

Copyright 1998 Society of Photo Instrumentation Engineers.

This paper was published in SPIE Proceedings, Volume 3386 and is made available as an electronic reprint with permission of SPIE. One print or electronic copy may be made for personal use only. Systemic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

Data flow architecture for high-speed optical processors

Kipp A. Bauchert and Steven A. Serati

Boulder Nonlinear Systems, Inc., Boulder CO 80301

1. Abstract

For optical processor applications outside of laboratory experiments, it is desirable to streamline the data flow in order to obtain the highest possible throughput from the system. This paper presents the data flow architectures for two optical processors designed and built by Boulder Nonlinear Systems, as well as the processor designs and some experimental data.

Keywords: spatial light modulators, optical correlators, liquid crystal displays, manufacturing inspection, optical processing, multispectral analysis, security monitoring, machine vision, data flow

2. Introduction

Optical and digital processors are commonly viewed as two completely different approaches to process data. This paper shows that under certain conditions these two types of processors have base commonality. The principals of a data flow processing architecture are applied to an optical processor. A data flow architecture attempts to improve throughput by reducing the control signals to a minimum with processing elements that perform a predefined operation when all of the data is presented. This concept is very similar to the highly parallel nature of an optical processor. The data flow architecture is commonly viewed through the use of activity templates and program graphs. These same techniques are then modified and applied to an optical processor. Two optical processors that were designed and built by Boulder Nonlinear Systems are presented along with some experimental correlation data.

3. Data flow architecture

Data flow architectures were developed in the 1970s as a new scheme for improving the throughput of computers. The basic premise is that there are very few control signals and the processing is performed as soon as all of the necessary operands, or data, are present in the processor element¹. The processor elements can be predefined or programmed on the fly to perform a single task such as addition, multiplication, comparison, etc. Tokens are associated with each piece of data to ensure that a new process is not started until all of the operands are present and the previous result has been passed to the next processing element. These processing elements can then be interconnected to perform complex programs at very high throughputs due to the minimal amount of overhead. A simple program is represented in Figure 1 that calculates the value:

$$z = (x + y) * (x - y)$$

Each table is referred to as an activity template and contains fields for the operation to be carried out, one field for each of the operands, and one field for each of the results. Each result field contains an address to the input field of another activity template along with the result data.

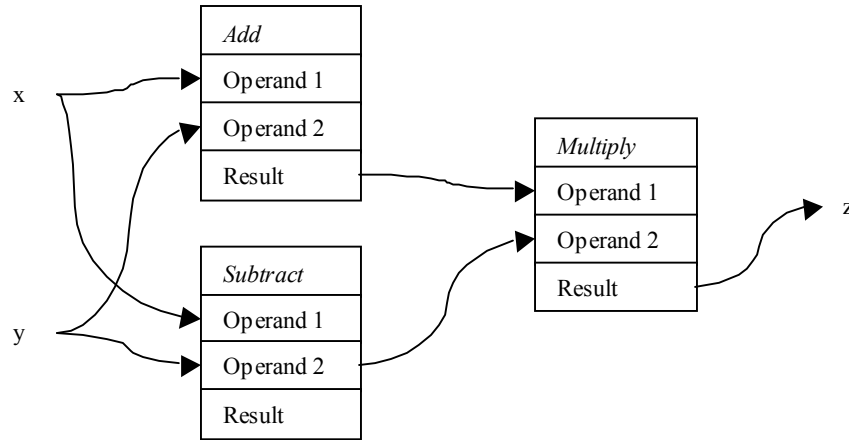


Figure 1 - Simple program representing a data flow architecture implemented with activity templates.

A basic processing element is depicted in Figure 2. The activity store holds all of the necessary activity templates for the data flow program. Each activity template has a unique address that is entered into the instruction queue FIFO unit in the order of desired operation. The fetch unit receives these addresses from the instruction queue, fetches the appropriate activity template from the activity store, and formats the information into an operation packet for the operation unit. The operation unit then performs the defined operation according to the received packet, generates the appropriate result packet, and is then ready to receive the next operation packet. The update unit interprets the result packet and updates the input fields of all of the appropriate activity templates in the activity store. The update unit also tests for the availability of all operands for a given activity template and places the activity template address into the instruction queue as soon as all operands are available. This circular pipeline mechanism allows each unit to be constantly busy as long as the instruction queue is not empty.

