Optical implementation of a single-iteration thresholding algorithm with applications to parallel data-base/knowledge-base processing

Ahmed Louri and James A. Hatch, Jr.

Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona 85721

Received January 22, 1993

Threshold (or relative magnitude) search is traditionally performed iteratively in a bit-serial manner in optical data-base/knowledge-base machines, which results in an execution time proportional to the operand size. We present a single-step threshold search algorithm and its optical implementation. The proposed algorithm performs magnitude comparison in constant time, independent of the operand size, and consequently it greatly increases the performance of optical data-base/knowledge-base processing operations such as searching, selection, retrieving, and sorting.

The information explosion seen in recent years has stimulated the development of computer-based information systems to assist in the creation, storage, modification, classification, and retrieval of mainly textual data. These are known as data base management systems (DBMS's) or knowledge-based systems (KBS's). For example, applications such as real-time command and control and on-line handling of financial data demand the rapid manipulation of mainly symbolic data. The increased amount of data handled by current information systems, coupled with the growing need for more-sophisticated processing functionality, has made current DBMS's and KBS's unable to cope with the needed performance. It has been argued that optics, with its parallelism, speed, and storage capacity, possesses the potential for a permanent solution of data-base/knowledge-base systems.[1–3]

Data-base processing consists mainly of the manipulation of tables, known as relations that resemble files storing data.[4] For example, each row, or tuple, of a relation representing an employee data base stores an entire employee record. Data representing employee name, number, age, etc. are stored in columns, called attributes. Data bases are accessed by manipulation of tuples of one or more relations through a set of operators known as the relational algebra. Composed of union, intersection, difference, selection, projection, and join, the relational operations utilize two types of comparison, equivalence and threshold. In the rest of this Letter the terms equality and inequality, denoted by $=$ and $\neq$, respectively, refer to the equality/inequality of two words; the terms greater than, less than, greater than or equal to, and less than or equal to, denoted by $>$, $<$, $\geq$, and $\leq$, respectively, are referred to as thresholding.[5]

As an illustration of a single important DBMS/KMS operation, we use the selection operation. Selection is important in DBMS's because it retrieves information from tuples through the application of a selection criteria. The criterion is referred to as a theta operation,[4] in which theta is any operator of the set: $>$, $<$, $\geq$, and $\leq$. For example, with the employee data base, a simple application of the selection operation would be the retrieval of the names of all employees who have been employed more than six years. The comparand attribute (the number six) is compared by use of the $>$ operator to the attribute field storing length of employment information in each tuple. The employee name field of successfully matching tuples is retrieved as the operation's output array.

While the equality/inequality comparison in selection algorithms is performed through the bit-parallel comparison of the two words, thresholding is performed in a bit-serial manner, beginning with the most significant bit. This creates different execution times for the selection algorithm. Optical thresholding has received much attention in Refs. 1–4 and 6. However, these methods utilized the bit-serial approach. In this Letter we present an optical implementation method that performs threshold operations in constant time. Since relative magnitude comparisons are a major component of searching and sorting algorithms, their bit-parallel execution will substantially improve DBMS performance. Before we proceed with the explanation of the method, we first explain how words are optically compared. To encode the data optically, we use a combination of polarization and intensity encoding schemes. Additional information on polarization-based logic may be found in Ref. 7. Throughout this discussion, we make use of the previous employee data base.

