Three additional facts about tensor products that we shall use are as follows:
If A and B are matrices, then

$$(A \otimes B)' = A' \otimes B'. \tag{6.2.3}$$

If A and B are nonsingular matrices, then

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \tag{6.2.4}$$

If A and B are permutation matrices, then so is

$$A \otimes B\,. \tag{6.2.5}$$

These results are not difficult to verify and we leave it to the reader to convince himself of their validity.

We now turn our attention to some basic properties of the Fourier matrix. If $F = \psi(\mathbf{f})$ denotes the Fourier matrix corresponding to the Fourier template $\mathbf{f}$ defined by Eq. 6.1.12, then the image algebra formulation of the DFT (Eq. 6.1.13) is equivalent to the vector-matrix product

$$\hat{\mathbf{a}} = \mathbf{a} \cdot F\,. \tag{6.2.6}$$

The vector-matrix representation of the DFT facilitates the analysis of the transform. In order to represent Eq. 6.2.6 more concisely, we let $F_n = (f_{kj})_{n\times n}$ denote the $n \times n$ Fourier matrix, where

$$f_{kj} = \mathbf{f}_j(k) = e^{-\frac{2\pi i}{n}kj},\ \ 0 \le j, k \le n-1\,.$$

Replacing F by F_n in Eq. 6.2.6 specifies the size of both the transform matrix and signal length.

The representation of Fourier matrices is commonly simplified by setting $\omega_n = e^{-\frac{2\pi i}{n}}$ and noting that

$$\begin{gathered} f_{kj} = \omega_n^{kj} \\ \omega_n^0 = \omega_n^n = \omega_n^{kn} = 1 \\ \omega_n^{n/2} = -1 \\ \omega_n^{n/4} = -i \\ \omega_n^{3n/4} = i \\ \omega_n^{k+n} = \omega_n^k \end{gathered}$$

Thus,

$$F_1 = (1),\ \ F_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and

$$F_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega_4 & \omega_4^2 & \omega_4^3 \\ 1 & \omega_4^2 & \omega_4^4 & \omega_4^6 \\ 1 & \omega_4^3 & \omega_4^6 & \omega_4^9 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}.$$

Similarly,

$$F_8 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_8 & \omega_8^2 & \omega_8^3 & -1 & -\omega_8 & -\omega_8^2 & -\omega_8^3 \\ 1 & \omega_8^2 & -1 & \omega_8^2 & 1 & \omega_8^2 & -1 & -\omega_8^2 \\ 1 & \omega_8^3 & -\omega_8^2 & \omega_8 & -1 & -\omega_8^3 & \omega_8^2 & -\omega_8 \\ - & - & - & - & + & - & - & - & - \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\omega_8 & \omega_8^2 & -\omega_8^3 & -1 & \omega_8 & -\omega_8^2 & \omega_8^3 \\ 1 & -\omega_8^2 & -1 & \omega_8^2 & 1 & -\omega_8^2 & -1 & \omega_8^2 \\ 1 & -\omega_8^3 & -\omega_8^2 & -\omega_8 & -1 & \omega_8^3 & \omega_8^2 & \omega_8 \end{array}\right).$$

The *conjugate* and the *conjugate transpose* of a matrix $A = (a_{ij}) \in \mathbb{C}_{m\times n}$ are denoted by

$$A^* = \left(a_{ij}^*\right) \text{ and } A^H = \left(a_{ji}^*\right),$$

respectively. We say that *A* is *symmetric* if $A' = A$, *Hermitian* if $A^H = A$, and *unitary* if $A^H = A^{-1}$.

6.2.1 Theorem. *F_n is symmetric.*

Proof: This is obvious since $f_{kj} = \omega_n^{kj} = \omega_n^{jk} = f_{jk}$. Therefore, $F_n' = F_n$.

Q.E.D.

As an easy corollary we have that $F_n^H = (F_n^*)' = F_n^*$. Thus, F_n is not Hermitian if $n > 2$. However, since F_n is a real-valued matrix for $n = 1$ or 2, F_n is Hermitian whenever $n = 1$ or 2.

6.2.2 Theorem. $F_n^H \cdot F_n = n \cdot I_n$.

Proof: Let $A = F_n^H \cdot F_n$ and consider the (l, j)th entry

$$a_{lj} = \sum_{k=0}^{n-1} \left[\left(\omega_n^{kl}\right)^* \cdot \omega_n^{kj}\right] = \sum_{k=0}^{n-1} \omega_n^{k(j-l)}.$$

If $j = l$, then $a_{lj} = n$. If $j \neq l$, then $\omega_n^{j-l} \neq 1$. Therefore, $\omega_n^{j-l} - 1 \neq 0$. However,

$$\left(1 - \omega_n^{j-l}\right) a_{lj} = a_{lj} - \omega_n^{j-l} a_{lj} = \sum_{k=0}^{n-1} \omega_n^{k(j-l)} - \sum_{k=1}^{n} \omega_n^{k(j-l)} = 1 - \omega_n^{(j-l)n} = 0.$$

Therefore, $a_{lj} = 0$ whenever $j \neq l$.
Thus,

$$A = \begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & n \end{pmatrix} = n \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Q.E.D.

According to this theorem, $I_n = \frac{1}{n} F_n^H \cdot F_n$ or, equivalently,

$$F_n^{-1} = \frac{1}{n} F_n^H \,. \tag{6.2.7}$$

Thus, F_n is *almost* unitary since it is a *scaled* unitary matrix.

Multiplying the Fourier transform equation $\hat{\mathbf{a}} = \mathbf{a} \cdot F_n$ by F_n^{-1} one obtains

$$\hat{\mathbf{a}} \cdot F_n^{-1} = \mathbf{a} \,. \tag{6.2.8}$$

Accordingly, **a** can be obtained from $\hat{\mathbf{a}}$ using matrix inversion.

Using Eqns. 6.2.7 and 6.2.8 one obtains

$$\mathbf{a} = \hat{\mathbf{a}} \cdot F_n^{-1} = \frac{1}{n} \hat{\mathbf{a}} \cdot F_n^H = \frac{1}{n} \hat{\mathbf{a}} \cdot (F_n^*)' = \frac{1}{n} \hat{\mathbf{a}} \cdot F_n^*$$

or

$$\mathbf{a} = \frac{1}{n} \hat{\mathbf{a}} \cdot F_n^* \tag{6.2.9}$$

which is, of course, equivalent to the image algebra expression $\mathbf{a} = \frac{1}{n}(\hat{\mathbf{a}} \bigoplus \mathbf{f}^*)$. The significance of Eq. 6.2.9 is that the original signal **a** may be recovered from the transformed version *without* the need for matrix inversion. This property is especially useful when computing two-dimensional transforms where the transformation matrices may be quite large and, therefore, the computation of general inverses impractical.

6.3 Shuffle Permutations and the Radix Splitting of Fourier Matrices

If *p* divides *n*, then it is possible to relate F_n to $F_{n/p}$ and F_p. This relationship, known as the *radix-p splitting* of F_n, allows for the computation of an *n*-point DFT from DFTs of smaller size. Repetition of this process is the heart of today's algorithms for the efficient computation of the DFT. The construct that provides the link between F_n and the Fourier matrices of smaller dimensions is a permutation matrix that *shuffles* or *sorts* the column (or rows) of a matrix in some desired order.

Suppose $n = pm$ with $1 < p < m$. Let

$$r : \mathbb{Z}_p \times \mathbb{Z}_{n/p} \to \mathbb{Z}_n \text{ and } c : \mathbb{Z}_p \times \mathbb{Z}_{n/p} \to \mathbb{Z}_n$$

denote the row and column scanning maps defined by

$$r : (k,l) \rightarrowtail k\frac{n}{p} + l \;\; and \;\; c : (k,l) \rightarrowtail lp + k \,,$$

respectively. The (p,n)–*shuffle permutation* $\sigma_{p,n} : \mathbb{Z}_n \to \mathbb{Z}_n$ is defined as $\sigma_{p,n} = r \circ c^{-1}$. Equivalently,

$$\sigma_{p,n}(i) = j \;\Leftrightarrow\; \exists (k,l) \in \mathbb{Z}_p \times \mathbb{Z}_{n/p} \;\; s.t. \; c(k,l) = i \text{ and } r(k,l) = j.$$

An easy way of visualizing the (p,n)–shuffle permutation is to label the points of the array $\mathbf{X} = \mathbb{Z}_p \times \mathbb{Z}_{n/p}$ in row scan order, then taking the transpose of **X** and listing the elements of the

transposed array in row–scan order. The mapping of the subscripts corresponds to the permutation of elements of $\mathbb{Z}_n$ by $\sigma_{p,n}$. Thus, if

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{\frac{n}{p}-1} \\ \mathbf{x}_{\frac{n}{p}} & \mathbf{x}_{\frac{n}{p}+1} & \mathbf{x}_{\frac{n}{p}+2} & \cdots & \mathbf{x}_{2\frac{n}{p}-1} \\ \mathbf{x}_{2\frac{n}{p}} & \mathbf{x}_{2\frac{n}{p}+1} & \mathbf{x}_{2\frac{n}{p}+2} & \cdots & \mathbf{x}_{3\frac{n}{p}-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{(p-1)\frac{n}{p}} & \mathbf{x}_{(p-1)\frac{n}{p}+1} & \mathbf{x}_{(p-1)\frac{n}{p}+2} & \cdots & \mathbf{x}_{n-1} \end{pmatrix},$$

then

$$\sigma_{p,n} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{p-1} & \mid & \mathbf{x}_p & \mathbf{x}_{p+1} & \cdots & \mathbf{x}_{2p-1} & \mid & \cdots & \mid & \mathbf{x}_{n-p} & \cdots & \mathbf{x}_{n-1} \\ \mathbf{x}_0 & \mathbf{x}_{\frac{n}{p}} & \mathbf{x}_{2\frac{n}{p}} & \cdots & \mathbf{x}_{(p-1)\frac{n}{p}} & \mid & \mathbf{x}_1 & \mathbf{x}_{\frac{n}{p}+1} & \cdots & \mathbf{x}_{(p-1)\frac{n}{p}+1} & \mid & \cdots & \mid & \mathbf{x}_{\frac{n}{p}-1} & \cdots & \mathbf{x}_{n-1} \end{pmatrix}$$

Equivalently, we have

$$\sigma_{p,n} = \begin{pmatrix} 0 & 1 & 2 & \cdots & p-1 & \mid & p & p+1 & \cdots & 2p-1 & \mid & \cdots & \mid & n-p & \cdots & n-1 \\ 0 & \frac{n}{p} & \frac{2n}{p} & \cdots & \frac{(p-1)n}{p} & \mid & 1 & \frac{n}{p}+1 & \cdots & (p-1)\frac{n}{p}+1 & \mid & \cdots & \mid & \frac{n}{p}-1 & \cdots & n-1 \end{pmatrix}$$

The horizontal bars are added to emphasize the structure of the shuffle permutation.

The permutation matrix $\Pi(p,n) \equiv P_{\sigma_{p,n}}$ is called the *mod p perfect shuffle*.

6.3.1 Examples:

(i) Suppose $n = 10$ and $p = 2$. In this case we have

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \\ \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 & \mathbf{x}_8 & \mathbf{x}_9 \end{pmatrix}.$$

Hence

$$\begin{aligned} \sigma_{2,10} &= \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mid & \mathbf{x}_2 & \mathbf{x}_3 & \mid & \mathbf{x}_4 & \mathbf{x}_5 & \mid & \mathbf{x}_6 & \mathbf{x}_7 & \mid & \mathbf{x}_8 & \mathbf{x}_9 \\ \mathbf{x}_0 & \mathbf{x}_5 & \mid & \mathbf{x}_1 & \mathbf{x}_6 & \mid & \mathbf{x}_2 & \mathbf{x}_7 & \mid & \mathbf{x}_3 & \mathbf{x}_8 & \mid & \mathbf{x}_4 & \mathbf{x}_9 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & \mid & 2 & 3 & \mid & 4 & 5 & \mid & 6 & 7 & \mid & 8 & 9 \\ 0 & 5 & \mid & 1 & 6 & \mid & 2 & 7 & \mid & 3 & 8 & \mid & 4 & 9 \end{pmatrix}. \end{aligned}$$

Since $\Pi(2,10) = P_{\sigma_{2,10}} = (p_{ij})$ is defined by

$$p_{ij} = \begin{cases} 1 & \text{if } \sigma_{2,10}(i) = j \\ 0 & \text{otherwise,} \end{cases}$$

we have

$$\Pi(2,10) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

(ii) If $n = 15$ and $p = 3$, then

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \\ \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 & \mathbf{x}_8 & \mathbf{x}_9 \\ \mathbf{x}_{10} & \mathbf{x}_{11} & \mathbf{x}_{12} & \mathbf{x}_{13} & \mathbf{x}_{14} \end{pmatrix}.$$

and

$$\sigma_{3,15} = \left(\begin{array}{ccc|ccc|ccc|ccc|ccc} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 & \mathbf{x}_8 & \mathbf{x}_9 & \mathbf{x}_{10} & \mathbf{x}_{11} & \mathbf{x}_{12} & \mathbf{x}_{13} & \mathbf{x}_{14} \\ \mathbf{x}_0 & \mathbf{x}_5 & \mathbf{x}_{10} & \mathbf{x}_1 & \mathbf{x}_6 & \mathbf{x}_{11} & \mathbf{x}_2 & \mathbf{x}_7 & \mathbf{x}_{12} & \mathbf{x}_3 & \mathbf{x}_8 & \mathbf{x}_{13} & \mathbf{x}_4 & \mathbf{x}_9 & \mathbf{x}_{14} \end{array}\right)$$

$$= \left(\begin{array}{ccc|ccc|ccc|ccc|ccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 0 & 5 & 10 & 1 & 6 & 11 & 2 & 7 & 12 & 3 & 8 & 13 & 4 & 9 & 14 \end{array}\right).$$

Suppose *A* is a $k \times 10$ matrix and $A(j)$ denotes the *j*th column of *A*. Then according to the above example,

$$\begin{aligned} A \cdot \Pi(2,10) &= (A(0), A(1), \ldots, A(9)) \cdot \Pi(2,10) \\ &= (A(0), A(2), A(4), A(6), A(8) \,|\, A(1), A(3), A(5), A(7), A(9)). \end{aligned}$$

Thus, $A \cdot \Pi(2,10)$ is just *A* with its even-indexed columns grouped first. In general, if $n = 2m$, then

$$A \cdot \Pi(2,n) = (A(0), A(2), \ldots, A(n-2) \,|\, A(1), A(3), \ldots, A(n-1)). \tag{6.3.1}$$

This is the reason that $\Pi(2,n)$ is referred to as the *even-odd sort* permutation matrix. It is customary to denote the even-odd sort permutation matrix $\Pi(2,n)$ by Π_n.

As mentioned, the shuffle permutation matrix plays a key role in linking F_n with $F_{n/p}$. Consider the case $n = 4$. Here we have

$$\Pi_4' \cdot F_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \\ 1 & i & -1 & -i \end{pmatrix} = \begin{pmatrix} F_4(0) \\ F_4(2) \\ F_4(1) \\ F_4(3) \end{pmatrix},$$

where $F_4(j)$ denotes the *j*th row of F_4. Thus, $\Pi_4' \cdot F_4$ is just F_4 with its even-indexed rows grouped first. Of course, since $(\Pi_4' \cdot F_4)' = F_4' \cdot (\Pi_4')' = F_4 \cdot \Pi_4$, this observation also follows from Eq. 6.3.1.

Note that in our case (i.e., $n = 4$)

$$\Pi_4' \cdot F_4 = \left(\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ - & - & - & - \\ 1 & -i & -1 & i \\ 1 & i & -1 & -i \end{array}\right) = \begin{pmatrix} F_2 & F_2 \\ F_2 \cdot \Omega_2 & -F_2 \cdot \Omega_2 \end{pmatrix}, \tag{6.3.2}$$

Where

$$\Omega_2 = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} = diag(1, \omega_4).$$

Thus each block of $\Pi_4' \cdot F_4$ is either F_2 or a diagonal scaling of F_2. Furthermore, the right-most matrix in Eq. 6.3.2 factors as

$$\begin{pmatrix} F_2 & F_2 \\ F_2 \cdot \Omega_2 & -F_2 \cdot \Omega_2 \end{pmatrix} = \begin{pmatrix} F_2 & \mathrm{O} \\ \mathrm{O} & F_2 \end{pmatrix} \begin{pmatrix} I_2 & I_2 \\ \Omega_2 & -\Omega_2 \end{pmatrix},$$

where I_2 denotes the 2×2 identity matrix. Hence, multiplying Eq. 6.3.2 by $\Pi_4^{-1} = \Pi_4' = \Pi_4$ we obtain

$$F_4 = \Pi_4 \begin{pmatrix} F_2 & \mathrm{O} \\ \mathrm{O} & F_2 \end{pmatrix} \begin{pmatrix} I_2 & I_2 \\ \Omega_2 & -\Omega_2 \end{pmatrix}.$$

Therefore, if $\mathbf{a} = (\mathbf{a}(0), \mathbf{a}(1), \mathbf{a}(2), \mathbf{a}(3))$, then

$$\begin{aligned} \mathbf{a} \cdot F_4 &= \mathbf{a} \cdot \Pi_4 \begin{pmatrix} F_2 & \mathrm{O} \\ \mathrm{O} & F_2 \end{pmatrix} \begin{pmatrix} I_2 & I_2 \\ \Omega_2 & -\Omega_2 \end{pmatrix} \\ &= (\mathbf{a}(0), \mathbf{a}(2), \mathbf{a}(1), \mathbf{a}(3)) \begin{pmatrix} F_2 & \mathrm{O} \\ \mathrm{O} & F_2 \end{pmatrix} \begin{pmatrix} I_2 & I_2 \\ \Omega_2 & -\Omega_2 \end{pmatrix} \\ &= [(\mathbf{a}(0), \mathbf{a}(2)) \cdot F_2, (\mathbf{a}(1), \mathbf{a}(3)) \cdot F_2] \begin{pmatrix} I_2 & I_2 \\ \Omega_2 & -\Omega_2 \end{pmatrix}. \end{aligned} \tag{6.3.3}$$

This shows how a four-point DFT can be computed in terms of two two-point DFTs. As the next theorem shows, this holds in general; i.e., if $n = 2m$, then F_n can be split into two half-sized DFT matrices.

Theorem. *If $n = 2m$ and $\Omega_m = diag\left(1, \omega_n, \ldots, \omega_n^{m-1}\right)$, then*

$$\Pi_n' \cdot F_n = \begin{pmatrix} F_m & F_m \\ F_m \cdot \Omega_m & -F_m \cdot \Omega_m \end{pmatrix}.$$

Proof: Let

$$\Pi_n' \cdot F_n = \begin{pmatrix} A & B \\ B & C \end{pmatrix},$$

where *A*, *B*, *C*, and *D* are $m \times m$ matrices. Since $F_n \cdot \Pi_n$ is just F_n with its even-indexed columns grouped first and $\Pi_n' \cdot F_n = (F_n \cdot \Pi_n)'$, $\Pi_n' \cdot F_n$ is just F_n with its even-indexed rows grouped first. Thus, if $A = (a_{jk})$ with $0 \le j < m$ and $0 \le k < m$, then $a_{jk} = \omega_n^{2jk}$.

But since $\omega_n^2 = \omega_m$, $\omega_n^{2jk} = \omega_m^{jk}$. Therefore, $A = F_m$.

Similarly, if $B = (b_{jk})$, then $b_{jk} = \omega_n^{2j(k+m)}$ for $0 \le j < m$ and $0 \le k < m$. Since

$$\omega_n^{2j(k+m)} = \omega_m^{j(k+m)} = \omega_m^{jk+jm} = \omega_m^{jk} \cdot \omega_m^{jm} = \omega_m^{jk},$$

$B = F_m$.

Since Ω_m is a diagonal matrix, $F_m \cdot \Omega_m = \left(\omega_m^{jk} \cdot \omega_n^k\right)$, where $0 \le j < m$ and $0 \le k < m$. Now if $C = (c_{jk})$, then

$$c_{jk} = \omega_n^{(2j+1)k} = \omega_n^{2jk} \cdot \omega_n^k = \omega_m^{jk} \cdot \omega_n^k.$$

Therefore, $C = F_m \cdot \Omega_m$.

Finally, let $D = (d_{jk})$. Using the fact that $\omega_n^m = -1$, we obtain

$$d_{jk} = \omega_n^{(2j+1)(k+m)} = \omega_n^{2jk} \cdot \omega_n^k \cdot \omega_n^{2mj} \cdot \omega_n^m = -\omega_m^{jk} \cdot \omega_n^k ,$$

and, hence, $D = -F_m \cdot \Omega_m$.

Q.E.D.

It now follows that

$$\Pi_n' F_n = \begin{pmatrix} F_m & \mathrm{O} \\ \mathrm{O} & F_m \end{pmatrix} \begin{pmatrix} I_m & I_m \\ \Omega_m & -\Omega_m \end{pmatrix}. \tag{6.3.4}$$

If we multiply this equation by Π_n we obtain

$$F_n = \Pi_n \cdot \begin{pmatrix} F_m & \mathrm{O} \\ \mathrm{O} & F_m \end{pmatrix} \begin{pmatrix} I_m & I_m \\ \Omega_m & -\Omega_m \end{pmatrix}. \tag{6.3.5}$$

Hence, if $\mathbf{a} = (\mathbf{a}(0), \mathbf{a}(1), \ldots, \mathbf{a}(n-1))$, then

$$\mathbf{a} \cdot F_n = (\mathbf{a}(0), \mathbf{a}(2), \ldots, \mathbf{a}(n-2), \mathbf{a}(1), \mathbf{a}(3) \ldots, \mathbf{a}(n-1)) \begin{pmatrix} F_m & \mathrm{O} \\ \mathrm{O} & F_m \end{pmatrix} \begin{pmatrix} I_m & I_m \\ \Omega_m & -\Omega_m \end{pmatrix}. \tag{6.3.6}$$

This establishes the following corollary:

6.3.3 Corollary. *If $n = 2m$ and $\mathbf{a} \in \mathbb{C}^n$, then*

$$\mathbf{a} \cdot F_n = (\mathbf{a}_0 \cdot F_m, \mathbf{a}_1 \cdot F_m) \begin{pmatrix} I_m & I_m \\ \Omega_m & -\Omega_m \end{pmatrix}, \tag{6.3.7}$$

where $\mathbf{a}_0 = (\mathbf{a}(0), \mathbf{a}(2), \ldots, \mathbf{a}(n-2))$ and $\mathbf{a}_1 = (\mathbf{a}(1), \mathbf{a}(3), \ldots, \mathbf{a}(n-1))$.

The idea of splitting the DFT can be applied again if *m* is even, relating $\mathbf{a}_0 \cdot F_m$ to $\mathbf{a}_{00} \cdot F_{m/2}$ and $\mathbf{a}_{01} \cdot F_{m/2}$, and $\mathbf{a}_1 \cdot F_m$ to $\mathbf{a}_{10} \cdot F_{m/2}$ and $\mathbf{a}_{11} \cdot F_{m/2}$. In case *n* is a power of 2, we can divide and conquer all the way down to a sequence of one-point DFTs.

In case *n* is not a power of 2, then the radix-2 divide-and-conquer process breaks down. However, as long as *n* is not a prime number, it is still fairly easy to express F_n as a function of smaller DFT matrices. Specifically, if $n = pm$ with $1 < p < m$, then F_n can be expressed as a function of the DFT matrices F_m and F_p. The key is to use tensor notation. Note that

$$\begin{aligned} diag(I_m, \Omega_m)(F_2 \otimes I_m) &= \begin{pmatrix} I_m & \mathrm{O} \\ \mathrm{O} & I_m \end{pmatrix} \left[\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes I_m \right] \\ &= \begin{pmatrix} I_m & O \\ O & \Omega_m \end{pmatrix} \begin{pmatrix} I_m & I_m \\ I_m & -I_m \end{pmatrix} \\ &= \begin{pmatrix} I_m & I_m \\ \Omega_m & -\Omega_m \end{pmatrix}. \end{aligned}$$

Thus, Eq. 6.3.4 can be rewritten as

$$\Pi_n' F_n = (I_2 \otimes F_m) \cdot diag(I_m, \Omega_m) \cdot (F_2 \otimes I_m) \tag{6.3.8}$$

and Eq. 6.3.7 as

$$\mathbf{a} \cdot F_n = (\mathbf{a}_0 \cdot F_m, \mathbf{a}_1 \cdot F_m) \cdot B_n , \tag{6.3.9}$$

where $B_n = diag(I_m, \Omega_m) \cdot (F_2 \otimes I_m)$.

The general formulation of Eq. 6.3.8 is given by the next theorem.

6.3.4 **Theorem.** *If* $n = pm$ *and* $\Omega_{p,m} = diag\big(1, \omega_n, \ldots, \omega_n^{m-1}\big)$, *then*

$$\Pi'(p,n) \cdot F_n = (I_p \otimes F_m) \cdot diag\big(I_m, \Omega_{p,m}, \ldots, \Omega_{p,m}^{p-1}\big) \cdot (F_p \otimes I_m)\,. \tag{6.3.10}$$

Proof: Let $(A_{rs}) = \Pi'(p,n) \cdot F_n$ and

$$(B_{rs}) = (I_p \otimes F_m) \cdot diag\big(I_m, \Omega_{p,m}, \ldots, \Omega_{p,m}^{p-1}\big) \cdot (F_p \otimes I_m)\,,$$

where $0 \le r, s < p$ and each $A_{rs},\ B_{rs}$ is an $m \times m$ matrix such that (A_{rs}) and (B_{rs}) are $p \times p$ block matrices.

Let $[A_{rs}]_{jk}$ and $[B_{rs}]_{jk}$ denote the (j,k)th entry of A_{rs} and B_{rs}, respectively. we need to show that $[A_{rs}]_{jk} = [B_{rs}]_{jk}$.

By definition of B_{rs},

$$[B_{rs}]_{jk} = [F_m]_{jk} \cdot \left[\Omega_{p,m}^r\right]_{kk} \cdot \omega_p^{rs} = \omega_m^{jk} \cdot \omega_n^{rk} \cdot \omega_p^{rs}\,.$$

Since $\Pi'(p,n) \cdot F_n$ is just F_n with its rows sorted mod *p*, we also have

$$(A_{rs}) = (F_n(0), F_n(p), \ldots, F_n(n-p) \,|\, F_n(1), F_n(p+1), \ldots, F_n(n-p+1) \,|\, \ldots$$
$$\ldots \,|\, F_n(p-1), F_n(2p-1), \ldots, F_n(n-1))'\,,$$

where $F_n(j)$ denotes the *j*th row of F_n. Thus, if f_{uv} denotes the (u,v)th entry of F_n, then

$$[A_{rs}]_{jk} = f_{r+jp,sm+k} = \omega_n^{(r+jp)(sm+k)} = \omega_n^{pjk} \cdot \omega_n^{rk} \cdot \omega_n^{rsm} \cdot \omega_n^{mpjs}\,.$$

Since $\omega_n^{mpjs} = (\omega_n^{mp})^{js} = (\omega_n^n)^{js} = 1$, $\omega_n^{rsm} = \omega_p^{rs}$, and $\omega_n^{pjk} = \omega_m^{jk}$, we have

$$[A_{rs}]_{jk} = \omega_m^{jk} \cdot \omega_n^{rk} \cdot \omega_p^{rs}\,.$$

Q.E.D.

In analogy with the radix-2 splitting, radix-p splitting can be used to obtain an *n*–point DFT by gluing together *p* DFTs of length $m = n/p$.

6.3.5 **Corollary.** *If* $n = pm$ *and* $\mathbf{a} \in \mathbb{C}^n$, *then*

$$\mathbf{a} \cdot F_n = (\mathbf{a}_0 \cdot F_m, \mathbf{a}_1 \cdot F_m, \ldots, \mathbf{a}_{p-1} \cdot F_m) B_{p,n}\,,$$

where $\mathbf{a}_j = (\mathbf{a}(j), \mathbf{a}(p+j), \ldots, \mathbf{a}(n-p+j))$ *and*

$$B_{p,n} = diag\big(I_m, \Omega_{p,m}, \ldots, \Omega_{p,m}^{p-1}\big) \cdot (F_p \otimes I_m)\,.$$

Proof: Multiply both sides of Eq. 6.3.10 by $\mathbf{a} \cdot \Pi(p,n)$.

Q.E.D.

The matrix $B_{p,n}$ is known as the *radix-p butterfly matrix* and derives its name from the graphical representation of its action on an input vector (Section **6.5**, Fig. **6.5.1**). Note that for $p = 2$, $B_{p,n} = B_n$, where B_n is the radix-2 butterfly matrix defined by Eq. 6.3.9.

6.4 Radix-2 Factorization, Perfect Shuffles, and Bit Reversals

According to Theorem 6.3.4 and its corollary,

$$\mathbf{a} \cdot F_n = \mathbf{a} \cdot \Pi(p,n) \cdot (I_p \otimes F_m) \cdot B_{p,n} = (\mathbf{a}_0 \cdot F_m,\, \mathbf{a}_1 \cdot F_m,\, \ldots,\, \mathbf{a}_{p-1} \cdot F_m) \cdot B_{p,n}\,, \tag{6.4.1}$$

where $m = n/p$ and $1 < p < m$. Now if $m_1 = m$, $p_1 = p$, and there exist integers m_2 and p_2 such that $m_1 = p_2 m_2$ with $1 < p_2 < m_2$, then we can reapply Eq. 6.4.1 to each of the expressions $\mathbf{a}_j \cdot F_{m_1}$, replacing each DFT of length p_1 by DFTs of length p_2. This procedure can always be employed until for some index *k*, m_k cannot be factored further.

In order to better understand actual implementation of the split-and-merge process, we restrict our attention to the simple case where $n = 2^k$ and $p = 2$. The methodology derived from the analysis of this case can then be generalized to derive implementations of the general formulation expressed by Eq. 6.4.1 (e.g., [6, 10]). This approach is analogous to our derivation of Eq. 6.3.10 from Eq. 6.3.8.

For $p = 2$, Eq. 6.4.1 becomes

$$\mathbf{a} \cdot F_n = \mathbf{a} \cdot \Pi_n \cdot (I_2 \otimes F_m) \cdot B_n = (\mathbf{a}_0 \cdot F_m,\, \mathbf{a}_1 \cdot F_m) \cdot B_n\,. \tag{6.4.2}$$

Assuming $n = 2^k$, the split-and-merge procedure can be replaced by applying Corollary 6.3.3 to $\mathbf{a}_j \cdot F_m$ $(j = 0,\, 1)$ which splits each $\mathbf{a}_j \cdot F_m$ into

$$\mathbf{a}_j \cdot F_m = \left(\mathbf{a}_{j0} \cdot F_{m/2},\, \mathbf{a}_{j1} \cdot F_{m/2}\right) \cdot B_m\,. \tag{6.4.3}$$

Merging Eq. 6.4.3 with Eq. 6.4.2 results in

$$\mathbf{a} \cdot F_n = \left[\left(\mathbf{a}_{00} \cdot F_{m/2},\, \mathbf{a}_{01} \cdot F_{m/2}\right) \cdot B_m\,,\, \left(\mathbf{a}_{10} \cdot F_{m/2},\, \mathbf{a}_{11} \cdot F_{m/2}\right) \cdot B_m\right] \cdot B_n\,. \tag{6.4.4}$$

Using tensor notation, we may rewrite Eqs. 6.4.2 and 6.4.4 as

$$\mathbf{a} \cdot F_n = \left(\mathbf{a}_0 \cdot F_{n/2},\, \mathbf{a}_1 \cdot F_{n/2}\right) \cdot (I_1 \otimes B_n) \tag{6.4.5}$$

and

$$\mathbf{a} \cdot F_n = \left(\mathbf{a}_{00} \cdot F_{n/4},\, \mathbf{a}_{01} \cdot F_{n/4},\, \mathbf{a}_{10} \cdot F_{n/4},\, \mathbf{a}_{11} \cdot F_{n/4}\right) \cdot \left(I_2 \otimes B_{n/2}\right) \cdot (I_1 \otimes B_n)\,, \tag{6.4.6}$$

respectively.

Since $F_1 = (1)$, continuation of the split-and-merge process $k - 3$ more times leads to the following formulation of the DFT:

$$\mathbf{a} \cdot F_n = \left(\mathbf{a}_{j_0},\, \mathbf{a}_{j_1},\, \ldots,\, \mathbf{a}_{j_{n-1}}\right) \cdot \left(I_{n/2} \otimes B_2\right) \cdots \left(I_2 \otimes B_{n/2}\right) \cdot (I_1 \otimes B_n)\,, \tag{6.4.7}$$

where each of the subscripts j_r is some binary sequence of length *k*.

For $s = (s_0,\, s_1,\, \ldots,\, s_{k-1}) \in \prod_{i=0}^{k-1} \mathbb{Z}_2$, let $(s)_2$ denote the base 2 expansion of *s*; i.e.,

$$(s)_2 = s_0 + s_1 2 + \cdots + s_{k-1} 2^{k-1}\,.$$

Using induction on *k*, it can be shown that $\mathbf{a}_{j_r} = \mathbf{a}((j_r)_2)$ for $r = 0, 1, \ldots, n-1$ (e.g., Theorem **6.4.3**). This implies that

$$\left(\mathbf{a}_{j_0}, \mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_{n-1}}\right) = \left(\mathbf{a}(\rho_n(0)), \mathbf{a}(\rho_n(1)), \ldots, \mathbf{a}(\rho_n(n-1))\right), \tag{6.4.8}$$

where $\rho_n : \mathbb{Z}_n \to \mathbb{Z}_n$ is the permutation defined by $\rho_n(r) = (j_r)_2$.

Equivalently, we have

$$\left(\mathbf{a}_{j_0}, \mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_{n-1}}\right) = \mathbf{a} \cdot P_n, \tag{6.4.9}$$

where P_n is the permutation matrix defined by $P_n \equiv P_{\rho_n}$. This means that the DFT of $\mathbf{a} \in \mathbb{C}^n$ consists of a permutation of the components of **a** followed by a product of *k* sparse matrices.

Since P_n plays a central role in the efficient computation of the DFT, we provide a formal definition and an image algebra specification for its implementation. The following example will provide the necessary insight into understanding the action of P_n on the coordinates of a vector.

6.4.1 **Example:** Suppose $n = 2^4$ and $\mathbf{a} \in \mathbb{C}^{16}$. Then

$$\mathbf{a} \cdot F_{16} = (\mathbf{a}_0 \cdot F_8, \mathbf{a}_1 \cdot F_8) \cdot (I_1 \otimes B_{16}),$$

where

$$\mathbf{a}_0 = (\mathbf{a}(0), \mathbf{a}(2), \mathbf{a}(4), \mathbf{a}(6), \mathbf{a}(8), \mathbf{a}(10), \mathbf{a}(12), \mathbf{a}(14))$$

and

$$\mathbf{a}_1 = (\mathbf{a}(1), \mathbf{a}(3), \mathbf{a}(5), \mathbf{a}(7), \mathbf{a}(9), \mathbf{a}(11), \mathbf{a}(13), \mathbf{a}(15)).$$

Using the butterfly matrix B_8, we can synthesize each 8–point DFT, $\mathbf{a}_j \cdot F_8$, from a pair of 4–point DFTs as follows:

$$(\mathbf{a}_0 \cdot F_8, \mathbf{a}_1 \cdot F_8) = (\mathbf{a}_{00} \cdot F_4, \mathbf{a}_{01} \cdot F_4, \mathbf{a}_{10} \cdot F_4, \mathbf{a}_{11} \cdot F_4) \cdot (I_2 \otimes B_8),$$

where

$$\mathbf{a}_{00} = (\mathbf{a}(0), \mathbf{a}(4), \mathbf{a}(8), \mathbf{a}(12)), \quad \mathbf{a}_{01} = (\mathbf{a}(2), \mathbf{a}(6), \mathbf{a}(10), \mathbf{a}(14)),$$
$$\mathbf{a}_{10} = (\mathbf{a}(1), \mathbf{a}(5), \mathbf{a}(9), \mathbf{a}(13)), \text{ and } \mathbf{a}_{11} = (\mathbf{a}(3), \mathbf{a}(7), \mathbf{a}(11), \mathbf{a}(15)).$$

Repeating this process, we obtain

$$(\mathbf{a}_{00} \cdot F_4, \mathbf{a}_{01} \cdot F_4, \mathbf{a}_{10} \cdot F_4, \mathbf{a}_{11} \cdot F_4)$$
$$= (\mathbf{a}_{000} \cdot F_2, \mathbf{a}_{001} \cdot F_2, \mathbf{a}_{010} \cdot F_2, \mathbf{a}_{011} \cdot F_2, \mathbf{a}_{100} \cdot F_2, \mathbf{a}_{101} \cdot F_2, \mathbf{a}_{110} \cdot F_2, \mathbf{a}_{111} \cdot F_2) \cdot (I_4 \otimes B_4),$$

where

$$\mathbf{a}_{000} = (\mathbf{a}(0), \mathbf{a}(8)), \quad \mathbf{a}_{001} = (\mathbf{a}(4), \mathbf{a}(12)), \quad \mathbf{a}_{010} = (\mathbf{a}(2), \mathbf{a}(10)),$$
$$\mathbf{a}_{011} = (\mathbf{a}(6), \mathbf{a}(14)), \quad \mathbf{a}_{100} = (\mathbf{a}(1), \mathbf{a}(9)), \quad \mathbf{a}_{101} = (\mathbf{a}(5), \mathbf{a}(13)),$$
$$\mathbf{a}_{110} = (\mathbf{a}(3), \mathbf{a}(11)), \text{ and } \mathbf{a}_{111} = (\mathbf{a}(7), \mathbf{a}(15)).$$

The final stage is now given by

$$\begin{aligned}
&(\mathbf{a}_{000}\cdot F_2,\, \mathbf{a}_{001}\cdot F_2,\, \mathbf{a}_{010}\cdot F_2,\, \mathbf{a}_{011}\cdot F_2,\, \mathbf{a}_{100}\cdot F_2,\, \mathbf{a}_{101}\cdot F_2,\, \mathbf{a}_{110}\cdot F_2,\, \mathbf{a}_{111}\cdot F_2)\\
&= (\mathbf{a}_{0000},\, \mathbf{a}_{0001},\, \mathbf{a}_{0010},\, \mathbf{a}_{0011},\, \mathbf{a}_{0100},\, \mathbf{a}_{0101},\, \mathbf{a}_{0110},\, \mathbf{a}_{0111},\, \mathbf{a}_{1000}, \mathbf{a}_{1001},\, \mathbf{a}_{1010},\, \mathbf{a}_{1011},\\
&\qquad \mathbf{a}_{1100},\, \mathbf{a}_{1101},\, \mathbf{a}_{1110},\, \mathbf{a}_{1111})\cdot(I_8\otimes B_2)\\
&= (\mathbf{a}(0),\, \mathbf{a}(8),\, \mathbf{a}(4),\, \mathbf{a}(12),\, \mathbf{a}(2),\, \mathbf{a}(10),\, \mathbf{a}(6),\, \mathbf{a}(14),\, \mathbf{a}(1),\, \mathbf{a}(9),\, \mathbf{a}(5),\, \mathbf{a}(13),\\
&\qquad \mathbf{a}(3),\, \mathbf{a}(11),\, \mathbf{a}(7),\, \mathbf{a}(15))\cdot(I_8\otimes B_2)\,.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbf{a}\cdot P_{16} &= (\mathbf{a}(0),\, \mathbf{a}(8),\, \mathbf{a}(4),\, \mathbf{a}(12),\, \mathbf{a}(2),\, \mathbf{a}(10),\, \mathbf{a}(6),\, \mathbf{a}(14),\, \mathbf{a}(1), \mathbf{a}(9),\, \mathbf{a}(5),\, \mathbf{a}(13),\\
&\qquad \mathbf{a}(3),\, \mathbf{a}(11),\, \mathbf{a}(7),\, \mathbf{a}(15))\\
&= (\mathbf{a}(\rho_{16}(0)),\, \mathbf{a}(\rho_{16}(1)),\, \mathbf{a}(\rho_{16}(2)),\, \ldots,\, \mathbf{a}(\rho_{16}(13)),\, \mathbf{a}(\rho_{16}(14)),\, \mathbf{a}(\rho_{16}(15)))\,.
\end{aligned}$$

Expressing *i* and $\rho_{16}(i)$ in terms of binary notation elucidates the action of P_n. Specifically, we have the following relationship:

i	$\rho_{16}(i)$	binary i	binary $\rho_{16}(i)$
0	0	0000	0000
1	8	0001	1000
2	4	0010	0100
3	12	0011	1100
4	2	0100	0010
5	10	0101	1010
6	6	0110	0110
7	14	0111	1110
8	1	1000	0001
9	9	1001	1001
10	5	1010	0101
11	13	1011	1101
12	3	1100	0011
13	11	1101	1011
14	7	1110	0111
15	15	1111	1111

Figure 6.4.1 The permutation ρ_{16} and its corresponding binary evaluation

It follows from the table that $\rho_{16}(i)$ is obtained from *i* by reversing the order (reading from right-to-left) of the four bits in the binary expression of *i*.

Because of its action on the indices of a vector, the permutation matrix P_n is commonly known as the *bit-reversing* permutation matrix.

