Thus, at each point $\mathbf{y} \in \mathbb{Z}^2$ the templates $\mathbf{s} \boxed{\vee} \mathbf{t}$ and $\mathbf{s} \boxed{\wedge} \mathbf{t}$ are identical except off their support. The reason for this is that $\forall \mathbf{x},\mathbf{y} \in \mathbb{Z}^2$ the set $S_{-\infty}(\mathbf{s_x}) \cap S_{-\infty}(\mathbf{t_y})$ is either empty or consists of a single point.

The template $\mathbf{t}$ is not an $\mathbb{R}_{\infty}^{\geq 0}$–valued template. To provide an example of the template product $\mathbf{s} \ovee \mathbf{t}$, we redefine $\mathbf{t}$ as

$$\mathbf{t_y} = \begin{array}{|c|}\hline 1 \\ \hline 3 \\ \hline 1 \\ \hline \end{array}$$

Then $\mathbf{r} = \mathbf{s} \ovee \mathbf{t}$ is given by

$$\mathbf{r_y} = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 3 & 6 & 3 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Again, as for the products $\boxed{\vee}$ and $\boxed{\wedge}$, it is easy to see that the templates $\mathbf{r} = \mathbf{s} \ovee \mathbf{t}$ and $\mathbf{u} = \mathbf{s} \owedge \mathbf{t}$ are identical except off their respective supports, and that $S(\mathbf{r_y}) = S_{\infty}(\mathbf{u_y})$.

In order to obtain an example where $\mathbf{s} \boxed{\vee} \mathbf{t}$ and $\mathbf{s} \boxed{\wedge} \mathbf{t}$ are not identical on their supports, we need to use templates $\mathbf{s}$ and $\mathbf{t}$ such that $card[S_{-\infty}(\mathbf{s_x}) \cap S_{-\infty}(\mathbf{t_y})] > 1$ for some pair of points $\mathbf{x}$ and $\mathbf{y}$. Let $\mathbf{s}$ be as before and define $\mathbf{t}$ by

$$\mathbf{t_y} = \begin{array}{|c|c}\cline{1-1} 1 & \\ \hline 3 & \multicolumn{1}{c|}{1} \\ \hline \end{array}$$

Then $\mathbf{r} = \mathbf{s} \boxed{\vee} \mathbf{t}$ is given by

$$\mathbf{r_y} = \begin{array}{|c|c|c|}\hline 2 & 3 & 2 \\ \hline 4 & 5 & 4 \\ \hline \end{array}$$

while $\mathbf{u} = \mathbf{s} \boxed{\wedge} \mathbf{t}$ is given by

$$\mathbf{u_y} = \begin{array}{|c|c|c|}\hline 2 & 3 & 2 \\ \hline 4 & 2 & 3 \\ \hline \end{array}$$

According to Theorem **4.9.4**, if $(\mathbb{F}, \gamma, \bigcirc)$ is a ring (or semiring), then $\left(\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}, \gamma, \bigcirc\right)$ is also a ring (or semiring). The operations γ and $\bigcirc$ on $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}$ are pointwise induced operations. Moreover, if $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, then $\left(\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}, \gamma, \bigcirc\right)$ is isomorphic to the matrix ring $(\mathrm{M}_{m\times n}(\mathbb{F}), \gamma, \bigcirc)$, where $\bigcirc$ denotes Hadamard multiplication in the ring $\mathrm{M}_{m\times n}(\mathbb{F})$. Now if $m = n$, then $(\mathrm{M}_{m\times m}(\mathbb{F}), \gamma, \circledgamma)$ is also a ring under generalized matrix multiplication (Eq. 3.14.15). This follows from Example **3.5.6**. Thus, it is natural to ask about the relationship between $\left(\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}, \gamma, \circledgamma\right)$ and $(\mathrm{M}_{m\times m}(\mathbb{F}), \gamma, \circledgamma)$. Since $\left(\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}, \gamma\right)$ and $(\mathrm{M}_{m\times m}(\mathbb{F}), \gamma)$ are isomorphic under the map ψ defined by Eq. 4.9.1, we only need to investigate the relationship between the generalized matrix product and the template product.

Suppose $\mathbf{r} = \mathbf{s} \circledgamma \mathbf{t}$, where $\mathbf{s}, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$. Let $(s_{ij})_{m\times m} = \mathrm{S} = \psi(\mathbf{s})$, $(t_{ij})_{m\times m} = \mathrm{T} = \psi(\mathbf{t})$, and $(r_{ij})_{m\times m} = \mathrm{R} = \mathrm{S} \circledgamma \mathrm{T}$. Then for $1 \le i, j \le m$ we have

$$\mathbf{r}_{\mathbf{x}_j}(\mathbf{x}_i) = \Gamma_{\mathbf{x}\in\mathbf{X}} (\mathbf{s}_{\mathbf{x}}(\mathbf{x}_i) \bigcirc \mathbf{t}_{\mathbf{x}_j}(\mathbf{x})) = \Gamma_{k=1}^{m} (\mathbf{s}_{\mathbf{x}_k}(\mathbf{x}_i) \bigcirc \mathbf{t}_{\mathbf{x}_j}(\mathbf{x}_k)) = \Gamma_{k=1}^{m} s_{ik} \bigcirc t_{kj} = r_{ij}\,.$$

Therefore,

$$\psi(\mathbf{s} \circledgamma \mathbf{t}) = \psi(\mathbf{r}) = \mathrm{R} = \mathrm{S} \circledgamma \mathrm{T} = \psi(\mathbf{s}) \circledgamma \psi(\mathbf{t})\,.$$

This shows that ψ preserves the multiplication operation $\circledgamma$.

It now follows that if $\mathbf{r}, \mathbf{s}, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$, then

$$\begin{aligned}
\mathbf{r} \circledgamma (\mathbf{s} \circledgamma \mathbf{t}) &= \psi^{-1}[\psi(\mathbf{r} \circledgamma (\mathbf{s} \circledgamma \mathbf{t}))] \\
&= \psi^{-1}[\psi(\mathbf{r}) \circledgamma \psi(\mathbf{s} \circledgamma \mathbf{t})] \\
&= \psi^{-1}[\psi(\mathbf{r}) \circledgamma (\psi(\mathbf{s}) \circledgamma \psi(\mathbf{t}))] \\
&= \psi^{-1}[(\psi(\mathbf{r}) \circledgamma \psi(\mathbf{s})) \circledgamma \psi(\mathbf{t})] \\
&= \psi^{-1}[\psi(\mathbf{r} \circledgamma \mathbf{s}) \circledgamma \psi(\mathbf{t})] \\
&= \psi^{-1}[\psi((\mathbf{r} \circledgamma \mathbf{s}) \circledgamma \mathbf{t})] \\
&= (\mathbf{r} \circledgamma \mathbf{s}) \circledgamma \mathbf{t}\,.
\end{aligned}$$

Therefore, $\circledgamma$ is associative in $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

Also,

$$\begin{aligned}
\mathbf{r} \circledgamma (\mathbf{s}\gamma\mathbf{t}) &= \psi^{-1}[\psi(\mathbf{r} \circledgamma (\mathbf{s}\gamma\mathbf{t}))] \\
&= \psi^{-1}[\psi(\mathbf{r}) \circledgamma \psi(\mathbf{s}\gamma\mathbf{t})] \\
&= \psi^{-1}[\psi(\mathbf{r}) \circledgamma (\psi(\mathbf{s})\gamma\psi(\mathbf{t}))] \\
&= \psi^{-1}[(\psi(\mathbf{r}) \circledgamma \psi(\mathbf{s}))\gamma(\psi(\mathbf{r}) \circledgamma \psi(\mathbf{t}))] \\
&= \psi^{-1}[\psi(\mathbf{r} \circledgamma \mathbf{s})\gamma\psi(\mathbf{r} \circledgamma \mathbf{t})] \\
&= \psi^{-1}[\psi((\mathbf{r} \circledgamma \mathbf{s})\gamma(\mathbf{r} \circledgamma \mathbf{t}))] \\
&= (\mathbf{r} \circledgamma \mathbf{s})\gamma(\mathbf{r} \circledgamma \mathbf{t})\,.
\end{aligned}$$

Similarly, $(\mathbf{s}\gamma\mathbf{t}) \circledgamma \mathbf{r} = (\mathbf{s} \circledgamma \mathbf{r})\gamma(\mathbf{t} \circledgamma \mathbf{r})$. Hence, $\circledgamma$ distributes over γ in $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

This establishes the following theorem.

4.10.3 Theorem. *If* $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, *then* $\psi : \left(\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}, \gamma, \circledgamma\right) \to (\mathrm{M}_{m\times m}(\mathbb{F}), \gamma, \circledgamma)$ *is a ring isomorphism.*

Since in general matrix multiplication is not commutative, we have, in contrast to Theorem **4.9.4**, that $\left(\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}, \gamma, \circledgamma\right)$ is not a commutative ring even if $(\mathbb{F}, \gamma, \bigcirc)$ is a commutative ring. However, if $\mathbb{F}$ has an identity with respect to $\bigcirc$, then $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ also has an identity with respect to $\circledgamma$. If 1 denotes the identity of $\mathbb{F}$ with respect to $\bigcirc$, then the template $\mathbb{1} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ defined by

$$\mathbb{1}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{y} \\ 0 & \text{otherwise}. \end{cases} \tag{4.10.9}$$

is the identity of $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ with respect to the operation $\circledgamma$. This template is also called the *unit template* of $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

The image-template product is closely related to the template–template product. In order to investigate its relationship to matrix multiplication, let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, and let $\nu : \mathbb{F}^{\mathbf{X}} \to \mathbb{F}^m$ and $\mu : \mathbb{F}^{\mathbf{Y}} \to \mathbb{F}^n$ denote the vector maps defined by $\nu(\mathbf{a}) = (\mathbf{a}(\mathbf{x}_1), \ldots, \mathbf{a}(\mathbf{x}_m))$ and $\mu(\mathbf{b}) = (\mathbf{b}(\mathbf{y}_1), \ldots, \mathbf{b}(\mathbf{y}_n))$, respectively. For $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}$ set $(t_{ij})_{m\times n} = \mathrm{T} = \psi(\mathbf{t})$, $\mathbf{b} = \mathbf{a} \circledgamma \mathbf{t}$, and $b = (b_1, \ldots, b_n) = \nu(\mathbf{a}) \circledgamma \psi(\mathbf{t})$. Then

$$b_j = \mathop{\Gamma}_{k=1}^{m} \mathbf{a}(\mathbf{x}_k) \bigcirc t_{kj} = \mathop{\Gamma}_{k=1}^{m} \mathbf{a}(\mathbf{x}_k) \bigcirc \mathbf{t}_{\mathbf{y}_j}(\mathbf{x}_k) = \mathop{\Gamma}_{\mathbf{x}\in\mathbf{X}} \mathbf{a}(\mathbf{x}) \bigcirc \mathbf{t}_{\mathbf{y}_j}(\mathbf{x}) = \mathbf{b}(\mathbf{y}_j).$$

Thus,

$$\mu(\mathbf{a} \circledgamma \mathbf{t}) = \mu(\mathbf{b}) = \nu(\mathbf{a}) \circledgamma \psi(\mathbf{t}) \tag{4.10.10}$$

or, since μ is one-to-one,

$$\mathbf{a} \circledgamma \mathbf{t} = \mu^{-1}(\nu(\mathbf{a}) \circledgamma \psi(\mathbf{t})).$$

It now follows that

$$(\mathbf{a}\gamma\mathbf{b}) \circledgamma \mathbf{t} = (\mathbf{a} \circledgamma \mathbf{t})\gamma(\mathbf{b} \circledgamma \mathbf{t}). \tag{4.10.11}$$

If $\mathbf{X} = \mathbf{Y}$, then $\nu = \mu$ and Eq. 4.10.10 has form

$$\nu(\mathbf{a} \circledgamma \mathbf{t}) = \nu(\mathbf{a}) \circledgamma \psi(\mathbf{t}). \tag{4.10.12}$$

Hence, whenever $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and $\mathbf{s}, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$, we have that

$$\begin{aligned} \nu(\mathbf{a} \circledgamma (\mathbf{s} \circledgamma \mathbf{t})) &= \nu(\mathbf{a}) \circledgamma \psi(\mathbf{s} \circledgamma \mathbf{t}) \\ &= \nu(\mathbf{a}) \circledgamma (\psi(\mathbf{s}) \circledgamma \psi(\mathbf{t})) \\ &= (\nu(\mathbf{a}) \circledgamma \psi(\mathbf{s})) \circledgamma \psi(\mathbf{t}) \\ &= \nu(\mathbf{a} \circledgamma \mathbf{s}) \circledgamma \psi(\mathbf{t}) \\ &= \nu((\mathbf{a} \circledgamma \mathbf{s}) \circledgamma \mathbf{t}). \end{aligned}$$

Therefore,

$$\mathbf{a} \circledgamma (\mathbf{s} \circledgamma \mathbf{t}) = (\mathbf{a} \circledgamma \mathbf{s}) \circledgamma \mathbf{t}. \tag{4.10.13}$$

The next example demonstrates the utility of Eq. 4.10.13.

4.10.4 **Example:** Suppose $\mathbf{X} = \{(x_1, x_2) \in \mathbb{Z}^2 : 1 \leq x_1 \leq m,\ 1 \leq x_2 \leq n\}$, $\mathbf{a} \in \mathbb{R}^{\mathbf{X}}$, and $\mathbf{r} \in \left(\mathbb{R}^{\mathbb{Z}^2}\right)^{\mathbb{Z}^2}$ is defined by

$$\mathbf{r_y} = \begin{array}{|c|c|c|} \hline 4 & 6 & -4 \\ \hline 6 & 9 & -6 \\ \hline -4 & -6 & 4 \\ \hline \end{array}$$

$\forall \mathbf{y} \in \mathbb{Z}^2$. Then according to Eq. 4.10.13,

$$\mathbf{a} \oplus \mathbf{r} = \mathbf{a} \oplus (\mathbf{s} \oplus \mathbf{t}) = (\mathbf{a} \oplus \mathbf{s}) \oplus \mathbf{t}\,,$$

where

$$\mathbf{s_y} = \begin{array}{|c|c|c|} \hline 2 & 3 & -2 \\ \hline \end{array} \qquad \mathbf{t_y} = \begin{array}{|c|} \hline 2 \\ \hline 3 \\ \hline -2 \\ \hline \end{array}$$

The construction of the new image $\mathbf{b} := \mathbf{a} \oplus \mathbf{r}$ requires nine multiplications and eight additions per pixel (if we ignore boundary pixels). In contrast, the computation of the image $\mathbf{b} := (\mathbf{a} \oplus \mathbf{s}) \oplus \mathbf{t}$ requires only six multiplications and four additions per pixel. For large images (e.g., size 1024×1024) this amounts to significant savings in computation.

Boundary problems will occur when convolving images defined on finite arrays with templates defined over the whole discrete plane $\mathbb{Z}^2$. Equation 4.10.13 holds for images $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and templates $\mathbf{s}, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$. In particular, the equality $\mathbf{a} \oplus (\mathbf{s} \oplus \mathbf{t}) = (\mathbf{a} \oplus \mathbf{s}) \oplus \mathbf{t}$ in the last example need not hold for arbitrary $\mathbf{s}, \mathbf{t} \in \left(\mathbb{R}^{\mathbb{Z}^2}\right)^{\mathbb{Z}^2}$. As an example, suppose $\mathbf{s}, \mathbf{t} \in \left(\mathbb{R}^{\mathbb{Z}^2}\right)^{\mathbb{Z}^2}$ are two translation invariant templates defined by

$$\mathbf{s_y} = \begin{array}{ccc} & & \boxed{3} \\ & \boxed{2} & \\ \boxed{1} & & \end{array} \qquad \mathbf{t_y} = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Then $\mathbf{r} = \mathbf{s} \oplus \mathbf{t}$ is given by

$\mathbf{r_y}$ =

		3	3	3
	2	5	8	3
1	3	8	5	3
1	4	3	2	
1	1	1		

Let $\mathbf{a} \in \mathbb{R}^{\mathbf{X}}$ with $\mathbf{a}(1, n) = 3$ and values near (1,*n*) as shown in Figure **4.10.2**.

	2	1	2	3
	2	4	2	0
	2	3	1	1
	1	1	0	2

Figure 4.10.2 Part of the image **a** near the corner point (1,*n*) with the shaded cell representing the point (1,*n*).

Now if $\mathbf{b} := \mathbf{a} \oplus \mathbf{r}$, then $\mathbf{b}(1, n) = \sum_{(x_1,x_2)\in \mathbf{X}\cap S(\mathbf{r}_{(m,n)})} \mathbf{a}(x_1, x_2) \cdot \mathbf{r}_{(1,n)}(x_1, x_2) = 48$ since

$\mathbf{r}_{(1,n)}\,|_{\mathbf{X}}$ =

1	3	8
1	4	3
1	1	1

On the other hand, if $\mathbf{c} := (\mathbf{a} \oplus \mathbf{s}) \oplus \mathbf{t}$, then $\mathbf{c}(1, n) = 32$ since

$\mathbf{s}_{(1,n)}\,|_{\mathbf{X}}$ =

	2
1	

and

$$\mathbf{t}_{(1,n)}\big|_{\mathbf{X}} = \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 1 & 1 \\ \hline \end{array}$$

Therefore, $\mathbf{a} \oplus (\mathbf{s} \oplus \mathbf{t}) \neq (\mathbf{a} \oplus \mathbf{s}) \oplus \mathbf{t}$.

However, if $\mathbf{s}', \mathbf{t}' \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ are defined by $\mathbf{s}' = \mathbf{s}|_{(\mathbf{X},\mathbb{R}^{\mathbf{X}})}$ and $\mathbf{t}' = \mathbf{t}|_{(\mathbf{X},\mathbb{R}^{\mathbf{X}})}$ (i.e., $\mathbf{s}'_{\mathbf{y}} = \mathbf{s}_{\mathbf{y}}|_{\mathbf{X}}$ and $\mathbf{t}'_{\mathbf{y}} = \mathbf{t}_{\mathbf{y}}|_{\mathbf{X}}$ $\forall \mathbf{y} \in \mathbf{X}$), then $\mathbf{a} \oplus (\mathbf{s}' \oplus \mathbf{t}') = (\mathbf{a} \oplus \mathbf{s}') \oplus \mathbf{t}'$. Note that if $\mathbf{r}' = \mathbf{s}' \oplus \mathbf{t}'$, then

$$\mathbf{r}'_{(1,n)} = \begin{array}{|c|c|c|}\cline{2-3} \multicolumn{1}{c|}{} & 2 & 4 \\ \hline 1 & 4 & 2 \\ \hline 1 & 1 & \multicolumn{1}{c}{} \\ \cline{1-2} \end{array}$$

Thus, $\mathbf{r}' \neq \mathbf{r}|_{(\mathbf{X},\mathbb{R}^{\mathbf{X}})}$ (e.g., equality fails for points in $\mathbf{X}$ that are on the boundary of $\mathbf{X}$).

4.11 Spatial Transformations of Templates

Spatial transformations are applied to a template in the same way as they are applied to any other image; if $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}$ and $f : \mathbf{Z} \to \mathbf{Y}$, then

$$\mathbf{r} = \mathbf{t} \circ f \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Z}} \tag{4.11.1}$$

is defined by

$$\mathbf{r}_{\mathbf{z}} = \mathbf{t}_{f(\mathbf{z})},$$

where $\mathbf{z} \in \mathbf{Z}$. The corresponding diagram is given in Figure **4.11.1**.

$$\begin{tikzcd} \mathbf{Z} \arrow[r, "f"] \arrow[dr, "\mathbf{r}"'] & \mathbf{Y} \arrow[d, "\mathbf{t}"] \\ & \mathbb{F}^{\mathbf{X}} \end{tikzcd}$$

Figure 4.11.1 The spatial transformation $\mathbf{r} = \mathbf{t} \circ f$ of the template $\mathbf{t}$.

It follows that every spatial transform $f : \mathbf{Z} \to \mathbf{Y}$ induces a unary operation $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}} \to \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Z}}$ defined by $\mathbf{t} \rightarrowtail \mathbf{t} \circ f$ $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}} \to \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Z}}$. If f is a transformation from **Z** to **X** instead from **Z** to **Y**, then f induces a unary operation $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}} \to \left(\mathbb{F}^{\mathbf{Z}}\right)^{\mathbf{Y}}$ defined by $\mathbf{t} \rightarrowtail \mathbf{r}$, where

$$\mathbf{r}_{\mathbf{y}} = \mathbf{t}_{\mathbf{y}} \circ f \quad \forall \mathbf{y} \in \mathbf{Y} . \tag{4.11.2}$$

Thus, $\mathbf{r}_{\mathbf{y}}(\mathbf{z}) = \mathbf{t}_{\mathbf{y}}(f(\mathbf{z})) \quad \forall \mathbf{z} \in \mathbf{Z}$.

There are obvious differences between the transformations defined by Eqs. 4.11.1 and 4.11.2. Computationally, 4.11.2 is far more complex since functional composition has to be performed for each point **y**. However, as the example below illustrates, for translation invariant templates we only need to compose once.

Spatial transformations can be a useful tool for redefining templates. For instance, the templates $\mathbf{t}^{\backslash}$ and $\mathbf{s}^{\backslash}$ in Example **4.8.4** were obtained using spatial transforms. The next example illustrates a similar application.

4.11.1 Example: (*Edge mask determination.*) Edge information conveys image information and, consequently, scene content. Edge detection contributes significantly to algorithms for feature detection, segmentation, and motion analysis. In a continuous image, a sharp intensity transition between neighboring pixels as shown in Figure **4.11.2**(a) would be considered an edge. Such steep changes in intensities can be detected by analyzing the derivatives of the signal function. In sampled waveforms such as shown in Figure **4.11.2**(b), approximations to the derivative, such as finite difference methods, are used to detect the existence of edges. However, due to sampling, high frequency components are introduced, and every pair of pixels with different intensities could be considered an edge. For this reason, smoothing before edge enhancement followed by thresholding after enhancement are an important part of many edge detection schemes.

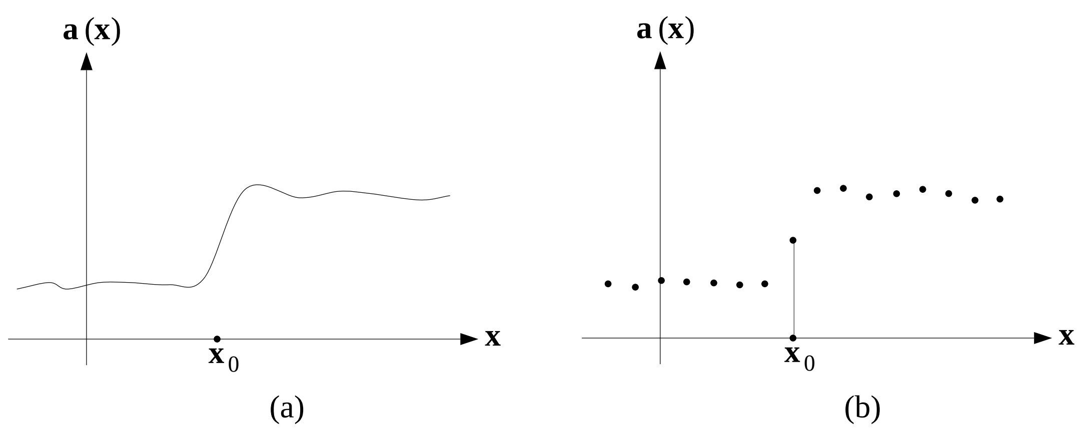


Figure 4.11.2 (a) Continuous image with edge phenomenon. (b) Sampled image function.

The gradient of an image **a** is defined in terms of direction oriented spatial derivatives as

$$\nabla \mathbf{a}(\mathbf{x}) = \frac{\partial \mathbf{a}(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial \mathbf{a}(x_1, x_2)}{\partial x_1} \\ \frac{\partial \mathbf{a}(x_1, x_2)}{\partial x_2} \end{pmatrix} .$$

One discrete approximation of the gradient is given in terms of the centered differences

$$d(x_1) = \frac{\mathbf{a}(x_1 + \triangle x_1, x_2) - \mathbf{a}(x_1 - \triangle x_1, x_2)}{2(\triangle x_1)}$$

and

$$d(x_2) = \frac{\mathbf{a}(x_1, x_2 + \triangle x_2) - \mathbf{a}(x_1, x_2 - \triangle x_2)}{2(\triangle x_2)}.$$

The centered differences $d(x_1)$ and $d(x_2)$ can be implemented using the templates

$\mathbf{t_y} =$

-1		1

$\mathbf{s_y} =$

1
-1

The pixel values at location $\mathbf{x} = (x_1, x_2)$ of $\mathbf{a} \oplus \mathbf{t}$ and $\mathbf{a} \oplus \mathbf{s}$ are the centered differences $d(x_1)$ and $d(x_2)$, respectively (see also Example **3.6.6**). This concept forms the basis for extensions to various templates used for edge detection. Variants of the centered difference templates are the 3×3 templates

$\mathbf{t_y} =$

-1		1
-1		1
-1		1

$\mathbf{s_y} =$

1	1	1
-1	-1	-1

which form *smoothened* or *averaged* central difference operators.

The templates **t** and **s** are orthogonal; one is sensitive to vertical edges and the other to horizontal edges. We may view **t** and **s** as corresponding to evaluating the averaged derivatives in the 0° and 90° direction, respectively. Evaluation of the derivative in directions other than 0° or 90° becomes necessary if one is interested in detecting edges that are neither horizontal or vertical. One simple way of obtaining a template that can be used for evaluating the averaged central difference in a desired direction θ is to simply rotate the image $\mathbf{t}_{(0,0)}$ through the angle θ and use the resulting image values for the weights of the new template. For example, applying the rotation f defined by Eq. 4.5.4 to the image $\mathbf{t}_{(0,0)}$ with θ= 90° results in the image $\mathbf{t}_{(0,0)} \circ f$ which is identical to the image $\mathbf{s}_{(0,0)}$. Since **s** is translation invariant, the image $\mathbf{t}_{(0,0)} \circ f$ completely determines **s**. However, for angles θ other than 90° the resulting image $\mathbf{t}_{(0,0)} \circ f$ may not be a satisfactory approximation of the averaged central difference in that direction; for θ= 30° the image $\mathbf{t}_{(0,0)} \circ f$ is basically the same as $\mathbf{t}_{(0,0)}$. The reason for this is the small size of the support of $\mathbf{t}_{(0,0)}$ and associated interpolation errors. In order to obtain a more accurate representation, one scheme is to enlarge the image $\mathbf{t}_{(0,0)}$ to the image

-1	-1	0	1	1
-1	-1	0	1	1
-1	-1	0	1	1
-1	-1	0	1	1
-1	-1	0	1	1

and rotate this image using linear interpolation. For θ= 30° the resulting image $\mathbf{t}_{(0,0)} \circ f$ is of form

-0.73	0.13	1	1	1
-1	-0.37	0.5	1	1
-1	-0.87	0	0.87	1
-1	-1	-0.5	0.37	1
-1	-1	-1	-0.13	0.73

The final 3×3 template **r** is obtained by restricting its support to a 3×3 neighborhood of the center pixel location **y**, namely

$\mathbf{r_y}$ =

-0.37	0.5	1
-0.87	0	0.87
-1	-0.5	0.37

While **t** will result in maximal enhancement of vertical or 90° edges, **r** will be more sensitive to 120° edges. Likewise, rotation through an angle of 60° results in the template

$\mathbf{s_y}$ =

0.37	0.87	1
-0.5	0	0.5
-1	-0.87	-0.37

which is sensitive to 150° edges.

4.12 Multivalued Images

Although each value **a**(**x**) of an image $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ is a unique element of $\mathbb{F}$, **a**(**x**) may be composed of values from several value sets. For example, if $\mathbb{F} = \mathbb{R}^n$, then **a**(**x**) is a vector of form $\mathbf{a}(\mathbf{x}) = (\mathbf{a}_1(\mathbf{x}), \ldots, \mathbf{a}_n(\mathbf{x}))$ where for each $i = 1, \ldots, n$, $\mathbf{a}_i(\mathbf{x}) \in \mathbb{R}$. Thus, with each *vector-value* **a**(**x**) there are associated *n* real values $\mathbf{a}_i(\mathbf{x})$. A real vector valued image represents a special case of a multivalued image.

4.12.1 **Definition.** If $\mathbb{F}$ is a value set and $\mathbb{F}$ can be written as a Cartesian product of value sets $\mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$ with $n > 1$, then $\mathbb{F}$ is called a *multivalued* set. If $\mathbb{F}$ cannot be written as such a product, then $\mathbb{F}$ is called a *single-valued* set.

If $\mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$ is a multivalued set and $\mathbb{F}_1 = \mathbb{F}_2 = \cdots = \mathbb{F}_n$, then $\mathbb{F}$ is called a *homogeneous* multivalued set or simply a *vector-valued* set. A multivalued set which is not a homogeneous multivalued set is referred to as a *non-homogeneous* or *heterogeneous* or *mixed* multivalued set.

If $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and $\mathbb{F}$ is a multivalued set, then **a** is called a *multivalued* image. If $\mathbb{F}$ is a homogeneous multivalued set, then **a** is also called a *vector-valued* image.

If $\mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$ and $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$, then **a** is of form $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_n)$ where for each $i = 1, \ldots, n$, $\mathbf{a}_i \in \mathbb{F}_i^{\mathbf{X}}$ is given by $\mathbf{a}_i(\mathbf{x}) = p_i(\mathbf{a}(\mathbf{x}))$. Thus, every multivalued image can be viewed as a vector whose components are single-valued images.

Conversely, if for $i = 1, \ldots, n$ there exists an image $\mathbf{a}_i : \mathbf{X} \to \mathbb{F}_i$, then the collection $\{\mathbf{a}_i : i = 1, \ldots, n\}$ determines a multivalued image $\mathbf{a} : \mathbf{X} \to \prod_{i=1}^{n} \mathbb{F}_i$ which is defined by $\mathbf{a}(\mathbf{x}) = (\mathbf{a}_1(\mathbf{x}), \ldots, \mathbf{a}_n(\mathbf{x}))$ (see also **2.5.7**). This means that we can construct a multivalued image from any array of images defined over the same point set.

Another method of forming multivalued images is through functions that map a single-valued set into a multivalued set. More precisely, if $\mathbf{a} \in \mathbb{E}^{\mathbf{X}}$, where $\mathbb{E}$ is a single-valued set, and $f : \mathbb{E} \to \prod_{i=1}^{n} \mathbb{F}_i$, then f induces the unary operation

$$f(\mathbf{a}) = f \circ \mathbf{a} \in \left(\prod_{i=1}^{n} \mathbb{F}_i \right)^{\mathbf{X}} .$$

For example, if $\mathbf{a} \in \mathbb{R}^{\mathbf{X}}$ and $f : \mathbb{R} \to \mathbb{R} \times \mathbb{Z}_2$ is defined by $f(r) = \left(r, \chi_{\geq k}(r)\right)$, then the image $\mathbf{b} = f(\mathbf{a})$ has values $\mathbf{b}(\mathbf{x}) = \left(\mathbf{a}(\mathbf{x}), \chi_{\geq k}(\mathbf{a}(\mathbf{x}))\right)$. This example is, of course, a special case of the general unary operation defined by Eq. 4.7.3. Obviously, any unary operation on single or multivalued images on a given spatial domain **X** is a function of the type described by Eq. 4.7.3.

Suppose $f : \mathbb{F} \to \mathbb{G}$, where $\mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$ and $\mathbb{G} = \prod_{j=1}^{m} \mathbb{G}_j$. Then f generates a sequence of functions $f_j : \mathbb{F} \to \mathbb{G}_j$ defined by $f_j = p_j \circ f$. Thus, every function $f : \mathbb{F} \to \mathbb{G}$ induces a unary operation $f : \mathbb{F}^{\mathbf{X}} \to \mathbb{G}^{\mathbf{X}}$ and a family of unary operations $f_j : \mathbb{F}^{\mathbf{X}} \to \mathbb{G}_j^{\mathbf{X}}$ (as defined by Eq. 4.7.3).

Conversely, given a sequence of functions $f_j : \mathbb{F} \to \mathbb{G}_j$, then we obtain a sequence of induced unary operations $f_j : \mathbb{F}^{\mathbf{X}} \to \mathbb{G}_j^{\mathbf{X}}$ and a unary operation $f : \mathbb{F}^{\mathbf{X}} \to \mathbb{G}^{\mathbf{X}}$ which is defined by

$$f(\mathbf{a}) = (f_1(\mathbf{a}), \ldots, f_m(\mathbf{a})). \tag{4.12.1}$$

Equivalently,

$$f(\mathbf{a}) = \{(\mathbf{x}, \mathbf{b}(\mathbf{x})) : \mathbf{b}(\mathbf{x}) = (f_1(\mathbf{a}(\mathbf{x})), \ldots, f_m(\mathbf{a}(\mathbf{x})))\}.$$

Induced unary operations can be constructed from an even lower level of specification. Suppose that for some indexing function $\lambda : \{1, \ldots, m\} \to \{1, \ldots, n\}$ there exists a sequence of functions

$$\bar{f}_j : \mathbb{F}_{\lambda(j)} \to \mathbb{G}_j\,, \quad j = 1, \ldots, m.$$

Then the sequence of induced unary operations $\bar{f}_j : \mathbb{F}_{\lambda(j)}^{\mathbf{X}} \to \mathbb{G}_j^{\mathbf{X}}$ induces a sequence of unary operations

$$f_j : \mathbb{F}^{\mathbf{X}} \to \mathbb{G}_j^{\mathbf{X}}$$

which is defined by $f_j = \bar{f}_j \circ p_{\lambda(j)}$, where $p_{\lambda(j)}$ denotes the projection $p_{\lambda(j)} : \mathbb{F}^{\mathbf{X}} \to \mathbb{F}_{\lambda(j)}^{\mathbf{X}}$. Obviously, the sequence $\{\bar{f}_j : j = 1, \ldots, m\}$ also induces a unary operation $f : \mathbb{F}^{\mathbf{X}} \to \mathbb{G}^{\mathbf{X}}$ which is defined by

$$f(\mathbf{a}) = (f_1(\mathbf{a}), \ldots, f_m(\mathbf{a})). \tag{4.12.2}$$

Equivalently,

$$\begin{aligned} f(\mathbf{a}) &= \left(\bar{f}_1 \circ p_{\lambda(1)}(\mathbf{a}), \ldots, \bar{f}_m \circ p_{\lambda(m)}(\mathbf{a})\right) \\ &= \left(\bar{f}_1\left(\mathbf{a}_{\lambda(1)}\right), \ldots, \bar{f}_m\left(\mathbf{a}_{\lambda(m)}\right)\right). \end{aligned} \tag{4.12.2}$$

For homogeneous multivalued sets we obtain a special class of unary operations. Suppose $\mathbb{F}$ and $\mathbb{G}$ are homogeneous multivalued sets, $m = n$, λ denotes the identity function $\lambda(j) = j$, and $\bar{f}_1 = \bar{f}_2 = \cdots = \bar{f}_n$. Then the induced operation *f* is called a *homogeneous unary operation*, otherwise *f* is called a *heterogeneous* or *mixed unary operation*. If *f* is a homogeneous unary operation then we let the coordinate function $\bar{f}_i$ represent *f*.

4.12.2 Examples:

(i) If $\mathbf{a} \in (\mathbb{R}^n)^{\mathbf{X}}$ and for $i = 1, \ldots, n$ $\bar{f}_i = sin : \mathbb{R} \to \mathbb{R}$, then

$$sin(\mathbf{a}) = (sin(\mathbf{a}_1), \ldots, sin(\mathbf{a}_n)).$$

Similarly, if for $i = 1, \ldots, n$ $\bar{f}_i = \chi_{\geq k}$, then

$$\chi_{\geq k}(\mathbf{a}) = \left(\chi_{\geq k}(\mathbf{a}_1), \ldots, \chi_{\geq k}(\mathbf{a}_n)\right).$$

Note that $\chi_{\geq k}(\mathbf{a})$ is a boolean vector valued image. For a boolean vector valued image $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in (\mathbb{Z}_2^n)^{\mathbf{X}}$, the boolean complement of $\mathbf{b}$, denoted by $\mathbf{b}^c$, is defined as $\mathbf{b}^c = (\mathbf{b}_1^c, \ldots, \mathbf{b}_n^c)$, where the *i*th coordinate $\mathbf{b}_i^c$ is defined by Eq. 4.4.5.

(ii) Suppose *n* is an even positive integer and $\mathbf{a} \in (\mathbb{R}^n)^{\mathbf{X}}$. For $i = 1, \ldots, \frac{n}{2}$ and $r \in \mathbb{R}$ define $\bar{f}_{2i}(r) = r^2$ and $\bar{f}_{2i-1}(r) = r$. Then

$$f(\mathbf{a}) = \left(\mathbf{a}_1, \mathbf{a}_2^2, \ldots, \mathbf{a}_{n-1}, \mathbf{a}_n^2\right).$$

(iv) Define $f_1 : \mathbb{R}^2 \to \mathbb{R}$ by $f_1(x,y) = sin(x) + cosh(y)$ and $f_2 : \mathbb{R}^2 \to \mathbb{R}$ by $f_2(x,y) = cos(x) + sinh(y)$. Then the induced function $f : \left(\mathbb{R}^2\right)^{\mathbf{X}} \to \left(\mathbb{R}^2\right)^{\mathbf{X}}$ given by $f = (f_1, f_2)$. Applying *f* to an image $\mathbf{a} \in \left(\mathbb{R}^2\right)^{\mathbf{X}}$ results in the image

$$f(\mathbf{a}) = \{(\mathbf{x}, \mathbf{b}(\mathbf{x})) : \mathbf{b}(\mathbf{x}) = (sin(\mathbf{a}_1(\mathbf{x})) + cosh(\mathbf{a}_2(\mathbf{x})), cos(\mathbf{a}_1(\mathbf{x})) + sinh(\mathbf{a}_2(\mathbf{x}))), \ \mathbf{x} \in \mathbf{X}\}.$$

Thus, if we represent complex numbers as points in $\mathbb{R}^2$ and **a** denotes a complex valued image, then $f(\mathbf{a}) = sin(\mathbf{a})$, where *sin* denotes the complex sine function.

