DISCRETE POLYNOMIAL TRANSFORM REPRESENTATION USING BINARY MATRICES AND FLOW DIAGRAMS

Maurice F. Aburdene
Richard J. Kozick
Ryan S. Magargle
Justin D. Maloney-Hahn
Christian M. Coviello

Bucknell University

ABSTRACT

This paper presents a new method for computing discrete polynomial transforms. The method is shown for the Hermite, binomial, and Laguerre transforms. The new method factors Pascal's matrix into binary matrices. Constructing the flow diagrams for the transform matrices requires only additions and N-2 multipliers for N-point Hermite and binomial transforms, and 2N multipliers for an N-point Laguerre transform. The method involves a three-stage process where stages 1 and 3 are identical for all three transforms.

1. INTRODUCTION

The discrete Hermite, binomial, and Laguerre transforms are used in control, signal processing, and communications [1,2]. Pascal's matrix can be used to compute these discrete transforms [3]. This paper presents a method for computing the transform matrices using binary matrices to facilitate the creation of flow diagrams.

These transforms are based on the discrete forms of the Hermite, binomial, and Laguerre polynomials. The polynomials are:

Hermite [4]:

$$p(x,i,N)=\sum_{l=0}^{i}(-2)^l\binom{i}{l}\frac{x^{(l)}}{N^{(l)}}$$

Binomial [4]:

$$p(x,i,N)=\binom{N}{x}\sum_{l=0}^{i}(-2)^l\binom{i}{l}\frac{x^{(l)}}{N^{(l)}}$$

Laguerre [3]:

$$p(x,i,\theta)=\theta^i\sum_{l=0}^{i}(-1)^l\binom{i}{l}\left(\frac{1-\theta}{\theta}\right)\binom{x}{l},$$

in which |θ|<1 and x=0,1…,N, and i=0,1, …,N. θ is a parameter of the Laguerre polynomial. $\binom{a}{b}=\frac{a!}{(a-b)!b!}$, i is the order of the polynomial, N is a parameter of the polynomial, and $x^{(l)}$ is the backward factorial function defined by

$$x^{(l)}=x(x-1)(x-2),...,(x-l+1)\ .$$

Using these discrete polynomials, the (N+1) x (N+1) transform matrices are defined as:

$$[\mathrm{T}]=\begin{bmatrix} p(0,0,N) & p(1,0,N) & \dots & p(N,0,N) \\ p(0,1,N) & p(1,1,N) & \dots & p(N,1,N) \\ p(0,2,N) & p(1,2,N) & \dots & p(N,2,N) \\ \vdots & \vdots & \ddots & \vdots \\ p(0,N,N) & p(1,N,N) & \cdots & p(N,N,N) \end{bmatrix}$$

where p(x,i,N) represents the general discrete polynomial evaluated at x,i,N. It was shown that the T matrices can be computed using the following definitions [3]:

$[A]_{i,j}=\binom{i+j}{j}$, Pascal's matrix

$[B]_{i,j}=\binom{i}{j}$, Pascal's matrix (shifted), where $B^TB=A$ (B^T is the transpose of B)

$[I^-]_{j,j}=(-1)^j,\ j=0,1,...,N$

$[Q]_{i,i}=\theta^i, i=0,1,...,N$

$\left[\tilde{I}\right]_{i,N-i}=1$ and 0 elsewhere

Defining the Hermite, binomial, and Laguerre matrices as H_N, B_N, and L_N respectively and using the above definitions and equations, it was shown that the following discrete transform matrices are formed [3]:

Hermite:

$$H_N = [I^{-}QB]^T[Diag[\tilde{I}(B^TB)]]^{-1}B \qquad (1)$$

Binomial:

$$B_N = [I^{-}QB]^T B \qquad (2)$$

Laguerre:

$$L_N = Q[I^{-}Diag[[I^{-}B]^T\Theta^T]Q^{-1}B]^T B\ , \qquad (3)$$

where $\Theta = [\theta\ \ \theta^1\ \ \theta^2\ \ \ldots\theta^N]$ and Diag[X] is a diagonal matrix, whose elements are the diagonal elements of X.