For the formal specification of P_n, let $n = 2^k$ and for $j = 1, 2, \ldots, k$ set

$$Q_j = I_{2^{k-j}} \otimes \Pi_{2^j} .$$

Define

$$P_n = Q_k \cdot Q_{k-1} \cdots Q_2 \cdot Q_1 . \tag{6.4.10}$$

Thus, P_n is composed of copies of the perfect shuffle permutation matrices $\Pi_2, \Pi_4, \ldots, \Pi_n$ which, according to Eq. 6.2.5, makes P_n a permutation matrix.

Note that if $k = 1$, then

$$P_2 = Q_1 = I_1 \otimes \Pi_2 = \Pi_2 ,$$

while for $k = 2$,

$$P_4 = Q_2 \cdot Q_1 = (I_1 \otimes \Pi_4) \cdot (I_2 \otimes \Pi_2) = \Pi_4 \cdot (I_2 \otimes P_2) .$$

This suggests the following theorem:

6.4.2 Theorem. *If $n = 2^k$ and $m = n/2$, then*

$$P_n = \Pi_n \cdot (I_2 \otimes P_m) .$$

Proof: Let $h = k - 1$. Then $m = 2^h$ and

$$P_m = \tilde{Q}_h \cdot \tilde{Q}_{h-1} \cdots \tilde{Q}_2 \cdot \tilde{Q}_1 ,$$

where

$$\tilde{Q}_j = I_{2^{h-j}} \otimes \Pi_{2^j} .$$

Using Eq. 6.2.1, we obtain

$$\begin{aligned} I_2 \otimes \tilde{Q}_j &= I_2 \otimes (I_{2^{h-j}} \otimes \Pi_{2^j}) \\ &= I_2 \otimes (I_{2^{k-1-j}} \otimes \Pi_{2^j}) \\ &= I_{2^{k-j}} \otimes \Pi_{2^j} \\ &= Q_j . \end{aligned}$$

Hence, using Eq. 6.2.2, we can factor $I_2 \otimes P_m$ as

$$\begin{aligned} I_2 \otimes P_m &= I_2 \otimes \tilde{Q}_h \cdot \tilde{Q}_{h-1} \cdots \tilde{Q}_2 \cdot \tilde{Q}_1 \\ &= \left(I_2 \otimes \tilde{Q}_h\right) \cdots \left(I_2 \otimes \tilde{Q}_2\right) \cdot \left(I_2 \otimes \tilde{Q}_1\right) \\ &= Q_{k-1} \cdots Q_2 \cdot Q_1 . \end{aligned}$$

Since $Q_k = I_{2^0} \otimes \Pi_{2^k} = \Pi_n$, we now have

$$\Pi_n \cdot (I_2 \otimes P_m) = Q_k \cdot Q_{k-1} \cdots Q_2 \cdot Q_1 .$$

Q.E.D.

The recursive characterization of P_n given by this theorem makes it easy to prove the bit-reversing action of P_n.

6.4.3 **Theorem.** *If $n = 2^k$ and P_n is the permutation matrix defined by Eq. 6.4.10, then the permutation ρ_n defined by*

$$\rho_n(i) = j \;\Leftrightarrow\; [P_n]_{ij} = 1$$

has the property that

$$\rho_n((i_0, i_1, \ldots, i_{k-1})_2) = (i_{k-1}, i_{k-2}, \ldots, i_0)_2 .$$

Proof: If $k = 1$, then $P_2 = I_2$ and, therefore, $\rho_n((i_0)_2) = (i_0)_2$.

We now proceed by using induction, assuming the result to hold for all integers less than $k - 1$, where $k \geq 2$.

Let $n = 2^k,\ \mathbf{a} \in \mathbb{C}^n,\ m = 2^{k-1}$, and $\mathbf{b} = \mathbf{a} \cdot P_n$. We must show that

$$\mathbf{b}((i_{k-1}, i_{k-2}, \ldots, i_0)_2) = \mathbf{a}((i_0, i_1, \ldots, i_{k-1})_2) .$$

By Theorem **6.4.2**,

$$\mathbf{b} = \mathbf{a} \cdot P_n = \mathbf{a} \cdot \Pi_n \cdot (I_2 \otimes P_m) = (\mathbf{a}_0, \mathbf{a}_1) \cdot (I_2 \otimes P_m) = (\mathbf{a}_0 \cdot P_m, \mathbf{a}_1 \cdot P_m) ,$$

where $\mathbf{a}_0(i) = \mathbf{a}(2i)$ and $\mathbf{a}_1(i) = \mathbf{a}(2i + 1)$ for $i = 0, 1, \ldots, m - 1$.

Let $i = (i_0, i_1, \ldots, i_{k-1})_2$. Suppose *i* is even. Then Π_n maps $\mathbf{a}(i)$ to $\mathbf{a}_0(i/2)$. Since *i* is even, $i_0 = 0$. Hence, $i/2 = i_1 + i_2 2 + \cdots i_{k-1} 2^{k-2}$ and $\mathbf{a}(i)$ maps to $\mathbf{a}_0((i_1, i_2, \ldots, i_{k-1})_2)$. By induction hypothesis,

$$\rho_m((i_1, i_2, \ldots, i_{k-1})_2) = (i_{k-1}, i_{k-2}, \ldots, i_1)_2 .$$

But since $i_0 = 0$, $(i_{k-1}, i_{k-2}, \ldots, i_1)_2 = (i_{k-1}, i_{k-2}, \ldots, i_1, i_0)_2$. Therefore,

$$\mathbf{b}((i_{k-1}, i_{k-2}, \ldots, i_0)_2) = \mathbf{a}((i_0, i_1, \ldots, i_{k-1})_2) .$$

If *i* is odd, then $i_0 = 1$ and Π_n maps $\mathbf{a}(i)$ to $\mathbf{a}_1\left(\frac{i-1}{2}\right)$, where

$$(i - 1)/2 = \left(1 + i_1 2 + \cdots i_{k-1} 2^{k-1} - 1\right)/2 = (i_1, i_2, \ldots, i_{k-1})_2 .$$

By induction hypothesis we again have

$$\rho_m((i_1, i_2, \ldots, i_{k-1})_2) = (i_{k-1}, i_{k-2}, \ldots, i_1)_2 .$$

However, the coordinate position $(i_{k-1}, i_{k-2}, \ldots, i_1)_2$ in the vector $\mathbf{a}_1 \cdot P_m$ corresponds to the coordinate position $(i_{k-1}, i_{k-2}, \ldots, i_1)_2 + m$ in the vector $\mathbf{b} = (\mathbf{a}_0 \cdot P_m, \mathbf{a}_1 \cdot P_m)$. Thus, since $i_0 = 1$,

$$\begin{aligned}(i_{k-1}, i_{k-2}, \ldots, i_1)_2 + m &= \left(i_{k-1} + i_{k-2} 2 + \cdots + i_1 2^{k-2}\right) + i_0 2^{k-1} \\ &= (i_{k-1}, i_{k-2}, \ldots, i_1, i_0)_2 .\end{aligned}$$

Therefore,

$$\mathbf{b}((i_{k-1}, i_{k-2}, \ldots, i_0)_2) = \mathbf{a}((i_0, i_1, \ldots, i_{k-1})_2) .$$

Q.E.D.

Reversing an index bit twice results in the original index. Thus, $\mathbf{a} \cdot P_n \cdot P_n = \mathbf{a}$ and, hence, $P_n \cdot P_n = I_n$. Therefore, $P_n = P_n^{-1} = P_n'$. This proves the following corollary.

6.4.4 Corollary. *P_n is symmetric.*

Having examined the structure of P_n, we turn our attention to the computation of $\mathbf{a} \cdot P_n$.

The following algorithm computes the permutation function ρ_n.

6.4.5 Algorithm (Evaluating $\rho_n(i)$)

For $n = 2^k$ and $i = 0, 1, \ldots, n-1$ define $\rho_n(i)$ as:

begin

$j := 0$
$m := i$
for l **:=** `0` **to** $log_2(n) - 1$ **loop**
$h := \lfloor \frac{m}{2} \rfloor$
$j := 2j + (m - 2h)$
$m := h$
end loop
return j

end

For $\mathbf{a} \in \mathbb{C}^n$, the computation of $\mathbf{a} \cdot P_n$ can now be accomplished by using the simple image algebra specification:

$$\mathbf{b} := \mathbf{a} \circ \rho_n$$

Of course, by specifying the template $\mathbf{p} = \psi^{-1}(P_n)$, we could just as well compute $\mathbf{a} \cdot P_n$ in terms of the convolution product $\mathbf{a} \oplus \mathbf{p}$.

In view of Eq. 6.4.7, the DFT of $\mathbf{a} \in \mathbb{C}^n$ can be obtained by first computing $\mathbf{a} \cdot P_n$ followed by a sequence of *k* computations of type $\mathbf{v} = \mathbf{w} \cdot (I_j \otimes B_m)$, where each column of $I_j \otimes B_m$ has exactly two nonzero entries. Since each evaluation of $\rho_n(\cdot)$ involves $O(log_2 n)$ integer operations, Algorithm **6.4.5** requires $O(n \cdot log_2 n)$ integer operations. Thus, the amount of integer arithmetic involved in computing $\mathbf{a} \circ \rho_n$ is of the same order of magnitude as the amount of floating point arithmetic in computing $\mathbf{w} \cdot (I_j \otimes B_m)$. Hence the overhead associated with bit reversal is nontrivial with respect to the computation of $\mathbf{a} \cdot F_n$, often amounting for 10% to 30% of the total computation time.

6.5 The Fast Fourier Transform

The vector-matrix product $\mathbf{a} \cdot F_n$ involves $O(n^2)$ complex computations. The *Fast Fourier Transform*, or FFT, is a method that drastically reduces the number of computations when forming the product $\mathbf{a} \cdot F_n$.

Using the results of the previous section, we can write $\mathbf{a} \cdot F_n$ as

$$\mathbf{a} \cdot F_n = \mathbf{a} \cdot P_n \cdot A_1 \cdot A_2 \cdots A_k \,, \tag{6.5.1}$$

where $A_i = I_j \otimes B_m$, $n = 2^k$, $i = 1, 2, \ldots, k$, $m = 2^i$, and $j = n/m$. Thus, we may use the following algorithm for computing $\mathbf{a} \cdot F_n$:

6.5.1 Algorithm (Radix-2 FFT)

$\mathbf{a} := \mathbf{a} \circ \rho_n$ (Algorithm **6.4.5**)
for $i := 1$ **to** k **loop**
    $\mathbf{a} := \mathbf{a} \cdot A_i$
end loop

Since each A_i has only two nonzero entries per column and each complex add involves two floating point operations, or *flops*, while each complex multiply involves six flops, it is not difficult to ascertain that this algorithm requires only $O(n \cdot log_2 n)$ flops if the sparsity of the A_is is exploited. This is the primary reason why sparse factorizations of F_n form the mathematical framework for the efficient computation of the DFT. These algorithms are collectively called *Fast Fourier Transforms*. Algorithm **6.5.1** represents a high-level version of the *Cooley-Tukey radix-2* FFT algorithm [1, 4, 3].

The FFT ranks as one of the great computational developments of this Century. Although Cooley and Tukey are, deservedly, given credit for the modern development of the FFT as represented by Algorithm **6.5.1**, Carl Friedrich Gauss had developed identical as well as more general methods for the efficient computation of the Fourier transform more than 150 years earlier [7]. Consequently, much time and effort could have been saved during the 1950s and '60s had researchers been familiar with Gauss's work. In addition, the general computational framework for the FFT could have been established much earlier.

Different FFT algorithms correspond to different factorizations of F_n and, hence, to different factorizations of the Fourier template $\mathbf{f}$. While template operations provide a highly structured environment for studying image transform theory, template factorizations provide a general framework for studying key aspects of advanced image transform computation; e.g., vectorization, localization, and parallelization. Some of these issues have already been discussed in previous sections while others will be discussed in more detail in subsequent sections. In this section we focus our attention on the image algebra specification of Algorithm **6.5.1**.

The image algebra equivalent of Eq. 6.5.1 is given by

$$\mathbf{a} \oplus \mathbf{f} = (\mathbf{a} \circ \rho_n) \oplus \mathbf{t}(1) \oplus \mathbf{t}(2) \oplus \cdots \oplus \mathbf{t}(k)\,, \tag{6.5.2}$$

where $\mathbf{t}(i) = \psi^{-1}(A_i)$. Here we use the notation $\mathbf{t}(i)$ instead of $\mathbf{t}_i$ to denote a sequence $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k$ of templates. As will be shown subsequently, the sequence $\mathbf{t}(1), \mathbf{t}(2), \ldots, \mathbf{t}(k)$ can be represented by a single parametrized template $\mathbf{t}(p)$, where *p* is of form 2^i. Thus, the notation $\mathbf{t}(i)$ will accustom us to the idea that $\mathbf{t}(i)$ is a parametrized template.

To compute Eq. 6.5.2, we may use the following algorithm which looks very much like Algorithm **6.5.1**.

6.5.2 Algorithm

$\mathbf{a} := \mathbf{a} \circ \rho_n$ (Algorithm **6.4.5**)
for $i := 1$ **to** k **loop**
    $\mathbf{a} := \mathbf{a} \oplus \mathbf{t}(i)$
end loop

To complete the specification of this algorithm, we need to examine the parameters defining the templates $\mathbf{t}(i) = \psi^{-1}(A_i)$.

Suppose $n = 2^k$ and $k = 3$. In this case we need to specify $\mathbf{t}(1)$, $\mathbf{t}(2)$, and $\mathbf{t}(3)$ corresponding to the matrices A_1, A_2, and A_3, respectively. Using the tensor representation of these matrices, we obtain

$$A_1 = I_4 \otimes B_2 = I_4 \otimes \begin{pmatrix} I_1 & I_1 \\ \Omega_1 & -\Omega_1 \end{pmatrix} = I_4 \otimes \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix},$$

$$A_2 = I_2 \otimes B_4 = I_2 \otimes \begin{pmatrix} I_2 & I_2 \\ \Omega_2 & -\Omega_2 \end{pmatrix} = I_2 \otimes \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & \omega_4 & 0 & -\omega_4 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \omega_4 & 0 & -\omega_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_4 & 0 & -\omega_4 \end{pmatrix},$$

and

$$A_3 = I_1 \otimes B_8 = I_1 \otimes \begin{pmatrix} I_4 & I_4 \\ \Omega_4 & -\Omega_4 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & \omega_8 & 0 & 0 & 0 & -\omega_8 & 0 & 0 \\ 0 & 0 & \omega_8^2 & 0 & 0 & 0 & -\omega_8^2 & 0 \\ 0 & 0 & 0 & \omega_8^3 & 0 & 0 & 0 & -\omega_8^3 \end{pmatrix}.$$

At the *i*th stage of the loop in Algorithm **6.5.2** we compute the product $\mathbf{b}_i = \mathbf{b}_{i-1} \bigoplus \mathbf{t}(i)$, where $\mathbf{b}_0 = \mathbf{a} \circ \rho_8$. By definition of the product operator $\bigoplus$, the value $\mathbf{b}_i(j)$ is given by

$$\mathbf{b}_i(j) = \sum_{l \in S(\mathbf{t}(i)_j)} \mathbf{b}_{i-1}(l) \cdot \mathbf{t}(i)_j(l)\,. \tag{6.5.3}$$

The cardinality of $S\left(\mathbf{t}(i)_j\right)$ is two since the template image $\mathbf{t}(i)_j$ corresponds to the *j*th column of A_i. Thus, using Eq. 6.5.3, the computation of each value $\mathbf{b}_i(j)$ involves two multiplications and one addition which shows that the image algebra formulation $\mathbf{a} := \mathbf{a} \bigoplus \mathbf{t}(i)$ exploits the sparsity of A_i.

To complete the specification of the template $\mathbf{t}(i)$, we need to define the weights $\mathbf{t}(i)_j(l)$. Let $\mathbf{b}_i = \mathbf{b}_{i-1} \bigoplus \mathbf{t}(i)$, where $\mathbf{b}_0 = \mathbf{a} \circ \rho_8$. By carefully considering the columns of the matrices A_i, it is not difficult to see that $\mathbf{b}_i(j)$ and $\mathbf{b}_i\left(j + 2^{i-1}\right)$ can be computed in terms of the following two-dimensional vector product

$$\left(\mathbf{b}_i(j),\, \mathbf{b}_i\left(j + 2^{i-1}\right)\right) = \left(\mathbf{b}_{i-1}(j),\, \mathbf{b}_{i-1}\left(j + 2^{i-1}\right)\right) \cdot \begin{pmatrix} 1 & 1 \\ w & -w \end{pmatrix}, \tag{6.5.4}$$

where $w = \omega_{2^i}^{j mod 2^{i-1}}$. Equation 6.5.4 can be represented graphically as illustrated in Figure **6.5.1**.

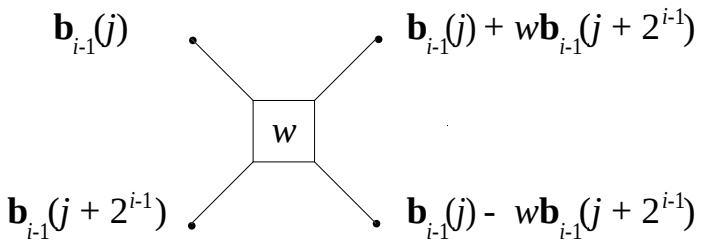


Figure 6.5.1 Graphical representation of the butterfly operation.

With a little imagination, the term "butterfly" becomes evident when viewing this schematic representation of Eq. 6.5.4. The complete set of butterfly computations in the eight–point Cooley-Tukey FFT is shown in Fig. **6.5.2**. Here we exploited the symmetries of $\omega_{2^i}^{j mod 2^{i-1}}$ in order to express each butterfly weight *w* in terms of ω_8.

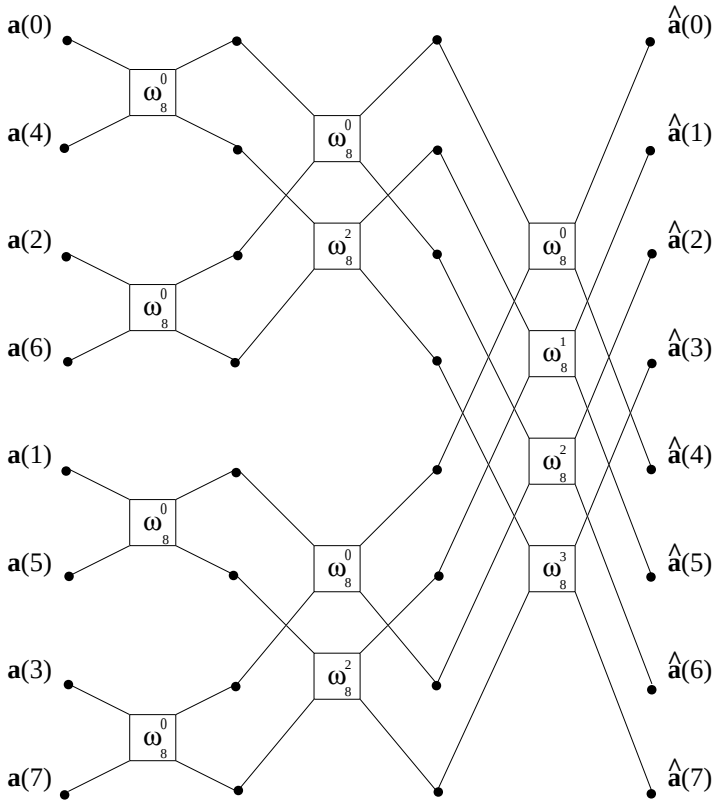


Figure 6.5.2 The Cooley-Tukey data flow graph for $n = 8$.

Equation 6.5.4 is the key for representing the templates $\mathbf{t}(1)$, $\mathbf{t}(2)$, and $\mathbf{t}(3)$ in terms of a single parametrized template $\mathbf{t}(p)$. Specifically, let $P = \{2^i : i = 0, 1, 2\}$ and for $p \in P$, define $\mathbf{t}(p)$ by

$$\mathbf{t}(p)_j(l) = \begin{cases} 1 & \text{if } \lfloor j/p \rfloor \text{ is even and } l = j \\ w(j,p) & \text{if } \lfloor j/p \rfloor \text{ is even and } l = j + p \\ -w(j,p) & \text{if } \lfloor j/p \rfloor \text{ is odd and } l = j \\ 1 & \text{if } \lfloor j/p \rfloor \text{ is odd and } l = j - p \\ 0 & \text{otherwise}, \end{cases} \tag{6.5.5}$$

where $w(j,p) = \omega_{2p}^{jmodp}$. Using the convention $w(j,1) = 0 \ \forall j$ and simple inspection verifies that for the parameters $p = 2^0$, 2^1, and 2^2, the templates generated by $\mathbf{t}(p)$ correspond to the templates $\psi^{-1}(A_1)$, $\psi^{-1}(A_2)$, and $\psi^{-1}(A_3)$, respectively.

The eight-point radix-2 FFT can easily be generalized. In the general case we have $A_i = I_j \otimes B_m$, where $m = 2^i$ and $j = n/m = 2^{k-i}$. Thus, A_i is the $2^{k-i} \times 2^{k-i}$ block matrix with blocks of size $m \times m$ given by

$$A_i = \begin{pmatrix} B_m & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & B_m & \cdots & \mathrm{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \mathrm{O} & \cdots & B_m \end{pmatrix},$$

where

$$B_m = \begin{pmatrix} I_{m/2} & I_{m/2} \\ \Omega_{m/2} & -\Omega_{m/2} \end{pmatrix}$$

$$= \left(\begin{array}{cccc|c|cccc} 1 & 0 & \cdots & 0 & & 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & & 0 & 0 & \cdots & 1 \\ - & - & - & - & + & - & - & - & - \\ 1 & 0 & \cdots & 0 & & -1 & 0 & \cdots & 0 \\ 0 & \omega_m & \cdots & 0 & & 0 & -\omega_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_m^{\frac{m}{2}-1} & & 0 & 0 & \cdots & -\omega_m^{\frac{m}{2}-1} \end{array}\right).$$

Since the number of zeros between the two nonzero entries in each column of B_m is $2^{i-1} - 1$, the spacing of the two nonzero entries in each column of A_i is exactly 2^{i-1}. But the nonzero entries in the *j*th column of A_i corresponds to the nonzero weights of the template image $\left[\psi^{-1}(A_i)\right]_j$. Thus, Eq. 6.5.5 also defines $\mathbf{t}(p)$ for $p \in P = \left\{2^i : i = 0, 1, \ldots, k-1\right\}$.

With the specification of $\mathbf{t}(p)$ completed, we are now able to formulate the Cooley-Tukey radix-2 FFT using the language of image algebra.

6.5.3 **Algorithm** (Radix-2 FFT). If $\mathbf{a} \in \mathbb{C}^n$, $n = 2^k$, and $\mathbf{t}(p)$ is specified by Eq. 6.5.5, then the following algorithm computes $\mathbf{a} \bigoplus \mathbf{f}$.

$\mathbf{a} := \mathbf{a} \circ \rho_n$
for $i := 1$ **to** $log_2 n$ **loop**
    $\mathbf{a} := \mathbf{a} \bigoplus \mathbf{t}\left(2^{i-1}\right)$
end loop

The appealing aspect of the image algebra version of the FFT is that even though it completely specifies the computation of the FFT, it retains the same high level appearance as Algorithm **6.5.1**.

Although we are only interested in *general* algebraic and computational frameworks, we need to remind the reader that many important issues arise when trying to obtain highly optimized versions of Algorithm **6.5.3** for *specific* implementations. We briefly discuss two of these issues.

Note that as *i* increases, the spacings between the nonzero values of $\mathbf{t}\left(2^{i-1}\right)$ increases by a factor of 2^{i-1}. This spacing is called the *stride* of the butterfly operation at the *i*th stage in the loop of Algorithm **6.5.3**. More generally, *strides* refer to the spacings of vector components that are named in a vector reference. In many advanced computer architectures, large power-of-two strides can severely degrade performance. Machines with interleaved memories serve as good examples of this problem. The stride issue can often be resolved by reordering the algorithm in order to achieve unit stride. However, this reordering typically results in algorithms having a high level of redundant arithmetic. This represents the typical dilemma of many current high performance computing environments: one

procedure is arithmetically efficient but has nonunit stride, thus severely degrading performance due to memory accessing, while the alternative has attractive stride properties but an excess of arithmetic [2, 8, 9].

Languages as well as computer architectures pose problems for achieving optimized FFTs. Some languages such as FORTRAN store complex vectors in stride-2 fashion. For example, an *n*-dimensional complex vector $\mathbf{u}+i\mathbf{v}$ is stored as an array of length 2*n* of real numbers $(u_0,\ v_0,\ u_1,\ v_1,\ \ldots,\ u_{n-1},\ v_{n-1})$. Thus, the extraction of either the real or imaginary vector components involves stride-2 access. Algorithms involving complex vectors inherit, therefore, all the problems associated with this access. In addition, an algorithm describing the butterfly operation (Eq. 6.5.4) that needs to explicitly reference the real and imaginary parts using stride-2 access will obviously destroy the simplicity of Eq. 6.5.4 and obscure the key algorithmic point.

A second issue concerns on-line versus off-line computations. We may either precompute the template weights $w(j,p)$ once and for all so that at execution time they may be recalled by simple look-up. This is the *off-line paradigm*. However, the off-line paradigm requires vector workspace. The vector workspace requirement can be greatly reduced since the weight s associated with $\mathbf{t}\left(2^{i-1}\right)$ are a subset of the weights associated with $\mathbf{t}\left(2^{i}\right)$. This follows from the observation that $\omega_{p_2}^{2j} = \omega_{p_1}^{j}$, where $p_2 = 2^i$ and $p_1 = 2^{i-1}$.

If not sufficient workspace is available, it may be necessary to use the *on-line paradigm* which assumes that the weights for $\mathbf{t}\left(2^{i-1}\right)$ are generated by direct call during the *i*th stage of the loop. Of course, on-line computation will increase the total computation time of Algorithm **6.5.3**.

We conclude this section with the image algebra version of the two-dimensional FFT and its inverse.

In our observation following Eq. 6.1.22 (Section **6.1**) we noted that the two-dimensional DFT can be computed in two steps by successive applications of the one-dimensional DFT; first along each row followed by a one-dimensional DFT along each column. Thus, to obtain a fast Fourier transform for two-dimensional images, we need to apply Algorithm **6.5.3** in simple succession. However, in order to perform the operations $\mathbf{a}\circ\rho_n$ and $\mathbf{a}\oplus\mathbf{t}(p)$ specified by the algorithm, it becomes necessary to extend the functions ρ_n and $\mathbf{t}(p)$ to two-dimensional arrays. For this purpose, suppose that $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$, where $m = 2^h$ and $n = 2^k$, and assume without loss of generality that $n \leq m$.

Let $P = \left\{2^i : i = 0,\ 1,\ \ldots,\ log_2 m - 1\right\}$ and for $p \in P$ define the parameterized row template $\mathbf{t}(p) : \mathbf{X} \to \mathbb{C}^{\mathbf{X}}$ by

$$\mathbf{t}(p)_{(u,v)}(x,y) = \begin{cases} 1 & \text{if } \lfloor u/p \rfloor \text{ is even and } (x,y) = (u,v) \\ w(u,p) & \text{if } \lfloor u/p \rfloor \text{ is even and } (x,y) = (u+p,v) \\ -w(u,p) & \text{if } \lfloor u/p \rfloor \text{ is odd and } (x,y) = (u,v) \\ 1 & \text{if } \lfloor u/p \rfloor \text{ is odd and } (x,y) = (u-p,v) \\ 0 & \text{otherwise}, \end{cases} \tag{6.5.6}$$

Note that for each $p \in P$, $\mathbf{t}(p)$ is a row template which is essentially identical to the template defined by Eq. 6.5.5.

The permutation ρ is extended to a function $\mathbf{X} \to \mathbf{X}$ in a similar fashion by restricting its actions to the rows of $\mathbf{X}$. In particular, define

$$r_m : \mathbf{X} \to \mathbf{X}$$
$$\text{by } r_m(i,j) = (\rho_m(i), j)\,.$$

With the definitions of *r* and **t** completed, we are now in a position to specify the two-dimensional radix-2 FFT in terms of image algebra notation.

6.5.4 **Algorithm** (2–D FFT). If $\mathbf{X}$, r_m and $\mathbf{t}$ are specified as above and $\mathbf{a} \in \mathbb{C}^{\mathbf{X}}$, then the following algorithm computes the two-dimensional Fourier transform $\mathbf{a} \bigoplus \mathbf{f}$.

$\mathbf{a} := \mathbf{a} \circ r_m$
for $i := 1$ **to** $log_2 m$ **loop**
$\quad \mathbf{a} := \mathbf{a} \bigoplus \mathbf{t}\left(2^{i-1}\right)$
end loop
$\mathbf{a} := \mathbf{a}' \circ r_n$
for $i := 1$ **to** $log_2 n$ **loop**
$\quad \mathbf{a} := \mathbf{a} \bigoplus \mathbf{t}\left(2^{i-1}\right)$
end loop
$\hat{\mathbf{a}} := \mathbf{a}'$

Following Eq. 6.1.23 (Section **6.1**) we also observed that the inverse Fourier transform can be computed in terms of the Fourier transform by simple conjugation. The next algorithm computes the inverse FFT, $\frac{1}{mn}(\hat{\mathbf{a}} \bigoplus \mathbf{f}^*)$, of $\hat{\mathbf{a}} \in \mathbb{C}^{\mathbf{X}}$.

6.5.5 **Algorithm** (2–D inverse FFT).

$\mathbf{a} := \hat{\mathbf{a}}^*$
`Algorithm 6.5.4`
$\mathbf{a} := \left(\frac{1}{mn}\mathbf{a}\right)^*$

When using parallel machines, it may be advantageous to implement the forward product in order to generate the data that is broadcast to other processors [5, 11]. In this case, the statement $\mathbf{a} := \mathbf{a} \bigoplus \mathbf{t}\left(2^{i-1}\right)$ in the preceding algorithms is replaced by $\mathbf{a} := \mathbf{t}'\left(2^{i-1}\right) \bigoplus \mathbf{a}$.

6.6 Radix-4 Factorization

According to Theorem 6.3.4 and its corollary,

$$\mathbf{a} \cdot F_n = \mathbf{a} \cdot \Pi(p, n) \cdot (I_p \otimes F_m) \cdot B_{p,n} = (\mathbf{a}_0 \cdot F_m,\, \mathbf{a}_1 \cdot F_m,\, \ldots,\, \mathbf{a}_{p-1} \cdot F_m) \cdot B_{p,n}\,, \tag{6.6.1}$$

where $m = n/p$ and $1 < p < m$.

Now, we consider the case where $n = 4^k$ and $p = 4$. The factorization is very similar to the radix-2 case.

For $p = 4$, Eq. 6.4.1 becomes

$$\begin{aligned} \mathbf{a} \cdot F_n &= \mathbf{a} \cdot \Pi_{4,n} \cdot (I_4 \otimes F_m) \cdot B_{4,n} \\ &= (\mathbf{a}_0 \cdot F_m,\, \mathbf{a}_1 \cdot F_m, \mathbf{a}_2 \cdot F_m,\, \mathbf{a}_3 \cdot F_m) \cdot B_{4,n}\,. \end{aligned} \tag{6.6.2}$$

Assuming $n = 4^k$, the split-and-merge procedure can be replaced by applying Corollary 6.3.3 to $\mathbf{a}_j \cdot F_m$ $(j = 0,\, 1, 2, 3)$ which splits each $\mathbf{a}_j \cdot F_m$ into

$$\mathbf{a}_j \cdot F_m = \left(\mathbf{a}_{j0} \cdot F_{m/2},\, \mathbf{a}_{j1} \cdot F_{m/2}, \mathbf{a}_{j2} \cdot F_{m/2},\, \mathbf{a}_{j3} \cdot F_{m/2}\right) \cdot B_{4,m}\,. \tag{6.6.3}$$

Merging 6.6.3 with Eq. 6.6.2 results in

$$\begin{aligned}\mathbf{a}\cdot F_n = [&\big(\mathbf{a}_{00}\cdot F_{m/4},\, \mathbf{a}_{01}\cdot F_{m/4}, \mathbf{a}_{02}\cdot F_{m/4},\, \mathbf{a}_{03}\cdot F_{m/4}\big)\cdot B_{4,m}\,,\\ &\big(\mathbf{a}_{10}\cdot F_{m/4},\, \mathbf{a}_{11}\cdot F_{m/4}, \mathbf{a}_{12}\cdot F_{m/4},\, \mathbf{a}_{13}\cdot F_{m/4}\big)\cdot B_{4,m},\\ &\big(\mathbf{a}_{20}\cdot F_{m/4},\, \mathbf{a}_{21}\cdot F_{m/4}, \mathbf{a}_{22}\cdot F_{m/4},\, \mathbf{a}_{23}\cdot F_{m/4}\big)\cdot B_{4,m},\\ &\big(\mathbf{a}_{30}\cdot F_{m/4},\, \mathbf{a}_{31}\cdot F_{m/4}, \mathbf{a}_{32}\cdot F_{m/4},\, \mathbf{a}_{33}\cdot F_{m/4}\big)\cdot B_{4,m}]\cdot B_{4,n}\,.\end{aligned} \tag{6.6.4}$$

Using tensor notation, we may rewrite Eqs. 6.6.2 and 6.6.4 as

$$\mathbf{a}\cdot F_n = \big(\mathbf{a}_0\cdot F_{n/4},\, \mathbf{a}_1\cdot F_{n/4}, \mathbf{a}_2\cdot F_{n/4},\, \mathbf{a}_3\cdot F_{n/4}\big)\cdot (I_1\otimes B_{4,n}) \tag{6.6.5}$$

and

$$\begin{aligned}\mathbf{a}\cdot F_n = (&\mathbf{a}_{00}\cdot F_{n/4^2},\, \mathbf{a}_{01}\cdot F_{n/4^2}, \mathbf{a}_{02}\cdot F_{n/4^2},\, \mathbf{a}_{03}\cdot F_{n/4^2}\,,\\ &\mathbf{a}_{10}\cdot F_{n/4^2},\, \mathbf{a}_{11}\cdot F_{n/4^2}, \mathbf{a}_{12}\cdot F_{n/4^2},\, \mathbf{a}_{13}\cdot F_{n/4^2},\\ &\mathbf{a}_{20}\cdot F_{n/4^2},\, \mathbf{a}_{21}\cdot F_{n/4^2}, \mathbf{a}_{22}\cdot F_{n/4^2},\, \mathbf{a}_{23}\cdot F_{n/4^2}.\\ &\mathbf{a}_{30}\cdot F_{n/4^2},\, \mathbf{a}_{31}\cdot F_{n/4^2}, \mathbf{a}_{32}\cdot F_{n/4^2},\, \mathbf{a}_{33}\cdot F_{n/4^2})\cdot \big(I_4\otimes B_{4,n/4}\big)\cdot (I_1\otimes B_{4,n}),\end{aligned} \tag{6.6.6}$$

respectively.

Since $F_1 = (1)$, continuation of the split-and-merge process $k-3$ more times leads to the following formulation of the DFT:

$$\mathbf{a}\cdot F_n = \big(\mathbf{a}_{j_0},\, \mathbf{a}_{j_1},\, \ldots,\, \mathbf{a}_{j_{n-1}}\big)\cdot \big(I_{n/4}\otimes B_{4,4}\big)\cdots\big(I_4\otimes B_{4,n/4}\big)\cdot (I_1\otimes B_{4,n})\,, \tag{6.6.7}$$

where each of the subscripts j_r is length *k* sequence of base 4 numbers.

For $s = (s_0,\, s_1,\, \ldots,\, s_{k-1}) \in \prod_{i=0}^{k-1} \mathbb{Z}_4$, let $(s)_4$ denote the base 4 expansion of *s*; i.e.,

$$(s)_4 = s_0 + s_1 4 + \cdots + s_{k-1}4^{k-1}\,.$$

Using induction on *k*, it can be shown that $\mathbf{a}_{j_r} = \mathbf{a}((j_r)_4)$ for $r = 0,\, 1,\, \ldots,\, n-1$. Defining $\rho_{4,n} : \mathbb{Z}_n \to \mathbb{Z}_n$ by $\rho_{4,n}(r) = (j_r)_4$, we have

$$\big(\mathbf{a}_{j_0},\, \mathbf{a}_{j_1},\, \ldots,\, \mathbf{a}_{j_{n-1}}\big) = \mathbf{a}\cdot P_{4,n}\,, \tag{6.6.8}$$

where $P_{4,n}$ is the permutation matrix defined by $P_{4,n} \equiv P_{\rho_{4,n}}$. This means that the DFT of $\mathbf{a} \in \mathbb{C}^n$ consists of a permutation of the components of **a** followed by a product of *k* sparse matrices.

The following example will provide the necessary insight into understanding the action of $P_{4,n}$ on the coordinates of a vector.

6.6.1 **Example:** Suppose $n = 4^2$ and $\mathbf{a} \in \mathbb{C}^{16}$. Then

$$\mathbf{a}\cdot F_{16} = (\mathbf{a}_0\cdot F_4,\, \mathbf{a}_1\cdot F_4, \mathbf{a}_2\cdot F_4,\, \mathbf{a}_3\cdot F_4)\cdot (I_1\otimes B_{4,16})\,,$$

where

$$\mathbf{a}_0 = (\mathbf{a}(0),\, \mathbf{a}(4),\, \mathbf{a}(8),\, \mathbf{a}(12)),$$

$$\mathbf{a}_1 = (\mathbf{a}(1),\, \mathbf{a}(5),\, \mathbf{a}(9),\, \mathbf{a}(13)),$$

$$\mathbf{a}_2 = (\mathbf{a}(2), \mathbf{a}(6), \mathbf{a}(10), \mathbf{a}(14)),$$

and

$$\mathbf{a}_3 = (\mathbf{a}(3), \mathbf{a}(7), \mathbf{a}(11), \mathbf{a}(15)).$$

Repeating this process, we obtain

$$\begin{aligned}
&(\mathbf{a}_0 \cdot F_4, \mathbf{a}_1 \cdot F_4, \mathbf{a}_2 \cdot F_4, \mathbf{a}_3 \cdot F_4) \\
&= (\mathbf{a}_{00} \cdot F_1, \mathbf{a}_{01} \cdot F_1, \mathbf{a}_{02} \cdot F_1, \mathbf{a}_{03} \cdot F_1, \\
&\quad \mathbf{a}_{00} \cdot F_1, \mathbf{a}_{01} \cdot F_1, \mathbf{a}_{02} \cdot F_1, \mathbf{a}_{03} \cdot F_1, \\
&\quad \mathbf{a}_{00} \cdot F_1, \mathbf{a}_{01} \cdot F_1, \mathbf{a}_{02} \cdot F_1, \mathbf{a}_{03} \cdot F_1, \\
&\quad \mathbf{a}_{00} \cdot F_1, \mathbf{a}_{01} \cdot F_1, \mathbf{a}_{02} \cdot F_1, \mathbf{a}_{03} \cdot F_1) \cdot (I_4 \otimes B_{4,4}) \\
&= (\mathbf{a}_{00}, \mathbf{a}_{01}, \mathbf{a}_{02}, \mathbf{a}_{03}, \\
&\quad \mathbf{a}_{10}, \mathbf{a}_{11}, \mathbf{a}_{12}, \mathbf{a}_{13}, \\
&\quad \mathbf{a}_{20}, \mathbf{a}_{21}, \mathbf{a}_{22}, \mathbf{a}_{23}, \\
&\quad \mathbf{a}_{30}, \mathbf{a}_{31}, \mathbf{a}_{32}, \mathbf{a}_{33}) \cdot (I_4 \otimes B_{4,4}) \\
&= (\mathbf{a}(0), \mathbf{a}(4), \mathbf{a}(8), \mathbf{a}(12), \\
&\quad \mathbf{a}(1), \mathbf{a}(5), \mathbf{a}(9), \mathbf{a}(13), \\
&\quad \mathbf{a}(2), \mathbf{a}(6), \mathbf{a}(10), \mathbf{a}(14), \\
&\quad \mathbf{a}(3), \mathbf{a}(7), \mathbf{a}(11), \mathbf{a}(15)) \cdot (I_4 \otimes B_{4,4}).
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbf{a} \cdot P_{4,16} &= (\mathbf{a}(0), \mathbf{a}(4), \mathbf{a}(8), \mathbf{a}(12), \\
&\quad \mathbf{a}(1), \mathbf{a}(5), \mathbf{a}(9), \mathbf{a}(13), \\
&\quad \mathbf{a}(2), \mathbf{a}(6), \mathbf{a}(10), \mathbf{a}(14), \\
&\quad \mathbf{a}(3), \mathbf{a}(7), \mathbf{a}(11), \mathbf{a}(15)) \\
&= (\mathbf{a}(\rho_{4,16}(0)), \mathbf{a}(\rho_{4,16}(1)), \mathbf{a}(\rho_{4,16}(2)), \mathbf{a}(\rho_{4,16}(3)), \\
&\quad \mathbf{a}(\rho_{4,16}(4)), \mathbf{a}(\rho_{4,16}(5)), \mathbf{a}(\rho_{4,16}(6)), \mathbf{a}(\rho_{4,16}(7)), \\
&\quad \mathbf{a}(\rho_{4,16}(8)), \mathbf{a}(\rho_{4,16}(9)), \mathbf{a}(\rho_{4,16}(10)), \mathbf{a}(\rho_{4,16}(11)), \\
&\quad \mathbf{a}(\rho_{4,16}(12)), \mathbf{a}(\rho_{4,16}(13)), \mathbf{a}(\rho_{4,16}(14)), \mathbf{a}(\rho_{4,16}(15))).
\end{aligned}$$

Expressing i and $\rho_{4,16}(i)$ in terms of base 4 representations elucidates the action of $P_{4,n}$. Specifically, we have the following relationship:

i	$\rho_{4,16}(i)$	i (base 4)	$\rho_{4,16}(i)$ (base 4)
0	0	00	00
1	4	01	10
2	8	02	20
3	12	03	30
4	1	10	01

Figure 6.6.1 The permutation $\rho_{4,16}$ and its corresponding binary evaluation (Continued) . . .

i	$\rho_{4,16}(i)$	i (base 4)	$\rho_{4,16}(i)$ (base 4)
5	5	11	11
6	9	12	21
7	13	13	31
8	2	20	02
9	6	21	12
10	10	22	22
11	14	23	32
12	3	30	03
13	7	31	13
14	11	32	23
15	15	33	33

Figure 6.6.1 The permutation $\rho_{4,16}$ and its corresponding binary evaluation

It follows from the table that $\rho_{4,16}(i)$ is obtained from *i* by reversing the order (reading from right-to-left) of the digits in the base 4 expression of *i*.

