Appendices

A. Minimal Filtering Algorithms

In this supplemental material we will first illustrate how minimal filtering algorithms are derived, using $F(4,3)$ as an example, which is more complex than the typical textbook example. The polynomial math gets fairly tedious for large filters, so we used Mathematica to perform the symbolic math.

Then in the following sections, we give transforms for the algorithms that were mentioned in the paper but not explicitly detailed due to space limitations.

A.1. F(4,3)

Recall the minimal filtering algorithm $F(4\times 4, 3\times 3)$ is created by nesting 1D algorithm $F(4,3)$ with itself. In this section we derive $F(4,3)$ using the Chinese Remainder Theorem technique pioneered by Winograd, using a notation similar to [2, p. 155].

The 3 element filter g and 4 element signal d can be represented as polynomials,

$$\begin{aligned} g(x) &= g_2x^2 + g_1x + g_0 \\ d(x) &= d_3x^3 + d_2x^2 + d_1x + d_0 \end{aligned} \tag{24}$$

and the linear convolution $g * d$ is equal to the coefficients of the polynomial product

$$y(x) = g(x)d(x) \tag{25}$$

For polynomial $m(x)$ of degree 6, this is equal to

$$y(x) = g(x)d(x) \mod m(x) \tag{26}$$

If instead $m(x)$ is of degree 5, we can write

$$y(x) = g(x)d(x) \mod m(x) + R_{m(x)}[y(x)] \tag{27}$$

where $R_m(x)[y(x)]$ is the remainder of $y(x)$ divided by $m(x)$.

We choose

$$\begin{aligned} m(x) &= m^{(0)}(x)m^{(1)}(x)m^{(2)}(x)m^{(3)}(x)m^{(4)}(x)m^{(5)}(x) \\ &= x(x-1)(x+1)(x-2)(x+2)(x-\infty) \end{aligned} \tag{28}$$

which uses the popular convention of writing $x-\infty$ in place of $R_{m(x)}[y(x)]$.

The residues of $g(x)$ and $d(x)$ with respect to $m^{(i)}(x)$ are

$$\begin{aligned} g^{(0)}(x) &= g(x) \mod m^{(0)} = g_0 \\ g^{(1)}(x) &= g(x) \mod m^{(1)} = g_0 + g_1 + g_2 \\ g^{(2)}(x) &= g(x) \mod m^{(2)} = g_0 - g_1 + g_2 \\ g^{(3)}(x) &= g(x) \mod m^{(3)} = g_0 + 2g_1 + 4g_2 \\ g^{(4)}(x) &= g(x) \mod m^{(4)} = g_0 - 2g_1 + 4g_2 \\ g^{(5)}(x) &= g(x) \mod m^{(5)} = g_2 \end{aligned} \tag{29}$$

and

$$\begin{aligned} d^{(0)}(x) &= d(x) \mod m^{(0)} = d_0 \\ d^{(1)}(x) &= d(x) \mod m^{(1)} = d_0 + d_1 + d_2 + d_3 \\ d^{(2)}(x) &= d(x) \mod m^{(2)} = d_0 - d_1 + d_2 - d_3 \\ d^{(3)}(x) &= d(x) \mod m^{(3)} = d_0 + 2d_1 + 4d_2 + 8d_3 \\ d^{(4)}(x) &= d(x) \mod m^{(4)} = d_0 - 2d_1 + 4d_2 - 8d_3 \\ d^{(5)}(x) &= d(x) \mod m^{(5)} = d_3 \end{aligned} \tag{30}$$

We can represent the residues $d^{(i)}(x)$ in matrix form as

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 2 & 4 & 8 \\ 1 & -2 & 4 & -8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

Define $M^{(i)}(x) = m(x)/m^{(i)}(x)$, yielding:

$$\begin{aligned} M^{(0)}(x) &= x^4 - 5x^2 + 4 \\ M^{(1)}(x) &= x^4 + x^3 - 4x^2 - 4x \\ M^{(2)}(x) &= x^4 - x^3 - 4x^2 + 4x \\ M^{(3)}(x) &= x^4 + 2x^3 - x^2 - 2x \\ M^{(4)}(x) &= x^4 - 2x^3 - x^2 + 2x \\ m(x) &= x^5 - 5x^3 + 4x \end{aligned} \tag{32}$$

Construct the matrix B such that column B_i is the coefficients of $M^{(i-1)}$ and column B_6 is the coefficients of m, yielding:

$$B = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4 & 4 & -2 & 2 & 4 \\ -5 & -4 & -4 & -1 & -1 & 0 \\ 0 & 1 & -1 & 2 & -2 & -5 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

In order to apply the Chinese Remainder Theorem, we must solve for $n^{(i)}(x)$, $N^{(i)}(x)$, such that

$$n^{(i)}(x)m^{(i)}(x) + n^{(i)}(x)M^{(i)}(x) = 1 \tag{34}$$

yielding:

$$
\begin{aligned}
n^{(0)}(x) &= \frac{1}{4}\left(5x - x^3\right) \\
n^{(1)}(x) &= \frac{1}{6}\left(x^3 + 2x^2 - 2x - 6\right) \\
n^{(2)}(x) &= \frac{1}{6}\left(x^3 - 2x^2 - 2x + 6\right) \\
n^{(3)}(x) &= \frac{1}{24}\left(-x^3 - 4x^2 - 7x - 12\right) \\
n^{(4)}(x) &= \frac{1}{24}\left(-x^3 + 4x^2 - 7x + 12\right) \\
N^{(0)}(x) &= \frac{1}{4} \\
N^{(1)}(x) &= -\frac{1}{6} \\
N^{(2)}(x) &= -\frac{1}{6} \\
N^{(3)}(x) &= \frac{1}{24} \\
N^{(4)}(x) &= \frac{1}{24}
\end{aligned}
\tag{35}
$$

