribute to the rank of a_j, and hence should be negative. Note that we need to consider modifying only the upper triangle portion of difference matrix D. To do so, we create new matrix U, where $U_{i,j} = 1$ if $i < j$, and 0 otherwise.

$$U = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Next, we subtract U from D, again, by the addition of a negative quantity.

$$D' = D + (-U) = \begin{bmatrix} 0 & 0 & -6 & 0 & -3 \\ -1 & 0 & -7 & -1 & -4 \\ 5 & 6 & 0 & 5 & 2 \\ -1 & 0 & -6 & 0 & -4 \\ 2 & 3 & -3 & 3 & 0 \end{bmatrix}$$

In this new difference matrix D', the negative elements of D remain negative, and hence contribute nothing to the rank of an element. The positive elements remain positive, or become 0. Thus, if we consider that all zeros contribute one to the rank of an element, the positive elements still contribute the same amount to the rank of an element. The zeros in the upper triangular portion of matrix D become negative in D'. This resolves the ties between nonunique ranks in the manner we described. The new rank matrix is then

$$R' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

where $R_{i,j} = 0$ indicates that a_i appears in the sorted output after a_j, and $R_{i,j} = 1$ indicates the opposite. Summing each column of R', we get the fully resolved ranks of each elements in $\underline{a}$:

7 ranks 3,
8 ranks 5,
2 ranks 1,
8 ranks 4, and
5 ranks 2.

The result from the algorithm is the generation of rank vector $\underline{r}$ = [3 5 1 4 2], which contains the positions of each of the data elements in the sorted output. This algorithm and an accompanying optical system are complete when they are capable of rearranging the input data to the order reported in $\underline{r}$. The problem of physically reordering the data reduces to the task of determining, in parallel, which column of $\underline{r}$ contains the number 1, the number 2, and so on, so that we know which element is first, second, and so on in the sorted output. We accomplish this by comparing each element of $\underline{r}$ to the numbers $(1, \ldots, n)$. Mathematically, this is illustrated as spreading the $\underline{r}$ vector n times and subtracting the vector $(1, \ldots, n)^T$, spread horizontally n times, as in the following:

$$S = \begin{bmatrix} 3 & 5 & 1 & 4 & 2 \\ 3 & 5 & 1 & 4 & 2 \\ 3 & 5 & 1 & 4 & 2 \\ 3 & 5 & 1 & 4 & 2 \\ 3 & 5 & 1 & 4 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 4 & \mathbf{0} & 3 & 1 \\ 1 & 3 & -1 & 2 & \mathbf{0} \\ \mathbf{0} & 2 & -2 & 1 & -1 \\ -1 & 1 & -3 & \mathbf{0} & -2 \\ -2 & \mathbf{0} & -4 & -1 & -3 \end{bmatrix} \quad \textbf{(1)}$$

In the resulting matrix of Equation 1, $S_{i,j} = 0$, highlighted by boldfacing, indicates in which column j, of $\underline{r}$, the value i exists. It tells us that a_i is to be relocated to row i of the sorted output, where we assume the sorted output is a column vector. Note that $S_{i,j} = 0$ for only one element per row of S. Say that we use S to select/discard elements from a copy of matrix A, such that elements $S_{i,j} = 0$ select elements $A_{i,j}$ and elements $S_{i,j} \neq 0$ discard elements $A_{i,j}$. Then, we have effectively rearranged the inputs to their positions in the sorted output. The following, where the selected elements are boldfaced, illustrates this operation:

$$A = \begin{bmatrix} 7 & 8 & \mathbf{2} & 8 & 5 \\ 7 & 8 & 2 & 8 & \mathbf{5} \\ \mathbf{7} & 8 & 2 & 8 & 5 \\ 7 & 8 & 2 & \mathbf{8} & 5 \\ 7 & \mathbf{8} & 2 & 8 & 5 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 \\ 5 \\ 7 \\ 8 \\ 8 \end{bmatrix} \quad \textbf{(2)}$$

Thus, the problem of reordering the data reduces to selecting the appropriate element from each row of A, since each row of A has a copy of each data element, and discarding the rest. We summarize the discrete steps for constant time sorting as follows:

Step 1, process the input:

- Generate matrix A by vertically spreading $\underline{a}$ n times.
- Generate matrix A^T by horizontally spreading $\underline{a}^T$ n times.

Step 2, compare every element of $\underline{a}$ with every element of $\underline{a}^T$ by computing the difference matrix $D = A + (-A^T)$.

Step 3, generate the U matrix, where $U_{i,j} = 1$ iff $i < j$.

Step 4, resolve nonunique ranks by computing matrix D' $= D + (-U)$.

Step 5, generate R' by thresholding D', where $R_{i,j} = 1$ iff $D'_{i,j} \geq 0$.

Step 6, form rank vector $\underline{r}$ by summing each column of the matrix R'.

Step 7, reorder the sorted data:

- Compare every element of $\underline{r}$ to every element of $(1, \ldots, n)^T$ by expanding both by n and subtracting the latter from the former to form matrix S.
- Use S to select/discard A where $S_{i,j} = 0$ indicates that data element $A_{i,j}$ should be transferred to row i in the sorted output.