The following algorithm computes the permutation function $\rho_{4,n}$.

6.6.2 Algorithm (Evaluating $\rho_{4,n}(i)$)

For $n = 4^k$ and $i = 0, 1, \ldots, n-1$ define $\rho_{4,n}(i)$ as:

```
begin
        j := 0
        m := i
        for l := 0 to log4(n) − 1 loop
                h := ⌊m/4⌋
                j := 4j + (m − 4h)
                m := h
        end loop
        return j
end
```

For $\mathbf{a} \in \mathbb{C}^n$, the computation of $\mathbf{a} \cdot P_{4,n}$ can now be accomplished by using the simple image algebra specification:

$$\mathbf{b} := \mathbf{a} \circ \rho_{4,n}$$

Of course, by specifying the template $\mathbf{p} = \psi^{-1}(P_{4,n})$, we could just as well compute $\mathbf{a} \cdot P_{4,n}$ in terms of the convolution product $\mathbf{a} \oplus \mathbf{p}$.

6.7 Radix-4 FFT Algorithm

Using the results of the previous section, we can write $\mathbf{a} \cdot F_n$ as

$$\mathbf{a} \cdot F_n = \mathbf{a} \cdot P_{4,n} \cdot A_1 \cdot A_2 \cdots A_k \,, \tag{6.7.1}$$

where $n = 4^k$ and $A_i = I_{4^i} \otimes B_{4,4^i}, \quad i = 1, 2, \ldots, k$. Thus, we may use the following algorithm to compute$\mathbf{a} \cdot F_n$:

6.7.1 **Algorithm** (Radix-4 FFT)

$\mathbf{a} := \mathbf{a} \cdot P_{4,n}$ (Algorithm **6.6.2**)
for $i := 1$ **to** k **loop**
    $\mathbf{a} := \mathbf{a} \cdot A_i$
end loop

Since each A_i has only four nonzero entries per column and each complex add involves two floating point operations, or *flops*, while each complex multiply involves six flops, it is not difficult to ascertain that this algorithm requires only $O(n \cdot log_4 n)$ flops if the sparsity of the A_is is exploited.

The image algebra equivalent of Eq. 6.7.1 is given by

$$\mathbf{a} \oplus \mathbf{f} = (\mathbf{a} \circ \rho_{4,n}) \oplus \mathbf{t}(1) \oplus \mathbf{t}(2) \oplus \cdots \oplus \mathbf{t}(k) \,, \tag{6.7.2}$$

where $\mathbf{t}(i) = \psi^{-1}(A_i)$.

To compute Eq. 6.7.2, we may use the following algorithm which looks very much like Algorithm **6.7.1**.

6.7.2 **Algorithm**

$\mathbf{a} := \mathbf{a} \circ \rho_{4,n}$ (Algorithm **6.6.2**)
for $i := 1$ **to** k **loop**
    $\mathbf{a} := \mathbf{a} \oplus \mathbf{t}(i)$
end loop

To complete the specification of this algorithm, we need to examine the parameters defining the templates $\mathbf{t}(i) = \psi^{-1}(A_i)$.

Suppose $k = 2$ and $n = 4^k$. In this case we need to specify $\mathbf{t}(1)$ and $\mathbf{t}(2)$ corresponding to the matrices A_1 and A_2, respectively. Using the tensor representation of these matrices, we obtain

$$
\begin{aligned}
A_1 &= I_4 \otimes B_{4,4} = I_4 \otimes diag\big(I_1, \Omega_{4,1}, \Omega_{4,1}^2, \Omega_{4,1}^3\big) \cdot (F_4 \otimes I_1) \\
&= I_4 \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \\
&= \begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -i & -1 & i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & i & -1 & -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -i & -1 & i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & i & -1 & -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -i & -1 & i & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & i & -1 & -i & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -i & -1 & i \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & i & -1 & -i
\end{pmatrix}
\end{aligned}
$$

and

$$
\begin{aligned}
A_2 &= I_1 \otimes B_{4,16} = I_1 \otimes diag\big(I_1, \Omega_{4,4}, \Omega_{4,4}^2, \Omega_{4,4}^3\big) \cdot (F_4 \otimes I_4) \\
&= \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & -i & 0 & 0 & 0 & -1 & 0 & 0 & 0 & i & 0 & 0 & 0 \\
0 & \omega_{16} & 0 & 0 & 0 & -i\omega_{16} & 0 & 0 & 0 & -\omega_{16} & 0 & 0 & 0 & i\omega_{16} & 0 & 0 \\
0 & 0 & \omega_{16}^2 & 0 & 0 & 0 & -i\omega_{16}^2 & 0 & 0 & 0 & -\omega_{16}^2 & 0 & 0 & 0 & i\omega_{16}^2 & 0 \\
0 & 0 & 0 & \omega_{16}^3 & 0 & 0 & 0 & -i\omega_{16}^3 & 0 & 0 & 0 & -\omega_{16}^3 & 0 & 0 & 0 & i\omega_{16}^3 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & \omega_{16}^2 & 0 & 0 & 0 & -\omega_{16}^2 & 0 & 0 & 0 & \omega_{16}^2 & 0 & 0 & 0 & -\omega_{16}^2 & 0 & 0 \\
0 & 0 & \omega_{16}^4 & 0 & 0 & 0 & -\omega_{16}^4 & 0 & 0 & 0 & \omega_{16}^4 & 0 & 0 & 0 & -\omega_{16}^4 & 0 \\
0 & 0 & 0 & \omega_{16}^6 & 0 & 0 & 0 & -\omega_{16}^6 & 0 & 0 & 0 & \omega_{16}^6 & 0 & 0 & 0 & -\omega_{16}^6 \\
1 & 0 & 0 & 0 & i & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\
0 & \omega_{16}^3 & 0 & 0 & 0 & i\omega_{16}^3 & 0 & 0 & 0 & -\omega_{16}^3 & 0 & 0 & 0 & -i\omega_{16}^3 & 0 & 0 \\
0 & 0 & \omega_{16}^6 & 0 & 0 & 0 & i\omega_{16}^6 & 0 & 0 & 0 & -\omega_{16}^6 & 0 & 0 & 0 & -i\omega_{16}^6 & 0 \\
0 & 0 & 0 & \omega_{16}^9 & 0 & 0 & 0 & i\omega_{16}^9 & 0 & 0 & 0 & -\omega_{16}^9 & 0 & 0 & 0 & -i\omega_{16}^9
\end{pmatrix}
\end{aligned}
$$

At the *i*th stage of the loop in Algorithm **6.7.2** we compute the product $\mathbf{b}_i = \mathbf{b}_{i-1} \bigoplus \mathbf{t}(i)$, where $\mathbf{b}_0 = \mathbf{a} \circ \rho_{4,16}$. By definition of the product operator $\bigoplus$, the value $\mathbf{b}_i(j)$ is given by

$$\mathbf{b}_i(j) = \sum_{l \in S\left(\mathbf{t}(i)_j\right)} \mathbf{b}_{i-1}(l) \cdot \mathbf{t}(i)_j(l)\,. \tag{6.7.3}$$

The cardinality of $S\left(\mathbf{t}(i)_j\right)$ is four since the template image $\mathbf{t}(i)_j$ corresponds to the *j*th column of A_i. Thus, using Eq. 6.7.3, the computation of each value $\mathbf{b}_i(j)$ involves four multiplications and three additions which shows that the image algebra formulation $\mathbf{a} := \mathbf{a} \bigoplus \mathbf{t}(i)$ exploits the sparsity of A_i.

To complete the specification of the template $\mathbf{t}(i)$, we need to define the weights $\mathbf{t}(i)_j(l)$. Let $\mathbf{b}_i = \mathbf{b}_{i-1} \bigoplus \mathbf{t}(i)$, where $\mathbf{b}_0 = \mathbf{a} \circ \rho_{4,16}$. Let

$$W(w) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ w & -iw & -w & iw \\ w^2 & -w^2 & w^2 & -w^2 \\ w^3 & iw^3 & -w^3 & -iw^3 \end{pmatrix}, \tag{6.7.4}$$

where $w = \omega_{4^i}^{j \bmod 4^{i-1}}$. By carefully considering the columns of the matrices A_i, it is not difficult to see that $\mathbf{b}_i(j)$, $\mathbf{b}_i\left(j + 4^{i-1}\right)$, $\mathbf{b}_i\left(j + 2 * 4^{i-1}\right)$, and $\mathbf{b}_i\left(j + 3 * 4^{i-1}\right)$ can be computed in terms of the following vector product

$$\begin{gathered} \left(\mathbf{b}_i(j),\, \mathbf{b}_i\left(j + 4^{i-1}\right), \mathbf{b}_i\left(j + 2 * 4^{i-1}\right), \mathbf{b}_i\left(j + 3 * 4^{i-1}\right)\right) = \\ \left(\mathbf{b}_{i-1}(j),\, \mathbf{b}_{i-1}\left(j + 4^{i-1}\right), \mathbf{b}_{i-1}\left(j + 2 * 4^{i-1}\right), \mathbf{b}_{i-1}\left(j + 3 * 4^{i-1}\right)\right) \cdot W(w)\,. \end{gathered} \tag{6.7.5}$$

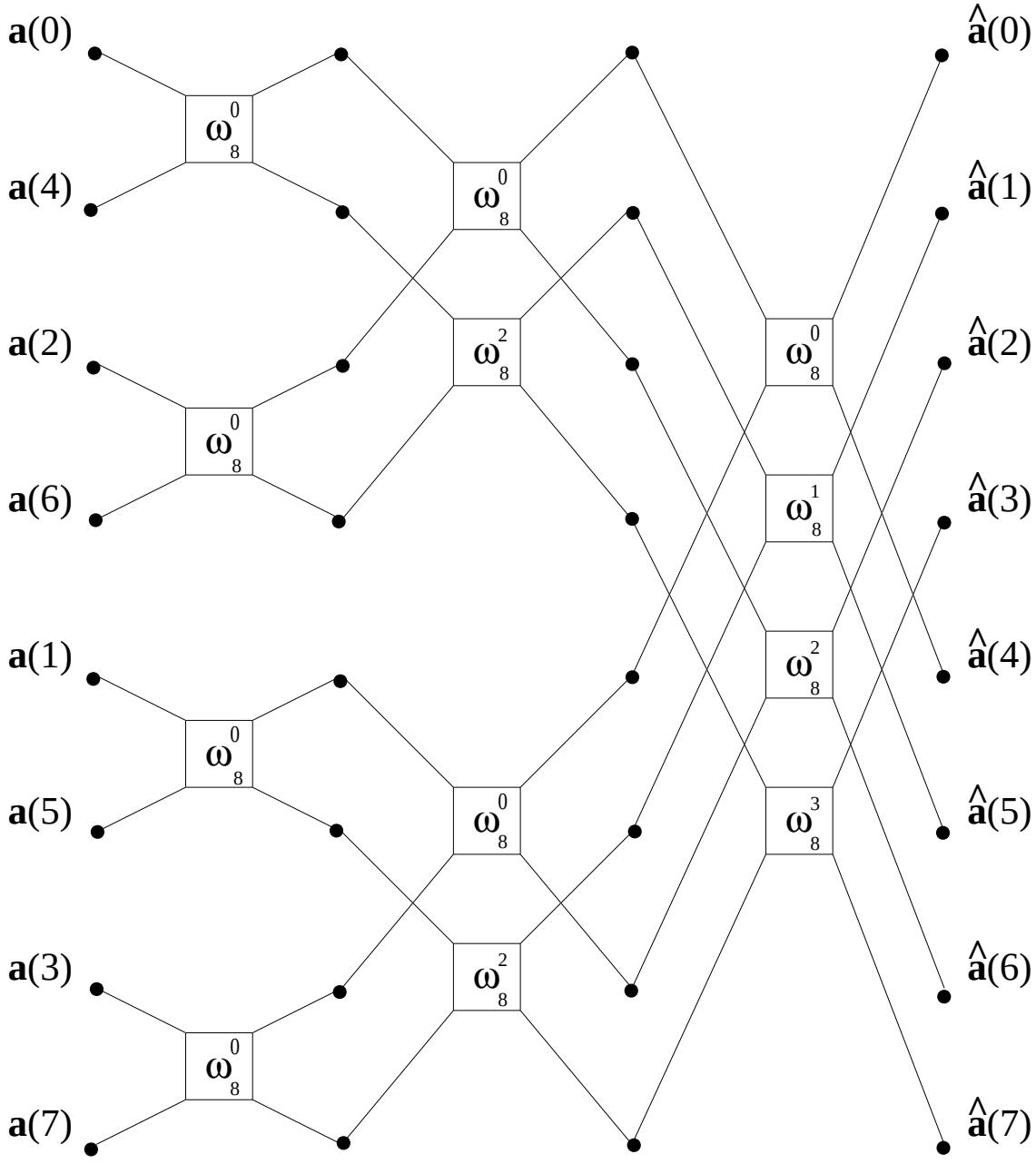


Figure 6.7.1 The Cooley-Tukey data flow graph for $n = 16$.

Equation 6.7.5 is the key for representing the templates $\mathbf{t}(1)$ and $\mathbf{t}(2)$ in terms of a single parametrized template $\mathbf{t}(p)$. Specifically, let $P = \{2^i : i = 0, 1\}$ and for $p \in P$, define $\mathbf{t}(p)$ by

$$\mathbf{t}(p)_j(l) = \begin{cases} \left[W\left(\omega_{4^p}^{j \bmod p}\right)\right]_{r,s} & \text{if } l - j + p \cdot \left(\left\lfloor \frac{j}{p} \right\rfloor \bmod 4\right) = 0, p, 2p, \text{or } 3p \\ 0 & \text{otherwise,} \end{cases} \tag{6.7.6}$$

where $r = \frac{l-j}{p} + \left(\left\lfloor \frac{j}{p} \right\rfloor \, mod \, 4 \right)$ and $s = \left\lfloor \frac{j}{p} \right\rfloor \, mod \, 4$. Using the convention $w(j,1) = 0 \; \forall j$ and simple inspection verifies that for the parameters $p = 4^0$ and 4^1, the templates generated by $\mathbf{t}(p)$ correspond to the templates $\psi^{-1}(A_1)$ and $\psi^{-1}(A_2)$, respectively.

The 16-point radix-4 FFT can easily be generalized.

With the specification of $\mathbf{t}(p)$ completed, we are now able to formulate the Cooley-Tukey radix-4 FFT using the language of image algebra.

6.7.3 **Algorithm** (Radix-4 FFT). If $\mathbf{a} \in \mathbb{C}^n$, $n = 4^k$, and $\mathbf{t}(p)$ is specified by Eq. 6.7.6, then the following algorithm computes $\mathbf{a} \bigoplus \mathbf{f}$.

$\mathbf{a} := \mathbf{a} \circ \rho_{4,n}$
for $i := 1$ **to** $log_4 n$ **loop**
    $\mathbf{a} := \mathbf{a} \bigoplus \mathbf{t}\left(4^{i-1}\right)$
end loop

Bibliography

[1] E.O. Brigham. *The Fast Fourier Transform*. Prentice-Hall, Englewood Cliffs, N.J., 1974.

[2] P. Budnik and D.J. Kuck. The organization and use of parallel memories. *IEEE Transactions on Computers*, C-20:1566–1569, 1971.

[3] J.W. Cooley, P.A. Lewis, and P.D. Welch. The fast Fourier transform and its applications. *IEEE Transactions on Education*, E-12(1):27–34, 1969.

[4] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[5] G.R. Fischer and M.R. Rowlee. Computation of disparity in stereo images on the Connection Machine. In *Image Algebra and Morphological Image Processing*, volume 1350 of *Proceedings of SPIE*, pages 328–334, 1990.

[6] P.D. Gader. *Image Algebra Techniques for Parallel Computation of Discrete Fourier Transforms and General Linear Transforms*. PhD thesis, University of Florida, Gainesville, FL, 1986.

[7] M.T. Heideman, D.H. Johnson, and C.S. Burrus. Gauss and the history of the fast fourier transform. *Arch. Hist. Exact Sci.*, 34:265–277, 1985.

[8] D.H. Lawrie. Access and alignment of data in an array processor. *IEEE Transactions on Computers*, C-24:99–109, 1975.

[9] D.H. Lawrie and C.R. Vora. The prime memory system for array access. *IEEE Transactions on Computers*, C-31:1435–1442, 1982.

[10] G.X. Ritter and P.D. Gader. Image algebra: Techniques for parallel image processing. *Journal of Parallel and Distributed Computing*, 4(5):7–44, April 1987.

[11] J.N. Wilson, G.R. Fischer, and G.X. Ritter. Implementation and use of an image processing algebra for programming massively parallel computers. In *Frontiers '88: The Second Symposium on the Frontiers of Massively Parallel Computation*, pages 587–594, Fairfax, VA, 1988.

CHAPTER 7
TRANSLATION INVARIANT TEMPLATES ON FINITE DOMAINS

In this chapter we are concerned with two important classes of templates, translation invariant and circulant templates. These templates are used to implement various convolutions and occur frequently in digital image processing. As we shall show, these templates are closely related to the discrete Fourier transform and its efficient computation. However, the main emphasis of this chapter is on the algebras associated with translation invariant templates on finite domains.

7.1 Translation Invariant Templates and Toeplitz Matrices

It follows from earlier chapters that translation invariant templates occur naturally in image processing. Translation invariance was defined in terms of a group $(\mathbf{X}, +)$ (Section **4.1**) such as $(\mathbb{Z} \times \mathbb{Z}, +)$. However, in most image processing tasks take place on finite subsets of the discrete plane $\mathbb{Z} \times \mathbb{Z}$ which are not closed sets under addition. For example, if $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n \subset \mathbb{Z} \times \mathbb{Z}$, then defining $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ to be translation invariant if $\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\mathbf{y}+\mathbf{z}}(\mathbf{x}+\mathbf{z})\ \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{X}$ makes little sense since $\mathbf{x}+\mathbf{z}$ or $\mathbf{y}+\mathbf{z}$ may not be points in $\mathbf{X}$. This poses the question as to the meaning of the term "translation invariant" or "shift invariant" in case the underlying point set $\mathbf{X}$ is not closed with respect to addition.

7.1.1 **Definition.** If $(\mathbf{Y}, +)$ is a group and $\mathbf{X} \subset \mathbf{Y}$, then $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is said to be *translation invariant on* $\mathbf{X}$ if and only if there exists a translation invariant template $\tilde{\mathbf{t}} \in \left(\mathbb{F}^{\mathbf{Y}}\right)^{\mathbf{Y}}$ such that $\tilde{\mathbf{t}}|_{(\mathbf{X}, \mathbb{F}^{\mathbf{X}})} = \mathbf{t}$.

Thus, translation invariant templates on $\mathbf{X}$ are restrictions of translation invariant templates on $\mathbf{Y}$. It follows that translation invariant templates on $\mathbf{X}$ *look very much like* translation invariant templates on $\mathbf{Y}$. For example, if $\mathbf{Y} = \mathbb{Z}^2$ and $\mathbf{1} \in \left(\mathbb{R}^{\mathbf{Y}}\right)^{\mathbf{Y}}$ denotes the unit template, then $\mathbf{1}|_{(\mathbf{X}, \mathbb{R}^{\mathbf{X}})}$ is also the unit template for the set $\left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$. However, the comment "look very much like" should not be taken too literally. Suppose $\mathbf{X} = \mathbb{Z}_3 \times \mathbb{Z}_4$ and $\tilde{\mathbf{t}} \in \left(\mathbb{R}^{\mathbb{Z}^2}\right)^{\mathbb{Z}^2}$ is defined by

$$\tilde{\mathbf{t}}_{(y_1,y_2)}(x_1, x_2) = \begin{cases} 1 & \text{if } (x_1, x_2) = (y_1, y_2) \\ 2 & \text{if } (x_1, x_2) = (y_1 \pm 10, y_2) \text{ or } (x_1, x_2) = (y_1, y_2 \pm 10) \\ 0 & \text{otherwise.} \end{cases}$$

If we restrict $\tilde{\mathbf{t}}$ to $\left(\mathbf{X}, \mathbb{R}^{\mathbf{X}}\right)$, then clearly $\tilde{\mathbf{t}}|_{(\mathbf{X}, \mathbb{R}^{\mathbf{X}})} = \mathbf{1}|_{(\mathbf{X}, \mathbb{R}^{\mathbf{X}})}$.

It therefore follows that if $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ is a translation invariant template, where $\mathbf{X} \subset \mathbb{Z}^2$ is finite, then there exists infinitely many templates in $\left(\mathbb{R}^{\mathbb{Z}^2}\right)^{\mathbb{Z}^2}$ whose restrictions to $\left(\mathbf{X}, \mathbb{R}^{\mathbf{X}}\right)$ equal $\mathbf{t}$. It is possible, however, to identify a translation invariant template with a unique translation invariant template $\tilde{\mathbf{t}}$ on $\mathbf{Y}$ which has *essentially* the same support on $\mathbf{X}$ as on $\mathbf{Y}$ (except, possibly, near the boundary of $\mathbf{X}$). Specifically, for each translation invariant template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$, let

$$R(\mathbf{t}) = \left\{ \mathbf{r} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}} : \mathbf{r} \text{ is translation invariant and } \mathbf{r}|_{(\mathbf{X}, \mathbb{F}^{\mathbf{X}})} = \mathbf{t} \right\},$$

and

$$S(\mathbf{t}, \mathbf{y}) = \bigcap_{\mathbf{r} \in R(\mathbf{t})} S(\mathbf{r}_{\mathbf{y}}), \quad \mathbf{y} \in \mathbf{Y}.$$

Suppose without loss of generality that $\mathbf{t}$ is not the zero template on $\mathbf{X}$. Then for some $\mathbf{y} \in \mathbf{X}$, $S(\mathbf{t_y}) \neq \emptyset$ and , therefore, $S(\mathbf{t},\mathbf{y}) \neq \emptyset$. Since each $\mathbf{r} \in R(\mathbf{t})$ is translation invariant, we have that $S(\mathbf{t},\mathbf{y}) \neq \emptyset \ \ \forall \mathbf{y} \in \mathbf{Y}$.

Now choose an arbitrary template $\mathbf{r} \in R(\mathbf{t})$ and note that if $\mathbf{x} \in S(\mathbf{t},\mathbf{y})$, then since $S(\mathbf{t},\mathbf{y}) \subset S(\mathbf{r_y})$, $\mathbf{r_y}(\mathbf{x}) \neq 0$. Define $\tilde{\mathbf{t}} \in \left(\mathbb{F}^{\mathbf{Y}}\right)^{\mathbf{Y}}$ by

$$\tilde{\mathbf{t}}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} \mathbf{r_y}(\mathbf{x}) & \text{if } \mathbf{x} \in S(\mathbf{t},\mathbf{y}) \\ 0 & \text{otherwise.} \end{cases}$$

It follows that $\tilde{\mathbf{t}} \in R(\mathbf{t})$, $\tilde{\mathbf{t}}|_{(\mathbf{X},\mathbb{F}^{\mathbf{X}})} = \mathbf{t}$, and that

$$S\left(\tilde{\mathbf{t}}_{\mathbf{y}}\right) \subset S(\mathbf{r_y}) \ \ \forall \mathbf{y} \in \mathbf{Y} \text{ and } \forall \mathbf{r} \in R(\mathbf{t}). \tag{7.1.1}$$

If $\mathbf{t}$ is the zero template on $\mathbf{X}$, then the unique template $\tilde{\mathbf{t}}$ corresponding to $\mathbf{t}$ is the zero element of $\left(\mathbb{F}^{\mathbf{Y}}\right)^{\mathbf{Y}}$. This shows that we can always identify a translation invariant template on $\mathbf{X}$ with a unique translation invariant template $\tilde{\mathbf{t}}$ on $\mathbf{Y}$, namely the one with *smallest* support as defined by Eq. 7.1.1.

Invariant templates can just as well be defined in terms of spatial transformations. If $(\mathbf{Y},+)$ is a group under point (vector) addition, then a function $\phi : \mathbf{Y} \to \mathbf{Y}$ is called a *translation* (or *shift*) if and only if there exists a point $\mathbf{z} \in \mathbf{Y}$ such that $\phi(\mathbf{y}) = \mathbf{y} + \mathbf{z} \ \ \forall \mathbf{y} \in \mathbf{Y}$. Thus, for each $\mathbf{z} \in \mathbf{Y}$, there exists a translation $\phi_{\mathbf{z}} : \mathbf{Y} \to \mathbf{Y}$ which is defined by $\phi_{\mathbf{z}}(\mathbf{y}) = \mathbf{y} + \mathbf{z} \ \ \forall \mathbf{y} \in \mathbf{Y}$.

If $\mathbf{y} \in \mathbf{Y}$ and $(\mathbf{Y},+)$ is an abelian group, then

$$(\phi_{\mathbf{z}} \circ \phi_{\mathbf{x}})(\mathbf{y}) = \phi_{\mathbf{z}}(\phi_{\mathbf{x}}(\mathbf{y})) = \phi_{\mathbf{z}}(\mathbf{y}+\mathbf{x}) = (\mathbf{y}+\mathbf{x})+\mathbf{z} = \mathbf{y}+(\mathbf{x}+\mathbf{z}) = \phi_{\mathbf{x}+\mathbf{z}}(\mathbf{y}).$$

Since $\mathbf{y}$ was arbitrarily chosen and $\mathbf{z}+\mathbf{x} = \mathbf{x}+\mathbf{z}$, this shows that

$$\phi_{\mathbf{z}} \circ \phi_{\mathbf{x}} = \phi_{\mathbf{x}+\mathbf{z}} = \phi_{\mathbf{z}+\mathbf{x}} = \phi_{\mathbf{x}} \circ \phi_{\mathbf{z}}\,. \tag{7.1.2}$$

As a trivial consequence of this equation we have the following result:

7.1.2 Theorem. *If $(\mathbf{Y},+)$ is an abelian group and $\Phi = \{\phi_{\mathbf{z}} : \mathbf{z} \in \mathbf{Y}\}$, then Φ is an abelian group under the operation of composition.*

Proof:

$$\begin{aligned}\phi_{\mathbf{z}} \circ (\phi_{\mathbf{x}} \circ \phi_{\mathbf{y}}) &= \phi_{\mathbf{z}} \circ \phi_{\mathbf{x}+\mathbf{y}} = \phi_{\mathbf{z}+(\mathbf{x}+\mathbf{y})} \\ &= \phi_{(\mathbf{z}+\mathbf{x})+\mathbf{y}} = \phi_{\mathbf{z}+\mathbf{x}} \circ \phi_{\mathbf{y}} = (\phi_{\mathbf{z}} \circ \phi_{\mathbf{x}}) \circ \phi_{\mathbf{y}}\,.\end{aligned}$$

This proves associativity. Commutativity follows from Eq. 7.1.2.

Since $\phi_{\mathbf{z}} \circ \phi_0 = \phi_{\mathbf{z}+0} = \phi_{\mathbf{z}}$, ϕ_0 is the identity of Φ.

Finally, since $\phi_{\mathbf{z}} \circ \phi_{-\mathbf{z}} = \phi_{\mathbf{z}+(-\mathbf{z})} = \phi_0$, each element of Φ has an inverse.

Q.E.D.

For $\mathbf{X} \subset \mathbf{Y}$, define

$$\Phi|_{\mathbf{X}} = \left\{\phi \,:\, \phi = \phi_{\mathbf{z}}|_{(\mathbf{X},\mathbf{X})}\,,\ \phi_{\mathbf{z}} \in \Phi\right\}.$$

Note that if $\phi \in \Phi|_{\mathbf{X}}$, then $\phi : domain(\phi) \to \mathbf{X}$, where $domain(\phi) = \{\mathbf{x} \in \mathbf{X} :\ \phi(\mathbf{x}) \in \mathbf{X}\} \subset \mathbf{X}$. Thus, in general, $\phi \in \Phi|_{\mathbf{X}}$ is not a function *from* $\mathbf{X}$ to $\mathbf{X}$.

We are now in a position of classifying translation invariant templates in terms of elements of $\Phi|_{\mathbf{X}}$.

7.1.3 **Theorem.** $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *is translation invariant* $\Leftrightarrow$ $\forall \phi \in \Phi|_{\mathbf{X}}$ *and* $\forall \mathbf{x}, \mathbf{y} \in domain(\phi)$, $\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\phi(\mathbf{y})}(\phi(\mathbf{x}))$.

Proof: Suppose that $\mathbf{t}$ is translation invariant. Let $\phi \in \Phi|_{\mathbf{X}}$ and $\mathbf{x}, \mathbf{y} \in domain(\phi)$. By definition of $\Phi|_{\mathbf{X}}$, there exists $\mathbf{z} \in \mathbf{Y}$ such that $\phi(\mathbf{u}) = \mathbf{u} + \mathbf{z}$ $\forall \mathbf{u} \in domain(\phi)$. Since $\mathbf{t}$ is translation invariant on $\mathbf{X}$, we now have

$$\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\mathbf{y}+\mathbf{z}}(\mathbf{x}+\mathbf{z}) = \mathbf{t}_{\phi(\mathbf{y})}(\phi(\mathbf{x}))\,.$$

To prove the converse, let $\mathbf{x}, \mathbf{y} \in \mathbf{Y}$ and define

$$T(\mathbf{x}, \mathbf{y}) = \{\mathbf{z} \in \mathbf{Y} \,:\, \mathbf{x} - \mathbf{z} \in \mathbf{X} \;\; and \;\; \mathbf{y} - \mathbf{z} \in \mathbf{X}\}\,.$$

Note that if $T(\mathbf{x}, \mathbf{y}) \neq \varnothing$, then $T(\mathbf{x}+\mathbf{w}, \mathbf{y}+\mathbf{w}) \neq \varnothing$ $\forall \mathbf{w} \in \mathbf{Y}$. Suppose $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ satisfies $\mathbf{t}_{\mathbf{y}_0}(\mathbf{x}_0) = \mathbf{t}_{\phi(\mathbf{y}_0)}(\phi(\mathbf{x}_0))$ $\forall \phi \in \Phi|_{\mathbf{X}}$ and $\forall \mathbf{x}_0, \mathbf{y}_0 \in domain(\phi)$. Observe that if $\mathbf{u}, \mathbf{v} \in T(\mathbf{x}, \mathbf{y})$, then

$$\mathbf{x} - \mathbf{u} \in \mathbf{X},\;\; \mathbf{y} - \mathbf{u} \in X,\;\; \mathbf{x} - \mathbf{v} \in \mathbf{X},\;\; \text{and}\;\; \mathbf{y} - \mathbf{v} \in \mathbf{X}\,. \tag{i}$$

Thus, if $\mathbf{z} = \mathbf{u} - \mathbf{v}$, then $\phi = \phi_{\mathbf{z}}|_{(\mathbf{X},\mathbf{X})} \in \Phi|_{\mathbf{X}}$ and the following equations hold:

$$\phi(\mathbf{x} - \mathbf{u}) = \mathbf{x} - \mathbf{v} \;\text{ and }\; \phi(\mathbf{y} - \mathbf{u}) = \mathbf{y} - \mathbf{v}\,. \tag{ii}$$

In particular, we have

$$\mathbf{t}_{\mathbf{y}-\mathbf{u}}(\mathbf{x} - \mathbf{u}) = \mathbf{t}_{\phi(\mathbf{y}-\mathbf{u})}(\phi(\mathbf{x} - \mathbf{u})) = \mathbf{t}_{\mathbf{y}-\mathbf{v}}(\mathbf{x} - \mathbf{v})\,. \tag{iii}$$

Now define $\tilde{\mathbf{t}} \in \left(\mathbb{F}^{\mathbf{Y}}\right)^{\mathbf{Y}}$ by

$$\tilde{\mathbf{t}}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} \mathbf{t}_{\mathbf{y}-\mathbf{u}}(\mathbf{x} - \mathbf{u}), & \text{where } \mathbf{u} \in T(\mathbf{x}, \mathbf{y}) \text{ is arbitrary, if } T(\mathbf{x}, \mathbf{y}) \neq \varnothing \\ 0 & \text{if } T(\mathbf{x}, \mathbf{y}) = \varnothing. \end{cases}$$

It follows from our observation (Eqs. (i), (ii), and (iii)) that $\tilde{\mathbf{t}}$ is well defined.

Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{Y}$. If $T(\mathbf{x}, \mathbf{y}) = \varnothing$, then $T(\mathbf{x}+\mathbf{z}, \mathbf{y}+\mathbf{z}) = \varnothing$ and , hence, $\tilde{\mathbf{t}}_{\mathbf{y}}(\mathbf{x}) = 0 = \tilde{\mathbf{t}}_{\mathbf{y}+\mathbf{z}}(\mathbf{x}+\mathbf{z})$. If $T(\mathbf{x}, \mathbf{y}) \neq \varnothing$, let $\mathbf{u} \in T(\mathbf{x}, \mathbf{y})$, $\mathbf{v} \in T(\mathbf{x}+\mathbf{z}, \mathbf{y}+\mathbf{z})$, $\mathbf{w} = \mathbf{z} - \mathbf{v} + \mathbf{u}$, and $\phi = \phi_{\mathbf{w}}|_{(\mathbf{X},\mathbf{X})}$. Then $\mathbf{x} - \mathbf{u} \in \mathbf{X}$ and $\phi(\mathbf{x} - \mathbf{u}) = (\mathbf{x} - \mathbf{u}) + \mathbf{w} = (\mathbf{x} + \mathbf{z}) - \mathbf{v} \in \mathbf{X}$. Likewise, we have $\phi(\mathbf{y} - \mathbf{u}) = (\mathbf{y} - \mathbf{u}) + \mathbf{w} = (\mathbf{y} + \mathbf{z}) - \mathbf{v} \in \mathbf{X}$. Thus, $\mathbf{x} - \mathbf{u}$, $\mathbf{y} = \mathbf{u} \in domain(\phi)$ and

$$\tilde{\mathbf{t}}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\mathbf{y}-\mathbf{u}}(\mathbf{x} - \mathbf{u}) = \mathbf{t}_{\phi(\mathbf{y}-\mathbf{u})}(\phi(\mathbf{x} - \mathbf{u})) = \mathbf{t}_{(\mathbf{y}+\mathbf{z})-\mathbf{v}}((\mathbf{x} + \mathbf{z}) - \mathbf{v}) = \tilde{\mathbf{t}}_{\mathbf{y}+\mathbf{z}}(\mathbf{x} + \mathbf{z})\,.$$

Therefore, $\tilde{\mathbf{t}}$ is translation invariant on $\mathbf{Y}$.

Clearly, if $\mathbf{x}, \mathbf{y} \in \mathbf{X}$, then $\tilde{\mathbf{t}}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\mathbf{y}}(\mathbf{x})$. Hence, $\tilde{\mathbf{t}}|_{(\mathbf{X},\mathbb{F}^{\mathbf{X}})} = \mathbf{t}$. Therefore $\mathbf{t}$ is translation invariant on $\mathbf{X}$.

Q.E.D.

Translation invariant templates on a finite point set $\mathbf{X}$ can also be classified in terms of special types of Toeplitz matrices.

7.1.4 **Definition.** Let $A = (a_{ij})_{n \times n}$. We say that *A* is a *Toeplitz matrix* if and only if for every pair $i, j \in \mathbb{Z}_n$ and for every integer *k* with the property $i + k, j + k \in \mathbb{Z}_n$ it follows that $a_{ij} = a_{i+k,j+k}$.

According to this definition, Toeplitz matrices are constant along their diagonals.

7.1.5 **Definition.** An $mn \times mn$ matrix $A = (a_{ij})$ is said to be *block Toeplitz with Toeplitz blocks* if and only if *A* is of form

$$A = \begin{pmatrix} A_0 & A_1 & \cdots & A_{m-1} \\ A_{-1} & A_0 & \cdots & A_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{-(m-1)} & A_{-(m-2)} & \cdots & A_0 \end{pmatrix},$$

where for each $i \in \mathbb{Z}_{\pm(m-1)}$ A_i denotes a Toeplitz matrix.

For the remainder of this section let $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$. Considering Examples **4.9.5** and **6.1.3**, the next theorem should not be very surprising.

7.1.6 **Theorem.** *$\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is translation invariant $\Leftrightarrow$ $\psi(\mathbf{t})$ is block Toeplitz with Toeplitz blocks.*

Proof: Suppose **t** is translation invariant on **X**. Since $\psi(\mathbf{t})$ is an $mn \times mn$ matrix, we can write $\psi(\mathbf{t})$ in terms of a block structured matrix

$$\psi(\mathbf{t}) = \begin{pmatrix} A_{00} & A_{01} & \cdots & A_{0,m-1} \\ A_{10} & A_{11} & \cdots & A_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m-1,0} & A_{m-1,1} & \cdots & A_{m-1,m-1} \end{pmatrix}, \qquad \text{(i)}$$

where each A_{ij} is an $n \times n$ matrix. Our first goal is to show that for each pair $i, j \in \mathbb{Z}_m$, A_{ij} is a Toeplitz matrix.

Let $A_{ij} = (a_{st})$, where $s, t \in \mathbb{Z}_n$. It follows from the definition of the isomorphism ψ that $a_{st} = \mathbf{t}_{(j,t)}(i, s)$. Fix *s* and *t*, and let *k* be an integer such that $s + k, t + k \in \mathbb{Z}_n$. Since **t** is translation invariant,

$$a_{st} = \mathbf{t}_{(j,t)}(i, s) = \mathbf{t}_{(j,t+k)}(i, s + k) = a_{s+k,t+k}\,,$$

which implies that A_{ij} is a Toeplitz matrix.

To show that $\psi(\mathbf{t})$ is block Toeplitz, let *k* be any integer such that $i + k, j + k \in \mathbb{Z}_m$. If $A_{ij} = (a_{st})$ and $A_{i+k,j+k} = (b_{st})$, then for $s, t \in \mathbb{Z}_n$ we have

$$a_{st} = \mathbf{t}_{(j,t)}(i, s) = \mathbf{t}_{(j+k,t)}(i + k, s) = b_{st}\,.$$

This shows that $A_{ij} = A_{i+k,j+k}$.

Conversely, suppose that $\psi(\mathbf{t})$ is block Toeplitz with Toeplitz blocks and written in terms of Eq. (i). Let $\mathbf{y} = (j,t)$, $\mathbf{x} = (i,s) \in \mathbb{Z}_m \times \mathbb{Z}_n$ and suppose that $\mathbf{z} = (k,l) \in \mathbb{Z} \times \mathbb{Z}$ such that $\mathbf{y} + \mathbf{z} \in \mathbb{Z}_m \times \mathbb{Z}_n$ and $\mathbf{x} + \mathbf{z} \in \mathbb{Z}_m \times \mathbb{Z}_n$. By definition of $\psi(\mathbf{t})$, $\mathbf{t}_{(j,t)}(i,s) = a_{st}$, where a_{st} is the (s,t) entry of A_{ij}.

Since $\psi(\mathbf{t})$ is block Toeplitz with Toeplitz blocks, $a_{st} = b_{s+l,t+l}$, where $b_{s+l,t+l}$ denotes the $(s+l, t+l)$ entry of $A_{i+k,j+k}$. Therefore,

$$\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = a_{st} = b_{s+l,t+l} = \mathbf{t}_{(j+k,t+l)}(i+k, s+l) = \mathbf{t}_{\mathbf{y}+\mathbf{z}}(\mathbf{x}+\mathbf{z}).$$

Q.E.D.

Since ψ is an isomorphism, a template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ has an inverse if and only if the corresponding matrix $\psi(\mathbf{t})$ has an inverse. The *inverse* of $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ (if it exists) will be denoted by $\mathbf{t}^{-1}$. Thus, if $\mathbf{t}$ has an inverse, then $\mathbf{t}^{-1} = \psi^{-1}\left(\psi(\mathbf{t})^{-1}\right)$. Obviously, if T is an $mn \times mn$ block Toeplitz matrix with Toeplitz blocks, then $\psi^{-1}(T)$ is an invariant template on $\mathbf{X}$. However, since the inverse *matrix* of a block Toeplitz matrix with Toeplitz blocks is not necessarily a Toeplitz matrix (see for example [4, 9]), Theorem **7.1.6** has the following implication:

7.1.7 Corollary. *The inverse of a translation invariant template is not necessarily translation invariant.*

7.2 Circulant Templates

Implementation of many translation invariant transformations often involve circular convolutions since circular convolutions can be computed using fast transform methods. Circulant templates, the topic of this section, are used to implement circular convolutions.