(v) Suppose $\mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$ and $\mathbb{G} = \prod_{k=1}^{m} \mathbb{F}_{j_k}$, where $\{j_1, j_2, \ldots, j_m\} \subset \{1, 2, \ldots, n\}$ with $j_1 < j_2 < \cdots < j_m$ and $\mathbb{F}_{j_k}$ denotes the $j_k\mathrm{th}$ factor of $\mathbb{F}$. Define

$$p_{j_1 j_2 \cdots j_m} : \mathbb{F} \to \mathbb{G}$$

by

$$p_{j_1 j_2 \cdots j_m}(r_1, r_2, \ldots, r_n) = (r_{j_1}, r_{j_2}, \ldots, r_{j_m}),$$

where $r_i \in \mathbb{F}_i$ and $r_{j_k} \in \mathbb{F}_{j_k}$.

The map $p_{j_1 j_2 \cdots j_m}$ deletes $n - m$ components from an element of $\mathbb{F}$. Since for $m = 1$ $p_{j_1} : \mathbb{F} \to \mathbb{F}_{j_1}$, the function $p_{j_1 j_2 \cdots j_m}$ generalizes the notion of the *i*th projection function.

The induced unary operation $p_{j_1 j_2 \cdots j_m} : \mathbb{F}^{\mathbf{X}} \to \mathbb{G}^{\mathbf{X}}$ deletes $n - m$ image components from a vector valued image $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$.

The general framework for binary operations on multivalued images is similar to that for unary operations. Suppose $\bigcirc : \mathbb{E} \times \mathbb{G} \to \mathbb{F}$, where $\mathbb{E} = \prod_{i=1}^{n} \mathbb{E}_i$, $\mathbb{G} = \prod_{i=1}^{\eta} \mathbb{G}_i$, and $\mathbb{F} = \prod_{j=1}^{m} \mathbb{F}_j$. Then the binary operation $\bigcirc$ generates a sequence of binary operations $\bigcirc_j : \mathbb{E} \times \mathbb{G} \to \mathbb{F}_j$ defined by $\bigcirc_j = p_j \circ \bigcirc$; i.e., $\mathbf{a} \bigcirc_j \mathbf{b} = p_j(\mathbf{a} \bigcirc \mathbf{b})$. Thus, every binary operation $\bigcirc : \mathbb{E} \times \mathbb{G} \to \mathbb{F}$ induces a binary operation $\bigcirc : \mathbb{E}^{\mathbf{X}} \times \mathbb{G}^{\mathbf{X}} \to \mathbb{F}^{\mathbf{X}}$ and a family of binary operations

$$\bigcirc_j : \mathbb{E}^{\mathbf{X}} \times \mathbb{G}^{\mathbf{X}} \to \mathbb{F}_j^{\mathbf{X}}.$$

Conversely, given a sequence of binary operations $\bigcirc_j : \mathbb{E} \times \mathbb{G} \to \mathbb{F}_j$, then this sequence induces a sequence of binary operations $\bigcirc_j : \mathbb{E}^{\mathbf{X}} \times \mathbb{G}^{\mathbf{X}} \to \mathbb{F}_j^{\mathbf{X}}$ and a binary operation $\bigcirc : \mathbb{E}^{\mathbf{X}} \times \mathbb{G}^{\mathbf{X}} \to \mathbb{F}^{\mathbf{X}}$ which is defined by

$$\mathbf{a} \bigcirc \mathbf{b} = (\mathbf{a} \bigcirc_1 \mathbf{b}, \mathbf{a} \bigcirc_2 \mathbf{b}, \ldots, \mathbf{a} \bigcirc_m \mathbf{b}),$$

$\forall \mathbf{a} \in \mathbb{E}^{\mathbf{X}}$ and $\forall \mathbf{b} \in \mathbb{G}^{\mathbf{X}}$.

As for unary operations, binary operations can also be induced from coordinate level binary operations. Suppose that for some indexing function $\lambda : \{1, \ldots, m\} \to \{1, \ldots, n\} \times \{1, \ldots, \eta\}$ there exists a sequence of binary operations

$$\bar{\bigcirc}_j : \mathbb{E}_{p_1(\lambda(j))} \times \mathbb{G}_{p_2(\lambda(j))} \to \mathbb{F}_j.$$

Then the sequence of induced binary operations $\bar{\bigcirc}_j : \mathbb{E}^{\mathbf{X}}_{p_1(\lambda(j))} \times \mathbb{G}^{\mathbf{X}}_{p_2(\lambda(j))} \to \mathbb{F}^{\mathbf{X}}_j$ induces another sequence of binary operations $\bigcirc_j : \mathbb{E}^{\mathbf{X}} \times \mathbb{G}^{\mathbf{X}} \to \mathbb{F}^{\mathbf{X}}_j$, which is defined by $\mathbf{a} \bigcirc_j \mathbf{b} = \mathbf{a}_{p_1(\lambda(j))} \bar{\bigcirc}_j \mathbf{b}_{p_2(\lambda(j))}$ $\forall \mathbf{a} \in \mathbb{E}^{\mathbf{X}}$ and $\forall \mathbf{b} \in \mathbb{G}^{\mathbf{X}}$. This new sequence then induces the binary operation $\bigcirc : \mathbb{E}^{\mathbf{X}} \times \mathbb{G}^{\mathbf{X}} \to \mathbb{F}^{\mathbf{X}}$ which is defined by

$$\mathbf{a} \bigcirc \mathbf{b} = (\mathbf{a} \bigcirc_1 \mathbf{b}, \mathbf{a} \bigcirc_2 \mathbf{b}, \ldots, \mathbf{a} \bigcirc_m \mathbf{b})\,.$$

Equivalently, we have that

$$\mathbf{a} \bigcirc \mathbf{b} = \left(\mathbf{a}_{p_1(\lambda(1))} \bar{\bigcirc}_1 \mathbf{b}_{p_2(\lambda(1))}, \ldots, \mathbf{a}_{p_1(\lambda(m))} \bar{\bigcirc}_m \mathbf{b}_{p_2(\lambda(m))}\right)\,. \tag{4.12.3}$$

In analogy with unary operations, if $m = n = \eta$, $\mathbb{E}$, $\mathbb{G}$ and $\mathbb{F}$ are homogeneous value sets, $\bar{\bigcirc}_i = \bar{\bigcirc}_j \ \ \forall i,j \in \mathbb{Z}^+_n$, and $\lambda(j) = (j,j) \ \ \forall i,j \in \mathbb{Z}^+_n$, then the induced operation 4.12.3 is said to be a *homogeneous* binary operation. Otherwise $\bigcirc$ is called a *mixed* or *heterogeneous* binary operation. If $\bigcirc$ is a homogeneous binary operation, then we let the coordinate operation $\bar{\bigcirc}_j$ represent the operation $\bigcirc$.

4.12.3 Examples:

(i) If $\mathbf{a}, \mathbf{b} \in (\mathbb{R}^n)^{\mathbf{X}}$ and for $j = 1, \ldots, n \ \ \bar{\bigcirc}_j = +$, then

$$\mathbf{a} + \mathbf{b} = (\mathbf{a}_1 + \mathbf{b}_1, \ldots, \mathbf{a}_n + \mathbf{b}_n)\,.$$

Similarly,

$$\mathbf{a} \cdot \mathbf{b} = (\mathbf{a}_1 \cdot \mathbf{b}_1, \ldots, \mathbf{a}_n \cdot \mathbf{b}_n)\,,$$
$$\mathbf{a} \vee \mathbf{b} = (\mathbf{a}_1 \vee \mathbf{b}_1, \ldots, \mathbf{a}_n \vee \mathbf{b}_n)\,,$$

and

$$\mathbf{a} \wedge \mathbf{b} = (\mathbf{a}_1 \wedge \mathbf{b}_1, \ldots, \mathbf{a}_n \wedge \mathbf{b}_n)\,.$$

If $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{R}^n$, then according to Eq. 4.3.3 we also have

$$\mathbf{r} + \mathbf{a} = (r_1 + \mathbf{a}_1, \ldots, r_n + \mathbf{a}_n)\,,$$
$$\mathbf{r} \cdot \mathbf{a} = (r_1 \cdot \mathbf{a}_1, \ldots, r_n \cdot \mathbf{a}_n)\,,$$

etc.

In the special case where $\mathbf{r} = (r, r, \ldots, r)$, we simply use the scalar $r \in \mathbb{R}$ and define $r + \mathbf{a} \equiv \mathbf{r} + \mathbf{a}$, $r \cdot \mathbf{a} \equiv \mathbf{r} \cdot \mathbf{a}$, and so on. Of course, these operations can be generalized to combine a scalar of any multivalue set $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$ with an image $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$.

(ii) Suppose that for $j = 1, \ldots, n$, $\bigcirc_j : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is defined by

$$(x_1, \ldots, x_n) \bigcirc_j (y_1, \ldots, y_n) = max\{x_i \vee y_j \,:\, 1 \le i \le j\}\,.$$

Then for $\mathbf{a}, \mathbf{b} \in (\mathbb{R}^n)^{\mathbf{X}}$ and $\mathbf{c} = \mathbf{a} \bigcirc \mathbf{b}$, the components of $\mathbf{c}(\mathbf{x}) = (\mathbf{c}_1(\mathbf{x}), \ldots, \mathbf{c}_n(\mathbf{x}))$ have values

$$\mathbf{c}_j(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \bigcirc_j \mathbf{b}(\mathbf{x}) = max\{\mathbf{a}_i(\mathbf{x}) \vee \mathbf{a}_j(\mathbf{x}) \,:\, 1 \le i \le j\}$$

for $j = 1, \ldots, n$.

(iii) Let $\bigcirc_1$ and $\bigcirc_2$ be two binary operations of $\mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ defined by

$$(x_1, x_2) \bigcirc_1 (y_1, y_2) = x_1 y_1 - x_2 y_2$$

and

$$(x_1, x_2) \bigcirc_2 (y_1, y_2) = x_1 y_2 + x_2 y_1 .$$

Now if $\mathbf{a}, \mathbf{b} \in \left(\mathbb{R}^2\right)^{\mathbf{X}}$ represent two complex valued images, then the product $\mathbf{c} = \mathbf{a} \bigcirc \mathbf{b}$ represents pointwise complex multiplication, namely

$$\mathbf{c}(\mathbf{x}) = (\mathbf{a}_1(\mathbf{x})\mathbf{b}_1(\mathbf{x}) - \mathbf{a}_2(\mathbf{x})\mathbf{b}_2(\mathbf{x}), \, \mathbf{a}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x}) + \mathbf{a}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})) .$$

Basic operations on single and multivalued images can be combined to form image processing operations of arbitrary complexity. Two such operations that have proven to be extremely useful in processing real vector valued images are the *winner take all* *j*th-coordinate maximum and minimum of two images. In order to define these two special operations, let $q_n : \mathbb{R} \to \mathbb{R}^n$ be defined by $q_n(r) = (r, r, \ldots, r)$. Now if $\mathbf{a}, \mathbf{b} \in (\mathbb{R}^n)^{\mathbf{X}}$, then the *jth-coordinate maximum* of **a** and **b** is defined as

$$\mathbf{a} \vee |_j \mathbf{b} = \mathbf{a} \cdot q_n\left[p_j\left(\chi_{\geq \mathbf{b}}(\mathbf{a})\right)\right] + \mathbf{b} \cdot q_n\left[p_j\left(\chi_{\geq \mathbf{b}}(\mathbf{a})\right)^c\right] , \tag{4.12.4}$$

where $1 \leq j \leq n$.

The *jth-coordinate minimum* is defined as

$$\mathbf{a} \wedge |_j \mathbf{b} = \mathbf{a} \cdot q_n\left[p_j\left(\chi_{\leq \mathbf{b}}(\mathbf{a})\right)\right] + \mathbf{b} \cdot q_n\left[p_j\left(\chi_{\leq \mathbf{b}}(\mathbf{a})\right)^c\right] , \tag{4.12.5}$$

where $1 \leq j \leq n$.

An easy interpretation of the images defined by Eqs. 4.12.4 and 4.12.5 is given by the following equivalent statements

$$\mathbf{a} \vee |_j \mathbf{b} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) \, : \, \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \text{ if } \mathbf{a}_j(\mathbf{x}) \geq \mathbf{b}_j(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = \mathbf{b}(\mathbf{x})\}$$

and

$$\mathbf{a} \wedge |_j \mathbf{b} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) \, : \, \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \text{ if } \mathbf{a}_j(\mathbf{x}) \leq \mathbf{b}_j(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = \mathbf{b}(\mathbf{x})\} ,$$

respectively. Since it is possible that for some $\mathbf{x} \in \mathbf{X}$ the pixel value $\mathbf{a}_j(\mathbf{x}) = \mathbf{b}_j(\mathbf{x})$, the operations of *j*th-coordinate maximum and minimum are not commutative.

The next example illustrates an application of a *j*th-coordinate maximum.

4.12.4 Example: (*Directional edge detection*). The output of this directional edge detection technique is a two-valued edge image in which each pixel has an intensity value as well as one of eight possible directional values. In this particular scheme, a gray scale image is convolved with the four 3×3 translation invariant templates shown in Figure 4.12.1

$\mathbf{t}(0) =$

1	1	1
-1	-1	-1

$\mathbf{t}(1) =$

1	1	
1		-1
	-1	-1

$\mathbf{t}(2) =$

-1		1
-1		1
-1		1

$\mathbf{t}(3) =$

	1	1
-1		1
-1	-1	

Figure 4.12.1 The four directional edge templates.

The resulting four images are then fused to form a single two-valued image **e**. Data fusion is accomplished by assigning to the resultant pixel the value of the largest magnitude (i.e. the largest absolute value) of the corresponding pixels in the four images *and* the direction θ associated with the template which yielded the largest magnitude if the pixel value is positive, or $\theta + 180° mod 360°$ if the value is negative. Thus, each pixel location of **e** has both an edge magnitude and an edge direction associated with it. It is customary to use the integers 0 through 7 to represent the eight directions 0° through 315°, respectively. The addition "$\theta + 180° mod 360°$" then becomes addition modulo eight, namely "$i + 4 mod 8$."

The image algebra translation of this algorithm is as follows. Let $f : \mathbb{R} \to \mathbb{R} \times \mathbb{Z}_8$ be defined by $f(r) = (|r|, 4\chi_{<0}(r))$, then

$\mathbf{a}_i := (0, i) + f(\mathbf{a} \oplus \mathbf{t}(i))$ for $i = 0, 1, 2, 3$, and

$\mathbf{e} := (\vee|_1)_{i=0}^{3} \mathbf{a}_i$.

Here each $\mathbf{a}_i$ is a vector valued image of form $\mathbf{a}_i(\mathbf{x}) = (\mathbf{a}_{i1}(\mathbf{x}), \mathbf{a}_{i2}(\mathbf{x}))$. More specifically, if $\mathbf{b}_i := \mathbf{a} \oplus \mathbf{t}(i)$, then

$$\mathbf{a}_i(\mathbf{x}) = \begin{cases} (|\mathbf{b}_i(\mathbf{x})|, 4 + i) & \text{if } \mathbf{b}_i(\mathbf{x}) < 0 \\ (\mathbf{b}_i(\mathbf{x}), i) & \text{if } \mathbf{b}_i(\mathbf{x}) \geq 0\,. \end{cases}$$

Note also that $(0, i) + f(\mathbf{b}_i)$ is simply the scalar addition of a vector and a vector valued image as described in Example **4.12.3**(i). Since $(\vee|_1)_{i=0}^{3} \mathbf{a}_i = ((\mathbf{a}_0 \vee|_1 \mathbf{a}_1) \vee|_1 \mathbf{a}_2) \vee|_1 \mathbf{a}_3$, it also follows that $\mathbf{e}(\mathbf{x}) = \mathbf{a}_k(\mathbf{x})$, where $\mathbf{a}_{k1}(\mathbf{x}) \geq \mathbf{a}_{i1}(\mathbf{x})$ for $i = 1, 2, 3$ and *k* denotes the smallest integer for which $\mathbf{a}_{k1}(\mathbf{x}) \geq \mathbf{a}_{i1}(\mathbf{x})$.

It should be obvious by now that multivalued image operations are special cases of image operations between different valued images as discussed in earlier sections of this chapter. For example, suppose that for $i = 1, \ldots, n$, each $(\mathbb{F}_i, \gamma_i)$ represents a commutative semigroup. Let $\mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$, and $\mathbf{r} = (r_1, \ldots, r_n)$ and $\mathbf{s} = (s_1, \ldots, s_n)$ denote two arbitrary elements of $\mathbb{F}$. Then for finite point sets the binary operation $\gamma : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ defined by

$$\mathbf{r}\gamma\mathbf{s} = (r_1\gamma_1 s_1, \ldots, r_n\gamma_n s_n)$$

induces a global reduce operation $\Gamma : \mathbb{F}^{\mathbf{X}} \to \mathbb{F}$ defined by

$$\Gamma\mathbf{a} = (\Gamma_1 \mathbf{a}_1, \ldots, \Gamma_n \mathbf{a}_n),$$

where $\Gamma_i : \mathbb{F}_i^{\mathbf{X}} \to \mathbb{F}_i$ is defined by Eq. 4.3.12. If $\mathbb{F}$ is homogeneous and $\gamma = (\gamma_1, \ldots, \gamma_n)$ is also homogeneous (i.e., $\gamma_i = \gamma_j \ \ \forall i, j \in \{1, \ldots, n\}$), then Γ is called a *homogeneous* reduce operation, otherwise Γ is said to be a *mixed* or *heterogeneous* reduce operation. If Γ is homogeneous, then we let Γ_i represent Γ. For example, if $(\mathbb{F}_i, \gamma_i) = (\mathbb{R}, +)$ for $i = 1, \ldots, n$, $\mathbf{a} \in (\mathbb{R}^n)^{\mathbf{X}}$, and $k = card(\mathbf{X})$, then

$$\begin{aligned} \Sigma\mathbf{a} &= (\Sigma\mathbf{a}_1, \ldots, \Sigma\mathbf{a}_n) \\ &= \left(\sum_{j=1}^{k} \mathbf{a}_1(\mathbf{x}_j), \ldots, \sum_{j=1}^{k} \mathbf{a}_n(\mathbf{x}_j) \right) \in \mathbb{R}^n . \end{aligned}$$

In contrast, the summation $\sum_{i=1}^{n} \mathbf{a}_i = \sum_{i=1}^{n} p_i(\mathbf{a}) \in \mathbb{R}^{\mathbf{X}}$ since each $\mathbf{a}_i \in \mathbb{R}^{\mathbf{X}}$.

In order to provide an example of a mixed reduce operation, let $\mathbb{F} = \mathbb{R} \times \mathbb{R}$, $\gamma_1 = +$, and $\gamma_2 = \vee$. Then for $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2) \in (\mathbb{R}^2)^{\mathbf{X}}$, we have

$$\Gamma\mathbf{a} = (\Sigma\mathbf{a}_1, \vee\mathbf{a}_2) = \left(\sum_{j=1}^{k} \mathbf{a}_1(\mathbf{x}_j), \bigvee_{j=1}^{k} \mathbf{a}_2(\mathbf{x}_j) \right) \in \mathbb{R}^2 .$$

4.13 Multivalued Templates

A template $\mathbf{t} \in (\mathbb{F}^{\mathbf{X}})^{\mathbf{Y}}$ is called a multivalued template if $\mathbb{F}$ can be written as a Cartesian product $\mathbb{F} = \prod_{i=1}^{n} \mathbb{F}_i$ with $n > 1$. In this case $\mathbf{t}$ is of form $\mathbf{t} = (\mathbf{t}_1, \ldots, \mathbf{t}_n)$, where $\mathbf{t}_i \in (\mathbb{F}_i^{\mathbf{X}})^{\mathbf{Y}}$ and $\mathbf{t}_{\mathbf{y}} = (\mathbf{t}_{1,\mathbf{y}}, \ldots, \mathbf{t}_{n,\mathbf{y}})$. If $\mathbb{F}$ is a homogeneous value set and $\mathbf{t}_1 = \mathbf{t}_2 = \cdots = \mathbf{t}_n$, then $\mathbf{t}$ is called a *homogeneous* template, otherwise $\mathbf{t}$ is said to be a *heterogeneous* template.

Since templates are special types of images, concepts and properties of multivalued images apply to multivalued templates as well. Specifically, operations on multivalued images are also operations on multivalued templates. The only operation not discussed in the multivalued setting is the product operator γ.

Template products of multivalued templates are governed by Eqs. 4.8.1 and 4.8.2. As for binary operations on multivalued images, an image-template product of a multivalued image and template induces a sequence of image-template products on the coordinate level, and products on the coordinate level induce products between multivalued images and templates. More precisely, suppose $(\mathbb{F}, \gamma)$ is a commutative semigroup, where $\mathbb{F} = \prod_{j=1}^{m} \mathbb{F}_j$, and γ denotes the operator $\gamma : \mathbb{E}^{\mathbf{X}} \times (\mathbb{G}^{\mathbf{X}})^{\mathbf{Y}} \to \mathbb{F}^{\mathbf{Y}}$

induced by $\bigcirc : \mathbb{E} \times \mathbb{G} \to \mathbb{F}$. Then $\bigcirc$ is of form $\bigcirc = (\bigcirc_1, \ldots, \bigcirc_m)$ and $\gamma = (\gamma_1, \ldots, \gamma_m)$, where $\bigcirc_j : \mathbb{E} \times \mathbb{G} \to \mathbb{F}_j$ and $\gamma_j : \mathbb{F}_j \times \mathbb{F}_j \to \mathbb{F}_j$. Thus $\circledgamma$ induces a sequence of products

$$\circledgamma_j : \mathbb{E}^{\mathbf{X}} \times \left(\mathbb{G}^{\mathbf{X}}\right)^{\mathbf{Y}} \to \mathbb{F}_j^{\mathbf{Y}},$$

where $\circledgamma_j$ is induced by $\bigcirc_j$ and γ_j.

Conversely, if for each $j = 1, \ldots, m$ $(\mathbb{F}_j, \gamma_j)$ is a commutative semigroup and $\bigcirc_j : \mathbb{E} \times \mathbb{G} \to \mathbb{F}_j$, then we obtain a sequence of induced products

$$\circledgamma_j : \mathbb{E}^{\mathbf{X}} \times \left(\mathbb{G}^{\mathbf{X}}\right)^{\mathbf{Y}} \to \mathbb{F}_j^{\mathbf{Y}}.$$

Thus, if $\mathbf{a} \in \mathbb{E}^{\mathbf{X}}$ and $\mathbf{t} \in \left(\mathbb{G}^{\mathbf{X}}\right)^{\mathbf{Y}}$, then the image $\mathbf{b} = \mathbf{a} \circledgamma_j \mathbf{t}$ is defined by

$$\mathbf{b}(\mathbf{y}) = \Gamma_j \left(\mathbf{a} \bigcirc_j \mathbf{t}_{\mathbf{y}}\right)$$

or, equivalently, by

$$\mathbf{b}(\mathbf{y}) = (\mathbf{a}(\mathbf{x}_1) \bigcirc_j \mathbf{t}_{\mathbf{y}}(\mathbf{x}_1))\gamma_j \cdots \gamma_j(\mathbf{a}(\mathbf{x}_n) \bigcirc_j \mathbf{t}_{\mathbf{y}}(\mathbf{x}_n)),$$

where $n = card(\mathbf{X})$.

If $\mathbb{E} = \prod_{i=1}^{n} \mathbb{E}_i$ and $\mathbb{G} = \prod_{i=1}^{\eta} \mathbb{G}_i$, then products can also be induced from coordinate level products. Following the method used for obtaining induced binary operations on multivalued images, suppose that for some indexing function $\lambda : \{1, \ldots, m\} \to \{1, \ldots, n\} \times \{1, \ldots, \eta\}$ there exists a sequence of binary operations

$$\bar{\bigcirc}_j : \mathbb{E}_{p_1(\lambda(j))} \times \mathbb{G}_{p_2(\lambda(j))} \to \mathbb{F}_j.$$

Then we obtain the operators

$$\bar{\circledgamma}_j : \mathbb{E}^{\mathbf{X}}_{p_1(\lambda(j))} \times (\mathbb{G}^{\mathbf{X}}_{p_2(\lambda(j))})^{\mathbf{Y}} \to \mathbb{F}_j^{\mathbf{X}} \quad (j = 1, \ldots, m), \tag{4.13.1}$$

$$\circledgamma_j : \mathbb{E}^{\mathbf{X}} \times (\mathbb{G}^{\mathbf{X}})^{\mathbf{Y}} \to \mathbb{F}_j^{\mathbf{X}} \quad (j = 1, \ldots, m), \tag{4.13.2}$$

and

$$\circledgamma : \mathbb{E}^{\mathbf{X}} \times (\mathbb{G}^{\mathbf{X}})^{\mathbf{Y}} \to \mathbb{F}^{\mathbf{X}}. \tag{4.13.3}$$

The operator $\bar{\circledgamma}_j$ in Eq. 4.13.1 is induced by $\bar{\bigcirc}_j$ and γ_j as defined by Eq. 4.8.1. The operator $\circledgamma_j$ in Eq. 4.13.2 is defined by

$$\mathbf{a} \circledgamma_j \mathbf{t} = \mathbf{a}_{p_1(\lambda(j))} \bar{\circledgamma}_j \mathbf{t}_{p_2(\lambda(j))}, \tag{4.13.4}$$

and the operator $\circledgamma$ (Eq. 4.13.3) is defined by

$$\mathbf{a} \circledgamma \mathbf{t} = (\mathbf{a} \circledgamma_1 \mathbf{t}, \ldots, \mathbf{a} \circledgamma_m \mathbf{t}), \tag{4.13.5}$$

where each $\circledgamma_j$ denotes the operation defined in Eq. 4.13.4. Note that if $\mathbf{b} = \mathbf{a} \circledgamma \mathbf{t}$, then

$$\begin{aligned}\mathbf{b}(\mathbf{y}) &= (\Gamma_1 (\mathbf{a} \bigcirc_1 \mathbf{t}_{\mathbf{y}}), \ldots, \Gamma_m (\mathbf{a} \bigcirc_m \mathbf{t}_{\mathbf{y}})) \\ &= \left(\Gamma_1 \left(\mathbf{a}_{p_1(\lambda(1))} \bar{\bigcirc}_1 \mathbf{t}_{p_2(\lambda(1)),\mathbf{y}}\right), \ldots, \Gamma_m \left(\mathbf{a}_{p_1(\lambda(m))} \bar{\bigcirc}_m \mathbf{t}_{p_2(\lambda(m)),\mathbf{y}}\right)\right),\end{aligned}$$

where

$$\begin{aligned}&\Gamma_j \left(\mathbf{a}_{p_1(\lambda(j))} \bar{\bigcirc}_j \mathbf{t}_{p_2(\lambda(j)),\mathbf{y}}\right) \\ &= \left[\mathbf{a}_{p_1(\lambda(j))}(\mathbf{x}_1) \bar{\bigcirc}_j \mathbf{t}_{p_2(\lambda(j)),\mathbf{y}}(\mathbf{x}_1)\right] \gamma_j \cdots \gamma_j \left[\mathbf{a}_{p_1(\lambda(j))}(\mathbf{x}_k) \bar{\bigcirc}_j \mathbf{t}_{p_2(\lambda(j)),\mathbf{y}}(\mathbf{x}_k)\right].\end{aligned}$$

If $\mathbb{E}$, $\mathbb{G}$, and $\mathbb{F}$ are homogeneous sets, $m = n = \eta$, and both $\gamma = (\gamma_1, \ldots, \gamma_m)$ and $\bar{\bigcirc} = \left(\bar{\bigcirc}_1, \ldots, \bar{\bigcirc}_m\right)$ are homogeneous operations, then $\circledgamma$ is called a *homogeneous* product operation, otherwise $\circledgamma$ is called a *mixed* or *heterogeneous* product operation. If $\circledgamma$ is a homogeneous product operation, then we represent $\circledgamma$ by $\bar{\circledgamma}_j$.

Left products and products between two multivalued templates are defined in a similar fashion.

4.13.1 Examples:

(i) Suppose $\mathbb{E} = \mathbb{G} = \mathbb{F} = \mathbb{R} \times \mathbb{R}_{\pm\infty}$ so that $\mathbb{E}_1 = \mathbb{G}_1 = \mathbb{F}_1 = \mathbb{R}$ and $\mathbb{E}_2 = \mathbb{G}_2 = \mathbb{F}_2 = \mathbb{R}_{\pm\infty}$. Let $\bar{\bigcirc}_1 : \mathbb{E}_1 \times \mathbb{G}_1 \to \mathbb{F}_1$ and $\bar{\bigcirc}_2 : \mathbb{E}_2 \times \mathbb{G}_2 \to \mathbb{F}_2$ denote multiplication and extended addition, respectively. If we use the semigroup structure $(\mathbb{F}, \gamma)$, where $\gamma = (+, \vee)$, then we obtain the heterogeneous image–template product

$$\mathbf{a} \,\textcircled{γ}\, \mathbf{t} = (\mathbf{a}_1 \oplus \mathbf{t}_1, \mathbf{a}_2 \boxed{\vee}\, \mathbf{t}_2)\,,$$

where $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2) \in (\mathbb{R} \times \mathbb{R}_{\pm\infty})^{\mathbf{X}}$ and $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2) \in \left[(\mathbb{R} \times \mathbb{R}_{\pm\infty})^{\mathbf{X}}\right]^{\mathbf{Y}}$.
In particular, if

$$\mathbf{t}_{1,\mathbf{y}} = \begin{array}{|c|c|c|}\hline -1 & & 1 \\ \hline -1 & & 1 \\ \hline -1 & & 1 \\ \hline \end{array} \qquad \mathbf{t}_{2,\mathbf{y}} = \begin{array}{ccc} & \boxed{0} & \\ \boxed{0} & \boxed{0} & \boxed{0} \\ & \boxed{0} & \end{array}$$

then the first coordinate of $\mathbf{a} \,\textcircled{γ}\, \mathbf{t}$ represents a vertical edge enhancement and the second coordinate a dilation or local maximization.

(ii) Suppose that $\mathbf{a} \in \mathbb{R}^{\mathbf{X}}$, $\mathbf{t} = (\mathbf{t}_1, \ldots, \mathbf{t}_n) \in \left((\mathbb{R}^n)^{\mathbf{X}}\right)^{\mathbf{Y}}$, and $\bigcirc : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ denotes the binary operation of scalar multiplication of *n*-dimensional vectors. If $(\mathbb{F}, \gamma) = (\mathbb{R}^n, +)$, where + denotes vector addition, then we obtain the homogeneous image–template product

$$\mathbf{a} \oplus \mathbf{t} = (\mathbf{a} \oplus \mathbf{t}_1, \ldots, \mathbf{a} \oplus \mathbf{t}_n) \in (\mathbb{R}^n)^{\mathbf{Y}}$$

which turns a single-valued image into a vector-valued image. The value of the image $\mathbf{b} = \mathbf{a} \oplus \mathbf{t}$ at a point $\mathbf{y} \in \mathbf{Y}$ is given by

$$\begin{aligned}
\mathbf{b}(\mathbf{y}) &= \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x}) \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \cdot (\mathbf{t}_{1,\mathbf{y}}(\mathbf{x}), \ldots, \mathbf{t}_{n,\mathbf{y}}(\mathbf{x})) \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \Big(\mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{1,\mathbf{y}}(\mathbf{x}), \ldots, \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{n,\mathbf{y}}(\mathbf{x})\Big) \\
&= \left(\sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{1,\mathbf{y}}(\mathbf{x}), \ldots, \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{n,\mathbf{y}}(\mathbf{x})\right).
\end{aligned}$$

4.14 The Algebra of Lists

A common activity in computer vision, and computer science in general, is the arrangement and representation of data in some ordered fashion. The ordering usually depends on both, the particular task at hand as well as the properties of the data. One of the most utilitarian of such ordered data structures are *lists*. In this section we define several basic concepts associated with lists and show how these concepts fit into the larger framework of image algebra.

Given a set L, a *list* over L is a finite sequence of elements of L. Such lists are denoted by angle brackets containing the ordered, comma separated elements. For example,

$$\langle 1, 3, 5, 7, 9 \rangle$$

represents the list over the integers containing elements 1, 3, 5, 7, and 9 in that order. The *empty* list, containing no elements, is denoted by $\langle \rangle$.

More formally, a *list* over the set L is an element of the set

$$L^* = \bigcup_{n=0}^{\infty} L^{\mathbb{Z}_n},$$

where we define $\mathbb{Z}_0 = \varnothing$ so that $L^{\mathbb{Z}_0} = \{\langle\rangle\}$.

If $l \in L^*$, then $l \in L^{\mathbb{Z}_n}$ for some $n \in \mathbb{N}$ and $l(i) \in L$ for $i = 0, 1, \ldots, n-1$. It is customary to denote the elements $l(i)$ of L by l_i and to specify the function l by enumerating the l_i's using the natural order of $\mathbb{Z}_n$. That is, we define $l_i \equiv l(i)$ and specify l by $l = \langle l_0, l_1, \ldots, l_{n-1} \rangle$.

Since lists are images, namely functions from a point set to a value set (see Theorem **4.14.1**), many of the concepts defined previously apply to lists as well. For example, the *length* of a list $l = \langle l_0, l_1, \ldots, l_{n-1} \rangle$, which is denoted by $\|l\|$, is defined as the number of elements in the sequence defining l. Hence $\|l\|$ can be computed using the following statement:

$$\|l\| := card[domain(l)].$$

Note that if $l = \langle \rangle$, then $\|\langle\rangle\| = 0$, and if $l = \langle l_0, l_1, \ldots, l_{n-1} \rangle$, then $\|l\| = n$.

The fundamental binary operation on L^* is concatenation. The concatenation operator $\cdot$ joins two lists together. If, for example, if l is the list $l = \langle l_0, \ldots, l_k \rangle$ and l' denotes the list $l' = \langle l'_0, \ldots, l'_n \rangle$ then $l \cdot l'$ denotes the list

$$\langle l_0, \ldots, l_k, l_{k+1}, \ldots, l_{k+n+1} \rangle,$$

where $l_{k+i} = l'_{i-1}$ for $i = 1, \ldots, n+1$. It follows that $\langle\rangle \cdot l = l \cdot \langle\rangle = l$ for any $l \in L^*$ and that $\|l \cdot l'\| = \|l\| + \|l'\|$.

Since concatenation is associative and the empty list acts as an identity under the operation of concatenation, we have the following result:

4.14.1 Theorem. $(L^*, \cdot)$ *is a monoid.*

4.14.2 Definition. Two lists $l, l' \in L^*$ are said to be *range equivalent* whenever $range(l) = range(l')$. Two lists l and l' are said to be *equivalent* if $\|l\| = \|l'\|$ and there exists a permutation $\sigma : \mathbb{Z}_{\|l\|} \to \mathbb{Z}_{\|l\|}$ such that $l_{\sigma(i)} = l'_i \ \forall\, i \in \mathbb{Z}_{\|l\|}$.

Obviously, equivalence implies range equivalence while the converse need not be true.

For a given string $l \in L^*$ and an arbitrary element $x \in L$, the *occurrence number* of x in l, denoted by $n(x, l)$, is defined as $n(x, l) = card\big[l^{-1}(x)\big]$. Thus the occurrence number tells the number of times an element of L occurs in a string l.

4.14.3 Theorem. $\sum_{x \in L} n(x,l) = \|l\|$.

Proof: If $x \notin range(l)$, then $n(x,l) = 0$. Therefore

$$\sum_{x \in L} n(x,l) = \sum_{x \in range(l)} n(x,l).$$

Let $n = \|l\|$ and $\{x_1, x_2, \ldots, x_m\} = range(l)$. If $x_i \neq x_j$, then

$$l^{-1}(x_i) \cap l^{-1}(x_j) = \varnothing,$$

for otherwise there exists an integer $k \in l^{-1}(x_i) \cap l^{-1}(x_j)$, which means that contrary to our assumption we have $x_i = l(k) = x_j$.

Since

$$\bigcup_{i=1}^{m} l^{-1}(x_i) = \mathbb{Z}_n,$$

it follows that $\left\{l^{-1}(x_i) : i = 1, 2, \ldots, m\right\}$ is a partition of $\mathbb{Z}_n$. Therefore,

$$\begin{aligned} \|l\| = n = card(\mathbb{Z}_n) &= card\left(\bigcup_{i=1}^{m} l^{-1}(x_i)\right) \\ &= card\big(l^{-1}(x_1)\big) + card\big(l^{-1}(x_2)\big) + \cdots + card\big(l^{-1}(x_m)\big) \\ &= \sum_{x \in \{x_1, x_2, \ldots, x_m\}} n(x,l). \end{aligned}$$

Q.E.D.

The basic unary operation on lists is the projection map π_i defined by

$$\pi_i(l) = \begin{cases} \langle l_i \rangle & \text{if } 0 \leq i < \|l\| \\ \langle\rangle & \text{if } i \geq \|l\|, \end{cases} \tag{4.14.1}$$

where $i \in \mathbb{N}$.

List projections belong to a class of important unary operations on lists that do not increase the range of a list. Specifically, a function $r : L^* \to L^*$ is called *range-limited* if $range[r(l)] \subset range(l) \quad \forall\, l \in L^*$.