Optical implementation

Consider an optical system that implements these steps and uses the rank vector to physically reorder the input data in constant time. We use currently available components to demonstrate the feasibility of such a system. The system contains a mixture of optics and electronics. We use photonics for highly parallel, noninterfering interconnections, while we use electronics, integrated into arrays of "smart pixels," for processing. The literature discusses the development status of these devices.[12,13]

Smart pixels are optoelectronic devices that combine optical detectors, simple processing electronics, and optical modulators or sources on a single substrate for high-speed processing of 2D data arrays.[6,14] These devices take advantage of the strengths of both the optical and electrical domains. Photons do not readily interact with each other in the optical domain, making it an ideal communications environment. However, this also makes photonics difficult for switching and control. Electrons, on the other hand, interact very easily, which makes them convenient for controlling each other. Smart pixels capitalize on the benefits of electronics for switching and control, and photonics for highly parallel communication. The integration of optical sources provides signal regeneration to the following stage. Furthermore, each of their components can be readily fabricated with conventional silicon and gallium arsenide processing techniques. Some examples of these are SEED-based,[13] Si/PLZT-based,[15] and VSTEP-based[16] smart pixels.

A promising class of lasers that may be used as optical sources both within smart pixels and as system input is vertical-cavity surface-emitting microlasers.[17] They are capable of being integrated in densities of over one million microlasers with dimensions of a few μm on a single chip. Their development is an important milestone in the realization of high-density optically interconnected systems.

Optical algebraic operations. To implement steps 2, 4,

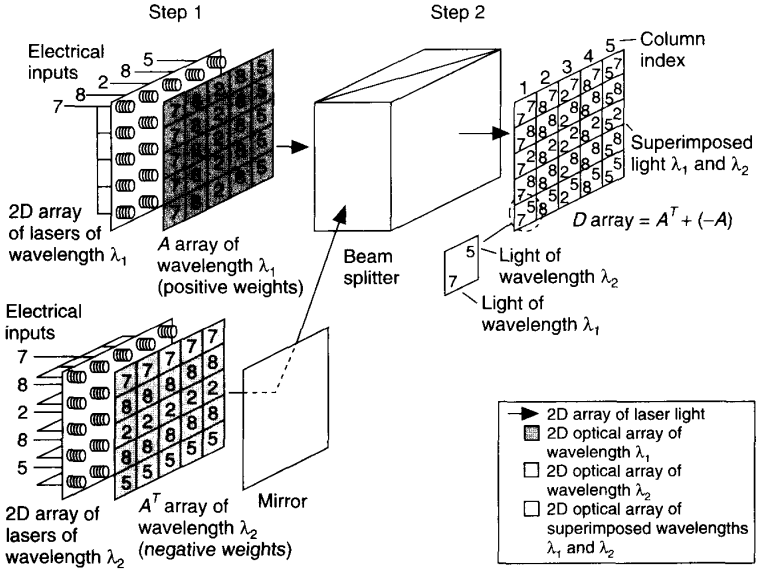

Figure 1. The optical hardware for implementing steps 1 and 2 of the algorithm. Using $\underline{a}$ and $\underline{a}^T$ to modulate 2D laser arrays, with wavelengths λ_1 and λ_2, forms A and A^T. These are "summed" to form the difference matrix D.

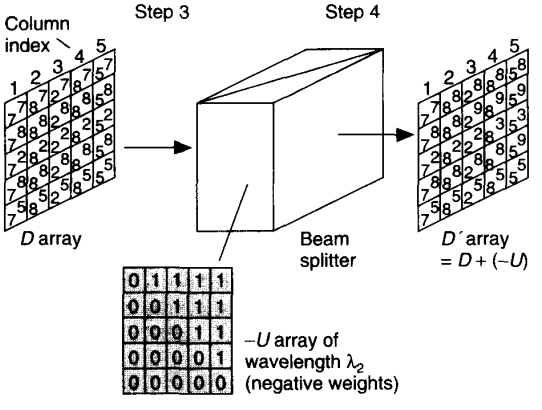

Figure 2. Illustration of the optical implementation of step 3 and part of step 4 of our algorithm. To resolve nonunique ranks, we "add" U to D to form D', where the actual subtraction takes place in the next stage, shown in Figure 3.

and 7 of the algorithm, we must have a way of subtracting two numbers. Since the intensity of light cannot be less than zero, we use a dual-wavelength scheme for representing both positive and negative results.[18] For ease of explanation, we assume that we are only sorting positive values. Negative values will then occur only as the subtraction result of two positive numbers. The system can be modified slightly to accommodate negative data elements, but it is more cumbersome to explain, and we avoided it here.

The positive light intensity level of one wavelength, λ_1, represents positive data numbers, while an equal positive light intensity level of a second wavelength, λ_2, represents their negated values. Thus, a number is considered negative merely by the fact that it is encoded on λ_2. Subtraction is performed by first superimposing (summing), via a beam splitter, the two modulated wavelengths. Since photons do not interact in free space, the actual subtraction of the absolute values written to λ_1 and λ_2 is performed electronically.

The light of wavelength λ_1 impinges upon a photodetector preceded by a λ_1 color filter, while the light of λ_2 impinges upon a photodetector preceded by a λ_2 color filter. The color filters eliminate any cross talk from the opposing wavelength. In general, the two wavelengths should be chosen so that their separation is larger than the pass bandwidth of the filters. Positive photodetector outputs $V(\lambda_1)$ and $V(\lambda_2)$ are then fed into the positive and negative terminals of an operation amplifier, where the subtraction of absolute values $|V(\lambda_1)| - |V(\lambda_2)|$ occurs.

Generating rank vector $\underline{r}$. Figure 1 illustrates an optical implementation for the first two steps of the algorithm.