Unless otherwise specified, for the remainder of this section $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n \subset \mathbb{Z} \times \mathbb{Z}$. For $\mathbf{x} = (i,j) \in \mathbb{Z} \times \mathbb{Z} = \mathbb{Z}^2$ we define $\mathbf{x} mod(m,n) \equiv (i(mod m), j(mod n))$.

7.2.1 Definition. A function $\varphi : \mathbf{X} \to \mathbf{X}$ is called a *circulant translation* if and only if $\exists\ \mathbf{z} \in \mathbf{X}$ such that $\varphi(\mathbf{x}) = (\mathbf{x}+\mathbf{z}) mod(m,n)\ \ \forall \mathbf{x} \in \mathbf{X}$

Note that if $\mathbf{z} \in \mathbb{Z}^2$ and $\mathbf{z} \notin \mathbf{X}$, then $\mathbf{z} mod(m,n) \in \mathbf{X}$ and $(\mathbf{x}+\mathbf{z}) mod(m,n) = (\mathbf{x}+\mathbf{y}) mod(m,n)\ \ \forall \mathbf{x} \in \mathbf{X}$, where $\mathbf{y} = \mathbf{z} mod(m,n)$. Therefore, we shall sometimes define circulant translations in terms of $\mathbf{z} \in \mathbb{Z}^2 \backslash \mathbf{X}$.

7.2.2 Theorem. *If $\Psi(\mathbf{X}) = \{\varphi : \varphi\ is\ a\ circulant\ translation\ on\ \mathbf{X}\}$ and $(\mathbf{X}, +)$ is the additive group of vector addition $mod(m,n)$, then $(\Psi(\mathbf{X}), \circ)$ is isomorphic $(\mathbf{X}, +)$.*

Proof: For each $\mathbf{z} \in \mathbf{X}$, let $\varphi_{\mathbf{z}}$ denote the circulant translation defined by $\varphi_{\mathbf{z}}(\mathbf{x}) = (\mathbf{x}+\mathbf{z}) mod(m,n)\ \ \forall \mathbf{x} \in \mathbf{X}$. Note that $\varphi \in \Psi(\mathbf{X}) \Leftrightarrow \varphi = \varphi_{\mathbf{z}}$ for some $\mathbf{z} \in \mathbf{X}$. Thus,

if we define $h : \mathbf{X} \to \Psi(\mathbf{X})$ by $h(\mathbf{z}) = \varphi_{\mathbf{z}}$, then *h* is onto. Also, since $\varphi_{\mathbf{z}} \neq \varphi_{\mathbf{y}}$ whenever $\mathbf{z} \neq \mathbf{y}$, $h(\mathbf{z}) \neq h(\mathbf{y})$. Therefore, *h* is one-to-one.

By definition of *h*, $h(\mathbf{z}+\mathbf{y}) = \varphi_{\mathbf{z}+\mathbf{y}}$. But

$$\begin{aligned}\varphi_{\mathbf{z}+\mathbf{y}}(\mathbf{x}) &= [\mathbf{x}+(\mathbf{z}+\mathbf{y})]mod(m,n)\\ &= [(\mathbf{x}+\mathbf{y})mod(m,n)+\mathbf{z}]mod(m,n)\\ &= \varphi_{\mathbf{z}}[(\mathbf{x}+\mathbf{y})mod(m,n)]\\ &= \varphi_{\mathbf{z}}(\varphi_{\mathbf{y}}(\mathbf{x}))\\ &= (\varphi_{\mathbf{z}} \circ \varphi_{\mathbf{y}})(\mathbf{x})\end{aligned}$$

$\forall \mathbf{x} \in \mathbf{X}$. Therefore, $h(\mathbf{z}+\mathbf{y}) = \varphi_{\mathbf{z}} \circ \varphi_{\mathbf{y}}$.

Q.E.D.

7.2.3 **Definition.** A template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is said to be *circulant* if and only if $\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\varphi(\mathbf{y})}(\varphi(\mathbf{x}))$ $\forall \mathbf{x}, \mathbf{y} \in \mathbf{X}$ and $\forall \varphi \in \Psi(\mathbf{X})$. We denote the set of all $\mathbb{F}$–valued circulant templates on $\mathbf{X}$ by $C(\mathbb{F}, \mathbf{X})$.

Suppose $\mathbf{t} \in C(\mathbb{F}, \mathbf{X})$ and $\phi \in \Phi|_{\mathbf{X}}$. If $\mathbf{x}, \mathbf{y} \in domain(\phi)$, then there exists a point $\mathbf{z} \in \mathbb{Z}^2$ such that $\mathbf{x}+\mathbf{z} = \phi(\mathbf{x}) \in \mathbf{X}$ and $\mathbf{y}+\mathbf{z} = \phi(\mathbf{y}) \in \mathbf{X}$. Since $\mathbf{x}+\mathbf{z} \in \mathbf{X}$ and $\mathbf{y}+\mathbf{z} \in \mathbf{X}$, $(\mathbf{x}+\mathbf{z})mod(m,n) = \mathbf{x}+\mathbf{z}$ and $(\mathbf{y}+\mathbf{z})mod(m,n) = \mathbf{y}+\mathbf{z}$. Thus, using the *circulant* translation $\varphi_{\mathbf{z}} \in \Psi(\mathbf{X})$, we have that

$$\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\varphi_{\mathbf{z}}(\mathbf{y})}(\varphi_{\mathbf{z}}(\mathbf{x})) = \mathbf{t}_{\mathbf{y}+\mathbf{z}}(\mathbf{x}+\mathbf{z}) = \mathbf{t}_{\phi(\mathbf{y})}(\mathbf{x})\,.$$

This proves the following theorem:

7.2.4 **Theorem.** *Every circulant template on* $\mathbf{X}$ *is translation invariant on* $\mathbf{X}$.

7.2.5 **Definition.** An $n \times n$ matrix $C = (c_{ij})$ is said to be a *circulant matrix of order n* if and only if for every integer *k*,

$$c_{ij} = c_{(i+k)modn,(j+k)modn}\,.$$

It follows from this definition that if *C* is a circulant matrix of order *n*, then *C* is of form

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & \cdots & c_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{pmatrix}.$$

Thus, we can use the simpler notation $C = circ(c_0, c_1, \ldots, c_{n-1})$ to uniquely represent the matrix *C*.

7.2.6 **Definition.** An $mn \times mn$ matrix A is said to be *block circulant with circulant blocks* of type (m, n) if and only if A can be expressed as a block matrix

$$A = \begin{pmatrix} A_0 & A_1 & \cdots & A_{m-1} \\ A_{m-1} & A_0 & \cdots & A_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ A_1 & A_2 & \cdots & A_0 \end{pmatrix},$$

such that each $A_i = circ\big(a_{i_0}, a_{i_1}, \ldots, a_{i_{n-1}}\big)$.

We shall also use the notation $A = circ(A_0, A_1, \ldots, A_{m-1})$ to represent a block circulant matrix with circulant blocks. The next theorem shows that circulant templates deserve their name.

7.2.7 **Theorem.** *If* $\mathbf{t} \in C(\mathbb{F}, \mathbf{X})$*, then* $\psi(\mathbf{t})$ *is block circulant with circulant blocks.*

Proof: Since **t** is translation invariant (Theorem **7.2.2**), $\psi(\mathbf{t})$ is block Toeplitz with Toeplitz blocks (Theorem **7.1.6**)

Let

$$\psi(\mathbf{t}) = \begin{pmatrix} A_0 & A_1 & \cdots & A_{m-1} \\ A_{-1} & A_0 & \cdots & A_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{-(m-1)} & A_{-(m-2)} & \cdots & A_0 \end{pmatrix}.$$

We shall show that for each $l \in \mathbb{Z}_{\pm(m-1)}$, A_l is circulant.

Since $l \in \mathbb{Z}_{\pm(m-1)}$, A_l is of form

$$A_l = \begin{pmatrix} \mathbf{t}_{(j,0)}(i,0) & \mathbf{t}_{(j,1)}(i,0) & \cdots & \mathbf{t}_{(j,s)}(i,0) & \cdots & \mathbf{t}_{(j,n-1)}(i,0) \\ \mathbf{t}_{(j,0)}(i,1) & \mathbf{t}_{(j,1)}(i,1) & \cdots & \mathbf{t}_{(j,s)}(i,1) & \cdots & \mathbf{t}_{(j,n-1)}(i,1) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{t}_{(j,0)}(i,r) & \mathbf{t}_{(j,1)}(i,r) & \cdots & \mathbf{t}_{(j,s)}(i,r) & \cdots & \mathbf{t}_{(j,n-1)}(i,r) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{t}_{(j,0)}(i,n-1) & \mathbf{t}_{(j,1)}(i,n-1) & \cdots & \mathbf{t}_{(j,s)}(i,n-1) & \cdots & \mathbf{t}_{(j,n-1)}(i,n-1) \end{pmatrix},$$

where $l = j - i$ and $i, j \in \mathbb{Z}_m$.

Let $k \in \mathbb{Z}$, $\mathbf{z} = (0, k)$, and consider the circulant translation $\varphi_{\mathbf{z}} : \mathbf{X} \to \mathbf{X}$ defined by $\varphi_{\mathbf{z}}(\mathbf{x}) = (\mathbf{x} + \mathbf{z}) mod(m, n)$. If $\mathbf{x} = (x_1, x_2) \in \mathbf{X}$, then $\varphi_{\mathbf{z}}(\mathbf{x}) = (x_1, x_2 + k) mod(m, n) = (x_1, (x_2 + k) mod n)$. Let A_l be denoted by $A_l = (a_{rs})$. Since **t** is circulant, we have

$$a_{rs} = \mathbf{t}_{(j,s)}(i,r) = \mathbf{t}_{\varphi_{\mathbf{z}}(j,s)}(\varphi_{\mathbf{z}}(i,r)) = \mathbf{t}_{(j,(s+k) mod n)}(i,(r+k) mod n) = a_{(r+k) mod n,(s+k) mod n}\,.$$

Therefore, A_l is circulant

We now show that $\psi(\mathbf{t})$ is block circulant. Let $l \in \{1, 2, \ldots, m-1\}$ and $A_l = (a_{rs})$. Since $l > 0$ and $\psi(\mathbf{t})$ is block Toeplitz, we have that

$$A_l = \begin{pmatrix} \mathbf{t}_{(l,0)}(0,0) & \cdots & \mathbf{t}_{(l,n-1)}(0,0) \\ \vdots & \ddots & \vdots \\ \mathbf{t}_{(l,0)}(0,n-1) & \cdots & \mathbf{t}_{(l,n-1)}(0,n-1) \end{pmatrix}.$$

Thus, $a_{rs} = \mathbf{t}_{(l,s)}(0,r)$. We need to show that $A_l = A_{-(m-l)}$.

Let $A_{-(m-l)} = (b_{rs})$, $\mathbf{z} = (l,0)$, and define the circulant translation $\varphi_{\mathbf{z}} : \mathbf{X} \to \mathbf{X}$ by $\varphi_{\mathbf{z}}(\mathbf{x}) = (\mathbf{x} + \mathbf{z}) mod(m,n)$. Then

$$b_{rs} = \mathbf{t}_{(0,s)}(m-l,r) = \mathbf{t}_{\varphi_{\mathbf{z}}(0,s)}(\varphi_{\mathbf{z}}(m-l,r)) = \mathbf{t}_{(l,s)}(0,r) = a_{rs}\,.$$

Q.E.D.

The set $C(m,n)$ of all block circulant matrices with circulant blocks of type (m,n) forms a commutative ring with unity [6]. Since $\psi : C(\mathbb{F}, \mathbf{X}) \to C(m,n)$ is an isomorphism, Theorem **7.2.7** implies the following corollary:

7.2.8 Corollary. *$(C(\mathbb{F}, \mathbf{X}), \bigoplus, +)$ is a commutative ring with unity.*

As mentioned previously, circulant templates are used to implement circular convolutions. In fact, if $\mathbf{t} \in C(\mathbb{F}, \mathbf{X})$, then the mapping $\mathbf{a} \rightarrowtail \mathbf{a} \bigoplus \mathbf{t}$ is the *circular convolution* of the two-dimensional sequences $a_{ij} = \mathbf{a}(i,j)$ and $t_{ij} = \mathbf{t}_{(0,0)}[(-i,-j) mod(m,n)]$. These circular convolutions can be used for approximating translation invariant template operations [1, 14]. The simplest techniques, which are adequate for small convolutions (i.e., templates with small support), are either to pad the array with zeros or to ignore the boundary effects.

Circular templates are closely related to the discrete Fourier transform and the theory of fast convolutions. The next set of definitions and theorems help to establish this relationship.

If $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ and $\mathbf{f} \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$ denotes the two-dimensional Fourier template, then the *two-dimensional Fourier matrix of order* $m \otimes n$ is denoted by $F_{m \otimes n}$ and defined by $F_{m \otimes n} = \psi(\mathbf{f})$. We use the tensor notation $m \otimes n$ in order to avoid confusion with the one-dimensional Fourier matrix F_{mn} defined in Section **6.2**. Note that in general $F_{m \otimes n} \neq F_{mn}$. The relation between $F_{m \otimes n}$ and the one-dimensional Fourier transform matrix is given by the following equation:

$$F_{m \otimes n} = F_m \otimes F_n = (F_m \otimes I_n)(I_m \otimes F_n). \tag{7.2.1}$$

This relation corresponds to our previous observation (Eq. 6.1.22) that the two-dimensional Fourier transform can be computed in two successive applications of the one-dimensional Fourier transform.

For each $C = circ(c_0, c_1, \ldots, c_{n-1})$, let $f_C(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}$. Thus, if $P = circ(0, 1, 0, \ldots, 0)$ is of size $n \times n$, then $f_C(P) = c_0 P^0 + c_1 P + \cdots + c_{n-1} P^{n-1} = C$, where $P^0 \equiv I_n$. Furthermore, since

$$\sum_{j=0}^{n-1} \omega_n^{jk} = \begin{cases} n & \text{if } k \equiv 0 mod n \\ 0 & \text{otherwise,} \end{cases}$$

it is easy to show that $P = F_n \Omega F_n^*$, where $\Omega = diag\left(\omega_n^j\right)$ with $j = 0, 1, \ldots, n-1$. It now follows that

$$C = F_n \Lambda F_n^* , \tag{7.2.2}$$

where $\Lambda = diag\left(f_C\left(\omega_n^j\right)\right)\ j = 0, 1, \ldots, n-1$.

Equation 7.2.2 represents the matrix formulation of the circular convolution theorem. Similarly, for each block circulant matrix C with circulant blocks of type (m, n) we can define a polynomial $f_C(x, y)$ by

$$\begin{aligned} f_C(x,y) = c_{00} + c_{01}y + \cdots + c_{0,n-1}y^{n-1} + x\left(c_{10} + c_{11}y + \cdots + c_{1,n-1}y^{n-1}\right) + \cdots \\ \cdots + x^{m-1}\left(c_{m-1,0} + c_{m-1,1}y + \cdots + c_{m-1,n-1}y^{n-1}\right), \end{aligned} \tag{7.2.3}$$

such that

$$C = (F_m \otimes F_n) D (F_m \otimes F_n)^*, \tag{7.2.4}$$

where $D = diag\left(f_C\left(\omega_m^j, \omega_n^k\right)\right)$. Thus, circulant and block circulant matrices are diagonalizable by Fourier matrices. In the language of image algebra this says that if $\mathbf{t} \in C(\mathbb{F}, \mathbf{X})$ and $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$, then there exists a template $\mathbf{s} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ such that $S(\mathbf{s_y}) \subset \{\mathbf{y}\}\ \ \forall \mathbf{y} \in \mathbf{X}$ and

$$\mathbf{a} \oplus \mathbf{t} = ((\mathbf{a} \oplus \mathbf{f}) \oplus \mathbf{s}) \oplus \mathbf{f}^{-1}. \tag{7.2.5}$$

Note that the computation of $\mathbf{b} \oplus \mathbf{s}$ can be implemented using pointwise multiplication. Equation 7.2.5 shows that the FFT can be used to facilitate the computation of convolutions.

The Fourier template $\mathbf{f}$ induces an isomorphism $C(\mathbb{C}, \mathbf{X}) \rightarrow \mathbb{C}^{\mathbf{X}}$. This corresponds to the image algebra version of the convolution theorem.

7.2.9 Theorem. *The function $\eta : C(\mathbb{C}, \mathbf{X}) \rightarrow \mathbb{C}^{\mathbf{X}}$ defined by $\eta(\mathbf{t}) = \mathbf{t}_{(0,0)} \oplus \mathbf{f}$ is a ring isomorphism.*

Proof: If $\mathbf{s}, \mathbf{r} \in C(\mathbb{C}, \mathbf{X})$ such that $\mathbf{t} = \mathbf{s} + \mathbf{r}$, then $\mathbf{t}_{(0,0)} = \mathbf{s}_{(0,0)} + \mathbf{r}_{(0,0)}$. Using this and Eq. 5.1.1 we obtain

$$\eta(\mathbf{s} + \mathbf{r}) = \eta(\mathbf{t}) = \left(\mathbf{s}_{(0,0)} + \mathbf{r}_{(0,0)}\right) \oplus \mathbf{f} = \left(\mathbf{s}_{(0,0)} \oplus \mathbf{f}\right) + \left(\mathbf{r}_{(0,0)} \oplus \mathbf{f}\right) = \eta(\mathbf{s}) + \eta(\mathbf{r}) .$$

From the definition of the Fourier template we obtain

$$\eta(1) = 1_{(0,0)} \oplus \mathbf{f} = \mathbf{1} .$$

Next suppose that $\mathbf{t} = \mathbf{s} \oplus \mathbf{r}$, $\mathbf{z} = (i, j)$, $\mathbf{x} = (x, y) \in \mathbf{X}$, and $\varphi_{\mathbf{z}}(\mathbf{x}) = (\mathbf{x} - \mathbf{z}) mod(m, n)$. Since $C(\mathbb{C}, \mathbf{X})$ is a commutative ring, $\mathbf{s} \oplus \mathbf{r} = \mathbf{r} \oplus \mathbf{s}$. Hence,

$$\begin{aligned} \mathbf{t}_{(0,0)}(\mathbf{x}) &= \sum_{\mathbf{z} \in \mathbf{X}} \mathbf{s}_{(0,0)}(\mathbf{z}) \cdot \mathbf{r}_{\mathbf{z}}(\mathbf{x}) = \sum_{\mathbf{z} \in \mathbf{X}} \mathbf{s}_{(0,0)}(\mathbf{z}) \cdot \mathbf{r}_{\varphi_{\mathbf{z}}(\mathbf{z})}(\varphi_{\mathbf{z}}(\mathbf{x})) \\ &= \sum_{(i,j) \in \mathbf{X}} \mathbf{s}_{(0,0)}(i, j) \cdot \mathbf{r}_{(0,0)}((x - i) mod m, (y - j) mod n) . \end{aligned}$$

Therefore, $\mathbf{t}_{(0,0)}(\mathbf{x})$ can be represented as the cyclic convolution of the two-dimensional sequences $\{\mathbf{s}_{(0,0)}(\mathbf{z})\}_{\mathbf{z}\in\mathbf{X}}$ and $\{\mathbf{r}_{(0,0)}(\mathbf{z})\}_{\mathbf{z}\in\mathbf{X}}$ (see also Section **3.6** and Example **3.10.3**(ii)). By the Convolution Theorem we now have

$$\mathbf{t}_{(0,0)} \oplus \mathbf{f} = (\mathbf{s}_{(0,0)} \oplus \mathbf{f}) \cdot (\mathbf{r}_{(0,0)} \oplus \mathbf{f})$$

and, hence,

$$\eta(\mathbf{s} \oplus \mathbf{r}) = \eta(\mathbf{s}) \cdot \eta(\mathbf{r}) . \tag{7.2.6}$$

Thus, η preserves the ring operations.

To see that η is onto, let $\mathbf{b} \in \mathbb{C}^{\mathbf{X}}$. Set $\mathbf{a} = \mathbf{b} \oplus \mathbf{f}^{-1}$ and define $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ by setting $\mathbf{t}_{(0,0)}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \ \ \forall \mathbf{x} \in \mathbf{X}$ and extending $\mathbf{t}$ circularly by defining

$$\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{(0,0)}[(\mathbf{x} - \mathbf{y}) mod(m, n)] \quad \forall \mathbf{y} \in \mathbf{X} .$$

Then $\eta(\mathbf{t}) = \mathbf{t}_{(0,0)} \oplus \mathbf{f} = \mathbf{a} \oplus \mathbf{f} = (\mathbf{b} \oplus \mathbf{f}^{-1}) \oplus \mathbf{f} = \mathbf{b}$.

To show that η is one-to-one, suppose that $\eta(\mathbf{s}) = \eta(\mathbf{r})$. Then $\mathbf{s}_{(0,0)} \oplus \mathbf{f} = \mathbf{r}_{(0,0)} \oplus \mathbf{f}$ and, since $\mathbf{f}$ is invertible, $\mathbf{s}_{(0,0)} = \mathbf{r}_{(0,0)}$. However, circulant templates are determined by their values at one point. Therefore, $\mathbf{s} = \mathbf{r}$.

Q.E.D.

One of the most useful properties of the Fourier transform is that it converts convolutions into pointwise multiplications, a fact which is expressed by Eq. 7.2.6. Because of this, local algorithms for computing convolutions can be derived by deriving local algorithms for computing discrete Fourier transforms.

For $\mathbf{t} \in (\mathbb{F}^{\mathbf{X}})^{\mathbf{X}}$, define $\mathbf{a}(\mathbf{t}) \in \mathbb{F}^{\mathbf{X}}$ by $\mathbf{a}(\mathbf{t}) = \Sigma\mathbf{t}$. Note that if $\mathbf{s} \in (\mathbb{F}^{\mathbf{X}})^{\mathbf{X}}$ is a one-point template with $S(\mathbf{s}_{\mathbf{y}}) = \{\mathbf{y}\} \ \ \forall \mathbf{y} \in \mathbf{X}$, then $\mathbf{a}(\mathbf{s})(\mathbf{x}) = \mathbf{s}_{\mathbf{x}}(\mathbf{x})$ and $\mathbf{b} \oplus \mathbf{s} = \mathbf{a}(\mathbf{s}) \cdot \mathbf{b} \ \ \forall \mathbf{b} \in \mathbb{F}^{\mathbf{X}}$. In addition, $\psi(\mathbf{s})$ is a diagonal template.

One-point templates and invertible templates provide a structure which is essentially the same as the ring $C(\mathbb{C}, \mathbf{X})$. Suppose $\mathbf{t} \in (\mathbb{C}^{\mathbf{X}})^{\mathbf{X}}$ is invertible. Define $D(\mathbf{t}, \mathbf{X}) = \left\{\mathbf{t}^{-1} \oplus \mathbf{s} \oplus \mathbf{t} \ : \ \mathbf{s} \in (\mathbb{C}^{\mathbf{X}})^{\mathbf{X}} \text{ is a one-point template}\right\}$. The structure of $D(\mathbf{t}, \mathbf{X})$ is revealed by the following theorem:

7.2.10 Theorem. *$D(\mathbf{t}, \mathbf{X})$ is a commutative ring which is isomorphic to $\mathbb{C}^{\mathbf{X}}$.*

Proof: The proof, which proceeds in two steps, is routine and we leave the details to the reader. The first step consists of showing that $\psi(D(\mathbf{t}, \mathbf{X})) = \left\{\psi(\mathbf{t})^{-1} \cdot S \cdot \psi(\mathbf{t}) \ : \ S \text{ is a diagonal matrix}\right\}$ is a commutative ring (this is trivial), and that $\psi : D(\mathbf{t}, \mathbf{X}) \to \psi(D(\mathbf{t}, \mathbf{X}))$ is an isomorphism. This shows that $D(\mathbf{t}, \mathbf{X})$ is a commutative ring.

The second step consists of verifying that the function $\zeta : D(\mathbf{t}, \mathbf{X}) \rightarrow \mathbb{C}^{\mathbf{X}}$ defined by $\zeta(\mathbf{u}) = \mathbf{a}(\mathbf{r})$, where $\mathbf{r} = \mathbf{t} \oplus \mathbf{u} \oplus \mathbf{t}^{-1}$, is an isomorphism. This is also straight forward. For suppose $\mathbf{u}, \mathbf{v} \in D(\mathbf{t}, \mathbf{X})$ with $\mathbf{u} = \mathbf{t}^{-1} \oplus \mathbf{s}_1 \oplus \mathbf{t}$ and $\mathbf{v} = \mathbf{t}^{-1} \oplus \mathbf{s}_2 \oplus \mathbf{t}$. If $\mathbf{u}_1 = \mathbf{t} \oplus \mathbf{u} \oplus \mathbf{t}^{-1}$ and $\mathbf{v}_1 = \mathbf{t} \oplus \mathbf{v} \oplus \mathbf{t}^{-1}$, then $\mathbf{u}_1 = \mathbf{t} \oplus \big(\mathbf{t}^{-1} \oplus \mathbf{s}_1 \oplus \mathbf{t}\big) \oplus \mathbf{t}^{-1} = \mathbf{s}_1$ and, similarly, $\mathbf{v}_1 = \mathbf{s}_2$. Thus,

$$\begin{aligned}
\zeta(\mathbf{u}) + \zeta(\mathbf{r}) &= \mathbf{a}(\mathbf{u}_1) + \mathbf{a}(\mathbf{v}_1) \\
&= \Sigma\mathbf{u}_1 + \Sigma\mathbf{v}_1 \\
&= \Sigma\mathbf{s}_1 + \Sigma\mathbf{s}_2 \\
&= \Sigma(\mathbf{s}_1 + \mathbf{s}_2) \\
&= \Sigma\mathbf{t} \oplus \big[\big(\mathbf{t}^{-1} \oplus \mathbf{s}_1 \oplus \mathbf{t}\big) + \big(\mathbf{t}^{-1} \oplus \mathbf{s}_2 \oplus \mathbf{t}\big)\big] \oplus \mathbf{t}^{-1} \\
&= \Sigma\mathbf{t} \oplus (\mathbf{u} + \mathbf{v}) \oplus \mathbf{t}^{-1} \\
&= \zeta(\mathbf{u} + \mathbf{r})\,.
\end{aligned}$$

Therefore, $\zeta(\mathbf{u}) + \zeta(\mathbf{r}) = \zeta(\mathbf{u} + \mathbf{r})$. Note that we used Eq. 5.1.1 to establish this equality. In a similar fashion one can show that $\zeta(\mathbf{u}) \cdot \zeta(\mathbf{r}) = \zeta(\mathbf{u} \oplus \mathbf{r})$.

The verification that ζ is one-to-one and onto is just as routine.

Q.E.D.

Thus, if methods can be found for implementing invertible transforms locally, then these methods can be used to implement a larger class of linear transforms locally.

7.3 Circulant Templates and Polynomials

In the preceding two sections we defined translation invariant and circulant templates and outlined their relationship to the matrix algebra associated with the discrete Fourier transform. In this section we describe a family of relationships between circulant templates and the quotient rings of polynomial rings. These relationships show that the problem of finding decompositions of circulant templates is equivalent to factoring multivariate polynomials. For separable circulant templates, the problem reduces to the single variable case and is, therefore, equivalent to finding roots of polynomials in one variable. This is analogous to the polynomial factorization methodologies developed in Chapter **5**. However, the underlying techniques are different. For circulant templates we use shifts and the Fundamental Theorem of Algebra in order to obtain minimal local decompositions.

Throughout this section, **x**, **y**, and **z** will denote elements of $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$, $\mathbf{0} = (0, 0)$, *x* and *y* will denote indeterminates, and $\mathbb{C}[x, y]$ the ring of polynomials in two variables with coefficients in $\mathbb{C}$. We let $\mathbb{C}[x, y]/(x^m - 1, y^n - 1)$ denote the quotient ring of polynomials $mod(x^m - 1)$ and $mod(y^n - 1)$ (see Section **3.6**). The *circulant von Neumann configuration* is the function $N : \mathbf{X} \rightarrow 2^{\mathbf{X}}$ defined by

$$N(i, j) = \{(i, j), ((i \pm 1) mod m, j), (i, (j \pm 1) mod n)\}.$$

In this section, the term "local" shall mean local with respect to *N*.

7.3.1 **Definition.** For each $\mathbf{z} \in \mathbf{X}$, define $\theta_{\mathbf{z}} : C(\mathbb{C}, \mathbf{X}) \to \mathbb{C}[x, y]/(x^m - 1, y^n - 1)$ by $\theta_{\mathbf{z}}(\mathbf{t}) = p_{\mathbf{t}_{\mathbf{z}}}$, where

$$p_{\mathbf{t}_{\mathbf{z}}}(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{t}_{\mathbf{z}}(i, j) x^i y^j .$$

$\theta_{\mathbf{z}}(\mathbf{t})$ or, equivalently, $p_{\mathbf{t}_{\mathbf{z}}}$ is called the *polynomial representative* of **t**.

Note that by defining $c_{ij} \equiv \mathbf{t}_{\mathbf{z}}(i, j)$ we obtain

$$\begin{aligned} p_{\mathbf{t}_{\mathbf{z}}}(x, y) = c_{00} + c_{01} y + \cdots + c_{0,n-1} y^{n-1} + x\left(c_{00} + c_{01} y + \cdots + c_{0,n-1} y^{n-1}\right) + \cdots \\ \cdots + x^{m-1}\left(c_{m-1,0} + c_{m-1,1} y + \cdots + c_{m-1,n-1} y^{n-1}\right) \end{aligned} \tag{7.3.1}$$

which is identical to Eq. 7.2.3

7.3.2 **Example:** Let $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ be given by

$\mathbf{t} =$

1	2	3
2	4	6
3	6	9

Then

$$\theta_{(0,0)}(\mathbf{t}) = 4 + 6y + 2y^{n-1} + 6x + 9xy + 3xy^{n-1} + 2x^{m-1} + 3x^{m-1}y + x^{m-1}y^{n-1},$$

while

$$\theta_{(1,1)}(\mathbf{t}) = 1 + 2y + 3y^2 + 2x + 4xy + 6xy^2 + 3x^2 + 6x^2y + 9x^2y^2.$$

The following properties are easy to ascertain:

7.3.3 **Properties of θ_0.**

1. If $\theta_{(0,0)}(\mathbf{t}) = x + a$ or $\theta_{(0,0)}(\mathbf{t}) = y + a$, then **t** is local.
2. If $\theta_{(0,0)}(\mathbf{t}) = a_0 + a_1 x + a_2 x^2$ or $\theta_{(0,0)}(\mathbf{t}) = a_0 + a_1 y + a_2 y^2$, then **t** is local.
3. If $\theta_{(0,0)}(\mathbf{t}) = x^i y^j$, then the circulant transform $\mathbf{a} \mapsto \mathbf{a} \bigoplus \mathbf{t}$ shifts all pixel values *i* units vertically and *j* units horizontally.

7.3.4 **Theorem.** *If* $\mathbf{s}, \mathbf{r} \in C(\mathbb{C}, \mathbf{X})$, *then* $\theta_0(\mathbf{s} \bigoplus \mathbf{r}) = \theta_0(\mathbf{r}) \cdot \theta_0(\mathbf{s})$.

Proof: For $\mathbf{x} \in \mathbf{X}$, let $\varphi_{\mathbf{x}}$ denote the circulant translation defined by $\varphi_{\mathbf{x}}(\mathbf{z}) = (\mathbf{z} - \mathbf{x}) mod(m, n)$. If $\mathbf{t} = \mathbf{s} \bigoplus \mathbf{r}$, then $\mathbf{t}_0(\mathbf{z}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{s}_{\mathbf{x}}(\mathbf{z}) \cdot \mathbf{r}_0(\mathbf{x})$. Since **s** is circulant, $\mathbf{s}_{\mathbf{x}}(\mathbf{z}) = \mathbf{s}_{\varphi_{\mathbf{x}}(\mathbf{x})}(\varphi_{\mathbf{x}}(\mathbf{z})) = \mathbf{s}_0((\mathbf{z} - \mathbf{x}) mod(m, n))$. Thus,

$$\mathbf{t}_0(\mathbf{z}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{r}_0(\mathbf{x}) \cdot \mathbf{s}_0((\mathbf{z} - \mathbf{x}) mod(m, n)) . \tag{I}$$

By definition of polynomial products in $\mathbb{C}[x,y]/(x^m-1, y^n-1)$ (see Section **3.6**), the numbers $\mathbf{t}_0(\mathbf{z})$, for $\mathbf{z} \in \mathbf{X}$, given by Eq. (I) are exactly the coefficients of the polynomial product $p_{\mathbf{r}_0}(x,y) \cdot p_{\mathbf{s}_0}(x,y)$.

Q.E.D.

7.3.5 **Theorem.** *For every* $\mathbf{z} = (k,l) \in \mathbf{X}$, $\theta_{\mathbf{z}}$ *is one-to-one and onto. Moreover, if* $\mathbf{s}, \mathbf{r} \in C(\mathbb{C}, \mathbf{X})$, *then*

1. $\theta_{\mathbf{z}}(\mathbf{s}+\mathbf{r}) = \theta_{\mathbf{z}}(\mathbf{s}) + \theta_{\mathbf{z}}(\mathbf{r})$
2. $\theta_{\mathbf{z}}(\mathbf{r}) = x^k y^l \theta_{\mathbf{0}}(\mathbf{r})$
3. $x^k y^l \theta_{\mathbf{z}}(\mathbf{s} \bigoplus \mathbf{r}) = \theta_{\mathbf{z}}(\mathbf{r}) \cdot \theta_{\mathbf{z}}(\mathbf{s})$.

Proof: It is not difficult to check that $\theta_{\mathbf{z}}$ is one-to-one and onto, and we leave the details to the reader. Similarly, Eq. 1 follows from the fact that addition of templates is defined pointwise.

To prove Eq. 2, let $\varphi_{\mathbf{z}}$ be the circulant translation defined by $\varphi_{\mathbf{z}}(\mathbf{x}) = (\mathbf{x}+\mathbf{z}) mod(m,n)$, and note that $\varphi_{\mathbf{z}}(\mathbf{0}) = \mathbf{z}$. Now, if $\mathbf{r} \in C(\mathbb{C}, \mathbf{X})$, then since $x^m = 1 = y^n$, we have that

$$\begin{aligned}
\theta_{\mathbf{z}}(\mathbf{r}) &= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{r}_{\mathbf{z}}(i,j) x^i y^j \\
&= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{r}_{\mathbf{z}}((i+k) mod m, (j+l) mod n) x^{i+k} y^{j+l} \\
&= x^k y^l \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{r}_{\mathbf{0}}(i,j) x^i y^j \\
&= x^k y^l \cdot p_{\mathbf{t}_0}(x,y) = x^k y^l \theta_{\mathbf{0}}(\mathbf{r}) \,.
\end{aligned}$$

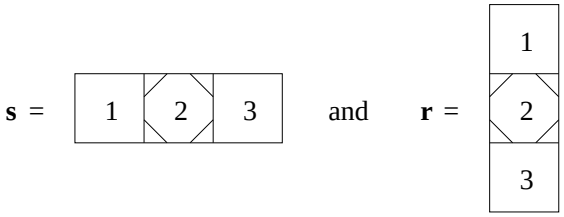
To prove Eq. 3, observe that since $\theta_{\mathbf{0}}(\mathbf{s} \bigoplus \mathbf{r}) = \theta_{\mathbf{0}}(\mathbf{r}) \cdot \theta_{\mathbf{0}}(\mathbf{s})$ and $\theta_{\mathbf{z}}(\mathbf{t}) = x^k y^l \theta_{\mathbf{0}}(\mathbf{t})$, we have that

$$\begin{aligned}
\theta_{\mathbf{z}}(\mathbf{s} \bigoplus \mathbf{r}) &= x^k y^l \theta_{\mathbf{0}}(\mathbf{s} \bigoplus \mathbf{r}) \\
&= x^k y^l \theta_{\mathbf{0}}(\mathbf{r}) \cdot \theta_{\mathbf{0}}(\mathbf{s}) \\
&= x^k y^l \Big[x^{m-k} y^{n-l} \theta_{\mathbf{z}}(\mathbf{r}) \cdot x^{m-k} y^{n-l} \theta_{\mathbf{z}}(\mathbf{s}) \Big] \\
&= x^{m-k} y^{n-l} \theta_{\mathbf{z}}(\mathbf{r}) \cdot \theta_{\mathbf{z}}(\mathbf{s}) \,.
\end{aligned}$$

The desired result now follows by multiplying the equation by $x^k y^l$.

Q.E.D.

7.3.6 **Example:** If **t** denotes the circulant template from Example **7.3.2**, then $\theta_{(1,1)}(\mathbf{t}) = xy\theta_{(0,0)}(\mathbf{t})$. Also, $\mathbf{t} = \mathbf{s} \oplus \mathbf{r}$, where



Thus, $\theta_{(1,1)}(\mathbf{s}) = x + 2xy + 3xy^2$ and $\theta_{(1,1)}(\mathbf{r}) = y + 2xy + 3x^2y$. Simple polynomial multiplication shows that

$$\theta_{(1,1)}(\mathbf{r}) \cdot \theta_{(1,1)}(\mathbf{s}) = xy\theta_{(1,1)}(\mathbf{t}) = xy\theta_{(1,1)}(\mathbf{s} \oplus \mathbf{r}) .$$

The following facts are direct consequences of Theorems **7.3.4** and **7.3.5**.

7.3.7 **Corollary.** *θ_0 is an isomorphism.*

7.3.8 **Corollary.** *Let $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$, $\mathbf{y} = (i, j)$, and $\mathbf{z} = (k, l)$. If $p_{\mathbf{t_y}}(x, y) = p_1(x, y) \cdot p_2(x, y)$, then $p_{\mathbf{t_z}}(x, y) = q_1(x, y) \cdot q_2(x, y)$, where $q_1(x, y) = x^{k-i}y^{l-j}p_1(x, y)$ and $q_2(x, y) = p_2(x, y)$.*

7.3.9 **Corollary.** *If $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$, $\mathbf{y} = (i, j)$, and $p_{\mathbf{t_y}}(x, y) = p(x, y) \cdot q(x, y)$, then $\mathbf{t} = \mathbf{t}_1 \oplus \mathbf{t}_2 \oplus \mathbf{s}$, where $\mathbf{t}_1 = \theta_{\mathbf{y}}^{-1}(p(x, y))$, $\mathbf{t}_2 = \theta_{\mathbf{y}}^{-1}(q(x, y))$, and **s** represents a circular shift $m - i$ units horizontally and $n - j$ units vertically. Moreover, **s** can be replaced by a template $\tilde{\mathbf{s}}$ which represents a circular shift i units horizontally and j units vertically.*

The collection $\{\theta_{\mathbf{z}} : \mathbf{z} \in \mathbf{X}\}$ constitutes a class of mappings between image algebra and polynomial algebra that are almost isomorphisms. For $\mathbf{z} \neq \mathbf{0}$, the operation of multiplication is not preserved (Theorem **7.3.5** Eq. 3). However, for $\mathbf{z} \neq \mathbf{0}$, they differ from the isomorphism $\theta_{\mathbf{0}}$ only by shifts. Thus, if any polynomial representative $p_{\mathbf{t_z}}$ of a template **t** can be factored, then the template **t** can also be factored. It follows that the template decomposition problem for circulant templates is equivalent to the problem of factoring multivariable polynomials. We will use this fact in order to show how any separable circulant template can be implemented locally with respect to the von Neumann configuration. Since the von Neumann restriction simulates a nearest neighbor mesh-connected array of $m \times n$ processors, the methods we develop can be used on such machines. Additionally, we shall provide upper bounds on the number of parallel steps required.

7.3.10 **Definition.** If $p(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} a_{ij}x^i y^j$, then

1. $deg(p(x, y)) \equiv max\{i + j : a_{ij} \neq 0\}$,
2. $deg_x(p(x, y)) \equiv max\{i : a_{ij} \neq 0\}$, and
3. $deg_y(p(x, y)) \equiv max\{j : a_{ij} \neq 0\}$.

7.3.11 **Definition.** If $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ and $deg\big(p_{\mathbf{t_y}}(x,y)\big) \leq deg(p_{\mathbf{t_z}}(x,y))\ \forall \mathbf{z} \in \mathbf{X}$, then we say that **y** is a *minimal point* for **t**.

Since θ_0 is an isomorphism, $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ is separable if and only if there exists polynomials *p* and *q* such that $p_{\mathbf{t_0}}(x,y) = p(x) \cdot q(y)$. It follows from the preceding corollaries that if **t** is separable, then for every $\mathbf{z} \in \mathbf{X}$ there exists polynomials $p_{\mathbf{z}}$ and $q_{\mathbf{z}}$ such that $p_{\mathbf{t_z}}(x,y) = p_{\mathbf{z}}(x) \cdot q_{\mathbf{z}}(y)$. We shall use this observation to provide a systematic method for factoring any separable circulant transform into local transforms with respect to 4–connected processor arrays and give upper bounds for the number of parallel steps required.

