Parallel Computation of 2-D Discrete Convolutions

Andrew P. Papliński, Member, IEEE
Department of Robotics and Digital Technology, Monash University

October 20, 1995

Abstract

A 2-D discrete convolution is a linear operation and as such can be expressed as a product of two matrices, namely, the left and right convolution matrices, respectively. Such a decomposition provides a way of parallelising computation of a 2-D discrete convolution. The general forms of the left and right convolution matrices and related memory reference mechanisms have been considered.

EDICS number: SPL.IP.1.2

Author's address: Department of Robotics and Digital Technology
Monash University
Clayton 3168
Australia
Fax: +61 3 9905 3574
E-mail: app@rdt.monash.edu.au

1 A fully parallel algorithm

The convolution of a pair of two-dimensional discrete functions, $f(\mathbf{x})$ and $h(\mathbf{x})$ is a linear operation defined as [1, 2]:

$$f(\mathbf{x}) \odot h(\mathbf{x}) = \sum_{\mathbf{u}} f(\mathbf{u}) h(\mathbf{x} - \mathbf{u}) \tag{1}$$

Given an $R \times C$ image F: $\overset{C}{\underset{R}{\boxed{\mathbf{F}}}}$ and an $r \times c$ filter H: $\overset{c}{\underset{r}{\boxed{\mathbf{H}}}}$, the 2-D discrete convolution can be expressed as a product of two specially formed block-matrices, called the **left convolution matrix**, lcm, and the **right convolution matrix**, rcm, as follows

$$F \odot H = \mathrm{lcm}(F, r) \cdot \mathrm{rcm}(H, C) = \mathrm{lcm}(H, R) \cdot \mathrm{rcm}(F, c) \tag{2}$$

The **left convolution matrix** $\mathrm{lcm}(F, r)$ is composed of the image matrix F replicated r times, each block shifted down by one row in the following way:

$$\mathrm{lcm}(F,r) \;=\; \overset{\longleftarrow\; r \cdot C \;\longrightarrow}{\begin{array}{|ccccc|}\hline \multirow{2}{*}{\mathbf{F}} & 0 & 0 & \cdots & 0 \\ & \multirow{2}{*}{\mathbf{F}} & 0 & \cdots & 0 \\ & & \multirow{2}{*}{$\cdots$} & 0 & 0 \\ 0 & & & \multirow{2}{*}{\mathbf{F}} & 0 \\ 0 & 0 & & & \multirow{2}{*}{\mathbf{F}} \\ 0 & \cdots & 0 & & \\ 0 & \cdots & 0 & 0 & \\ \hline \end{array}} \begin{array}{l} \uparrow \\ r-1 \\ \downarrow \\ \uparrow \\ R \\ \downarrow \end{array} \tag{3}$$

The size of the left convolution matrix is $(r \cdot C) \times (R + r - 1)$.

The **right convolution matrix**, $\mathrm{rcm}(H, C)$, consists of one-dimensional convolution matrices of order C formed from rows of the filter H:

$$\mathrm{rcm}(H,C) \;=\; \overset{\longleftarrow\; C+c-1 \;\longrightarrow}{\begin{bmatrix} \langle H(1,:)\rangle_C \\ \langle H(2,:)\rangle_C \\ \cdots \\ \langle H(r,:)\rangle_C \end{bmatrix}} \begin{array}{l} \uparrow \\ r \cdot C \\ \downarrow \end{array} \tag{4}$$

where $H(m,:)$ denotes the m-th row of the matrix H, and the angle brackets denote the 1-D convolution matrix. For a row vector $\mathbf{h}$ of length c, the one-dimensional convolution matrix of order C, also known as the **Sylvester resultant matrix** [3, **?**], is formed from the shifted vector $\mathbf{h}$ in the following way:

$$\langle \mathbf{h} \rangle_C \;=\; \begin{array}{l} \begin{bmatrix} h_1 & h_2 & \cdots & h_c & & \\ & h_1 & h_2 & \cdots & h_c & \mathbf{0} \\ & \mathbf{0} & \ddots & \ddots & & \ddots \\ & & & h_1 & h_2 & \cdots & h_c \end{bmatrix} \begin{array}{l} \uparrow \\ C \\ \downarrow \end{array} \\ \overset{\longleftarrow\; C+c-1 \;\longrightarrow}{} \end{array} \tag{5}$$

Therefore, the right convolution matrix is of size $(r \cdot C) \times (C + c - 1)$ and the 2-D convolution matrix will have size $(R + r - 1) \times (C + c - 1)$.