2. FACTORING THE B MATRIX

These transforms can be further simplified by factoring B into a series of binary matrices [6]. This simplification reduces the multiplication associated with the B matrices into a series of additions. To construct the binary matrices representing the $(N + 1)$ x $(N + 1)$ B matrix we use,

$$[I_N]_{i,j} = \begin{cases} 1 \text{ for } i=j \\ 0 \text{ elsewhere} \end{cases}$$

the $(k + 1)$ x $(k + 1)$ matrix, S_k, whose elements are given by:

$$[S_k]_{i,j} = \begin{cases} 1 \text{ for } j \geq i \\ 0 \text{ for } j < i \end{cases}$$

and

$$G_k = \begin{bmatrix} I_{N-1-k} & 0 \\ 0 & S_k \end{bmatrix}$$

where $k = 1,2,\ldots,N$.

In addition, the factorization of B is given by

$$B_N = \prod_{i=1}^{N} G_i$$

For example, the G expansion of a 4x4 (N=3) shifted Pascal's matrix (B) is

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and}$$

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, G_3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$B=G_1G_2G_3$.

The Hermite transform matrix is given by

$$H_N = [I^{-}QG_1G_2..G_N]^T[Diag[\tilde{I}(B^TB)]]^{-1}G_1G_2..G_N \qquad (4)$$

Using the identity $[ABC]^T=C^TB^TA^T$, we can further simplify the expression in equation 4 to a series of matrix multiplications. The only problem left to resolve is the diagonal operation in the middle of the transform, $Diag[\tilde{I}(B^TB)]$.

For the Hermite transform, the diagonal elements of the $Diag[\tilde{I}(B^TB)]$ operation are binomial coefficients.

$$\binom{N}{k}, \text{ where } k=0,1,\ldots,N.$$

The inverse of a diagonal matrix, $[Diag[\tilde{I}(B^TB)]]^{-1}$, is simply the inverse of the diagonal elements, which gives the W_N matrix as:

$$W_N = [Diag[\tilde{I}(B^TB)]]^{-1} = \begin{bmatrix} \frac{1}{\binom{N}{0}} & 0 & 0 & 0 \\ 0 & \frac{1}{\binom{N}{1}} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{\binom{N}{N}} \end{bmatrix}$$

For the Laguerre transform matrix, W_N is a little more complex because the diagonal operation, $Diag[[I^{-}B]^T\Theta^T]$, is dependent on Θ, where $\Theta = \left[\theta\ \theta^1\ \theta^2 \ldots \theta^N\right]$ and $|\theta| < 1$. Observing the general forms of the Γ, B, and Q components leads to the following general form for the diagonals of W_N for the Laguerre transform:

$$\theta + \sum_{k=1}^{j}(-1)^k\theta^k\binom{j}{k},$$

where $j=0,1,\ldots,N$ and is the current row in the W_N matrix. The general form of W_N for the Laguerre transform is

$W_N = Diag[[I^{-}B]^T\Theta^T] =$

$$\begin{bmatrix} \theta & 0 & 0 & 0 & 0 \\ 0 & \theta+\sum_{k=1}^{1}(-1)^k\theta^k\binom{1}{k} & 0 & 0 & 0 \\ 0 & 0 & \theta+\sum_{k=1}^{2}(-1)^k\theta^k\binom{2}{k} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \theta+\sum_{k=1}^{N}(-1)^k\theta^k\binom{N}{k} \end{bmatrix}$$

So the final simplified form of the transforms will be:

$$H_N = G_N^T ... G_2^T G_1^T Q I^{-} W_N G_1 G_2 .. G_N \quad (5)$$

$$L_N = Q G_N^T ... G_2^T G_1^T Q^{-} W_N I^{-} G_1 G_2 .. G_N \quad (6)$$

$$B_N = G_N^T ... G_2^T G_1^T Q I^{-} G_1 G_2 .. G_N \quad (7)$$

2.1 Factoring the *B* Matrix to Form the Discrete Hermite Transform Flow Diagram

As noted earlier, the shifted Pascal matrix *B* can be represented as a multiplication of the binary matrices, G_k. In this section the factored matrices of *B* will be used in order to create a flow diagram that will detail a simplified method of computation for the discrete Hermite polynomial transform.