7.3.12 **Theorem.** *Suppose* $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ *is separable and* $\mathbf{y} = (k,l)$ *is a minimal point for* **t**. *Additionally, let* $i = deg_x\big(p_{\mathbf{t_y}}(x,y)\big)$ *and* $j = deg_y\big(p_{\mathbf{t_y}}(x,y)\big)$. *If* $\alpha = min\{k,\ m-k\}$ *and* $\beta = min\{l,\ n-l\}$, *then the circulant transform* $\mathbf{a} \mapsto \mathbf{a} \bigoplus \mathbf{t}$ *can be computed in at most* $\alpha + \beta + i + j + 1$ *local parallel steps. Moreover,*

1. $\alpha + \beta$ *of these steps consists of horizontal or vertical shifts of the entire array by one location,*
2. $i + j$ *of these steps consists of at most one addition and one multiplication (possibly complex) per pixel, and*
3. *one of these steps consists of at most one multiplication per pixel.*

Proof: Since **t** is separable, there exists polynomials *p* and *q* such that $p_{\mathbf{t_y}}(x,y) = p(x) \cdot q(y)$. By the Fundamental Theorem of Algebra

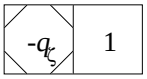
$$p(x) = a(x-p_1)(x-p_2)\cdots(x-p_i) \text{ and } q(y) = b(y-q_1)(y-q_2)\cdots(y-q_j),$$

where $a,\ b,\ p_1,\ p_2,\ \ldots,\ p_i,\ q_1,\ \ldots,\ q_j \in \mathbb{C}$.

Define circulant templates $\mathbf{p},\ \mathbf{q},\ \mathbf{p}_1,\ \ldots,\ \mathbf{p}_i,\ \mathbf{q}_1,\ \ldots,\ \mathbf{q}_j$ such that $\theta_0(\mathbf{p}) = p(x),\ \theta_0(\mathbf{q}) = q(y),\ \theta_0(\mathbf{p}_\zeta) = (x - p_\zeta)$ for $\zeta = 1,\ 2,\ \ldots,\ i$ and $\theta_0(\mathbf{q}_\zeta) = (y - q_\zeta)$ for $\zeta = 1,\ 2,\ \ldots,\ j$. By property **7.3.3**(1), each of the templates $\mathbf{p}_1,\ \ldots,\ \mathbf{p}_i,\ \mathbf{q}_1,\ \ldots,\ \mathbf{q}_j$ is a local template. In fact, it should be obvious that these templates have the following geometric representations:

$\mathbf{p}_\zeta$ =

$\mathbf{q}_\zeta$ =



By our choice of **p** and **q**, $\theta_{\mathbf{y}}(\mathbf{t}) = \theta_0(\mathbf{p}) \cdot \theta_0(\mathbf{q})$. Furthermore, by Theorem **7.3.4**, $\theta_0(\mathbf{p}) = a\left(\prod_{\zeta=1}^{i} \theta_0(\mathbf{p}_\zeta)\right)$, which implies that $\mathbf{p} = a\left(\bigoplus_{\zeta=1}^{i} \mathbf{p}_\zeta\right)$. Similarly, $\mathbf{q} = b\left(\bigoplus_{\zeta=1}^{j} \mathbf{q}_\zeta\right)$. Hence, by Corollary **7.3.9**,

$$\mathbf{t} = ab\left[\left(\bigoplus_{\zeta=1}^{i} \mathbf{p}_\zeta\right) \oplus \left(\bigoplus_{\zeta=1}^{j} \mathbf{q}_\zeta\right) \oplus \mathbf{s}\right].$$

This equation expresses $\mathbf{t}$ as a product of local templates. The number of steps required to implement the circular convolution transform $\mathbf{a} \rightarrowtail \mathbf{a} \bigoplus \mathbf{t}$ is also given by this equation; namely, $i + j + 1$ multiplication steps and $\alpha + \beta$ shifts.

Q.E.D.

To provide a specific example, suppose that $\mathbf{X}$ represents a 512×512 array and $\mathbf{t}$ a rectangular 30×30 separable circulant template. Then $\mathbf{y} = (14, 14)$ and $\alpha = \beta = 14$. Hence, if $\mathbf{a} \in \mathbb{C}^{\mathbf{X}}$, then $\mathbf{a} \rightarrowtail \mathbf{a} \bigoplus \mathbf{t}$ can be computed locally in 60 parallel steps consisting of at most one multiplication and addition per pixel, one step consisting of one multiplication per pixel, and a total of 28 unit vertical or horizontal circular shifts. Since the computation in its original form required 900 multiplications per pixel, it is clear that the decomposition is far more efficient with respect to the number of arithmetic operations as well as parallelism. Nontrivial examples of such templates are the discretization of the Marr-Hildreth edge operators [16].

Another systematic method for decomposing separable circulant transforms into local transforms is given by the next theorem. This theorem also avoids complex multiplications.

7.3.13 Theorem. *Suppose* $\mathbf{t} \in C(\mathbb{R}, \mathbf{X})$ *is separable and* $p_{\mathbf{t_z}}(x, y) = p_{\mathbf{z}}(x)$ *is a polynomial in one variable* $\forall \mathbf{z} \in \mathbf{X}$. *Let* $(l, 0)$ *be a minimal point for* $\mathbf{t}$.

1. *If* $deg\big(p_{(l,0)}(x)\big) = 2k$ *for some* $k \in \mathbb{Z}^+$ *and* $\alpha = min\{m - |k - l|, |k - l|\}$, *then the transform* $\mathbf{a} \rightarrowtail \mathbf{a} \bigoplus \mathbf{t}$ *can be computed in* $\alpha + k + 1$ *local parallel steps. Furthermore,* α *of these steps consist of vertical, circular unit shifts of the entire array, k of these steps consist of at most two real multiplications and additions per pixel, and one of step consists of at most one real multiplication per pixel.*
2. *If* $deg\big(p_{(l,0)}(x)\big) = 2k + 1$ *for some* $k \in \mathbb{Z}^+$, *then the conclusion of part* (1) *holds with k replaced by* $k + 1$.

Proof: We can write $\theta_{(l,0)}(\mathbf{t})$ as $\theta_{(l,0)}(\mathbf{t}) = c \prod_{i=1}^{k} p_i(x)$, where each $p_i(x)$ is a monic quadratic polynomial with real coefficients and $c \in \mathbb{R}$. For each $i \in \mathbb{Z}_k^+$, define $\mathbf{p}_i = \theta_{(1,0)}^{-1}(p_i(x))$. By property **7.3.3**(2), each $\mathbf{p}_i$ is a local template. Furthermore,

$$\theta_{\mathbf{0}}(\mathbf{t}) = x^{-l}\theta_{(l,0)}(\mathbf{t}) = cx^{-l} \prod_{i=1}^{k} \theta_{(1,0)}(\mathbf{p}_i) = cx^{k-l} \prod_{i=1}^{k} \theta_{\mathbf{0}}(\mathbf{p}_i)\,.$$

Therefore, $\mathbf{t} = c\left(\bigoplus_{i=1}^{k} \mathbf{p}_i\right) \bigoplus \mathbf{s}$, where $\mathbf{s}$ is a shift template.

If $k > l$, then we can choose $\mathbf{s}$ to be a shift of $\alpha = min\{k - l,\ m - (k - l)\}$ units. If $k \leq l$, then we can let $\mathbf{s}$ be a shift of $\alpha = min\{l - k,\ m - (l - k)\}$ units. In either case, the shift can be executed in α steps.

If the second hypothesis of our theorem holds, write $\theta_{(l,0)}(\mathbf{t})$ as $\theta_{(l,0)}(\mathbf{t}) = c\left[\prod_{i=1}^{k} p_i(x)\right] \cdot p(x)$, where each $p_i(x)$ is a monic quadratic polynomial with real coefficients, $c \in \mathbb{R}$, and $p(x)$ is

a monic linear polynomial with real coefficients. Define local templates $\mathbf{p}_i$ as before and set $\mathbf{p} = \theta_0^{-1}(p(x))$. Then

$$\theta_0(\mathbf{t}) = cx^{-l}\left[\prod_{i=1}^{k}\theta_{(1,0)}(\mathbf{p}_i)\right]\cdot\theta_0(\mathbf{p}) = cx^{k-l}\left[\prod_{i=1}^{k}\theta_0(\mathbf{p}_i)\right]\cdot\theta_0(\mathbf{p})\,.$$

The remainder of the proof is now identical to the latter part of the proof of part 1.

Q.E.D.

Obviously, an analogous theorem holds for $\mathbf{t} \in C(\mathbb{R}, \mathbf{X})$ with $p_{\mathbf{t_z}}(x, y) = q_{\mathbf{z}}(y)\ \forall \mathbf{z} \in \mathbf{X}$.

If $r \in \mathbb{R}$, let $\lceil r \rceil$ denote the smallest integer greater than or equal to *r*. The following is an easy consequence of Theorem **7.3.13**.

7.3.14 Corollary. *Suppose* $\mathbf{t}$ *is a separable circulant template and* $\mathbf{z} = (k, l)$ *is a minimal point for* $\mathbf{t}$. *If* $i = \left\lceil deg_x\left(\frac{p_{\mathbf{t_z}}(x,y)}{2}\right)\right\rceil$, $j = \left\lceil deg_y\left(\frac{p_{\mathbf{t_z}}(x,y)}{2}\right)\right\rceil$, $\alpha = min\{m - |i-k|, |i-k|\}$, *and* $\beta = min\{n - |j-l|, |j-l|\}$, *then the circulant transform* $\mathbf{a} \rightarrowtail \mathbf{a} \bigoplus \mathbf{t}$ *can be computed in* $\alpha + \beta + i + j + 1$ *local parallel steps. Moreover,* $\alpha + \beta$ *of these steps consist of vertical and horizontal circular unit shifts of the entire array,* $i + j$ *of these steps consist of at most two multiplications and additions per pixel, and one of step consists of at most one multiplication per pixel.*

The templates corresponding to the quadratic polynomials that result from the decomposition of $\mathbf{t}$ as given by Theorem **7.3.13** or its corollary are depicted in Figure 7.3.1.

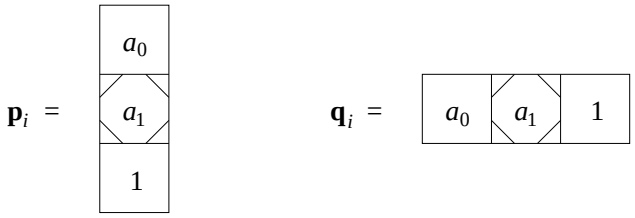


Figure 7.3.1 The quadratic templates resulting from the decomposition technique used in Theorem **7.3.13** and its corollary.

The rationale for establishing the relationship between circulant templates and the quotient ring of polynomials in two variables $mod(x^m - 1,\ y^n - 1)$ is the same as that for establishing the relationship between templates and polynomials discussed in Chapter **5**; the relationship provides a method for decomposing separable circulant templates by factoring (or finding the roots) of corresponding polynomials in one variable. Although finding roots of polynomials can be a numerically unstable procedure, many computerized techniques have been developed that are capable of factoring polynomials exactly [5, 10, 13]. These techniques can be applied in our methodology for developing parallel algorithms for computing convolutions.

7.4 G-Templates

In this section we generalize the notion of a circulant template by introducing the concept of *G*-templates. The concept of *G*-templates was first introduced by P. Gader [7]. Gader's generalization of circulant templates grew out of an awareness of the possible uses of Cayley networks as models for parallel computer architectures and the importance of translation invariant transformations in parallel processing; these generalizations were also influenced by observations that the discrete Fourier transform is related to the theory of group representations [2].

G-templates are translation invariant with respect to Cayley networks, which are networks whose underlying graphs are the group graphs of some finite group *G*. Since Cayley networks have been investigated as possible models for parallel computer architectures [15], *G*–templates have potential applications in the field of parallel image processing.

Roughly speaking, a *G*-template is a template which is translation invariant with respect to a digraph induced by a neighborhood configuration which admits a group structure. In the case were the configuration is the von Neumann configuration, the group turns out to be $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ and the set of all *G*-templates will be the set of all circulant templates on $\mathbf{X}$. For general configurations, the set of complex-valued *G*-templates is a linear algebra over $\mathbb{C}$.

The details of this section reflect much of Gader's initial work. For the remainder of our discussion on *G*-templates we let $\mathbf{X}$ be a finite point set, *N* will denote a neighborhood configuration on $\mathbf{X}$, *G* will denote a finite group (written multiplicatively), and images will be complex valued.

7.4.1 **Definition.** Let $\Gamma = \{g_1, g_2, \ldots, g_k\}$ be a set of generators for the group *G*. The *Cayley color graph*, or *group graph, of G* with respect to Γ is the graph $D_\Gamma(G) = (V, E)$, where $V = G$ and $(x, y) \in E \Leftrightarrow \exists\ g_i \in \Gamma$ such that $xg_i = y$.

If $(x, y) \in E$ and $xg_i = y$, then we say that (x, y) is *colored* by g_i, or that (x, y) *has the color* g_i.

7.4.2 **Definition.** An *automorphism* of a digraph $D = (V, E)$ is a permutation $\sigma : V \to V$ with the property that $(x, y) \in E \Leftrightarrow (\sigma(x), \sigma(y)) \in E$.

A *color-preserving automorphism* of a Cayley color graph $D_\Gamma(G) = (V, E)$ is an automorphism σ of the digraph $D_\Gamma(G)$ with the additional property that for every edge (x, y), (x, y) and $(\sigma(x), \sigma(y))$ have the same color.

It is not difficult to ascertain that the set of color-preserving automorphisms of $D_\Gamma(G)$ is a group under composition which is isomorphic to *G*. Such an isomorphism can be defined by using the mapping rule $g \mapsto \sigma_g$, where $g \in G$ and σ_g is the automorphism of $D_\Gamma(G)$ defined by $\sigma_g \equiv gx$. Note that this implies that the group of color-preserving automorphisms is the same for every group graph constructed using the generating sets of *G* [3].

7.4.3 **Definition.** Let D_1 and D_2 be digraphs. We say that D_1 is *isomorphic* to D_2 if and only if there exists a one-to-one and onto function $\varsigma : V(D_1) \to V(D_2)$ with the property that $(x, y) \in E(D_1) \Leftrightarrow (\varsigma(x), \varsigma(y)) \in E(D_2)$.

In Section 5.8 we defined the digraph $D(N)$ of a neighborhood configuration *N* on $\mathbf{X}$. If $D(N)$ is isomorphic (as a digraph) to $D_\Gamma(G)$ for some group *G* and generating set Γ, then the isomorphism between the digraphs induces a group structure on $\mathbf{X}$ which is isomorphic to *G*. In order to demonstrate this we shall need the following definition:

7.4.4 **Definition.** The pair $(\mathbf{X}, N)$ is said to *simulate* the group *G* if and only if $D(N)$ is isomorphic to $D_\Gamma(G)$ for some generating set Γ.

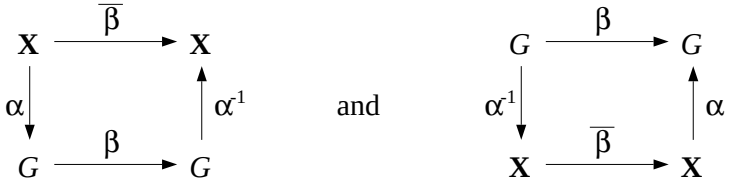
If $(\mathbf{X}, N)$ simulates *G*, then we assign to each arc in $D(N)$ the same color as the associated arc in $D_\Gamma(G)$.

7.4.5 **Theorem.** *If $(\mathbf{X}, N)$ simulates G, then there exists a binary operation $*$ on $\mathbf{X}$ such that $(\mathbf{X}, *)$ is a group which is isomorphic to G.*

Proof: Since $(\mathbf{X}, N)$ simulates *G*, there exists a one-to-one and onto function $\alpha : V(D(N)) \to V(D_\Gamma(G))$ for some generating set Γ of *G*. By definition of the respective digraphs, $V(D(N)) = \mathbf{X}$ and $V(D_\Gamma(G)) = G$. Hence, $\alpha : \mathbf{X} \to G$.

Defining the desired binary operation $*$ by $\mathbf{x} * \mathbf{y} \equiv \alpha^{-1}(\alpha(\mathbf{x}) \cdot \alpha(\mathbf{y}))$, we have that $\alpha(\mathbf{x} * \mathbf{y}) = \alpha(\mathbf{x}) \cdot \alpha(\mathbf{y})$. To check that $(\mathbf{X}, *)$ is indeed a group is a routine matter and, therefore, omitted. Q.E.D.

Unless otherwise mentioned, we shall identify $\mathbf{X}$ with *G* whenever $(\mathbf{X}, N)$ simulates *G*. Also, whenever we write $\mathbf{X}$ as a linearly ordered set $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{p-1}\}$, we shall assume that $\mathbf{x}_0 = e$ (i.e., $\alpha(\mathbf{x}_0) = e$), where *e* denotes the identity of *G*. Finally, note that if $\beta : G \to G$ is any one-to-one and onto function, then there exists a corresponding function $\bar{\beta} : \mathbf{X} \to \mathbf{X}$ defined by $\bar{\beta}(\mathbf{x}) = \alpha^{-1}(\beta(\alpha(\mathbf{x})))$. Conversely, given a one-to-one and onto function $\bar{\beta} : \mathbf{X} \to \mathbf{X}$, then there exists a corresponding function $\beta : G \to G$ defined by $\beta(g) = \alpha\big(\bar{\beta}\big(\alpha^{-1}(g)\big)\big)$. Pictorially we have the following two commutative diagrams representing these cases:



For this reason we shall use β to denote either map.

7.4.6 **Definition.** Suppose $(\mathbf{X}, N)$ simulates *G*. A template $\mathbf{t} \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$ is called a *G-template* if and only if for every color-preserving automorphism $\beta : \mathbf{X} \to \mathbf{X}$, the equation $\mathbf{t}_{\beta(\mathbf{y})}(\beta(\mathbf{x})) = \mathbf{t}_{\mathbf{y}}(\mathbf{x})$ holds. We denote the set of all *G*-templates on $\mathbf{X}$ by $G(\mathbb{C}, \mathbf{X})$.

The set of *G*-templates is, in a sense, the same as the set of templates that are translation invariant with respect to the group graph $D_\Gamma(G)$ or $D(N)$ since color-preserving automorphisms are essentially translations within the group.

The next theorem shows that *G*-templates are indeed generalizations of circulant templates.

7.4.7 Theorem. *If* $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ *and N denotes the von Neumann configuration, then* $(\mathbf{X}, N)$ *simulates the group* $G = \mathbb{Z}_m \times \mathbb{Z}_n$ *(written additively). Furthermore,* $G(\mathbb{C}, \mathbf{X}) = C(\mathbb{C}, \mathbf{X})$.

Proof: Let $\Gamma = \{(1,0), (0,1), (m-1,0), (0,n-1)\}$, and $\alpha : \mathbf{X} \to G$ the identity map. Thus, $\alpha : V(D(N)) \to V(D_\Gamma(G))$ is one-to-one and onto. Hence we need to show that $(\mathbf{x},\mathbf{y}) \in E(D(N)) \;\Leftrightarrow\; (\mathbf{x},\mathbf{y}) \in E(D_\Gamma(G))$.

Let $\mathbf{y} = (i,j) \in \mathbf{X}$ and $(\mathbf{x},\mathbf{y}) \in E(D(N))$. Then $\mathbf{x} \in N(\mathbf{y})$ or, equivalently, $\mathbf{x} \in \{((i \pm 1) mod m,\, j),\, (i,\, (j \pm 1) mod n)\}$. Therefore, $(\mathbf{y} - \mathbf{x}) mod(m,n) \in \Gamma$. But this says that $\mathbf{y} = (\mathbf{x} + \mathbf{z}) mod(m,n)$ for some $\mathbf{z} \in \Gamma$. This implies that $(\mathbf{x},\mathbf{y}) \in E(D_\Gamma(G))$. The converse is just as easy to show. Therefore, $(\mathbf{X}, N)$ simulates *G*.

To verify that $G(\mathbb{C}, \mathbf{X}) = C(\mathbb{C}, \mathbf{X})$, recall that the set of all circulant translations on $\mathbf{X}$ is a group under composition which is isomorphic to $G = (\mathbf{X}, +)$ (Theorem **7.2.2**). Also, *G* is isomorphic to the group of color-preserving automorphisms of $D_\Gamma(G)$ for any Γ; this follows from the assertion made immediately following Definition **7.4.2**. Therefore,

$$(\Psi(\mathbf{X}),\, \circ) \approx (\mathbf{X},\, +) = G \approx D_\Gamma(G),$$

where the symbol $\approx$ is interpreted as "is isomorphic to." Thus, a mapping $\varphi : \mathbf{X} \to \mathbf{X}$ is a color-preserving automorphism if and only if φ is a circulant translation. By definition, $\mathbf{t} \in C(\mathbb{C}, \mathbf{X}) \;\Leftrightarrow\; \mathbf{t}_{\varphi(\mathbf{y})}(\varphi(\mathbf{x})) = \mathbf{t}_\mathbf{y}(\mathbf{x})$ holds $\forall \mathbf{x}, \mathbf{y} \in \mathbf{X}$. Therefore, $G(\mathbb{C}, \mathbf{X}) = C(\mathbb{C}, \mathbf{X})$.

Q.E.D.

The theorem establishes the algebraic structure of the set of *G*–templates for the special case where *N* is the von Neumann configuration on $\mathbb{Z}_m \times \mathbb{Z}_n$. In this case $G(\mathbb{C}, \mathbf{X})$ is identical to the ring of circulant templates. The next theorem reveals the algebraic structure of the set of *G*-templates for the general case.

7.4.8 Theorem. $G(\mathbb{C}, \mathbf{X})$ *is a linear algebra over* $\mathbb{C}$.

Proof: Let $\mathbf{s}, \mathbf{t} \in G(\mathbb{C}, \mathbf{X})$. Since template addition is defined pointwise, it is clear that $\mathbf{s} + \mathbf{t} \in G(\mathbb{C}, \mathbf{X})$. The zero template $\emptyset \in G(\mathbb{C}, \mathbf{X})$. Hence, using Theorem **4.9.3**, we have that $(G(\mathbb{C}, \mathbf{X}), +)$ is a commutative group.

Since scalar multiplication is also defined pointwise, we have that $c\mathbf{t} \in G(\mathbb{C}, \mathbf{X})\ \forall c \in \mathbb{C}$. It now routine to check that $G(\mathbb{C}, \mathbf{X})$ is a vector space over $\mathbb{C}$. This shows that the first axiom of a linear algebra is satisfied (Section **3.9**).

Let $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z} \in \mathbf{X}$, φ a color-preserving automorphism of $D(N)$, and $\mathbf{r} = \mathbf{s} \bigoplus \mathbf{t}$. Note that

$$\mathbf{t}_{\varphi(\mathbf{y})}(\mathbf{x}) = \mathbf{t}_{\mathbf{y}}\left(\varphi^{-1}(\mathbf{x})\right) \text{ and } \mathbf{s}_{\mathbf{x}}(\varphi(\mathbf{z})) = \mathbf{s}_{\varphi^{-1}(\mathbf{x})}(\mathbf{z}).$$

Thus, since φ^{-1} is onto, we have

$$\begin{aligned}\mathbf{r}_{\varphi(\mathbf{y})}(\varphi(\mathbf{z})) = \sum_{\mathbf{x}\in\mathbf{X}} \mathbf{s}_{\mathbf{x}}(\varphi(\mathbf{z}))\mathbf{t}_{\varphi(\mathbf{y})}(\mathbf{x}) &= \sum_{\mathbf{x}\in\mathbf{X}} \mathbf{s}_{\varphi^{-1}(\mathbf{x})}(\mathbf{z})\mathbf{t}_{\mathbf{y}}\left(\varphi^{-1}(\mathbf{x})\right) \\ &= \sum_{\mathbf{w}\in\mathbf{X}} \mathbf{s}_{\mathbf{w}}(\mathbf{z})\mathbf{t}_{\mathbf{y}}(\mathbf{w}) = \mathbf{r}_{\mathbf{y}}(\mathbf{z})\,.\end{aligned}$$

Therefore, $\mathbf{r} = \mathbf{s} \bigoplus \mathbf{t} \in G(\mathbb{C}, \mathbf{X})$, which shows that $G(\mathbb{C}, \mathbf{X})$ is closed under multiplication. According to Eqs. 5.1.1, template multiplication is associative and both left and right distributive over template addition. This proves that the second and third axioms of a linear algebra are satisfied.

The one-point unit template $1 \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$ is clearly an element of $G(\mathbb{C}, \mathbf{X})$. In view of the pointwise multiplication of a template by a scalar, it is also a simple routine to check that $(c\mathbf{t}) \bigoplus \mathbf{s} = \mathbf{t} \bigoplus (c\mathbf{s}) = c(\mathbf{t} \bigoplus \mathbf{s})$ $\forall c \in \mathbb{C}$. This verifies that the remaining two axioms of a linear algebra are also satisfied.

Q.E.D.

This establishes the fundamental algebraic structure of generalized circulant templates. In the next section we shall delve deeper into the algebra of these templates.

7.5 Group Algebras and G-Templates

Group algebras were first discussed in Section **3.10**. In this section we will show that the set of all *G*-templates for some configuration *N* is isomorphic to the group algebra over $\mathbb{C}$ of the group corresponding to *N*.

Group algebras arise quite naturally in the study of *linear representations* of groups. These representations furnish another description of the set of *G*-templates. They also provide necessary and sufficient conditions for determining the invertibility of *G*-templates.

The topic of group representations has been in existence since before the turn of the century when Froboenius first defined group characters [12]. Since its early formulation, this topic has seen applications in physics and chemistry as well as in group theory. It is fascinating to realize that a subject developed at the turn of the century has now applications in the theory of parallel computing.

Let **X** be a finite point set that has a multiplicative operation which provides **X** with an abelian group structure. Let *G* denote this group and let $p = card(\mathbf{X})$. The elements of the group algebra $\mathbb{C}(G)$ are formal sums of the form

$$\alpha = \sum_{i=0}^{p-1} a_i \mathbf{x}_i\,, \tag{7.5.1}$$

where $\mathbf{x}_i \in \mathbf{X}$ and $a_i \in \mathbb{C}$ (see Section **3.10**).

The map $h : \mathbb{C}(G) \to \mathbb{C}^p$ defined by

$$h(\alpha) = (a_0, a_1, \ldots, a_{p-1}),$$

where α is given by Eq. 7.5.1, is clearly one-to-one and onto. Also, if

$$\beta = \sum_{i=0}^{p-1} b_i \mathbf{x}_i$$

and $c \in \mathbb{C}$, then

$$\begin{aligned} h(\alpha + \beta) &= h\left(\sum_{i=0}^{p-1} a_i \mathbf{x}_i + \sum_{i=0}^{p-1} b_i \mathbf{x}_i\right) \\ &= h\left(\sum_{i=0}^{p-1} (a_i + b_i)\mathbf{x}_i\right) \\ &= (a_0 + b_0, a_1 + b_1, \ldots, a_{p-1} + b_{p-1}) \\ &= (a_0, a_1, \ldots, a_{p-1}) + (b_0, b_1, \ldots, b_{p-1}) \\ &= h(\alpha) + h(\beta) \end{aligned}$$

and

$$\begin{aligned} c \cdot h(\alpha) &= c \cdot (a_0, a_1, \ldots, a_{p-1}) \\ &= (ca_0, ca_1, \ldots, ca_{p-1}) \\ &= h\left(\sum_{i=0}^{p-1} (ca_i)\mathbf{x}_i\right) \\ &= h\left(c \cdot \sum_{i=0}^{p-1} a_i \mathbf{x}_i\right) \\ &= h(c\alpha)\,. \end{aligned}$$

This shows that *h* is a vector space isomorphism. According to Theorem **4.4.3**, the vector space $\mathbb{C}^p$ is also isomorphic to $\mathbb{C}^{\mathbf{X}}$. We have, therefore, established the following result:

7.5.1 **Theorem.** $\mathbb{C}(\mathbf{X})$ *and* $\mathbb{C}^{\mathbf{X}}$ *are isomorphic vector spaces.*

We let $f : \mathbb{C}(\mathbf{X}) \to \mathbb{C}^{\mathbf{X}}$ denote the natural vector space isomorphism so that if α is given by Eq. 7.5.1, then $f(\alpha) = \mathbf{a} \in \mathbb{C}^{\mathbf{X}}$ is given by $\mathbf{a}(\mathbf{x}_i) = a_i \;\; \forall i = 0, 1, \ldots, p-1$. For the remainder of this section we shall write images in terms of their vector representation $\mathbf{a} = (\mathbf{a}(\mathbf{x}_0), \mathbf{a}(\mathbf{x}_1), \ldots, \mathbf{a}(\mathbf{x}_{p-1}))$.

Clearly, $\mathbb{C}(\mathbf{X})$ and $\mathbb{C}^{\mathbf{X}}$ are *not* isomorphic as algebras; image multiplication, which is defined pointwise, does not correspond to group convolutions under *f*. However, we can define a new product on $\mathbb{C}^{\mathbf{X}}$ that will turn $\mathbb{C}^{\mathbf{X}}$ into an algebra which is isomorphic to $\mathbb{C}(\mathbf{X})$. Consider the group convolution

$$\alpha * \beta = \sum_{i=0}^{p-1} \left(\sum_{\mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i} a_j b_k \right) \mathbf{x}_i = \gamma\,.$$

The element γ is of form $\gamma = \sum_{i=0}^{p-1} c_i \mathbf{x}_i$, where $c_i = \sum_{\mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i} a_j b_k$, while $f(\gamma) = \mathbf{c} \in \mathbb{C}^{\mathbf{X}}$ is given by $\mathbf{c}(\mathbf{x}_i) = c_i$. But

$$c_i = \sum_{\mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i} a_j b_k = \sum_{\mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i} \mathbf{a}(\mathbf{x}_j)\mathbf{b}(\mathbf{x}_k) = \sum_{\mathbf{x}_j \in \mathbf{X}} \mathbf{a}(\mathbf{x}_j)\mathbf{b}\left(\mathbf{x}_i \cdot \mathbf{x}_j^{-1}\right),$$

since $\mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i \;\Leftrightarrow\; \mathbf{x}_k = \mathbf{x}_j^{-1} \cdot \mathbf{x}_i = \mathbf{x}_i \cdot \mathbf{x}_j^{-1}$. By defining the *convolution product* of two elements $\mathbf{a}, \mathbf{b} \in \mathbb{C}^{\mathbf{X}}$ by $\mathbf{c} = \mathbf{a} * \mathbf{b}$ such that $\mathbf{c}(\mathbf{x}_i) = \sum_{\mathbf{u} \in \mathbf{X}} \mathbf{a}(\mathbf{u})\mathbf{b}\left(\mathbf{x}_i \cdot \mathbf{u}^{-1}\right)$, we have that $f(\alpha * \beta) = f(\alpha) * f(\beta)$. It now follows that the algebra $\left(\mathbb{C}^{\mathbf{X}}, +, *\right)$ is isomorphic to the group algebra $\mathbb{C}(G)$. This isomorphism provides us with the convenience of letting the elements of $\mathbb{C}^{\mathbf{X}}$ represent the elements of $\mathbb{C}(G)$.

7.5.2 **Example:** Let *G* denote the group $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ with the operation of addition $mod(m, n)$. We make $\mathbf{X}$ into a multiplicative group by defining

$$\mathbf{x}_i \cdot \mathbf{x}_j = \mathbf{x}_k \;\Leftrightarrow\; (\mathbf{x}_i + \mathbf{x}_j) mod(m, n) = \mathbf{x}_k \,.$$

If $\mathbf{a}, \mathbf{b} \in \mathbb{C}^{\mathbf{X}}$, then $\mathbf{c} = \mathbf{a} * \mathbf{b}$ is given by

$$\mathbf{c}(\mathbf{x}_i) = \sum_{\mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i} \mathbf{a}(\mathbf{x}_j)\mathbf{b}(\mathbf{x}_k) = \sum_{\mathbf{x}_j \in \mathbf{X}} \mathbf{a}(\mathbf{x}_j)\mathbf{b}\left(\mathbf{x}_i \cdot \mathbf{x}_j^{-1}\right),$$

where $\mathbf{x}_i \cdot \mathbf{x}_j^{-1} = (\mathbf{x}_i - \mathbf{x}_j) mod(m, n)$. Equivalently,

$$\mathbf{c}(\mathbf{x}_i) = \sum_{\mathbf{u} \in \mathbf{X}} \mathbf{a}(\mathbf{u})\mathbf{b}((\mathbf{x} - \mathbf{u}) mod(m, n)) \,.$$

Thus, $\mathbf{a} * \mathbf{b}$ is a circular convolution.

7.5.3 **Lemma.** *Suppose* $(\mathbf{X}, N)$ *simulates G,* $\varphi_{\mathbf{z}} : \mathbf{X} \to \mathbf{X}$ *denotes the color-preserving automorphism corresponding to* $\mathbf{z} \in \mathbf{X}$*, and* $\mathbf{a} \in \mathbb{C}^{\mathbf{X}}$*. If* $\mathbf{t}(\mathbf{a}) \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$ *denotes the parametrized template defined by*

$$\mathbf{t}(\mathbf{a})_{\mathbf{y}}(\mathbf{x}) = \begin{cases} \mathbf{a}(\mathbf{x}) & if\ \mathbf{y} = \mathbf{x}_0 \\ \mathbf{a}\left(\varphi_{\mathbf{y}}^{-1}(\mathbf{x})\right) & if\ \mathbf{y} \neq \mathbf{x}_0 \,, \end{cases}$$

where $\mathbf{x}_0$ *represents the identity of* $\mathbf{X}$*, then* $\mathbf{t}(\mathbf{a})$ *is a G-template.*

Proof: Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{X}$. Since the map $\mathbf{z} \mapsto \varphi_{\mathbf{z}}$ is a group isomorphism, we have that

$$\varphi^{-1}_{\varphi_{\mathbf{z}}(\mathbf{y})}(\varphi_{\mathbf{z}}(\mathbf{x})) = \varphi_{(\varphi_{\mathbf{z}}(\mathbf{y}))^{-1}}(\varphi_{\mathbf{z}}(\mathbf{x})) \,.$$

Therefore,

$$\varphi^{-1}_{\varphi_{\mathbf{z}}(\mathbf{y})}(\varphi_{\mathbf{z}}(\mathbf{x})) = (\varphi_{\mathbf{z}}(\mathbf{y}))^{-1} \cdot \varphi_{\mathbf{z}}(\mathbf{x}) = (\mathbf{z}\mathbf{y})^{-1}\mathbf{z}\mathbf{x} = \mathbf{y}^{-1}\mathbf{x} = \varphi_{\mathbf{y}}^{-1}(\mathbf{x}) \,,$$

and

$$\mathbf{t}(\mathbf{a})_{\varphi_{\mathbf{z}}(\mathbf{y})}(\varphi_{\mathbf{z}}(\mathbf{x})) = \mathbf{a}\left(\varphi^{-1}_{\varphi_{\mathbf{z}}(\mathbf{y})}(\varphi_{\mathbf{z}}(\mathbf{x}))\right) = \mathbf{a}\left(\varphi_{\mathbf{y}}^{-1}(\mathbf{x})\right) = \mathbf{t}(\mathbf{a})_{\mathbf{y}}(\mathbf{x}) \,.$$

Q.E.D.

7.5.4 Theorem. *If $(\mathbf{X}, N)$ simulates G, then $G(\mathbb{C}, \mathbf{X})$ is isomorphic (as an algebra) to $\mathbb{C}(G)$.*

Proof: Let $\mathbf{t} \in G(\mathbb{C}, \mathbf{X})$ and define $\xi : G(\mathbb{C}, \mathbf{X}) \to \mathbb{C}(G)$ by

$$\xi(\mathbf{t}) = (\mathbf{t}_{\mathbf{x}_0}(\mathbf{x}_0), \mathbf{t}_{\mathbf{x}_0}(\mathbf{x}_1), \ldots, \mathbf{t}_{\mathbf{x}_0}(\mathbf{x}_{p-1})).$$

For each $\mathbf{a} = (\mathbf{a}(\mathbf{x}_0), \mathbf{a}(\mathbf{x}_1), \ldots, \mathbf{a}(\mathbf{x}_{p-1})) \in \mathbb{C}(G)$, let $\mathbf{t}(\mathbf{a})$ denote the corresponding parametrized *G*-template guarantied by Lemma **7.5.3**. By definition of ξ we have that $\xi(\mathbf{t}(\mathbf{a})) = \mathbf{a}$. This shows that ξ is onto. In order to show that ξ is one-to-one, suppose that $\xi(\mathbf{t}) = \xi(\mathbf{s})$. Again by definition of ξ, $\mathbf{t}_{\mathbf{x}_0}(\mathbf{x}_i) = \mathbf{s}_{\mathbf{x}_0}(\mathbf{x}_i)$ for each $i = 0, 1, \ldots, p-1$. Let $\mathbf{y} \in \mathbf{X}$ and β a color-preserving automorphism with the property $\beta(\mathbf{x}_0) = \mathbf{y}$. Then $\mathbf{t}_{\mathbf{y}}(\beta(\mathbf{x}_i)) = \mathbf{s}_{\mathbf{y}}(\beta(\mathbf{x}_i))$ for each $i = 0, 1, \ldots, p-1$. Since β is an automorphism, we have that $\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{s}_{\mathbf{y}}(\mathbf{x})$ $\forall \mathbf{x} \in \mathbf{X}$. Since $\mathbf{y}$ was arbitrarily chosen this shows that $\mathbf{t} = \mathbf{s}$.

We have left to show that ξ preserves the algebraic operations. Since addition and scalar multiplication are defined pointwise on both $G(\mathbb{C}, \mathbf{X})$ and $\mathbb{C}(G)$, the equations $\xi(\mathbf{s} + \mathbf{t}) = \xi(\mathbf{s}) + \xi(\mathbf{t})$ and $c \cdot \xi(\mathbf{t}) = \xi(c \cdot \mathbf{t})$ follow immediately.

Let $\mathbf{s}, \mathbf{t} \in G(\mathbb{C}, \mathbf{X})$, $\mathbf{a} = \xi(\mathbf{s})$, $\mathbf{b} = \xi(\mathbf{t})$, $\mathbf{r} = \mathbf{t} \oplus \mathbf{s}$, $\mathbf{c} = \xi(\mathbf{r})$, and $\mathbf{d} = \xi(\mathbf{s}) * \xi(\mathbf{t})$. Since

$$\mathbf{t}_{\mathbf{u}}(\mathbf{x}) = \mathbf{t}_{\mathbf{x}_0}(\varphi_{\mathbf{u}^{-1}}(\mathbf{x})) = \mathbf{t}_{\mathbf{x}_0}\left(\mathbf{x} \cdot \mathbf{u}^{-1}\right) = \mathbf{b}\left(\mathbf{x} \cdot \mathbf{u}^{-1}\right)$$

and $\mathbf{s}_{\mathbf{x}_0}(\mathbf{u}) = \mathbf{a}(\mathbf{u})$, we have that

$$\begin{aligned}\mathbf{c}(\mathbf{x}) = \mathbf{r}_{\mathbf{x}_0}(\mathbf{x}) &= \sum_{\mathbf{u} \in \mathbf{X}} \mathbf{t}_{\mathbf{u}}(\mathbf{x}) \mathbf{s}_{\mathbf{x}_0}(\mathbf{u}) = \sum_{\mathbf{u} \in \mathbf{X}} \mathbf{b}\left(\mathbf{x} \cdot \mathbf{u}^{-1}\right) \mathbf{a}(\mathbf{u}) \\ &= \sum_{\mathbf{u} \in \mathbf{X}} \mathbf{a}(\mathbf{u}) \mathbf{b}\left(\mathbf{x} \cdot \mathbf{u}^{-1}\right) = \mathbf{d}(\mathbf{x}).\end{aligned}$$

Thus, $\xi(\mathbf{t} \oplus \mathbf{s}) = \xi(\mathbf{s}) * \xi(\mathbf{t})$. However, since *G* is a commutative group, we have by Theorem **3.10.1** that $\xi(\mathbf{s}) * \xi(\mathbf{t}) = \xi(\mathbf{t}) * \xi(\mathbf{s})$ and, therefore, $\xi(\mathbf{t} \oplus \mathbf{s}) = \xi(\mathbf{t}) * \xi(\mathbf{s})$.

Q.E.D.

7.5.5 Corollary. *If* $\mathbf{s}$ *and* $\mathbf{t}$ *are G-templates, then* $\mathbf{s} \oplus \mathbf{t} = \mathbf{t} \oplus \mathbf{s}$.

Note that if $G = \mathbb{Z}_m \times \mathbb{Z}_n$, then ξ is identical to the map θ_0 defined in Section **7.3**.

At the end of Section **3.4** we mentioned that every finite group *G* is isomorphic to some subgroup of S_n or, equivalently, to some subgroup G' of S_G, the group of all permutations of *G*. This fact, known as Cayley's Theorem, allows for the *representation* of *G* in terms of G'. The study of group representations has become a formal theory in its own right. The *theory of representations* is closely connected with the theory of algebras. It deals with the problem of mapping an abstract group, ring, or group algebra homomorphically (i.e., without destroying the algebraic structure) into the more concrete group or ring of matrices or (equivalently) linear transformations of a vector space. For our purposes, this theory provides another description of *G*-templates in terms of the representation of the group algebra $\mathbb{C}(G)$.

7.5.6 **Definition.** A *linear representation* of a group *G* over $\mathbb{C}$ is a group homomorphism $\alpha : G \to M_{k\times k}(\mathbb{C})$ for some integer $k \geq 1$. A *linear representation* of the group algebra $\mathbb{C}(G)$ over $\mathbb{C}$ is an algebra homomorphism $\alpha : G \to M_{k\times k}(\mathbb{C})$; that is, α preserves sums, products, and scalar multiplication.