Implementing step 1. Here, one-dimensional input $\underline{a}$ modulates the columns of a 2D laser array of wavelength λ_1 to form the A array. Meanwhile, $\underline{a}^T$ modulates the rows of a 2D laser array of wavelength λ_2 to form the $-A^T$ array (where the '−' is inherent in the use of λ_2). This differs slightly from our algorithm, which suggested that the two vectors be vertically and horizontally spread n times with optics. We avoided spreading because the beam intensity and "distance" between analog levels are reduced to $1/n$ times the levels of the original beam. The latter reduction makes analog levels more difficult to distinguish at the detectors. Furthermore,

the beam-spreading approach results in more cross talk (assuming non-ideal filters) for this particular system since multiple wavelengths would overlap on the same detector.

Implementing step 2. "Summing" arrays A and $-A^T$ optically forms the difference array D of step 2 by merging their optical data planes with a beam splitter. Recall that at this stage, the beams are merely superimposed and the actual subtraction will be performed later. In Figure 1, each element of the D array contains two numbers that represent the superposition of the two colors. The number in the upper right corner represents the intensity level of light component λ_2, while the number in the lower left corner represents light component λ_1.

Implementing steps 3, 4, 5, and 6. Figure 2 illustrates the implementation of step 3 and part of step 4. Modulating a 2D laser array (not shown) of wavelength λ_2 forms the $-U$ array of step 3. The second beam splitter in Figure 2 sums D and $-U$ in step 4. Figure 3 illustrates, for a single pixel, the subtraction by the smart pixel of the absolute values in D'. Notice the integration of the photodetectors, modulation electronics, and the surface-emitting laser in this close-up view of a single smart pixel. The wavelengths of the D' array are separated by a prism and imaged onto the photodetectors within the smart pixel array. The operation amplifier subtracts the detected value of λ_2, $V(\lambda_2)$ from the detected value of λ_1, $V(\lambda_1)$. The output is then thresholded by a CMOS gate (not shown). The digital output from the thresholding operation of step 5 then modulates the surface-emitting laser for communication to the next stage.

Figure 4 illustrates steps 4, 5, and 6 on a full scale. Here, the labeling conventions of the D' array have been altered since it is being viewed from the back. The output of the electronic subtraction and threshold-

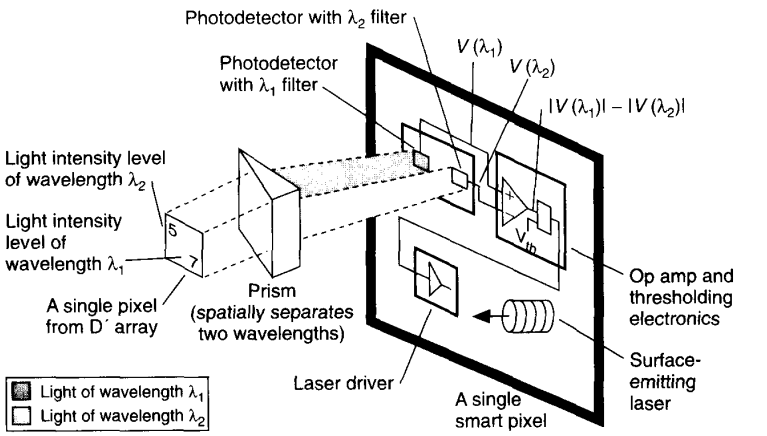

Figure 3. Illustration of the implementation of the actual subtraction from step 4 and all of step 5 for a single pixel. The D' pixel passes through a prism that separates the two wavelengths. These light components impinge upon their corresponding photodetectors in the smart pixel to generate $V(\lambda_1)$ and $V(\lambda_2)$. The operation amplifier performs the subtraction $|V(\lambda_1)| - |V(\lambda_2)|$. The output is thresholded, and the result modulates the laser driver.

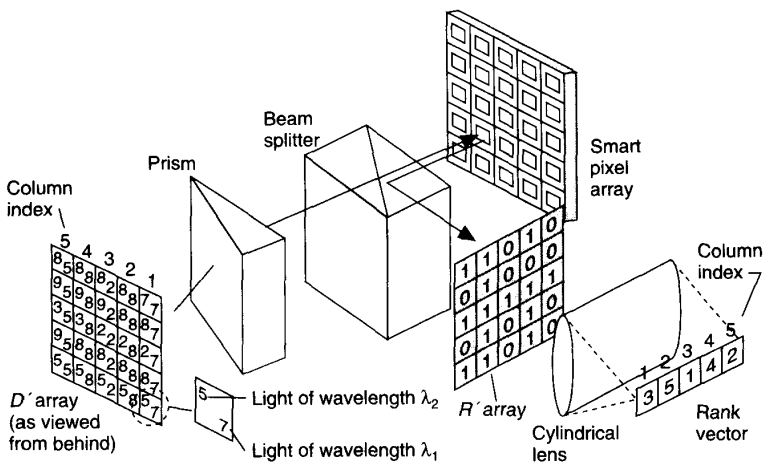

Figure 4. The optical system for implementing the actual subtraction phase of step 4 along with steps 5 and 6 on a full scale. The prism separates the wavelengths so that they may impinge upon their corresponding photodetectors on the smart pixel array. The integrated electronics in the smart pixel array perform the subtraction and thresholding of steps 4 and 5. The surface-emitting laser writes the result to the R' array, which then reflects off the beam splitter and is vertically summed by the cylindrical lens to form rank vector $\underline{r}$ in step 6. Note that the labeling conventions on the D' array in this figure have been changed to mirror the fact that we are now viewing it from behind.