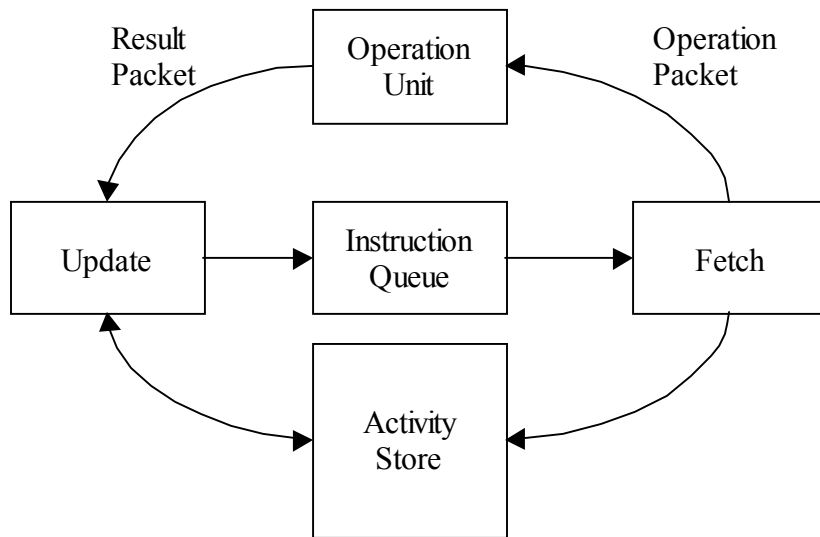


Figure 2 - Basic processing element for a data flow architecture.

4. Optical processor

An optical processor is a highly parallel data dependant processor and can therefore be considered as a special purpose version of a data flow processor. The optical processor described here contains only three basic types of operations, a Fourier

transform, a multiplication, and a modulus squared. The data flow program graph for this optical processor is depicted in Figure 3 and a conceptual drawing is shown in Figure 4.

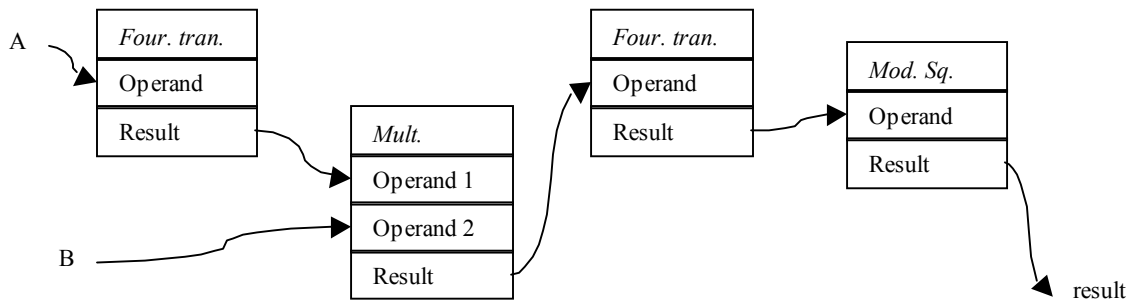


Figure 3 - The data flow program graph for an optical processor.

The Operand A is introduced into Plane A via a programmable Spatial Light Modulator (SLM), and Operand B is introduced into Plane B via another SLM. Diffraction propagation from the SLM in Plane A results in the first Fourier transform operation depicted by the first activity template in Figure 3. The lens L1 scales and locates this Fourier transform plane to a more useful form. The second Fourier transform operation is created in a similar fashion with the SLM at Plane B and the lens L2. The multiplication of the Fourier transform of Plane A with Plane B occurs at Plane B prior to the second Fourier transform. The final modulus squared is a result of the square law intensity detection with a CCD detector. Coherent optical processing is performed by use of a coherent laser as the illumination source for the optical processor.

The optical processor performs the operations defined in the activity templates at the speed of light as soon as the operands are present. Hence for a typical optical processor, two complex two-dimensional Fourier transforms and multiplications are performed in approximately 1 nanosecond, regardless of the number of pixels in the SLMs or detector. This optical processor can be utilized for several different calculations, see Table 1, depending solely on the input data for Operands A and B.