The proof of equation (2) follows an illustrative example.

Example

The calculations were carried out using MATLAB [4].

R = 5 ; C = 4 ; r = 3 ; c = 2;

$$\mathsf{F} = \begin{matrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \\ 51 & 52 & 53 & 54 \end{matrix} ; \quad \mathsf{H} = \begin{matrix} 0.11 & 0.12 \\ 0.21 & 0.22 \\ 0.31 & 0.32 \end{matrix} ;$$

$$\mathsf{lcmF} = \begin{matrix} 11 & 12 & 13 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 21 & 22 & 23 & 24 & 11 & 12 & 13 & 14 & 0 & 0 & 0 & 0 \\ 31 & 32 & 33 & 34 & 21 & 22 & 23 & 24 & 11 & 12 & 13 & 14 \\ 41 & 42 & 43 & 44 & 31 & 32 & 33 & 34 & 21 & 22 & 23 & 24 \\ 51 & 52 & 53 & 54 & 41 & 42 & 43 & 44 & 31 & 32 & 33 & 34 \\ 0 & 0 & 0 & 0 & 51 & 52 & 53 & 54 & 41 & 42 & 43 & 44 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 51 & 52 & 53 & 54 \end{matrix}$$

rcmH = [convmtx(H(1, :), C); convmtx(H(2, :), C); convmtx(H(3, :), C)]

$$= \begin{matrix} 0.11 & 0.12 & 0 & 0 & 0 \\ 0 & 0.11 & 0.12 & 0 & 0 \\ 0 & 0 & 0.11 & 0.12 & 0 \\ 0 & 0 & 0 & 0.11 & 0.12 \end{matrix}$$

$$\begin{matrix} 0.21 & 0.22 & 0 & 0 & 0 \\ 0 & 0.21 & 0.22 & 0 & 0 \\ 0 & 0 & 0.21 & 0.22 & 0 \\ 0 & 0 & 0 & 0.21 & 0.22 \end{matrix}$$

$$\begin{matrix} 0.31 & 0.32 & 0 & 0 & 0 \\ 0 & 0.31 & 0.32 & 0 & 0 \\ 0 & 0 & 0.31 & 0.32 & 0 \\ 0 & 0 & 0 & 0.31 & 0.32 \end{matrix}$$

G = conv2(F, H) = lcmF*rcmH

$$= \begin{matrix} 1.21 & 2.64 & 2.87 & 3.10 & 1.68 \\ 4.62 & 9.88 & 10.54 & 11.20 & 5.96 \\ 11.23 & 23.72 & 25.01 & 26.30 & 13.84 \\ 17.53 & 36.62 & 37.91 & 39.20 & 20.44 \\ 23.83 & 49.52 & 50.81 & 52.10 & 27.04 \\ 23.42 & 48.28 & 49.34 & 50.40 & 25.96 \\ 15.81 & 32.44 & 33.07 & 33.70 & 17.28 \end{matrix}$$

Proof

The proof is inductive over r. Firstly, for $r = 1$ equation (2) is reduced to the well-known 1-D case, namely, to convolution of each row of matrix F with a row vector H, and can be written as

$$F \odot H_{1 \times c} = F \cdot \langle H \rangle_C$$

Assuming now that equation (2) is true for some r, we will show that it holds for $r+1$. In this case we have:

$$F \odot H_{r+1\times c} = \begin{array}{|c|c|}\hline \multirow{2}{*}{$\mathrm{lcm}(F,r)$} & $0_{r\times C}$ \\ \cline{2-2} & \multirow{2}{*}{F} \\ \cline{1-1} 0 & \\ \hline \end{array} \cdot \begin{array}{|c|}\hline \mathrm{rcm}(H,R) \\ \hline \langle H(r+1,:)\rangle_C \\ \hline \end{array} =$$

$$\left[\begin{array}{c} \mathrm{lcm}(F,r) \\ 0 \end{array}\right] \mathrm{rcm}(H,R) + \left[\begin{array}{c} 0 \\ F \end{array}\right] H(r+1,:)\rangle_C = \left[\begin{array}{c} F \odot H_{r\times c} \\ 0 \end{array}\right] + \left[\begin{array}{c} 0 \\ F \odot H(r+1,:) \end{array}\right]$$

which completes the proof.