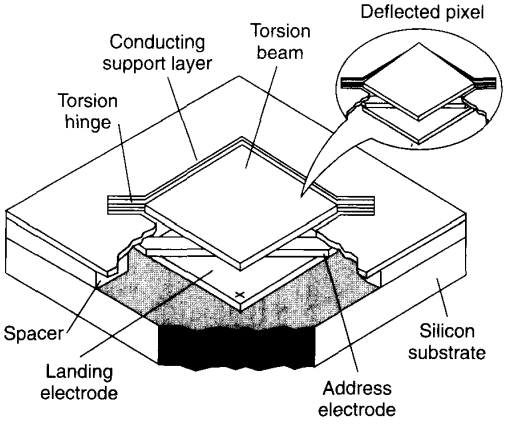

Figure 5. The architecture of a single torsion-beam DMD pixel.[8]

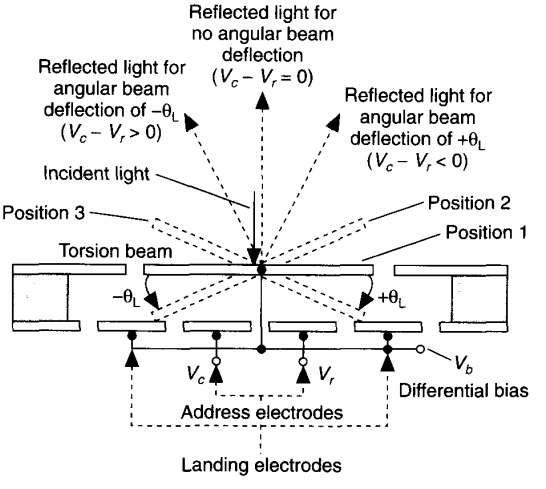

Figure 6. Cross-sectional view of a torsion-beam DMD pixel illustrating the effects of the three mirror positions on an incident light beam.[8]

ing of D' by the smart pixel array modulates the surface-emitting lasers to generate the R' array. Since the lasers are integrated on the same side of the substrate as the photodetectors, the R' array propagates back into the system and reflects off the beam splitter. The cylindrical lens, which focuses a beam only in one dimension, vertically sums the ones in the R' matrix to form rank vector $\underline{r}$, in accordance with step 6. To this end, the rank vector $\underline{r}$ is generated.

Physical reordering of the input data. Next, we need to reorder the input data according to the rank vector. We should note that this additional step may not be required for all applications. For example, if the data elements are sorted in a content-addressable memory, any data element can be recalled directly by its rank, and if this is what is needed, physical reordering is not necessary.

Physical reordering has received little attention previously since prior systems generally proceeded as far as generating the rank vector and recommending its use as a pointer. Our system uses a novel technique involving torsion-beam DMDs for accomplishing this final step. DMDs are micromechanical arrays of mirror elements that are capable of mechanical movement in response to an electrical input. The direction of the incident light is modulated by reflection from rotated pixels.

DMDs are addressed optically or by integrated circuits. In an optically addressed device, integrated photodetectors convert incident beams of light into electrical control signals for the mirror elements. DMDs are also highly efficient light modulation devices since they modulate under reflection. The devices exhibit a reflectivity of approximately 90 percent[10] with a contrast ratio of 2,500:1[9] for the torsion-beam configuration. The device's pixel-switching time is comparable with other optical light modulators. Furthermore, DMDs are fabricated with conventional silicon and GaAs processing techniques and have been reported in sizes of $1{,}920 \times 1{,}080$ elements.[19]

Figure 5 illustrates the architecture of a single mirror. In the torsion-beam configuration, a square pixel is suspended over an air gap by two thin torsion hinges connected to two rigid supports. As voltages are applied to the address electrodes, the pixel deflects about the hinge in response to the electric field, causing a change in the direction of the incident light.

A cross-sectional view of a single mirror is shown in Figure 6. The differential voltage on the address electrodes causes the torsion beam to be attracted toward the more positive electrode. At increasing voltage levels, the angular deflection of the beam increases until it contacts the landing electrode, which prevents any further deflection. This corresponds to angular deflection θ of approximately ±10 degrees. The torsion beam and landing electrodes are held at the same potential to prevent current flow upon contact.

By applying a differential bias to the beam and landing electrodes, we can reduce the voltage required for maximum angular deflection to TTLs (transistor-transistor levels). This differential bias also determines the number of digital deflection states. In the bistable condition, the beam is in a state of equilibrium at an angular deflection of $\theta = \pm 10$ degrees, meaning that it must be deflected to either the right or the left. In the tristable condition, the beam is in a state of equilibrium at angular deflections of $\theta = 0$ degrees and $\theta = \pm 10$

degrees. We use the tristable condition in our design since the subtraction result of step 7a can be negative, zero, or positive.

Figure 6 also illustrates the operation of a DMD pixel for the three addressing conditions, $V_c - V_r > 0$, $V_c - V_r = 0$, and $V_c - V_r < 0$. Zero potential differences across electrodes $V_c - V_r = 0$ do not result in angular deflection (position 1) of the torsion beam. However, a positive potential difference $V_c - V_r > 0$ results in a counterclockwise deflection (position 2) of the torsion beam by 10 degrees. In this position, the incident light is reflected to the left. Similarly, a negative potential difference deflects the torsion beam counterclockwise by 10 degrees (position 3).