A comparand attribute is compared with an attribute field of multiple employees simultaneously with the optical system in Fig. 1. This system is basically a vector–matrix multiplier with a polarizer (P1) and a beam splitter (BS1) inserted between the second spatial light modulator (SLM2) and the cylindrical lens (CL3). The added polarizer darkens any vertically polarized light, while the beam splitter duplicates the data plane before the light passes through CL3. The data plane immediately after SLM2 is called the preprocessed data plane, while the data plane duplicated by the beam splitter is called the partially processed data plane. The

0146-9592/93/120992-03$6.00/0 © 1993 Optical Society of America

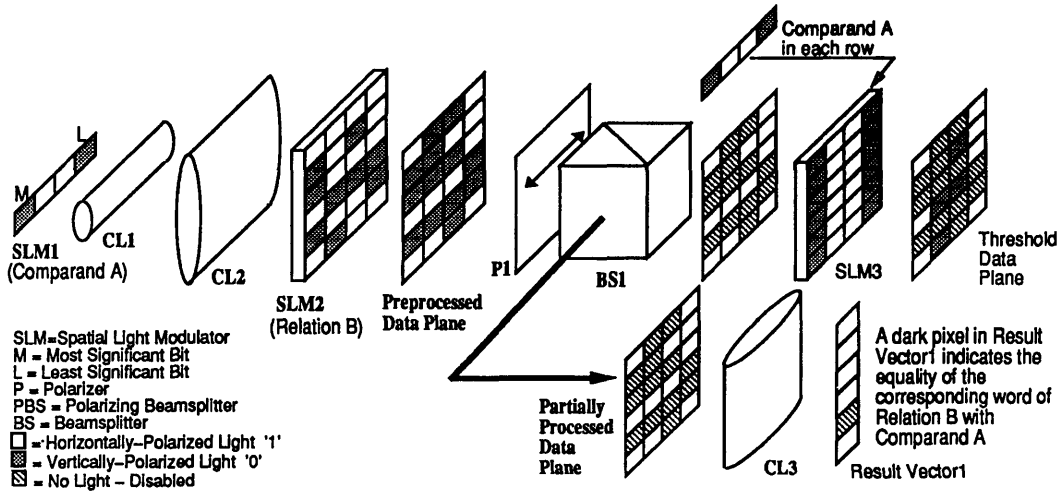

Fig. 1. Optical system for comparing an argument, Comparand A, with the attributes of Relation B.

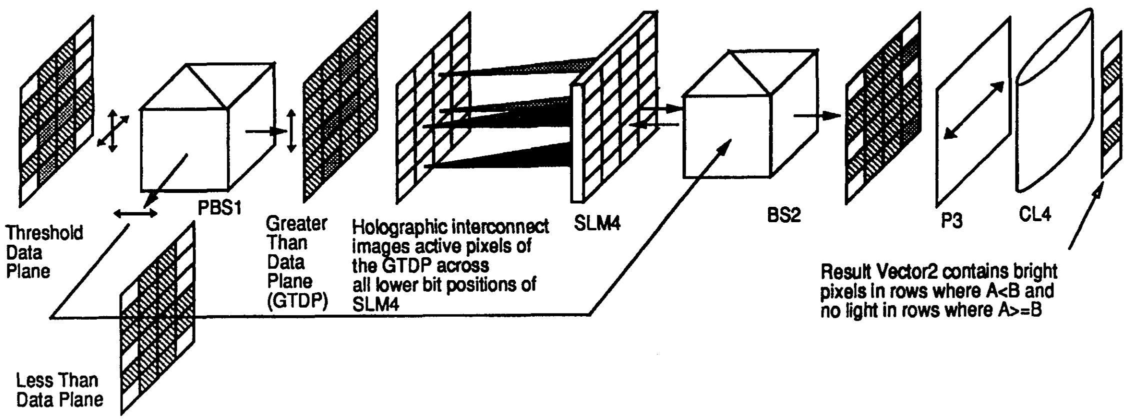

Fig. 2. Optical implementation of the single-iteration threshold unit.

SLM's of our system are pixellated electrically addressed ferroelectric liquid crystals configured to rotate the incident light by 90 deg in bit positions containing a logical value 1, and 0 deg for those containing a logical value 0. A uniform beam of vertically polarized light impinges upon SLM1. The logical values of SLM1 are encoded such that vertically polarized light emanating from the device represents a 0, while horizontally polarized light represents a 1. The combinations of possible polarization rotations experienced by a beam passing through SLM1 and SLM2 are functionally equivalent to a Boolean XOR operation. Since CL1 and CL2 image a bit position of SLM1 onto a bit-slice (column) of SLM2, the preprocessed data plane expresses the result of a bitwise matrix of XOR gates.