Range–limited maps can be conveniently defined in terms of a parameterized projection map $\pi : \mathbb{N}^* \times L^* \to L^*$ which is defined as follows: For each $(s,l) \in \mathbb{N}^* \times L^*$, with $s = \langle s_0, \ldots, s_k \rangle$, define

$$\pi(s,l) = \pi_{s_0}(l) \cdot \pi_{s_1}(l) \cdot \cdots \cdot \pi_{s_k}(l). \tag{4.14.2}$$

It will be convenient to use the notation l_s, and $l_{\langle s_0, \ldots, s_k \rangle}$ to denote $\pi(s,l)$. Some confusion may arise when using the notation l_s without specifying *s*; e.g., if *s* is a list, then $l_s \in L^*$, but if *s* represents an integer, then $l_s \in L$. It is also common practice to view $\pi(s,\cdot)$ as a unary operation on L^*, where the unary operation is denoted by π_s, and defined by

$$\begin{aligned} \pi_s &: L^* \to L^*, \text{ where} \\ \pi_s(l) &= \pi(s,l)\,. \end{aligned} \tag{4.14.3}$$

Again, if *s* is not specified, then it may not be clear whether π_s refers to the map defined by Eq. 4.14.1 or to the map defined by Eq. 4.14.3. Hopefully, the content of discussion leaves no doubt as to which of the two maps an author is referring to. For example, if we specify *s* as the single integer list $s = \langle i \rangle$ and $0 \le i < \|l\|$, then it should be clear that

$$\pi_s(l) = \pi_{\langle i \rangle}(l) = l_s = l_{\langle i \rangle} = \langle l_i \rangle = \pi_i(l)\,.$$

Since π_s is defined in terms of concatenation, various relationships between π_s and concatenation exist. The following two properties are obvious:

1. $l_{\langle j_0,\ldots,j_k \rangle} = l_{\langle j_0,\ldots,j_i \rangle} \cdot l_{\langle j_{i+1},\ldots,j_k \rangle}$, for any $i = 0,\ldots,k-1$
2. $\pi_{s\cdot t}(l) = \pi_s(l) \cdot \pi_t(l)$, where $l \in L^*$, $s \in \mathbb{N}^*$, and $t \in \mathbb{N}^*$

Such properties can be used to manipulate strings in a coherent and concise fashion.

Lists are often viewed or defined as sequences. For instance, if $n = \|l\|$, then *l* can be viewed as an element of $\prod_{i=1}^{n} L$. Thus, in addition to the above defined projections, we also can apply the standard projections

$$\begin{aligned} p_i &: L^{\mathbb{Z}_n} \to L \\ p_i &: l \rightarrowtail l_i \end{aligned}$$

to lists. The maps π_i and p_i are distinct but similar in behavior. For example, if $x \in L$ and $l \in L^*$ such that $\|l\| = n-1$, then

$$\pi_i(\langle x \rangle \cdot l) = \begin{cases} \langle x \rangle & \text{if } i = 0 \\ \langle l_{i-1} \rangle & \text{if } 0 < i \le \|l\| \end{cases} \quad \text{while} \quad p_i(\langle x \rangle \cdot l) = \begin{cases} x & \text{if } i = 0 \\ l_{i-1} & \text{if } 0 < i \le \|l\|\,. \end{cases}$$

The operations of concatenation and projections can be used to define more sophisticated operations on lists. The next example provides an illustration of this.

4.14.4 Example: (*Median filter*). The median filter is a nonlinear technique for noise suppression. It consists of replacing each pixel value $\mathbf{a}(\mathbf{y})$ of an image $\mathbf{a} \in \mathbb{R}^{\mathbf{X}}$ with the median value of the pixels in a neighborhood *N* of $\mathbf{y}$. The filtered image, $\mathbf{b}$, is given by $\mathbf{b}(\mathbf{y}) = median(\{\mathbf{a}(\mathbf{x}) : \mathbf{x} \in N(\mathbf{y})\})$, where $N : \mathbf{X} \to 2^{\mathbf{X}}$ is a neighborhood configuration function about the point $\mathbf{y} \in \mathbf{X}$.

In order to provide an algorithm that expresses the median filter in the language of the list algebra, let $\mathbb{L}$ be the set of lists with real valued elements and define the binary operation $\bigcirc : \mathbb{R} \times \mathbb{Z}_2 \to \mathbb{L}$ by

$$\bigcirc(r,i) = \begin{cases} \langle r \rangle & \text{if } i = 1 \\ \langle\rangle & \text{if } i = 0. \end{cases}$$

Define $\gamma \ : \ \mathbb{L} \times \mathbb{L} \to \mathbb{L}$ recursively as follows:

$$\gamma(k,l) = \begin{cases} k & \text{if } l = \langle\rangle \\ l & \text{if } k = \langle\rangle \\ \langle r_1 \rangle \cdot \gamma(k', \ l) & \text{if } k = \langle r_1 \rangle \cdot k', \ \ l = \langle r_2 \rangle \cdot l', \text{ and } r_1 \leq r_2 \\ \langle r_2 \rangle \cdot \gamma(k, \ l') & \text{if } k = \langle r_1 \rangle \cdot k', \ \ l = \langle r_2 \rangle \cdot l', \text{ and } r_1 > r_2 \,. \end{cases}$$

The algorithm now proceeds as follows:
Let $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ be defined as

$$\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in N(\mathbf{y}) \\ 0 & \text{otherwise.} \end{cases}$$

Define the median function for a list l of length $n = \|l\|$ by

$$median(l) = \begin{cases} p_{\frac{n+1}{2}}(l) & \text{if } n \text{ is odd} \\ \frac{p_{\frac{n}{2}}(l) + p_{\frac{n}{2}+1}(l)}{2} & \text{if } n \text{ is even.} \end{cases}$$

Then, the filtered image $\mathbf{b}$ is given by

$$\mathbf{b} := median(\mathbf{a} \circledgamma \mathbf{t}).$$

The median filter defined in the above example should not be confused with the mean or local averaging filter described in Example **4.8.1**. While the local averaging filter is generally more effective in suppressing smoothly generated noise, the median filter is much more effective in reducing the effects of discrete impulse noise. Figure **4.14.1** provides a comparative example of the effects of these filters on an image containing a fair amount of discrete impulse noise.

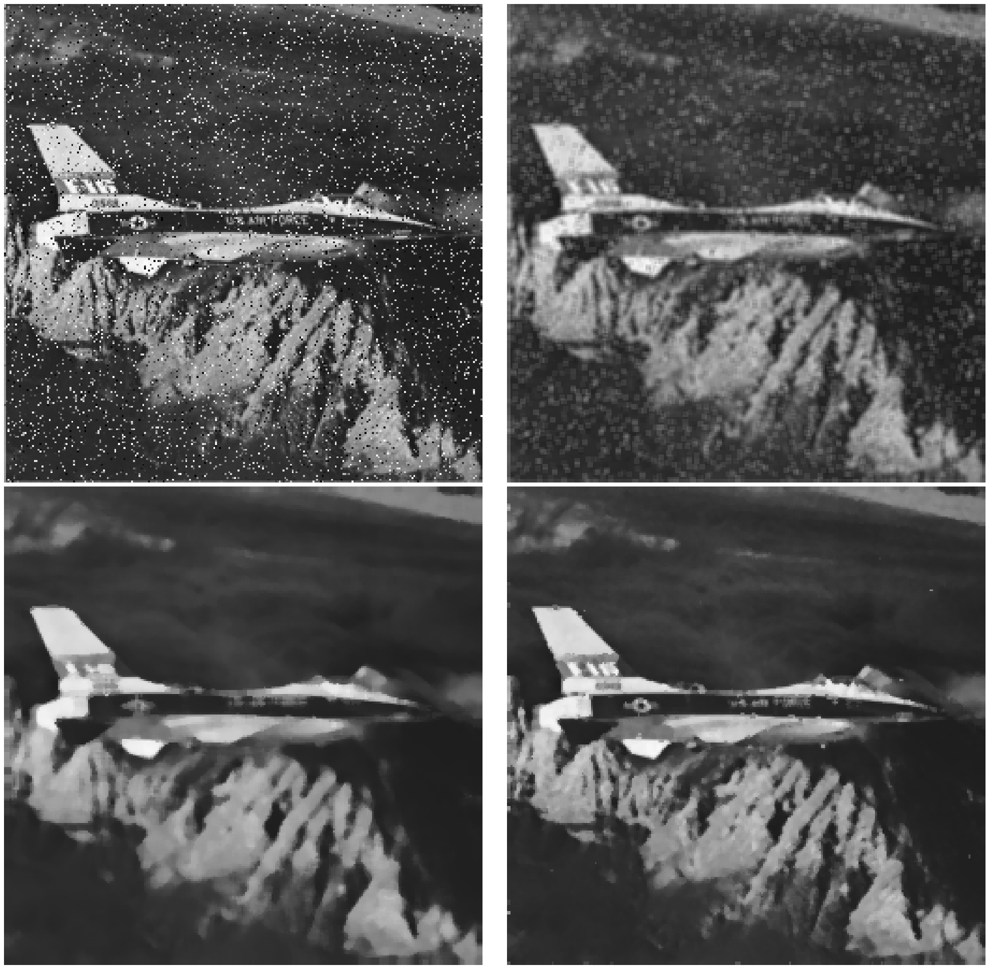
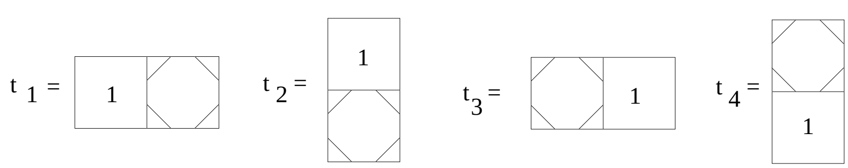


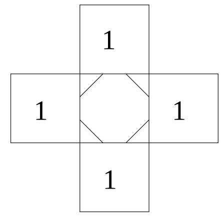
Figure 4.14.1 Example of mean vs. median filter. The top row shows the noisy source image on the left and the mean filtered image using a 3×3 local averaging template is shown on the right. The bottom row shows the effect of median filtering. The image on the left is median filtered using a 5×5 neighborhood, while the image on the right is median filtered using a 3×3 neighborhood.

Since the median filter represents an image-to-image transformation, we could have used previous methodologies for an algebraic specification of the median filter algorithm. For example, if $N(\mathbf{y})$ represents a von Neumann neighborhood, then for $i = 1, 2, 3, 4$, let $\mathbf{t}_i$ be defined as follows:



Note that if $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3 + \mathbf{t}_4$, then $\mathbf{t}$ is given by

$\mathbf{t_y} =$



If $\mathbf{a}_i := \mathbf{a} \bigoplus \mathbf{t}_i \;\; \forall i \in \{1, 2, 3, 4\}$, then the median image $\mathbf{m}$ of **a** can be computed as follows:

$\mathbf{a}_0 := \mathbf{a}$
$\mathbf{m} := \mathbf{a}_0 \vee \mathbf{a}_1 \vee \mathbf{a}_2$
`for` $i := 0$ to 2 `loop`
  `for` $j := i + 1$ to 3 `loop`
    `for` $k := j + 1$ to 4 `loop`
      $\mathbf{m} := (\mathbf{a}_i \vee \mathbf{a}_k \vee \mathbf{a}_j) \wedge \mathbf{m}$
    `end loop`
  `end loop`
`end loop`

This scheme can be generalized by decomposing any size template into the appropriate number of one-point templates. However, the number of convolutions (iterations) for k single-pixel templates (with k odd) is

$$\binom{k+1}{\frac{k+1}{2}}.$$

This provides for an extremely inefficient algorithm for most computers. For some computer architectures, however, this neighborhood decomposition into single-pixel templates is a very efficient method for implementing order statistic filters [2, 1].

Bibliography

[1] P.C. Coffield. An architecture for processing image algebra operations. In *Image Algebra and Morphological Image Processing III*, volume 1769 of *Proceedings of SPIE*, pages 178–189, San Diego, CA, July 1992.

[2] P.C. Coffield. *An Electro-Optical Image Algebra Processing System for Automatic Target Recognition*. Ph.D. dissertation, University of Florida, Gainesville, August 1992.

[3] J. Dugundji. *Topology*. Allen & Bacon, Inc., Boston, 1966.

[4] G.X. Ritter, J.L. Davidson, and J.N. Wilson. Beyond mathematical morphology. In *Visual Communication and Image Processing II*, volume 845 of *Proceedings of SPIE*, pages 260–269, Cambridge, MA, October 1987.

[5] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

CHAPTER 5
TECHNIQUES FOR THE COMPUTATION OF GENERAL LINEAR TRANSFORMS

This chapter presents methods for the computation of discrete linear image transformations. Linear image transformations represent a large class of fundamental image processing operations in which images are transformed by the use of linear combinations of pixels. Special consideration is given to the linear operator $\oplus$ to represent and optimize linear image processing transforms.

5.1 Image Algebra and Linear Algebra

In Sections **4.9** and **4.10** we established natural connections between matrix algebra and the algebra of templates. A consequence of these connections is that for any given field $(\mathbb{F}, +, \times)$, the ring $\left(\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}, +, \oplus\right)$ is algebraically the same as the matrix algebra $(\mathrm{M}_{mn\times mn}(\mathbb{F}), +, \times)$, where $m = card(\mathbf{X})$, $n = card(\mathbf{Y})$, and $\times$ denotes both (induced) matrix multiplication and multiplication in $\mathbb{F}$. Since any linear algebra of dimension *mn* is isomorphic to a subalgebra of $\mathrm{M}_{mn\times mn}(\mathbb{F})$ (Section **3.9**), linear algebra can be viewed as a special case or as a subalgebra of image algebra. Thus, any relationship which can be expressed using the notation of finite-dimensional linear algebra can also be expressed using the usual notation of image algebra. Image algebra notation, however, differs from the conventional matrix/vector notation in that it reflects the way computations are carried out in digital image processing.

In comparison to image algebra, linear algebra is an ancient and well-established area of mathematics encompassing a great wealth of accumulated knowledge in terms of theorems and computational techniques. The relationship between image algebra and linear algebra provides the link to this knowledge and offers direct methods for optimizing and implementing linear image transforms on digital computers.

The pertinent link between image algebra and linear algebra is the pair of isomorphisms $\psi : \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathrm{M}_{n\times n}(\mathbb{F})$ and $\nu : \mathbb{F}^{\mathbf{X}} \to \mathbb{F}^n$ defined in Section **4.9**. These isomorphisms, however, are *not* unique. Their definitions are based on the ordering of the points of $\mathbf{X}$. Distinguishing the points of $\mathbf{X}$ by subscripts, say $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, provides for a linear ordering of $\mathbf{X}$ by use of the rule: $\mathbf{x}_i \prec \mathbf{x}_j \Leftrightarrow i < j$. The isomorphism ψ is dependent on this linear ordering since by definition, $\psi(\mathbf{t}) = (t_{ij})_{n\times n}$, where $t_{ij} = \mathbf{t}_{\mathbf{x}_j}(\mathbf{x}_i)$. A similar comment applies to the vector space isomorphism ν. It follows that for each different ordering of $\mathbf{X}$ one obtains a different isomorphism. The next theorem describes the relationship between them.

5.1.1 Theorem. *Suppose* $\mathbf{X}_1 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ *and* $\mathbf{X}_2 = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ *are two different orderings of* $\mathbf{X}$. *Let* $\psi_1 : \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathrm{M}_{n\times n}(\mathbb{F})$ *and* $\psi_2 : \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathrm{M}_{n\times n}(\mathbb{F})$ *be two isomorphisms defined relative to the ordering of* $\mathbf{X}_1$ *and* $\mathbf{X}_2$*, respectively. Then there exists a permutation matrix* $P \in \mathrm{M}_{n\times n}(\mathbb{F})$ *such that*

$$\psi_1(\mathbf{t}) = P\psi_2(\mathbf{t})P'$$

$\forall\, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

Conversely, if there exists a permutation matrix $P \in \mathrm{M}_{n\times n}(\mathbb{F})$ *and for every template* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *a matrix* $A_{\mathbf{t}} \in \mathrm{M}_{n\times n}(\mathbb{F})$ *with the property that* $\psi_1(\mathbf{t}) = PA_{\mathbf{t}}P'$*, then there exists an ordering* *on* $\mathbf{X}$ *such that if* $\psi : \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathrm{M}_{n\times n}(\mathbb{F})$ *is defined relative to this ordering, then* $\psi(\mathbf{t}) = A_{\mathbf{t}}$ $\forall\, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

Proof: Let $\sigma \in S_n$ be the permutation defined by $\mathbf{x}_{\sigma(i)} = \mathbf{y}_i$ for $i \in \{1, \ldots, n\}$. Recall that the permutation matrix $P_\sigma = (p_{ij})_{n\times n}$ is defined by

$$p_{ij} = \begin{cases} 1 & \text{if } j = \sigma(i) \\ 0 & \text{if } j \neq \sigma(i)\,, \end{cases}$$

where 0 and 1 denote the additive and multiplicative identity of $\mathbb{F}$, respectively.

Let $\psi_1(\mathbf{t}) = (t_{ij})_{n\times n}$ and $\psi_2(\mathbf{t}) = (s_{ij})_{n\times n}$. Then

$$P_\sigma \psi_1(\mathbf{t}) P'_\sigma = \left(t_{\sigma(i),\sigma(j)}\right)_{n\times n}.$$

But $s_{ij} = \mathbf{t}_{\mathbf{y}_j}(\mathbf{y}_i) = \mathbf{t}_{\mathbf{x}_{\sigma(j)}}\left(\mathbf{x}_{\sigma(i)}\right) = t_{\sigma(i),\sigma(j)}$. Therefore $\psi_2(\mathbf{t}) = P_\sigma \psi_1(\mathbf{t}) P'_\sigma$ or, equivalently, $\psi_1(\mathbf{t}) = P\psi_2(\mathbf{t})P'$, where $P = P_\sigma^{-1}$. This proves the first part of the theorem.

In order to prove the second part, let *P* be the permutation matrix with the property $\psi_1(\mathbf{t}) = PA_{\mathbf{t}}P'$. Since *P* is a permutation matrix, there exists a permutation $\sigma \in S_n$ such that $P = P_\sigma$. Define an ordering $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ on $\mathbf{X}$ by setting $\mathbf{y}_i = \mathbf{x}_{\sigma^{-1}(i)}$ and an isomorphism $\psi : \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathrm{M}_{n\times n}(\mathbb{F})$ relative to this ordering. By the first part of the theorem we have

$$P'\psi_1(\mathbf{t})P = \psi(\mathbf{t}) \quad \forall\, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$$

and by the hypothesis of the second part of the theorem we have that

$$P'\psi_1(\mathbf{t})P = A_{\mathbf{t}} \quad \forall\, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}.$$

Hence $\psi(\mathbf{t}) = A_{\mathbf{t}}$ $\forall\, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

Q.E.D.

According to Theorem **5.1.1**, two isomorphisms defined in terms of distinct linear orderings of a point set differ only by a permutation. For this reason we shall use the symbol ψ to denote the isomorphism $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathrm{M}_{n\times n}(\mathbb{F})$ described in Chapter **4**. The ordering on $\mathbf{X}$ will be understood as given and no mention of it will be made in general. The special case where $\mathbf{X}$ is an $m \times n$ array, $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ or $\mathbf{X} = \mathbb{Z}_m^+ \times \mathbb{Z}_n^+$, is of particular interest. Unless otherwise specified, in these two cases we shall always assume the lexicographical or row-scanning order.

The isomorphisms ν and ψ allowed for the easy proofs of basic properties governing the general convolution operator $\circledgamma$ (Chapter **4**). We conclude this section by reformulating the most pertinent of these properties in terms of the field operations $+$ and $\times$.

Suppose $(\mathbb{F}, +, \times)$ is a field, $\mathbf{a}, \mathbf{b} \in \mathbb{F}^{\mathbf{X}}$, $\mathbf{r}, \mathbf{s}, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$, and $c \in \mathbb{F}$. Then

$$\begin{array}{ll} 1. & (\mathbf{a} + \mathbf{b}) \oplus \mathbf{t} = (\mathbf{a} \oplus \mathbf{t}) + (\mathbf{b} \oplus \mathbf{t}) \\ 2. & (c \cdot \mathbf{a}) \oplus \mathbf{t} = c \cdot (\mathbf{a} \oplus \mathbf{t}) \\ 3. & \mathbf{a} \oplus (\mathbf{s} + \mathbf{t}) = (\mathbf{a} \oplus \mathbf{s}) + (\mathbf{a} \oplus \mathbf{t}) \\ 4. & \mathbf{a} \oplus (\mathbf{s} \oplus \mathbf{t}) = (\mathbf{a} \oplus \mathbf{s}) \oplus \mathbf{t} \\ 5. & \mathbf{r} \oplus (\mathbf{s} \oplus \mathbf{t}) = (\mathbf{r} \oplus \mathbf{s}) \oplus \mathbf{t} \\ 6. & \mathbf{r} \oplus (\mathbf{s} + \mathbf{t}) = (\mathbf{r} \oplus \mathbf{s}) + (\mathbf{r} \oplus \mathbf{t}) \\ 7. & (\mathbf{r} + \mathbf{s}) \oplus \mathbf{t} = (\mathbf{r} \oplus \mathbf{t}) + (\mathbf{s} \oplus \mathbf{t}) \end{array} \tag{5.1.1}$$

Properties 1 and 2 of Eq. 5.1.1 express the linearity of the convolution operator $\oplus$. Specifically, the transformation $F_{\mathbf{t}} : \mathbb{F}^{\mathbf{X}} \to \mathbb{F}^{\mathbf{X}}$, defined by $F_{\mathbf{t}}(\mathbf{a}) = \mathbf{a} \oplus \mathbf{t}$ is linear since it follows from Property 1 that $F_{\mathbf{t}}(\mathbf{a} + \mathbf{b}) = F_{\mathbf{t}}(\mathbf{a}) + F_{\mathbf{t}}(\mathbf{b})$ and Property 2 that $F_{\mathbf{t}}(c \cdot \mathbf{a}) = c \cdot F_{\mathbf{t}}(\mathbf{a})$.

The remaining five properties provide the necessary tools for linear transform optimization and inversion.

5.2 Fundamentals of Template Decomposition

Template decomposition plays a fundamental role in image processing algorithm optimization. It provides a method for reducing the cost in computation and, therefore, increases the computational efficiency of image processing algorithms. This goal can be achieved in two ways, either by reducing the number of arithmetic computations in an algorithm or by restructuring an algorithm so as to match the structure of a special image processing architecture optimally.

Intuitively, the problem of template decomposition is that for a given template **t**, one needs to find a sequence of smaller templates $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$ such that convolving an image **a** with **t** is equivalent to the sequence of convolutions $(\cdots((\mathbf{a} \circledcirc \mathbf{t}_1) \circledcirc \mathbf{t}_2) \cdots \circledcirc \mathbf{t}_{n-1}) \circledcirc \mathbf{t}_n$. In other words, **t** can be expressed in terms of the $\mathbf{t}_i$'s as $\mathbf{t} = \mathbf{t}_1 \circledcirc \cdots \circledcirc \mathbf{t}_n$.

One of the reasons for template decomposition is that some current image processors can only handle very small templates efficiently. For example, ERIM's Cytocomputer [17, 8], Martin Marietta's GAPP [2], and the National Bureau of Standard's PIPE [14] cannot deal with templates of size larger than 3×3 at each pipeline stage. Thus, a large template has to be decomposed into a sequence of 3×3 or smaller templates before it can be effectively processed.

A more important motivation for template decomposition is to speed up template operations. For large convolution masks, that is templates with large supports, the computation cost resulting from implementation can be prohibitive. However, in many instances, this cost can be significantly reduced by decomposing the templates into a sequence of smaller templates. For instance, the linear convolution of an image with a template whose support is an $n \times n$ square array at each point requires n^2 multiplications and $n^2 - 1$ additions in order to compute one new image pixel value; while the same convolution computed with a template whose support is a $1 \times n$ row, followed by a convolution with a template having support consisting of an $n \times 1$ column at each point, takes only $2n$ multiplications and $2(n-1)$ additions for each new pixel value (Example **4.10.4** illustrates this concept). This cost saving may still hold for mesh connected array processors, where the cost of a convolution is proportional to the size the template's support.

The problem of template decomposition has been investigated by several researchers. Ritter and Gader [16] presented some very efficient methods for decomposing FFT and general linear convolution templates. Wiejak and Buxton [20] proposed a method to decompose a 2-dimensional (or higher dimensional) Marr-Hildreth convolution operator into two one-dimensional convolutions. Zhuang and Haralick [22] gave an algorithm based on a tree search that can find an optimal decomposition of an arbitrary morphological structuring element if such a decomposition exists. Wilson and Manseur [11] provided optimal decompositions of 5×5 templates into 3×3 templates, while Wei and Lucas [21, 9] gave methods for decomposing templates on hexagonal arrays. In this section we discuss necessary and sufficient conditions for decomposing separable rectangular templates.

As mentioned in Section **5.1**, properties 3 through 7 of Eq. 5.1.1 provide the necessary tools for linear transform optimization. They can be used when exploring the possibilities of computing template operations in different ways. For example, if $(\mathbb{F}, +, \cdot)$ is a field, $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$, and we know that $\mathbf{t} = \mathbf{r} \bigoplus \mathbf{s}$, then by Property 4, we could apply **r** and **s** sequentially to **a** instead of computing $\mathbf{a} \bigoplus \mathbf{t}$ directly since, in general, **r** and **s** have much smaller support than **t**. In practice, however, one usually does not know **r** and **s** for a given **t**. A programmer may start with a large template **t** — i.e., a template with large support — and is faced with the task of finding two smaller templates **r** and **s** such that $\mathbf{t} = \mathbf{r} \bigoplus \mathbf{s}$. In many cases, the task of decomposing a template into smaller pieces may be very difficult if not impossible.

5.2.1 **Definition.** A *linear decomposition* (or $\bigoplus$*–decomposition*) of a template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is a sequence $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ such that $\mathbf{t} = \mathbf{t}_1 \bigoplus \cdots \bigoplus \mathbf{t}_n$.
A *weak linear decomposition* (or *weak* $\bigoplus$*–decomposition*) of **t** is a sequence of templates $\mathbf{t}_1, \ldots, \mathbf{t}_{k_1}, \ldots, \mathbf{t}_{k_{n-1}+1}, \ldots, \mathbf{t}_{k_n} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ such that

$$\mathbf{t} = (\mathbf{t}_1 \bigoplus \cdots \bigoplus \mathbf{t}_{k_1}) + (\mathbf{t}_{k_1+1} \bigoplus \cdots \bigoplus \mathbf{t}_{k_2}) + \cdots + \left(\mathbf{t}_{k_{n-1}+1} \bigoplus \cdots \bigoplus \mathbf{t}_{k_n}\right).$$

For example, if $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ has a weak decomposition $\mathbf{t} = \mathbf{t}_1 \bigoplus \mathbf{t}_2 + \mathbf{t}_3 \bigoplus \mathbf{t}_4$, then according to properties 3 and 4 of Eq. 5.1.1, $\mathbf{a} \bigoplus \mathbf{t}$ can be computed as

$$((\mathbf{a} \bigoplus \mathbf{t}_1) \bigoplus \mathbf{t}_2) + ((\mathbf{a} \bigoplus \mathbf{t}_3) \bigoplus \mathbf{t}_4).$$

Thus, when computing $\mathbf{a} \bigoplus \mathbf{t}$ one can compute the image $\mathbf{b} = (\mathbf{a} \bigoplus \mathbf{t}_1) \bigoplus \mathbf{t}_2$ and store it, then compute the image $\mathbf{c} = (\mathbf{a} \bigoplus \mathbf{t}_3) \bigoplus \mathbf{t}_4$, and finally add **b** and **c**. In general, a linear decomposition of **t** is preferred to a weak linear decomposition because usually more time and space is involved in computing with weak decompositions.

With the concept of template decomposition defined, we turn our attention to the decomposition of some commonly used templates. These templates belong to the class of shift-invariant templates with finite support.

In the subsequent discussion, we assume that $\mathbb{F}$ is a field, $\mathbf{X} = \mathbb{Z}^2$, and $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is a template with finite support at some point $\mathbf{y} \in \mathbf{X}$. Since $S(\mathbf{t}_{\mathbf{y}})$ is finite, the following are well defined:

$$\begin{aligned} i(\mathbf{y})_{min} &= inf\{p_1(\mathbf{x}) : \mathbf{x} \in S(\mathbf{t}_{\mathbf{y}})\}, \quad i(\mathbf{y})_{max} = sup\{p_1(\mathbf{x}) : \mathbf{x} \in S(\mathbf{t}_{\mathbf{y}})\}, \\ j(\mathbf{y})_{min} &= inf\{p_2(\mathbf{x}) : \mathbf{x} \in S(\mathbf{t}_{\mathbf{y}})\}, \quad j(\mathbf{y})_{max} = sup\{p_2(\mathbf{x}) : \mathbf{x} \in S(\mathbf{t}_{\mathbf{y}})\}. \end{aligned} \qquad (5.2.1)$$

Let $m(\mathbf{y}) = i(\mathbf{y})_{max} - i(\mathbf{y})_{min}$, $n(\mathbf{y}) = j(\mathbf{y})_{max} - j(\mathbf{y})_{min}$, and define

$$R(\mathbf{t}_{\mathbf{y}}) = \{(i(\mathbf{y})_{min} + i, j(\mathbf{y})_{min} + j) : 0 \le i \le m(\mathbf{y}), 0 \le j \le n(\mathbf{y}), i, j \in \mathbb{N}\}.$$

By definition, $R(\mathbf{t_y})$ is an $(m(\mathbf{y})+1) \times (n(\mathbf{y})+1)$ rectangular array and it is the *smallest* rectangular array containing $S(\mathbf{t_y})$. Figure **5.2.1** illustrates the concept of the smallest rectangular array containing the support of $\mathbf{t_y}$. As shown in this illustration, $\mathbf{y}$ need not be contained in $R(\mathbf{t_y})$.

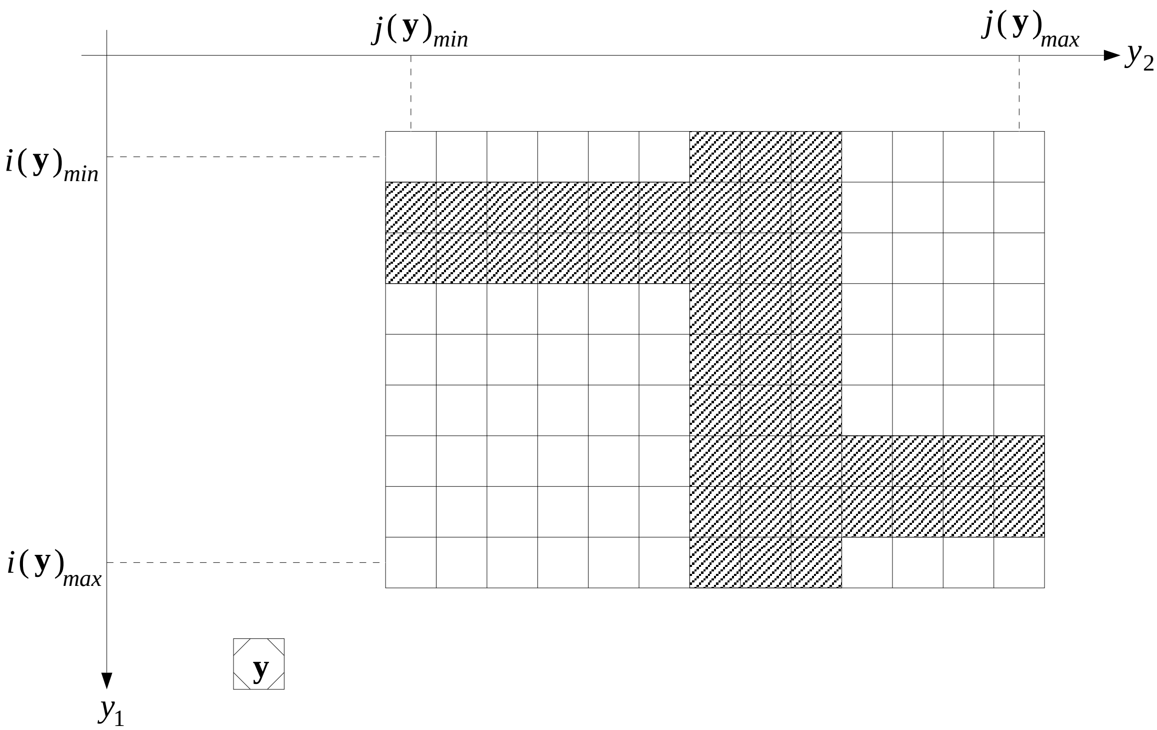


Figure 5.2.1 The rectangular array $R(\mathbf{t_y})$ containing $S(\mathbf{t_y})$. Here $S(\mathbf{t_y})$ is represented by the shaded region.

5.2.2 Definition. Let $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

1. A template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is said to have *finite support* if and only if $S(\mathbf{t_y})$ is finite $\forall\, \mathbf{y} \in \mathbf{X}$.
2. A template $\mathbf{t}$ with finite support is called a *rectangular* $m \times n$ *template*, or simply an $m \times n$ *template*, if $R(\mathbf{t_y})$ is of size $m \times n$ $\forall\, \mathbf{y} \in \mathbf{X}$.
3. An $m \times n$ template $\mathbf{t}$ is called *strictly rectangular* if and only if $S(\mathbf{t_y}) = R(\mathbf{t_y})$ $\forall\, \mathbf{y} \in \mathbf{X}$.
4. An $m \times n$ template $\mathbf{t}$ is called a *row template* whenever $m = 1$, and a *column template* whenever $n = 1$. If both $m = 1$ and $n = 1$, then $\mathbf{t}$ is called a *one-point template*.

Part 2 of the definition says that $m(\mathbf{y}) = m$ and $n(\mathbf{y}) = n$ $\forall\, \mathbf{y} \in \mathbf{X}$. In particular, it follows that if $\mathbf{t}$ is a shift-invariant template with finite support, then $\mathbf{t}$ is a rectangular $m \times n$ template for some pair of integers *m* and *n*. Thus **t** can be represented by an $m \times n$ matrix $T = (t_{ij})_{m \times n}$, where $t_{ij} = \mathbf{t}_{\mathbf{y}}(i(\mathbf{y})_{min} + (i-1), j(\mathbf{y})_{min} + (j-1))$, and $i(\mathbf{y})_{min}$ and $j(\mathbf{y})_{min}$ are as defined in Eq. 5.2.1. The matrix *T* is called the *weight matrix associated with* **t**. Note that if **t** is a strictly rectangular $m \times n$ template with weight matrix $T = (t_{ij})_{m \times n}$, then $t_{ij} \neq 0$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

The weight matrix associated with **t** should not be confused with the matrix $\psi(\mathbf{t})$ defined earlier. In fact, two distinct templates can have identical weight matrices. For example, the matrix

$$T = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 1 & 0 \end{pmatrix}$$

is a weight matrix associate with the two shift-invariant templates **s** and **t** shown in Figure **5.2.2**.

Figure 5.2.2 Two templates having identical weight matrix *T*.

It is clear that by choosing the smallest rectangular array of type $\mathbb{Z}_{\pm m} \times \mathbb{Z}_{\pm n} = \{(i,j) \in \mathbb{Z}^2 \,:\, -m \leq i \leq m,\ -n \leq j \leq n\}$ with the property that $S(\mathbf{t}_{(0,0)}) \subset \mathbb{Z}_{\pm m} \times \mathbb{Z}_{\pm n}$, then the matrix

$$T = \begin{pmatrix} t_{-m,-n} & \cdots & t_{-m,0} & \cdots & t_{-m,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{0,-n} & \cdots & t_{0,0} & \cdots & t_{0,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{m,-n} & \cdots & t_{m,0} & \cdots & t_{m,n} \end{pmatrix}, \tag{5.2.2}$$

defined by $t_{ij} = \mathbf{t}_{(0,0)}(i,j)$ for $-m \leq i \leq m$ and $-n \leq j \leq n$, provides for a one-to-one correspondence between shift–invariant templates and matrices defined this way. For example, the templates **s** and **t** defined in Figure **5.2.2** have template representations

$$S = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & 4 \\ 0 & 0 & 2 & 1 & 0 \end{pmatrix} \text{ and } T = \begin{pmatrix} 0 & 0 & 1 & 3 & 4 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

respectively. The matrix *T* defined by Eq. 5.2.2 is called the *centered* weight matrix corresponding to **t**. In contrast to weight matrices, centered weight matrices provide for the location of the target pixel, which is always located at the center of the matrix. For this reason the target pixel of a template is often referred to as the *center* pixel.

Among shift-invariant templates, rectangular templates are the simplest and most commonly used templates in image processing. A first step in various template decomposition schemes of such templates is to decompose the template into the product of a row and a column template. In light of this, it is important to note that although every template whose corresponding centered weight matrix is a row or column matrix *is* a row or column template, respectively, the converse need not necessarily be true. The template shown in Figure **5.2.3** is a row template but its corresponding centered weight matrix is a 3×3 matrix.

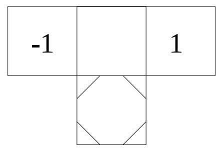


Figure 5.2.3 Example of a row template.

5.2.3 **Definition.** A template is called *linearly separable*, or simply *separable*, if it can be expressed as linear decomposition of a row template and a column template.

Ideally, to speed up the template operations, we would like to decompose an arbitrary rectangular template into two one–dimensional templates, namely a row template and a column template. Thus, for a given strictly rectangular $m \times n$ template, the number of arithmetic operations required for each template operation at each pixel location can be reduced from $O(mn)$ to $O(m+n)$. The next theorem provides necessary and sufficient conditions for the separability of strictly rectangular templates.

5.2.4 **Theorem.** *If* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *is a strictly rectangular* $m \times n$ *template with associated weight matrix* $T = (t_{ij})_{m \times n}$*, then* $\mathbf{t}$ *is linearly separable if and only if*

$$\frac{t_{ij}}{t_{1j}} = \frac{t_{i1}}{t_{11}} \tag{I}$$

for all $1 \leq i \leq m$ *and* $1 \leq j \leq n$.