Figure 7 shows a top view of the actual system. Incident light enters from the side where it is reflected by the beam splitter and impinges on the DMD pixel. The mirror then reflects it in the direction determined by its position. The reflected beam passes through the beam splitter, and the cylindrical lens focuses it to a point a distance f from the lens, where f is the focal length of the lens. Light reflected by the mirror in position 1 is focused to point F along the optical axis, while light reflected by mirror position 2 is focused to point F′. An opaque screen, with an on-axis opening, blocks the off-axis beam. Thus, applying a nonzero potential difference to the DMD's address electrodes can disable data so that the data beam is deflected and focused off axis, where it is then blocked by the screen.

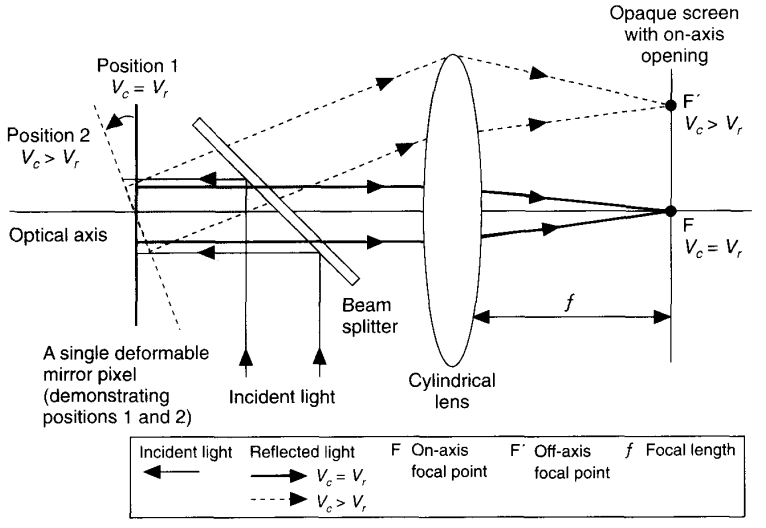

Figure 7. A top view of the optical system for reordering sorted data with a DMD separating the data elements. Incident light enters from the side where it reflects off the beam splitter and impinges on the DMD mirror. It is then reflected depending on the mirror's position. Light reflected from position 1 is focused by the cylindrical lens to point F along the optical axis, while the light reflected from position 2 is focused to F′. Thus, data can be spatially separated by controlling the mirror's position.

The optical system in Figure 8 (next page) builds upon these principles to perform the final step of the algorithm. The first task is the subtraction of the horizontally spread version of $(1, \ldots, n)^T$ from the vertically spread version of $\underline{r}$ as outlined in Equation 1. To facilitate this, we propose that the DMD be fabricated such that the V_c column-addressing electrodes of each cell in a column be connected. Similarly, the V_r row-addressing electrodes of each row should be connected. This allows us to address the entire array in parallel while also performing the vector-spreading operation required in Equation 1.

We load $\underline{r}$ onto the column electrodes and $[1\ 2\ 3\ 4\ 5]^T$ onto the row electrodes. A resistive network integrated onto the device substrate easily implements array $[1\ 2\ 3\ 4\ 5]^T$ since the array's values remain the same for each sort unless the order of the sort (ascending or descending) is changed. The column-addressing lines may be either electrically or optically addressed. In the optically addressed configuration, an array of photodetectors is integrated onto the DMD substrate that connects internally to the column addressing lines. To implement this, the $\underline{r}$ array formed in Figure 4 is imaged directly onto the integrated photodetectors. In the electrically addressed configuration, vector $\underline{r}$ is imaged onto an external photodetector array, which is then connected to the column-addressing lines through DMD chip I/O. The optically addressed configuration is preferred since it is directly scalable with array size, as opposed to the electrically addressed configuration, which is limited by the device pinout limits.

If we consider an element of S as the subtraction of a row electrode voltage from a column electrode voltage, we see that the DMD addressing condition $V_c - V_r$ implements the subtraction in Equation 1. Thus, S represents the potential difference across the addressing electrodes. Since elements $S_{i,j} = 0$ indicate selected elements—one per row—of A, the lenslet array and the cylindrical lens focus the selected elements to a column at the opening of the opaque screen. The mirror array deflects elements of A such that $S_{i,j} \neq 0$ and focuses them to off-axis columns. The opaque screen discards