Let us consider the implementation of the previous query (retrieval of the names of all employees with more than six years of employment). The comparison argument (the number six) is loaded as Comparand A into SLM1 with the most significant bit to the left. Additionally, the length of employment attribute field is loaded as Relation B in SLM2. Zeros (0's) in the preprocessed data plane indicate the equality of two corresponding bit positions, while ones (1's) denote their inequality. Since photodetectors respond to intensity levels rather than to differences in polarization, detecting the result of an operation requires logical values to be encoded as intensity variations instead of polarization directions. Polarizer P1 accomplishes this by darkening the vertically polarized light illuminating CL3 so that 0's are represented by the absence of light, while the 1's continue to be represented by its presence. P1 also serves the dual role of darkening bit positions in the partially processed data plane that indicate equality, a necessity for single-iteration thresholding.

To this end, the equality/inequality of two words is determined by focusing the preprocessed data plane to a vertical line with CL3 and detecting the presence, or absence, of light in Result Vector1. However, for two words that are not equal, the relative magnitude is not immediately known. For this, two extra steps with two extra SLM's are needed. The first step is a bitwise comparison of the bits of Comparand A and the bits of the PPDP. This comparison is performed by loading A into each row of SLM3 (see Fig. 1). This bitwise comparison determines whether bit A_j (for $j = 1, \ldots, n$, with $j = 1$ representing the least significant bit) is greater than or less than B_{ij} (the bit in the ith row and jth column of Relation B). The output of SLM3 is referred to as the threshold data plane and consists of vertically polarized light in bit positions where $A_j > B_{ij}$, horizontally polarized light in bit positions where $A_j < B_{ij}$, and no light where $A_j = B_{ij}$. The second extra step determines whether the entire word A is greater than or less than an entire word from a row of Relation B.

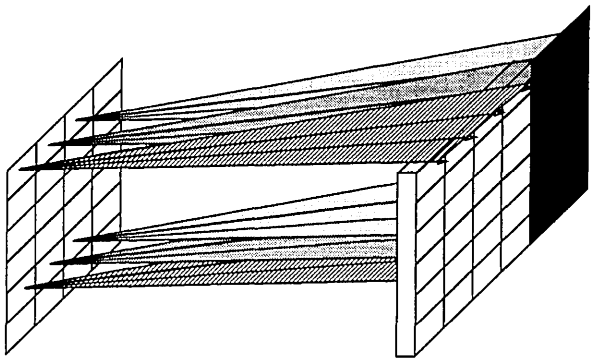

Fig. 3. Space-invariant hologram images each bit position of the incident data plane onto the corresponding lower bit positions of SLM4. Optics between the hologram and SLM4 remove unwanted diffraction orders and collimate the beams illuminating the SLM; these were omitted for clarity. The black wall attached to SLM4 serves as a baffle to block light beyond the SLM boundaries.

This depends on finding the first (deterministic) bit position, beginning with the most significant bit, where an inequality exists. The word containing a one in this position is the larger-valued word. For example, the latter of the binary patterns $A = 10100$ and $B = 11011$ is the larger-valued word because it contains a one in the highest bit position that results in an inequality. Even though the result of the second extra step indicates that $A_j > B_j$ in lower bit positions, these have no effect on the final outcome and must be simultaneously eliminated from the final result. Thus thresholding in constant time requires that the bit positions below the deterministic bit be disabled in parallel.