Suppose α is a representation of *G* and *e* denotes the identity of *G*. If $M \in range(\alpha)$ and $g \in G$ such that $\alpha(g) = M$, then

$$M \cdot \alpha(e) = \alpha(g \cdot e) = \alpha(g) = M = \alpha(e \cdot g) = \alpha(e) \cdot \alpha(g) = \alpha(e) \cdot M.$$

Therefore, $I_k = \alpha(e)$. Similarly,

$$I_n = \alpha(e) = \alpha\big(g \cdot g^{-1}\big) = \alpha(g) \cdot \alpha\big(g^{-1}\big) \ \ \forall g \in G\,,$$

which shows that $\alpha\big(g^{-1}\big) = [\alpha(g)]^{-1}$. This essentially shows that $range(\alpha)$ is a group. The group $range(\alpha)$ is referred to as the *k-dimensional linear representation* of *G*.

The map α is only a homomorphism. We now turn our attention to isomorphic representations. As before, we let *G* denote the multiplicative abelian group $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{p-1}\}$ with identity $\mathbf{x}_0$. For each $k \in \mathbb{Z}_p$, let $\tau_k \in S_p$ denote the permutation defined by the rule $\tau_k(i) = j \ \Leftrightarrow \ \mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i$.

7.5.7 **Definition.** A *right regular representation* of $G = \mathbf{X}$ is a representation $\alpha : G \to M_{p\times p}(\mathbb{C})$ of the form $\alpha(\mathbf{x}_k) = P_{\tau_k}$. A *right regular representation* of $\mathbb{C}(G)$ is a representation $\alpha^* : \mathbb{C}(G) \to M_{p\times p}(\mathbb{C})$ of the form

$$\alpha^*(\mathbf{a}(\mathbf{x}_0), \mathbf{a}(\mathbf{x}_1), \ldots, \mathbf{a}(\mathbf{x}_{p-1})) = \sum_{i=0}^{p-1} \mathbf{a}(\mathbf{x}_i)\alpha(\mathbf{x}_i)\,,$$

where α is a right regular representation of *G*.

Note that for a right regular representation of a group *G* the cardinality of *G* equals the dimensionality of the representation. In fact, right regular representations are isomorphisms. Note that there is a different right regular representation of *G* for every different ordering of *G* [8].

According to our definition, $\alpha^*(\mathbf{a}) = \sum_{i=0}^{p-1} \mathbf{a}(\mathbf{x}_i)\alpha(\mathbf{x}_i) = \sum_{i=0}^{p-1} \mathbf{a}(\mathbf{x}_i)P_{\tau_i}$. Thus, it is clear that any representation α of *G* can be extended to a representation α^* of $\mathbb{C}(G)$. In fact, *G* can be considered to be embedded in $\mathbb{C}(G)$ via the map $\mathbf{x}_i \rightarrowtail \mathbf{e}_{i+1}$, where $\mathbf{e}_{i+1}$ represents the *i*th element of the standard basis. Hence we can define $\alpha^*(\mathbf{e}_{i+1}) = \alpha(\mathbf{x}_i)$ and extend linearly. It can be shown that multiplication is preserved [11]. Henceforth, we use the same name for either representation; that is, we take $\alpha^* = \alpha$.

7.5.8 **Theorem.** *Let* $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{p-1}\}$ *and* $\psi : G(\mathbb{C}, \mathbf{X}) \to M_{p\times p}(\mathbb{C})$ *be defined relative to this ordering. If* α *is a right regular representation of* $\mathbb{C}(G)$*, then* $\psi = \alpha \circ \xi$.

The function $\psi : G(\mathbb{C}, \mathbf{X}) \to M_{p\times p}(\mathbb{C})$ denotes the restriction of the isomorphism $\psi : \big(\mathbb{C}^{\mathbf{X}}\big)^{\mathbf{X}} \to M_{p\times p}(\mathbb{C})$ defined in Section **4.9**. The conclusion of the theorem is that the diagram

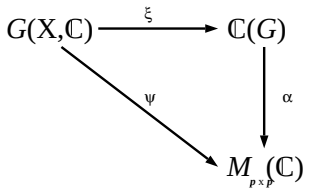


Figure 7.5.1

is a commutative diagram.

Proof: Let $\mathbf{a} \in \mathbb{C}(G)$, $M_{\mathbf{a}} = \alpha(\mathbf{a})$, $\mathbf{t}(\mathbf{a}) = \xi^{-1}(\mathbf{a})$, and $\psi(\mathbf{t}(\mathbf{a})) = (t_{ij})$. Then

$$t_{ij} = \mathbf{t}(\mathbf{a})_{\mathbf{x}_j}(\mathbf{x}_i) = \mathbf{t}(\mathbf{a})_{\mathbf{x}_0}\left(\varphi^{-1}(\mathbf{x}_i)\right) = \mathbf{a}\left(\mathbf{x}_i \cdot \mathbf{x}_j^{-1}\right).$$

Now let $M_{\mathbf{a}} = (m_{ij})$, $P_{\tau_k} = \left(p_{ij}^{(k)}\right)$, and suppose that $p_{ij}^{(k)} = 1$ for some triple (i, j, k). Then $\tau_k(i) = j$ and $\mathbf{x}_j \cdot \mathbf{x_k} = \mathbf{x_i}$. If we also have $p_{ij}^{(l)} = 1$ for some *l*, then $\mathbf{x}_j \cdot \mathbf{x_l} = \mathbf{x_i}$ and, therefore, $\mathbf{x_k} = \mathbf{x_j^{-1}} \cdot \mathbf{x_i} = \mathbf{x}_l$. This implies that $k = l$. Since

$$\alpha(\mathbf{a}) = \sum_{i=0}^{p-1} \mathbf{a}(\mathbf{x}_i) P_{\tau_i} = M_{\mathbf{a}},$$

we have $m_{ij} = \mathbf{a}(\mathbf{x}_k) \Leftrightarrow p_{ij}^{(k)} = 1 \Leftrightarrow \mathbf{x}_j \cdot \mathbf{x}_k = \mathbf{x}_i$. Therefore,

$$m_{ij} = \mathbf{a}\left(\mathbf{x}_i \cdot \mathbf{x}_j^{-1}\right) = \mathbf{t}(\mathbf{a})_{\mathbf{x}_j}(\mathbf{x}_i) = t_{ij}$$

and

$$\psi(\mathbf{t}(\mathbf{a})) = M_{\mathbf{a}} = \alpha(\mathbf{a}) = \alpha\left(\xi\left(\xi^{-1}(\mathbf{a})\right)\right) = \alpha \circ \xi(\mathbf{t}(\mathbf{a})).$$

Q.E.D.

This concludes our algebraic characterization of generalized circulant templates. In the next chapter we shall examine some practical applications of these theoretical considerations.

Bibliography

[1] H.C. Andrews. *Computer Techniques in Image Processing*. Academic Press, New York, 1970.

[2] L. Auslander and R. Tolmieri. Is computing with the finite fourier transform pure or applied mathematics? *Bulletin of the AMS*, 2(6):847–894, November 1979.

[3] M. Behzad, G. Chartrand, and L. Lesniak-Foster. *Graphs and Digraphs*. Wadsworth International Group, Belmont, CA, 1979.

[4] R. Blahut. *Fast Algorithms for Digital Signal Processing*. Addison-Wesley, Reading, MA, 1985.

[5] J.H. Davenport, Y. Siret, and E. Tournier. *Computer Algebra*. Academic Press, 1988.

[6] P.J. Davis. *Circulant Matrices*. John Wiley and Sons, New York, NY, 1979.

[7] P.D. Gader. *Image Algebra Techniques for Parallel Computation of Discrete Fourier Transforms and General Linear Transforms*. PhD thesis, University of Florida, Gainesville, FL, 1986.

[8] V.E. Hill. *Groups, Representations, and Characters*. Macmillian, New York, 1975.

[9] T. Kailath, B. Levy, L. Ljung, and M. Morf. The factorization and representation of operators in the algebra generated by toeplitz operators. *SIAM J. Appl. Math.*, 37(3):467–484, December 1979.

[10] E. Kaltofen. Polynomial time reduction from multivariate to bivariate to univariate integer polynomial factorization. *SIAM J. Comput*, 14:469–489, 1985.

[11] R. Keown. *An Introduction to Group Representation Theory*. Academic Press, New York, 1975.

[12] W. Ledermann. *Introduction to Group Characters*. Cambridge University Press, Cambridge, 1977.

[13] D.R. Musser. Multivariate polynomial factorization. *Journal of the Association for Computing Machinery*, 22(2):291–308, April 1975.

[14] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[15] H.B. Sexton. Cayley networks as parallel computer architectures. In *Cooperative Research Associateships tenable at the Naval Ocean Systems Center*. National Research Council, Washington, DC, 1986.

[16] S. Winograd. On computing the discrete fourier transform. *Math. Comp.*, 32:175–199, January 1978.

CHAPTER 8
INVERSION OF TRANSLATION INVARIANT TEMPLATES

Inverse problems have come to play a central role in modern applied mathematics such as mathematical physics, in imaging areas such as computerized tomography, seismic imaging, remote sensing, and image restoration. The importance of the Fourier transform would be drastically reduced were it not for its invertibility. In this chapter we are concerned with the inversion of transforms of form $\mathbf{a} \rightarrowtail \mathbf{a} \bigoplus \mathbf{t}$, where **t** belongs to the class of general translation invariant templates. Rosenfeld and Kak [17] explained the equivalence between the Wiener filter, least square methods, and the problem of inverting a block circulant matrix with circulant blocks. In 1973, Trapp [18] showed how the discrete Fourier transform can be used to diagonalize and invert a matrix that is either circulant or block circulant with circulant blocks. While the algebraic relationship between circulant matrices and polynomials was completely formulated by Davis [2], it was Gader and Ritter [5, 16] who established the connection between polynomials, circulant templates, and *G*-templates. A consequence of this connection is that a circulant template **t** defined on an $n \times n$ array is invertible if and only if its corresponding polynomial $p_{\mathbf{t}}(x, y)$ has the property that $p_{\mathbf{t}}\left(\omega_m^j, \omega_n^k\right) \neq 0$ for all $0 \leq j \leq m - 1$ and $0 \leq k \leq n - 1$ (Section **8.2**).

8.1 The Radon Transform

Our interest in the Radon transform stems from the role this transform plays in the invertibility of *G*-templates. Prior to examining this role, we provide a brief overview of some basic concepts associated with this transform.

In the classical setting, the Radon transform involves the evaluation of integrals of real-valued functions of *n* variables over $(n - 1)$–dimensional hyperplanes. Techniques from Radon transform theory have been successfully employed in medical imaging (particularly computer aided tomography), in radio astronomy, and in non-destructive testing. Less well-known are applications in representation of Lie groups and applied statistics [6, 10, 1, 4].

For functions of two variables — which are of major interest in image processing applications — the Radon transform reduces to a line integral. For a continuous function $\mathbf{a} : \mathbb{R}^2 \rightarrow \mathbb{R}$, the Radon transform of **a**, denoted by $\mathfrak{R}(\mathbf{a})$, can be found by integrating along a line **L** given by the normal equation

$$\rho = x \cos \phi + y \sin \phi \,, \tag{8.1.1}$$

where ϕ denotes the angle between the *x*–axis and the line perpendicular to **L**, and $\rho \geq 0$ denotes the distance from the origin to **L**. Figure **8.1.1** illustrates the normal form of **L**.

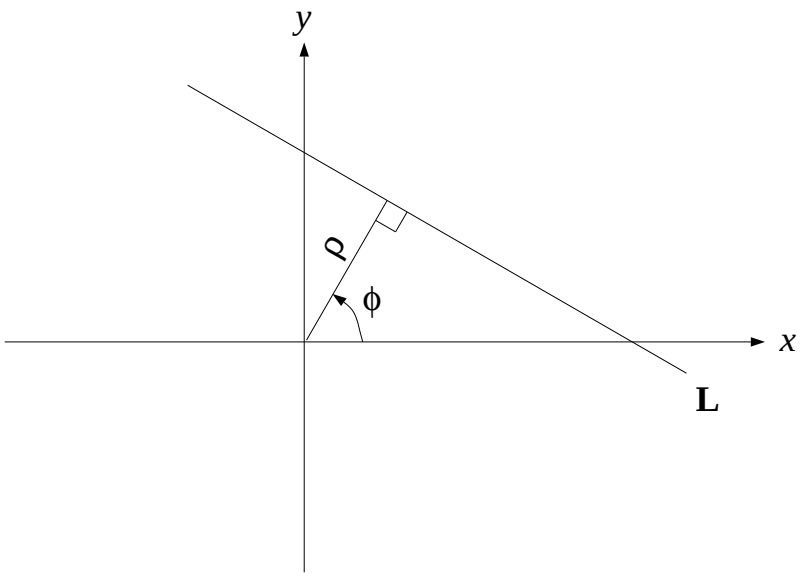


Figure 8.1.1 The normal-form representation of **L**.

The parametric equations for **L** are given by

$$z = \frac{y - \rho \sin \phi}{\cos \phi} = \frac{x - \rho \cos \phi}{-\sin \phi} \tag{8.1.2}$$

Solving for x and y, we obtain

$$\begin{aligned} x &= \rho \cos \phi - z \sin \phi \\ y &= \rho \sin \phi + z \cos \phi \end{aligned} \tag{8.1.3}$$

Therefore, the Radon transform $\mathfrak{R}(\mathbf{a})$ is given by the integral

$$[\mathfrak{R}(\mathbf{a})](\rho, \phi) = \int_{-\infty}^{\infty} \mathbf{a}(\rho \cos \phi - z \sin \phi,\ \rho \sin \phi + z \cos \phi) dz \tag{8.1.4}$$

8.1.1 **Example:** Let $\mathbf{a} : \mathbb{R}^2 \to \mathbb{R}$ be defined by

$$\mathbf{a}(\mathbf{x}) = \begin{cases} 1 & \text{if } \|\mathbf{x}\| \leq 1 \\ 0 & \text{if } \|\mathbf{x}\| > 1 \end{cases}$$

Graphically, **a** represents a disk of radius 1 (Figure **8.1.2**.)

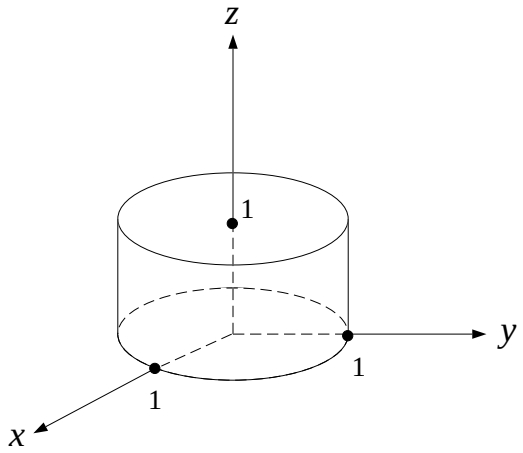


Figure 8.1.2 The graph of **a**.

Since $\mathbf{a}(\mathbf{x}) = 0$ for all points **x** outside the circle of radius 1 we have

$$[\mathfrak{R}(\mathbf{a})](\rho,\phi) = \int_{-\infty}^{\infty} \mathbf{a}(\rho\cos\phi - z\sin\phi,\ \rho\sin\phi + z\cos\phi)dz$$
$$= \int_{-\sqrt{1-\rho^2}}^{\sqrt{1-\rho^2}} dz = 2\sqrt{1-\rho^2}\,,$$

where $0 \leq \rho \leq 1$.

The limits of integration are found by substituting the right side of Eq. 8.1.3 for *x* and *y* in the equation $\|\mathbf{x}\|^2 = x^2 + y^2 = 1$. Substitution yields $\rho^2 + z^2 = 1$ and, hence, $z = \pm\sqrt{1-\rho^2}$ (see Figure **8.1.3**).

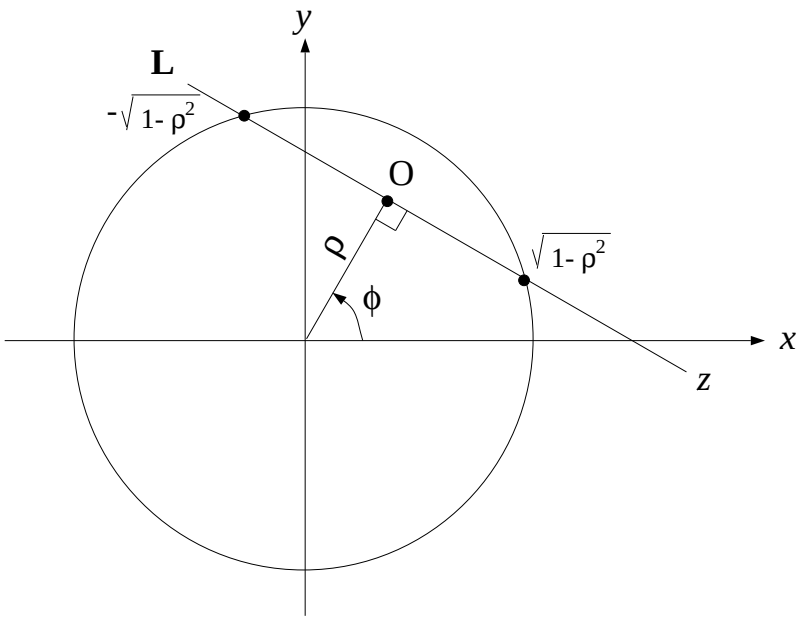


Figure 8.1.3 The limits of integration on the line **L**.

In practice, the Radon transform of an image **a** is given empirically by sensors. These sensors count the number of photons that are emitted from an x-ray source and pass through some material body such as human tissue. Different tissue densities will absorb different amounts of photons. By knowing the amount of photon energy emitted by the source versus the amount of energy received by the sensor, an approximation of $[\mathfrak{R}(\mathbf{a})](\rho,\phi)$ is obtained for a fixed position of the x-ray source and sensor. Specifically,

$$[\mathfrak{R}(\mathbf{a})](\rho,\phi) \approx lnC - lnA\,, \tag{8.1.5}$$

where *A* represents the number of photons that are detected by the sensor and *C* denotes the number of photons which are emitted from the source; i.e., *C* corresponds to the number of photons that would be received by the sensor if no obstruction were present. In actual tomographic imaging, the value *C* is usually precalibrated. The typical method by which data is collected for transverse section imaging in computer tomography is indicated in Figure **8.1.4**.

The goal in computer tomography is to reconstruct the image **a** from the measurements $-ln(A/C)$. Since $[\mathfrak{R}(\mathbf{a})](\rho,\phi)$ represents the amount of energy absorbed along the line **L**, it follows from Eq. 8.1.4 that $\mathbf{a}(\mathbf{x})$ corresponds to the absorption capacity of the tissue at location **x**. This absorption capacity corresponds to the tissue density at location **x**.

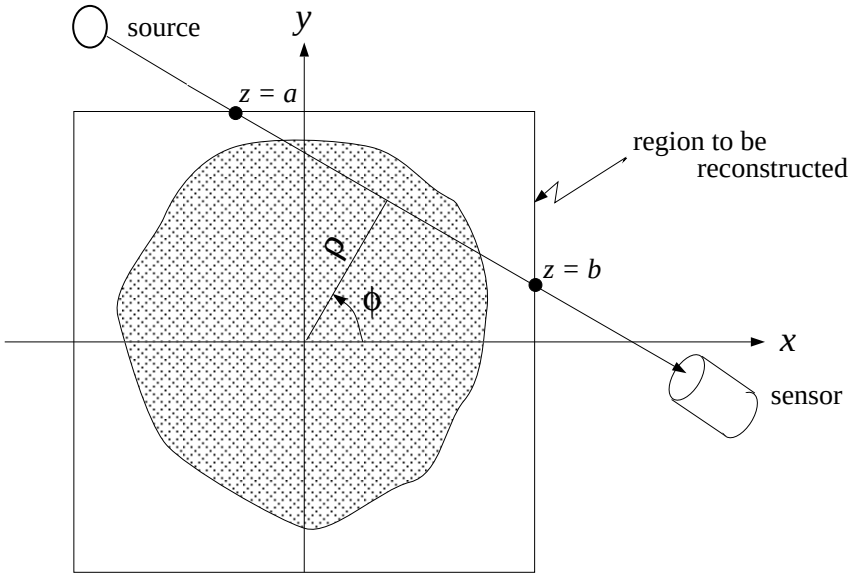


Figure 8.1.4 Data collection for computer tomography. The image **a** to be reconstructed is rectangular and contains the area of obstruction (shaded area) such as a cross section of the human body. The support of the image to be reconstructed corresponds to the area of obstruction.

In the formulation of Eq. 8.1.4 there is an important difference between the domains of the image function **a** and the function $\mathfrak{R}(\mathbf{a})$. In polar form, the image function **a** is defined for pairs of real numbers (r, θ) which represent the polar coordinates of points in the (x, y)–plane (Figure **8.1.5**). The pair of real numbers (ρ, ϕ) in the domain of $\mathfrak{R}(\mathbf{a})$ on the other hand, is *not* to be interpreted as polar coordinates in the (x, y)–plane. Roughly speaking, the operator $\mathfrak{R}$ associates with a function **a** over (r, θ)–space another function $\mathfrak{R}(\mathbf{a})$ over (ρ, ϕ)–space, where each point (ρ, ϕ) in (ρ, ϕ)–space corresponds to a unique line **L** (determined by (ρ, ϕ)) in (x, y)–space. The basic relation between (r, θ) and (ρ, ϕ) is given by

$$\rho = r \cos(\phi - \theta) \tag{8.1.6}$$

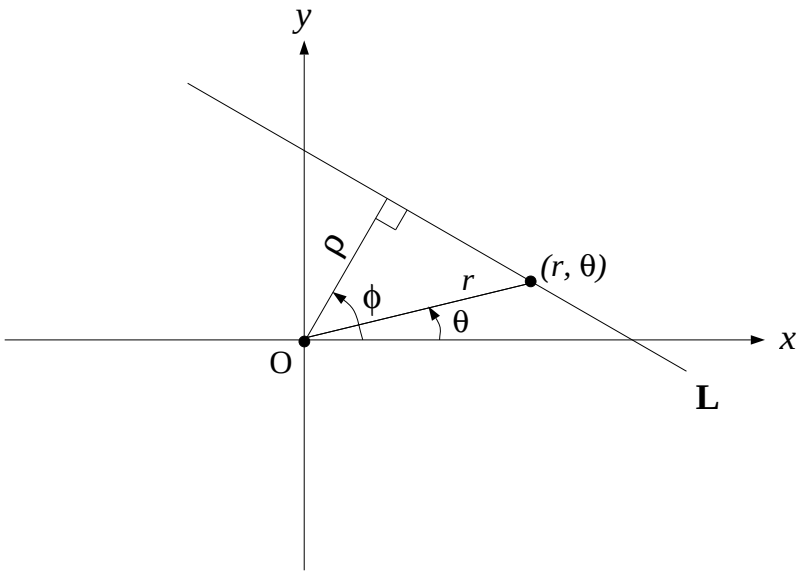


Figure 8.1.5 The relationship between (r, θ) and (ρ, ϕ).

It is easily verified that $(r, \theta) = \left(\sqrt{\rho^2 + z^2},\ \phi + \tan^{-1}(z/\rho)\right)$. Hence the polar form of the Radon transform is given by

$$[\mathfrak{R}(\mathbf{a})](\rho, \phi) = \begin{cases} \int_{-\infty}^{\infty} \mathbf{a}\left(\sqrt{\rho^2 + z^2},\ \phi + \tan^{-1}(z/\rho)\right) dz & if\ \ \rho \neq 0 \\ \int_{-\infty}^{\infty} \mathbf{a}(z,\ \phi + \pi/2) dz & if\ \ \rho = 0\,. \end{cases} \tag{8.1.7}$$

Reconstructing the image **a** corresponds to inverting the Radon transform $\mathfrak{R}(\mathbf{a})$. This problem was solved in 1917 by Radon [15]. Radon proved that

$$\mathbf{a}(r, \theta) = \frac{1}{2\pi^2} \lim_{\varepsilon \to 0} \int_{\varepsilon}^{\infty} \frac{1}{q} \int_{0}^{2\pi} \left[\frac{\partial}{\partial \rho} \mathfrak{R}(\mathbf{a})\right](r\cos(\phi - \theta) + q,\ \phi) d\phi dq\,, \tag{8.1.8}$$

where $q \in \mathbb{R}^+$. Note that the integral

$$\int_{0}^{2\pi} f(r\cos(\phi - \theta) + q,\ \phi) d\phi dq\,,$$

where $f = \frac{\partial}{\partial \rho}\mathfrak{R}(\mathbf{a})$, is an integral over a circle of radius q with center (r, θ).

Although the exact details of 8.1.8 may seem a bit obscure, the implication is clear: the value $\mathbf{a}(r, \theta)$ is uniquely determined by the set of *all* its line integrals; the integration is with respect to *all* values of ϕ $(0 \leq \phi \leq 2\pi)$ and the partial derivative is with respect to the *variable* ρ.

In computer tomography one deals with only a finite number of receivers. It is, therefore, impossible to calculate all the line integrals. The basic geometry for data collection is shown in Figure **8.1.6**. The source and sensors are on opposite sides of the object to be reconstructed. The sensors reside on a

detector strip and the source and detector strip move in unison around a common center of rotation. The x-ray source assumes *m* distinct positions during a stepwise rotation (indicated by $s_0,\ s_1,\ \ldots,\ s_{m-1}$ in Figure **8.1.6**). The detector strip contains $2n+1$ sensors spaced equally on an arc whose centers corresponds to the source position as shown. For each position of the source, $2n+1$ measurements are obtained. The total number of measurements (i.e., the number of line integrals) is, therefore, $m(2n+1)$ — which is not the infinite number of measurements required by Eq. 8.1.8. Hence, there is no hope for exact reconstruction.

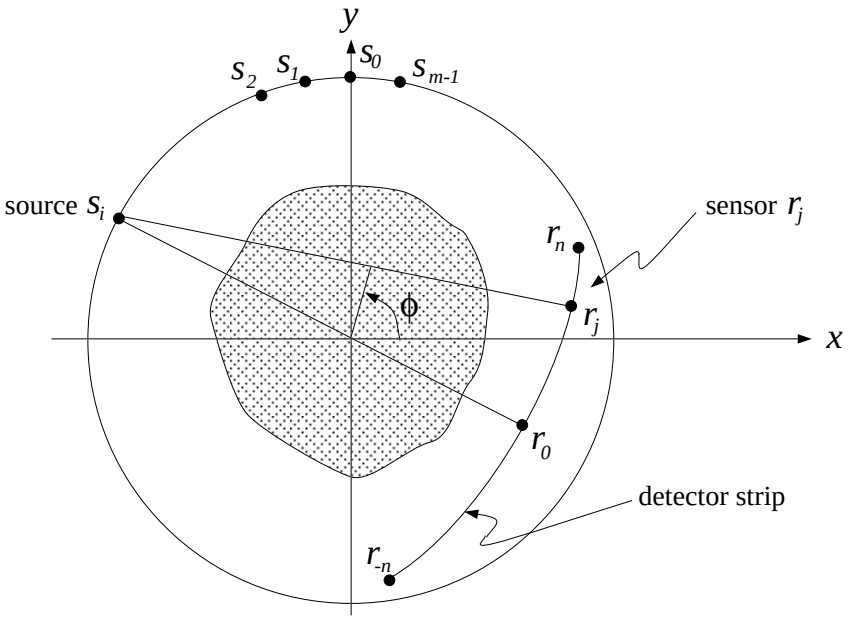


Figure 8.1.6 Fan beam with spinning scanner for data collection using a single source and multiple sensors.

There are other problems as well. According to Eq. 8.1.5, the actual measurements only approximate the corresponding line integrals. The left side of the equation corresponds more to a sum (count of the number of photons absorbed) than to an integral. Moreover, by using digital computers, only finitely many values of **a** can be reconstructed. These problems lead to the discrete treatment of the Radon transform and its inversion. The formulation of the discrete Radon transform provided below is the one most suitable for our subsequent discussions.

8.1.2 **Definition.** Let $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$, G a finite multiplicative group, $S \subset G$, and $S \cdot y = \{s \cdot y : s \in S\}$, where $y \in G$. If $\mathbf{a} : G \to \mathbb{F}$, then the *discrete Radon transform of* **a** *based on S* is defined as

$$[\mathfrak{R}(\mathbf{a})](y) = \sum_{x \in S \cdot y} \mathbf{a}(x)\,. \tag{8.1.9}$$

Note that in this definition the integral of Eq. 8.1.4 has been replaced by a sum and the line **L** by the set $S \cdot y$. In the case where G is the group $G = \mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ (with the operation of addition $mod(m, n)$) and $S \subset \mathbf{X}$ a digital line, the value $[\mathfrak{R}(\mathbf{a})](\mathbf{y})$ represents the sum over the line $S \cdot \mathbf{y}$. This

shows that Eq. 8.1.9 does indeed represent a simplified discrete version of the Radon transform. This simple version has found applications in applied statistics. For example, Diaconis [3] applied this version to the analysis of syllable counts in Plato's *Republic*. For each sentence in the book, the syllable pattern in the last five syllables was recorded. This provided a binary vector in $\mathbb{Z}_{2^5}$. A function $\mathbf{a} : \mathbb{Z}_{2^5} \to \mathbb{R}$ was defined which counts the number of sentences for each pattern. The function **a** was then analyzed (characterized) by considering the Radon transform for various choices of *S*.

It is natural to inquire about the choices of *S* that characterize the function **a** when the only available data consists of sums over the translates of *S*. We explore tis topic in the next section.

8.2 Invertibility of the Radon Transform and G-Templates.

The fundamental question that arise in connection with the discrete Radon transform is whether or not it is possible to invert it; i.e., whether one can recover (in principle) the function **a** from the data provided by $[\mathfrak{R}(\mathbf{a})]$. This question is closely related to the invertibility of *G*-templates. In this section we present necessary and sufficient conditions for the invertibility of both, the discrete Radon transform and *G*-templates.

8.2.1 **Lemma** *If* $\mathbf{t} \in G(\mathbb{C}, \mathbf{X})$*, then there exists a set* $S \subset \mathbf{X}$ *such that* $S \cdot \mathbf{y} = S(\mathbf{t}_\mathbf{y}) \ \ \forall \mathbf{y} \in \mathbf{X}$. *Conversely, given a non-empty set* $S \subset \mathbf{X}$*, then there exists a G-template* **t** *such that* $S \cdot \mathbf{y} = S(\mathbf{t}_\mathbf{y}) \ \ \forall \mathbf{y} \in \mathbf{X}$.

Proof: Let $\mathbf{t} \in G(\mathbb{C}, \mathbf{X})$, $\mathbf{x}_0 \in \mathbf{X}$ the identity of *G*, and define $S \equiv S(\mathbf{t}_{\mathbf{x}_0})$. Then

$$\begin{aligned} S \cdot \mathbf{y} &= \{\mathbf{z} \cdot \mathbf{y} \,:\, \mathbf{z} \in S\} \\ &= \{\mathbf{x} \in \mathbf{X} \,:\, \mathbf{x} = \mathbf{z} \cdot \mathbf{y},\ \mathbf{z} \in S\} \\ &= \left\{\mathbf{x} \in \mathbf{X} \,:\, \mathbf{x} \cdot \mathbf{y}^{-1} = \mathbf{z},\ \mathbf{z} \in S(\mathbf{t}_{\mathbf{x}_0})\right\} \\ &= \left\{\mathbf{x} \in \mathbf{X} \,:\, \mathbf{t}_{\mathbf{x}_0}\left(\mathbf{x} \cdot \mathbf{y}^{-1}\right) \neq 0\right\} \\ &= \{\mathbf{x} \in \mathbf{X} \,:\, \mathbf{t}_\mathbf{y}(\mathbf{x}) \neq 0\} \\ &= S(\mathbf{t}_\mathbf{y})\,. \end{aligned}$$

To prove the converse, suppose that $S \subset \mathbf{X}$ is non-empty. Define $\mathbf{a} \in \mathbb{C}^\mathbf{X}$ by

$$\mathbf{a}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in S \\ 0 & \text{if } \mathbf{x} \notin S\,. \end{cases}$$

According to Lemma **7.5.3**, there exists a *G*-template **t** such that $\mathbf{t}_{\mathbf{x}_0} = \mathbf{a}$. Hence, $S(\mathbf{t}_{\mathbf{x}_0}) = S$ and by the first part of our proof, $S \cdot \mathbf{y} = S(\mathbf{t}_\mathbf{y}) \ \ \forall \mathbf{y} \in \mathbf{X}$

Q.E.D.

Note that the image $\mathbf{a}$ defined in the proof of the Lemma is just the characteristic function on S. Hence, in view of Theorem **7.5.8**, the template $\mathbf{t}$ defined in terms of the parameter $\mathbf{a}$ has the property

$$\begin{aligned}\psi(\mathbf{t}) &= \alpha(\xi(\mathbf{t})) \\ &= \alpha\big(\mathbf{t}_{\mathbf{x}_0}(\mathbf{x}_0), \mathbf{t}_{\mathbf{x}_1}(\mathbf{x}_1), \ldots, \mathbf{t}_{\mathbf{x}_{p-1}}(\mathbf{x}_{p-1})\big) \\ &= \sum_{i=0}^{p-1} \mathbf{a}(\mathbf{x}_i)\alpha(\mathbf{x}_i) \\ &= \sum_{\mathbf{z}\in S} \alpha(\mathbf{z}),\end{aligned}$$

which corresponds to the discrete Radon transform based on S. Since $\mathbf{a}$ is determined by S and $\mathbf{t}$ is defined in terms of $\mathbf{a}$, we refer to $\mathbf{t}$ as the template *induced* by S.

The relationship between the template $\mathbf{t}$ induced by S and the discrete Radon transform is revealed by the next theorem.

8.2.2 **Theorem.** *Suppose $(\mathbf{X}, N)$ simulates G, $S \subset G$, and $\mathbf{t} \in G(\mathbb{C}, \mathbf{X})$ is induced by S. The discrete Radon transform based on S is invertible if and only if $\mathbf{t}$ is invertible.*

Proof: We identify $\mathbf{X}$ with G. Let $\mathbf{a} \in \mathbb{C}^{\mathbf{X}}$ and $\mathbf{b} = \mathbf{a} \bigoplus \mathbf{t}$. Then $[\mathfrak{R}(\mathbf{a})](\mathbf{y}) = \sum_{\mathbf{x}\in S\cdot\mathbf{y}} \mathbf{a}(\mathbf{x}) = \sum_{\mathbf{x}\in S(\mathbf{t}_\mathbf{y})} \mathbf{a}(\mathbf{x}) = \mathbf{b}(\mathbf{y})$. Thus, $\mathfrak{R}(\mathbf{a})$ is invertible if and only if the map $\mathbf{a} \mapsto \mathbf{b} = \mathbf{a} \bigoplus \mathbf{t}$ is invertible. This will be the case if and only if $\mathbf{t}$ is invertible.

Q.E.D.

Invertibility of the Radon transform is closely linked to irreducible matrices. A matrix $A \in M_{k\times k}(\mathbb{C})$ is said to be *reducible* if there exists a partition of $\mathbb{Z}_k^+$ into two non-empty subsets J_1 and J_2 (i.e. $J_1 \cap J_2 = \varnothing$ and $J_1 \cup J_2 = \mathbb{Z}_k^+$) such that $a_{ij} = 0$ wherever $i \in J_1$ and $j \in J_2$. Otherwise, A is said to be *irreducible*.

8.2.3 **Example:** Let A be a matrix of form

$$A = \begin{pmatrix} * & 0 & 0 & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & 0 & 0 & * & * \\ * & 0 & 0 & * & * \end{pmatrix}.$$

If $J_1 = \{1, 4, 5\}$ and $J_2 = \{2, 3\}$, then $J_1 \cup J_2 = \mathbb{Z}_5^+$, $J_1 \cap J_2 = \varnothing$, and $a_{ij} = 0$ wherever $i \in J_1$ and $j \in J_2$. Therefore, A is irreducible. On the other hand, the matrix

$$\begin{pmatrix} * & 0 & * & * & * \\ 0 & * & 0 & * & * \\ * & * & * & 0 & * \\ 0 & * & 0 & * & * \\ 0 & * & * & * & * \end{pmatrix}.$$

is irreducible.

Note that if P denotes the permutation matrix formed by interchanging the first and the third rows of the identity matrix I_5 and A is the matrix of the above example, then

$$PA = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & 0 & 0 & * & * \\ * & 0 & 0 & * & * \\ * & 0 & 0 & * & * \end{pmatrix}$$

and

$$PAP^{-1} = \left(\begin{array}{ccc|ccc} * & * & & * & * & * \\ * & * & & * & * & * \\ - & - & + & - & - & - \\ 0 & 0 & & * & * & * \\ 0 & 0 & & * & * & * \\ 0 & 0 & & * & * & * \end{array}\right).$$

In general, it can be shown that $A \in M_{k\times k}(\mathbb{C})$ is reducible if and only if there exists a permutation matrix P such that PAP^{-1} is of form

$$\begin{pmatrix} B & C \\ 0 & D \end{pmatrix}, \tag{8.2.1}$$

where B and D are square matrices [13]. Thus, an equivalent definition of reducibility of a matrix A is to simply require A to have the form expressed in Eq. 8.2.1.

We now turn our attention to the collection of matrices of a k–dimensional representation $\mathcal{A} = \{\alpha(\mathbf{x}) : x \in G\} \subset M_{k\times k}(\mathbb{C})$ of the group G. If the collection $\mathcal{A} = range(\alpha)$ is reducible, then each matrix $\alpha(x)$ may (by choice of an appropriate permutation) be presented in the form

$$\alpha(x) = \begin{pmatrix} \alpha_1(x) & c(x) \\ 0 & \alpha_2(x) \end{pmatrix},$$

where $\alpha_1(x)$ is an $m \times m$ matrix, $\alpha_2(x)$ is a $(k-m)\times(k-m)$ matrix, $c(x)$ is an $m \times (k-m)$ matrix, and 0 denotes the $(k-m)\times m$ zero matrix. Since $det(\alpha(x)) = det(\alpha_1(x)\cdot\alpha_2(x))$, the non-singularity of $\alpha(x)$ assumes us of the non-singularity of both $\alpha_1(x)$ and $\alpha_2(x)$ for every $x \in G$. Furthermore,

$$\alpha(x,y) = \alpha(x)\cdot\alpha(y) = \begin{pmatrix} \alpha_1(x)\cdot\alpha_1(y) & \alpha_1(x)\cdot c(y) + c(x)\cdot\alpha_2(y) \\ 0 & \alpha_2(x)\cdot\alpha_2(y) \end{pmatrix}$$

so that $\alpha_1(x,y) = \alpha_1(x)\cdot\alpha_1(y)$ and $\alpha_2(x,y) = \alpha_2(x)\cdot\alpha_2(y)$. Hence, the collection $a_1 = \{\alpha_1(x) : x \in G\}$ and $a_2 = \{\alpha_2(x) : x \in G\}$ furnish m and $(k-m)$–dimensional representations of our group G, respectively.

The following definitions are expressed in terms of group representations but can just as easily be expressed in terms of representations of group algebras.

8.2.4 **Definition.** Let $\alpha : G \to M_{k\times k}(\mathbb{C})$ be a representation of G. We say that α is *reducible* if and only if there exists a non-singular matrix $M \in M_{k\times k}(\mathbb{C})$ such that

$$\alpha(x) = M\begin{pmatrix} \alpha_1(x) & c(x) \\ 0 & \alpha_2(x) \end{pmatrix}M^{-1}.$$

If α is not reducible, then we say that α is *irreducible*.

8.2.5 **Definition.** Two k–dimensional representations α and β of G are said to be *similar* if and only if there exists an invertible matrix $M \in M_{k\times k}(\mathbb{C})$ such that $\alpha(x) = M\beta(x)M^{-1}\ \forall x \in G$.

It is well-known [9, 11] that if $\alpha : G \to M_{k\times k}(\mathbb{C})$ is a right regular representation of G, then there exists a non-singular matrix M and irreducible representations $\alpha_1, \alpha_2, \ldots, \alpha_m$ of G such that for every $x \in G$

$$\alpha(x) = M \begin{pmatrix} \alpha_1(x) & 0 & \cdots & 0 \\ 0 & \alpha_2(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_m(x) \end{pmatrix} M^{-1}.$$

Moreover, if β is any irreducible representation of G, then β is similar to α_i for some $i \in \mathbb{Z}_m^+$. The same statement holds true for $\mathbb{C}(G)$ as well.

The next result relates the invertibility of the Radon transform to irreducible representations.

