232-57

Accurate numerical computation by optical convolution

Demetri Psaltis, David Casasent, Deborah Neft and Mark Carlotto

Carnegie-Mellon University
Department of Electrical Engineering
Pittsburgh, Pennsylvania 15213

Abstract

Methods for improving the accuracy of optical processors are considered. The improvement in accuracy is achieved by increasing the space-bandwidth product requirement of the system. Since convolution is the operation most easily and efficiently implemented by optical systems, we concentrate on systems that achieve convolution with increased accuracy.

I. Introduction

One of the major shortcoming of optical data processors (ODP) is the relatively low accuracy with which they perform computations. The accuracy is limited primarily by the linear dynamic range of the devices used (light modulators, detectors, etc.) and optical and electronic noise that is present in the system, since ODPs are normally operated as analog systems. Improved device performance will lead to a corresponding improvement in accuracy, however it is unlikely that a dramatic change will come from this direction, in the foreseeable future. Current systems have a linear dynamic range of 30-40 dB which translates to 9-11 bits of accuracy, and the above numbers are rather optimistic. In order to construct optical systems that have 16 bits or 32 bits of accuracy, it is necessary to change our design philosophy. In a conventional ODP an optical wave in any point in space and/or time attains one of N distinguishable values. (N is a maximum set by the noise level and the linear dynamic range of the optical modulator or detector). Alternatively L separate points in space (or time), each attaining one of n_j distinguishable levels, can simultaneously represent the same variable encoded in a single point before provided that

$$N = \prod_{j}^{L} n_j, \tag{1}$$

where Π represents the product. Since all the n_j are positive integers $n_j \leq N$. Hence, instead of encoding each data point in a single location with a large dynamic range, we choose to represent it in L different locations each with a greatly reduced dynamic range requirement, which in turn results in increased overall accuracy. The L locations correspond either to different spatial resolution points or different times. Optical systems normally have high space-bandwidth product and speed. It is expected that trading either one for improved accuracy will be advantageous.

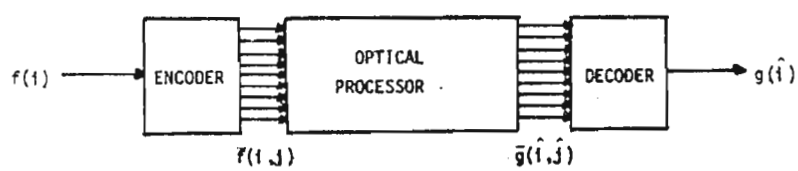

Figure 1 Optical processors operating on coded data.

In figure 1 the input data is denoted by f(i) where the index i is used to represent different data points. We desire to perform an arbitrary operation $\mathcal{F}$ on f(i) and produce an output function g(i). An encoder assigns a set of values $\bar{f}(i,j)$ to each input data point f(i), where the index j is used to represent the different encoded values. The optical system operates on the encoded function $\bar{f}(i,j)$. The output of the system must in general be decoded to produce $g(\hat{i})$. The critical step of course is the particular coding scheme used to obtain $\bar{f}(i,j)$. There are two criteria that we consider to evaluate a coding scheme: (a) We must be able to perform the encoding and decoding steps quickly and efficiently so that the high throughput rate of the optical system is maintained. (b) The coding used must be compatible with the optical system, which must be able to operate on the encoded data, $\bar{f}(i,j)$. In this paper we consider two separate coding schemes. In section

II we review residue arithmetic and we point out some of its advantages and disadvantages. In section III we present binary encoding and in section IV we apply this concept to the implementation of accurate optical correlators/convolvers.

II. Residue arithmetic

Residue arithmetic has received considerable attention recently [1, 2] as the basis for a coding scheme. The coding can be described by,

$$\bar{f}(i,j) = |f(i)|_{n_j} \tag{2}$$

where $|\cdot|_n$ represents the residue (remainder of the division of the number in brackets by n). The n_j are called moduli and they are relatively prime numbers. The maximum number f(i) that can be represented by this scheme is given by equation (1). The attractive attribute of residue arithmetic is that additions, subtractions and multiplications can be performed independently for each j with no carries required [1]. Thus if $\mathcal{F}$ consisted of these operations only, then the following two operations are equivalent.