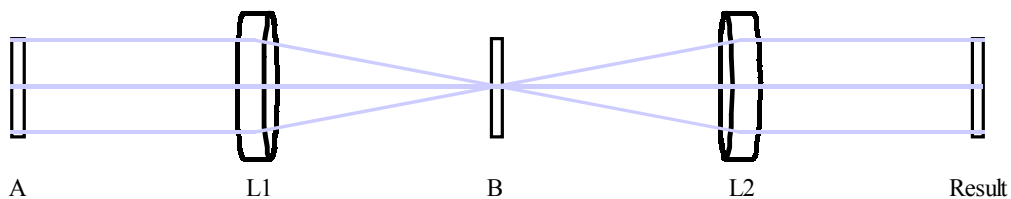


Figure 4 - Conceptual drawing of an optical processor.

Table 1 - Some possible calculations for the optical processor defined in Figure 3.

<u>Operand A</u>	<u>Operand B</u>	<u>Result</u>
$g(x,y)$	$H^*(f_x, f_y)$	Cross-correlation
$g(x,y)$	$H(f_x, f_y)$	Convolution
$g(x,y)$	$H(f_x + \alpha, f_y + \alpha) + H^*(f_x - \alpha, f_y - \alpha)$ (Vander Lugt filter ²)	Convolution and Cross-correlation separated by 2α
$g(x,y)$	$G^*(f_x, f_y)$	Auto-correlation
$g(x,y)$	$A(f_x, f_y)$ – amplitude mask	Spatially filtered version of $g(x,y)$

A data flow optical processing element is depicted in Figure 5. The data store holds all of the necessary input images and filters for the desired task. Each data template consists of an input image and filter combination and has a unique address that is entered into the instruction queue FIFO unit in the order of desired operation. The fetch unit receives these addresses from the instruction queue, fetches the appropriate data template from the data store, and formats the information into a data packet for the optical processor. The optical processor then generates the appropriate result data from the CCD detector at the correlation plane and is then ready to receive the next data packet.

An intelligent optical correlator system will have a more complex update unit than the example in Figure 2. The update unit must search every correlation image for valid correlation peaks. Then the update unit must either report these results or decide what additional data templates should be processed based on the correlation peak data. A brute force approach would simply run through a predefined set of input and filter combinations and report all of the results. The simplest approach to improving the throughput would be to utilize something on the order of a binary search tree for stepping through the data templates. The proper algorithm should be based on the type of data and the application scenario, no one algorithm will give the best results for every application.

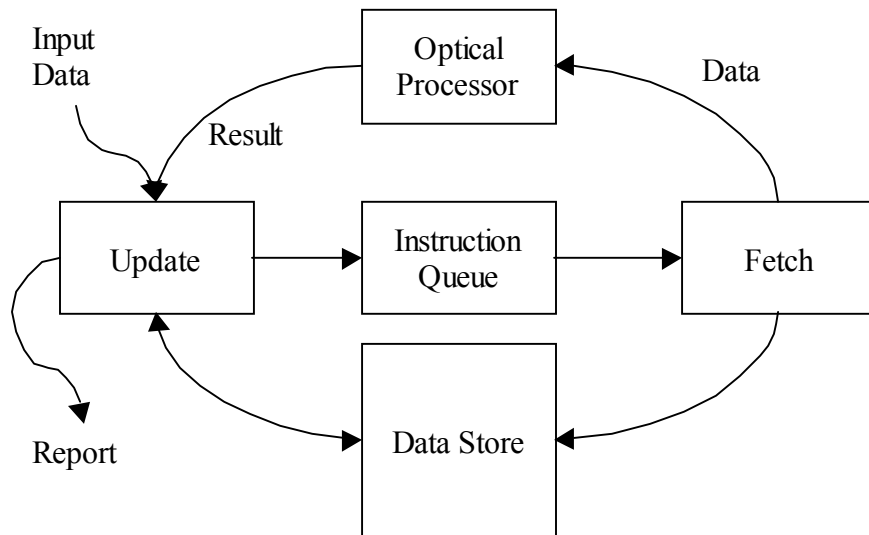


Figure 5 - A data flow optical processing element.