8.2.6 **Theorem.** *Let $S \subset G$ and suppose that $\mathbf{t} \in G(\mathbb{C}, \mathbf{X})$ is induced by S. The discrete Radon transform based on S is invertible if and only if $\beta(\xi(\mathbf{t}))$ is invertible for every irreducible representation β of $\mathbb{C}(G)$.*

Proof: Let $\xi(\mathbf{t}) = \mathbf{a}$ and β be an irreducible representation of $\mathbb{C}(G)$. If the discrete Radon transform based on S is invertible, then according to Theorem **8.2.2** $\mathbf{t}$ is also invertible. Let α be a right regular representation of G and $\psi : \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}} \to M_{p\times p}(\mathbb{C})$ be the isomorphism defined relative to the same ordering of G as α. by Theorem**7.5.8**, $\alpha(\mathbf{a}) = \psi(\mathbf{t})$. Since $\psi(\mathbf{t})$ is invertible, we now know that $\alpha(\mathbf{a})$ is also invertible. By our observation following Definition **8.2.5**, there exists a non-singular matrix $M \in M_{p\times p}(\mathbb{C})$ and irreducible representations $\alpha_1, \alpha_2, \ldots, \alpha_m$ of $\mathbb{C}(G)$ such that

$$\alpha(\mathbf{a}) = M \begin{pmatrix} \alpha_1(\mathbf{a}) & & & 0 \\ & \alpha_2(\mathbf{a}) & & \\ & & \ddots & \\ 0 & & & \alpha_m(\mathbf{a}) \end{pmatrix} M^{-1}$$

and $\beta = M_j \alpha_j M_j^{-1}$ for some $j \in \mathbb{Z}_m^+$. Since $\alpha(\mathbf{a})$ is invertible, $\alpha_i(\mathbf{a})$ is invertible for each $i \in \mathbb{Z}_m^+$. Therefore, $\beta(\xi(\mathbf{t})) = \beta(\mathbf{a}) = M_j \alpha_j(\mathbf{a}) M_j^{-1}$ is invertible.

Conversely, suppose that $\beta(\xi(\mathbf{t}))$ is invertible for every irreducible representation β of $\mathbb{C}(G)$. According to our observation

$$\psi(\mathbf{t}) = \alpha(\xi(\mathbf{t})) = M \begin{pmatrix} \alpha_1(\xi(\mathbf{t})) & & & 0 \\ & \alpha_2(\xi(\mathbf{t})) & & \\ & & \ddots & \\ 0 & & & \alpha_m(\xi(\mathbf{t})) \end{pmatrix} M^{-1},$$

where M is non-singular and each α_i is an irreducible representation of $\mathbb{C}(G)$. Thus, each $\alpha_i(\xi(\mathbf{t}))$ is invertible, which implies that $\psi(\mathbf{t})$ is invertible. Since ψ is an isomorphism, we

have that $\mathbf{t}$ is invertible. Therefore, by Theorem **8.2.2** the discrete Radon transform based on S is invertible.

Q.E.D.

Note that the proof of Theorem **8.2.6** yields the more general result given by the following corollary:

8.2.7 **Corollary.** *$\mathbf{t} \in G(\mathbb{C}, \mathbf{X})$ is invertible if and only if $\beta(\xi(\mathbf{t}))$ invertible for every irreducible representation β of $\mathbb{C}(G)$.*

8.2.8 **Definition.** If α is a representation of G, then the *contragredient representation* of α is the representation $\hat{\alpha}$ defined by $\hat{\alpha}(x) = \left[\alpha\left(x^{-1}\right)\right]'$.

It follows from the definition that $\alpha(x) = \left[\hat{\alpha}\left(x^{-1}\right)\right]'$ and that every representation is the contragredient representation of some representation.

8.2.9 **Theorem.** *Suppose α is an irreducible representation of $\mathbb{C}(G)$, $S \subset G$, and $\mathbf{t} \in G(\mathbb{C}, \mathbf{X})$ is induced by S. The matrix $\alpha(\xi(\mathbf{t}))$ is invertible if and only if $\sum_{x \in S} \hat{\alpha}\left(x^{-1}\right)$ is invertible.*

Proof: Let $\mathbf{a} = \xi(\mathbf{t})$. In view of our observation following Lemma **8.2.1** we can write $\alpha(\mathbf{a}) = \sum_{i=0}^{p-1} \mathbf{a}(\mathbf{x}_i)\alpha(\mathbf{x}_i)$. Furthermore, since $\mathbf{a} : \mathbf{X} \to \mathbb{C}$ corresponds to the characteristic function on S, $\alpha(\mathbf{a}) = \sum_{x \in S} \alpha(x)$. Therefore, $\alpha(\xi(\mathbf{t})) = \alpha(\mathbf{a})$ is invertible if and only $\sum_{x \in S} \alpha(x)$ is invertible. Moreover, $\sum_{x \in S} \alpha(x)$ is invertible if and only if $\sum_{x \in S} \left[\hat{\alpha}\left(x^{-1}\right)\right]' = \left[\sum_{x \in S} \hat{\alpha}\left(x^{-1}\right)\right]'$ is invertible. However, $\left[\sum_{x \in S} \hat{\alpha}\left(x^{-1}\right)\right]'$ is invertible if and only if $\sum_{x \in S} \hat{\alpha}\left(x^{-1}\right)$ is invertible.

Q.E.D.

As an immediate consequence, we now have the following result:

8.2.10 **Corollary.** *If $S \subset G$, then the discrete Radon transform based on S is invertible if and only if for every irreducible representation α of G, the matrix $\sum_{x \in S} \alpha\left(x^{-1}\right)$ is invertible.*

Proof: By Theorem 8.2.6, the discrete Radon transform based on S is invertible if and only if for every irreducible representation $\hat{\alpha}$ of G, the matrix $\hat{\alpha}(\xi(\mathbf{t}))$ is invertible. But by Theorem **8.2.9** this will happen if and only if $\sum_{x \in S} \alpha\left(x^{-1}\right)$ is invertible.

Q.E.D.

This corollary was first proved by Diaconis and Graham using different techniques [4].

The results developed in this section show that the invertibility of the discrete Radon transform is closely linked to the invertibility of G–templates. In the case of circulant templates the irreducible representations of abelian groups are all one-dimensional. As an example consider the group algebra $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ over $\mathbb{C}$. If $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ and α is a right regular representation, then according to Theorem **7.5.8** and Eqs. 7.2.4 and 7.3.1,

$$\alpha(\xi(\mathbf{t})) = \psi(\mathbf{t}) = F_{m\otimes n} D F^*_{m\otimes n}, \tag{8.2.2}$$

where $D = diag\Big(p_{\mathbf{t}_0}\Big(\omega_m^j, \omega_n^k\Big)\Big)$. Thus, the irreducible representation of $\mathbf{X}$ over $\mathbb{C}$ are the mn elements of the set $\Big\{\omega_m^j \omega_n^k : j, k \in \mathbb{Z}\Big\}$.

If $D = diag(d_1, d_2, \ldots, d_p)$ and D is invertible, then $D^{-1} = diag\Big(\frac{1}{d_1}, \frac{1}{d_2}, \ldots, \frac{1}{d_p}\Big)$. Therefore, $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ is invertible if and only if $p_{\mathbf{t}_0}\Big(\omega_m^j, \omega_n^k\Big) \neq 0$ for every j and k.

8.2.11 Example: If $\mathbf{t}$ denotes the *circulant* Moore template

$\mathbf{t} =$

1	1	1
1	1	1
1	1	1

,

then

$$\begin{aligned} p_{\mathbf{t}_0}(x, y) &= 1 + y + y^{n-1} + x\big(1 + y + y^{n-1}\big) + x^{m-1}\big(1 + y + y + y^{n-1}\big) \\ &= \big(1 + x + x^{m-1}\big)\big(1 + y + y^{n-1}\big). \end{aligned}$$

Let $p(x) = 1 + x + x^{n-1}$ and $q(y) = 1 + y + y^{n-1}$. Then $p\Big(\omega_m^j\Big) = 0 \iff \cos\Big(\frac{2\pi j}{m}\Big) = -\frac{1}{2} \iff \frac{m}{3} \in \mathbb{Z}$. Similarly, $q\big(\omega_n^k\big) = 0 \iff \frac{n}{3} \in \mathbb{Z}$. Therefore, $\mathbf{t}$ is invertible if and only if m and n are not divisible by 3. Hence, the invertibility of the averaging template *depends on the dimension* of the array $\mathbf{X}$!

The inverse of the circulant averaging template given in the above example can be found explicitly using the following method:

Let C_m denote the set of $m \times m$ circulant matrices with each $C = circ(c_0, c_1, \ldots, c_{m-1})$ associate a polynomial $c(x) = c_0 + c_1 x + \ldots + c_{m-1} x^{m-1}$. Now define $h : C_m \to \mathbb{R}[x]/(x^m - 1)$ by $h(C) = c(x)$. It is not difficult to show that h is a ring isomorphism. Thus,

$$h^{-1}(p(x)) = h^{-1}\big(1 + x + x^{m-1}\big) \in C_m.$$

The inverse $circ(c_0, c_1, \ldots, c_{m-1}) = \big[h^{-1}\big(1 + x + x^{m-1}\big)\big]^{-1}$ is given by

$$c_j = \frac{\sin(j\theta) + \sin[(m-j)\theta]}{2\sin\theta(1 - \cos(m\theta))},$$

where $\theta = \frac{2\pi}{3}$ [2]. For example, if $\mathbf{Y} = \mathbb{Z}_4 \times \mathbb{Z}_4$, then

$$p_{\mathbf{t}_0}(x,y) = \left(1 + x + x^3\right)\left(1 + y + y^3\right)$$

and

$$\left[h^{-1}\left(1 + x + x^3\right)\right]^{-1} = circ\left(\frac{1}{3}, \frac{1}{3}, -\frac{2}{3}, \frac{1}{3}\right).$$

Therefore, $h\left[circ\left(\frac{1}{3}, \frac{1}{3}, -\frac{2}{3}, \frac{1}{3}\right)\right] = \frac{1}{3}\left(1 + x - 2x^2 + x^3\right)$ and
$p_{\mathbf{t}_0^{-1}}(x,y) = \frac{1}{9}\left(1 + x - 2x^2 + x^3\right)\left(1 + y - 2y^2 + y^3\right)$. The template $\mathbf{t}^{-1}$ is shown in Fig. **8.2.1**.

$\mathbf{t}_0^{-1} =$

1	1	-2	1
1	1	-2	1
-2	-2	4	-2
1	1	-2	1

Figure 8.2.1 The inverse of the circulant 3×3 averaging template **t**.

Let $F_{\mathbf{t}} : \mathbb{R}^{\mathbf{X}} \to \mathbb{R}^{\mathbf{X}}$ be defined by $F_{\mathbf{t}}(\mathbf{a}) = \mathbf{a} \oplus \mathbf{t}$. If $\mathbf{t}$ is invertible, then since

$$F_{\mathbf{t}}\left(\mathbf{a} \oplus \mathbf{t}^{-1}\right) = \left(\mathbf{a} \oplus \mathbf{t}^{-1}\right) \oplus \mathbf{t} = \mathbf{a} = (\mathbf{a} \oplus \mathbf{t}) \oplus \mathbf{t}^{-1} = F_{\mathbf{t}^{-1}}(\mathbf{a} \oplus \mathbf{t}),$$

we obtain the commutative diagram shown in **8.2.2**.

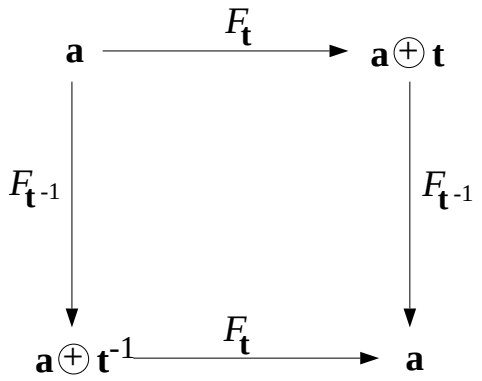


Figure 8.2.2 Recovering $\mathbf{a}$ from $\mathbf{a} \oplus \mathbf{t}$ and $\mathbf{a} \oplus \mathbf{t}^{-1}$.

Figure **8.2.3** provides an example of this commutative property. Here, $\mathbf{X} = \mathbb{Z}_{64} \times \mathbb{Z}_{64}$ with $\mathbf{t}$ corresponding to the 3×3 circulant averaging template. It may be somewhat surprising that the extremely noisy looking image $\mathbf{a} \oplus \mathbf{t}^{-1}$ (lower left hand image of Fig. **8.2.3**), when locally averaged, results in the image of the SR71 spy plane. This surprise dissipates in view of the fact that $\mathbf{a} \oplus \mathbf{t}^{-1}$ contains both positive and negative values. In order to display $\mathbf{a} \oplus \mathbf{t}^{-1}$, the pixel values have been linearly shifted

by the amount $|\wedge \mathbf{a}|$. Thus, the image displayed in the lower left hand corner is not really $\mathbf{a} \oplus \mathbf{t}^{-1}$, but $\mathbf{a} \oplus \mathbf{t}^{-1} + |\wedge \mathbf{a}|$.

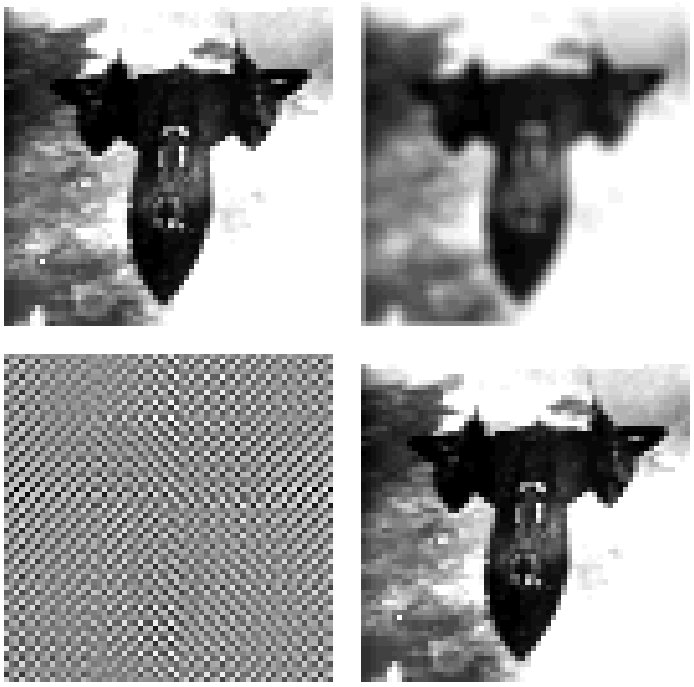


Figure 8.2.3 Inversion of a locally averaged image.

Suppose $\mathbf{t}$ is a circulant $\mathbb{Z}_2$–valued template. According to Theorem **8.2.2**, the transformation $F_{\mathbf{t}} : \mathbf{a} \rightarrowtail \mathbf{a} \oplus \mathbf{t}$ is invertible if and only if the Radon transform $\mathfrak{R}(\mathbf{a})$ based on $S = S(\mathbf{t_0})$ is invertible. Furthermore,

$$[\mathfrak{R}(\mathbf{a})](\mathbf{y}) = \sum_{\mathbf{x} \in S \cdot \mathbf{y}} \mathbf{a}(\mathbf{x}) = \sum_{\mathbf{x} \in S(\mathbf{t_y})} \mathbf{a}(\mathbf{x}) \mathbf{t_y}(\mathbf{x}) \; \forall \mathbf{y} \in \mathbf{X}.$$

As we have seen in Example **8.2.11**, if $S \cdot \mathbf{y}$ denotes the Moore neighborhood of $\mathbf{y}$, then the inversion if fairly straightforward. In general, however, proving invertibility can be rather complicated even for simple neighborhood configuration for averaging an image, the invertibility is not as easily established. Specifically, if

$$\mathbf{t_y} = \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & 1 & 1 \\ \hline & 1 & \\ \hline \end{array} \; ,$$

then $p_{\mathbf{t_0}}(x, y) = 1 + y + y^{n-1} + x + x^{m-1}$. For $m = n = 5$, we have

$$\begin{aligned} p_{\mathbf{t_0}}\left(\omega_5, \omega_5^2\right) &= 1 + \omega_5^2 + \omega_5^8 + \omega_5 + \omega_5^4 \\ &= 1 + \omega_5^2 + \omega_5^3 + \omega_5 + \omega_5^4 \\ &= 0. \end{aligned}$$

More generally,

$$p_{\mathbf{t_0}}\left(\omega_m^j, \omega_n^i\right) = 1 + 2\cos\left(\frac{2\pi j}{m}\right) + 2\cos\left(\frac{2\pi i}{n}\right).$$

Thus,

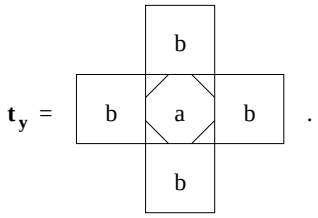
$$p_{\mathbf{t_0}}\left(\omega_m^j, \omega_n^i\right) = 0 \iff \cos\left(\frac{2\pi j}{m}\right) + \cos\left(\frac{2\pi i}{n}\right) = -\frac{1}{2}.$$

Therefore, if $m = n = 6$, then $p_{\mathbf{t_0}}\left(\omega_6, \omega_6^3\right) = 0$. It follows from those observations that if $m = n$ and 5 or 6 divides m, then the transform is not invertible.

The obvious question one may ask is: "Does it follow that whenever $p_{\mathbf{t_0}}\left(\omega_m^j, \omega_m^k\right) = 0$, then 5 or 6 divides m?" This question was answered in the affirmative by Z. Manseur and D. Wilson [12]. In Fact, Wilson and Manseur proved invertibility for a more general class of von Neumann neighborhood based templates. They considered the template $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ defined by

$$\mathbf{t_y}(\mathbf{x}) = \begin{cases} a & \text{if } \mathbf{x} = \mathbf{y} \\ b & \text{if } \|\mathbf{x} - \mathbf{y}\| = 1 \text{ or } m-1 \\ 0 & \text{otherwise,} \end{cases} \tag{8.2.3}$$

where $a, b \in \mathbb{R}$ with $b \neq 0$ and $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_m$. Pictorially, $\mathbf{t}$ has representation



Their results are summarized by the following theorem:

8.2.12 Theorem. *(Manseur and Wilson) Let* $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_m$ *and* $\mathbf{t} \in C(\mathbb{C}, \mathbf{X})$ *be defined by Eq. 8.2.3.*

1. *If* $a = b = 1$, *then* $\mathbf{t}$ *fails to be invertible if and only if either 5 or 6 divides* m.
2. *If* $a = 0$ *and* $b \neq 0$, *then* $\mathbf{t}$ *is invertible* $\iff m$ *is odd.*
3. *If* $\left|\frac{a}{b}\right| > 4$, *then* $\mathbf{t}$ *is invertible.*

The proofs of statements 2 and 3 follow the argument used to prove the invertibility of the Moore template. However, the proof of statement 1 is rather involved. Note also that the hypothesis requires a square array. It is still an open question whether or not similar results hold for rectangular (i.e. not necessarily square) arrays.

Another requirement in the above theorem is the symmetry of the pixel values about the center point $\mathbf{y}$. The invertibility of von Neumann templates having non-symmetric pixel values remains another open question.

8.3 Determinants and Inversion

Determinants can be used to establish the invertibility of a template. Specifically, if $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ and $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$, then $\mathbf{t}$ is invertible if and only if $det(\psi(\mathbf{t})) \neq 0$. Determinants can also be used to provide explicit expressions for the inverse of a given template.

If $A \in M_{mn \times mn}(\mathbb{R})$, then the determinant $det(A - \lambda I_{mn})$ is given by

$$(-1)^{mn}\lambda^{mn} + b_1\lambda^{mn-1} + \cdots + b_{mn-1}\lambda + b_{mn} = 0\,. \tag{8.3.1}$$

The term b_{mn} is obtained by setting $\lambda = 0$ and is, therefore, equal to $det(A)$. Multiplying Eq. 8.3.1 by $(-1)^{mn}$ results in the polynomial

$$p(\lambda) = \lambda^{mn} + c_{mn-1}\lambda^{mn-1} + \cdots + c_1\lambda + c_0\,, \tag{8.3.2}$$

where $c_0 = (-1)^{mn}det(A)$ and

$$c_{mn-1} = trace(A) = -(a_{11} + a_{22} + \cdots + a_{mn,mn}).$$

The polynomial $p(\lambda)$ corresponds to the characteristic polynomial of *A* (Section **3.8**).

8.3.1 **Theorem.** *Suppose* $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ *and* $A = \psi(\mathbf{t})$. *If* $det(A) \neq 0$, *then* $\mathbf{t}$ *is invertible and*

$$\mathbf{t}^{-1} = -\frac{1}{c_0}\left(\mathbf{t}^{mn-1} + c_{mn-2}\mathbf{t}^{mn-2} + \cdots + c_1 1\right),$$

where $\mathbf{t}^k = \bigoplus_{i=0}^{k} \mathbf{t}$ *and* $\mathbf{t}^0 = 1$.

Proof: It follows from the Cayley-Hamilton Theorem [8] that *A* satisfies its characteristic polynomial. Therefore,

$$p(A) = A^{mn} + c_{mn-1}A^{mn-1} + \cdots + c_1 A + c_0 I_{mn} = O_{mn}\,.$$

Since $det(A) \neq 0$, $c_0 \neq 0$. Hence,

$$I_{mn} = A \cdot \left[-\frac{1}{c_0}\left(A^{mn-1} + c_{mn-1}A^{mn-2} + \cdots + c_1 I_{mn}\right)\right],$$

and

$$A^{-1} = -\frac{1}{c_0}\left(A^{mn-1} + c_{mn-1}A^{mn-2} + \cdots + c_1 I_{mn}\right). \tag{8.3.3}$$

Since ψ is an isomorphism, $\psi^{-1}\left(A^{-1}\right) = \mathbf{t}^{-1}$ and $\psi^{-1}\left(A^k\right) = \mathbf{t}^k$ for $k = 0, 1, \ldots, mn-1$. The result now follows immediately by applying ψ^{-1} to Eq. 8.3.3.

Q.E.D.

In contrast to Section **8.2**, the templates considered here are translation invariant but not circulant. According to Corollary **7.1.7**, the inverse of a translation invariant template is not necessarily translation

invariant. Translation variant template operations run notoriously slow on current computer architectures. The above theorem provides a way to write the inverse of a translation invariant template in terms of invariant templates. However, speed-up in performance can generally not be achieved by simply replacing $\mathbf{a} \bigoplus \mathbf{t}^{-1}$ with

$$-\frac{1}{c_0}\left(\mathbf{a} \bigoplus \mathbf{t}^{mn-1}\right) + c_{mn-2}\left(\mathbf{a} \bigoplus \mathbf{t}^{mn-2}\right) + \cdots + c_1 \cdot \mathbf{a}$$

due to the increase in the number of convolutions that must be performed. Fortunately, it is often possible to decompose the polynomial representation of a template into fewer products. For example, using polynomial factorization, the polynomial

$$p(\mathbf{t}) = \mathbf{t}^3 + c_2\mathbf{t}^2 + c_1\mathbf{t} + c_0$$

can be factored as

$$p(\mathbf{t}) = \mathbf{t}[\mathbf{t}(\mathbf{t} + c_2) + c_1] + c_0\,.$$

Thus, the number of convolution products can be reduced by one-half, from 6 to 3.

A classical method of evaluating determinants of large matrices is by cofactor expansion. The next theorem is a consequence of the use of cofactor expansion on certain types of block Toeplitz matrices.

8.3.2 Theorem. *If A and B are $n \times n$ matrices and M_m is an $mn \times mn$ block Toeplitz matrix of form*

$$M_m = \begin{pmatrix} O & A & O & O & \cdots & O & O \\ B & O & A & O & \cdots & O & O \\ O & B & O & A & \cdots & O & O \\ O & O & B & O & \cdots & O & O \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & O & \cdots & O & A \\ O & O & O & O & \cdots & B & O \end{pmatrix},$$

then $det(M_m) = (-1)^n det(A) \cdot det(B) \cdot det(M_{m-2})$.

Proof:

$$det(M_m) = (-1)^n det\begin{pmatrix} B & O & A & O & \cdots & O \\ O & A & O & O & \cdots & O \\ O & B & O & A & \cdots & O \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$= (-1)^n det(B) \cdot det \underbrace{\begin{pmatrix} A & O & O & O & \cdots & O \\ B & O & A & O & \cdots & O \\ O & B & O & A & \cdots & O \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}}_{m-1 \ blocks}$$

$$= (-1)^n det(B) \cdot det(A) \cdot det \underbrace{\begin{pmatrix} O & A & O & O & \cdots & O \\ B & O & A & O & \cdots & O \\ O & B & O & A & \cdots & O \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}}_{m-2 \ blocks}$$

Q.E.D.

8.3.3 **Corollary.** *If A_n is an $n \times n$ tridiagonal matrix of form*

$$A_n = \begin{pmatrix} 0 & a & 0 & 0 & \cdots & 0 & 0 \\ b & 0 & a & 0 & \cdots & 0 & 0 \\ 0 & b & 0 & a & \cdots & 0 & 0 \\ 0 & 0 & b & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & a \\ 0 & 0 & 0 & 0 & \cdots & b & 0 \end{pmatrix},$$

then $det(A_n) = -ab \cdot det(A_{n-2})$.

Applying the reduction formula given by the above corollary to the matrices

$$A_1 = (0),\ A_2 = \begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix},\ A_3 = \begin{pmatrix} 0 & a & 0 \\ b & 0 & a \\ 0 & b & 0 \end{pmatrix},\ \text{and}\ A_4 = \begin{pmatrix} 0 & a & 0 & 0 \\ b & 0 & a & 0 \\ 0 & b & 0 & a \\ 0 & 0 & b & 0 \end{pmatrix}$$

we obtain

$$det(A_1) = 0,\ det(A_2) = -ab,\ det(A_3) = -ab \cdot det(A_1) = 0,$$

and $det(A_4) = -ab \cdot det(A_2) = (ab)^2$, respectively.

More generally, we have that

$$det(A_n) = \begin{cases} 0 & \text{if } n \text{ is odd} \\ (-ab)^{n/2} & \text{if } n \text{ is even}. \end{cases} \tag{8.3.4}$$

The above discussion provides a quick method for the determination of the invertibilty of certain templates. For example, consider the template $\mathbf{r} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ shown in Figure **8.3.1**.

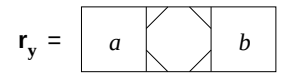


Figure 8.3.1 A template which is invertible on an array with an even number of columns.

If $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$, then the matrix $\psi(\mathbf{r})$ is the $mn \times mn$ block Toeplitz matrix given by

$$\psi(\mathbf{r}) = \begin{pmatrix} A_n & O & \cdots & O \\ O & A_n & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & A_n \end{pmatrix}, \tag{8.3.5}$$

where A_n is the $n \times n$ matrix

$$A_n = \begin{pmatrix} 0 & a & 0 & 0 & \cdots & 0 & 0 \\ b & 0 & a & 0 & \cdots & 0 & 0 \\ 0 & b & 0 & a & \cdots & 0 & 0 \\ 0 & 0 & b & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & a \\ 0 & 0 & 0 & 0 & \cdots & b & 0 \end{pmatrix}.$$

Since $det(A_n) \neq 0$ if and only if *n* is even and $det(\psi(\mathbf{r})) = [det(A_n)]^m$, the template **r** is invertible if and only if $a \cdot b \neq 0$ and the number of columns of the array **X** is even.

An analogous argument will show that the template **s** defined in Figure **8.3.2** is invertible if and only if $a \cdot b \neq 0$ and the number of rows of **X** is even.

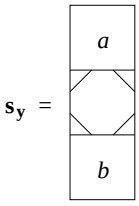


Figure 8.3.2 A template which is invertible on an array with an even number of rows.

Using the same reasoning also shows that the templates shown in Figure **8.3.3** are never invertible.

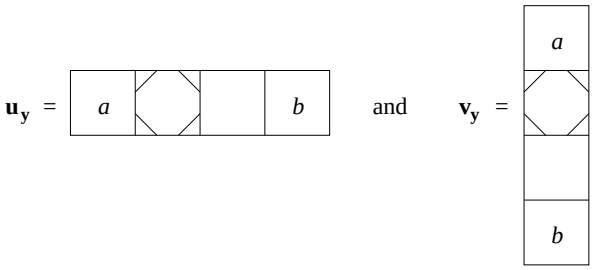
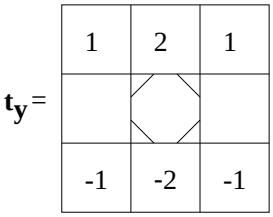
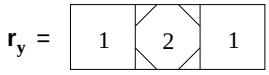


Figure 8.3.3 Two non-inveritible templates.

As another example, consider the Sobel edge template



Now $\mathbf{t} = \mathbf{r} \oplus \mathbf{s}$, where

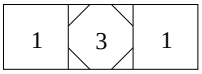


and **s** is the template shown in Figure **8.3.2** with $a = 1$ and $b = -1$. The matrix $\psi(\mathbf{r})$ is given by Eq. 8.3.5, where

$$A_n = \begin{pmatrix} 2 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 2 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 2 \end{pmatrix}.$$

It is well-known that A_n has *n* distinct eigenvalues strictly between 0 and 4 ([14], p. 101). Thus, $det(A_n) \neq 0$ for all *n* and, hence, $det(\psi(\mathbf{r})) = [det(A_n)]^m \neq 0$ for all *m* and *n*. Therefore, **r** is always invertible. Since **s** is invertible whenever the number of rows of **X** is even, the Sobel template is invertible if and only if the number of rows of **X** is even.

The matrix $\psi(\mathbf{r})$ in the above discussion is almost diagonally dominant. Recall that an $n \times n$ matrix $A = (a_{ij})$ is *diagonally dominant* if and only if $|a_{ii}| > \sum_{i \neq j} |a_{ij}|$ for all $i = 1, 2, \ldots, n$. Since diagonally dominant matrices are always invertible, templates whose corresponding matrices are diagonally dominant must also be invertible. In particular, the template

$\mathbf{t}_y =$ 

is invertible.

Our next result is also a consequence of straight forward cofactor expansion and induction.

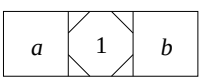
8.3.4 **Theorem.** *If A_n is an $n \times n$ tridiagonal matrix of form*

$$A_n = \begin{pmatrix} 1 & a & 0 & 0 & \cdots & 0 & 0 \\ b & 1 & a & 0 & \cdots & 0 & 0 \\ 0 & b & 1 & a & \cdots & 0 & 0 \\ 0 & 0 & b & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & a \\ 0 & 0 & 0 & 0 & \cdots & b & 1 \end{pmatrix},$$

then $det(A_n) = det(A_{n-1}) - ab \cdot det(A_{n-2})$.

Note that if A_n is as above, then $det(A_1) = 1$, $det(A_2) = 1 - ab$, and $det(A_3) = 1 - 2ab$.

8.3.5 **Theorem.** *Suppose $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$, $\mathbf{r} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ is defined by*

$\mathbf{r}_y =$ 

and A_n is as in Theorem **8.3.4**. *Then* **r** *is invertible if and only if $det(A_n) \neq 0$.*

Proof: The matrix $\psi(\mathbf{r})$ is an $mn \times mn$ block Toeplitz matrix of form

$$\psi(\mathbf{r}) = \begin{pmatrix} A_n & O & \cdots & O \\ O & A_n & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & A_n \end{pmatrix}.$$

The result now follows from the fact that $det(\psi(\mathbf{r})) = [det(A_n)]^m \neq 0 \Leftrightarrow det(A_n) \neq 0$.

Q.E.D.

As an easy consequence we have:

8.3.6 Corollary. *Suppose* $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$.
1. If $\mathbf{r} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ *is defined by*

$\mathbf{r}_y =$

1	1	1

then **r** *is invertible if and only if* $n \neq 2 mod 3$.

2. If $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ *is defined by*

$\mathbf{t} =$

1	1	1
1	1	1
1	1	1

,

then **t** *is invertible if and only if* $n \neq 2 mod 3$ *and* $m \neq 2 mod 3$.

Proof: (1) By Theorem **8.3.5**, $det(\psi(\mathbf{r})) \neq 0 \Leftrightarrow det(A_n) \neq 0$, where

$$A_n = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix}.$$

Let $y_n = det(A_n)$. Then by Theorem **8.3.4**, $y_n = y_{n-1} - y_{n-2}$, where $y_1 = 1$, $y_2 = 0$, and $y_3 = -1$.

It follows by induction that $y_{3k-1} = 0$, $y_{3k} = -y_{3k+1}$, and $y_{3k} = \pm 1$ with $y_{3k} = 1$ if k is even and $y_{3k} = -1$ if k is odd. For suppose the result holds for some integer k. According to Theorem **8.3.4**,

$$y_{3(k+1)} = y_{3(k+1)-1} - y_{3(k+1)-2} = y_{3k+2} - y_{3k+1} = (y_{3k+1} - y_{3k}) - y_{3k+1} = -y_{3k}.$$

Thus, $y_{3(k+1)} = -(\pm 1)$. Similarly,

$$y_{3(k+1)+1} = y_{3(k+1)} - y_{3(k+1)-1} = (y_{3(k+1)-1} - y_{3(k+1)}) - y_{3(k+1)-1} = -y_{3(k+1)},$$

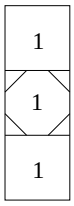
and

$$y_{3(k+1)-1} = y_{3(k+1)-2} - y_{3(k+1)-3} = -y_{3k-1} = 0.$$

Therefore, $y_n = 0 \quad \Leftrightarrow \quad n = 2 mod 3$.

(2) A similar argument shows that the template

$\mathbf{s_y} =$



is invertible if and only if $m \neq 2 mod 3$.

Since $\mathbf{t} = \mathbf{r} \oplus \mathbf{s}$, and $det(\psi(\mathbf{t})) = det(\psi(\mathbf{r})) \cdot det(\psi(\mathbf{s}))$, $\mathbf{t}$ is invertible if and only if $m \neq 2 mod 3$ and $n \neq 2 mod 3$.

Q.E.D.

The techniques discussed in this section can be used to determine whether a template is invertible. However, with the exception of Theorem **8.3.1**, they do not suggest a method for finding the inverse of a template. Methods for inverting templates are discussed in the next section.

8.4 A Class of Easily Invertible Templates

The preceding sections suggest the obvious: template inversion is — in general — a difficult and computationally inefficient process. Furthermore, the inverse of a translation invariant template is generally translation variant. The main objective of this section is to describe a large class of templates that can be easily inverted and have the additional property that their inverses can be factored into a small number of invariant templates. Moreover, the support of each template will have the same geometric configuration as the original template.

In order to establish the main goal of this section it will be useful to recall some elementary facts from linear algebra.

8.4.1 **Definition.** Let $R[x_1, x_2, \ldots, x_n]$ be a polynomial ring. The polynomial $p \in R[x_1, x_2, \ldots, x_n]$ is said to be *symmetric* if and only if

$$p(x_1, x_2, \ldots, x_n) = p\big(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}\big)$$

for every permutation $\sigma : \mathbb{Z}_n^+ \to \mathbb{Z}_n^+$.

There are *n* special symmetric polynomials with integer coefficients, called the *elementary symmetric functions of degree k*:

$$p(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \cdots + x_n = \sum_{i=1}^{n} x_i$$

$$p(x_1, x_2, \ldots, x_n) = x_1x_2 + x_1x_3 + \cdots + x_{n-1}x_n = \sum_{i<j} x_ix_j$$

$$p(x_1, x_2, \ldots, x_n) = x_1x_2x_3 + x_1x_3x_4 + \cdots + x_{n-2}x_{n-1}x_n = \sum_{i<j<k} x_ix_jx_k$$

$$\vdots$$

$$p(x_1, x_2, \ldots, x_n) = x_1x_2x_3 \cdots x_n$$

In particular, the elementary symmetric function of degree $k = 1, 2, \ldots, n$ is of form

$$p(x_1, x_2, \ldots, x_n) = \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} x_{i_1}x_{i_2} \cdots x_{i_k}.$$

If $c_1, c_2 \in \mathbb{C}$ and A is an $n \times n$ matrix, then the equation

$$(I_n + c_1A)(I_n + c_2A) = I_n + (c_1 + c_2)A + c_1c_2A^2 \tag{8.4.1}$$

is an immediate consequence of the associative and distributive laws of matrix multiplication. Equivalently, we have

$$\prod_{i=1}^{2} (I_n + c_iA) = I_n + p_1(c_1, c_2)A + p_2(c_1, c_2)A^2, \tag{8.4.2}$$

where p_1 and $p_2$are elementary symmetric functions of degree 1 and 2, respectively. An easy extension of Eq. 8.4.2 proves the following fact:

8.4.2 **Theorem.** *If $\{c_1, c_2, \ldots, c_n\} \subset \mathbb{C}$ and A is an $n \times n$ matrix, then*

$$\prod_{i=1}^{n} (I_n + c_iA) = I_n + p_1(c_1, \ldots, c_n)A + p_2(c_1, \ldots, c_n)A^2 + \cdots + p_n(c_1, \ldots, c_n)A^n,$$

where $p_k(c_1, \ldots, c_n)$ is an elementary function of degree k for $k = 1, 2, \ldots, n$.

An $n \times n$ matrix $A = (a_{ij})_{n \times n}$ is called *upper triangular* if $a_{ij} = 0$ whenever $i < j$ and *strictly upper triangular* $a_{ij} = 0$ whenever $i \le j$. The matrix *A* is said to be *nilpotent* if there exists an integer *k* such that $A^k = O_n$. If *k* is the smallest such integer, then *A* is said to be *nilpotent of order k*. With these notions in mind, the next theorem becomes a routine exercise.

8.4.3 **Theorem.** *If A is a strictly upper triangular $n \times n$ matrix, then A is nilpotent of order n. Moreover, the product of any n strictly upper triangular $n \times n$ matrices is the zero matrix.*

The next result follows immediately from this theorem.

8.4.4 **Corollary.** *If A is a block matrix with strictly upper triangular $n \times n$ blocks, then A is nilpotent of order n.*

8.4.5 **Theorem.** *If $\omega_1, \omega_2, \ldots, \omega_n$ denotes the list of the distinct nth roots of unity, then $p_k(\omega_1, \omega_2, \ldots, \omega_n) = 0$ for $k = 1, 2, \ldots, n-1$, where p_k denotes the elementary symmetric function of degree k.*

Proof: For $k = 1, 2, \ldots, n$, ω_k is a root of the polynomial $p(x) = x^n - 1$. Thus,

$$\begin{aligned} p(x) &= (x - \omega_1)(x - \omega_2) \cdots (x - \omega_n) \\ &= x^n - a_1 x^{n-1} + a_2 x^{n-2} - \cdots + (-1)^n a_n \,, \end{aligned}$$

where

$$\begin{aligned} a_1 &= \omega_1 + \omega_2 + \cdots + \omega_n = p_1(\omega_1, \ldots, \omega_n) \\ a_2 &= \omega_1\omega_2 + \omega_1\omega_3 + \cdots + \omega_{n-1}\omega_n = p_2(\omega_1, \ldots, \omega_n) \\ a_2 &= \omega_1\omega_2\omega_3 + \omega_1\omega_3\omega_4 + \cdots + \omega_{n-2}\omega_{n-1}\omega_n = p_3(\omega_1, \ldots, \omega_n) \\ &\vdots \\ a_n &= \omega_1\omega_2 \cdots \omega_n = p_n(\omega_1, \ldots, \omega_n) \end{aligned}$$

But $p(x) = x^n - 1$. Therefore, $a_1 = a_2 = \cdots = a_{n-1} = 0$.

Q.E.D.

These preliminary results provide the necessary means for proving the central theorem of this section.

8.4.6 **Theorem.** *Suppose A is a nilpotent matrix of order n. If $\omega_1, \omega_2, \ldots, \omega_n$ denotes the list of the distinct nth roots of unity, then*

$$(I + \omega_1 A)(I + \omega_2 A) \cdots (I + \omega_n A) = I \,.$$

In particular, for $k = 1, 2, \ldots, n-1$, the inverse of the matrix

$$T = (I + \omega_1 A)(I + \omega_2 A) \cdots (I + \omega_k A)$$

is given by

$$T^{-1} = (I + \omega_{k+1} A)(I + \omega_{k+2} A) \cdots (I + \omega_n A) \,.$$

Proof: By Theorems **8.4.2** and **8.4.5**,

$$(I + \omega_1 A)(I + \omega_2 A) \cdots (I + \omega_n A) = I + p_n(\omega_1, \omega_2, \ldots, \omega_n) A^n .$$

But $A^n = O$ since *A* is nilpotent of order *n*.

For the second part suppose that

$$T = (I + \omega_1 A)(I + \omega_2 A) \cdots (I + \omega_k A).$$

By the first part,

$$T \cdot [(I + \omega_{k+1} A)(I + \omega_{k+2} A) \cdots (I + \omega_n A)] = I\,.$$

Therefore, by the uniqueness of inverses,

$$T^{-1} = (I + \omega_{k+1} A)(I + \omega_{k+2} A) \cdots (I + \omega_n A)\,.$$

Q.E.D.