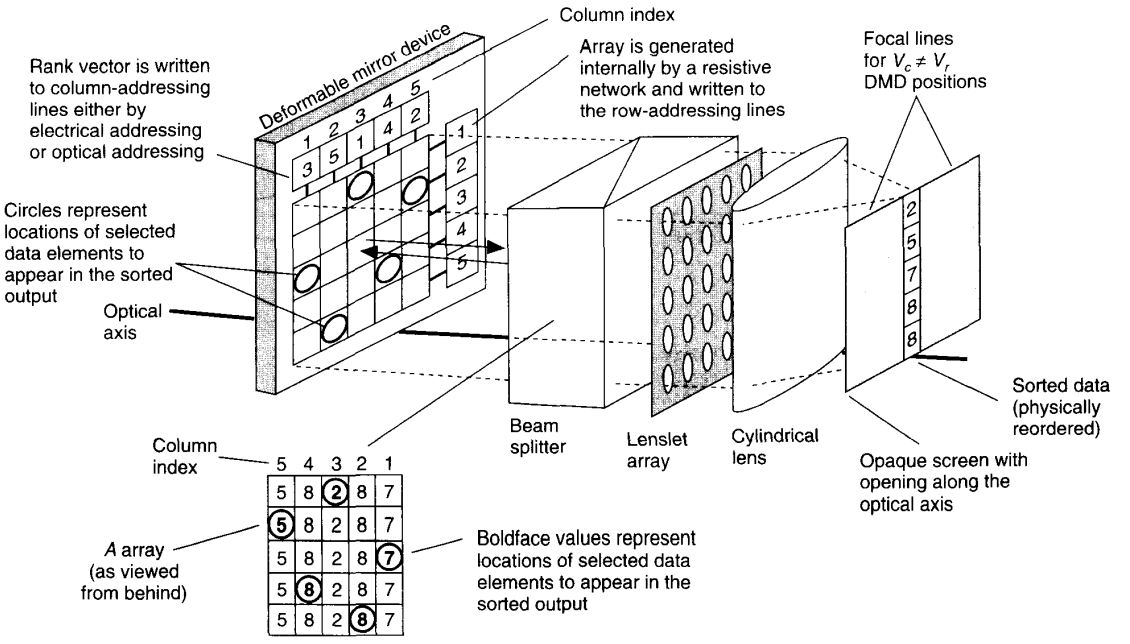

Figure 8. Optical setup for physically reordering the sorted input. The *A* array reflects off the beam splitter and is imaged onto the DMD. Selected elements of *A* reflect off the mirrors and are focused to a column that lies within the opening of the opaque screen by the lenslet array and the cylindrical lens. Elements of *A* to be discarded reflect off the mirror and are focused to off-axis columns where they are permanently blocked by the screen.

these, in accordance with Equation 2 by blocking them from the rest of the system. Thus, we have effectively demonstrated the implementation of Equation 2, the physical reordering of the sorted data.

Discussion

Figure 9 shows the physical layout of the proposed optical sorting system. The two V1 and V2 vertical-cavity, surface-emitting lasers (VCSELs) provide the 2D optical input data, as mentioned earlier. V1 generates the A array with wavelength λ_1. To reduce the amount of hardware required (a VCSEL and beam splitter BS2) while also reducing the power loss due to beam splitting, a single VCSEL generates the $-A^T$ and $-U$ arrays. Since U is an array of constants, it can easily be added to the A^T array as one of the inputs to the laser drivers for the VCSELs in V2. Since V2 is of wavelength λ_2, it provides the array $-(A^T + U)$ to be "added" to A to form the D' array.

Beam splitter BS1 generates two copies of the D' array. One copy is imaged onto SP by polarizing beam splitter PBS1 for the generation of the R' array. CL1 forms the rank vector and images it onto the column-addressing photodetectors of the DMD. To preserve the parity of the array, this 90-degree deflection can be implemented with a pentaprism (not shown) instead of a mirror. PBS2 images the other copy of D' onto the pixels of the DMD for the reordering step. Since this step requires the A array, a color filter that passes only light of wavelength λ_1 removes the $(A^T + U)$ elements from the D' array, leaving behind only the A array. After the DMD selects the appropriate elements of A, lenslet array LA and cylindrical lens CL2 image the sorted data onto the 1D detector array, DET.

The layout in Figure 9 suggests that the system volume should be very small. Using 1-in. aperture optics (lenses, beam splitter and half-wave plates) with a modest 1.5-in. component separation, we see that the entire system can fit in an area of about 20 cm × 20 cm. Although this measurement doesn't include the interface electronics or the heat removal components, the complete optical system should be small enough to be manufactured as an add-on unit for mainframe computers.

Next, we estimate the space bandwidth product (SBWP)

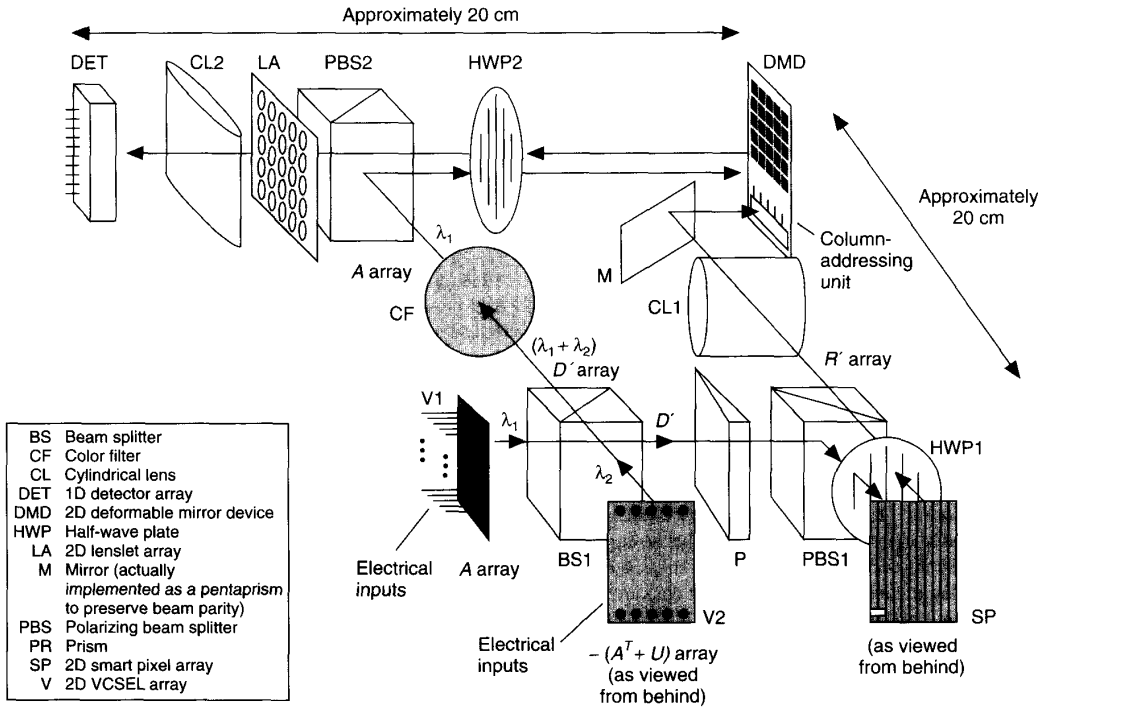

Figure 9. Layout of the proposed optical sorting system.