Proof: Suppose that $\mathbf{t}$ is separable and that $\mathbf{t} = \mathbf{r} \bigoplus \mathbf{s}$, where $\mathbf{r}$ is a $1 \times n$ row template with weights $r_1, \ldots, r_n$ and $\mathbf{s}$ is a $m \times 1$ column template with weights $s_1, \ldots, s_m$. By definition of template products, $t_{ij} = s_i \cdot r_j$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Therefore,

$$\frac{t_{ij}}{t_{1j}} = \frac{s_i \cdot r_j}{s_1 \cdot r_j} = \frac{s_i}{s_1} = \frac{s_i \cdot r_1}{s_1 \cdot r_1} = \frac{t_{i1}}{t_{11}}.$$

To prove sufficiency, suppose that Eq. (I) holds. Define a $1 \times n$ row template $\mathbf{r}$ and a $m \times 1$ column template $\mathbf{s}$ by

$$r_j = t_{1j} \quad \text{for } j = 1, \ldots, n \tag{i}$$

and

$$s_i = \frac{t_{i1}}{t_{11}} \quad \text{for } i = 1, \ldots, m\,, \tag{ii}$$

respectively. Thus, $t_{ij} = s_i \cdot r_j$ and, therefore, $\mathbf{t} = \mathbf{r} \bigoplus \mathbf{s}$.

Q.E.D.

Note that condition (I) is equivalent to saying that the rank of *T* is 1. The theorem provides a straight forward method for testing and decomposing linearly separable templates. Since Eq. (I) is equivalent to $t_{ij} \cdot t_{11} = t_{i1} \cdot t_{1j}$, it takes $2n(m-1)$ multiplications and $(m-1)(n-1)$ comparisons in order to check whether a rectangular template of size $m \times n$ is separable or not. If it is separable, then one can easily construct the corresponding row and column templates by using Eqs. (i) and (ii).

In many cases it is convenient to define large invariant templates by a *weight* function *w* of two variables. For example, the function

$$w(x, y) = x^2y - 2x^2 + 2xy - 4x - y + 2\,, \tag{5.2.5}$$

when evaluated at the integers $x = -1, 0, 1$ and $y = -1, 0, 1$, provides the weights of the 3×3 template **t** shown in Figure **5.2.4**. Note that if $T = (t_{ij})$ denotes the centered weight matrix corresponding to **t**, then $t_{ij} = w(i, j)$. In practice, weight functions are commonly restricted to rectangular domains centered at the origin.

$$\mathbf{t} = \begin{array}{|c|c|c|} \hline 6 & 4 & 2 \\ \hline 3 & 2 & 1 \\ \hline -6 & -4 & -2 \\ \hline \end{array}$$

Figure 5.2.4 The template **t** defined by $w(x, y) = x^2y - 2x^2 + 2xy - 4x - y + 2$.

The separability of a shift-invariant template can be reduced to the separability of its associated weight function. The following result is obvious.

5.2.5 **Theorem.** *Suppose* $\mathbf{X} = \mathbb{Z}^2$ *and* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *is an invariant template with finite support and weight function w. Then* $\mathbf{t}$ *is linearly separable if and only if* $w(x, y) = f(x) \cdot g(y)$ *for some* $\mathbb{F}$*–valued functions f and g.*

Note that this theorem is similar to Theorem **5.2.4** with the weight functions *f* and *g* specifying the row and column templates.

5.2.6 **Examples:**

(i) The template $\mathbf{t}$ with weight function w defined by Eq. 5.2.5 is separable since

$$w(x,y) = \left(x^2 + 2x - 1\right)(y-2) = f(x) \cdot g(y)\,,$$

where $f(x) = x^2 + 2x - 1$ and $g(y) = y - 2$. Thus, $\mathbf{t} = \mathbf{r} \bigoplus \mathbf{s}$, where the corresponding row and column templates are given by

$$\mathbf{r} = \begin{array}{|c|} \hline -2 \\ \hline -1 \\ \hline 2 \\ \hline \end{array} \qquad \text{and} \qquad \mathbf{s} = \begin{array}{|c|c|c|} \hline -3 & -2 & -1 \\ \hline \end{array}$$

(ii) Consider the $(2n+1) \times (2n+1)$ template $\mathbf{t}$ defined by

$$\begin{aligned}
w(x,y) &= \frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)e^{-(x^2+y^2)/2\sigma^2} \\
&= \frac{1}{2\pi\sigma^4}\left(1 - \frac{x^2}{\sigma^2}\right)e^{-x^2/2\sigma^2} \cdot e^{-y^2/2\sigma^2} + \frac{1}{2\pi\sigma^4}e^{-x^2/2\sigma^2}\left(1 - \frac{y^2}{\sigma^2}\right)e^{-y^2/2\sigma^2} \\
&= f_1(x)f_2(y) + f_3(x)f_4(y)\,,
\end{aligned}$$

where $-n \leq x, y \leq n$. This template is also known as a *Marr-Hildreth* template.

Clearly $\mathbf{t}$ is not separable but it still has a very efficient weak linear decomposition which is the sum of two separable templates, namely

$$\mathbf{t} = (\mathbf{t}_1 \bigoplus \mathbf{t}_2) + (\mathbf{t}_3 \bigoplus \mathbf{t}_4)\,,$$

where $\mathbf{t}_i$ is defined by f_i for $i = 1, 2, 3, 4$.

5.3 LU Decomposition of Templates

The *rank* of a template $\mathbf{t}$ is defined as the rank of its corresponding weight matrix *T*. It is easy to show that every $n \times n$ matrix *T* can be decomposed as a sum of at most *n* rank 1 matrices. Thus, if $\mathbf{t}$ is an $n \times n$ template, then $\mathbf{t}$ can be expressed as

$$\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_n\,, \tag{5.3.1}$$

where $\mathbf{t}_i$ is a rank 1 template for each $i = 1, 2, \ldots, n$. According to Theorem **5.2.4**, each $\mathbf{t}_i$ can in turn be decomposed into a product of a row and column template, namely $\mathbf{t}_i = \mathbf{r}_i \bigoplus \mathbf{s}_i$. Therefore,

$$\mathbf{t} = (\mathbf{r}_1 \bigoplus \mathbf{s}_1) + (\mathbf{r}_2 \bigoplus \mathbf{s}_2) + \cdots + (\mathbf{r}_n \bigoplus \mathbf{s}_n)\,. \tag{5.3.2}$$

This shows that every invariant template with finite support has an effective weak linear decomposition. However, the computational cost of replacing the convolution $\mathbf{a} \bigoplus \mathbf{t}$ by the $2n$ convolutions $(\mathbf{a} \bigoplus \mathbf{r}_i) \bigoplus \mathbf{s}_i$ is in most cases still prohibitive. It is therefore desirable to reduce the number of terms appearing on the right-hand side of Eq. 5.3.2. One effective method for accomplishing this is the LU factorization technique of standard numerical linear algebra.

For the remainder of this section *L* will always denote a lower triangular matrix and *U* an upper triangular matrix. A matrix *T* is said to have an *LU factorization* whenever *T* can be written as a matrix product $T = LU$.

The next theorem provides a first step in the reduction of the number of terms in Eq. 5.3.2.

5.3.1 **Theorem.** *If* $T = (t_{ij})_{n \times n}$ *has an LU factorization, then for any* $m = 1, 2, \ldots, n$ *there exist matrices* $T_1, \ldots, T_m$ *each of rank at most 1 and a matrix* $E_m = (e_{ij})_{n \times n}$ *with* $e_{ik} = e_{kj} = 0$ *for* $k = 1, 2, \ldots, m$ *such that*

$$T = T_1 + \cdots T_m + E_m .$$

Proof: Suppose $T = LU$. For $i = 1, \ldots, n$ define U_i to be the $n \times n$ matrix whose *i*th row is the *i*th row of *U* and whose remaining rows have zero entries. For $m = n$ let V_m denote the $n \times n$ zero matrix and for $m < n$ set

$$V_m = U_{m+1} + \cdots + U_n .$$

Then

$$\begin{aligned} T = LU &= L(U_1 + \cdots + U_m + V_m) \\ &= LU_1 + \cdots + LU_m + LV_m \\ &= T_1 + \cdots + T_m + E_m , \end{aligned}$$

where $T_i = LU_i$ and $E_m = LV_m$.

Q.E.D.

Thus, if the weight matrix *T* corresponding to the $n \times n$ template **t** has an LU factorization, then for any $m = 1, \ldots, n$, we can write **t** as the sum

$$\begin{aligned} \mathbf{t} &= \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_m + \mathbf{e}_m \\ &= (\mathbf{r}_1 \bigoplus \mathbf{s}_1) + \cdots + (\mathbf{r}_m \bigoplus \mathbf{s}_m) + \mathbf{e}_m , \end{aligned} \tag{5.3.3}$$

where $\mathbf{e}_m$ is at most of size $(n - m) \times (n - m)$. In particular, if $m = 1$, then

$$\mathbf{t} = (\mathbf{r} \bigoplus \mathbf{s}) + \mathbf{e} , \tag{5.3.4}$$

where **e** is of size $(n - 1) \times (n - 1)$; i.e., any template whose weight matrix has an LU decomposition can be decomposed into a product of a row and a column template plus a template of smaller size.

5.3.2 **Definition.** A template is said to have an *LU decomposition* if its corresponding weight matrix has a LU decomposition.

Using centered weight matrices in the LU decomposition of a template **t** simplifies the task of locating the support of the template $\mathbf{e}_m$ in relation to the location of the target pixel $\mathbf{y} = (y_1, y_2)$ since

$$S(\mathbf{e}_{m\mathbf{y}}) \subset \left\{ (x_1, x_2) \in \mathbb{Z}^2 \,:\, y_1 \le x_1 \le y_1 + (n - m),\ y_2 \le x_2 \le y_2 + (n - m) \right\} ;$$

i.e, $\mathbf{e}_m$ has form

$$\mathbf{e}_m \;=\; \begin{array}{|c|c|c|} \hline e_{1,1} & \cdots & e_{1,n\text{-}m} \\ \hline \vdots & \cdots & \vdots \\ \hline e_{n\text{-}m,1} & \cdots & e_{n\text{-}m,n\text{-}m} \\ \hline \end{array}$$

where $e_{ij} = \mathbf{e}_{m\mathbf{y}}(y_1 + (i-1), y_2 + (j-1))$ for $i, j = 1, 2, \ldots, n - m + 1$.

This observation implies that Eqs. 5.3.3 and 5.3.4 are not as good as they seem at first glance; even for templates with centered target pixels, the target pixel of $\mathbf{e}_m$ will always be in the upper right hand corner and thus requiring, in general, a shift of data in order to map the template operation to a particular architecture or to achieve optimal computation speed.

5.3.3 Example: Consider the 5×5 template

t =

1	1	2	1	1
2	3	5	3	2
3	4	7	4	3
2	3	5	3	2
1	1	2	1	1

The corresponding centered (as well as non-centered) weight matrix is given by

$$T = \begin{pmatrix} 1 & 1 & 2 & 1 & 1 \\ 2 & 3 & 5 & 3 & 2 \\ 3 & 4 & 7 & 4 & 3 \\ 2 & 3 & 5 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 \end{pmatrix},$$

and an LU decomposition of T is given by

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

For $m = 2$,

$$T_1 = LU_1 = \begin{pmatrix} 1 & 1 & 2 & 1 & 1 \\ 2 & 2 & 4 & 2 & 2 \\ 3 & 3 & 6 & 3 & 3 \\ 2 & 2 & 4 & 2 & 2 \\ 1 & 1 & 2 & 1 & 1 \end{pmatrix}, \quad T_2 = LU_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and $E_2 = (0)_{5\times 5}$.
Thus,

$$\mathbf{t} = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 2 & 1 & 1 \\ \hline 2 & 2 & 4 & 2 & 2 \\ \hline 3 & 3 & 6 & 3 & 3 \\ \hline 2 & 2 & 4 & 2 & 2 \\ \hline 1 & 1 & 2 & 1 & 1 \\ \hline \end{array} \quad + \quad \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Since each of the templates $\mathbf{t}_1$ and $\mathbf{t}_2$ is of rank 1, $\mathbf{t}$ can be decomposed further as

$$\mathbf{t} = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 3 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 2 & 1 & 1 \\ \hline \end{array} + \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline \end{array}$$

Although **t** is now a sum of products of row and column templates, it is not a sum of products of 3×3 or smaller templates which are better suited for the types of architectures mentioned in Section **5.2**. However, it is not difficult to show that

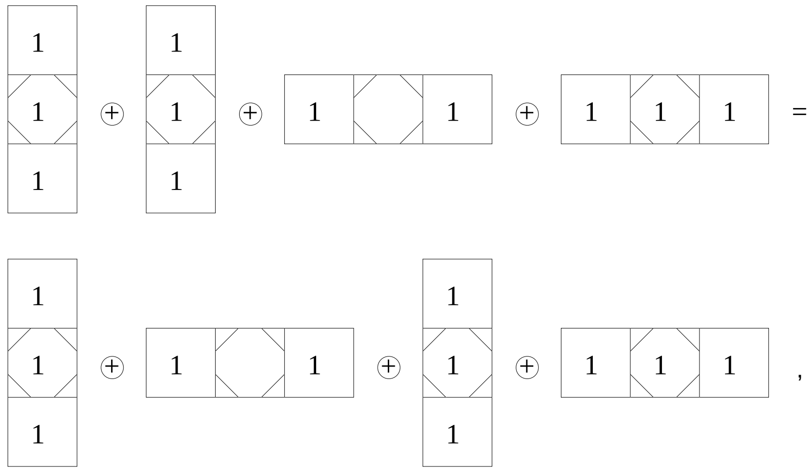
$$\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 3 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} = \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array} \oplus \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array}$$

and

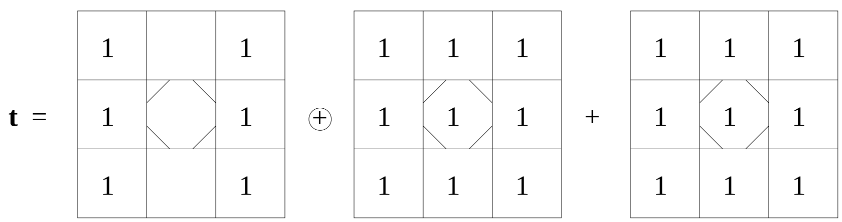
$$\begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 2 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 1 & & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline \end{array}$$

Thus, **t** can be written as a weak decomposition of 3×3 or smaller templates.

There are two noteworthy observations regarding Example **5.3.3**. First, since



the template **t** can be expressed as



Therefore, $\mathbf{t} = \mathbf{s} \bigoplus \mathbf{t}_2 + \mathbf{t}_2$, where $\mathbf{s}$ and $\mathbf{t}_2$ are 3×3 templates and $\mathbf{t}_2$ is as in the above example. In particular, the 5×5 rank 1 template $\mathbf{t}_1$ in Example **5.3.3** is now expressed as a product of two 3×3 templates; namely, $\mathbf{t}_1 = \mathbf{s} \bigoplus \mathbf{t}_2$. O'Leary showed that this holds in general [14].

5.3.4 **Theorem**(O'Leary)**.** *If* $\mathbf{t}$ *is a* 5×5 *template such that either* $\mathbf{t}$ *has rank 1 or* $\mathbf{t}$ *is a diagonal template, then* $\mathbf{t} = \mathbf{r} \bigoplus \mathbf{s}$, *where* $\mathbf{r}$ *and* $\mathbf{s}$ *are* 3×3 *templates.*

As a second observation we note that the template **t** in Example **5.3.3** does not satisfy condition (I) of Theorem **5.2.4** since, for example, $t_{33}/t_{13} \neq t_{31}/t_{11}$. In fact, **t** is a rank 2 template and it is for this reason that $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$, where $\mathbf{t}_1$ and $\mathbf{t}_1$ are rank 1 templates. The relationship between the number of terms in the expansion of **t** and the rank of **t** is given by the following corollary of Theorem **5.3.1**:

5.3.5 **Theorem.** *If T is an $n \times n$ matrix of rank r and T has an LU factorization, then for any* $m = 1, 2, \ldots, r$,

$$T = T_1 + \cdots + T_m + E_m \,,$$

where each T_i is an $n \times n$ rank 1 matrix, $E_m = (e_{ij})_{n \times n}$ has the property that $e_{ik} = e_{kj} = 0$ for $k = 1, 2, \ldots, m$, and E_m is the zero matrix whenever $m = r$.

5.3.6 **Example:** Let $\mathbf{t}$ be the rank 3 template defined by

$\mathbf{t}$ =

3	3	3	3	3
	2	2	2	
		1		
	-2	-2	-2	
-3	-3	-3	-3	-3

An LU decomposition for the corresponding weight matrix T is given by

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

For $m = 2$ we have $T = T_1 + T_2 + E_2$, where

$$T_1 = \begin{pmatrix} 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -3 & -3 & -3 & -3 & -3 \end{pmatrix}, \qquad T_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -2 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$E_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Thus,

$\mathbf{t}$ =

3	3	3	3	3
-3	-3	-3	-3	-3

+

2	2	2
-2	-2	-2

+

1

or $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \mathbf{e}_2$. Since $\mathbf{t}_1$ is a 5×5 rank 1 template, $\mathbf{t}_1$ can be further decomposed into $\mathbf{t}_1 = \mathbf{r} \bigoplus \mathbf{s}$, where $\mathbf{r}$ and $\mathbf{s}$ are 3×3 templates. Therefore, the convolution $\mathbf{a} \bigoplus \mathbf{t}$ can be accomplished using only 3×3 templates, namely

$$\mathbf{a} \bigoplus \mathbf{t} = 3(\mathbf{a} \bigoplus \mathbf{r}_1) \bigoplus \mathbf{s}_1 + 2(\mathbf{a} \bigoplus \mathbf{r}_2) + \mathbf{a}\,,$$

where

$$\mathbf{r}_1 = \begin{array}{|c|c|c|} \hline 1 & a & 1 \\ \hline & & \\ \hline -1 & -a & -1 \\ \hline \end{array}\,, \quad \mathbf{s}_1 = \begin{array}{|c|c|c|} \hline 1 & b & 1 \\ \hline & & \\ \hline 1 & b & 1 \\ \hline \end{array}\,, \quad \mathbf{r}_2 = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline & & \\ \hline -1 & -1 & -1 \\ \hline \end{array}\,,$$

$a = \frac{1+\sqrt{5}}{2}$, and $b = \frac{1-\sqrt{5}}{2}$. The templates $\mathbf{r}_1$ and $\mathbf{s}_1$ where obtained by first decomposing the template $\mathbf{r} = \frac{1}{3}\mathbf{t}_1$, whose pictorial representation is given by

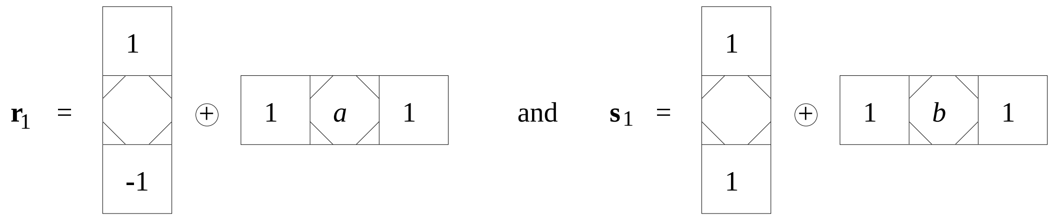
$$\mathbf{r} = \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline \end{array}$$

into the column and row templates

$$\begin{array}{|c|} \hline 1 \\ \hline \\ \hline \\ \hline \\ \hline -1 \\ \hline \end{array} \bigoplus \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} =$$

$$\begin{array}{|c|} \hline 1 \\ \hline \\ \hline -1 \\ \hline \end{array} \bigoplus \begin{array}{|c|} \hline 1 \\ \hline \\ \hline 1 \\ \hline \end{array} \bigoplus \begin{array}{|c|c|c|} \hline 1 & a & 1 \\ \hline \end{array} \bigoplus \begin{array}{|c|c|c|} \hline 1 & b & 1 \\ \hline \end{array}$$

and then recombining these templates in the order



A prime assumption in the LU decomposition of a template is that the weight matrix of the template has an LU decomposition. But given the weight matrix *T* of a template, how do we know whether or not *T* has an LU decomposition? A well known fact of matrix algebra is that every matrix *T* can be written as $TP = LU$ for some permutation matrix *P*. However, since a permutation matrix moves the template weights to different geometric locations, thus resulting in a different template, this fact is, in general, not directly applicable to the problem of template decomposition. Necessary conditions for the existence of an LU decomposition are given by the following theorem:

5.3.7 **Theorem.** *Let $T = (t_{ij})_{m \times n}$ be of rank r and let T_k denote the leading $k \times k$ principal submatrix of T. If T_k is nonsingular for $k = 1, 2, \ldots, r$, then there exists a lower triangular matrix L and an upper triangular matrix U such that $T = LU$.*

For a proof see [13] or [5].

5.4 Polynomial Factorization of Templates

Every template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ with centered weight matrix $T = (t_{ij})$ as defined by Eq. 5.2.2 can be represented by a polynomial $t(x, y)$ in two variables, where

$$t(x, y) = \sum_{i=0}^{2m} \sum_{j=0}^{2n} a_{ij} x^i y^j \,, \tag{5.4.1}$$

and $a_{ij} = t_{i-m, j-n}$. The coefficients of $t(x, y)$ define a matrix $A = (a_{ij})$ which corresponds to the matrix *T* shifted to the first quadrant.

The polynomial $t(x, y)$ defined by Eq. 5.4.1 is called the *weight polynomial of* **t**. The weight polynomial should not be confused with the weight function defined in Section **5.2**. However, just as for weight functions, the factorization of weight polynomials is equivalent to the decomposition of the corresponding templates. Since polynomial factorization results in polynomial factors of lower degree, the corresponding template decomposition results in template factors of smaller size.

The following polynomial decomposition theorem was proved by Manseur and Wilson [11].

5.4.1 **Theorem.** *If* $m \geq 3$, $n \geq 3$, *and* $t(x,y) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} x^i y^j$, *then there exists five polynomials* *u, v, p, q, and r such that*

$$t(x,y) = u(x,y) \cdot v(x,y) + p(x,y) \cdot q(x,y) + r(x,y)\,,$$

where

$$u(x,y) = \sum_{i=0}^{2} \sum_{j=0}^{2} u_{ij} x^i y^j\,, \qquad v(x,y) = \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} v_{ij} x^i y^j\,,$$
$$p(x,y) = \sum_{i=0}^{2} \sum_{j=0}^{2} p_{ij} x^i y^j\,, \qquad q(x,y) = \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} q_{ij} x^i y^j\,,$$

and

$$r(x,y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} r_{ij} x^i y^j\,.$$

Proof: If $a(y) = \sum_{j=0}^{n} a_{0j} y^j$, then by the Fundamental Theorem of Algebra, there exists polynomials a_1 and a_2 with $deg(a_1) \leq 2$, $deg(a_2) \leq n-2$, and such that $a(y) = a_1(y) \cdot a_2(y)$. Similarly, if $b(y) = \sum_{j=0}^{n} a_{mj} y^j$, then there exists polynomials b_1 and b_2 with $deg(b_1) \leq 2$, $deg(b_2) \leq n-2$, and such that $b(y) = b_1(y) \cdot b_2(y)$.

Define

$$u(x,y) = a_1(y) + x^2 b_1(y)\,,$$
$$v(x,y) = a_2(y) + x^{m-2} b_2(y)\,,$$

and

$$s(x,y) = t(x,y) - u(x,y) \cdot v(x,y)\,. \tag{i}$$

Multiplying *u* and *v*, we obtain

$$\begin{aligned} s(x,y) &= t(x,y) - \left[\left(a_1(y) + x^2 b_1(y)\right) \cdot \left(a_2(y) + x^{m-2} b_2(y)\right)\right] \\ &= t(x,y) - \left[a(y) + x^2 b_1(y) a_2(y) + x^{m-2} a_1(y) b_2(y) + x^m b(y)\right]. \end{aligned}$$

But

$$\begin{aligned} t(x,y) &= \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} x^i y^j \\ &= \sum_{i=1}^{m-1} \sum_{j=0}^{n} a_{ij} x^i y^j + \sum_{j=0}^{n} a_{0j} y^j + x^m \sum_{j=0}^{n} a_{mj} y^j \\ &= \sum_{i=1}^{m-1} \sum_{j=0}^{n} a_{ij} x^i y^j + a(y) + x^m b(y)\,. \end{aligned}$$

Thus,

$$s(x,y) = \sum_{i=1}^{m-1} \sum_{j=0}^{n} a_{ij} x^i y^j - x^2 b_1(y) a_2(y) - x^{m-2} a_1(y) b_2(y)\,,$$

which means that $s(x,y)$ is of form

$$s(x,y) = \sum_{i=1}^{m-1} \sum_{j=0}^{n} s_{ij} x^i y^j .$$

If $c(x) = \sum_{i=0}^{m-1} s_{i0} x^i$, then c can be decomposed as $c(x) = x \cdot c_1(x)$, where $deg(c_1) \leq m-2$. Similarly, if $d(x) = \sum_{i=0}^{m-1} s_{in} x^i$, then d can be decomposed as $d(x) = x \cdot d_1(x)$, where $deg(d_1) \leq m-2$. Let $p(x,y) = x + xy^2$, $q(x,y) = c_1(x) + d_1(x) y^{n-2}$, and

$$r(x,y) = s(x,y) - p(x,y) \cdot q(x,y) . \tag{ii}$$

Again, by multiplying $p(x,y)$ by $q(x,y)$ and noting that $s(x,y)$ can be written as

$$s(x,y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} s_{ij} x^i y^j + \sum_{i=1}^{m-1} s_{i0} x^i + y^n \sum_{i=1}^{m-1} s_{in} x^i ,$$

it can be ascertained that $r(x,y)$ is of form

$$r(x,y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} r_{ij} x^i y^j .$$

It now follows from Eqs. (i) and (ii) that

$$\begin{aligned} t(x,y) &= u(x,y) \cdot v(x,y) + s(x,y) \\ &= u(x,y) \cdot v(x,y) + p(x,y) \cdot q(x,y) + r(x,y) . \end{aligned}$$

Q.E.D.

Let **t** be an arbitrary 5×5 template. According to Eq. 5.3.1, **t** can be decomposed into a sum of five rank 1 templates

$$\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3 + \mathbf{t}_4 + \mathbf{t}_5 .$$

It follows from O'Leary's theorem (Theorem **5.3.4**) that for $i = 1, 2, \ldots, 5$, each template $\mathbf{t}_i$ can in turn be decomposed into a product of two 3×3 templates $\mathbf{t}_i = \mathbf{r}_i \bigoplus \mathbf{s}_i$. Thus, **t** can be written as a weak decomposition of 3×3 templates

$$\mathbf{t} = (\mathbf{r}_1 \bigoplus \mathbf{s}_1) + \cdots + (\mathbf{r}_5 \bigoplus \mathbf{s}_5) .$$

Obviously, Theorem **5.4.1** provides for a better result; the number of 3×3 templates needed is actually half of the number indicated by the rank method, namely

$$\mathbf{t} = \mathbf{u} \bigoplus \mathbf{v} + \mathbf{p} \bigoplus \mathbf{q} + \mathbf{r} , \tag{5.4.3}$$

where **u**, **v**, **p**, **q**, and **r** are of size at most 3×3. Example **5.6.18**(iv) shows that this result is the best achievable for certain 5×5 templates.

The methods of Theorem **5.4.1** can be applied several times in order to decompose templates of size larger than 5×5 into sums and products of 3×3 templates. For example, if **t** is a 7×7 template, then **t** can be decomposed as

$$\mathbf{t} = (\mathbf{r}_1 \oplus \mathbf{s}_1 \oplus \mathbf{u}_1) + (\mathbf{r}_2 \oplus \mathbf{s}_2 \oplus \mathbf{u}_2) + (\mathbf{r}_3 \oplus \mathbf{s}_3) + (\mathbf{r}_4 \oplus \mathbf{s}_4) + \mathbf{r}_5\,,$$

where each $\mathbf{r}_i$, $\mathbf{s}_i$, and $\mathbf{u}_i$ is a 3×3 template. By factoring the polynomials $a(y)$, $b(y)$, $c(x)$ and $d(x)$ in the proof of Theorem **5.4.1** into the product of quadratics, a general $(2n+1) \times (2n+1)$ template **t** can always be decomposed into a sum

$$\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_{2n-1}\,,$$

where $\mathbf{t}_{2n-1}$ is a 3×3 template and, for $i = 1, 2, \ldots, n-1$, the templates $\mathbf{t}_{2i-1}$ and $\mathbf{t}_{2i}$ are products of form

$$\mathbf{t}_{2i-1} = \mathbf{r}_{i_1} \oplus \mathbf{r}_{i_2} \oplus \cdots \oplus \mathbf{r}_{i_{(n+1-i)}} \quad \text{and} \quad \mathbf{t}_{2i} = \mathbf{s}_{i_1} \oplus \mathbf{s}_{i_2} \oplus \cdots \oplus \mathbf{s}_{i_{(n+1-i)}}\,,$$

where each $\mathbf{r}_{i_j}$ and $\mathbf{s}_{i_j}$ is a 3×3 template. In actual applications, further savings can be achieved by noting that in the proof of Theorem **5.4.1**, the polynomial $p(x, y) = x + xy^2$ appears in every decomposition. Thus, the distributive law can be used to further reduce the operation count in templates of size 7×7 or larger.

5.5 The Manseur-Wilson Theorem

The decomposition methods presented in the next section are based on a remarkable theorem first proven by Z. Manseur and D. Wilson [11]. While the statement of this theorem is somewhat lengthy, the key idea (as noted by Manseur and Wilson) is as follows: If the roots of the four *boundary polynomials* (Eq. (I) of Theorem **5.5.1**) of a template can be rearranged so that they match, then the template can be decomposed into a sum and product of three smaller templates.

5.5.1 **Theorem** (Manseur and Wilson). *Suppose* $t(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} x^i y^j$ *is a polynomial such that* a_{00}, a_{0n}, a_{m0}, *and* a_{mn} *are all nonzero. Define*

$$\begin{aligned} a(y) &= \sum_{j=0}^{n} a_{0j} y^j\,, \quad b(y) = \sum_{j=0}^{n} a_{mj} y^j\,, \\ c(x) &= \sum_{i=0}^{m} a_{in} x^i\,, \quad d(x) = \sum_{i=0}^{m} a_{i0} x^i\,, \end{aligned} \tag{I}$$

and let $\alpha_1, \ldots, \alpha_n$ *denote the roots of* $a(y)$, $\beta_1, \ldots, \beta_n$ *denote the roots of* $b(y)$, $\gamma_1, \ldots, \gamma_m$ *denote the roots of* $c(x)$, *and let* $\delta_1, \ldots, \delta_m$ *denote the roots of* $d(x)$.

If there exists an ordering of the roots such that

$$\alpha_1 \alpha_2 \delta_1 \delta_2 = \beta_1 \beta_2 \gamma_1 \gamma_2 \quad and \quad \alpha_3 \cdots \alpha_n \delta_3 \cdots \delta_m = \beta_3 \cdots \beta_n \gamma_3 \cdots \gamma_m \tag{II}$$

or such that

$$\alpha_1\alpha_2\gamma_1\gamma_2 = \beta_1\beta_2\delta_1\delta_2 \quad \textit{and} \quad \alpha_3 \cdots \alpha_n\gamma_3 \cdots \gamma_m = \beta_3 \cdots \beta_n\delta_3 \cdots \delta_m\,, \tag{III}$$

then there exists polynomials $p(x, y)$, $q(x, y)$, *and* $r(x, y)$ *such that*

$$t(x, y) = p(x, y) \cdot q(x, y) + r(x, y)\,,$$

where

$$p(x, y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^i y^j\,, \qquad q(x, y) = \sum_{i=0}^{m-2}\sum_{j=0}^{n-2} q_{ij}x^i y^j\,,$$

and

$$r(x, y) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} r_{ij}x^i y^j\,.$$

Conversely, if $t(x, y) = p(x, y) \cdot q(x, y) + r(x, y)$, *where p, q, and r are defined as above, then there exists an ordering of the roots as described in Eqs.* (II) *and* (III).

Proof: Factor $a(y)$, $b(y)$, $c(x)$, and $d(x)$ as follows:

$$\begin{aligned} a(y) &= a_{0n}[y^n + \cdots + (a_{01}/a_{0n})y + (a_{00}/a_{0n})] \\ &= a_{0n}\left[y^2 - (\alpha_1 + \alpha_2)y + \alpha_1\alpha_2\right]\left[y^{n-2} + \cdots + (-1)^n\alpha_3 \cdots \alpha_n\right], \end{aligned} \tag{i}$$

$$\begin{aligned} b(y) &= a_{mn}[y^n + \cdots + (a_{m1}/a_{mn})y + (a_{m0}/a_{mn})] \\ &= \left[y^2 - (\beta_1 + \beta_2)y + \beta_1\beta_2\right]\left[a_{mn}y^{n-2} + \cdots + (-1)^n a_{mn}\beta_3 \cdots \beta_n\right], \end{aligned} \tag{ii}$$

$$\begin{aligned} c(x) &= a_{mn}[x^m + \cdots + (a_{1n}/a_{mn})x + (a_{0n}/a_{mn})] \\ &= \left[x^2 - (\gamma_1 + \gamma_2)x + \gamma_1\gamma_2\right]\left[a_{mn}x^{m-2} + \cdots + (-1)^m a_{mn}\gamma_3 \cdots \gamma_m\right], \end{aligned} \tag{iii}$$

and

$$\begin{aligned} d(x) &= a_{m0}[x^m + \cdots + (a_{10}/a_{m0})x + (a_{00}/a_{m0})] \\ &= a_{m0}\left[x^2 - (\delta_1 + \delta_2)x + \delta_1\delta_2\right]\left[x^{m-2} + \cdots + (-1)^m\delta_3 \cdots \delta_m\right]. \end{aligned} \tag{iv}$$

By substituting appropriate roots in the above equations, one obtains the relations

$$a_{00}/a_{0n} = (-1)^n\alpha_1\alpha_2 \cdots \alpha_n\,, \tag{v}$$

$$a_{m0}/a_{mn} = (-1)^n\beta_1\beta_2 \cdots \beta_n\,, \tag{vi}$$

$$a_{0n}/a_{mn} = (-1)^m\gamma_1\gamma_2 \cdots \gamma_m\,, \tag{vii}$$

and

$$a_{00}/a_{m0} = (-1)^m\delta_1\delta_2 \cdots \delta_m\,. \tag{viii}$$

Now let

$$a_1(y) = \gamma_1\gamma_2 y^2 - \gamma_1\gamma_2(\alpha_1 + \alpha_2)y + \gamma_1\gamma_2\alpha_1\alpha_2$$

and

$$a_2(y) = (-1)^m a_{mn}\gamma_3 \cdots \gamma_m y^{n-2} + \cdots + (-1)^{m+n} a_{mn}\gamma_3 \cdots \gamma_m \alpha_3 \cdots \alpha_n \,.$$

Substituting Eq. (vii) into Eq. (iii) shows that $a(y) = a_1(y) \cdot a_2(y)$. Next let

$$d_1(x) = \beta_1\beta_2 x^2 - \beta_1\beta_2(\delta_1 + \delta_2)x + \beta_1\beta_2\delta_1\delta_2$$

and

$$d_2(x) = (-1)^n a_{mn}\beta_3 \cdots \beta_n x^{m-2} + \cdots (-1)^{m+n} a_{mn}\beta_3 \cdots \beta_n \delta_3 \cdots \delta_m \,.$$

Substituting Eq. (vi) into (iv) shows that $d(x) = d_1(x) \cdot d_2(x)$.

Using a similar argument it can be ascertained that Eqs. (ii) and (iii) can be written as products $b(y) = b_1(y) \cdot b_2(y)$ and $c(x) = c_1(x) \cdot c_2(x)$, respectively.

The polynomial $p(x, y)$ is now defined by defining p_{0j} to be the coefficient of y^j in $a_1(y)$, p_{2j} the coefficient of y^j in $b_1(y)$, p_{i0} to the coefficient of x^i in $d_1(x)$, p_{i2} the coefficient of x^i in $c_1(x)$, and $p_{11} = 0$.

In order to define the polynomial $q(x, y)$, set q_{0j} equal to the coefficient of y^j in $a_2(y)$, $q_{m-2,j}$ the coefficient of y^j in $b_2(y)$, q_{i0} the coefficient of x^i in $d_2(x)$, $q_{i,n-2}$ the coefficient of x^i in $c_2(x)$, and set all other coefficients equal to zero.

By defining $r(x, y) = t(x, y) - p(x, y) \cdot q(x, y)$ it follows from the conditions on the roots that the only indices for which the coefficients r_{ij} of $r(x, y)$ are possibly nonzero, occur when $1 \leq i \leq m - 1$ and $1 \leq j \leq n - 1$. That is, $r(x, y)$ can be written as

$$r(x, y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} r_{ij} x^i y^j \,. \tag{ix}$$

To prove the converse, observe that if $t(x, y) = p(x, y) \cdot q(x, y) + r(x, y)$, where $r(x, y)$ is as in Eq. (ix), then Eqs. (i)—(iv) can again be used to show that either $\alpha_1\alpha_2\delta_1\delta_2 = \beta_1\beta_2\gamma_1\gamma_2$ and $\alpha_3 \cdots \alpha_n\delta_3 \cdots \delta_m = \beta_3 \cdots \beta_n\gamma_3 \cdots \gamma_m$ or $\alpha_1\alpha_2\gamma_1\gamma_2 = \beta_1\beta_2\delta_1\delta_2$ and $\alpha_3 \cdots \alpha_n\gamma_3 \cdots \gamma_m = \beta_3 \cdots \beta_n\delta_3 \cdots \delta_m$.

Q.E.D.

According to Eq. 5.4.3, any 5×5 template can be decomposed into a weak decomposition of five 3×3 templates. However, given a 5×5 template **t** which satisfies the hypothesis of Theorem **5.5.1**, then **t** has a weak decomposition of only three 3×3 templates

$$\mathbf{t} = \mathbf{p} \bigoplus \mathbf{q} + \mathbf{r} \,.$$

5.6 Polynomial Decomposition of Special Types of Templates

For many templates, the conditions listed in the hypothesis of the Manseur-Wilson Theorem (Theorem **5.5.1**) are often difficult to establish. First the roots of the boundary polynomials $a(y)$, $b(y)$, $c(x)$, and $d(x)$ have to be determined and then their ordered products (Eqs. (II) and (III) of Theorem **5.5.1**) must be computed. Once this has been accomplished, the polynomials (templates) $p(x, y)$, $q(x, y)$ and $r(x, y)$ have to be constructed as in the proof of Theorem **5.5.1**. Some of these tasks can be simplified by using commercially available computer algebra software tools. However, there is a large class of commonly used templates whose weak decomposition can be ascertained by simple inspection. These templates exhibit specific symmetric properties.