4.1. 128x128 analog optical correlator

The 128x128 analog optical correlator utilizes two Spatial Light Modulators (SLMs) for inputting the data, see Figure 6 and Figure 7. The output of the correlator is a Dalsa 128x128 high-speed CCD camera with a maximum throughput of 830 Hz. The 128x128 analog SLMs can modulate light in amplitude-only, real-axis-only, phase-only, or a amplitude-phase-coupled modes³. The type of modulation is selected via the liquid crystal material and the input polarization state⁴. The analog nature of the device allows for many levels of modulation. Current drive electronics support 4-bit and 8-bit modulation depth while refreshing the SLM data at a rate of almost 9000 Hz and allowing useful frame rates of nearly 1500 Hz.

The theoretical full-frame load time of the VLSI chip is approximately 25 μsec , but has only been tested to 102.4 μsec . This results in a tested continuous full-frame load rate of 9766 Hz, or equivalently 1.3 gigabits/sec. However, this does not include time for the Liquid Crystal (LC) to optically respond to the electric field, or for actual viewing time. For a Chiral Smectic LC (CSLC) device, the typical response time (10% to 90% modulation) will be approximately 50 - 150 μsec , which is mainly a function of electric field strength and temperature. Our current drive electronics support a load time of 113.8 μsec and response and view times as short as one load time. This time coupled with an equivalent inverse image cycle for electrical balancing the LC results in a useful frame time of 682.6 μsec , or a rate of 1464 Hz.

For a Zero-Twist Nematic (ZTN) device, the rise time can be comparable to a CSLC, while the fall time limits the frame rate to approximately 500 Hz. Note that a nematic modulator responds to the amplitude of an AC field, unlike a CLSC device which also responds to the polarity. Therefore, the “true” and “inverse” images necessary for electrical balancing result in a static image for the ZTN devices.

This correlator also utilizes a Dalsa 128x128 camera as the input device for the input SLM, loading new images into the SLM at a frame rate of 732 Hz with every other frame being inverted to maintain a balanced electrical field resulting in a useful frame rate of 366 Hz. The filter SLM is driven from memory at 732 Hz, twice the rate of the input SLM, resulting in each filter being correlated with both the true and inverse input images. The output Dalsa camera is also driven at 732 Hz with each frame being captured by a frame grabber card. The captured images are then transferred to host memory for postprocessing with a peak detection algorithm.

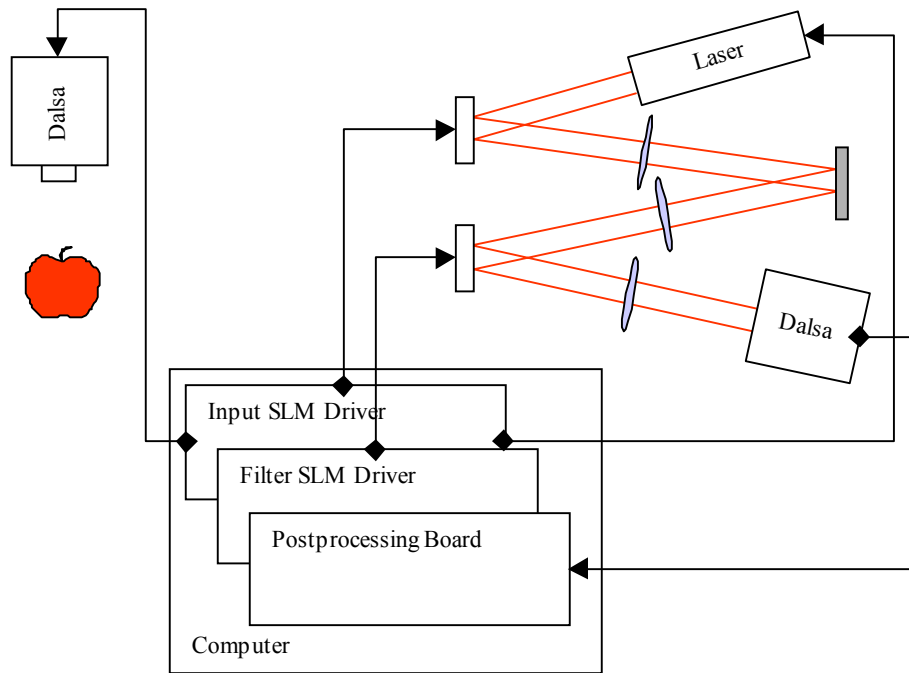


Figure 6 - Block diagram of 128x128 analog correlator.