of the proposed optical sorting system. For simplicity, we assume that the lengths of the pixel of the active components (VCSELs, SP array, and DMD) have the same value, which is a. Saleh and Teich[20] give diffraction-limited spot diameter D_d at the detector plane for the optical setup of Figure 10:

$$D_d = (2.44\,\lambda\,f_{cl}) / a$$

where λ denotes the wavelength of the light, and f_{cl} represents the focal length of the CL2 cylindrical lens. Then, the maximum height of the image at the h_M detector plane becomes

$$h_M = (f_{cl}/f_{la}) \times a$$

where f_{la} represents the focal length of each lens in the lenslet array. Therefore, the maximum SBWP is given by

$$SBWP = (h_M / D_d) = [a^2/(2.44\,\lambda\,f_{la})]$$

As discussed earlier, Boysel[19] reports a 1,920×1,080-element DMD with a pixel size of 17 μm. In a typical optoelectronic smart pixel, the power dissipation limited SPD is

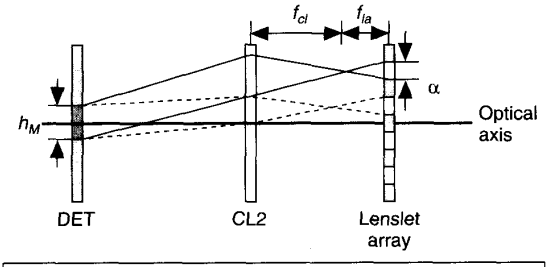

Figure 10. A top view of a part of the optical setup of Figure 9. The figure explains how the lenslet array and the cylindrical lens image a DMD pixel of size *a* onto a detector pixel of size h_M.

known to be about 200 pixels/cm^2.[21,22] Thus, the length of a square pixel of SP array becomes $(\sqrt{200})^{-1}$cm or 707 μm. Comparing these two pixel sizes, we should use the length of array SP as the value of a since it represents the worst-case pixel size in the system. Assuming l is 0.85 μm and f_{la} is 0.1 mm, the SBWP of the proposed optical sorting system becomes 2,410.

In addition to the SBWP, an equally important parameter is the bit capacity of the system, that is, the word size. Since the proposed optical sorting system is analog, the dynamic range of the detectors is mostly responsible for determining this value. Currently, light intensity may be produced, controlled, and detected in about 500 discrete levels.[23-25] This implies that the sorter has the capability of an 8-bit microprocessor. Obviously, this limits the word length of the data that can be sorted in constant time. However, this same restriction is also imposed upon electronic sorting systems. Note that the number of data elements to be sorted at once is not affected by this limitation. A typical sort string might have between 10 and 20 characters. Both optical and electronic sorters can only operate on a subset of this string at any time, a single character, for instance. Thus, the sorting time is inevitably related linearly to the number of characters.

In addition to the word-length restriction, it is also prohibitive to build a sorting system that can sort a sequence length of 10,000 data items without iteration, where a sequence length is the number of data items to be sorted. In general, the m sequence length will be much larger than the n number of items that the sorter can hold at any time. Thus, the data set will have to be divided into m/n smaller data subsets of sequence length n. These will then be sorted separately and compared to each other during multiple passes through the data set. At first glance, it would seem as though one needs to sort each data subset against itself and then against each other subsets. This approach would extend the execution time to $O(m^2/n^2)$ cycles.

ADVANCES IN OPTICAL COMPUTING have opened up new possibilities in several fields related to high-performance computing, high-speed communications, and parallel algorithm design. To design algorithms that execute faster for these applications, we must take into consideration the specific properties of optics, such as massive parallelism and global interconnections. Our algorithm and its optical implementation are an excellent example of what optics can achieve. As these optoelectronic and smart-pixel devices mature, we expect to see such algorithms having a major impact on sorting applications in the future.

Acknowledgments

NSF grant MIP 9113688 supported this research.