We begin our discussion by considering properties of symmetric and skew symmetric *column* templates. The treatment of symmetric and skew symmetric row templates is identical.

5.6.1 **Definition.** A polynomial $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ is said to be *symmetric with respect to n* whenever $a_i = a_{n-i}$ for $i = 0, 1, \ldots, n$. A row template is *symmetric* if its corresponding weight polynomial is symmetric.

According to Eq. 5.4.1, the weight polynomial corresponding to a column template **t** is of form

$$t(x) = \sum_{i=0}^{2m} a_i x^i \,. \tag{5.6.1}$$

Hence our interest will be focused on polynomials that are symmetric with respect to an even integer $n = 2m$.

Another important observation is that Definition **5.6.1** makes no provision for $p(x)$ to be of degree *n*. For example, the polynomial $p(x) = -x + 2x^2 - x^3$ satisfies the symmetric property with respect to $n = 4$ since $p(x) = 0 - x + 2x^2 - x^3 + 0 \cdot x^4$. This also shows that although *n* is even, the degree of $p(x)$ need not be even.

Closely associated with the concept of symmetry is the concept of skew symmetry.

5.6.2 **Definition.** A polynomial $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ is said to be *skew symmetric with respect to n* whenever $a_i = -a_{n-i}$ for $i = 0, 1, \ldots, n$. A column template is called *skew symmetric* if and only if its corresponding weight polynomial is skew symmetric.

Suppose $\alpha \neq 0$. If $p(x)$ is symmetric with respect to *n*, then

$$\begin{aligned}\frac{1}{\alpha^n} p(\alpha) &= \frac{a_0}{\alpha^n} + \frac{a_1}{\alpha^{n-1}} + \cdots + a_n \\ &= a_n \frac{1}{\alpha^n} + a_{n-1} \frac{1}{\alpha^{n-1}} + \cdots + a_0 \\ &= p(1/\alpha)\,.\end{aligned}$$

Similarly, if $p(x)$ is skew symmetric with respect to *n*, then $-\frac{1}{\alpha^n} p(\alpha) = p(1/\alpha)$. Therefore, $p(\alpha) = 0 \;\Leftrightarrow\; p(1/\alpha) = 0$ for both symmetric and skew symmetric polynomials. This proves the following theorem.

5.6.3 **Theorem.** *If $p(x)$ is a polynomial satisfying the symmetric or skew symmetric property with respect to n, then a nonzero number α is a root of $p(x)$ if and only if $1/\alpha$ is a root of $p(x)$.*

Suppose $p(x) = 3x^2 + x^3 + 3x^4$. Since $p(x) = 0 + 0 \cdot x + 3x^2 + x^3 + 3x^4 + 0 \cdot x^5 + 0 \cdot x^6$, $p(x)$ satisfies the symmetric condition with respect to $n = 6$. The polynomial $p(x)$ can be factored as $p(x) = 3x^2 q(x)$, where $q(x) = 1 + \frac{1}{3}x + x^2$. The factor $3x^2$ has degree $2 = n - deg(p(x))$, and leading coefficient equal to the leading coefficient of $p(x)$. In addition, $q(x)$ is of degree $deg(p(x)) - 2$ and is also symmetric with respect to $deg(p(x)) - 2$. More generally, the following result holds:

5.6.4 **Theorem.** *If $p(x)$ is a polynomial of degree k and $p(x)$ satisfies the symmetric property with respect to n, then*

$$p(x) = a_k x^j q(x)\,,$$

where $j = n - k$, a_k denotes the leading coefficient of $p(x)$, and $q(x)$ is a polynomial of degree $k - j$ which satisfies the symmetric (skew symmetric) property with respect to $k - j$.

Proof: Let $p(x) = a_0 + a_1 x + \cdots + a_k x^k$. If $p(x)$ is symmetric, then $a_j = a_{n-j} = a_{n-(n-k)} = a_k$. If $p(x)$ is skew symmetric, then $a_j = -a_k$. Thus, in either case $a_j \neq 0$. Therefore,

$$\begin{aligned} p(x) &= a_j x^j + a_{j+1} x^{j+1} \cdots + a_k x^k \\ &= a_k x^j q(x)\,, \end{aligned}$$

where

$$q(x) = \pm 1 + \frac{a_{j+1}}{a_k} x + \frac{a_{j+2}}{a_k} x^2 + \cdots + x^{k-j}\,,$$

and we use the plus sign if $p(x)$ is symmetric and the minus sign if $p(x)$ is skew symmetric. In either case, $q(x)$ is of degree $k - j$.

Let $q(x) = b_0 + b_1 x + \cdots + b_m x^m$, where $m = k - j$ and $b_i = a_{j+i}/a_k$ for $i = 0, 1, \ldots, m$. If $p(x)$ is symmetric, then

$$\begin{aligned} b_i &= \frac{a_{j+i}}{a_k} = \frac{a_{(n-k)+i}}{a_k} = \frac{a_{n-(k-i)}}{a_k} = \frac{a_{k-i}}{a_k} \\ &= \frac{a_{j+(k-j-i)}}{a_k} = b_{k-j-i} = b_{m-i}\,. \end{aligned}$$

Therefore, $q(x)$ is symmetric with respect to $m = k - j$.

Similarly, if $p(x)$ is skew symmetric, then

$$b_i = \frac{a_{j+i}}{a_k} = \frac{a_{n-(k-i)}}{a_k} = \frac{a_{k-i}}{a_k} = -b_{m-i}\,.$$

Hence $q(x)$ is skew symmetric with respect to $m = k - j$.

Q.E.D.

Note that this theorem is true for both even and odd integers *n*.

5.6.5 **Theorem.** *Suppose $p(x)$ is a polynomial which satisfies the symmetric property with respect to an even integer n. If the degree of $p(x)$ is k and $j = n - k$, then*

$$p(x) = a_k x^j q_0(x) q_1(x) \cdots q_{(k-j)/2}(x)\,,$$

where

$$q_i(x) = \begin{cases} 1 & if \;\; i = 0 \\ x^2 + b_i x + 1 & if \;\; k - j > 0 \;\; and \;\; 1 \le i \le \frac{n-j}{2}\,. \end{cases}$$

Proof: By Theorem **5.6.4**, $p(x) = a_k x^j q(x)$, where $deg(q(x)) = k - j$. If k is even, then $k - j$ is even since $j = n - k$ is even. If k is odd, then j is odd and, hence, $k - j$ is even. Thus, in either case, $k - j$ is divisible by 2.

Now if $k - j = 0$, then set $q_0(x) = 1 = q(x)$ and the result follows. If $k - j > 0$, then it follows from the Fundamental Theorem of Algebra and Theorem **5.6.3** that $q(x)$ can be factored as

$$\begin{aligned} q(x) &= (x - \alpha_1)(x - 1/\alpha_1)(x - \alpha_2)(x - 1/\alpha_2) \cdots \big(x - \alpha_{(k-j)/2}\big)\big(x - 1/\alpha_{(k-j)/2}\big) \\ &= \big(x^2 + b_1 x + 1\big)\big(x^2 + b_2 x + 1\big) \cdots \big(x^2 + b_{(k-j)/2} x + 1\big), \end{aligned}$$

where $b_i = -\alpha_i - 1/\alpha_i$.

Q.E.D.

In contrast to Theorem **5.6.4**, the requirement for *n* to be even is necessary. For example, if $p(x) = 3x^3 + 2x^4 + 2x^5 + 3x^6$, then $p(x)$ is of even degree but symmetric with respect to the odd integer $n = 9$. In this case we have $p(x) = 3x^3 q(x)$, where $q(x) = 1 + \frac{2}{3}x + \frac{2}{3}x^2 + x^3$ is of odd degree and can, therefore, not be factored into a sequence of quadratic polynomials.

Another observation is that in contrast to Theorem **5.6.4**, Theorem **5.6.5** does not hold for skew symmetric polynomials. The polynomial $p(x) = x^2 - 1$ is skew symmetric with respect to $n = 2$, but $x^2 - 1 \neq x^2 + bx + 1$ for any number *b*. In fact, the polynomial $x^2 - 1$ is a common factor of any polynomial which is skew symmetric with respect to an even integer.

5.6.6 **Examples:**

(i) According to Eq. 5.4.1, the weight polynomial corresponding to a row template **t** is of form

$$t(y) = \sum_{j=0}^{2n} a_j y^j\,. \qquad \text{I.}$$

If one allows for complex roots, then it follows from the Fundamental Theorem of Algebra that **t** has an exact factorization $\mathbf{t} = \mathbf{p} \bigoplus \mathbf{q}$, where **p** and **q** are of size at most $1 \times (n + 1)$ since $t(y) = p(y) \cdot q(y)$, where

$$p(y) = \sum_{j=0}^{n} p_j y^j \;\; and \;\; q(y) = \sum_{j=0}^{n} q_j y^j\,. \qquad \text{II.}$$

For example, the row template

$$\mathbf{t} = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline\end{array}$$

has weight polynomial $t(y) = 1 + y + y^2 + y^3 + y^4$ which factors as

$$t(y) = \left(1 + p_1 y + y^2\right)\left(1 + q_1 y + y^2\right),$$
$$\text{where } p_1 = \frac{1+\sqrt{5}}{2} \text{ and } q_1 = \frac{1-\sqrt{5}}{2}\,.$$

Thus,

$$\mathbf{t} = \begin{array}{|c|c|c|}\hline 1 & p_1 & 1 \\ \hline\end{array} \oplus \begin{array}{|c|c|c|}\hline 1 & q_1 & 1 \\ \hline\end{array}$$

(ii) If **d** is a *diagonal* template, then the weight polynomial of **d** is of form

$$d(x,y) = \sum_{i=0}^{2n} a_{ii} x^i y^i\,.$$

However, it is more convenient to represent diagonal templates in terms of polynomials of one variable, namely

$$d(x) = \sum_{i=0}^{2n} d_i x^i\,,$$

where $d_i = a_{ii}$.

It follows from (i) above that such templates can always be factored exactly into two diagonal templates of smaller size (see also O'Leary's Theorem (**5.3.4**)). For instance, a 5×5 diagonal template will have decomposition of form

$$\mathbf{d} = \begin{array}{|c|c|c|c|c|}\hline d_1 & & & & \\ \hline & d_2 & & & \\ \hline & & d_3 & & \\ \hline & & & d_4 & \\ \hline & & & & d_5 \\ \hline\end{array} = \begin{array}{|c|c|c|}\hline p_1 & & \\ \hline & p_2 & \\ \hline & & p_3 \\ \hline\end{array} \oplus \begin{array}{|c|c|c|}\hline q_1 & & \\ \hline & q_2 & \\ \hline & & q_3 \\ \hline\end{array}$$

We turn our attention to the decomposition of skew symmetric row or column templates.

5.6.7 Theorem. *If $p(x)$ is a polynomial which satisfies the skew symmetric property with respect to an even integer n, then*

$$p(x) = \left(x^2 - 1\right)q(x)\,,$$

where $q(x)$ is a polynomial which is symmetric with respect to $n-2$.

Proof: Let $n = 2m$ and $p(x) = a_0 + a_1 x + \cdots + a_n x^n$. Since $p(x)$ is skew symmetric, $a_i = -a_{n-i}$ for $i = 0, 1, \ldots, m$. Also, since $a_m = -a_m$, $a_m = 0$. Therefore,

$$\begin{aligned} p(1) &= \sum_{i=0}^{2m} a_i = \sum_{i=0}^{m} a_i + \sum_{i=m}^{2m} a_i \\ &= \sum_{i=0}^{m} a_i - \sum_{i=0}^{m} a_i = 0\,. \end{aligned}$$

Thus, $x = 1$ is a root of $p(x)$ and $p(x)$ has a common linear factor $x - 1$.

An analogous argument shows that $p(-1) = 0$. Therefore, $p(x)$ has a common linear factor $x + 1$. Hence $(x-1)(x+1) = x^2 - 1$ is a common factor of $p(x)$ and $p(x)$ can be written as

$$p(x) = \left(x^2 - 1\right)q(x)\,,$$

where $q(x) = q_0 + q_1 x + \cdots + q_{n-2}x^{n-2}$. It remains to be shown that $q(x)$ is symmetric with respect to $n-2$.

Note that

$$\begin{aligned} p(x) &= \left(x^2 - 1\right)\left(q_0 + q_1 x + \cdots q_{n-2}x^{n-2}\right) \\ &= -q_0 - q_1 x - q_2 x^2 - \cdots - q_{n-2}x^{n-2} + q_0 x^2 + q_1 x^3 + \cdots + q_{n-2}x^n \\ &= -q_0 - q_1 x + (q_0 - q_2)x^2 + (q_1 - q_3)x^3 + \cdots \\ &\quad \cdots + (q_{n-4} - q_{n-2})x^{n-2} + q_{n-3}x^{n-1} + q_{n-2}x^n \\ &= a_0 + a_1 x + \cdots + a_n x^n\,. \end{aligned}$$

Therefore,

$$\begin{aligned} &-q_0 = a_0 = -a_n = -q_{n-2}\,, \\ &-q_1 = a_1 = -a_{n-1} = -q_{n-3}\,, \\ &\text{and}\;\; q_{i-2} - q_i = a_i \;\;\text{for}\;\; i = 2, \ldots, n-2\,. \end{aligned}$$

Thus, we have $q_0 = q_{n-2}$ and $q_1 = q_{n-3}$.

We now proceed inductively. Assume that $q_i = q_{(n-2)-i}$ for some $i \geq 1$. Since

$$q_{i-1} - q_{i+1} = a_{i+1} = -a_{n-(i+1)}\,,$$

we have that

$$\begin{aligned} q_{i+1} &= a_{n-(i+1)} + q_{i-1} \\ &= \left(q_{[n-(i+1)]-2} - q_{n-(i+1)}\right) + q_{i-1} \\ &= q_{(n-2)-(i+1)} - \left(q_{n-(i+1)} - q_{i-1}\right) \\ &= q_{(n-2)-(i+1)} - \left(q_{n-(i+1)} - q_{n-(i+1)}\right) \\ &= q_{(n-2)-(i+1)} \end{aligned}$$

since, by induction, $q_{i-1} = q_{(n-2)-(i-1)} = q_{n-(i+1)}$.

Therefore $q(x)$ is symmetric with respect to $n-2$.

Q.E.D.

Combining the preceding three theorems, we obtain

5.6.8 **Theorem.** *If $p(x)$ is a polynomial of degree k satisfying the skew symmetric property with respect to an even positive integer n, then there exists an integer j such that*

$$p(x) = a_k x^j \left(x^2 - 1\right) q_0(x) q_1(x) \cdots q_{l/2}(x)\,,$$

where a_k denotes the leading coefficient of $p(x)$, $l = (k-j) - 2$, and $q_i(x)$ is of form

$$q_i(x) = \begin{cases} 1 & if \;\; i = 0 \\ x^2 + b_i x + 1 & if \;\; i = 1, \ldots, l/2\,. \end{cases}$$

Proof: Let *k* denote the degree of $p(x)$. Then according to Theorem **5.6.4**, $p(x) = a_k x^j q(x)$, where $j = n - k$, a_k denotes the leading coefficient of $p(x)$, and $q(x)$ is a polynomial of degree $k - j$ which satisfies the skew symmetric property with respect to $k - j$.

Since *n* is even, $k - j$ is even. Thus, we can apply the results of Theorem **5.6.7** to $q(x)$ in order to obtain

$$q(x) = \left(x^2 - 1\right) p_1(x)\,,$$

where $p_1(x)$ is a polynomial of degree $(k-j) - 2$ which is symmetric with respect to $l = (k-j) - 2$. Since *l* is even, Theorem **5.6.5** can be applied to $p_1(x)$. Since the degree of $p_1(x)$ is *l*,

$$p_1(x) = a_l q_0(x) q_1(x) \cdots q_{l/2}(x)\,,$$

where $q_i(x) = x^2 + b_i x + 1$ for $i \geq 1$. However, according to the proof of Theorem **5.6.4**, the leading coefficient $a_l = 1$. Therefore,

$$\begin{aligned} p(x) &= a_k x^j \left(x^2 - 1\right) p_1(x) \\ &= a_k x^j \left(x^2 - 1\right) q_0(x) q_1(x) \cdots q_{l/2}(x)\,. \end{aligned}$$

Q.E.D.

According to Theorem **5.6.5**, any symmetric column template whose weight polynomial is given by Eq. 5.6.1 can be decomposed into a product of at most $(k-j)/2+1$ templates

$$\mathbf{t} = \mathbf{s} \oplus \mathbf{t}_1 \oplus \mathbf{t}_2 \oplus \cdots \oplus \mathbf{t}_{(k-j)/2}\,, \tag{5.6.4}$$

where $\mathbf{s}$ denotes the one-point template defined by

$$\mathbf{s}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} a_k & \text{if } \mathbf{x} = \mathbf{y} \\ 0 & \text{if } \mathbf{x} \neq \mathbf{y}\,, \end{cases}$$

and $\mathbf{t}_i$ is of form

$$\mathbf{t}_i \;=\; \begin{array}{|c|} \hline 1 \\ \hline b_i \\ \hline 1 \\ \hline \end{array}$$

for $i = 1, \ldots, (k-j)/2$. In particular, the convolution $\mathbf{a} \oplus \mathbf{t}$ can be achieved in terms of at most $(k-j)/2$ convolutions using only 3×1 templates:

$$\mathbf{a} \oplus \mathbf{t} = (\mathbf{b} \oplus \mathbf{t}_1) \oplus \mathbf{t}_2 \oplus \cdots \oplus \mathbf{t}_{(k-j)/2}\,,$$

where $\mathbf{b} = a_k \cdot \mathbf{a}$.

Similarly, if $\mathbf{t}$ is skew symmetric, then according to Theorem 5.6.8, $\mathbf{t}$ is the product of at most $\frac{l}{2}+2$ templates

$$\mathbf{t} = \mathbf{s} \oplus \mathbf{r} \oplus \mathbf{t}_1 \oplus \mathbf{t}_2 \oplus \cdots \oplus \mathbf{t}_{l/2}\,, \tag{5.6.5}$$

where $\mathbf{s}$ and $\mathbf{t}_i$ are as above and

$$\mathbf{r} \;=\; \begin{array}{|c|} \hline 1 \\ \hline \\ \hline -1 \\ \hline \end{array}$$

There are certain polynomials which are *almost* skew symmetric and can be decomposed using the decomposition method for skew symmetric polynomials.

5.6.9 **Definition.** A polynomial $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{2m}x^{2m}$ is said to be *skew symmetric with respect to its pivot coefficient* a_m if $a_i = -a_{2m-i}$ for $i = 0, 1, \ldots, m-1$. A column template is *skew symmetric with respect to its target point* if its corresponding weight polynomial is skew symmetric with respect to its pivot coefficient.

For example, the polynomial $p(x) = x^2 + 2x - 1$ is not skew symmetric since $a_1 \neq -a_{2-1}$, but $p(x)$ is skew symmetric with respect to its pivot coefficient a_1.

If $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{2m}x^{2m}$ is skew symmetric with respect to its pivot coefficient a_m, then

$$p(x) = p_1(x) + u(x),$$

where $p_1(x)$ is skew symmetric with respect to $2m$ and $u(x) = a_m x^m$. Thus, $p_1(x)$ can be decomposed using the methods of Theorem **5.6.8**. Therefore, if **t** is a column template which is skew symmetric with respect to its target point, then $\mathbf{t} = \mathbf{v} + \mathbf{u}$, where **u** is the one point template

$$\mathbf{u}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} a_m & \text{if } \mathbf{x} = \mathbf{y} \\ 0 & \text{if } \mathbf{x} \neq \mathbf{y} \end{cases}$$

and **v** is the composition of at most $\frac{l}{2} + 2$ templates given by Eq. 5.6.5. The convolution $\mathbf{a} \oplus \mathbf{t}$ can, therefore, be accomplished in terms of at most $\frac{l}{2} + 1$ convolutions using only 3×1 templates:

$$\mathbf{a} \oplus \mathbf{t} = \big(\cdots((\mathbf{b} \oplus \mathbf{r}) \oplus \mathbf{t}_1) \cdots \oplus \mathbf{t}_{l/2}\big) + \mathbf{c},$$

where $\mathbf{b} = a_k \cdot \mathbf{a}$ and $\mathbf{c} = a_m \cdot \mathbf{a}$.

Having laid the foundation for the decomposition of symmetric and skew symmetric row and column templates, we now extend these ideas to two-dimensional templates.

5.6.10 **Definition.** A polynomial of form $p(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} x^i y^j = \sum_{i=0}^{m} x^i p_i(y)$, where $p_i(y) = \sum_{j=0}^{n} a_{ij} y^j$ for $i = 0, 1, \ldots, m$, is called *symmetric* (respectively *skew symmetric*) with respect to *y* if $p_i(y)$ is symmetric (respectively skew symmetric) with respect to *n* for all $i = 0, 1, \ldots, m$. A template **t** is said to be *symmetric* (respectively *skew symmetric*) with respect to *y* if its corresponding weight polynomial is symmetric (respectively skew symmetric) with respect to *y*.

A similar definition can be given for the variable *x*. However, one can always restrict the decomposition methodology to the *y* variable by simply transposing a template which is symmetric or skew symmetric with respect to *x* and then taking the transposes of the resulting smaller templates which are symmetric or skew symmetric with respect to *y*.

5.6.11 **Theorem.** *If $p(x, y)$ is any polynomial of two variables, then $p(x, y) = p_1(x, y) + p_2(x, y)$, where $p_1(x, y)$ is symmetric with respect to y and $p_2(x, y)$ is skew symmetric with respect to y.*

Proof: Suppose $p(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} a_{ij}x^i y^j$. If one rewrites $p(x,y)$ as

$$p(x,y) = \sum_{j=0}^{n} a_{0j}y^j + x\sum_{j=0}^{n} a_{1j}y^j + x^2\sum_{j=0}^{n} a_{2j}y^j + \cdots + x^m\sum_{j=0}^{n} a_{mj}y^j ,$$

then it is easy to see that each term $x^i \sum_{j=0}^{n} a_{ij}y^j$ can be decomposed as

$$\begin{aligned} x^i\sum_{j=0}^{n} a_{ij}y^j &= x^i\sum_{j=0}^{n}\left(\frac{a_{ij}+a_{i,n-j}}{2} + \frac{a_{ij}-a_{i,n-j}}{2}\right)y^j \\ &= x^i\sum_{j=0}^{n}\left(\frac{a_{ij}+a_{i,n-j}}{2}\right)y^j + x^i\sum_{j=0}^{n}\left(\frac{a_{ij}-a_{i,n-j}}{2}\right)y^j \\ &= x^i p_{1i}(y) + x^i p_{2i}(y) , \end{aligned}$$

where

$$p_{1i}(y) = \sum_{j=0}^{n}\left(\frac{a_{ij}+a_{i,n-j}}{2}\right)y^j \text{ and } p_{2i}(y) = \sum_{j=0}^{n}\left(\frac{a_{ij}-a_{i,n-j}}{2}\right)y^j .$$

It follows from the definition of these two polynomials that $p_{1i}(y)$ is symmetric with resect to *n* and that $p_{2i}(y)$ is skew symmetric with respect to *n* for all $i = 0, 1, \ldots, m$. Therefore, the polynomial

$$p_1(x,y) = \sum_{i=0}^{m} x^i p_{1i}(y)$$

is symmetric with respect to *y* and the polynomial

$$p_2(x,y) = \sum_{i=0}^{m} x^i p_{2i}(y)$$

is symmetric with respect to *y*.

Obviously, $p_1(x,y) + p_2(x,y) = p(x,y)$.

Q.E.D.

The theorem implies that *any* template $\mathbf{t}$ whose weight polynomial is given by Eq. 5.4.1 can be decomposed into a sum $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$, where $\mathbf{t}_1$ is symmetric and $\mathbf{t}_2$ is skew symmetric.

5.6.12 Example: Let $\mathbf{t}$ be the template defined by

$\mathbf{t}$ =

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

and let $\sum_{i=0}^{4} \sum_{j=0}^{4} a_{ij} x^i y^j$ represent the weight polynomial of $\mathbf{t}$.
Using the equations

$$b_{ij} = \sum_{j=0}^{4} \left(\frac{a_{ij} + a_{i,n-j}}{2} \right) \quad \text{and} \quad c_{ij} = \sum_{j=0}^{4} \left(\frac{a_{ij} - a_{i,n-j}}{2} \right)$$

to compute the coefficients of the templates $\mathbf{t}_1$ and $\mathbf{t}_2$, respectively, we obtain

$\mathbf{t}_1$ =

11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15

and $\mathbf{t}_2$ =

-10	-5		5	10
-10	-5		5	10
-10	-5		5	10
-10	-5		5	10
-10	-5		5	10

with $\mathbf{t}_1 + \mathbf{t}_2 = \mathbf{t}$.

The template $\mathbf{t}_2$ in the above example is skew symmetric with respect to y and can be decomposed into the product $\mathbf{t}_2 = \mathbf{r} \oplus \mathbf{s}$, where $\mathbf{r}$ and $\mathbf{s}$ are the row and column templates given by

$$\mathbf{r} = \begin{array}{|c|c|c|c|c|} \hline -1 & -1/2 & & 1/2 & 1 \\ \hline \end{array} \qquad \text{and} \qquad \mathbf{s} = \begin{array}{|c|} \hline 10 \\ \hline 10 \\ \hline 10 \\ \hline 10 \\ \hline 10 \\ \hline \end{array}$$

respectively.

The row template $\mathbf{r}$ is skew symmetric with corresponding weight polynomial given by $r(y) = y^4 + \frac{1}{2}y^3 - \frac{1}{2}y - 1$. According to Theorem **5.6.8**, $r(y) = \left(y^2 - 1\right)\left(y^2 + b_1 y + 1\right)$. Obviously, $b_1 = \frac{1}{2}$. Hence, $\mathbf{r}$ can be factored into the product $\mathbf{r} = \mathbf{r}_1 \oplus \mathbf{r}_2$, where $\mathbf{r}_1$ and $\mathbf{r}_2$ are defined by

$$\mathbf{r}_1 = \begin{array}{|c|c|c|} \hline -1 & & 1 \\ \hline \end{array} \qquad \text{and} \qquad \mathbf{r}_2 = \begin{array}{|c|c|c|} \hline 1 & 1/2 & 1 \\ \hline \end{array}$$

The template $\mathbf{s}$ on the other hand is symmetric, with corresponding weight polynomial given by $s(x) = 10x^4 + 10x^3 + 10x^2 + 10x + 10$. Applying Theorem **5.6.5**, one obtains

$$s(x) = 10q_1(x)q_2(x) = 10\left(x^2 + b_1 x + 1\right)\left(x^2 + b_2 x + 1\right).$$

Since $\left(x^2 + b_1 x + 1\right)\left(x^2 + b_2 x + 1\right) = x^4 + x^3 + x^2 + x + 1$, it is not difficult to ascertain that $b_1 = \frac{1+\sqrt{5}}{2}$ and $b_2 = \frac{1-\sqrt{5}}{2}$ satisfies this equation. It follows that $\mathbf{s} = 10(\mathbf{s}_1 \oplus \mathbf{s}_2)$, where

$$\mathbf{s}_1 = \begin{array}{|c|} \hline 1 \\ \hline b_1 \\ \hline 1 \\ \hline \end{array} \qquad \text{and} \qquad \mathbf{s}_2 = \begin{array}{|c|} \hline 1 \\ \hline b_2 \\ \hline 1 \\ \hline \end{array}$$

Thus, $\mathbf{t}_2 = 10(\mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{s}_1 \oplus \mathbf{s}_2)$. By defining $\mathbf{u} = \mathbf{r}_1 \oplus \mathbf{s}_1$ and $\mathbf{v} = \mathbf{r}_2 \oplus \mathbf{s}_2$, one may write $\mathbf{t}_2$ as a product of two 3×3 templates; namely $\mathbf{t}_2 = 10(\mathbf{u} \oplus \mathbf{v})$, where

$$\mathbf{u} \oplus \mathbf{v} = \begin{array}{|c|c|c|} \hline -1 & & 1 \\ \hline b_1 & & b_1 \\ \hline -1 & & 1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline 1 & 1/2 & 1 \\ \hline b_2 & b_2/2 & b_2 \\ \hline 1 & 1/2 & 1 \\ \hline \end{array}$$

The reason for the easy decomposition of $\mathbf{t}_2$ into the product of two 3×3 templates is that $\mathbf{t}_2$ is not only skew symmetric with respect to *y*, but also symmetric with respect to *x*. In contrast, the template $\mathbf{t}_1$ in Example **5.6.12** is symmetric with respect to *y* but has no symmetry property with respect to *x*. Hence, decomposing $\mathbf{t}_1$ into a product of a row and column template $\mathbf{t}_1 = \mathbf{r} \oplus \mathbf{s}$, where

$$\mathbf{r} = \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} \qquad \text{and} \qquad \mathbf{s} = \begin{array}{|c|} \hline 11 \\ \hline 12 \\ \hline 13 \\ \hline 14 \\ \hline 15 \\ \hline \end{array}$$

then $\mathbf{r}$ is necessarily symmetric with respect to *y*, while $\mathbf{s}$ has no symmetry property with respect to *x*. Although $\mathbf{r}$ can be decomposed into two 1×3 row templates by use of Theorem **5.6.5** (see also Example **5.6.6** for a decomposition of this template), we have not explored methods for decomposing the nonsymmetric column template $\mathbf{s}$ into two 3×1 column templates. Thus the question arises as to whether or not $\mathbf{t}_1$ can be decomposed into a product of 3×3 templates. The next theorem answers this question.

5.6.13 **Theorem.** *If* $t(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} x^i y^j$ *is symmetric or skew symmetric with respect to y and* $a_{00} \neq 0 \neq a_{m0}$, *then there exist polynomials* $p(x, y)$, $q(x, y)$, *and* $r(x, y)$ *of form*

$$p(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} p_{ij} x^i y^j, \quad q(x, y) = \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} q_{ij} x^i y^j, \quad \textit{and}$$

$$r(x, y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} r_{ij} x^i y^j$$

such that $t(x, y) = p(x, y) \cdot q(x, y) + r(x, y)$.

Proof: Since $t(x, y)$ is either symmetric or skew symmetric with respect to *y*, it follows that $a_{0n} \neq 0$ and $a_{mn} \neq 0$. Using the notation employed in the Manseur-Wilson Theorem, the boundary polynomial $a(y)$ is either symmetric or skew symmetric, depending upon whether $t(x, y)$ is symmetric or skew symmetric, respectively. In either case, according to Theorem **5.6.3**, each root of $a(y)$ can be paired with its reciprocal and we can choose $a_1(y)$ and $a_2(y)$ such that

$$\alpha_1 \alpha_2 = 1 \quad \text{and} \quad \alpha_3 \cdots \alpha_n = 1 \tag{i}$$

Similarly, since $b(y)$ is symmetric or skew symmetric, we can choose $b_1(y)$ and $b_2(y)$ such that

$$\beta_1 \beta_2 = 1 \quad \text{and} \quad \beta_3 \cdots \beta_n = 1 \tag{ii}$$

Combining (i) and (ii), we obtain

$$\alpha_1\alpha_2 = \beta_1\beta_2 \quad \text{and} \quad \alpha_3 \cdots \alpha_{\mathrm{n}} = \beta_3 \cdots \beta_{\mathrm{n}} \tag{iii}$$

Also, since $t(x,y)$ is symmetric or skew symmetric, $c(x) = \pm d(x)$. Therefore,

$$\delta_1\delta_2 = \gamma_1\gamma_2 \quad \text{and} \quad \delta_3 \cdots \delta_m = \gamma_3 \cdots \gamma_m \tag{iv}$$

But (iii) and (iv) imply

$$\alpha_1\alpha_2\delta_1\delta_2 = \beta_1\beta_2\gamma_1\gamma_2 \quad \text{and} \quad \alpha_3 \cdots \alpha_n\delta_3 \cdots \delta_m = \beta_3 \cdots \beta_n\gamma_3 \cdots \gamma_m$$

Thus, the conditions of the Manseur-Wilson Theorem are satisfied and the result follows.

Q.E.D.

The theorem shows that the 5×5 template $\mathbf{t}_1$ defined in Example **5.6.12** can be decomposed into a weak sum of at most three 3×3 templates.

The proof of the next theorem provides a simple method for decomposing rectangular templates that exhibit symmetry at their *corners*.

5.6.14 **Theorem.** *If* $t(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} a_{ij}x^iy^j$ *has the property that* $a_{00} = a_{0n} = a_{m0} = a_{mn} = 0$ *and* $a_{10} = a_{01}$, $a_{m,n-1} = a_{m-1,n}$, $a_{m-1,0} = a_{m,1}$, *and* $a_{0,n-1} = a_{1n}$, *then there exist polynomials* $p(x,y)$, $q(x,y)$, *and* $r(x,y)$ *such that*

$$t(x,y) = p(x,y) \cdot q(x,y) + r(x,y)\,,$$

where

$$p(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^iy^j\,, \qquad q(x,y) = \sum_{i=0}^{m-2}\sum_{j=0}^{n-2} q_{ij}x^iy^j\,,$$

and

$$r(x,y) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} r_{ij}x^iy^j\,.$$

Proof: Let $p(x,y) = x + y + xy^2 + x^2y$ and

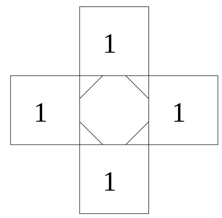
$$\begin{aligned} q(x,y) = {} & \left[a_{01} + a_{02}y + \cdots + a_{0,n-1}y^{n-2}\right] + \left[a_{2n}xy^{n-2} + a_{3n}x^2y^{n-2} + \cdots + a_{m-1,n}x^{m-1}y^{n-2}\right] \\ & + \left[a_{m1}x^{m-2} + a_{m2}x^{m-2}y + \cdots + a_{m,n-2}x^{m-2}y^{n-3}\right] + \left[a_{20}x + \cdots + a_{m-2,0}x^{m-3}\right]. \end{aligned}$$

Defining $r(x,y)$ by $r(x,y) = t(x,y) - p(x,y) \cdot q(x,y)$ provides for the desired conclusion.

Q.E.D.

The polynomial $c(x) = c_0 + c_1x + c_2x^2$, where $c_0 = a_{01}$, $c_1 = a_{00}$, and $c_2 = a_{10}$, is called the *upper left-hand corner polynomial of* $\mathbf{t}$. Upper right-hand corner and lower corner polynomials are defined in a likewise fashion. Thus, if all corner polynomials of a template **t** are symmetric and their pivot coefficients are zero, then **t** can be decomposed into a weak decomposition of three smaller templates **p**, **q**, and **r**, where **p** is given by

p =



and the templates **q** and **r** can be constructed using the description given in the proof of the theorem.

Introducing sign changes $(\pm)$ in the values of the templates **p** and **q** provides methods that are analogous to those given in the proof Theorem **5.6.14** for decomposing templates whose boundary and corner polynomials satisfy various symmetry conditions different from those given in the hypothesis of the theorem.

5.6.15 **Theorem.** *If* $t(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} x^i y^j$ *has the property that* $a_{00} = a_{0n} = a_{m0} = a_{mn} = 0$ *and* $t(x, y)$ *satisfies the following conditions:*

1. $a_{0j} \geq 0$ *and* $a_{mj} \leq 0$ *for* $j = 0, 1, \ldots, n$
2. $a_{i0} \leq 0$ *and* $a_{in} \geq 0$ *for* $i = 0, 1, \ldots, m$
3. *the upper left-hand corner and lower right-hand corner polynomials are skew symmetric*
4. *the upper right-hand corner and lower left-hand corner polynomials are symmetric,*

then there exist polynomials $p(x, y)$, $q(x, y)$, *and* $r(x, y)$ *such that*

$$t(x, y) = p(x, y) \cdot q(x, y) + r(x, y)\,,$$

where

$$p(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} p_{ij} x^i y^j\,, \qquad q(x, y) = \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} q_{ij} x^i y^j\,,$$

and

$$r(x, y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} r_{ij} x^i y^j\,.$$

Proof: Let $p(x, y) = x - y - xy^2 + x^2 y$ and

$$\begin{aligned} q(x, y) = {} & \left[a_{01} + a_{02} y + \cdots + a_{0,n-1} y^{n-2}\right] + \left[a_{2n} x y^{n-2} + a_{3n} x^2 y^{n-2} + \cdots + a_{m-1,n} x^{m-1} y^{n-2}\right] \\ & - \left[a_{m1} x^{m-2} + a_{m2} x^{m-2} y + \cdots + a_{m,n-2} x^{m-2} y^{n-3}\right] - \left[a_{20} x + \cdots + a_{m-2,0} x^{m-3}\right]. \end{aligned}$$

Defining $r(x, y)$ by $r(x, y) = t(x, y) - p(x, y) \cdot q(x, y)$ provides for the desired conclusion.

Q.E.D.

Note that according to the above proof, the template **p** is given by

p =

	-1	
1		-1
	1	

This reflects the appropriate corner symmetries of **t**. It should also be apparent how other corner symmetries and boundary conditions could be utilized in order to obtain analogous decomposition schemes. For instance, if the corners of **t** are all zero and the following conditions are satisfied

1. $a_{0j} \geq 0$ and $a_{mj} \leq 0$ for $j = 0, 1, \ldots, n$
2. $a_{i0} \geq 0$ and $a_{in} \leq 0$ for $i = 0, 1, \ldots, m$
3. the upper left-hand corner and lower right-hand corner polynomials are symmetric
4. the upper right-hand corner and lower left-hand corner polynomials are skew symmetric,

then setting $p(x, y) = -x - y + xy^2 + x^2y$,

$$\begin{aligned} q(x, y) = &\left[a_{01} + a_{02}y + \cdots + a_{0,n-1}y^{n-2}\right] - \left[a_{2n}xy^{n-2} + a_{3n}x^2y^{n-2} + \cdots + a_{m-1,n}x^{m-1}y^{n-2}\right] \\ &- \left[a_{m1}x^{m-2} + a_{m2}x^{m-2}y + \cdots + a_{m,n-2}x^{m-2}y^{n-3}\right] + \left[a_{20}x + \cdots + a_{m-2,0}x^{m-3}\right], \end{aligned}$$

and defining $r(x, y)$ by $r(x, y) = t(x, y) - p(x, y) \cdot q(x, y)$ results in the decomposition $t(x, y) = p(x, y) \cdot q(x, y) + r(x, y)$.