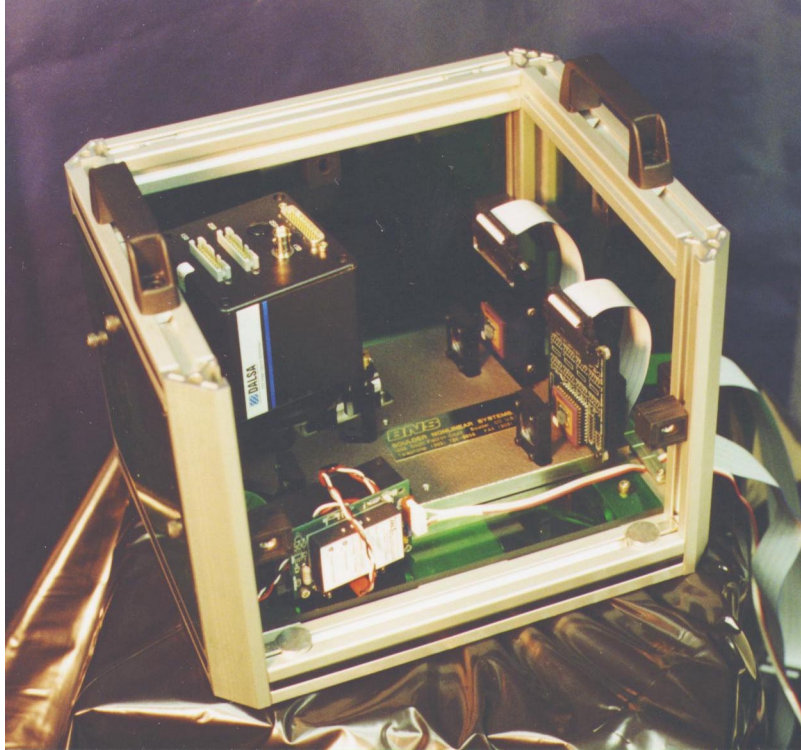


Figure 7 - Photograph of 128x128 analog correlator.

4.2. 256x256 binary optical correlator

The 256x256 binary optical correlator utilizes two Spatial Light Modulators (SLMs) for inputting the data, see Figure 8 and Figure 9. The output is a Dalsa 256x256 high-speed CCD camera with a maximum sustained throughput of 220 Hz. The 256x256 binary SLMs can modulate light in binary amplitude or binary phase⁵. The type of modulation is selected by rotating an output analyzer. Current drive electronics refresh the SLM at a sustained rate of over 18 KHz and support a useful frame rate of over 2000 Hz.

The theoretical full-frame load time of the 256x256 VLSI chip is approximately 25 μ sec, but has only been tested to 51.2 μ sec. This results in a tested continuous full-frame load rate of 19531 Hz, or equivalently 1.3 gigabits/sec. However, this does not include time for the LC to optically respond to the electric field, or for actual viewing time. Our current drive electronics support a load time of 55.4 μ sec and response and view times as short as one load time, and a total true/inverse cycle time of 8 load cycles. This time results in a useful frame time of 442.8 μ sec, or a rate of 2258 Hz.

Each SLM is driven from a memory bank of 512 images. The Dalsa 256x256 camera feeds into a frame grabber for capturing the correlation images. The captured images are then transferred to host memory for postprocessing with a peak detection algorithm. Some sample input and actual correlation images can be seen in Figure 10 and Figure 11.