References

1. F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, Calif., 1992.
2. K.E. Batcher, "Sorting Networks and Their Applications," *Proc. Spring Joint Computer Conf.*, Vol. 32, AFIPS Press, Reston, Va., 1968, pp. 307-314.
3. Y.B. Karasik, "Fast Sorting Algorithm Based on the Massive Parallelism of Optical Computing," *Proc. Second Symp. High-Performance Computing II*, M. Durand and F.E. Dabaghi, eds., Vol. 6, Elsevier Science, New York, 1991, pp. 83-89.
4. C.W. Stirk and R.A. Athale, "Sorting With Optical Compare-and-Exchange Modules," *Applied Optics*, Vol. 27, No. 9, 1988, pp. 1721-1726.
5. K.S. Huang, L. Liu, and J.K. Peir, "Optical Facility for Parallel Enumeration Sort," *IBM Tech. Disclosure Bulletin*, Vol. 33, No. 3A, IBM, New York, 1990, pp. 16-19.
6. H.S. Hinton, "Architectural Considerations for Photonic Switching Networks," *IEEE J. Selected Areas in Comm.*, Vol. 6, No. 7, 1988, pp. 1209-1226.
7. T.J. Cloonan et al., "A Complexity Analysis of Smart Pixel Switching Nodes for Photonic Extended Generalized Shuffle Switching Networks," *IEEE J. Quantum Electronics*, 1993, pp. 619-633.
8. L.J. Hornbeck, "Deformable-Mirror Spatial Light Modulators," *Proc. SPIE*, Vol. 1150, 1989, pp. 86-102.
9. R.M. Boysel, J.M. Florence, and W.-R. Wu, "Deformable Mirror Light Modulators for Image Processing," *Optical Information Processing Systems and Architectures*, Vol. 1151, SPIE, San Diego, Calif., 1989, pp. 183-194.
10. J.M. Younse, "Mirrors on a Chip," *IEEE Spectrum*, Nov. 1993, pp. 27-31.
11. U. Manber, *Introduction to Algorithms—A Creative Approach*, Addison-Wesley, Reading, Mass., 1989.
12. A. Lentine, "Self-Electro-Optic Effect Devices for Optical Information Processing," *Optical Computing Hardware*, J. Jahns and S.H. Lee, eds., ch. 2, Academic Press, San Diego, Calif., 1994, pp. 64-69.
13. D.A.B. Miller et al., "Field-Effect Transistor Self-Electrooptic Effect Device: Integrated Photodiode, Quantum Well Modulator and Transistor," *IEEE Photonics Technology Letters*, Vol. 1, 1989, pp. 62-64.
14. M.A. Neifeld, "Amorphous Silicon Carbide Multilayer Modulators for Silicon Smart Pixels," *Spatial Light Modulators and Applications, Technical Digest Series*, Vol. 6, Optical Society of America, Palm Springs, Calif., Mar. 1993, pp. 44-47.
15. S.H. Lee et al., "Two-Dimensional Silicon/PLZT Spatial Light Modulators: Design Considerations and Technology," *Optical Engineering*, Vol. 25, No. 2, 1986, pp. 250-260.

16. K. Kasahara, "VSTEP-Based Smart Pixels," *IEEE J. Quantum Electronics*, Vol. 29, No. 2, 1993, pp. 757-768.
17. J.L. Jewellet al., "Surface-Emitting Microlasers for Photonic Switching and Interchip Connections," *Optical Engineering*, Vol. 29, 1990, pp. 210-214.
18. D.R. Collins et al., "Deformable Mirror Device Spatial Light Modulators and Their Applicability to Optical Neural Networks," *Applied Optics*, Vol. 28, No. 22, 1989, pp. 4900-4907.
19. R.M. Boysel, "A 1920-by-1080-Element Deformable Mirror Device for High-Definition Displays," *IEEE Trans. Electron Devices*, Vol. 38, No. 12, 1991, p. 2715.
20. B.E.A. Saleh and M.C. Teich, *Fundamentals of Photonics*, John Wiley & Sons, New York, 1991.
21. S. Yu and S.R. Forrest, "Implementation of Smart Pixels for Optoelectronic Processors and Interconnection Systems I: Optoelectronic Gate Technology," *IEEE J. Lightwave Technology*, Vol. 11, 1993, pp. 1695-1669.
22. S. Yu and S.R. Forrest, "Implementation of Smart Pixels for Optoelectronic Processors and Interconnection Systems II: SEED-Based Technology and Comparison with Optoelectronic Gates," *IEEE J. Lightwave Technology*, Vol. 11, 1993, pp. 1670-1680.
23. E. Pochapsky and D. Casasent, "Linear Acousto-Optic Heterodyning Processors for Complex-Valued Data Processing," *Digital Optical Computing*, Vol. 752, SPIE, 1987, pp. 155-171.
24. D. Psaltis, "Input/Output Devices," *Optical Computing*, Vol. 456, SPIE, 1984, pp. 66-71.
25. D. Psaltis and R.A. Athale, "High Accuracy Computation with Linear Analog Optical Systems: A Critical Study," *Applied Optics*, Vol. 25, No. 18, 1986, pp. 3071-3077.

Ahmed Louri is an associate professor of electrical and computer engineering at the University of Arizona and director of the Optical Computing and Parallel Processing Laboratory. His research interests include computer architecture, parallel processing, optical computing, and optical interconnections. He has published numerous journal and conference articles on these topics.

Louri earned the Diplome D'engenieur (Engineer Degree) in electrical engineering from the University of Science and Technology, Oran, Algeria, and the master's and PhD degrees in computer engineering from the University of Southern California in Los Angeles. He received the 1988 National Science Foundation Research Initial Award and the Best Article of 1991 award from *IEEE Micro*. He is a member of the Institute of Electrical and Electronics Engineers, Association of Computing Machinery, Optical Society of America, and International Society for Optical Engineering.

James A. Hatch was a research assistant in the University of Arizona Optical Computing and Parallel Processing Laboratory when this article was written. He is now employed by Trimble Navigation, Austin, Texas. His research interests encompass the use of optical devices for computing and communications.

Hatch holds a BS degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, New York. He is a member of the IEEE LEOS and Computer societies and the Optical Society of America.

Jongwhoa Na works as a research associate in the Optical Computing and Parallel Processing Laboratory at the University of Arizona. His research interests include digital optical computers, artificial intelligence, and parallel computer architectures.

Na received the BS degree in electronics engineering from SoGang University, Seoul, the MS degree in computer engineering from Wayne State University, Detroit, and the PhD degree in electrical engineering from the University of Arizona. He is a member of the IEEE.

Direct questions concerning this article to Ahmed Louri, Dept. of E&CE, University of Arizona, Tucson, AZ 85721; louri@ece.arizona.edu.