5.6.16 Example: If **t** is defined by

t =

	1	2	1	
1	2	3	2	1
2	3	4	3	2
1	2	3	2	1
	1	2	1	

then the corner polynomials of **t** are symmetric. According to the proof of Theorem 5.6.14,

$$\begin{aligned} q(x, y) &= \left[a_{01} + a_{02}y + a_{03}y^2\right] + \left[a_{24}xy^2 + a_{34}x^2y^2\right] + \left[a_{41}x^2 + a_{42}x^2y\right] + \left[a_{20}x\right] \\ &= \left[1 + 2y + y^2\right] + \left[2xy^2 + x^2y^2\right] + \left[x^2 + 2x^2y\right] + \left[2x\right] \\ &= 1 + 2x + x^2 + 2y + y^2 + 2xy^2 + 2x^2y + x^2y^2\,. \end{aligned}$$

Hence, $\mathbf{q}$ is given by

$\mathbf{q}$ =

1	2	1
2		2
1	2	1

and $\mathbf{p} \bigoplus \mathbf{q}$ by

$\mathbf{p} \oplus \mathbf{q}$ =

	1	2	1	
1	4	2	4	1
2	2	8	2	2
1	4	2	4	1
	1	2	1	

Since $\mathbf{r} = \mathbf{t} - \mathbf{p} \bigoplus \mathbf{q}$, we have

$\mathbf{r}$ =

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

and $\mathbf{t} = \mathbf{p} \bigoplus \mathbf{q} + \mathbf{r}$.

In the above example, the template **t** is symmetric with respect to both *x* and *y*. In addition, the centered weight matrix corresponding to **t** is a symmetric matrix. It may therefore not be surprising that the template factors **p**, **q**, and **r** exhibit the same symmetric properties. Thus, it is natural to ask whether or not this phenomenon is true in general. The next theorem answers this question in the affirmative.

5.6.17 **Theorem.** *If* $t(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{n} a_{ij} x^i y^j$ *satisfies the symmetric property with respect to both x and y and if the matrix* $\mathrm{A} = (a_{ij})_{n \times n}$ *is symmetric, then there exist polynomials* $p(x, y)$, $q(x, y)$, *and* $r(x, y)$ *such that*

$$t(x, y) = p(x, y) \cdot q(x, y) + r(x, y)\,,$$

where

$$p(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} p_{ij}x^i y^j \,, \qquad q(x,y) = \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} q_{ij}x^i y^j \,,$$

and

$$r(x,y) = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} r_{ij}x^i y^j \,.$$

Proof: If $a_{00} \neq 0$, then by symmetry $a_{n0} \neq 0$ and the result follows from Theorem **5.6.13**. The symmetry of the polynomial follows from the fact that the polynomials $a(y)$, $b(y)$, $c(x)$, and $d(x)$ defined in the proofs of either Theorem **5.6.13** or the Manseur-Wilson Theorem all have, in this case, the same coefficients and, therefore, the same decomposition.
If $a_{00} = 0$, then again by symmetry we have that $a_{0n} = a_{n0} = a_{nn} = 0$. The result now follows from Theorem **5.6.14** and the definitions of $p(x,y)$, $q(x,y)$, and $r(x,y)$ in the proof of **5.6.14**.

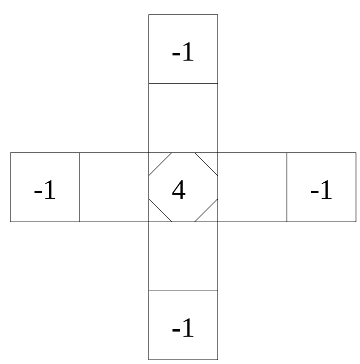
Q.E.D.

In contrast to Eq. 5.4.3, the number of 3×3 templates needed to decompose a 5×5 template is at most 3 if the template satisfies the symmetric conditions of Theorems **5.6.13**, **5.6.14**, or **5.6.17**. In general, the number of 3×3 templates needed to decompose a $(2n+1) \times (2n+1)$ template exhibiting the symmetry conditions described in Theorem **5.6.17** is at most $n(n+1)/2$.

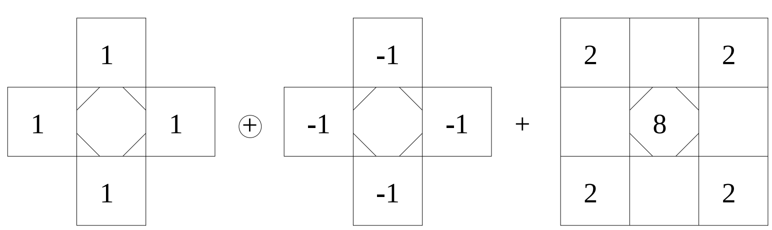
5.6.18 Examples:

(i) The Laplace template

l =



satisfies the conditions of Theorem **5.6.14** and can, therefore, be decomposed as



This represents the decomposition into templates that exhibit the same symmetric behavior as **l**. A simpler decomposition of **l** into diagonal templates is given by

$$
\begin{array}{|c|c|c|}\hline 1 & & \\ \hline & & \\ \hline & & 1 \\ \hline \end{array}
\oplus
\begin{array}{|c|c|c|}\hline & & -1 \\ \hline & & \\ \hline -1 & & \\ \hline \end{array}
+
\begin{array}{|c|}\hline 4 \\ \hline \end{array}
$$

This shows that the theorems discussed in this chapter need not provide the most efficient decompositions.

(ii) The template

$$
\mathbf{h} = \begin{array}{|c|c|c|c|c|}\hline -13 & 2 & 7 & 2 & -13 \\ \hline 2 & 17 & 22 & 17 & 2 \\ \hline 7 & 22 & 27 & 22 & 7 \\ \hline 2 & 17 & 22 & 17 & 2 \\ \hline -13 & 2 & 7 & 2 & -13 \\ \hline \end{array}
$$

used by Haralick [6] satisfies the hypothesis of Theorem **5.6.17**. Adopting the construction used in the proof of the Manseur-Wilson Theorem, results in the following decomposition of **h**:

$$
\begin{array}{|c|c|c|}\hline -13 & -19.73 & -13 \\ \hline -19.73 & -28.943 & -19.73 \\ \hline -13 & -19.73 & -13 \\ \hline \end{array}
\oplus
\begin{array}{|c|c|c|}\hline 1 & -1.672 & 1 \\ \hline -1.672 & 1.5411 & -1.672 \\ \hline 1 & -1.672 & 1 \\ \hline \end{array}
+
\begin{array}{|c|}\hline 8.39 \\ \hline \end{array}
$$

(iii) The template

$$\mathbf{t} = \begin{array}{|c|c|c|c|c|}\hline & 1 & & 1 & \\ \hline & & & & \\ \hline & & & & \\ \hline 1 & & & & 1 \\ \hline & & & & \\ \hline \end{array}$$

is symmetric with respect to *y*, but does not satisfy the hypothesis of either the Manseur-Wilson Theorem or the hypotheses of Theorems **5.6.13**, **5.6.14**, or **5.6.17**. It can be shown that **t** cannot be decomposed as $\mathbf{t} = \mathbf{p} \bigoplus \mathbf{q} + \mathbf{r}$, where each **p**, **q**, and **r** is of size at most 3×3 [10]. However, using *LU* factorization, **t** can still be decomposed as a weak decomposition of *four* 3×3 templates.

(iv) Similar to example (iii), the template

$$\mathbf{t} = \begin{array}{|c|c|c|c|c|}\hline & 1 & & 1 & \\ \hline & & 2 & & \\ \hline & 1 & 1 & 1 & \\ \hline 1 & 1 & & 1 & 1 \\ \hline & & & & \\ \hline \end{array}$$

is symmetric with respect to *y*, but does not satisfy the hypothesis of either the Manseur-Wilson Theorem or the hypotheses of Theorems **5.6.13**, **5.6.14**, or **5.6.17**. Z. Manseur has shown that **t** cannot be decomposed as $\mathbf{t} = (\mathbf{t}_1 \bigoplus \mathbf{t}_2) + (\mathbf{t}_3 \bigoplus \mathbf{t}_4)$ such that each $\mathbf{t}_i$ is of size at most 3×3 [10]. Of course, employing the constructive method used in the proof of Theorem **5.4.1**, **t** can be decomposed into a weak sum $\mathbf{t} = (\mathbf{t}_1 \bigoplus \mathbf{t}_2) + (\mathbf{t}_3 \bigoplus \mathbf{t}_4) + \mathbf{r}$ such that each $\mathbf{t}_i$ and **r** are 3×3 templates.

5.7 Decomposition of Variant Templates

In this section we shall be concerned with decomposition techniques for variant templates that are analogous to the techniques discussed in the preceding sections. We shall restrict our discussion to rectangular templates.

Rectangular $m \times n$ variant templates can be specified by a family of $m \times n$ matrices $T(\mathbf{y}) = \left(t(\mathbf{y})_{ij}\right)_{m\times n}$, where $\mathbf{y} \in \mathbb{Z}^2$, $t(\mathbf{y})_{ij} = \mathbf{t}_{\mathbf{y}}(i(\mathbf{y})_{min} + (i-1), j(\mathbf{y})_{min} + (j-1))$, and $i(\mathbf{y})_{min}$ and $j(\mathbf{y})_{min}$ are defined by Eq. 5.2.1. The matrix $T(\mathbf{y})$ is called the *weight matrix associated with* **t** *at* **y**. Whenever **y** is understood, we simply write T for $T(\mathbf{y})$ and t_{ij} for $t(\mathbf{y})_{ij}$.

In analogy with centered weight matrices associated with invariant templates, we also define the notion centered weight matrices for rectangular variant templates. In particular, if **t** is a rectangular template and *m* and *n* are positive integers such that $\mathbb{Z}_{\pm m} \times \mathbb{Z}_{\pm n}$ is the smallest rectangular array containing $S(\mathbf{t}_{(0,0)})$, then the *centered* weight matrix associated with **t** at $\mathbf{y} = (y_1, y_2)$ is defined as $T(\mathbf{y}) = \left(t(\mathbf{y})_{ij}\right)_{\pm m\times \pm n}$, where $t(\mathbf{y})_{ij} = \mathbf{t}_{\mathbf{y}}(y_1 + i, y_2 + j)$, $-m \le i \le m$, and $-n \le j \le n$. Note that since **t** is a rectangular template, the array $\{(y_1+i, y_2+j) : -m \le i \le m, -n \le j \le n\}$ is the smallest rectangular array which is *centered* at $\mathbf{y} = (y_1, y_2)$ and contains $S(\mathbf{t}_{\mathbf{y}})$.

Obviously, if $\mathbf{t} = \mathbf{r} \oplus \mathbf{s}$ and **t** is variant, then **r** and **s** cannot both be invariant. However, it is often possible to decompose **t** into a pair of non-trivial smaller templates such that one of the factors is an invariant template. Thus, again, the transform computation can be optimized and templates can be tailored to fit particular architectures. Having one factor invariant can further enhance the speed of computation.

As a first example, we decompose an arbitrary 5×5 template.

5.7.1 Theorem. *If* **t** *is a* 5×5 *template, then there exist templates* $\mathbf{t}_i$, $i = 1, 2, \ldots, 7$, *such that*

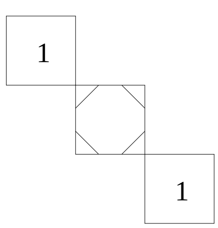
$$\mathbf{t} = (\mathbf{t}_1 \oplus \mathbf{t}_2) + (\mathbf{t}_3 \oplus \mathbf{t}_4) + (\mathbf{t}_5 \oplus \mathbf{t}_6) + \mathbf{t}_7\,,$$

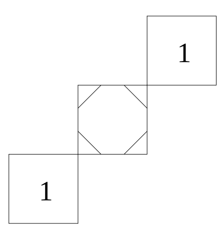
where $\mathbf{t}_1$, $\mathbf{t}_3$ *and* $\mathbf{t}_5$ *are invariant* 3×3 *templates, and* $\mathbf{t}_2$, $\mathbf{t}_4$, $\mathbf{t}_6$ *and* $\mathbf{t}_7$ *are variant* 3×3 *templates.*

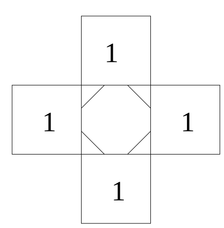
Proof: The centered weight matrix $T(\mathbf{y})$ associated with **t** at $\mathbf{y} = (y_1, y_2)$ is of form

$$T(\mathbf{y}) = \begin{pmatrix} t(\mathbf{y})_{-2,-2} & t(\mathbf{y})_{-2,-1} & t(\mathbf{y})_{-2,0} & t(\mathbf{y})_{-2,1} & t(\mathbf{y})_{-2,2} \\ t(\mathbf{y})_{-1,-2} & t(\mathbf{y})_{-1,-1} & t(\mathbf{y})_{-1,0} & t(\mathbf{y})_{-1,1} & t(\mathbf{y})_{-1,2} \\ t(\mathbf{y})_{0,-2} & t(\mathbf{y})_{0,-1} & t(\mathbf{y})_{0,0} & t(\mathbf{y})_{0,1} & t(\mathbf{y})_{0,2} \\ t(\mathbf{y})_{1,-2} & t(\mathbf{y})_{1,-1} & t(\mathbf{y})_{1,0} & t(\mathbf{y})_{1,1} & t(\mathbf{y})_{1,2} \\ t(\mathbf{y})_{2,-2} & t(\mathbf{y})_{2,-1} & t(\mathbf{y})_{2,0} & t(\mathbf{y})_{2,1} & t(\mathbf{y})_{2,2} \end{pmatrix}.$$

Define the three invariant 3×3 templates $\mathbf{t}_1$, $\mathbf{t}_3$ and $\mathbf{t}_5$ by

$\mathbf{t}_1$ = 

, $\mathbf{t}_3$ = 

, and $\mathbf{t}_5$ = 

Without loss of generality set $t_{ij} = t(\mathbf{y})_{ij}$ and define

$$\mathbf{t}_2 = \begin{array}{|c|c|c|} \hline t_{-2,-2} & t_{-2,-1} & \\ \hline t_{-1,-2} & & t_{1,2} \\ \hline & t_{2,1} & t_{2,2} \\ \hline \end{array} \;, \quad \mathbf{t}_4 = \begin{array}{|c|c|c|} \hline & t_{-2,1} & t_{-2,2} \\ \hline t_{1,-2} & & t_{-1,2} \\ \hline t_{2,-2} & t_{2,-1} & \\ \hline \end{array}$$

and

$$\mathbf{t}_6 = \begin{array}{|c|c|c|} \hline & t_{-2,0} & \\ \hline t_{0,-2} & & t_{0,2} \\ \hline & t_{2,0} & \\ \hline \end{array}$$

Now let $\mathbf{t}_7 = \mathbf{t} - [(\mathbf{t}_1 \bigoplus \mathbf{t}_2) + (\mathbf{t}_3 \bigoplus \mathbf{t}_4) + (\mathbf{t}_5 \bigoplus \mathbf{t}_6)]$. Then

$$(\mathbf{t}_7)_{\mathbf{y}} = \begin{cases} t_{-1,-1} - (t_{-2,0} + t_{0,-2}) & \text{if } \mathbf{x} = (y_1 - 1, y_2 - 1) \\ t_{-1,0} - (t_{1,2} + t_{1,-2}) & \text{if } \mathbf{x} = (y_1 - 1, y_2) \\ t_{-1,1} - (t_{-2,0} + t_{0,2}) & \text{if } \mathbf{x} = (y_1 - 1, y_2 + 1) \\ t_{0,-1} - (t_{2,1} + t_{-2,1}) & \text{if } \mathbf{x} = (y_1, y_2 - 1) \\ t_{0,0} - (t_{-2,-2} + t_{2,2} + t_{0,2} + t_{-2,0} + t_{0,-2} + t_{2,0}) & \text{if } \mathbf{x} = \mathbf{y} \\ t_{0,1} - (t_{2,1} + t_{-2,1}) & \text{if } \mathbf{x} = (y_1, y_2 + 1) \\ t_{1,-1} - (t_{0,-2} + t_{2,0}) & \text{if } \mathbf{x} = (y_1 + 1, y_2 - 1) \\ t_{1,0} - (t_{-1,2} + t_{-1,-2}) & \text{if } \mathbf{x} = (y_1 + 1, y_2) \\ t_{1,1} - (t_{2,0} + t_{0,2}) & \text{if } \mathbf{x} = (y_1 + 1, y_2 + 1) \\ 0 & \text{otherwise}. \end{cases}$$

It follows from these definitions that

$$\mathbf{t} = (\mathbf{t}_1 \bigoplus \mathbf{t}_2) + (\mathbf{t}_3 \bigoplus \mathbf{t}_4) + (\mathbf{t}_5 \bigoplus \mathbf{t}_6) + \mathbf{t}_7 .$$

Q.E.D.

Theorem **5.7.1** is void of any type of symmetry assumptions. If $\mathbf{t}$ satisfies certain symmetric conditions, then the number of computations can be significantly reduced. The next theorem is analogous to Theorem **5.6.14**.

5.7.2 **Theorem.** *If* **t** *is a* $(2m+1) \times (2n+1)$ *template whose associated weight matrix* $T(\mathbf{y})$ *satisfies the following conditions*

1. $t_{-m,-n} = t_{-m,n} = t_{m,-n} = t_{m,n} = 0$
2. $t_{-m+1,-n} = t_{-m,-n+1}$, $t_{m,n-1} = t_{m-1,n}$, $t_{m-1,-n} = t_{m,-n+1}$, *and* $t_{-m,n-1} = t_{-m+1,n}$

then there exist three templates **p**, **q**, *and* **r** *such that*

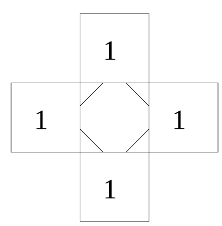
$$\mathbf{t} = (\mathbf{p} \bigoplus \mathbf{q}) + \mathbf{r} \,,$$

where **p** *is a* 3×3 *invariant template and* **q** *and* **r** *are* $(2m-1) \times (2n-1)$ *variant templates*

Proof: Define **q** in terms of the centered weight matrix $T(\mathbf{y})$ as follows:

$$\mathbf{q}_{\mathbf{y}}(i,j) = \begin{cases} t_{i-1,j} & \text{if } i = -(m-1) \text{ and } -(n-1) \le j \le n-1 \\ t_{i+1,j} & \text{if } i = m-1 \text{ and } -(n-1) \le j \le n-1 \\ t_{i,j-1} & \text{if } j = -(n-1) \text{ and } -(m-2) \le i \le m-2 \\ t_{i,j+1} & \text{if } j = n-1 \text{ and } -(m-2) \le i \le m-2 \\ 0 & \text{otherwise}\,. \end{cases}$$

Note that the support of $\mathbf{q}_{\mathbf{y}}$ consists of the non-zero boundary values of $T(\mathbf{y})$ and that the entries of $Q(\mathbf{y})$ (the matrix associated with $\mathbf{q}_{\mathbf{y}}$) are all zero except along the *boundary* of the matrix. Convolving **q** with

p = 

results in the template $\mathbf{p} \bigoplus \mathbf{q}$ which agrees with **t** along the boundary of $T(\mathbf{y})$. Thus, the template $\mathbf{r} = \mathbf{t} - (\mathbf{p} \bigoplus \mathbf{q})$ is of size $(2m-1) \times (2n-1)$

Q.E.D.

The remarks and observations that followed the proof of Theorem **5.6.14** also apply here. Variant templates whose corner polynomials exhibit specific symmetry properties and whose boundary values satisfy certain sign $(\pm)$ rules can be decomposed into weak sums as $(\mathbf{p} \bigoplus \mathbf{q}) + \mathbf{r}$ with appropriate sign changes in the values of the templates **p** and **q** defined in the proof of Theorem **5.7.2**.

In Sections **5.2** and **5.3** we observed that rank 1 invariant templates are linearly separable. This observation also holds for invariant templates.

5.7.3 **Definition.** A rectangular template $\mathbf{t}$ has *column rank 1* if and only if there exists a column vector $\mathbf{u}$ such that for every $\mathbf{y} \in \mathbb{Z}^2$ and for every column vector $\mathbf{v}(\mathbf{y})$ in $T(\mathbf{y})$, there exists scalars $\lambda_{\mathbf{v}(\mathbf{y})}$ such that $\mathbf{v}(\mathbf{y}) = \lambda_{\mathbf{v}(\mathbf{y})} \cdot \mathbf{u}$.

A similar definition can be formulated to define templates having *row rank 1.*Whenever $\mathbf{y}$ is understood, we write $\mathbf{v}$ for $\mathbf{v}(\mathbf{y})$ and $\lambda_{\mathbf{v}}$ for $\lambda_{\mathbf{v}(\mathbf{y})}$.

5.7.4 **Theorem.** *If* $\mathbf{t}$ *is a template having column rank 1, then there exists an invariant column template* $\mathbf{s}$ *and a row template* $\mathbf{r}$ *such that*

$$\mathbf{t} = \mathbf{s} \bigoplus \mathbf{r}\,.$$

Proof: Suppose $\mathbf{t}$ is of size $m \times n$, $\mathbf{y} \in \mathbb{Z}^2$ is arbitrary, and $T(\mathbf{y})$ is the $m \times n$ weight matrix associated with $\mathbf{t}$ at $\mathbf{y}$. Let $\mathbf{v}_1, \mathbf{v}_1, \ldots, \mathbf{v}_n$ denote the column vectors of $T(\mathbf{y})$. By hypothesis there exists a fixed column vector $\mathbf{u} = (u_1, u_2, \ldots, u_m)$ and scalars $\lambda_{\mathbf{v}_1}, \lambda_{\mathbf{v}_2}, \ldots, \lambda_{\mathbf{v}_n}$ such that $\mathbf{v}_i = \lambda_{\mathbf{v}_i} \cdot \mathbf{u}$ $(i = 1, 2, \ldots, n)$.

Therefore,

$$\begin{aligned} T(\mathbf{y}) &= (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n) \\ &= (\lambda_{\mathbf{v}_1} \cdot \mathbf{u}, \lambda_{\mathbf{v}_2} \cdot \mathbf{u}, \ldots, \lambda_{\mathbf{v}_n} \cdot \mathbf{u}) \qquad \text{(i)} \\ &= \mathbf{u} \cdot (\lambda_{\mathbf{v}_1}, \lambda_{\mathbf{v}_2}, \ldots, \lambda_{\mathbf{v}_n})\,. \end{aligned}$$

Define the templates $\mathbf{r}$ and $\mathbf{s}$ by

$$\mathbf{r}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} \lambda_{\mathbf{v}_i} & \text{if } p_2(\mathbf{x}) = j(\mathbf{y})_{min} + (j-1),\ 1 \le j \le n,\ \text{and } p_1(\mathbf{x}) = p_1(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \qquad \text{(ii)}$$

and

$$\mathbf{s}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} u_i & \text{if } p_1(\mathbf{x}) = i(\mathbf{y})_{min} + (i-1),\ 1 \le i \le m,\ \text{and } p_2(\mathbf{x}) = p_2(\mathbf{y}) \\ 0 & \text{otherwise}\,, \end{cases} \qquad \text{(iii)}$$

respectively.

It now follows from (i), (ii), and (iii) that

$$\mathbf{t} = \mathbf{s} \bigoplus \mathbf{r}\,.$$

Q.E.D.

5.7.5 **Example:** Let $\mathbf{X} = \mathbb{Z}^2$, $\mathbf{y} = (y_1, y_2) \in \mathbf{X}$, and let $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ be given by

$$\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} 2^{y_i} & \text{if } \mathbf{x} = (y_1 - 5, y_2 + (i+1)),\ i = 1, 2 \\ 2^{y_i+1} & \text{if } \mathbf{x} = (y_1 - 4, y_2 + (i+1)),\ i = 1, 2 \\ -3 \cdot 2^{y_i} & \text{if } \mathbf{x} = (y_1 - 3, y_2 + (i+1)),\ i = 1, 2\,. \end{cases}$$

It is easy to verify that $\mathbf{y} \notin R(\mathbf{t_y})$ and that $\mathbf{t}$ is strictly rectangular and has column rank 1. Figure **5.7.1** provides an illustration of $\mathbf{t}$ at location $\mathbf{y} = (1, 2)$.

$$\mathbf{t}_{(1,2)} = \begin{array}{|c|c|c|c|}\hline & & 2 & 4 \\ \hline & & 4 & 8 \\ \hline & & -6 & -12 \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline\end{array}$$

Figure 5.7.1 The function $\mathbf{t}_{(1,2)}$ with corresponding rectangle $R\big(\mathbf{t}_{(1,2)}\big)$ outlined in bold.

It is also not difficult to ascertain that the row template $\mathbf{r}$ in the decomposition $\mathbf{t} = \mathbf{s} \bigoplus \mathbf{r}$ is given by

$$\mathbf{r_y} = \begin{array}{|c|c|c|c|}\hline & & 2^{y_2} & 2^{y_1} \\ \hline\end{array}$$

while the column template $\mathbf{s}$ is given by

$$\mathbf{s_y} = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline -3 \\ \hline \\ \hline \\ \hline \\ \hline\end{array}$$

The notion of "column rank 1" has the following generalization.

5.7.6 **Definition.** A rectangular template has *column rank at most k* if and only if there exist *k* column vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ such that for each $\mathbf{y} \in \mathbb{Z}^2$ and every column vector $\mathbf{v}$ in $T(\mathbf{y})$ there exists scalars $\lambda_{\mathbf{v}}(j)$ such that

$$\mathbf{v} = \sum_{j=1}^{k} \lambda_{\mathbf{v}}(j) \cdot \mathbf{u}_j .$$

5.7.7 **Theorem.** *If* **t** *is a template having column rank at most k, then there exist k column rank 1 templates* $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k$ *such that*

$$\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_k .$$

Proof: Suppose **t** is of size $m \times n$, $\mathbf{y} \in \mathbb{Z}^2$ arbitrary, and suppose $T(\mathbf{y})$ is the $m \times n$ matrix associated with **t** at **y**. Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ denote the column vectors of $T(\mathbf{y})$. By hypothesis, there exist fixed column vectors $\mathbf{u}_j = (u_{j_1}, u_{j_2}, \ldots, u_{j_m})'$, $j = 1, 2, \ldots, k$, and scalars $\lambda_{\mathbf{v}_i}(j)$ such that

$$\mathbf{v}_i = \sum_{j=1}^{k} \lambda_{\mathbf{v}_i}(j) \cdot \mathbf{u}_j .$$

Define the templates $\mathbf{t}_l$, where by $l = 1, 2, \ldots, k$,

$$(\mathbf{t}_l)_{\mathbf{y}}(\mathbf{x}) = \begin{cases} \lambda_{\mathbf{v}_j}(l) \cdot u_{l_i} & \text{if } p_1(\mathbf{x}) = i(\mathbf{y}) + i \\ & \text{and } p_2(\mathbf{x}) = j(\mathbf{y}) + j \\ 0 & \text{otherwise} . \end{cases}$$

The weight matrix associated with $\mathbf{t}_l$ at **y** is given by

$$T_l(\mathbf{y}) = (\lambda_{\mathbf{v}_1}(l) \cdot \mathbf{u}_l \, , \lambda_{\mathbf{v}_2}(l) \cdot \mathbf{u}_l \, , \ldots , \lambda_{\mathbf{v}_n}(l) \cdot \mathbf{u}_l \,) .$$

Since

$$T_l(\mathbf{y}) = \mathbf{u}_l \cdot (\lambda_{\mathbf{v}_1}(l), \lambda_{\mathbf{v}_2}(l), \ldots , \lambda_{\mathbf{v}_n}(l) \,)$$

for each $\mathbf{y} \in \mathbb{Z}^2$, $\mathbf{t}_l$ has rank 1.

Also, since

$$\begin{aligned} \sum_{l=1}^{k} T_l(\mathbf{y}) &= \left(\sum_{l=1}^{k} \lambda_{\mathbf{v}_1}(l) \cdot \mathbf{u}_l \, , \sum_{l=1}^{k} \lambda_{\mathbf{v}_2}(l) \cdot \mathbf{u}_l \, , \ldots , \sum_{l=1}^{k} \lambda_{\mathbf{v}_n}(l) \cdot \mathbf{u}_l \right) . \\ &= (\mathbf{v}_1, \mathbf{v}_2, \ldots , \mathbf{v}_n) = T(\mathbf{y}) , \end{aligned}$$

we have that

$$\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_k .$$

Q.E.D.

Since each template $\mathbf{t}_i$ in the above theorem has column rank 1, it follows from Theorem **5.7.4** that each template $\mathbf{t}_i$ can be decomposed as $\mathbf{t}_i = \mathbf{s}_i \bigoplus \mathbf{r}_i$, where $\mathbf{s}_i$ is an invariant column template and $\mathbf{r}_i$ is a row template. The following corollary is, therefore, an immediate consequence of Theorems **5.7.4** and **5.7.7**.

5.7.8 **Corollary.** *If* **t** *is a rectangular template having column rank at most k, then there exist k invariant column templates* $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$ *and k invariant row templates* $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k$ *such that*

$$\mathbf{t} = \sum_{i=1}^{k} \mathbf{s}_i \bigoplus \mathbf{r}_i \,.$$

Similar results can, of course, be proved for row rank *k* templates.

5.8 Local Decomposition of Templates

In order to take advantage of the capabilities of computers with parallel, reconfigurable, or distributive architectures, there is a need to understand the process of computing global linear transforms locally. In this as well as the subsequent section we discuss necessary and sufficient conditions for the existence of local decompositions of linear transformations with respect to directed networks of processors.

For the remainder of this section let **X** be a finite set and $N : \mathbf{X} \to 2^{\mathbf{X}}$ a function such that $\mathbf{x} \in N(\mathbf{x}) \;\; \forall \mathbf{x} \in \mathbf{X}$. We shall also assume that $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$.

We may view the elements of **X** as processors in a network. The assumption that $\mathbf{x} \in N(\mathbf{x})$ can be interpreted as meaning that every processor in the network has direct access to its own memory. Similarly, the interpretation of $\mathbf{y} \in N(\mathbf{x})$ is that processor **x** has direct access to the memory of processor **y**. Thus, $N(\mathbf{x})$ is a collection of processors to which processor **x** has direct access. We call *N* a *processor neighborhood configuration.*

5.8.1 **Definition.** A template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is said to be *local* with respect to *N* if and only if $S(\mathbf{t}_{\mathbf{y}}) \subset N(\mathbf{y}) \;\; \forall \mathbf{y} \in \mathbf{X}$. If *N* is understood, then we say that **t** is a *local template.*

If $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ and **t** has a linear decomposition $\mathbf{t} = \mathbf{t}_1 \bigoplus \mathbf{t}_2 \bigoplus \cdots \bigoplus \mathbf{t}_n$ such that for each $i \in \{1, 2, \ldots, n\}$ $\mathbf{t}_i$ is local with respect to *N*, then we say that the sequence $\{\mathbf{t}_i\}_{i=1}^{n}$ is a *local decomposition* of **t** with respect to *N*. Again, if *N* is understood, then we say that **t** has a local decomposition whenever such a sequence exists.

Several concepts from graph theory are essential in our discussion of local decomposition of templates. The following list of definitions defines the necessary concepts.

5.8.2 **Definition.** A *directed graph*, or *digraph*, is a pair $D = (V, E)$, where $E \subset V \times V$ and *V* is a finite set. The elements of *V* are called the *vertices* of *D*. If $(x, y) \in E$, then (x, y) is called an *arc* (or *edge*) from *x* to *y*. If there is need, we may write $V(D) = V$ and $E(D) = E$ to emphasize that *V* and *E* are the sets of vertices and arcs associated with the digraph *D*, respectively.

5.8.3 **Definition.** A *graph G* is a *digraph* $G = (V, E)$ with the property that $(x, y) \in E \Rightarrow (y, x) \in E$. If $(x, y) \in E$, then we say that *xy*, or equivalently *yx*, is an *edge* of *G*.

5.8.4 **Definition.** Let $D = (V, E)$ be a digraph and $x, y \in V$. An *x-y walk* in *D*, or *a walk from x to y*, is a finite sequence of vertices $x = z_0, z_1, \ldots, z_{n-1}, z_n = y$, such that $(z_i, z_{i+1}) \in E$ for $i \in \{0, 1, \ldots, n-1\}$. An *x-y* walk is called a *closed walk* if $x = y$.

5.8.5 **Definition.** Let $D = (V, E)$ be a digraph and $x, y \in V$. An *x-y path* in *D*, or *a path from x to y*, is an *x-y* walk with distinct vertices, except possibly *x* and *y*.

5.8.6 **Definition.** Let $D = (V, E)$ be a digraph and $x, y \in V$. An *x-y path* in *D*, or *a path from x to y*, is an *x-y* walk with distinct vertices, except possibly *x* and *y*.

A *closed x-y path* is an *x-y* path such that $x = y$.

The sequence of vertices in an *x-y* walk will be denoted by $P(x, y)$; i.e., $P(x, y) = \{z_0, z_1, \ldots, z_n\}$, where $z_0 = x$ and $z_n = y$.

5.8.7 **Definition.** Let $D = (V, E)$ be a digraph and $x, y \in V$. We say that *y is reachable from x* (*in D*) if there exists a path from *x* to *y*. If *y* is reachable from *x* and *x* is reachable from *y*, then we say that the pair $\{x, y\}$ is *mutually reachable* (*in D*).

Note that if $G = (V, E)$ is a graph and $x, y \in V$ such that there exists an *x-y* path in *G*, then there exists a *y-x* path in *G*. Hence reachable and mutually reachable are equivalent notions in graphs.

5.8.8 **Definition.** Let $D = (V, E)$ be a digraph. We say that *D* is *strongly connected* if and only if every pair of vertices $\{x, y\} \subset V$ is mutually reachable. If *D* is a graph that is strongly connected, then *D* is called a *connected* graph.

We now set up a connection between the processor neighborhood configuration function $N : \mathbf{X} \to 2^{\mathbf{X}}$ and digraphs.

5.8.9 **Definition.** Let $N : \mathbf{X} \to 2^{\mathbf{X}}$ be a neighborhood configuration. For each $\mathbf{x} \in \mathbf{X}$, let $E(N(\mathbf{x})) = \{(\mathbf{y}, \mathbf{x}) : \mathbf{y} \in N(\mathbf{x})\}$. The *digraph of N*, denoted by $D(N)$, is the digraph defined by $D(N) = (\mathbf{X}, E(N))$, where $E(N) = \bigcup\limits_{\mathbf{x} \in \mathbf{X}} E(N(\mathbf{x}))$.

A neighborhood configuration *N* is called *symmetric* if and only if for every pair $\mathbf{x}, \mathbf{y} \in \mathbf{X}$, $\mathbf{x} \in N(\mathbf{y}) \Rightarrow \mathbf{y} \in N(\mathbf{x})$. This notion has the following consequence.

5.8.10 **Theorem.** *$D(N)$ is a graph $\Leftrightarrow$ N is symmetric.*

Proof: Suppose $D(N)$ is a graph. Let $\mathbf{x}_0, \mathbf{y}_0 \in \mathbf{X}$ with $\mathbf{x}_0 \in N(\mathbf{y}_0)$. By definition, $(\mathbf{x}_0, \mathbf{y}_0) \in E(N)$. Since $D(N)$ is a graph, $(\mathbf{y}_0, \mathbf{x}_0) \in E(N)$ and, hence, $(\mathbf{y}_0, \mathbf{x}_0) \in E(N(\mathbf{x}_0))$. Therefore, $\mathbf{y}_0 \in N(\mathbf{x}_0)$ and *N* is symmetric.

Conversely, assume that *N* is symmetric and that $(\mathbf{x}_0, \mathbf{y}_0) \in E(N)$. Then $\mathbf{x}_0 \in N(\mathbf{y}_0)$ and, by symmetry, $\mathbf{y}_0 \in N(\mathbf{x}_0)$. Thus, $(\mathbf{y}_0, \mathbf{x}_0) \in E(N(\mathbf{x}_0)) \subset E(N)$. Therefore, $(\mathbf{y}_0, \mathbf{x}_0) \in E(N)$ and $D(N)$ is a graph.

Q.E.D.

If *N* is symmetric, then we let $G(N)$ denote the graph of *N*.

The next theorem provides necessary and sufficient conditions for the local decomposition of a template with respect to a given processor neighborhood configuration. Although the theorem is disguised in the language of image algebra, it was first stated and proved by M. Tchuente [19, 18].

5.8.11 **Theorem** (Tchuente)**.** *If* $\mathbf{X}$ *is a finite point set and* $N : \mathbf{X} \to 2^{\mathbf{X}}$ *a symmetric neighborhood configuration, then every* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *has a local decomposition with respect to N* $\Leftrightarrow G(N)$ *is connected.*

Tchuente's theorem is not as general as one might like. The assumption that *N* is symmetric restricts the type of multiprocessors being modeled to those in which the data can flow in both directions along any communication links. In many important cases, such as pipeline and parallel-pipeline computers, and systolic arrays, this assumption is not valid [7, 12, 15, 3, 4]. In order to provide for effective computations of linear transforms of these types of architectures, P. Gader proved the following generalization of Tchuente's theorem [3, 4].