First, the G_k matrices should be defined and calculated. The matrix is formed by placing two matrices within an (*N*+1) x (*N*+1) matrix:

$$G_k = \begin{bmatrix} I_{N-1-k} & 0 \\ 0 & S_k \end{bmatrix}, \text{ where } k=1,2,\ldots,N$$

$$[I_N]_{i,i} = 1$$

$$[S_k]_{i,j} = \begin{cases} 1 \text{ if } j \geq i \\ 0 \text{ if } j < i \end{cases}, \text{ where } [S_k] \text{ is a } (k+1) \text{ x } (k+1) \text{ matrix.}$$

For this example *N*=2. Therefore, we will need to find G_k for *k* = 1 and 2.

For G_1 we have $G_1 = \begin{bmatrix} I_{2-1-1} & 0 \\ 0 & S_1 \end{bmatrix}$, where $I_0 = [1]$ and $S_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

Placing I_0 and S_1 into the G_1 matrix yields $G_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

Following a similar process for G_2 we have $G_2 = \begin{bmatrix} I_{2-1-2} & 0 \\ 0 & S_2 \end{bmatrix}$,

where, by definition, I_{-1} does not exist and $S_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

Therefore, placing S_2 into the matrix yields $G_2 = [S_2]$,

$$G_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

The shifted Pascal's matrix, *B*, can then be represented as the product of the binary summation matrices, G_k :

$$B_2 = G_1 G_2$$

Therefore substituting into equation 1 yields,

$$H_N = G_2^T G_1^T Q^T I^{-T} W_2 G_1 G_2 ,$$

$$H_N = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} Q^T I^{-T} \; W_2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

Starting the multiplication with the sampled point vector will yield the necessary information to create the *N*=2 discrete Hermite transform flow diagram as shown in Figure 1.

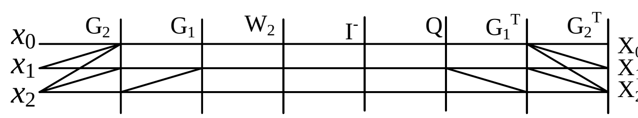

Figure 1. 3x3 Example Hermite Transform Flow Diagram.

3. FLOW DIAGRAM REPRESENTATION NOTES

The flow diagrams represent the matrix operations in an easy to read algorithmic form. This allows for an easy translation into low-level chip programming. The flow diagram also makes it easier to identify similar operations between the transforms so that they may be implemented only once on-chip. In Figure 2, a flow diagram for the 8x8 Hermite transform is shown.

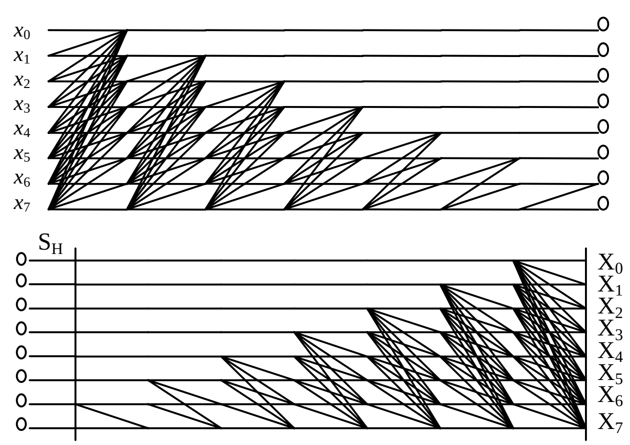

Figure 2. 8x8 Hermite Transform Flow Diagram.

It can be noticed by drawing flow diagrams for the Hermite, binomial, and Laguerre transforms that, except for the scaling factors, the transforms have identical summation series. The form is shown below in Figure 3.

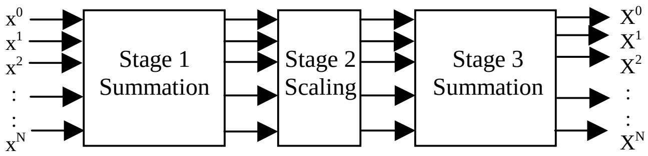

Figure 3. Standard Form for Transform Flow Diagrams.