$$g(\hat{i}) = \mathcal{F}\,[f(i)] \tag{3a}$$

$$\bar{g}(\hat{i},\hat{j}) = \mathcal{F}\,[\bar{f}(i,j)], \tag{3b}$$

where

$$\bar{g}(\hat{i},\hat{j}) = \tilde{\mathcal{F}}\,|g(\hat{i})|_{n_j} \tag{4}$$

and $\tilde{\mathcal{F}}$ operates on i only. Addition and multiplication in the residue system differ from the corresponding algebraic operations in that the residue of the algebraic result must be obtained. As a result the same optical system capable of implementing (3a) cannot be used to implement (3b). A number of optical systems have been proposed [1, 2, 3] that allow implementation of the elementary operations (+, -, x). However efficient integration of these building blocks into a system than performs more complicated signal processing operations (e.g. Fourier transform, filtering) has not been achieved. A system proposed by Huang, et al. (figure 12, reference 1) is the most promising attempt to date, since it does utilize the parallelism of optics to some extent. A second limitation of residue arithmetic is the difficulty in performing the coding and particularly the decoding steps. The method proposed by Psaltis and Casasent [2] utilizes optical convolution to achieve both operations. A time-integrating acousto-optic convolver [5] in the encoding section, and a space-integrating [5] version in the decoding section can convert 20-bit numbers in less than 1 msec from decimal to residue and visa-versa. Extensions of such systems can realize addition and multiplication in residue.

III. Binary arithmetic

We now turn our attention to a different encoding scheme that results from setting $n_j = 2$ for all j in (1). This of course is the familiar binary number system. The maximum dynamic range obtainable in this case is $N = 2^L$. The encoded function $\bar{f}(i,j)$ is related to f(i) by

$$f(i) = \sum_{j=0}^{L} \bar{f}(i,j)2^j, \tag{5}$$

where $\bar{f}(i,j)$ is 0 or 1. The obvious advantage of this coding scheme is that the minimum possible dynamic range is used ($n_j = 2$). In addition the encoding and decoding steps can be easily realized or they may even be unnecessary, if the optical system is interfaced with a digital machine. What must be determined is whether optical systems can operate efficiently on data that has been encoded as described in (5) and the advantages and disadvantages of such systems. This paper advances initial remarks on these issues.

The simplest way to add two data points (numbers), f(1) and f(2) with an optical system is by the use of the spatial integrating properly of a lens. This can be achieved if the two points spatially modulate the optical beam. If temporal modulation is used, the time integrating property of detectors can be utilized to implement the desired summation. Optical systems of course can add many numbers simultaneously. Algebraic addition is thus very easy to realize optically, whereas the same is not true for binary addition. For this reason we use algebraic addition to obtain the binary sum.

To describe how this is done, we write the addition of 2 data points as:

$$f(1) + f(2) = \sum_{j=0}^{L} [\bar{f}(1,j) + \bar{f}(2,j)]2^j. \tag{6}$$

From (6) we conclude that to perform the addition of f(1) and f(2) we can simply add (algebraically) the corresponding bits in the binary representation. This can be easily done with an optical system. Subtraction is similar to addition.

The multiplication of two numbers f(1) and f(2) is given by

$$f(1)\ f(2) = \left[\sum_{\ell=0}^{L} \bar{f}(1,\ell)2^{\ell}\right]\left[\sum_{k=0}^{L} \bar{f}(2,k)2^{k}\right]$$

$$= \sum_{j=0}^{2L}\left[\sum_{k=0}^{L} \bar{f}(1,j-k)\ \bar{f}(2,k)\right]2^{j} \tag{7}$$

where the substitution $\ell + k = j$ or $\ell = j-k$ has been used to obtain the last part of equation (7). The term in brackets in equation (7) can be recognized as the discrete convolution of $\bar{f}(1,j)$ and $\bar{f}(2,j)$. From (7) we see that to multiply two binary numbers we can convolve the bits representing the two numbers. The concept of performing binary multiplication using analog convolution was proposed by Whitehouse and Speiser [4], for the implementation of fast multipliers using charge-coupled-device convolvers. Hence both addition/subtraction and multiplication can be easily performed by optical computers when the coding is binary. The compatibility of binary coding and optical processing offers many advantages over residue coding. The relative disadvantage of this scheme is that the outputs are not in binary form. The sum of M numbers in binary will produce L output data points each requiring a dynamic range M. Similarly the product of two such numbers will produce 2L output points, the maximum dynamic range being L. The output dynamic range requirement is thus greater than the input one and it increases as more operations are performed on the input data. However there is a dramatic drop in the dynamic range requirement of this system when compared to a system where coding is not employed. To appreciate the advantage of such a system we consider a specific example. To multiply two numbers, each with a dynamic range $N = 2^{L} = 2^{16} \simeq 65{,}000$ we need an output dynamic range $N^2 = 2^{2L} = 2^{32} = 4.2 \times 10^{9}$. With the binary encoding the input and output dynamic ranges are 2 and 16 correspondingly.