5.8.12 **Theorem** (Gader)**.** *Suppose* $\mathbf{X}$ *is a finite point set and* $N : \mathbf{X} \to 2^{\mathbf{X}}$ *a neighborhood configuration. Then every template* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *has a local decomposition with respect to N* $\Leftrightarrow D(N)$ *is strongly connected.*

Although Gader's and Tchuente's appear almost identical, it happens — as is often true in graph theory — that the case for directed graphs is significantly different than for graphs [1]. There are two differences between graphs and digraphs which emerge when one tries to use a straightforward generalization of Tchuente's proof. One difference is that any permutation graph can be accomplished by a sequence of adjacent transpositions. This is not necessarily true for digraphs (e.g., consider the directed cycle). Another difference is that for a connected graph there is always at least one vertex which can be removed, along with an edge incident to it, such that the resulting graph is still connected. This is not necessarily true for digraphs (consider again the directed cycle). Tchuente relies on both of these facts in order to prove his theorem.

5.9 Necessary and Sufficient Conditions for the Existence of Local Decompositions

This section is devoted to the proof of Gader's theorem. With the exception of some slight changes, we essentially follow Gader's proof by proving the theorem in stages. We first prove that strong connectivity is a necessary condition. We then establish a sequence of theorems which show that strong connectivity is sufficient for permutation matrices. Finally, we show how certain matrix decompositions can be used in conjunction with the results for permutation matrices to deduce that strong connectivity is sufficient in general.

5.9.1 **Theorem.** *Let* $\mathbf{X}$ *be a finite point set and let* $N : \mathbf{X} \to 2^{\mathbf{X}}$ *a neighborhood configuration. If every* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *has a local decomposition with respect to N, then* $D(N)$ *is strongly connected.*

Proof: Suppose to the contrary that $D(N)$ is not strongly connected, but that every $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ has a local decomposition with respect to *N*. Since $D(N)$ is not strongly connected there exists points $\mathbf{x}, \mathbf{y} \in \mathbf{X}$ such that **x** is *not* reachable from **y**.

Let $\mathbf{r} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ be defined by

$$\mathbf{r}_{\mathbf{z}}(\mathbf{w}) = \begin{cases} 1 & \text{if } \mathbf{z} = \mathbf{x} \text{ and } \mathbf{w} = \mathbf{y} \\ 0 & \text{otherwise,} \end{cases}$$

and let $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ be defined by

$$\mathbf{a}(\mathbf{w}) = \begin{cases} 1 & \text{if } \mathbf{w} = \mathbf{y} \\ 0 & \text{otherwise.} \end{cases}$$

Observe that if **b** denotes the image $\mathbf{b} = \mathbf{a} \bigoplus \mathbf{r}$, then **b** is given by

$$\mathbf{b}(\mathbf{z}) = \sum_{\mathbf{w} \in S(\mathbf{r}_{\mathbf{z}})} \mathbf{a}(\mathbf{w})\mathbf{r}_{\mathbf{z}}(\mathbf{w}) = \begin{cases} \mathbf{a}(\mathbf{y}) = 1 & \text{if } \mathbf{z} = \mathbf{x} \\ 0 & \text{otherwise.} \end{cases}$$

According to our assumption, **r** can be factored as $\mathbf{r} = \mathbf{r}_1 \bigoplus \mathbf{r}_2 \bigoplus \cdots \bigoplus \mathbf{r}_k$ such that each $\mathbf{r}_j$ is local with respect to *N*.

Define $C(\mathbf{x}) = \{\mathbf{z} \in \mathbf{X} : \mathbf{x} \text{ is reachable from } \mathbf{z}\}$ and for $j = 1, 2, \ldots, k$, let $\mathbf{b}_j = \mathbf{a} \bigoplus (\mathbf{r}_1 \bigoplus \mathbf{r}_2 \bigoplus \cdots \bigoplus \mathbf{r}_j)$. We shall show that $\mathbf{b}_j(\mathbf{z}) = 0 \ \forall \mathbf{z} \in C(\mathbf{x})$ and $\forall j = 1, 2, \ldots, k$.

Suppose that for some $\mathbf{z} \in C(\mathbf{x})$, $\mathbf{b}_1(\mathbf{z}) \neq 0$. Then there exists $\mathbf{w} \in \mathbf{X}$ such that $\mathbf{r}_{1,\mathbf{z}}(\mathbf{w}) \neq 0$ and $\mathbf{a}(\mathbf{w}) \neq 0$. Since $\mathbf{r}_{1,\mathbf{z}}(\mathbf{w}) \neq 0$, $\mathbf{w} \in S(\mathbf{r}_{1,\mathbf{z}}) \subset N(\mathbf{z})$. Hence, **z** is reachable from **w**. However, according to the definition of **a**, since $\mathbf{a}(\mathbf{w}) \neq 0$, we must have $\mathbf{w} = \mathbf{y}$. But this means that **x** is reachable from **y**, contradicting the fact that **x** is not reachable from **y**. Therefore, $\mathbf{b}_1(\mathbf{z}) = 0 \ \forall \mathbf{z} \in C(\mathbf{x})$.

Using induction, suppose that the claim holds for $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{j-1}$ and for some $j \in \{1, 2, \ldots, k\}$. If the claim does not hold for $\mathbf{b}_j$, then there exists $\mathbf{z} \in C(\mathbf{x})$ such that $\mathbf{b}_j(\mathbf{z}) \neq 0$. Thus, again, there exists $\mathbf{w} \in \mathbf{X}$ such that $\mathbf{r}_{j,\mathbf{z}}(\mathbf{w}) \neq 0$ and $\mathbf{b}_{j-1}(\mathbf{w}) \neq 0$. By induction hypothesis, $\mathbf{w} \notin C(\mathbf{x})$. Since $\mathbf{r}_{j,\mathbf{z}}(\mathbf{w}) \neq 0$, $\mathbf{w} \in S(\mathbf{r}_{j,\mathbf{z}}) \subset N(\mathbf{z})$ and, hence, **z** is reachable from **w**. Also, since $\mathbf{z} \in C(\mathbf{x})$, **x** is reachable from **z**. Thus, **x** is reachable from **w**, which leads to the contradiction that $\mathbf{w} \in C(\mathbf{x})$. Therefore, $\mathbf{b}_j(\mathbf{z}) = 0 \ \forall \mathbf{z} \in C(\mathbf{x})$ and $\forall j \in \{1, 2, \ldots, k\}$. Since $\mathbf{b}_k = \mathbf{b}$, we now have that $\mathbf{b}(\mathbf{z}) = 0 \ \forall \mathbf{z} \in C(\mathbf{x})$. But this contradicts the fact that $\mathbf{b}(\mathbf{x}) = 1$. Since this contradiction follows from our assumption that **r** has a local decomposition with respect to *N*, our assumption must be false. But this in turn contradicts the hypothesis of the theorem. Hence, our original assumption that $D(N)$ is not strongly connected must be false.

Q.E.D.

5.9.2 **Definition.** If $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$, then $\mathbf{t}$ is called a *permutation template* if and only if $\psi(\mathbf{t})$ is a permutation matrix.

It follows from Theorem **5.1.1** that if $\psi_1(\mathbf{t})$ is a permutation matrix for some ordering of **X**, then $\psi_2(\mathbf{t})$ is also a permutation matrix for any other ordering of **X**. Recall also that every permutation can be factored into a product of transpositions (Section **3.3**). Thus, every permutation template can be factored into a product of templates corresponding to transpositions. We identify these templates in the next definition.

5.9.3 **Definition.** Let $P = \mathbf{X} \times \mathbf{X}$. For each parameter $p = (\mathbf{x}, \mathbf{y}) \in P$, define $\mathbf{t} : P \to \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ by

$$\mathbf{t}(\mathbf{x}, \mathbf{y})_{\mathbf{z}}(\mathbf{w}) = \begin{cases} 1_{\mathbf{x}}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{y} \\ 1_{\mathbf{y}}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{x} \\ 1_{\mathbf{z}}(\mathbf{w}) & \text{otherwise.} \end{cases} \qquad \textbf{(5.9.1)}$$

the template $\mathbf{t}(p)$ is called the *exchange template* associated with *p*.

Observe that the exchange template $\mathbf{t}(\mathbf{x}, \mathbf{y})$ is the permutation template corresponding to the transposition $\sigma \in S_{\mathbf{X}}$ defined by $\sigma = (\mathbf{x}, \mathbf{y})$. That is, σ interchanges $\mathbf{x}$ with $\mathbf{y}$ and leaves the remaining elements of **X** fixed. This observation follows from Eq. 5.9.1 which says that $\mathbf{t}(\mathbf{x}, \mathbf{y})_{\mathbf{z}}(\mathbf{w}) = 1$ if $\mathbf{x} = \mathbf{w}$ and $\mathbf{z} = \mathbf{y}$, or if $\mathbf{y} = \mathbf{w}$ and $\mathbf{z} = \mathbf{x}$, or if $\mathbf{z} = \mathbf{w}$. Thus, if $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and $\mathbf{b} = \mathbf{a} \oplus \mathbf{t}$, then

$$\mathbf{b}(\mathbf{z}) = \begin{cases} \mathbf{a}(\mathbf{x}) & \text{if } \mathbf{z} = \mathbf{y} \\ \mathbf{a}(\mathbf{y}) & \text{if } \mathbf{z} = \mathbf{x} \\ \mathbf{a}(\mathbf{z}) & \text{otherwise.} \end{cases}$$

5.9.4 **Theorem.** *Let $N : \mathbf{X} \to 2^{\mathbf{X}}$ a neighborhood configuration on* **X** *and $\mathbf{x}, \mathbf{y} \in \mathbf{X}$. Suppose there exists an* **x**–**y** *path*

$$P_1 : \ \mathbf{x} = \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{x}_k = \mathbf{y}$$

and an **y**–**x** *path*

$$P_2 : \ \mathbf{y} = \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}_j = \mathbf{x}$$

in $D(N)$. If

$$P : \ \mathbf{x} = \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{y}_1, \ldots, \mathbf{y}_{j-1}, \mathbf{x}_j = \mathbf{x}$$

is a closed path, then the exchange template $\mathbf{t}(\mathbf{x}, \mathbf{y})$ is local with respect to N

Proof: The proof is by construction. Relabel the closed path P as
$P: \ \mathbf{x} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1}, \mathbf{p}_n = \mathbf{x}$, so that $\mathbf{p}_k = \mathbf{y}, \ \mathbf{p}_{k+1} = \mathbf{y}_1$, and $\mathbf{p}_n = \mathbf{p}_{k+j} = \mathbf{x}$. Then

$$\mathbf{t}(\mathbf{x},\mathbf{y}) = \mathbf{t}(\mathbf{p}_0,\mathbf{p}_1) \bigoplus \mathbf{t}(\mathbf{p}_1,\mathbf{p}_2) \bigoplus \cdots \bigoplus \mathbf{t}(\mathbf{p}_{k-1},\mathbf{p}_k) \bigoplus \mathbf{t}(\mathbf{p}_{k-2},\mathbf{p}_{k-1}) \bigoplus \cdots \bigoplus \mathbf{t}(\mathbf{p}_1,\mathbf{p}_2) \bigoplus \mathbf{t}(\mathbf{p}_0,\mathbf{p}_1).$$

We shall show that each template $\mathbf{t}(\mathbf{p}_{i-1},\mathbf{p}_i)$ has a local decomposition with respect to *N*. It is interesting to note that the factors of the decomposition are not permutation templates. Figure **5.9.1** is helpful in understanding the action of the templates we are about to describe. We begin by decomposing $\mathbf{t}(\mathbf{p}_0,\mathbf{p}_1)$ into a product of local templates $\mathbf{t}_{1,1} \bigoplus \mathbf{t}_{1,2} \bigoplus \cdots \bigoplus \mathbf{t}_{1,2n-1}$.
For $j = 1, 2, \ldots, n-1$, define

$$(\mathbf{t}_{1,j})_{\mathbf{z}}(\mathbf{w}) = \begin{cases} 1_{\mathbf{p}_j}(\mathbf{w}) + 1_{\mathbf{p}_{j-1}}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{p}_j \\ 1_{\mathbf{z}}(\mathbf{w}) & \text{if } \mathbf{z} \neq \mathbf{p}_j \, . \end{cases}$$

For $j = n$, define

$$(\mathbf{t}_{1,n})_{\mathbf{z}}(\mathbf{w}) = \begin{cases} 1_{\mathbf{p}_{n-1}}(\mathbf{w}) - 1_{\mathbf{p}_0}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{p}_0 \\ 1_{\mathbf{p}_i}(\mathbf{w}) - 1_{\mathbf{p}_{i-1}}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{p}_i, \ i = 2, 3, \ldots, n-1 \\ 1_{\mathbf{z}}(\mathbf{w}) & \text{otherwise.} \end{cases}$$

For $j = 1, 2, \ldots, n-3$, define

$$(\mathbf{t}_{1,n+j})_{\mathbf{z}}(\mathbf{w}) = \begin{cases} 1_{\mathbf{p}_{j+2}}(\mathbf{w}) + 1_{\mathbf{p}_{j+1}}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{p}_{j+2} \\ 1_{\mathbf{z}}(\mathbf{w}) & \text{otherwise} \, . \end{cases}$$

For $j = 2n-2$, define

$$(\mathbf{t}_{1,2n-2})_{\mathbf{z}}(\mathbf{w}) = \begin{cases} 1_{\mathbf{p}_0}(\mathbf{w}) - 1_{\mathbf{p}_{n-1}}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{p}_0 \\ 1_{\mathbf{p}_i}(\mathbf{w}) - 1_{\mathbf{p}_{i-1}}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{p}_i, \ i = 3, 4, \ldots, n-1 \\ 1_{\mathbf{z}}(\mathbf{w}) & \text{otherwise,} \end{cases}$$

and for $j = 2n-1$, define

$$(\mathbf{t}_{1,n+j})_{\mathbf{z}}(\mathbf{w}) = \begin{cases} 1_{\mathbf{p}_1}(\mathbf{w}) + 1_{\mathbf{p}_0}(\mathbf{w}) & \text{if } \mathbf{z} = \mathbf{p}_1 \\ 1_{\mathbf{z}}(\mathbf{w}) & \text{if } \mathbf{z} \neq \mathbf{p}_1 \, . \end{cases}$$

By definition, each $\mathbf{t}_{1,j}$ is a local template. We need to show that $\mathbf{t}(\mathbf{p}_0,\mathbf{p}_1) = \mathbf{t}_{1,1} \bigoplus \mathbf{t}_{1,2} \bigoplus \cdots \bigoplus \mathbf{t}_{1,2n-1}$.
Let $\mathbf{b}_1 = \mathbf{a} \bigoplus (\mathbf{t}_{1,1} \bigoplus \mathbf{t}_{1,2} \bigoplus \cdots \bigoplus \mathbf{t}_{1,n-1})$ and set $a_i = \mathbf{a}(\mathbf{p}_i)$. Then $\mathbf{b}_1$ is given by

$$\mathbf{b}_1(\mathbf{z}) = \begin{cases} \sum_{i=0}^{j} a_i & \text{if } \mathbf{z} = \mathbf{p}_j, \ j = 0, 1, \ldots, n-1 \\ \mathbf{a}(\mathbf{z}) & \text{otherwise.} \end{cases}$$

If $\mathbf{b}_2 = \mathbf{b}_1 \bigoplus \mathbf{t}_{1,n}$, then $\mathbf{b}_2$ is given by

$$\mathbf{b}_2(\mathbf{z}) = \begin{cases} \sum_{i=1}^{n-1} a_i & \text{if } \mathbf{z} = \mathbf{p}_0 \\ a_0 + a_1 & \text{if } \mathbf{z} = \mathbf{p}_1 \\ \mathbf{b}_1(\mathbf{z}) & \text{if } \mathbf{z} = \mathbf{p}_j \text{ and } 0 \neq j \neq 1 \\ \mathbf{a}(\mathbf{z}) & \text{otherwise.} \end{cases}$$

Let $\mathbf{b}_3 = \mathbf{b}_2 \bigoplus (\mathbf{t}_{1,n+1} \bigoplus \mathbf{t}_{1,n+2} \bigoplus \cdots \bigoplus \mathbf{t}_{1,2n-3})$. Then

$$\mathbf{b}_3(\mathbf{z}) = \begin{cases} \sum_{i=2}^{j} a_i & \text{if } \mathbf{z} = \mathbf{p}_j, \; j = 2, 3, \ldots, n-1 \\ \mathbf{b}_2(\mathbf{z}) & \text{if } \mathbf{z} = \mathbf{p}_0 \text{ or } \mathbf{z} = \mathbf{p}_1 \\ \mathbf{a}(\mathbf{z}) & \text{otherwise.} \end{cases}$$

Next set $\mathbf{b}_4 = \mathbf{b}_3 \bigoplus \mathbf{t}_{1,2n-2}$. Then

$$\mathbf{b}_3(\mathbf{z}) = \begin{cases} a_1 & \text{if } \mathbf{z} = \mathbf{p}_0 \\ a_0 + a_1 & \text{if } \mathbf{z} = \mathbf{p}_1 \\ \mathbf{a}(\mathbf{z}) & \text{otherwise.} \end{cases}$$

Finally, let $\mathbf{b}_5 = \mathbf{b}_4 \bigoplus \mathbf{t}_{1,2n-1}$. Then

$$\mathbf{b}_5(\mathbf{z}) = \begin{cases} a_1 & \text{if } \mathbf{z} = \mathbf{p}_0 \\ a_0 & \text{if } \mathbf{z} = \mathbf{p}_1 \\ \mathbf{a}(\mathbf{z}) & \text{otherwise.} \end{cases}$$

Since we also have $\mathbf{b}_5 = \mathbf{a} \bigoplus \mathbf{t}(\mathbf{p}_0, \mathbf{p}_1)$, it follows that

$$\mathbf{t}(\mathbf{p}_0, \mathbf{p}_1) = \mathbf{t}_{1,1} \bigoplus \mathbf{t}_{1,2} \bigoplus \cdots \bigoplus \mathbf{t}_{1,2n-1} .$$

Therefore, $\mathbf{t}(\mathbf{p}_0, \mathbf{p}_1)$ has a local decomposition.
Using an identical argument, we can construct for each $i = 2, 3, \ldots, k-1$ a local decomposition $\{\mathbf{t}_{i,j}\}_{j=1}^{2n-1}$ of $\mathbf{t}(\mathbf{p}_{i-1}, \mathbf{p}_i)$. We then have

$$\begin{aligned} \mathbf{t}(\mathbf{x}, \mathbf{y}) &= \left[\bigoplus_{i=1}^{k} \mathbf{t}(\mathbf{p}_{i-1}, \mathbf{p}_i) \right] \oplus \left[\bigoplus_{i=k-1}^{1} \mathbf{t}(\mathbf{p}_{i-1}, \mathbf{p}_i) \right] \\ &= \left[\bigoplus_{i=1}^{k} \left(\bigoplus_{j=1}^{2n-1} \mathbf{t}_{i,j} \right) \right] \oplus \left[\bigoplus_{i=k-1}^{1} \left(\bigoplus_{j=1}^{2n-1} \mathbf{t}_{i,j} \right) \right] . \end{aligned}$$

Q.E.D.

Figure **5.9.1** below illustrates the action of the local decomposition of $\mathbf{t}(\mathbf{p}_0, \mathbf{p}_1)$ on an arbitrary input image **a**.

$\mathbf{a}$	$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_3$	$\mathbf{b}_4$	$\mathbf{b}_5$
a_0	a_0	$\sum_{i=1}^{n-1} a_i$	$\sum_{i=1}^{n-1} a_i$	a_1	a_1
a_1	$a_0 + a_1$	$a_0 + a_1$	$a_0 + a_1$	$a_0 + a_1$	a_0
a_2	$\sum_{i=0}^{2} a_i$	a_2	a_2	a_2	a_2
a_3	$\sum_{i=0}^{3} a_i$	a_3	$a_2 + a_3$	a_3	a_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_{n-2}	$\sum_{i=0}^{n-2} a_i$	a_{n-2}	$\sum_{i=2}^{n-2} a_i$	a_{n-2}	a_{n-2}
a_{n-1}	$\sum_{i=0}^{n-1} a_i$	a_{n-1}	$\sum_{i=2}^{n-1} a_i$	a_{n-1}	a_{n-1}
$\mathbf{a}(\mathbf{z})$	$\mathbf{a}(\mathbf{z})$	$\mathbf{a}(\mathbf{z})$	$\mathbf{a}(\mathbf{z})$	$\mathbf{a}(\mathbf{z})$	$\mathbf{a}(\mathbf{z})$

Figure 5.9.1 The mapping stages of an input image **a** in the local implementation of $\mathbf{t}(\mathbf{p}_0, \mathbf{p}_1)$.

5.9.5 Theorem. *If $D(N)$ is strongly connected, then $\mathbf{t}(\mathbf{x}, \mathbf{y})$ has a local decomposition with respect to N.*

Proof: Since $D(N)$ is strongly connected, every pair $(\mathbf{x}, \mathbf{y})$is mutually reachable in $D(N)$. Thus, there exists **x**–**y** and **y**–**x** paths

$$P_1 : \mathbf{x} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1}, \mathbf{x}_n = \mathbf{y}$$

and

$$P_2 : \mathbf{y} = \mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{m-1}, \mathbf{q}_m = \mathbf{x}\,.$$

If P_1 and P_2 have only vertices **x** and **y** in common, then the result follows from Theorem **5.9.4**. Otherwise, label the common vertices $\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \ldots, \mathbf{p}_{i_k}$ such that $i_1 < i_2 < \cdots < i_k$. For sake of convenience, set $i_0 = 0$ and $i_{k+1} = n$. Define $j_1, j_2, \ldots, j_k$ by $\mathbf{q}_{j_l} = \mathbf{p}_{i_l}$, where

$l = 1, 2, \ldots, k$. By our choice of the $\mathbf{p}_{i_l}$s, the closed walks

$$\begin{aligned}
&W_0: \ \mathbf{x} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{i_1} = \mathbf{q}_{j_1}, \mathbf{q}_{j_1+1}, \ldots, \mathbf{q}_m = \mathbf{x} \\
&W_1: \ \mathbf{p}_{i_1}, \mathbf{p}_{i_1+1}, \ldots, \mathbf{p}_{i_2} = \mathbf{q}_{j_2}, \mathbf{q}_{j_2+1}, \ldots, \mathbf{q}_{j_1} \\
&\vdots \\
&W_k: \ \mathbf{p}_{i_k}, \mathbf{p}_{i_k+1}, \ldots, \mathbf{p}_{n-1}, \mathbf{p}_n = \mathbf{y} = \mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{j_k}
\end{aligned}$$

are all closed paths. Furthermore,

$$\begin{aligned}
\mathbf{t}(\mathbf{x},\mathbf{y}) = \mathbf{t}(\mathbf{x},\mathbf{p}_{i_1}) \oplus \mathbf{t}(\mathbf{p}_{i_1},\mathbf{p}_{i_2}) \oplus \cdots \oplus \mathbf{t}\big(\mathbf{p}_{i_{k-1}},\mathbf{p}_{i_k}\big) \oplus \mathbf{t}(\mathbf{p}_{i_k},\mathbf{y}) \oplus \mathbf{t}\big(\mathbf{p}_{i_{k-1}},\mathbf{p}_{i_k}\big) \oplus \cdots \\
\cdots \oplus \mathbf{t}(\mathbf{p}_{i_1},\mathbf{p}_{i_2}) \oplus \mathbf{t}(\mathbf{x},\mathbf{p}_{i_1}).
\end{aligned}$$

Since for each $i = 0, 1, \ldots, k$, each W_i is a closed path, each of the templates $\mathbf{t}\big(\mathbf{p}_{i_l},\mathbf{p}_{i_{l+1}}\big)$ has, according to Theorem **5.9.4**, a local decomposition for $l = 0, 1, \ldots, k-1$. Hence $\mathbf{t}(\mathbf{x},\mathbf{y})$ has a local decomposition with respect to *N*.

Q.E.D.

If $\mathbf{t} \in \big(\mathbb{F}^{\mathbf{X}}\big)^{\mathbf{X}}$ is a permutation template, then **t** can be factored into a product of exchange templates. Thus, if $D(N)$ is strongly connected, then according to the theorem each of these exchange templates has a local decomposition with respect to *N*. Therefore, **t** has a local decomposition with respect to *N*. This proves the following corollary:

5.9.6 **Corollary.** *If $D(N)$ is strongly connected and $\mathbf{t} \in \big(\mathbb{F}^{\mathbf{X}}\big)^{\mathbf{X}}$ is a permutation template, then* **t** *has a local decomposition with respect to N.*

It now follows that strong connectivity of $D(N)$ is a necessary and sufficient condition for the existence of local decompositions of all permutation templates. We now show that this holds in general.

5.9.7 **Lemma.** *Suppose $card(\mathbf{X}) = n > 1$ and $D(N)$ is strongly connected. If for some $j = 1, 2, \ldots, n-1$ there exists templates $\mathbf{r}_j, \mathbf{s}_j \in \big(\mathbb{F}^{\mathbf{X}}\big)^{\mathbf{X}}$ such that*

$$\psi(\mathbf{r}_j) = \begin{pmatrix} I_{j-1} & | & O & | & & O & \\ - & + & - & + & - & - & - \\ O & | & r_j & | & r_{j+1} & \cdots & r_n \\ - & + & - & + & - & - & - \\ O & | & O & | & & I_{n-j} & \end{pmatrix}$$

and

$$\psi(\mathbf{s}_j) = \begin{pmatrix} I_{j-1} & | & O & | & O \\ - & + & - & + & - \\ O & | & s_j & | & O \\ - & + & - & + & - \\ & | & s_{j+1} & | & \\ O & | & \vdots & | & I_{n-j} \\ & | & s_n & | & \end{pmatrix},$$

where each O denotes a zero-matrix of the appropriate dimension, then $\mathbf{r}_j$ *and* $\mathbf{s}_j$ *have local decompositions with respect to N.*

Proof: We consider $\mathbf{s}_j$ first. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote the ordering used to define ψ. If $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and $\mathbf{b} = \mathbf{a} \bigoplus \mathbf{s}_j$, then $\mathbf{b}$ is given by

$$\mathbf{b}(\mathbf{y}) = \begin{cases} \mathbf{a}(\mathbf{y}) & \text{if } \mathbf{y} \neq \mathbf{x}_j \\ \sum_{k=j}^{n} s_k \cdot \mathbf{a}(\mathbf{x}_k) & \text{if } \mathbf{y} = \mathbf{x}_j \,. \end{cases}$$

Since $card(\mathbf{X}) > 1$ and $D(N)$ is strongly connected, there exists $\mathbf{x}_m \in \{\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$ such that $\mathbf{x}_m \neq \mathbf{x}_j$ and $\mathbf{x}_m \in N(\mathbf{x}_j)$. The idea of the proof is to first multiply $\mathbf{a}(\mathbf{x}_j)$ by s_j and leave everything else fixed. Each $\mathbf{a}(\mathbf{x}_k)$ $(k = j+1, \ldots, n)$ is then *moved* to the location $\mathbf{x}_m$, $s_k \cdot \mathbf{a}(\mathbf{x}_k)$ is added to the current value at the location $\mathbf{x}_j$, and $\mathbf{a}(\mathbf{x}_k)$ is then moved back to location $\mathbf{x}_k$. All steps can be carried out locally.

We now define the templates for the local decomposition of $\mathbf{s}_j$. Define $\mathbf{t}_j$ by

$$(\mathbf{t}_j)_{\mathbf{y}} = \begin{cases} 1_{\mathbf{y}} & \text{if } \mathbf{y} \neq \mathbf{x}_j \\ s_j \cdot 1_{\mathbf{x}_j} & \text{if } \mathbf{y} = \mathbf{x}_j \,. \end{cases}$$

Obviously, $\mathbf{t}_j$ is local and if $\mathbf{b} = \mathbf{a} \bigoplus \mathbf{t}_j$, then $\mathbf{b}$ is given by

$$\mathbf{b}(\mathbf{y}) = \begin{cases} \mathbf{a}(\mathbf{y}) & \text{if } \mathbf{y} \neq \mathbf{x}_j \\ s_j \cdot \mathbf{a}(\mathbf{x}_j) & \text{if } \mathbf{y} = \mathbf{x}_j \,. \end{cases}$$

For each $k = j+1, j+2, \ldots, n$ define templates $\mathbf{u}_k$ and $\mathbf{t}_k$ by

$$(\mathbf{u}_k)_{\mathbf{y}} = \begin{cases} 1_{\mathbf{y}} & \text{if } \mathbf{y} \neq \mathbf{x}_j \\ 1_{\mathbf{x}_j} + s_k \cdot 1_{\mathbf{x}_m} & \text{if } \mathbf{y} = \mathbf{x}_j \end{cases}$$

and

$$\mathbf{t}_k = \mathbf{t}(\mathbf{x}_k, \mathbf{x}_m) \bigoplus \mathbf{u}_k \bigoplus \mathbf{t}(\mathbf{x}_m, \mathbf{x}_k) \,.$$

Since $\mathbf{x}_m \in N(\mathbf{x}_j)$, each $\mathbf{u}_k$ is local with respect to *N*. Furthermore, by Theorem **5.9.5**, each $\mathbf{t}(\mathbf{x}_k, \mathbf{x}_m)$ and $\mathbf{t}(\mathbf{x}_m, \mathbf{x}_k)$ is local with respect to *N*. Therefore $\mathbf{t}_k$ has a local decomposition with respect to *N* for each $k = j+1, j+2, \ldots, n$.

Now, if $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and $\mathbf{b} = \mathbf{a} \bigoplus \mathbf{t}_k$ for some $k = j+1, j+2, \ldots, n$, then

$$\mathbf{b}(\mathbf{y}) = \begin{cases} \mathbf{a}(\mathbf{y}) & \text{if } \mathbf{y} \neq \mathbf{x}_j \\ s_k \cdot \mathbf{a}(\mathbf{x}_k) & \text{if } \mathbf{y} = \mathbf{x}_j \,. \end{cases}$$

Hence, if $\mathbf{b} = \mathbf{a} \bigoplus (\mathbf{t}_j \bigoplus \mathbf{t}_{j+1} \bigoplus \cdots \bigoplus \mathbf{t}_n)$, then

$$\mathbf{b}(\mathbf{y}) = \begin{cases} \mathbf{a}(\mathbf{y}) & \text{if } \mathbf{y} \neq \mathbf{x}_j \\ \sum_{k=j}^{n} s_k \cdot \mathbf{a}(\mathbf{x}_k) & \text{if } \mathbf{y} = \mathbf{x}_j \,. \end{cases}$$

Therefore, $\mathbf{s}_j = \mathbf{t}_j \bigoplus \mathbf{t}_{j+1} \bigoplus \cdots \bigoplus \mathbf{t}_n$, which enables us to conclude that $\mathbf{s}_j$ has a local decomposition with respect to *N*.

We now prove the lemma for $\mathbf{r}_j$. The concept is the same; there are some points which must be modified to account for the transposition. Note that if $\mathbf{b} = \mathbf{a} \bigoplus \mathbf{r}_j$, then $\mathbf{b}$ is given by

$$\mathbf{b}(\mathbf{y}) = \begin{cases} \mathbf{a}(\mathbf{y}) & \text{if } \mathbf{y} = \mathbf{x}_k,\ k = 1, 2, \ldots, j-1 \\ r_j \cdot \mathbf{a}(\mathbf{x}_j) & \text{if } \mathbf{y} = \mathbf{x}_j \\ r_k \cdot \mathbf{a}(\mathbf{x}_j) + \mathbf{a}(\mathbf{x}_k) & \text{if } \mathbf{y} = \mathbf{x}_k,\ k = j+1, j+2, \ldots, n. \end{cases}$$

Next define a local template $\mathbf{w}_{n+1}$ by

$$(\mathbf{w}_{n+1})_{\mathbf{y}} = \begin{cases} 1_{\mathbf{y}} & \text{if } \mathbf{y} \neq \mathbf{x}_j \\ r_j \cdot 1_{\mathbf{x}_j} & \text{if } \mathbf{y} = \mathbf{x}_j\,. \end{cases}$$

Again, since $card(\mathbf{X}) > 1$ and $D(N)$ is strongly connected, for every $k = j+1, j+2, \ldots, n$ there exists an $m_k \in \{1, 2, \ldots, n\}$ such that $\mathbf{x}_{m_k} \in N(\mathbf{x}_k)$ and $\mathbf{x}_{m_k} \neq \mathbf{x}_k$.

For $k = j+1, j+2, \ldots, n$ define templates $\mathbf{v}_k$ and $\mathbf{w}_k$ by

$$(\mathbf{v}_k)_{\mathbf{y}} = \begin{cases} 1_{\mathbf{y}} & \text{if } \mathbf{y} \neq \mathbf{x}_k \\ 1_{\mathbf{x}_k} + r_k \cdot 1_{\mathbf{x}_{m_k}} & \text{if } \mathbf{y} = \mathbf{x}_k \end{cases}$$

and

$$\mathbf{w}_k = \mathbf{t}(\mathbf{x}_j, \mathbf{x}_{m_k}) \bigoplus \mathbf{v}_k \bigoplus \mathbf{t}(\mathbf{x}_{m_k}, \mathbf{x}_j)\,.$$

As before, since $\mathbf{v}_k$ is local and each exchange template has a local decomposition, $\mathbf{w}_k$ has a local decomposition for each $k = j+1, j+2, \ldots, n$. Furthermore, if $\mathbf{b} = \mathbf{a} \bigoplus \mathbf{w}_k$, where $j+1 \leq k \leq n$, then $\mathbf{b}$ is given by

$$\mathbf{b}(\mathbf{y}) = \begin{cases} \mathbf{a}(\mathbf{y}) & \text{if } \mathbf{y} \neq \mathbf{x}_k \\ \mathbf{a}(\mathbf{x}_k) + r_k \cdot \mathbf{a}(\mathbf{x}_j) & \text{if } \mathbf{y} = \mathbf{x}_k\,. \end{cases}$$

Thus for every $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$,

$$\mathbf{a} \bigoplus \mathbf{r}_j = \mathbf{a} \bigoplus (\mathbf{w}_{j+1} \bigoplus \mathbf{w}_{j+2} \bigoplus \cdots \bigoplus \mathbf{w}_{n+1})\,.$$

Therefore, $\mathbf{r}_j = \mathbf{w}_{j+1} \bigoplus \mathbf{w}_{j+2} \bigoplus \cdots \bigoplus \mathbf{w}_{n+1}$ and, hence, $\mathbf{r}_j$ has a local decomposition with respect to *N*.

Q.E.D.

We need one more lemma before proving sufficiency. In the following, let R_j and S_j denote $n \times n$ matrices of the form $\psi(\mathbf{r}_j)$ and $\psi(\mathbf{s}_j)$, respectively, as given in the hypothesis of Lemma 5.9.7.

5.9.8 **Lemma.** *If $A \in M_{n \times n}(\mathbb{F})$ with $n > 1$, then for every $j = 1, 2, \ldots, n-1$ there exist $n \times n$ matrices R_j and S_j of the above form and $n \times n$ permutation matrices P_j and Q_j such that*

$$A = \left[\prod_{j=1}^{n-1} P_j \cdot R_j\right] \cdot \begin{pmatrix} & & & | & 0 \\ & I_{n-1} & & | & \vdots \\ & & & | & 0 \\ - & - & - & + & - \\ 0 & \cdots & 0 & | & c \end{pmatrix} \cdot \left[\prod_{j=1}^{n-1} S_j \cdot Q_j\right]$$

for some $c \in \mathbb{F}$.

Proof: We make use of the following observation made by Tchuente [19]: Given $A \in M_{n\times n}(\mathbb{F})$, then there exist permutation matrices *P* and *Q*, constants $s_1, s_2, \ldots, s_n$ and $r_1, r_2, \ldots, r_n$, and an $(n-1)\times(n-1)$ matrix *C* such that

$$A = P\begin{pmatrix} r_1 & | & r_2 & \cdots & r_n \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & I_{n-1} & \\ 0 & | & & & \end{pmatrix}\begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & C & \\ 0 & | & & & \end{pmatrix}\begin{pmatrix} s_1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ s_2 & | & & & \\ \vdots & | & & I_{n-1} & \\ s_n & | & & & \end{pmatrix}Q \quad \text{(i)}$$

The proof proceeds by induction on *n*. If $n = 2$, then the result follows from Eq. (i).