In Figure 3, the input vector consists of the sampled data points. The points enter stage 1, which is identical for all three transforms and is a summation series from $G_N..G_2G_1$. The scaling matrices specific to each transform are:

$$S_B = I^{-}Q$$
$$S_H = W_N I^{-}Q$$
$$S_L = I^{-}W_N Q^{-},$$

Within the scaling diagonal matrices above, W_N and Q are transform specific and have been previously defined. Finally, there is stage 3, which is also identical for all three transforms and is a G^T summation from $G_1^T G_2^T ... G_N^T$. The result yields the transform coefficients X^0 to X^N. It should be noted that for the Laguerre transform a multiplication with Q is added after stage 3.

4. INVERSE TRANSFORMS

To construct the inverse transform matrices, the transpose of the forward transform is utilized. The Laguerre , Hermite, and binary matrices are orthogonal with respect to weighting sequences [4]. By using appropriate normalizing factors for the transform matrices, we can obtain the inverse transforms [4,5]. In this paper, we did not include the normalizing factors to demonstrate the factorization method.

Taking the transpose of the three transforms matrices yields:

$$H_N^T = G_N^T ... G_2^T G_1^T W_N I^{-} Q G_1 G_2 .. G_N$$
$$L_N^T = G_N^T ... G_2^T G_1^T I^{-} W_N Q^{-} G_1 G_2 .. G_N Q$$
$$B_N^T = G_N^T ... G_2^T G_1^T I^{-} Q G_1 G_2 .. G_N$$

The general form of the inverse transform is the same as the forward transform and can be seen in Figure 4.

The difference in the forward and inverse transforms is in Stage 2. While the forward transform has S_B, S_H, and S_L, the inverse transform has S_B^T, S_H^T, and S_L^T. Also, the Q multiplier that was at the end of the forward Laguerre transform is now at the beginning, before Stage 1.

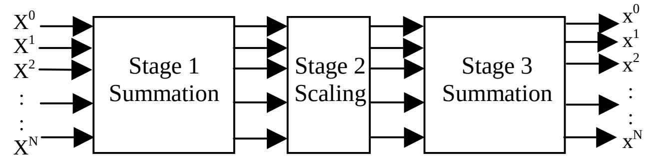

Figure 4. Standard Form for Inverse Transform Flow Diagrams.

5. CONCLUSION

This paper has defined the discrete Hermite, binomial, and Laguerre transforms using binary matrices. This allows flow diagrams to be constructed that provide visual models that demonstrate the ease of hardware implementation for the transforms. Fortunately, all three transforms have the same form and can thus share the same hardware when implemented. It was also shown that the inverse transform for each can be easily computed using the same flow diagrams with only the scalar multiplier being changed.

6. REFERENCES

[1] G. Mandyam and N. Ahmed. “The Discrete Laguerre Transform: Derivation and Applications.” *IEEE Transactions on Signal Processing*, vol. 44, pp. 2925-2931, Dec 1996.

[2] J.B. Martens. “Local Orientation Analysis in Images by Means of the Hermite Transform.” *IEEE Transactions on Image Processing*, vol. 6, pp. 1103-1116, August 1997.

[3] M.F. Aburdene and J.E. Dorband. “Unification of Legendre, Laguerre, Hermite, and Binomial Discrete Transforms Using Pascal’s Matrix.” *Multidimensional Systems and Signal Processing*, vol. 5, pp. 301-305, 1994.

[4] R.A. Haddad and T.W. Parsons. *Digital Signal Processing: Theory, Applications, and Hardware*. New York: Computer Science Press, 1991.

[5] N. Morrison. *Introduction to Sequential Smoothing and Prediction*. New York: McGraw-Hill, 1969.

[6] R. Brawer and M. Pirovino. “The Linear Algebra of the Pascal Matrix.” *Linear Algebra and Its Application*, vol. 174, pp. 13-23, 1992.

.