Since the output of the binary-coded system is not in common binary form, we must address the decoding step. There are two reasons for this. First we normally wish to have our results in a conventional number system such as decimal or binary. Second, since the dynamic range of the computed numbers increases as more and more computations are performed, it may be desirable to convert them at selected stages of the processing to the binary representation before performing additional operations on these numbers. The output, $\bar{g}(j)$, of any operand in the binary-coded system has an equivalent value of the form

$$g = \sum_{j}^{L'} \bar{g}(j)2^{j} \tag{8}$$

where the numbers $\bar{g}(j)$ can have a dynamic range greater than two. If we wish to convert these numbers to an analog form, we simply implement (8) by electronic circuits. This circuit is similar to a digital-to-analog convertor. The conversion of the $\bar{g}(j)$'s in (8) to the true binary representation of g, is more complex. One way to achieve this is to A/D convert each $\bar{g}(j)$ and multiply the binary representation of $\bar{g}(j)$ by 2^{j} which is equivalent to upshifting by j bits. We can then use a digital adder to perform the summation in (8). Alternatively we can use analog circuitry to perform the decoding by the following algorithm

$$\bar{g}'(j+1) = \left|\bar{g}(j+1)\right|_2 + \frac{g(j) - \left|g(j)\right|_2}{2}, \tag{9}$$

where $\bar{g}'(j+1)$ denotes the value assigned to $\bar{g}(j+1)$ after the operation. Repeated application of (9) reduces all the $\bar{g}(j)$'s to either zero or one, i.e. the binary representation.

IV. System implementation using binary arithmetic

Equations (6) and (7) provide us the basic building blocks that we can use to implement more complex processing systems. The simplest system is a binary multiplier, which can be implemented using optical convolution. Such a multiplier can be very fast (< 100 nsec for a full 16 bit multiplication is possible). However, such an element would probably be too expensive and bulky to be competitive with digital multipliers. The parallel processing capability of the optics must be utilized in order to gain a clear advantage. This can be done by performing more complicated operations on large sets of data simultaneously, rather a simple multiplication of two numbers. Specifically we consider computing the convolution of two functions, f(x) and h(x), very accurately with an optical processor using the binary coding scheme.

The convolution is written as

$$g(\hat{x}) = \int f(x)\ h(\hat{x}-x)dx. \tag{10}$$

Using (5) we substitute the binary representations of $f(x)$ and $h(\hat{x}-x)$ to obtain

$$g(\hat{x}) = \int\left[\sum_{j=0}^{L} \bar{f}_j(x)2^{j}\right]\left[\sum_{k=0}^{L} \bar{h}_k(\hat{x}-x)2^{k}\right]dx$$

$$= \int \sum_{j=0}^{2L}\sum_{k=0}^{L} \bar{f}_{j-k}(x)\ h_k(\hat{x}-x)dx$$

$$= \sum_{j=0}^{2L}\left[\sum_{k=0}^{L} \int \bar{f}_{j-k}(x)\ \bar{h}_k(\hat{x}-x)dx\right]2^{j} \tag{11}$$

The term in brackets in the last part of the above equation is recognized as the two dimensional convolution of $\bar{f}_j(x)$ and $\bar{h}_j(x)$ in x and j (a continuous convolution in x and a discrete convolution in j). Equation (11)

is also recognized to be in the form predicted in (8). Consequently the decoding schemes that were described earlier apply here as well.

There are several OSP architectures [5] that can perform 2-D convolution. The result in (11) shows that we can utilize one of these architectures to convolve one-dimensional signals with very good accuracy. The specific architecture we propose for this uses acousto-optic (AO) devices. AO devices are at a much higher level of development than 2-D spatial light modulators and thus this particular system is preferable to other 2-D convolver architectures that require the use of 2-D devices. The proposed system is shown in figure 2. It utilizes N AO delay lines at P_1, one for each bit. Each AO line is driven by the corresponding $\bar{f}_j(t)$.

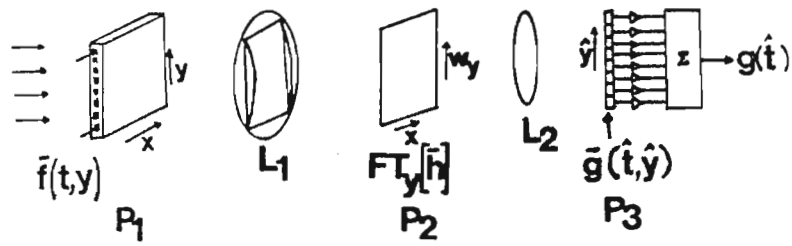

Figure 2 Two dimensional convolver using acousto-optic devices.