Now assume that the result holds for $j = 2, \ldots, n-1$. By Eq. (i), there exist permutation matrices P_1 and Q_1, and an $(n-1)\times(n-1)$ matrix *C* such that

$$A = P_1 R_1 \cdot \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & C & \\ 0 & | & & & \end{pmatrix} \cdot S_1 Q_1 \,. \quad \text{(ii)}$$

By induction hypothesis, for every $j = 1, \ldots, n-2$, there exist permutation matrices $\hat{P}_j$, $\hat{Q}_j$, matrices $\hat{R}_j$ and $\hat{S}_j$, and a constant $c \in \mathbb{F}$ such that

$$C = \left[\prod_{j=1}^{n-2} \hat{P}_j \cdot \hat{R}_j\right] \cdot \begin{pmatrix} & & & | & 0 \\ & I_{n-2} & & | & \vdots \\ & & & | & 0 \\ - & - & - & + & - \\ 0 & \cdots & 0 & | & c \end{pmatrix} \cdot \left[\prod_{j=1}^{n-2} \hat{S}_j \cdot \hat{Q}_j\right]. \quad \text{(iii)}$$

For $j = 1, \ldots, n-2$, define

$$P_{j+1} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{P}_j & \\ 0 & | & & & \end{pmatrix}, \quad Q_{j+1} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{Q}_j & \\ 0 & | & & & \end{pmatrix},$$

$$S_{j+1} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{S}_j & \\ 0 & | & & & \end{pmatrix}, \quad and \quad R_{j+1} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{R}_j & \\ 0 & | & & & \end{pmatrix}$$

Then

$$P_{j+1}\cdot R_{j+1} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{P}_j & \\ 0 & | & & & \end{pmatrix} \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{R}_j & \\ 0 & | & & & \end{pmatrix} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{P}_j\cdot\hat{R}_j & \\ 0 & | & & & \end{pmatrix},$$

and

$$S_{j+1}\cdot Q_{j+1} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{S}_j & \\ 0 & | & & & \end{pmatrix} \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{Q}_j & \\ 0 & | & & & \end{pmatrix} = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \hat{S}_j\cdot\hat{Q}_j & \\ 0 & | & & & \end{pmatrix}.$$

Therefore,

$$\prod_{j=2}^{n-1} P_jR_j = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \prod_{j=1}^{n-2}\hat{P}_j\hat{R}_j & \\ 0 & | & & & \end{pmatrix} \text{ and } \prod_{j=2}^{n-1} S_jQ_j = \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \prod_{j=1}^{n-2}\hat{S}_j\hat{Q}_j & \\ 0 & | & & & \end{pmatrix} \qquad \text{(iv)}$$

It now follows from Eqs. (iii) and (iv) that

$$\begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & C & \\ 0 & | & & & \end{pmatrix} =$$

$$\begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \prod_{j=1}^{n-2}\hat{P}_j\hat{R}_j & \\ 0 & | & & & \end{pmatrix} \begin{pmatrix} 1 & | & 0 & \cdots & 0 & & 0 \\ - & + & - & - & - & + & - \\ 0 & | & & & & | & 0 \\ \vdots & | & & I_{n-2} & & | & \vdots \\ 0 & | & & & & | & 0 \\ - & + & - & - & - & + & - \\ 0 & | & 0 & \cdots & 0 & | & c \end{pmatrix} \begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & \prod_{j=1}^{n-2}\hat{S}_j\hat{Q}_j & \\ 0 & | & & & \end{pmatrix}$$

or, equivalently,

$$\begin{pmatrix} 1 & | & 0 & \cdots & 0 \\ - & + & - & - & - \\ 0 & | & & & \\ \vdots & | & & C & \\ 0 & | & & & \end{pmatrix} = \left[\prod_{j=2}^{n-1} P_jR_j\right] \begin{pmatrix} & & & | & 0 \\ & I_{n-1} & & | & \vdots \\ & & & | & 0 \\ - & - & - & + & - \\ 0 & \cdots & 0 & | & c \end{pmatrix} \left[\prod_{j=2}^{n-1} S_jQ_j\right]. \qquad \text{(v)}$$

Substituting Eq. (v) into Eq. (ii) yields

$$A = \left[\prod_{j=1}^{n-1} P_j R_j\right] \begin{pmatrix} & & & | & 0 \\ & I_{n-1} & & | & \vdots \\ & & & | & 0 \\ - & - & - & + & - \\ 0 & \cdots & 0 & | & c \end{pmatrix} \left[\prod_{j=1}^{n-1} S_j Q_j\right].$$

Q.E.D.

We now show that strong connectivity is sufficient in general.

5.9.9 **Theorem.** *If $D(N)$ is strongly connected, then every template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ has a local decomposition with respect to N.*

Proof: Let $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$. According to Lemma 5.9.8,

$$\psi(\mathbf{t}) = \left[\prod_{j=1}^{n-1} P_j R_j\right] \begin{pmatrix} & & & | & 0 \\ & I_{n-1} & & | & \vdots \\ & & & | & 0 \\ - & - & - & + & - \\ 0 & \cdots & 0 & | & c \end{pmatrix} \left[\prod_{j=1}^{n-1} S_j Q_j\right],$$

where P_j and Q_j are permutation matrices, and R_j and S_j are matrices of the form given in Lemma 5.9.7.

Set

$$\mathbf{s} = \psi^{-1}\left[\begin{pmatrix} & & & | & 0 \\ & I_{n-1} & & | & \vdots \\ & & & | & 0 \\ - & - & - & + & - \\ 0 & \cdots & 0 & | & c \end{pmatrix}\right],$$

and for $j = 1, 2, \ldots, n-1$ let

$$\mathbf{r}_j = \psi^{-1}(R_j),\ \mathbf{s}_j = \psi^{-1}(S_j),\ \mathbf{p}_j = \psi^{-1}(P_j),\ \text{and}\ \mathbf{q}_j = \psi^{-1}(Q_j).$$

Since $\psi(\mathbf{s})$ is a diagonal matrix, **s** is local with respect to *N*. By Lemma **5.9.7**, each $\mathbf{r}_j$ and $\mathbf{s}_j$ has a local decomposition with respect to *N*. By Corollary **5.9.6**, each $\mathbf{p}_j$ and $\mathbf{q}_j$ has a local decomposition with respect to *N*. Thus, since

$$\mathbf{t} = \left[\bigoplus_{j=1}^{n-1} (\mathbf{p}_j \oplus \mathbf{r}_j)\right] \oplus \mathbf{s} \oplus \left[\bigoplus_{j=1}^{n-1} (\mathbf{s}_j \oplus \mathbf{q}_j)\right],$$

it follows that **t** has a local decomposition with respect to *N*.

Q.E.D.

In summary, the last sections show how correspondences between directed graphs, neighborhood configurations, network of processors, and matrices can be used to provide necessary and sufficient conditions for the existence of local decompositions of linear transforms. In contrast to earlier sections of this chapter, the existence theorems of this section do not readily suggest methods for constructing efficient algorithms for decomposing templates into local factors. This topic will again be addressed in the next chapter.

Bibliography

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

[2] E.L. Cloud and W. Holsztynski. Higher efficiency for parallel processors. In *Proceedings IEEE Southcon 84*, pages 416–422, Orlando, FL, March 1984.

[3] 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.

[4] P.D. Gader. Necessary and sufficient conditions for the existence of local matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, pages 305–313, July 1988.

[5] G.H. Golub and C.F. van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, 1983.

[6] R. M. Haralick. Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing*, 22:28–38, 1983.

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

[8] R.M. Lougheed and D.L. McCubbrey. The cytocomputer: A practical pipelined image processor. In *Proceedings of the Seventh International Symposium on Computer Architecture*, pages 411–418, 1980.

[9] 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.

[10] Z.Z. Manseur. *Decomposition and Inversion of Convolution Operators*. PhD thesis, University of Florida, 1990.

[11] Z.Z. Manseur and D.C. Wilson. Invertibility of a special class of mean filter operators. *Journal of Mathematical Imaging and Vision*, 1(2):137–144, 1992.

[12] J.H. McClellan and edit. C.M. Rader. *Number Theory in Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

[13] C.L. Nikias, A.P. Chrysafis, and A.N. Venetssanopoulos. The lu decomposition theorem and its implication to the realization of two-dimansional digital filters. *IEEE Transactions on Accoustics Speech and Signal Processing*, COM-26(3):694–711, 1985.

[14] D.P. O'Leary. Some algorithms for approximating convolutions. *Computer Vision Graphics and Image Processing*, 41(3):333–345, 1988.

[15] W. Rheinboldt. *Future Directions in Computational Mathematics, Algorithms, and Software*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1985.

[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] S.R. Sternberg. Overview of image algebra and related issues. In S. Levialdi, editor, *Integrated Technology for Parallel Image Processing*. Academic Press, London, 1985.

[18] M. Tchuente. Parallel calculation of a linear mapping on a computer network. *Linear Algebra and Its Applications*, 28:223–247, 1979.

[19] M. Tchuente. Parallel realization of permutations over trees. *Discrete Math.*, 39:211–214, 1982.

[20] J.S. Wiejak, H. Buxton, and B.F. Buxton. Convolution with separable masks for early image processing. *Computer Vision, Graphics, and Image Processing*, 32:279–290, 1985.

[21] 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.

[22] X. Zhuang and R. M. Haralick. Morphological structuring element decomposition. *Computer Vision, Graphics, and Image Processing*, 35:370–382, 1986.

CHAPTER 6
TECHNIQUES FOR THE COMPUTATION OF THE DISCRETE FOURIER TRANSFORM

In this chapter we examine the discrete version of the Fourier transform. It is no exaggeration to say that the discovery of efficient algorithms for computing the discrete Fourier transform has revolutionized the practice of digital signal processing and digital signal analysis. Collectively, these algorithms are known as *Fast Fourier Transforms* (or *FFT's*) and they are significant for several reasons. They allow for real-time or near real-time spectral analysis and filtering. Efficient computation of many spatial domain computations can be achieved in the frequency domain using fast Fourier transform methods. This is due to the well known result that the convolution of two waveforms in the spatial domain corresponds to the equivalent operation of multiplication of their respective Fourier transforms in the Fourier (or frequency) domain. Just as importantly, the study of methodologies of the various fast Fourier transform algorithms have stimulated a reformulation of many signal processing concepts and algorithms in terms of discrete mathematics. This resulted in a shift away from the notion that signal processing on a digital computer is merely an approximation to analog signal processing techniques. It became evident that discrete signal processing is an important field of investigation in its own right, involving properties and mathematical methods that are exact in the disrete-time domain and independent upon any continuity assumptions. The methods and techniques discussed in this chapter can be considered part of this new field of investigation.

6.1 Linear Separability and the Discrete Fourier Transform

The *Fourier Transform* (FT) was first introduced in Section **3.8** (Example **3.8.13**). Although other transforms are discussed in some detail, we place special emphasis on the Fourier transform because of its wide range of applications in image and signal processing. One of the basic properties of the two-dimensional Fourier transform is its linear separability. This property is a key ingredient in the efficient computation of the discrete version of the two-dimensional Fourier transform. For this reason we begin our discussion with some observations concerning linear separability.

For the remainder of this section $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$ and ψ will denote the ring isomorphism $\psi : \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathrm{M}_{mn \times mn}(\mathbb{F})$ discussed in Section **5.1**. Note that since $\mathbf{X}$ is finite, $S(\mathbf{t_y})$ is finite $\forall\, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ and $\forall\, \mathbf{y} \in \mathbf{X}$. However, because the shape of $S(\mathbf{t_y})$ differs in most cases if **y** is on or near the boundary of **X** from the shape of $S(\mathbf{t_y})$ when **y** is an interior point of **X** which is far removed from the boundary, we need to modify the definition of a row or column template slightly.

6.1.1 **Definition.** If $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}$, then **t** is called a *row template on* **X** if and only if for every $\mathbf{y} \in \mathbf{Y}$, $\exists\, k \in \mathbb{Z}_m$ such that $\mathbf{t_y}(i,j) = 0$ whenever $i \neq k$.

Similarly, **t** is called a *column template on* **X** if and only if for every $\mathbf{y} \in \mathbf{Y}$ $\exists\, l \in \mathbb{Z}_n$ such that$\mathbf{t_y}(i,j) = 0$ whenever $j \neq l$.

Observe that a row (column) template as defined in Definition **5.2.2** also satisfies this definition of a row (column) template when restricted to the finite array **X**. Also, if **t** is both a row and a column template, then **t** is a *one-point* template unless $S(\mathbf{t_y}) = \varnothing\ \forall\, \mathbf{y} \in \mathbf{X}$, in which case **t** is the zero template.

The most common row (or column) templates used by algorithm designers, which are also the templates we are concerned with in this section, are those for which $\mathbf{Y} = \mathbf{X}$ and $\mathbf{t}_{(k,l)}(i,j) = 0$

whenever $i \neq k$ (or $\mathbf{t}_{(k,l)}(i,j) = 0$ whenever $j \neq l$). Therefore, unless otherwise stated all row and column templates considered henceforth are assumed to satisfy these latter conditions.

Recall that if **t** is a row (column) template, then the weight matrix *T* associated with **t** is a row (column) matrix. The isomorphic image $\psi(\mathbf{t})$, however, does not satisfy these conditions.

6.1.2 **Theorem.** *Suppose* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$.

1. If **t** *is a row template on* **X**, *then* $\psi(\mathbf{t})$ *is a block diagonal matrix consisting of* $m \times m$ *blocks and each block is an* $n \times n$ *matrix.*

2. If **t** *is a column template on* **X**, *then* $\psi(\mathbf{t})$ *is a block diagonal matrix consisting of* $m \times m$ *blocks and each block is an* $n \times n$ *diagonal matrix.*

The proof is obvious as it follows directly from the definitions of row and column templates and the definition of the map ψ.

6.1.3 **Example:** If

$$\mathbf{r} = \begin{array}{|c|c|c|}\hline 2 & 1 & 3 \\ \hline \end{array}$$

and $\mathbf{X} = \mathbb{Z}_3 \times \mathbb{Z}_4$, then

$$\psi(\mathbf{r}) = \left(\begin{array}{cccc|cccc|cccc}
1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 3 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3 & 1 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1
\end{array}\right).$$

If

$$\mathbf{s} = \begin{array}{|c|}\hline 2 \\ \hline 1 \\ \hline 3 \\ \hline \end{array}$$

then

$$\psi(\mathbf{s}) = \left(\begin{array}{cccc|cccc|cccc}
1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
\hline
3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \\
\hline
0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 1
\end{array}\right).$$

It follows from Theorem **6.1.2** that the matrix $\psi(\mathbf{t})$ should be highly structured whenever **t** is the product of a row and a column template. For example, if **r** and **s** are as in Example **6.1.3**, then $\psi(\mathbf{t}) = \psi(\mathbf{r} \bigoplus \mathbf{s}) = \psi(\mathbf{r}) \cdot \psi(\mathbf{s})$ has form

$$\psi(\mathbf{r}) \cdot \psi(\mathbf{s}) = \left(\begin{array}{cccc|cccc|cccc}
1 & 2 & 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 1 & 2 & 0 & 6 & 2 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 1 & 2 & 0 & 6 & 2 & 4 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 1 & 0 & 0 & 6 & 2 & 0 & 0 & 0 & 0 \\
\hline
3 & 6 & 0 & 0 & 1 & 2 & 0 & 0 & 2 & 4 & 0 & 0 \\
9 & 3 & 6 & 0 & 3 & 1 & 2 & 0 & 6 & 2 & 4 & 0 \\
0 & 9 & 3 & 6 & 0 & 3 & 1 & 2 & 0 & 6 & 2 & 4 \\
0 & 0 & 9 & 3 & 0 & 0 & 3 & 1 & 0 & 0 & 6 & 2 \\
\hline
0 & 0 & 0 & 0 & 3 & 6 & 0 & 0 & 1 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 9 & 3 & 6 & 0 & 3 & 1 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 9 & 3 & 6 & 0 & 3 & 1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 9 & 3 & 0 & 0 & 3 & 1
\end{array}\right). \qquad (6.1.1)$$

Equation 6.1.1 shows that each block of $\psi(\mathbf{t})$ is a matrix which is void of zero rows or zero columns. More generally, if **r** is a row template on $\mathbf{X} = \mathbb{Z}_m^+ \times \mathbb{Z}_n^+$, **s** is a column template on **X**, and $\mathbf{t} = \mathbf{r} \bigoplus \mathbf{s}$, then

$$\begin{aligned}
\psi(\mathbf{r}) \cdot \psi(\mathbf{s}) &= \begin{pmatrix} R_{11} & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & R_{22} & \cdots & \mathrm{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \mathrm{O} & \cdots & R_{mm} \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1m} \\ S_{21} & S_{22} & \cdots & S_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ S_{m1} & S_{m2} & \cdots & S_{mm} \end{pmatrix} \\
&= \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1m} \\ T_{21} & T_{22} & \cdots & T_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ T_{m1} & T_{m2} & \cdots & T_{mm} \end{pmatrix} = \psi(\mathbf{t}) \,,
\end{aligned}$$

where $T_{ij} = R_{ii}S_{ij}$ $(i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n)$, each S_{ij} is a diagonal $n \times n$ matrix, and O denotes the zero matrix.

Suppose $s_{ij}(k)$ denotes the kth diagonal entry of S_{ij} and $r_{lk}(i)$ denotes the (l, k)th entry of the $n \times n$ matrix R_{ii}. Then the kth column vector

$$\mathbf{v}(i,j)_k = \left(\mathbf{t}_{(j,k)}(i,1),\, \mathbf{t}_{(j,k)}(i,2),\, \ldots,\, \mathbf{t}_{(j,k)}(i,n)\right)'$$

of $T_{ij} = R_{ii}S_{ij}$ can be expressed as

$$\mathbf{v}(i,j)_k = (r_{1k}(i),\, r_{2k}(i),\, \ldots,\, r_{nk}(i))' \cdot s_{ij}(k) \tag{6.1.2}$$

while the kth column vector $\mathbf{v}(i,j')_k$ of $T_{ij'}$ has form

$$\mathbf{v}\big(i,j'\big)_k = (r_{1k}(i),\, r_{2k}(i),\, \ldots,\, r_{nk}(i))' \cdot s_{ij'}(k)\,. \tag{6.1.3}$$

From Eqs. 6.1.2 and 6.1.3 we obtain:

$$\mathbf{v}\big(i,j'\big)_k = \frac{s_{ij'}(k)}{s_{ij}(k)} \cdot \mathbf{v}(i,j)_k\,. \tag{6.1.4}$$

Since j and j' were arbitrary, it follows that the kth column vector of T_{ij} is a scalar product of the kth column vector of T_{i1} for $j = 1, 2, \ldots, m$. Thus, if we form the $n \times m$ matrix A_{ik} by defining the jth column of A_{ik} to be the kth column of T_{ij}, where $j = 1, 2, \ldots, m$, then A_{ik} has rank 1 for all $i = 1, 2, \ldots, m$ and all $k = 1, 2, \ldots, n$. Hence, a necessary condition for linear separability of a template $\mathbf{t}$ on $\mathbf{X}$ is that each matrix $A_{ik} = \Big(a(i,k)_{lj}\Big)_{n\times m}$, where $\Big(a(i,k)_{lj}\Big) = \mathbf{t}_{(j,k)}(i,l)$, has rank 1. This condition is also sufficient.

6.1.4 **Theorem.** *If* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ *where* $\mathbf{X} = \mathbb{Z}_m^+ \times \mathbb{Z}_n^+$*, then* $\mathbf{t}$ *is linearly separable if and only if for* $i = 1, 2, \ldots, m$ *and* $k = 1, 2, \ldots, n$*, each matrix* A_{ik} *has rank 1.*

Proof: We have already shown that if $\mathbf{t}$ is linearly separable, then A_{ik} has rank 1 for $i = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, n$.

In order to prove sufficiency, suppose that A_{ik} has rank 1. Then there exists a column vector $\mathbf{a}(i,k)$ of A_{ik} such that each column vector of A_{ik} is a multiple of $\mathbf{a}(i,k)$. Thus, if $\mathbf{a}(i,k)_j$ denotes the jth column of A_{ik}, then there exists a scalar $s_{ij}(k)$ such that

$$\mathbf{a}(i,k)_j = \mathbf{a}(i,k) \cdot s_{ij}(k)\,. \tag{i}$$

Now, if

$$T_{ij} = \begin{pmatrix} \mathbf{t}_{(j,1)}(i,1) & \mathbf{t}_{(j,2)}(i,1) & \cdots & \mathbf{t}_{(j,n)}(i,1) \\ \mathbf{t}_{(j,1)}(i,2) & \mathbf{t}_{(j,2)}(i,2) & \cdots & \mathbf{t}_{(j,n)}(i,2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{t}_{(j,1)}(i,n) & \mathbf{t}_{(j,2)}(i,n) & \cdots & \mathbf{t}_{(j,n)}(i,n) \end{pmatrix},$$

then by definition of A_{ik},

$$\mathbf{a}(i,k)_j = \left(\mathbf{t}_{(j,k)}(i,1),\, \mathbf{t}_{(j,k)}(i,2),\, \ldots\,,\, \mathbf{t}_{(j,k)}(i,n)\right)'.$$

In view of Eq. (i), we have

$$\left(\mathbf{t}_{(j,k)}(i,1),\, \mathbf{t}_{(j,k)}(i,2),\, \ldots\,,\, \mathbf{t}_{(j,k)}(i,n)\right)' = \mathbf{a}(i,k) \cdot s_{ij}(k)\,. \tag{ii}$$

Define the $n \times n$ matrices R_{ii} and S_{ij} by

$$R_{ii} = (\mathbf{a}(i,1),\, \mathbf{a}(i,2),\, \ldots\,, \mathbf{a}(i,n))$$

and

$$S_{ij} = \begin{pmatrix} s_{ij}(1) & 0 & \cdots & 0 \\ 0 & s_{ij}(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{ij}(n) \end{pmatrix}.$$

Hence, by Eq. (ii),

$$T_{ij} = R_{ii} \cdot S_{ij} \;\text{ for }\; i = 1, \ldots\,, m \;\text{ and }\; j = 1, \ldots\,, m\,.$$

Let

$$T = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1m} \\ T_{21} & T_{22} & \cdots & T_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ T_{m1} & T_{m2} & \cdots & T_{mm} \end{pmatrix},\; R = \begin{pmatrix} R_{11} & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & R_{22} & \cdots & \mathrm{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \mathrm{O} & \cdots & R_{mm} \end{pmatrix},$$

and

$$S = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1m} \\ S_{21} & S_{22} & \cdots & S_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ S_{m1} & S_{m2} & \cdots & S_{mm} \end{pmatrix}.$$

Then $\psi(\mathbf{t}) = T = R \cdot S$ and, therefore, $\mathbf{t} = \psi^{-1}(R) \bigoplus \psi^{-1}(S)$.

Q.E.D.

If **t** is a template on **X** such that each of the matrices A_{ik} has rank less or equal to some positive integer *p*, then an argument analogous to the one given in the proof of Theorem **6.1.4** can be used to show that each column of A_{ik} is a linear combination of $q \leq p$ column vectors $\left(\mathbf{a}^1(i,k),\, \mathbf{a}^2(i,k),\, \ldots\,, \mathbf{a}^q(i,k)\right)$. If we continue to reason along the same line as in the proof of the theorem, we obtain the following corollary:

6.1.5 **Corollary.** *If* $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$, *then* **t** *can be written as the sum of* $q \leq p$ *separable templates if and only if for* $i = 1, 2, \ldots\,, m$ *and* $k = 1, 2, \ldots\,, n$, *each matrix* A_{ik} *has rank less or equal to p.*

In Section **3.8** (Example **3.8.13**) we introduced the Fourier transform $\mathcal{F}(f) = \hat{f}$ of a function $f \in C(\mathbb{R}^1)$. Here we reformulate the definition of the Fourier transform by defining $\mathcal{F}(f) = \hat{f}$ in terms of the equation

$$\hat{f}(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi iux} dx\,. \tag{6.1.6}$$

Given $\hat{f}$, then *f* can be recovered by using the inverse Fourier transform which is given by the equation

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(u) e^{2\pi ixu} du\,. \tag{6.1.7}$$

Equations 6.1.6 and 6.1.7 are called the *Fourier transform pair*. This transform pair exists if *f* and $\hat{f}$ satisfy the conditions given in Example **3.8.13**.

The reader must be warned that there is no single customary definition of the Fourier transform and its inverse. The variations arise in the treatment of the factor $\frac{1}{2\pi}$ and the sign of the exponential. For instance, in Example **3.8.13**, the term 2π is missing in the exponent. However, for each definition of the Fourier transform, there is only one reasonable definition of its inverse. The various definitions have the form

$$\hat{f}(u) = \alpha \int_{-\infty}^{\infty} f(x) e^{i\beta ux} dx \tag{6.1.8}$$

for some constants α and β, with the corresponding inverse transform being

$$f(x) = \frac{|\beta|}{2\pi\alpha} \int_{-\infty}^{\infty} \hat{f}(u) e^{-i\beta xu} du\,. \tag{6.1.9}$$

We have taken $\alpha = 1$ and $\beta = -2\pi$. Other definitions use various combinations of $\alpha = 1$, $\alpha = \frac{1}{2\pi}$, $\alpha = \frac{1}{\sqrt{2\pi}}$, $\beta = \pm 1$, $\beta = \pm 2\pi$. The factor $|\beta|/2\pi\alpha$ is selected so that the inverse Fourier transform of *f* is actually equal to *f* for a general class of functions.

Suppose $\mathbf{a} : \mathbb{Z}_n \to \mathbb{R}$. The *Discrete Fourier Transform*, abbreviated as DFT, of **a** is defined as

$$\hat{\mathbf{a}}(j) = \sum_{k=0}^{n-1} \mathbf{a}(k) e^{-2\pi jki/n}\,, \tag{6.1.10}$$

where $j = 0, 1, \ldots, n-1$. The *Discrete Inverse Fourier Transform*, or DIFT, is given by

$$\mathbf{a}(k) = \frac{1}{n} \sum_{j=0}^{n-1} \hat{\mathbf{a}}(j) e^{2\pi kji/n}\,, \tag{6.1.11}$$

where $k = 0, 1, \ldots, n-1$.

In digital signal processing, **a** is usually viewed as having been obtained from a continuous function $f \in C(\mathbb{R}^1)$ by sampling *f* at some finite number of points $\{x_0, x_1, \ldots, x_{n-1}\} \subset \mathbb{R}$ and setting

$\mathbf{a}(k) = f(x_k)$ so that Eqs. 6.1.10 and 6.1.11 represent the finite analogues of Eqs. 6.1.6 and 6.1.7, respectively. As we are only interested in the computational aspects of the DFT and its inverse, we view $\mathbf{a}$ as a vector of finite length

$$\mathbf{a} = (\mathbf{a}(0), \mathbf{a}(1), \ldots, \mathbf{a}(n-1)).$$

Defining the template $\mathbf{f} \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$, where $\mathbf{X} = \mathbb{Z}_n$, by

$$\mathbf{f}_j(k) = e^{-2\pi kji/n}, \tag{6.1.12}$$

we obtain equivalent image algebra formulation

$$\hat{\mathbf{a}} = \mathbf{a} \bigoplus \mathbf{f} \tag{6.1.13}$$

of Eq. 6.1.10.

The template $\mathbf{f}$ defined by Eq. 6.1.12 is called the *one-dimensional Fourier template*. It follows directly from the definition of $\mathbf{f}$ that $\mathbf{f}' = \mathbf{f}$ and $(\mathbf{f}^*)' = \mathbf{f}^*$, where $\mathbf{f}^*$ denotes the complex conjugate of $\mathbf{f}$ defined by $\mathbf{f}_j^*(k) = (\mathbf{f}_j(k))^* = e^{2\pi kji/n}$. Hence, the image algebra equivalent of the DIFT is given by

$$\mathbf{a} = \frac{1}{n}(\hat{\mathbf{a}} \bigoplus \mathbf{f}^*). \tag{6.1.14}$$

The notion of the Fourier transform and its inverse can be easily extended to functions $f \in C\left(\mathbb{R}^2\right)$. In particular, the *two-dimensional* Fourier transform pair is given by

$$\hat{f}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(ux+vy)}dxdy \tag{6.1.15}$$

and

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(u,v)e^{2\pi i(xu+yv)}dudv. \tag{6.1.16}$$

For discrete functions $\mathbf{a} : \mathbb{Z}_m \times \mathbb{Z}_n \to \mathbb{R}$ we have the two-dimensional discrete Fourier transform

$$\hat{\mathbf{a}}(u,v) = \sum_{k=0}^{n-1}\sum_{j=0}^{m-1} \mathbf{a}(j,k)e^{-2\pi i\left(j\frac{u}{m}+k\frac{v}{n}\right)}, \tag{6.1.17}$$

with the inverse transform specified by

$$\mathbf{a}(j,k) = \frac{1}{nm}\sum_{v=0}^{n-1}\sum_{u=0}^{m-1} \hat{\mathbf{a}}(u,v)e^{2\pi i\left(u\frac{j}{m}+v\frac{k}{n}\right)}. \tag{6.1.18}$$

The image algebra equivalent formulation of the transform pair 6.1.17 and 6.1.18 is given by

$$\hat{\mathbf{a}} = \mathbf{a} \bigoplus \mathbf{f} \tag{6.1.19}$$

and

$$\mathbf{a} = \frac{1}{nm}(\hat{\mathbf{a}} \bigoplus \mathbf{f}^*), \tag{6.1.20}$$

where $\mathbf{f} \in \left(\mathbb{C}^{\mathbf{X}}\right)^{\mathbf{X}}$ is defined by

$$\mathbf{f}_{(u,v)}(j,k) = e^{-2\pi i\left(j\frac{u}{m}+k\frac{v}{n}\right)}, \tag{6.1.21}$$

and $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$.

The *two-dimensional Fourier template* $\mathbf{f}$ defined by Eq. 6.1.21 satisfies the conditions of Theorem **6.1.4**.

6.1.6 **Example:** Suppose $\mathbf{X} = \mathbb{Z}_2 \times \mathbb{Z}_3$. Then

$$\mathbf{f}_{(u,v)}(j,k) = e^{-2\pi i\left(j\frac{u}{2}+k\frac{v}{3}\right)},$$

where $0 \leq j,\, u \leq 1$ and $0 \leq k,\, v \leq 2$. The matrix $F = \psi(\mathbf{f})$, called the *Fourier matrix* corresponding to **f**, is given by

$$\psi(\mathbf{f}) = F = \begin{pmatrix} 1 & 1 & 1 & | & 1 & 1 & 1 \\ 1 & e^{-\frac{2}{3}\pi i} & e^{-\frac{4}{3}\pi i} & | & 1 & e^{-\frac{2}{3}\pi i} & e^{-\frac{4}{3}\pi i} \\ 1 & e^{-\frac{4}{3}\pi i} & e^{-\frac{2}{3}\pi i} & | & 1 & e^{-\frac{4}{3}\pi i} & e^{-\frac{2}{3}\pi i} \\ - & - & - & + & - & - & - \\ 1 & 1 & 1 & | & -1 & -1 & -1 \\ 1 & e^{-\frac{2}{3}\pi i} & e^{-\frac{4}{3}\pi i} & | & -1 & -e^{-\frac{2}{3}\pi i} & -e^{-\frac{4}{3}\pi i} \\ 1 & e^{-\frac{4}{3}\pi i} & e^{-\frac{2}{3}\pi i} & | & -1 & -e^{-\frac{4}{3}\pi i} & -e^{-\frac{2}{3}\pi i} \end{pmatrix} = \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix}.$$

The matrices A_{ik} derived from F are given by

$$A_{11} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix},\ A_{12} = \begin{pmatrix} 1 & 1 \\ e^{-\frac{2}{3}\pi i} & e^{-\frac{2}{3}\pi i} \\ e^{-\frac{4}{3}\pi i} & e^{-\frac{4}{3}\pi i} \end{pmatrix},\ A_{13} = \begin{pmatrix} 1 & 1 \\ e^{-\frac{4}{3}\pi i} & e^{-\frac{4}{3}\pi i} \\ e^{-\frac{2}{3}\pi i} & e^{-\frac{2}{3}\pi i} \end{pmatrix},$$
$$A_{21} = \begin{pmatrix} 1 & -1 \\ 1 & -1 \\ 1 & -1 \end{pmatrix},\ A_{22} = \begin{pmatrix} 1 & -1 \\ e^{-\frac{2}{3}\pi i} & -e^{-\frac{2}{3}\pi i} \\ e^{-\frac{4}{3}\pi i} & -e^{-\frac{4}{3}\pi i} \end{pmatrix},\ A_{13} = \begin{pmatrix} 1 & -1 \\ e^{-\frac{4}{3}\pi i} & -e^{-\frac{4}{3}\pi i} \\ e^{-\frac{2}{3}\pi i} & -e^{-\frac{2}{3}\pi i} \end{pmatrix}.$$

Clearly, each A_{ik} has rank 1. Choosing

$$\mathbf{a}(1,1) = \mathbf{a}(2,1) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},\ \mathbf{a}(1,2) = \mathbf{a}(2,2) = \begin{pmatrix} 1 \\ e^{-\frac{2}{3}\pi} \\ e^{-\frac{4}{3}\pi} \end{pmatrix},$$

and

$$\mathbf{a}(1,3) = \mathbf{a}(2,3) = \begin{pmatrix} 1 \\ e^{-\frac{4}{3}\pi} \\ e^{-\frac{2}{3}\pi} \end{pmatrix},$$

then $s_{ij}(k) = 1$ for $i,\, j = 1,\, 2,\ k = 1,\, 2,\, 3$ as long as $i \neq 2$ whenever $i = j$, and $s_{22}(k) = -1$ for $k = 1, 2, 3$.

Hence $S_{22} = -I_3$ and $S_{ij} = I_3$ in the remaining cases.

Also,

$$R_{11} = R_{22} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{-\frac{2}{3}\pi} & e^{-\frac{4}{3}\pi} \\ 1 & e^{-\frac{4}{3}\pi} & e^{-\frac{2}{3}\pi} \end{pmatrix}.$$

According to the proof of Theorem **6.1.4**

$$\begin{aligned} F &= \begin{pmatrix} R_{11} & \mathrm{O} \\ \mathrm{O} & R_{22} \end{pmatrix} \cdot \begin{pmatrix} S_{11} & \mathrm{S}_{12} \\ \mathrm{S}_{21} & S_{22} \end{pmatrix} \\ &= \begin{pmatrix} R_{11} & \mathrm{O} \\ \mathrm{O} & R_{22} \end{pmatrix} \cdot \begin{pmatrix} I_3 & I_3 \\ I_3 & -I_3 \end{pmatrix} = R \cdot S. \end{aligned}$$

Setting $\mathbf{r} = \psi^{-1}(R)$ and $\mathbf{s} = \psi^{-1}(S)$ we obtain $\mathbf{f} = \mathbf{r} \bigoplus \mathbf{s}$, where

$$\mathbf{s}_{(0,v)} = \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline \end{array}\,, \qquad \mathbf{s}_{(1,v)} = \begin{array}{|c|} \hline 1 \\ \hline -1 \\ \hline \end{array}\,,$$

and $\mathbf{r}_{(u,v)}(j,k) = e^{-\frac{2}{3}i\pi vk}$.

In the above example, the row template $\mathbf{r}$ does not depend on the variables u and j and is, therefore, equivalent to the one-dimensional Fourier template $\mathbf{r}_v(k) = e^{-\frac{2i\pi}{n}vk}$ $(n = 3)$. This holds in general and follows from the law of exponents; i.e., $e^{x+y} = e^x \cdot e^y$. More precisely, since

$$\mathbf{f}_{(u,v)}(j,k) = e^{-2\pi i\left(\frac{u}{m}j+\frac{v}{n}k\right)} = e^{-2\pi i\frac{v}{n}k} \cdot e^{-2\pi i\frac{u}{m}j} = \mathbf{r}_v(k) \cdot \mathbf{s}_u(j),$$

the template $\mathbf{t}$ can always be written as the product of a row and a column template, namely $\mathbf{f} = \mathbf{r} \bigoplus \mathbf{s}$. Therefore,

$$\hat{\mathbf{a}} = (\mathbf{a} \bigoplus \mathbf{r}) \bigoplus \mathbf{s}\,. \tag{6.1.22}$$

Equation 6.1.22 says that the two-dimensional Fourier transform can be computed in two steps by successive applications of the one-dimensional Fourier transform; first along each row of **a** followed by a one-dimensional Fourier transform along the columns of the result. Observe that the roles of **r** and **s** can be reversed; the same result can be obtained by first taking the one-dimensional Fourier transform along the columns of **a** and then along the rows of the result.

As a final observation we note that Eq. 6.1.20 implies that

$$\mathbf{a} = \frac{1}{nm}(\hat{\mathbf{a}}^* \bigoplus \mathbf{f})^*\,. \tag{6.1.23}$$

This means that the inverse Fourier transform can be computed in terms of the Fourier transform and simple conjugation.

6.2 Block Structured Matrices and the Fourier Matrix

The isomorphism ψ maps many commonly used templates to matrices that display highly regular structures. Many of these highly regular matrices can be subdivided into *blocks* of submatrices that exhibit similarities or common properties that prove very useful in transform optimization. The Fourier matrix is a particularly elegant and useful example of a block structured matrix. Due to the particular block structure and intrinsic properties of the Fourier matrix, block structured matrices arise quite naturally in the analysis and optimization of the discrete Fourier transform.

In this section we digress temporarily to study a few concepts from the theory of block matrices and investigate some properties of the Fourier matrix.

Column and row partitioning of a matrix are examples of matrix blocking. The notation

$$A = \begin{pmatrix} A_{00} & A_{01} & \cdots & A_{0,n-1} \\ A_{10} & A_{11} & \cdots & A_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m-1,0} & A_{m-1,1} & \cdots & A_{m-1,n-1} \end{pmatrix}$$

is used to denote an $m \times n$ *block matrix*, where each block A_{ij} is an $m_i \times n_j$ matrix of scalars. The special case where $m = 1$ or $n = 1$ provides for row and column block vectors.

The manipulation of block matrices is analogous to the manipulation of scalar matrices. For example, if $A = (A_{ij})$ is a $m \times p$ block matrix and $B = (B_{ij})$ is a $p \times n$ block matrix, then the matrix $C = A \cdot B$ is the $m \times n$ block matrix $C = (C_{ij})$ defined by

$$C_{ij} = \sum_{k=0}^{p-1} A_{ik} \cdot B_{kj} \, ,$$

where the individual matrix pairs A_{ik} , B_{kj} are assumed to have dimensions such that the product $A_{ik} \cdot B_{kj}$ exists.

One of the simplest type of block matrices are block diagonal matrices. Recall that if $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{C}^n$, then the matrix $A = diag(\mathbf{a}) = diag(a_1, a_2, \ldots, a_{n-1})$ is the $n \times n$ diagonal matrix with diagonal entries $a_0, a_1, \ldots, a_{n-1}$. This concept generalizes to block vectors. If $(A_0, A_1, \ldots, A_{k-1})$ is a block vector such that each $A_i \in M_{n \times n}(\mathbb{C})$, then $A = diag(A_0, A_1, \ldots, A_{k-1})$ is the $kn \times kn$ block diagonal matrix

$$A = \begin{pmatrix} A_0 & 0 & \cdots & 0 \\ 0 & A_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{k-1} \end{pmatrix}.$$

This should not be confused with $diag(A)$ when $A \in M_{n \times n}(\mathbb{C})$, in which case $diag(A)$ is a *vector* of scalars given by $diag(A) = (a_{00}, a_{11}, \ldots, a_{n-1,n-1})$.

A key role in the theory of block matrices is played by the tensor product and we shall make heavy use of it in subsequent sections. For example, if I_n denotes the $n \times n$ identity matrix, then $I_n \otimes A$ is the block diagonal matrix

$$I_n \otimes A = \begin{pmatrix} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A \end{pmatrix}.$$

More generally, we have

$$I_k \otimes (I_n \otimes A) = I_{kn} \otimes A. \tag{6.2.1}$$

This follows from the even more general property

$$(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D), \tag{6.2.2}$$

where we assume that the ordinary matrix products are defined.