To envision the disabling process, consider the creation of two data planes, the greater-than data plane (GTDP) and the less-than data plane (LTDP). The GTDP contains a 1 in each bit position where $A_j > B_j$ and 0's elsewhere, while the LTDP contains 1's in bit positions where $A_j < B_j$ and 0's elsewhere. In this example, the GTDP is 00100 while the LTDP is 01011. The 1's in the GTDP then force all the LTDP bits in lower positions to 0's, resulting in the plane 01000 for the current example. If this plane still contains 1's in any bit position(s), then a $A_j < B_j$ result occurred in a higher bit position than the first $A_j > B_j$ and $A < B$. If all the bright LTDP bits are forced to 0, then the GTDP has a 1 in the highest bit position, which results in an inequality. However, there may not have been any bright bits in the LTDP to begin with, i.e., the two words are equal. Thus a fully disabled LTDP row indicates only that $A \geq B$.

The optical system of Fig. 2 accomplishes this data plane generation and bit disabling process. The polarizing beam splitter (PBS1) creates two data planes of separate polarizations: the GTDP containing vertically polarized light in positions where $A_j > B_{ij}$ and darkness elsewhere, and the LTDP containing only horizontally polarized light in positions where $A_j < B_{ij}$. A space-invariant computer-generated holographic interconnect images bits of the GTDP onto their corresponding lower bit positions of SLM4, which is an optically addressed ferroelectric liquid-crystal SLM. Figure 3 illustrates the functionality of the holographic interconnect. Spatial filtering and collimating optics between the holographic interconnect and SLM4 are used to remove unwanted diffraction orders while collimating the diffracted beams. The diffracted beams from the hologram impinge upon the photosensitive side of SLM4 while BS2 images the LTDP onto the reflective side of SLM4. The electric fields established in SLM4 by the diffracted beams rotate the polarization of the LTDP bits in the locations to turn the appropriate 1's to 0's. BS2 then transmits the reflected data plane, and polarizer P3 removes any rotated bits (0's) from further evaluation. CL4 searches for any remaining bright LTDP bits by focusing each row to a point forming Result Vector2. A bright Result Vector2 bit means that $A < B_i$, while a dark bit means that $A \geq B_i$. Since Result Vector1 denotes the equality of the words, if it is inverted and NOR'ed with Result Vector2, a bright bit results in rows of the output where $A > B_i$. Thus the relative magnitude comparison of multiple words is performed in a single pass through the optical system. The method is independent of the number of bits in a word, which makes it desirable for large-scale parallel-processing systems.

In this Letter we proposed a single-iteration thresholding algorithm and its optical implementation. The algorithm provides optical data-base systems with the parallel thresholding of a comparand with an attribute field of each tuple in a relation in constant time. Thus, for an attribute field of length m (m bits wide), the proposed algorithm provides a speed-up factor of m over the conventional iterative algorithms. The proposed algorithm would potentially provide a significant speed improvement for many symbolic algorithms, such as searching and sorting in large data bases, where thresholding operations are extensively used. Since the optical system of Fig. 1 is already a fundamental component of optical DBMS/KMS systems, the additional hardware required for single-iteration thresholding represents a reasonable amount of added complexity and cost for the performance enhancements offered.

This research was supported by National Science Foundation grant MIP 9113688.

References

1. P. B. Berra, K.-H. Brenner, W. T. Cathey, H. J. Caulfield, S. H. Lee, and H. Szu, Appl. Opt. **29**, 195 (1990).
2. P. B. Berra, A. Ghafoor, P. A. Mitkas, S. J. Marcinkowski, and M. Guisani, IEEE Trans. Knowl. Data Eng. **1**, 111 (1989).
3. P. A. Mitkas and P. B. Berra, in *Proceedings of the 25th Hawaii International Conference on System Sciences* (IEEE Computer Society, New York, 1992), p. 338.
4. C. J. Date, *An Introduction to Database Systems,* 4th ed. (Addison-Wesley, Reading, Mass., 1986), p. 245.
5. A. Louri, Appl. Opt. **29**, 3241 (1992).
6. R. A. Athale and M. W. Haney, Opt. Lett. **17**, 1611 (1992).
7. A. W. Lohmann, Appl. Opt. **25**, 1594 (1986).