Thus the index j corresponds to the y-direction in figure 2. The light transmittance of P_1 is given by $\bar{f}_{y=j}(t-x/v) = \bar{f}(t-x/v,y)$ (where v is the velocity of sound). This function is Fourier transformed in y and imaged in x by the lens combination L_1. The light entering plane P_2 is thus described by

$$F(t-\frac{x}{v},w_y) = \int \bar{f}(t-\frac{x}{v},y)e^{-jw_y y}dy. \tag{12}$$

A two dimensional transparency is placed at P_2 whose complex transmittance is given by:

$$H(x,w_y) = \int \bar{h}(x,y=j)e^{-jw_y y}dy. \tag{13}$$

The light amplitude immediately following plane P_2 is simply the product of F and H. The lens L_2 takes the 2-D Fourier transform of this product and displays it at the output plane P_3. A one dimensional detector array in y is placed at x = 0, at plane P_3. The light detected is a function of time and space $\hat{y}$ (different detector elements). This P_3 pattern is described by:

$$\begin{aligned}\bar{g}(t,\hat{y}=j) &= \frac{1}{2\pi} \iint F(t-\tfrac{x}{v},w_y)\, H(x,w_y)e^{-jw_y y}dxdw_y \\ &= \iint \bar{f}(t-\tfrac{x}{v}\,\hat{y}-y)\, \bar{h}(x,y)dxdy\Big|_{\hat{y}=j}\end{aligned} \tag{14}$$

where $\hat{y}$ is the vertical spatial variable at the output. Equation (14) is the desired 2-D convolution (equation (11)). The outputs from the different detector elements can be weighted appropriately and added by analog electronics to provide an overall dynamic range much higher than conventional OSPs.

V. Experiment

We performed a simple experiment to demonstrate the advantages and feasibility of binary coding. A conventional Van der Lugt convolver/correlator [6] was used. Specifically, we computed the autocorrelation of the function f(x) shown in Fig. 3. The binary representation of f(x) is shown in Fig. 4. A 2-D transparency of the function $\bar{f}(x,y)$ in Fig. 4 was prepared and an off-axis Fourier transform hologram of it was constructed.

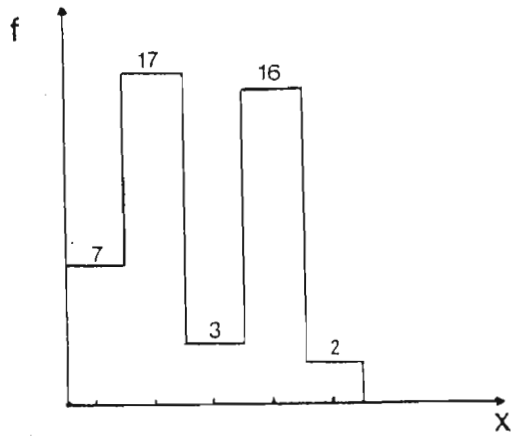

Figure 3 Input function, f(x) used in the experimental demonstration.

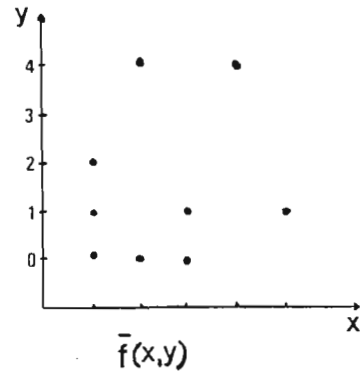

Figure 4 Binary-coded function $\bar{f}(x,y)$.

The hologram was placed in the Fourier plane of the Van der Lugt system and $\bar{f}(x, -y)$ was placed in the input plane. The output light distribution is the autocorrelation of $\bar{f}(x,y)$ in the x-direction and its autoconvolution in y. In Fig. 5 cross sectional scans of the output function $\bar{g}(\hat{x},\hat{y})$ along the $\hat{y}$ direction (the different bits) are shown for all appropriate values of $\hat{x}$ (the correlation shift variable). The same values for

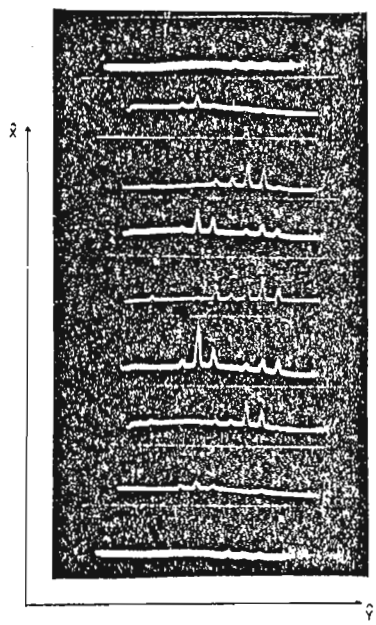

Figure 5 Cross sectional scans of the optically computed function $\bar{g}(x,y)$.