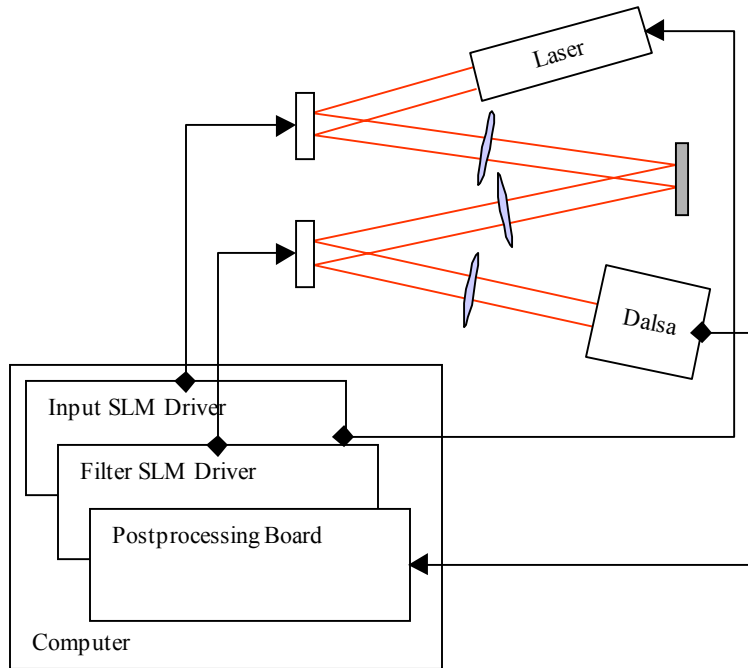


Figure 8 - Block diagram of 256x256 binary correlator.

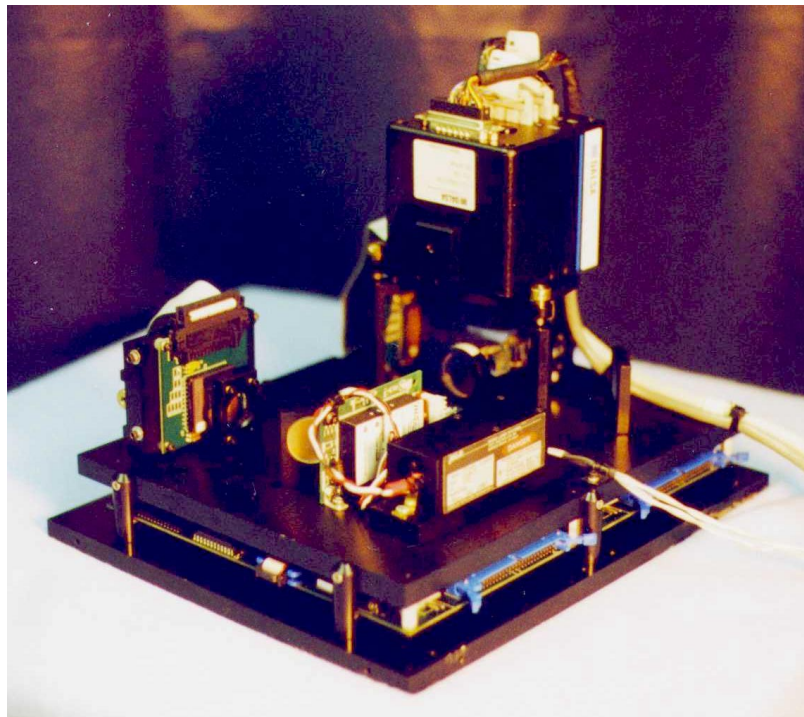


Figure 9 - Photograph of 256x256 binary correlator.

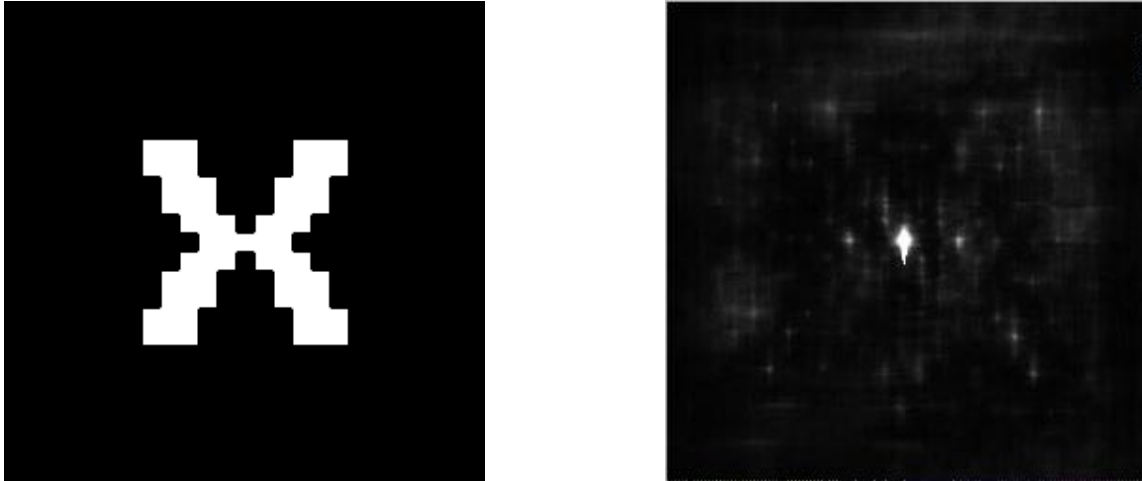


Figure 10 - BigX input and correlation from 256x256 binary optical correlator.