Matrix G is constructed by setting row G_i equal to the coefficients of $g^{(i-1)}$ multiplied by $N^{(i-1)}$:

$$
G = \begin{bmatrix}
\frac{1}{4} & 0 & 0 \\
-\frac{1}{6} & -\frac{1}{6} & -\frac{1}{6} \\
-\frac{1}{6} & \frac{1}{6} & -\frac{1}{6} \\
\frac{1}{24} & \frac{1}{12} & \frac{1}{6} \\
\frac{1}{24} & -\frac{1}{12} & \frac{1}{6} \\
0 & 0 & 1
\end{bmatrix}
\tag{36}
$$

A.2. F(3x3,3x3)

A minimal algorithm for $F(3,3)$ uses the matrices:

$$
\begin{aligned}
B^T &= \begin{bmatrix}
2 & -1 & -2 & 1 & 0 \\
0 & -2 & -1 & 1 & 0 \\
0 & 2 & -3 & 1 & 0 \\
0 & -1 & 0 & 1 & 0 \\
0 & 2 & -1 & -2 & 1
\end{bmatrix} \\
G &= \begin{bmatrix}
\frac{1}{2} & 0 & 0 \\
-\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
-\frac{1}{6} & \frac{1}{6} & -\frac{1}{6} \\
\frac{1}{6} & \frac{1}{3} & \frac{2}{3} \\
0 & 0 & 1
\end{bmatrix} \\
A^T &= \begin{bmatrix}
1 & 1 & 1 & 1 & 0 \\
0 & 1 & -1 & 2 & 0 \\
0 & 1 & 1 & 4 & 1
\end{bmatrix}
\end{aligned}
\tag{37}
$$

The algorithm uses 5 multiplications, while standard algorithm uses $3 \times 3 = 9$. The data transform uses 9 floating point instructions, the filter transform uses 7, and the inverse transform uses 7.

Applying the nesting formula (8) yields a minimal algorithm for $F(3 \times 3, 3 \times 3)$ using $5 \times 5 = 25$ multiplies. The standard algorithm uses $3 \times 3 \times 3 \times 3$ = 81 multiplies. This is an arithmetic complexity reduction of $3.24X$.

The data transform for the 2D algorithm uses $9(5+5) = 90$ floating point instructions, the filter transform uses $7(5+3) = 56$, and the inverse transform uses $7(5+3) = 56$.

A.3. F(6x6,3x3)

$F(6 \times 6, 3 \times 3)$, has quite large coefficients.

$$
\begin{aligned}
B^T &= \begin{bmatrix}
-36 & 0 & 49 & 0 & -14 & 0 & 1 & 0 \\
0 & 36 & 36 & -13 & -13 & 1 & 1 & 0 \\
0 & -36 & 36 & 13 & -13 & -1 & 1 & 0 \\
0 & 18 & 9 & -20 & -10 & 2 & 1 & 0 \\
0 & -18 & 9 & 20 & -10 & -2 & 1 & 0 \\
0 & 12 & 4 & -15 & -5 & 3 & 1 & 0 \\
0 & -12 & 4 & 15 & -5 & -3 & 1 & 0 \\
0 & -36 & 0 & 49 & 0 & -14 & 0 & 1
\end{bmatrix} \\
G &= \begin{bmatrix}
-\frac{1}{36} & 0 & 0 \\
\frac{1}{48} & \frac{1}{48} & \frac{1}{48} \\
\frac{1}{48} & -\frac{1}{48} & \frac{1}{48} \\
-\frac{1}{120} & -\frac{1}{60} & -\frac{1}{30} \\
-\frac{1}{120} & \frac{1}{60} & -\frac{1}{30} \\
\frac{1}{720} & \frac{1}{240} & \frac{1}{80} \\
\frac{1}{720} & -\frac{1}{240} & \frac{1}{80} \\
0 & 0 & 1
\end{bmatrix} \\
A^T &= \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & -1 & 2 & -2 & 3 & -3 & 0 \\
0 & 1 & 1 & 4 & 4 & 9 & 9 & 0 \\
0 & 1 & -1 & 8 & -8 & 27 & -27 & 0 \\
0 & 1 & 1 & 16 & 16 & 81 & 81 & 0 \\
0 & 1 & -1 & 32 & -32 & 243 & -243 & 1
\end{bmatrix}
\end{aligned}
\tag{38}
$$

This minimal algorithm for $F(6,3)$ uses 8 multiplies. The data transform uses 26 floating point instructions, the filter transform uses 13, and the inverse transform uses 20.

By nesting, $F(6 \times 6, 3 \times 3)$ uses $8 \times 8 = 64$ multiplies, while the standard algorithm uses $6 \times 6 \times 3 \times 3 = 324$. This is an arithmetic complexity reduction of $5.06X$.

The 2D data transform uses $26(8+8) = 416$ floating point instructions, the filter transform uses $13(3+8) = 143$, and the inverse transform uses $20(8+6) = 280$.