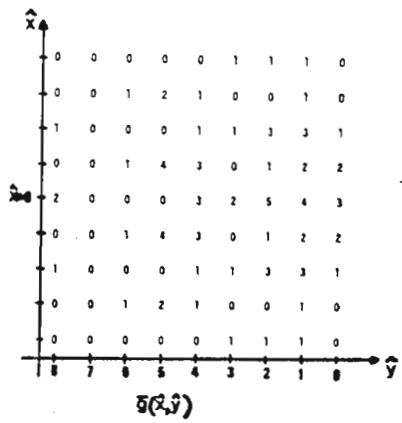

Figure 6 Numerical evaluation of the function $\bar{g}(\hat{x},\hat{y})$.

$\bar{g}(\hat{x},\hat{y})$ were calculated separately and they are tabulated in Fig. 6. Comparing Figs. 5 and 6 we find that the optically computed values are absolutely accurate and hence from these values the autocorrelation function $R(\hat{x}) = g(\hat{x})$ can be obtained using equation (8). R(x) is shown in Fig. 7.

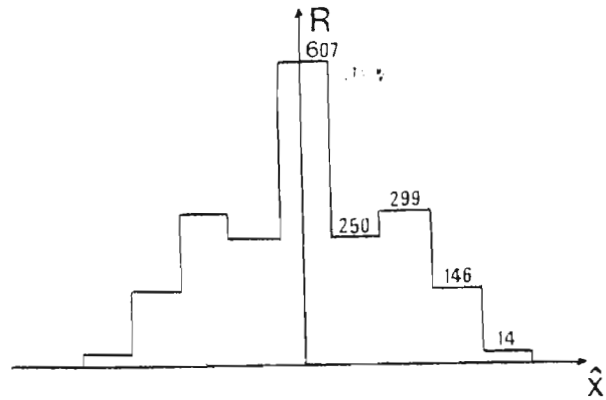

Figure 7 Autocorrelation function of f(x).

This experiment points out some interesting characteristics of binary coding. If coding is not employed the input function must be recorded with 17:1 accuracy and the autocorrelation must be computed and detected with a 607:1 accuracy. With coding these requirements are only 2:1 and 5:1 respectively. In addition the values of $\bar{g}(\hat{x},\hat{y})$ for the most significant bits (e.g. for $\hat{y}$ = 6-8) are 0, 1 or 2 whereas the larger values of $\bar{g}(\hat{x},\hat{y})$ appear at less significant bits. This will in general be desirable since it will be less likely that an error will occur at the higher bits which would result in a larger overall error. In general, if the space-bandwidth product of f(x) is N and we use L bits to represent it, the function $\bar{g}(\hat{x},\hat{y})$ will have a maximum value N at the most significant bit and an overall maximum value N x L.

$$\max[\bar{g}(\hat{x},\hat{y})] = NL. \tag{15}$$

The maximum value of $g(\hat{x})$ is given by

$$\max[g(\hat{x})] = N(2^L-1)^2. \tag{16}$$

Thus the use of binary coding results in a decrease in the dynamic range requirement at the output of the system by a factor $L/(2^L-1)^2$.

VI. Acknowledgments

The support of the National Science Foundation for the research reported upon is gratefully acknowledged. We thank Harper Whitehouse and Jeffrey Speiser of NOSC for introducing us to the idea of digital multiplication by convolution. This arose during a joint CMU/NOSC program on generalized optical signal processors.

VII. References

1. A. Huang, Y. Tsunoda, and J. W. Goodman, Appl. Opt., 18, 148 (1979).
2. D. Psaltis and D. Casasent, Appl. Opt., 18, 163 (1979).
3. S. Collins, Proc. Soc. Photo-Opt. Instrum. Eng., 128 (1977).
4. H. J. Whitehouse and J. Speiser, "Aspects of Signal Processing with Emphasis on Underwater Acoustics", Ed. G. Tacconi, Part II, Reidel Publishing Company.
5. D. Psaltis and D. Casasent, Optical Engineering, February 1980.
6. A. Van der Lugt, IEEE Trans. Info. Theory, IT-10, 139 (1964).