1.1 Implementation comments

The significance of equations (2), (3) and (4) is twofold. Firstly, at the conceptual level, they show in a direct way how the 2-D convolution can be represented by a single matrix multiplication. Secondly, at the implementation level, these equations can be seen as a detailed description of a parallel version of the convolution algorithm.

The direct formation of the **lcm** and **rcm** matrices is rather impractical for real images due to their size. It is possible, however, to replace the replicated storage of the blocks and elements of the convolution matrices by suitable memory reference mechanisms. We will describe these reference mechanisms for three convolution matrices specified in equations (5), (4) and (3), respectively.

If $D = \langle \mathbf{h} \rangle_C$ is a 1-D convolution matrix, as specified in equation (5), then the memory reference mechanism can be described as follows:

$$D\,[m,n] = \begin{cases} h\,[n-m+1]\ , & \text{if } \ 1 \le n-m+1 \le c \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

If $B = \mathrm{rcm}(H,C)$ is the right convolution matrix defined by equation (4), then the memory reference mechanism for this matrix can be specified in the following way:

$$B\,[m,n] = \begin{cases} H\,[1+\alpha\ ,\ n-\beta]\ , & \text{if } \ 1 \le n-\beta \le c \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where α and β are the quotient and remainder from the division of $(m-1)$ by C, that is:

$$(m-1) = \alpha \cdot C + \beta\ ; \quad \alpha = 0,\ldots,r-1\ ; \ \ \beta = 0,\ldots,C-1.$$

Finally, if $A = \mathrm{lcm}(F,r)$ is the left convolution matrix defined by equation (3), then the memory reference mechanism is of the following form:

$$A\,[m,n] = \begin{cases} H\,[m-\gamma\ ,\ 1+\delta]\ , & \text{if } \ 1 \le m-\gamma \le R \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where γ and δ are the quotient and remainder from the division of $(n-1)$ by C, that is:

$$(n-1) = \gamma \cdot C + \delta\ ; \quad \gamma = 0,\ldots,r-1\ ; \ \ \delta = 0,\ldots,C-1.$$

2 A row-by-row algorithm

In some situations the preference may be given to the partially vectorised algorithm in which only one row (or column) of the convolution is computed at each step. The row-by-row algorithm can be described using the MATLAB language in the following way

- Scan the filter vertically, transpose it, and reverse the order of its elements

```
h = fliplr(H(:)')
```

For example:

H = 0.11 0.12
0.21 0.22
0.31 0.32

h = 0.32 0.22 0.12 0.31 0.21 0.11

- Calculate the $c \times (C - c + 1)$ matrix p of pointers to the image columns which are to be multiplied by the scanned filter h

```
cp = convmtx(1:C, c)
p = flipud(cp(:, c:C))
```

For example:

cp = 1 2 3 4 5 6 7 0 p = 1 2 3 4 5 6
0 1 2 3 4 5 6 7 2 3 4 5 6 7

For each row $y = 1, \ldots, R - r + 1$ of the convolution matrix:

- Extract the $r \times C$ strip from the image

```
f = F(y:y+r-1, :)
```

For example:

f = 21 22 23 24 25 26 27 28
31 32 33 34 35 36 37 38
41 42 43 44 45 46 47 48

- Re-arrange columns of the strip matrix f as pointed to by the matrix of pointers, p:

```
gg(:) = f(:, p)
```

For example:

gg = 21 22 23 24 25 26 27
31 32 33 34 35 36 37
41 42 43 44 45 46 47

22 23 24 25 26 27 28
32 33 34 35 36 37 38
42 43 44 45 46 47 48

- Pre-multiply gg by h in order to obtain the y-th row of the convolution of F and H:

```
g(y,:) = h*gg
```

For example:

g(y,:) = 44.65 45.94 47.23 48.52 49.81 51.10 52.39

3 Summary

Representation of the two-dimensional convolution as a product of two appropriately formed matrices specified in equations (3) and (4) is equivalent to a compact description of a parallel convolution algorithm. Subsequently, it has been shown that a direct formation of convolution matrices can be implemented using suitable memory reference mechanisms. Finally, a row-by-row version of the convolution algorithm has been presented.

References

[1] A. K. Jain, *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Addison-Wesley, 1992.

[3] A. P. Papliński, "Matrix multiplication and division of polynomials," *IEE Proceedings*, vol. 132, Pt.D, pp. 95–99, May 1985.

[4] MathWorks, *MATLAB Reference Guide.* The MathWorks Inc., 2004.