A particular consequence of this theorem is that if ω is a primitive *n*th root of unity, then

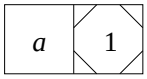
$$(I + \omega A)\big(I + \omega^2 A\big) \cdots (I + \omega^n A) = I.$$

Thus, if $T = I + A$, where $A^n = O$, then

$$T^{-1} = (I + \omega A)\big(I + \omega^2 A\big) \cdots \big(I + \omega^{n-1} A\big)\,.$$

8.4.7 **Example:** Suppose $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$, ω is a primitive *n*th root of unity, and $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ is given by

$\mathbf{t}$ =



Let *U* and *V* denote the $n \times n$ upper triangular matrices defined by

$$U = \begin{pmatrix} 0 & a & 0 & \cdots & 0 & 0 \\ 0 & 0 & a & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & a \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \; and \; V = \begin{pmatrix} 1 & a & 0 & \cdots & 0 & 0 \\ 0 & 1 & a & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} = I_n + U\,.$$

Thus, if $T = \psi(\mathbf{r})$, then

$$T = \begin{pmatrix} V & O & \cdots & O \\ O & V & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & V \end{pmatrix} = \begin{pmatrix} I_n & O & \cdots & O \\ O & I_n & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & I_n \end{pmatrix} + \begin{pmatrix} U & O & \cdots & O \\ O & U & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & U \end{pmatrix} = I_{mn} + A,$$

where $A = diag(U, U, \ldots, U)$ and O denotes the $n \times n$ zero matrix. According to Corollary **8.4.4**, A is nilpotent of order n. Hence, $T^{-1} = T_1 \cdot T_2 \cdots T_{n-1}$, where each T_i is an $mn \times mn$ block matrix, consisting of $m \times m$ blocks. Specifically,

$$T_i = \begin{pmatrix} W_i & O & \cdots & O \\ O & W_i & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & W_i \end{pmatrix},$$

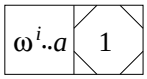
where O denotes the $n \times n$ zero matrix and W_i is the $n \times n$ upper triangular matrix of form

$$W_i = \begin{pmatrix} 1 & \omega^i a & 0 & \cdots & 0 & 0 \\ 0 & 1 & \omega^i a & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \omega^i a \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Therefore,

$$\mathbf{t}^{-1} = \mathbf{t}_1 \oplus \mathbf{t}_2 \oplus \cdots \oplus \mathbf{t}_{n-1},$$

where $\mathbf{t}_i = \psi^{-1}(T_i)$ is of form

$\mathbf{t}_i =$ 

Thus, each $\mathbf{t}_i$ is invariant and has the same geometric configuration as $\mathbf{t}$. Furthermore, the number of factors in the decomposition of $\mathbf{t}^{-1}$ is one less than the number of columns. Note also that the primitive *n*th root of unity can be chosen in the field $\mathbb{Z}_p$, where *p* is a large prime. This means that all calculations can be accomplished using only integer arithmetic, therefore avoiding round-off errors and dealing with complex numbers.

The above example has the following generalizations:

8.4.8 **Theorem.** *Suppose* $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ *and* $\mathbf{t} \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$ *is translation invariant. If* $\psi(\mathbf{t})$ *is upper triangular and* $\mathbf{t}_{\mathbf{y}}(\mathbf{y}) = 1$, *then* $\mathbf{t}^{-1}$ *can be factored into a product of at most mn invariant templates. Furthermore, the support of each template in the factorization will be identical to the support of* $\mathbf{t}$.

Proof: By hypothesis, $\psi(\mathbf{t}) = I_{mn} + A$, where A is a strictly upper triangular $mn \times mn$ matrix. Therefore, A is nilpotent of order $k \leq mn$. Hence, by Theorem **8.4.6**, the inverse of $\psi(\mathbf{t})$ can be factored into a product of at most *mn* matrices of form $I_{mn} + \omega_i A$, where ω_i is a root of unity for $i = 1, 2, \ldots, k$. Thus, $\mathbf{t}^{-1}$ can be factored as

$$\mathbf{t}^{-1} = \mathbf{t}_1 \oplus \mathbf{t}_2 \oplus \cdots \oplus \mathbf{t}_k,$$

where $\mathbf{t}_i = \psi^{-1}(I_{mn} + \omega_i A)$.

Q.E.D.

8.4.9 **Theorem.** *Suppose* $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ *and* $\mathbf{t} \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$ *is translation invariant. If* $1 = \omega_1, \omega_2, \ldots, \omega_n$ *is a list of the nth roots of unity and* $\mathbf{t}$ *has factorization*

$$\mathbf{t} = \mathbf{t}_1 \bigoplus \mathbf{t}_2 \bigoplus \cdots \bigoplus \mathbf{t}_k ,$$

such that

$\mathbf{t}_i =$

$\omega_i \cdot f$	$\omega_i \cdot e$	
$\omega_i \cdot d$	$\omega_i \cdot c$	
$\omega_i \cdot b$	$\omega_i \cdot a$	1

then

$$\mathbf{t}^{-1} = \mathbf{t}_{k+1} \bigoplus \mathbf{t}_{k+2} \bigoplus \cdots \bigoplus \mathbf{t}_n ,$$

where

$\mathbf{t}_j =$

$\omega_j \cdot f$	$\omega_j \cdot e$	
$\omega_j \cdot d$	$\omega_j \cdot c$	
$\omega_j \cdot b$	$\omega_j \cdot a$	1

for $j = k+1, k+2, \ldots, n$.

Proof: Let $A = \frac{1}{\omega_i}[\psi(\mathbf{t}_i) - I_{mn}]$. Thus A is a block matrix with strictly upper triangular $n \times n$ blocks and, by Corollary **8.4.4**, nilpotent of order n. According to Theorem **8.4.6** we now have that the inverse of

$$\begin{aligned}\psi(\mathbf{t}) &= \psi(\mathbf{t}_1) \cdot \psi(\mathbf{t}_2) \cdots \psi(\mathbf{t}_k) \\ &= (I_{mn} + \omega_1 A)(I_{mn} + \omega_2 A) \cdots (I_{mn} + \omega_k A)\end{aligned}$$

is given by

$$[\psi(\mathbf{t})]^{-1} = (I_{mn} + \omega_{k+1} A)(I_{mn} + \omega_{k+2} A) \cdots (I_{mn} + \omega_n A).$$

Therefore, $\mathbf{t}^{-1} = \mathbf{t}_{k+1} \bigoplus \mathbf{t}_{k+2} \bigoplus \cdots \bigoplus \mathbf{t}_n$ where $\mathbf{t}_j = \psi^{-1}(I_{mn} + \omega_j A)$ for $j = k+1, k+2, \ldots, n$.

Q.E.D.

Note that Theorem **8.4.9** is valid for a much wider class of templates than indicated by the hypothesis. In particular, if $\psi(\mathbf{t}_i) - I_{mn}$ is a strictly upper triangular block Toeplitz matrix with strictly upper triangular Toeplitz blocks, then the conclusion of the theorem still holds.

Identical results could also have been obtained using lower triangular matrices instead of upper triangular matrices. Furthermore, by Schur's Theorem [7] every matrix with real eigenvalues is equal to the sum of a diagonalizable matrix and a nilpotent matrix. Therefore we could have stated many of the results in this more general setting. However, the benefits of doing so do not warrant the added complications.

8.4.10 **Example:** Let $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$, $\mathbf{a} \in \mathbb{R}^{\mathbf{X}}$, and **r**, **s**, **t**, and **u** be parameterized real-valued templates defined by

$\mathbf{r}(i) =$

$i \cdot d$	$i \cdot c$	
$i \cdot b$	$i \cdot a$	1

, $\mathbf{s}(i) =$

1	$i \cdot a$	$i \cdot b$
	$i \cdot c$	$i \cdot d$

$\mathbf{t}(i) =$

$i \cdot b$	$i \cdot a$	1
$i \cdot d$	$i \cdot c$	

, and $\mathbf{u}(i) =$

	$i \cdot c$	$i \cdot d$
1	$i \cdot a$	$i \cdot b$

where *i*, *a*, *b*, *c*, and *d* are integers.

Consider the transform

$$\mathbf{b} := \mathbf{a} \oplus \mathbf{r}(1) \oplus \cdots \oplus \mathbf{r}(k) \oplus \mathbf{s}(1) \oplus \cdots \oplus \mathbf{s}(k) \oplus \mathbf{t}(1) \oplus \cdots \oplus \mathbf{t}(k) \oplus \mathbf{u}(1) \oplus \cdots \oplus \mathbf{u}(k).$$

This transform is invertible whenever the hypotheses of the theorems presented in this section are satisfied. Observe that parts of the hypotheses are satisfied since $\psi(\mathbf{r}(i)) - I_{mn}$ and $\psi(\mathbf{t}(i)) - I_{mn}$ are block matrices with strictly upper triangular $n \times n$ blocks, $\mathbf{r}(i)_{\mathbf{y}}(\mathbf{y}) = 1$ and $\mathbf{t}(i)_{\mathbf{y}}(\mathbf{y}) = 1$. Similarly, $\psi(\mathbf{s}(i)) - I_{mn}$ and $\psi(\mathbf{u}(i)) - I_{mn}$ are block matrices with strictly lower triangular $n \times n$ blocks. Now suppose that **a** is a 64×64 image with pixel values between 0 and 31. As noted earlier, the theorems hold for the finite field $\mathbb{Z}_p$, where *p* is a prime. This lets us avoid complex arithmetic and round-off errors. To invert the transform exactly, we need only choose a prime number *p* which is larger than both the number of columns and the maximum pixel value (assuming $k < p$). Thus, the most reasonable prime in this case would be the integer $p = 67$. Since the integers $1, 2, \ldots, 66$ are all the roots of unity in the field $\mathbb{Z}_{67}$, it follows from the results of this section that the transform is invertible if we set $\omega_i = i$ for $i = 1, 2, \ldots, 66$ and $k < p$. Furthermore, the inverse transform is given by

$$\mathbf{a} := \mathbf{b} \oplus \mathbf{u}(k+1) \oplus \cdots \oplus \mathbf{u}(66) \oplus \mathbf{t}(k+1) \oplus \cdots \oplus \mathbf{t}(66) \oplus \mathbf{s}(k+1) \oplus \cdots \oplus \mathbf{s}(66) \oplus \mathbf{r}(k+1) \oplus \cdots \oplus$$

Figure **8.4.1** shows the result of this transform and its inversion for $k = 1$ and $k = 33$. Note that for $k = 33$ the transform completely disguises the airplane. Having used only integer

arithmetic $mod67$ provides for an exact inversion of the transform in both cases and thus allows the recovery of the original image.

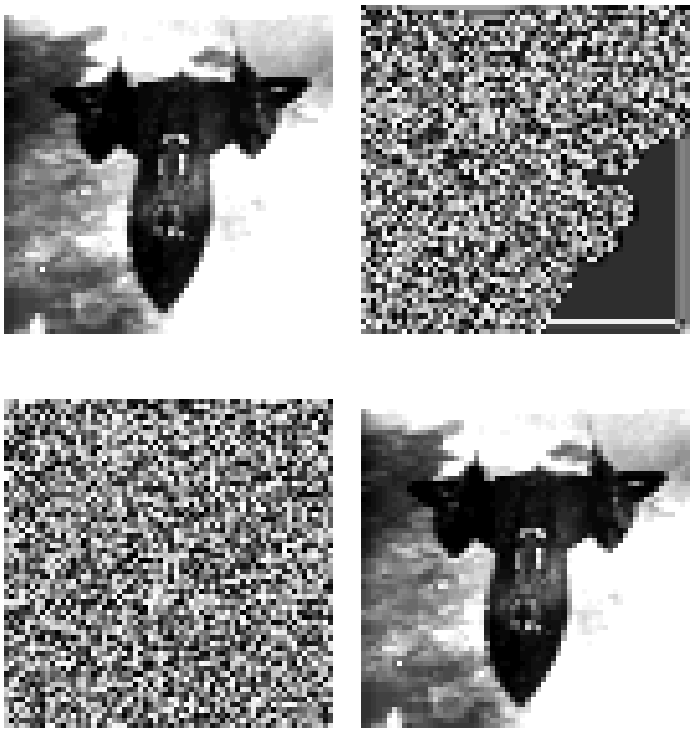


Figure 8.4.1 Masking and inversion transform. The top row shows the input image on the left and the resulting image for break value $k = 1$ on the right. The lower row shows the effect when setting the break value at $k = 33$ (left image) and the inverted image wich is identical to the source image for both breakvalues.

The foregoing example indicates how images can be encrypted and decrypted. Using large primes, the choice of combinations becomes extremely large, thus making the decoding task computationally prohibitive unless the basic template configurations, values, and *k* are known. Of course, messages as well as images can be encrypted using these techniques. One may simply equate the letter *A* with the integer 1, the letter *B* with 2, etc. If the message is entered into an array with 80 columns, then the integer 83 will be the smallest prime that will encode and recover the message exactly.

Bibliography

[1] E. Bolker. The finite radon transform. In *Proc. AMS Summer Conference on Integral Geometry*, 1984.

[2] P.J. Davis. *Circulant Matrices*. John Wiley and Sons, New York, NY, 1979.

[3] P. Diaconis. Projection pursuit for discrete data. Technical Report 148, Stanford University, Department of Statistics, 1983.

[4] P. Diaconis and R.L. Graham. The discrete radon transform on z_{2^k}. *Pacific Journal of Mathematics*, 118(2):323–345, June 1985.

[5] P.D. Gader. *Image Algebra Techniques for Parallel Computation of Discrete Fourier Transforms and General Linear Transforms*. PhD thesis, University of Florida, Gainesville, FL, 1986.

[6] I.M. Gelfand, M.I. Graev, and N. Y. Vilenkin. *Generalized Functions*. Academic Press, New York, NY, 1966.

[7] K. Hoffman and R. Kunze. *Linear Algebra*. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[8] T.W. Hungerford. *Algebra*. Springer-Verlag, New York, 1974.

[9] R. Keown. *An Introduction to Group Representation Theory*. Academic Press, New York, 1975.

[10] J. Kung. The radon transform of a combinatorial geometry. *Journal of Combinatorial Theory*, Series A:97–102, 1974.

[11] W. Ledermann. *Introduction to Group Characters*. Cambridge University Press, Cambridge, 1977.

[12] Z.Z. Manseur and D.C. Wilson. Decomposition methods for convolution operators. *Computer Vision, Graphics, and Image Processing*, 53(5):428–434, 1991.

[13] F.D. Murnaghan. *The Theory of Group Representations*. Dover Publications, Inc., New Yorck, 1963.

[14] J.M. Ortega. *Numerical Analysis: A second course*. Academic Press, New York, London, 1972.

[15] J. Radon. Ueber die bestimmung von functionen durh integralwerte langs gewisser mannigfaltigkeiten. *Ber. Verb. der Saechischen Akademie der Wissenschaften*, Math. Phys. Kl.(69):262–277, 1917.

[16] G.X. Ritter and P.D. Gader. Image algebra: Techniques for parallel image processing. *Journal of Parallel and Distributed Computing*, 4(5):7–44, April 1987.

[17] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. Academic Press, New York, NY, 2nd edition, 1982.

[18] G. E. Trapp. Inverses of circulant matrices and block circulant matrices. *Kyungpook Math. Journal*, 13(1):11–20, 1973.

CHAPTER 9
DECOMPOSITION AND INVERSION OF TEMPLATES OVER HEXAGONALLY SAMPLED IMAGES

Most digitally sampled image representations employed in digital image processing are based on placements of pixels in a rectangular grid form that corresponds to tiling the plane with squares. Reasons for interest in hexagonal grids in image processing are that digital scenes in low resolution images often look more *natural* when pixels are presented in hexagonal rather than rectangular arrangement, hexagons can be grouped into natural aggregates for fast addressing, and since each point in a hexagonal grid has six equal-distance neighbors, the 4–neighbor/8–neighbor problem does not exist.

Hexagonal grids have been examined and used by various researchers for several decades. In 1963, B.H. McCormick proposed hexagonal grids as a possible array representation for planar images [11]. in 1969, M.J.E. Golay proposed a parallel computer for hexagonal array processing and developed a basic theory for hexagonal pattern transformations [5]. Kendall Preston developed a special purpose computer architecture in 1971 which was based on hexagonal pattern transformations in order to achieve high-speed image processing routines [12]. In the early 1980's, D. Lucas and L. Gibson exploited the geometric advantages of the hexagonal representation in applications to automatic target recognition [2, 3, 4, 8, 9]. Narendra Ahuja investigated polygonal decompositions in 1983 for hierarchical image representations in triangular, square, and hexagonal lattices [1]. In 1983, D.K. Scholten and S.G. Wilson showed that the hexagonal lattice outperforms the rectangular lattice as a basis for performing chain code quantization of line drawings [13]. Informative summaries of the properties of hexagonal arrays with emphasis on image analysis in the morphological domain can be found in J. Serra's work on mathematical morphology [14, 15].

The main emphasis of this chapter is on the problem of template decomposition and inversion over hexagonal arrays. Hexagons organize themselves naturally into a hierarchy of snowflake shaped regions. These tile the plane and consequently yield a simpler definition of circulancy. Unlike the circulancy of rectangular tiling of the plane, which yields a toroidal topology, the hexagonal analogue yields the topology of a circle. As a result, circulant templates are mapped isomorphically into a quotient of the ring of polynomials in one variable. These polynomials are products of linear factors over the complex numbers. A polynomial will be invertible in the quotient ring whenever each of its linear factors is invertible. This results in a simple criterion for template invertibility.

9.1 Generalized Balanced Ternary

Balanced ternary is an addressing system proposed by D.E. Knuth for locating integers using three symbols [6]. Taking these symbols to be the integers 0, 1, and 2, the need to address negative integers requires that 2 will be used in the role of the negative integer -1. The resulting notation for the integers is shown in Figure **9.1.1**. A look at this figure suggests how to construct addition and multiplication tables for these symbols which, when applied sequentially in the manner of decimal operations, will yield standard arithmetic. In order to make things work analogous to integer arithmetic on the discrete line we must have $1 + 1 = 12$, $1 + 2 = 0$, $2 \cdot 2 = 1$, etc.

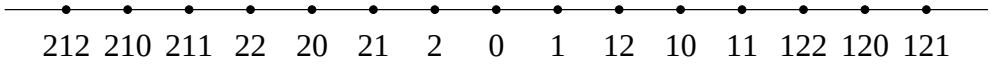


Figure 9.1.1 Balanced ternary notation for integers close to 0.

Generalized Balanced Ternary, abbreviated *GBT*, extends the algebraic and geometric properties of balanced ternary to higher dimensions [7]. The 2–dimensional generalized binary ternary numbers , abbreviated GBT_2, where developed by L. Gibson and D. Lucas as a method for addressing a hexagonal tiling of Euclidean 2–space [2, 3, 8, 9, 4]. They describe the hexagonal tiling as a hierarchy of cells, where at each level in the hierarchy new cells are constructed according to a rule of aggregation. A hexagon and its six neighbors form the first level in this hierarchy (Fig. **9.1.2**). Note that a first level aggregate can also tile 2–space and has the same uniform adjacency property that the hexagonal tiling possesses. A first level aggregate and its six first level aggregate neighbors form a second level aggregate (Fig. **9.1.3**). The hierarchy continues in the obvious way. Figure **9.1.4** shows a third level aggregate.

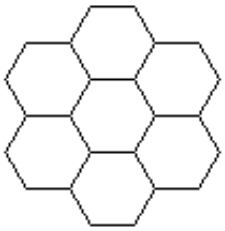


Figure 9.1.2 The first level aggregate.

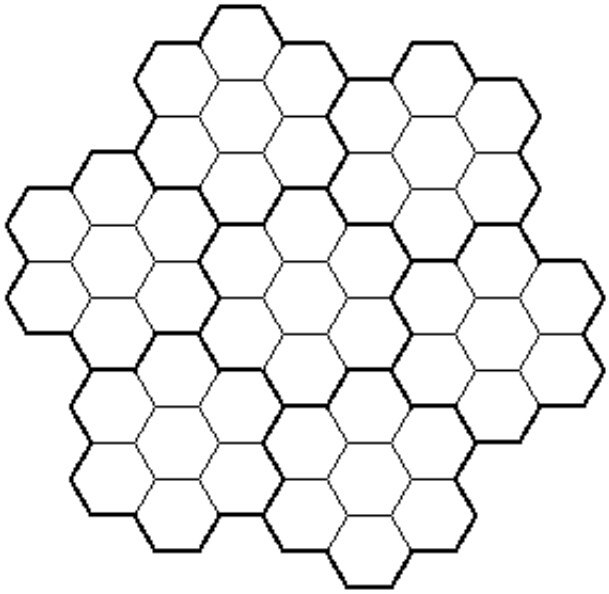


Figure 9.1.3 The second level aggregate.

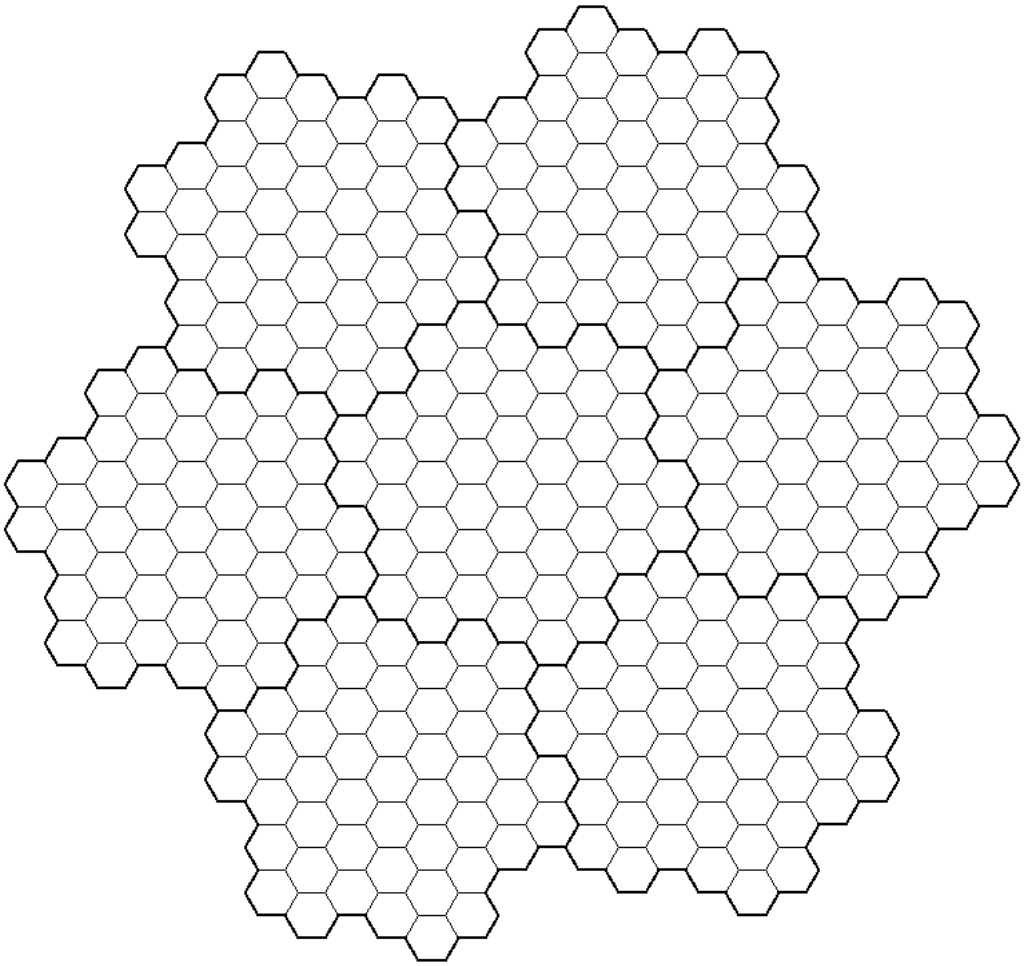


Figure 9.1.4 The third level aggregate.

The GBT_2 addressing method of the hierarchical tiling is based on the following scheme. A first level aggregate L_1 is chosen and labeled with the integers 0 through 6 as shown in Figure **9.1.5**. The center hexagon labeled 0 is considered as the origin.

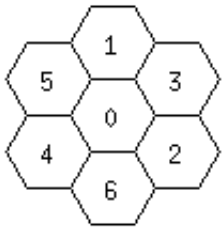


Figure 9.1.5 The GBT address of the first level aggregate.

The six first level aggregates neighboring L_1 are labeled with the digits shown in Figure **9.1.6** and form a second level aggregate L_2; each digit is some integer from $\mathbb{Z}_7$. Reading these digits from right to left, the first digit corresponds to where the labeled hexagon is in its first level aggregate L_1 and the second digit corresponds to where L_1 is in the second level aggregate L_2.

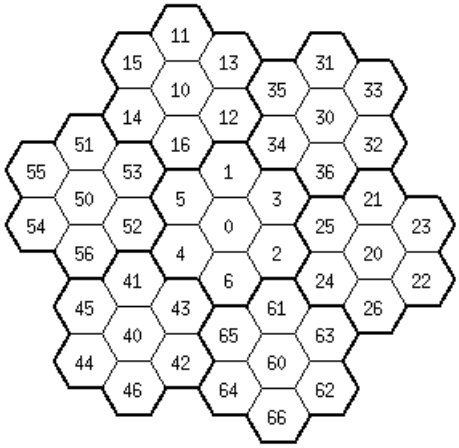


Figure 9.1.6 The GBT address of the second level aggregate.

Figure **9.1.7** illustrates the labeling scheme of the third level aggregate *centered* at L_1. Continuing in this manner, every hexagon in the tiling will correspond to a unique finite sequence (an address) with

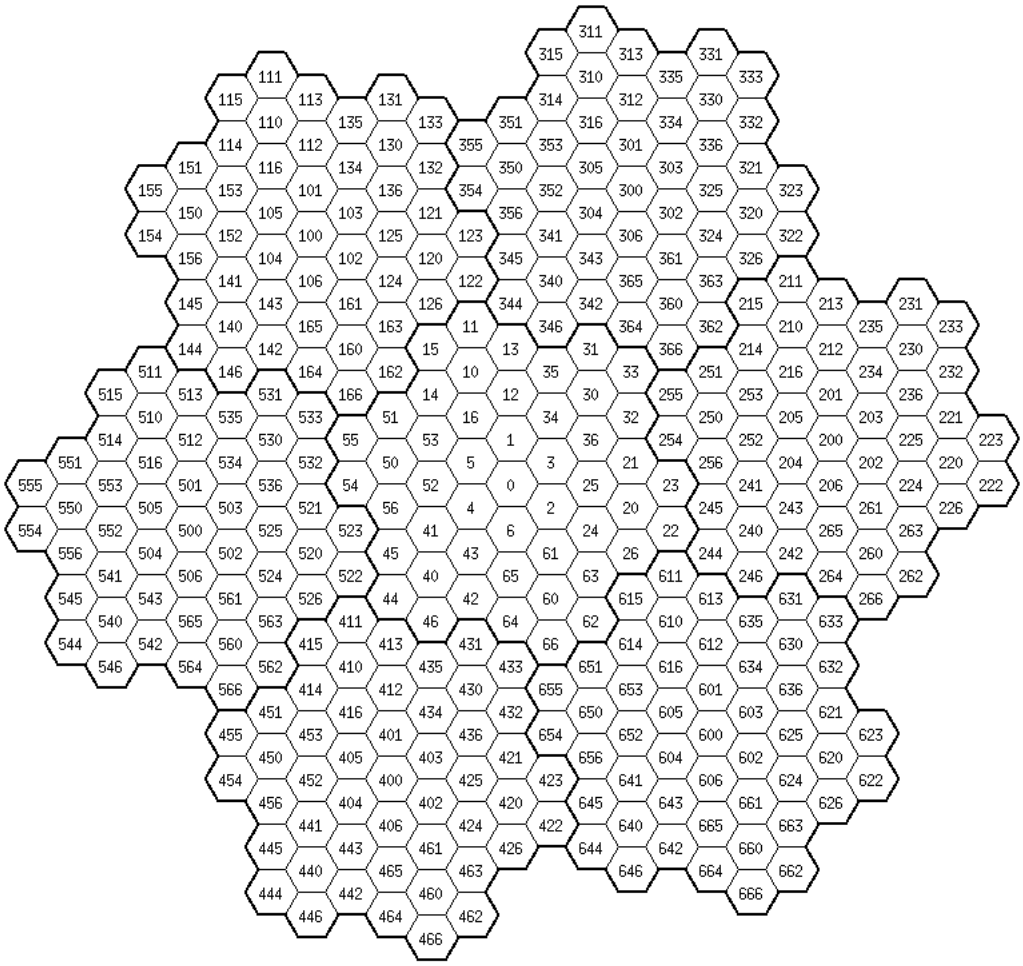


Figure 9.1.7 The GBT address of the third level aggregate.

entries an integer from $\mathbb{Z}_7$. Thus, the GBT_2 system can be thought of as a set of all finite sequences with entries from the set $\mathbb{Z}_7$.

9.2 GBT$_2$ Arithmetic

The rationale for using the GBT_2 addressing scheme described in the previous section is rooted in geometry and a desire for an efficient addressing system for hexagonal array processing. The geometric basis becomes clear if one considers Figure **9.2.1** and the parallelogram law of vector addition. If the hexagon with address 0 is centered at the origin of the plane, then the resultant of addition of the vector with terminal point at the center of the hexagon labeled 1 with the vector whose terminal point is located at the center of the hexagon labeled 2 is the vector whose terminal point is at the center of the hexagon labeled 3; i.e., $1 + 2 = 3$.

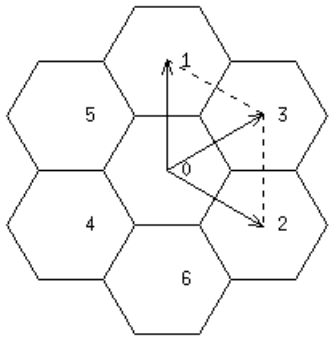


Figure 9.2.1 Addition of first level aggregate GBT_2 addresses.

A quick check shows that whenever the resultant of two vectors whose initial points and terminal points are the origin and the center of one of the hexagons in the first aggregate, respectively, is another vector in the first aggregate, then the GBT_2 address addition that yields the corresponding correct vector addition is simply addition $mod 7$. In particular, $5 + 6 = 4$, $3 + 6 = 2$, and $4 + 3 = 0$.

Proceeding to the second level aggregate, we note that if we use vector addition we should obtain $1 + 3 = 34$. By examining the first aggregate and its surrounding hexagons (Fig. **9.2.2**), one obtains a simple mechanism for defining a new addition on the set $\mathbb{Z}_7$.

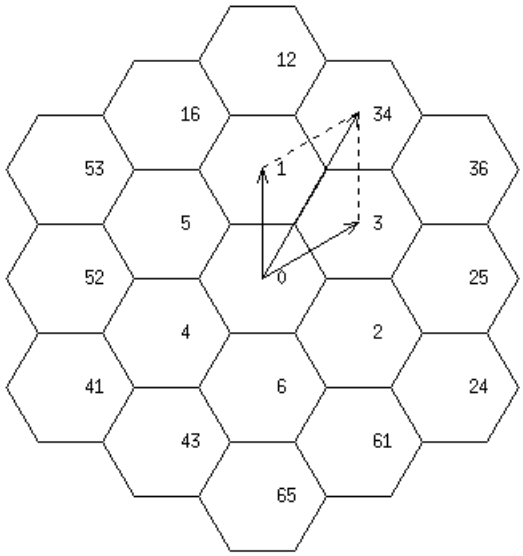


Figure 9.2.2 The first level aggregate and its surrounding hexagons.

The addition table corresponding to the vector addition in the first level aggregate is shown in Figure **9.2.3**. We may view the addition defined by this table much in the same manner as addition in the decimal system. The sum of any two digits yields a *remainder* digit and, potentially, a nonzero *carry* digit. For example $1 + 3 = 34$ has a remainder of 4 and a carry of 3. Note that the remainder corresponds to the addition $(1 + 3) mod 7$, while the carry is one of the two digits to be added.

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	12	3	34	5	16	0
2	2	3	24	25	6	0	61
3	3	34	25	36	0	1	2
4	4	5	6	0	41	52	43
5	5	16	0	1	52	53	4
6	6	0	61	2	43	4	65

Figure 9.2.3 GBT_2 addition table.

The digitwise addition decomposes the addition table into a remainder and a carry table as illustrated in Figure **9.2.4**. The digitwise arithmetic allows for the vectorially correct addition of any two GBT_2

addresses. For example, the addition of the GBT_2 addresses 416 and 346 is given by

$$\begin{array}{r} (0)(6) \\ 4\ \ 1\ 6 \\ +\underline{3\ \ 4\ 6} \\ 0\ \ 4\ 5 \end{array}$$

where (x) denotes the carry digit x. Note that nonzero carry digits in single digit addition indicate that the vector sum lies in a different first level aggregate.

+	1	2	3	4	5	6
1	2	3	4	5	6	0
2	3	4	5	6	0	1
3	4	5	6	0	1	2
4	5	6	0	1	2	3
5	6	0	1	2	3	4
6	0	1	2	3	4	5

+	1	2	3	4	5	6
1	1	0	3	0	1	0
2	0	2	2	0	0	6
3	3	2	3	0	0	0
4	0	0	0	4	5	4
5	1	0	0	5	5	0
6	0	6	0	4	0	6

Figure 9.2.4 Digitwise operation on GBT_2 with remainder table on the left and carry table on the right.

Negation of *GBT* addresses is accomplished on a digit by digit basis. The negative of 0 is 0. The negative of any other *GBT* digit *d* is the integer difference between 7 and *d*; i.e., $7 - d$. Looking at Figure **9.1.5**, one can easily infer that the negative of a given digit lies on the opposite side of 0. To form the negative of any other GBT_2 address one simply takes the negative of each of its digits. For instance, the negative of 123 is 654.

Subtraction is defined in terms of addition combined with negation in the usual way. Thus $r - s = r + (negative\ of\ s)$. In particular, $65 - 43 = 65 + 34 = 2$.

Multiplication of GBT_2 addresses is similar to addition in that it is a digitwise operation. The multiplication table (Figure **hexa12**) shows that the *GBT* product of two digits is just their integer product modulo 7. Multidigit multiplication in GBT_2 is best explained by example. To multiply 254 by 62 we proceed as follows:

$$\begin{array}{rl} 254 & \\ \times\ \underline{\ 62} & \\ 431 & (= 2 \times 254) \\ \underline{523}\ \ & (= 6 \times 254) \\ 5261 & (= GBT_2\ sum) \end{array}$$

Similarly, $255 \times 25 = 604$. Note that the multiplication of two GBT_2 addresses can be viewed as

×	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5

Figure 9.2.5 GBT_2 multiplication table. (Continued) . . .

3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Figure 9.2.5 GBT_2 multiplication table.

multiplication of vectors in 2–dimensional space; it is identical with multiplication of complex numbers. That is, it adds angles and multiplies magnitudes. In particular, it is commutative, associative, and distributes over GBT_2 addition. As a result, we a commutative ring structure which is isomorphic to a subring of the complex numbers with addition corresponding to vector addition and multiplication corresponding to complex multiplication.

9.3 Images on Hexagonal Arrays and Polynomial Rings

Let I_k denote the subset of GBT_2 addresses consisting of all sequences whose first rightmost *k* entries are equal to zero. The set I_k is closed under addition and multiplication. Therefore, I_k is an ideal of the quotient ring GBT_2. If $\mathbf{X}_k$ denotes the corresponding quotient ring GBT_2/I_k, then $\mathbf{X}_k$ consists of 7^k equivalence classes of addresses and these addresses are in one-to-one correspondence with the hexagons of the *k*th level aggregate L_k [10, 17]. The effect of the quotient ring structure on GBT_2 arithmetic operations in a level *k* aggregate is to ignore any carries beyond the *k* least significant digits. This causes the aggregate to wrap on itself in a toroidal fashion; i.e., any arithmetic operation which would normally result in a departure from the aggregate to another level will now reenter the aggregate at some other location. For example, $13 + 1 = 344$ in the full GBT_2 ring. In $\mathbf{X}_2$ the carry into the third position is ignored so that $13 + 1 = 44$.

Addition of two images $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{\mathbf{X}_7}$ is defined as the usual pointwise addition:

$$\mathbf{a} + \mathbf{b} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) \,:\, \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) + \mathbf{b}(\mathbf{x})\}\,.$$

Multiplication can also be defined pointwise, yielding a commutative ring structure on $\mathbb{R}^{\mathbf{X}_7}$. Another ring structure on $\mathbb{R}^{\mathbf{X}_7}$, which is pertinent to this chapter, is a convolution ring. Addition in this ring is again pointwise addition. However, multiplication is defined in terms of convolution as

$$\mathbf{a} * \mathbf{b} = \left\{ (\mathbf{x}, \mathbf{c}(\mathbf{x})) \,:\, \mathbf{c}(\mathbf{x}) = \sum_{\mathbf{z} \in \mathbf{X}_k} \mathbf{a}(\mathbf{z}) \cdot \mathbf{b}(\mathbf{x} - \mathbf{z}) \right\},$$

where $\mathbf{x} - \mathbf{z}$ denotes the subtraction in the quotient ring $\mathbf{X}_k = GBT_2/I_k$. This multiplication links the properties of the ring $\left(\mathbb{R}^{\mathbf{X}_7}, +, *\right)$ closely to the structure of the 7^k element ring $\mathbf{X}_k$.

Another ring having 7^k elements is $\mathbb{Z}_{7^k}$. An isomorphism $\eta : \mathbb{Z}_{7^k} \rightarrow \mathbf{X}_k$ can be defined by setting $\eta(0) = 0$, $\eta(1) = 1$, and $\eta(i) = \sum_{1}^{i} \eta(1)$. Thus, $\eta(3) = 1 + 1 + 1 = 13$, $\eta(4) = 1 + 1 + 1 + 1 = 44$, etc. The fact that η is an isomorphism can be shown directly [10] but it is also a consequence of a deeper isomorphism theorem linking the 7–adic integers with an extended version of the GBT_2 which allows infinite strings of nonzero digits [16].

The inverse of the isomorphism η, $\xi = \eta^{-1} : \mathbf{X}_k \rightarrow \mathbb{Z}_{7^k}$ provides for an elegant correspondence between images on the hexagonal array $\mathbf{X}_k$ and polynomials. With each image $\mathbf{a} \in \mathbb{R}^{\mathbf{X}_7}$, one can associate a polynomial $p_{\mathbf{a}}$ defined by

$$p_{\mathbf{a}}(x) = \sum_{\mathbf{z} \in \mathbf{X}_k} \mathbf{a}(\mathbf{z}) x^{\xi(\mathbf{z})} .$$

For two images **a** and **b** we have

$$p_{\mathbf{a}}(x) \cdot p_{\mathbf{b}}(x) = \sum_{\mathbf{z} \in \mathbf{X}_k} \mathbf{c}(\mathbf{z}) x^{\xi(\mathbf{z})} ,$$

where $\mathbf{c} = \mathbf{a} * \mathbf{b}$ by definition of polynomial multiplication. Also,

$$p_{\mathbf{a}}(x) + p_{\mathbf{b}}(x) = \sum_{\mathbf{z} \in \mathbf{X}_k} [\mathbf{a}(\mathbf{z}) + \mathbf{b}(\mathbf{z})] x^{\xi(\mathbf{z})} .$$

Therefore, the ring $\left(\mathbb{R}^{\mathbf{X}_7}, +, *\right)$ is isomorphic to the quotient ring of polynomials $\mathbb{R}[x] / \left(x^{7^k} - 1\right)$.

Bibliography

[1] N. Ahuja. On approaches to polygonal decomposition for hierarchical image representation. *Computer Vision, Graphics and Image Processing*, 24:200–214, 1983.

[2] L. Gibson and D. Lucas. Spatial data processing using generalized balanced ternary. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 566–571, June 1982.

[3] L. Gibson and D. Lucas. Vectorization of raster images using hierarchical methods. *Computer Graphics and Image Processing*, 20:82–89, 1982.

[4] L. Gibson and D. Lucas. Pyramidal algorithms for automated target recognition. In *Proceedings of the IEEE National Aerospace and Electronics Conference—NAE-CON*, pages 215–219, Dayton, Ohio, 1986.

[5] M.J.E. Golay. Hexagonal parallel pattern transformations. *IEEE Transactions on Computers*, C-18(8):733–740, August 1969.

[6] D.E. Knuth. *The Art of Computer Programming, Sorting and Searching*, volume 3. Addison-Wesley, Reading, MA, 1973.

[7] D. Lucas. A multiplication in n-space. In *Proceedings AMS*, number 74, pages 1–8, 1979.

[8] D. Lucas and L. Gibson. Image pyramid partitions. In *Seventh International Conference on Pattern Recognition*, pages 230–233, Montreal, Canada, July 1984.

[9] D. Lucas and L. Gibson. Techniques to exploit the relation between polynomial representations and moments of pictures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 183–143, San Francisco, CA, July 1985.

[10] D. Lucas and L. Gibson. Template decomposition and inversion over hexagonal sampled images. In *Image Algebra and Morphological Image Processing II*, volume 1568 of *Proceedings of SPIE*, pages 157–163, San Diego, CA, July 1991.

[11] B.H. McCormick. The Illinois pattern recognition computer — ILLIAC III. *IEEE Transactions on Electronic Computers*, 12:791–813, 1963.

[12] K. Preston. Feature extraction by golay hexagonal pattern transforms. *IEEE Transactions on Computers*, C-20(9):1007–1014, September 1971.

[13] D.K. Scholten and S.G. Wilson. Chain coding with a hexagonal lattice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(5):526–533, 1983.

[14] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[15] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, New York, 1988.

[16] W. Zhang-Kitto. *An Isomorphism Theorem between the Extended Generalized Balanced Ternary Numbers and the p-adic Integers*. Ph.D. dissertation, University of Florida, Gainesville, FL, 1991.

[17] W. Zhang-Kitto and D.C. Wilson. An isomorphism theorem between the 7–adic integers and the ring associated with a hexagonal lattice. *Journal of Applicable Algebra in Engineering, Communication, and Computation*, pages 269–285, April 1992.