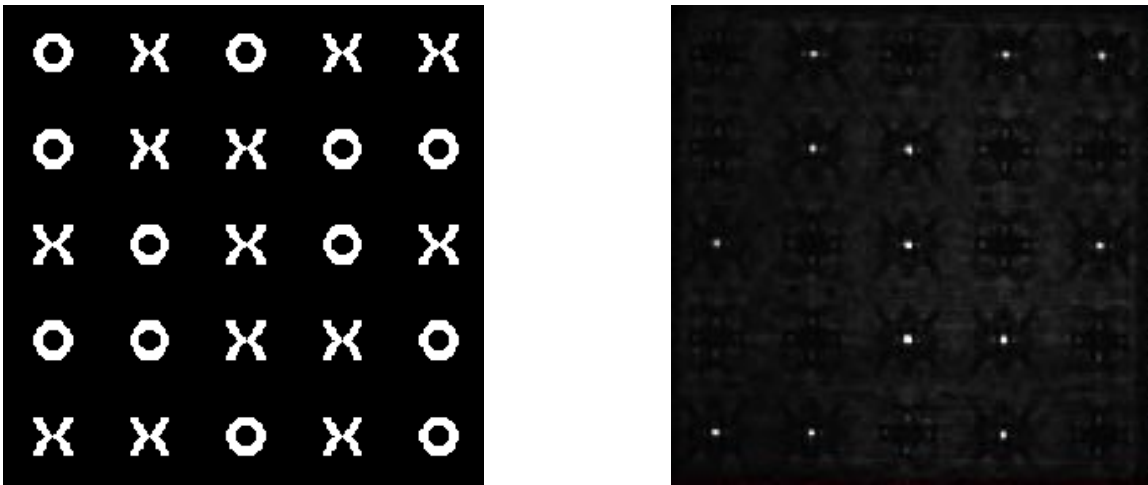


Figure 11 - SmallXO input and correlation with a SmallX filter in the 256x256 binary optical correlator.

5. Summary

Similarities between data flow processors and optical processors have been drawn. Both achieve high throughputs by processing data in a highly parallel fashion as soon as the data is presented. Some background was given on the description of a data flow processor with through the use of activity templates and program graphs. These same principals were applied to describe a generic optical processor. The designs for both the 128x128 analog and the 256x256 binary optical processors have been described in detail. Experimental correlation results were presented for the 256x256 binary optical processor.

6. Acknowledgments

The authors would like to thank the United States Department of Agriculture (contract #96-33610-3088) and Pacific Northwest National Labs (contract #323592-A-A6) for funding the development of the two optical processors.

7. References

-
- ¹ J. B. Dennis, "Data Flow Supercomputers", *Computer*, Volume 13, Number 11, pp. 48-56, IEEE, November 1980.
 - ² J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, San Francisco, 1968.
 - ³ S. A. Serati, G. D. Sharp, & R. A. Serati, "128 x 128 analog liquid crystal spatial light modulator", *Optical Pattern Recognition VI*, Volume 2490, pp. 378-387, SPIE, Bellingham, April 1995.
 - ⁴ K. A. Bauchert, S. A. Serati, G. D. Sharp, & D. J. McKnight, "Complex phase/amplitude spatial light modulator advances and use in a multispectral optical correlator", *Optical Pattern Recognition VIII*, Volume 3073, pp. 170-177, SPIE, Bellingham, April 1997.
 - ⁵ D. J. McKnight, K. M. Johnson, & R. A. Serati, "256 x 256 liquid-crystal-on-silicon spatial light modulator", *Applied Optics*, Volume 33, Number 14, pp. 2775-2784, May 1994.