

2次元信号処理における演算子の表現について

- 行列による表現とテンソル積による表現 -

宮崎 明雄

九州大学 工学部 情報工学科

(〒812 福岡市東区箱崎 6-10-1)

あらし 画像信号は空間的な2次元信号であり、画像処理の殆どは2次元信号の一種の変換であると考えられる。このような観点から、2次元画像信号の変換を演算子によって定義し、画像処理問題を抽象的に議論するための数学的手法が既に知られている。しかし、実際の画像信号処理問題では、演算子を2次元システムとして具体的に表現することが必要である。本稿では、画像信号処理における演算子の表現問題を、筆者が従来展開している2次元信号の取り扱いに対する作用素代数的アプローチに基づいて取り扱い、演算子が行列のテンソル積によって表現されることを示す。その結果を応用することにより、SIMD(単一命令複数データ流)制御方式のアーキテクチャをもつ計算機システムに適した2次元信号処理アルゴリズムを構成することができる。

Representation and Implementation of Two-Dimensional Signal Processing Operations by using the Tensor Product of Matrices

Akio MIYAZAKI

Department of Computer Science and Communication Engineering

Faculty of Engineering, Kyushu University

(6-10-1 Hakozaki, Higashi-ku, Fukuoka 812, JAPAN)

Abstract This paper deals with the representation problem of operators in two-dimensional (2D) signal processing, which specifies the transformation of 2D signals in the space domain as the ordinary picture or image signal. We first present their representation by means of the tensor product of matrices on the basis of the fundamentals of operator-algebraic treatment of 2D signals established already in our previous contributions. Using the representation, we then give a practical algorithm for effectively implementing 2D signal processing operations on a SIMD multi-processor for which identical processors obtain instructions from a single instruction stream and perform operations on multiple data streams.

1. まえがき

画像信号は膨大な量の情報をもっている。したがって、画像情報の表示、伝送、記録・蓄積などの種々の画像処理問題に関連して、画像信号およびその処理システムの効率的なモデルおよび表現方法を与えることは重要であり、これまでに多くの研究発表がなされている[1]-[3]。

画像信号は空間的な2次元信号であり、画像のエッジ抽出、復元、平滑化などの画像処理の殆どは2次元信号の一種の変換

$$T : \mathcal{D}_1 \rightarrow \mathcal{D}_2 \quad (1.1)$$

であると考えられる。ここで、 \mathcal{D}_1 および \mathcal{D}_2 は任意の2次元画像の集合、写像 T は画像処理モデルを表す演算子である。

画像処理問題を式(1.1)のモデルによって定式化し、演算子 T を用いて抽象的に議論するための数学的手法が既に知られている[2][3]。しかし、実際の信号処理問題では、 T を2次元信号処理システムとして具体的に表現することが必要である。

周知のように、式(1.1)の信号処理モデルが行列によって表現されれば、ストリックアレイなどのアレイプロセッサ[4]に適した信号処理アルゴリズムを構成することができる。他方、信号処理モデルが行列のテンソル積によって表現されれば、SIMD(単一命令複数データ流、single instruction multiple data stream)制御方式のアーキテクチャをもつ計算機システムに適した信号処理アルゴリズムを構成することができる[5]。このような観点から、2次元画像処理における演算子の表現問題を取り扱うことは重要であると考えられる。

筆者らは、最近の研究[6]-[8]において、2次元システムのインパルス応答を用いることにより演算子 T が陽に表現されることを示し、その行列による表現式を与えた。さらに、 T の合成および分解に関する基本的性質を明らかにした。本稿では、このような2次元信号の取り扱いに対する作用素代数的アプローチに基づいて、 T が行列のテンソル積によって表現されることを示す。さらに、その結果を応用することにより、SIMD制御方式の計算機システムに適した2次元信号処理アルゴリズムの構成を行う。

2. 2次元デジタル画像

画像は2次元平面(画面)上に書かれているとし、白黒の濃度の分布によって表現されているとする。このような画像は周知のように、生理学的、物理学的理由により空間周波数に関して帯域制限を受けているために2次元平面上の間隔 d の格子上的画素(標本点)によって表

現される。一つの画像における画素は、離散的に2次元平面上に配置されている。本稿では、画面の左上隅から下へ行番号、右へ列番号をつける。そこで、第 i 行 j 列画素を (i,j) 、その濃度値が $f(i,j)$ で与えられる2次元画像を行列を用いて

$$F = [f(i,j)] \quad (i, j = 1, 2, \dots, N) \quad (2.1)$$

と書く。また、2次元画像の集合を $\mathcal{D} = \{F\}$ で表す。

実際の画像の大きさ(画素数)は有限であるが、理論上、実際上の両面からこの大きさ以上の領域にわたって画像が定義されている必要が生じる。このとき、実際に与えられた画像を外部へ必要に応じて拡大する。このような拡大法には色々なものが考えられ、画像処理の目的に応じて適当に選ばれる。例えば、①外部はすべて0で埋める(0補充)、②実際の画像の周期的反復(周期的拡大)、③実際の画像の縁と同じ値を拡張する(縁伝搬)などの方法が揚げられる。

3. 2次元信号処理における演算子の行列による表現

画像などの2次元信号の処理モデルを写像

$$T : \mathcal{D}_1 \rightarrow \mathcal{D}_2 \quad (3.1a)$$

あるいは、 $\forall F = [f(i,j)] \in \mathcal{D}_1$ に対して、

$$T(F) = G = [g(i,j)] \in \mathcal{D}_2 \quad (3.1b)$$

によって定義する。ここで、 $\mathcal{D}_1, \mathcal{D}_2$ は \mathcal{D} の部分集合を表す。

本稿では、演算子 T が線形であり、かつ並列型の演算子、すなわち G の各成分が入力信号 F のみの値から計算される場合を考える。このとき、 G の (i,j) 成分 $g(i,j)$ は

$$g(i,j) = \sum_{p=-P}^P \sum_{q=-Q}^Q h_{ij}(p,q) f(i-p, j-q) \quad (3.2)$$

と書くことができる。ここで、 P, Q は $P < N/2, Q < N/2$ を満たす正整数であるとする。また、 $h_{ij}(p,q)$ は2次元インパルス応答であり、一般に位置可変な関数である。

注意：式(3.2)の計算を行う場合、式(2.1)で定義された画像 F を前節で述べたやり方で外部へ拡大することが必要である。

筆者らは、演算子 T の表現に関して研究を行い、 T が次のような行列表現をもつことを示した[6][7]。

$$G = T(F) = \sum_{k=1}^K A_k F B_k \quad (3.3)$$

ここで、 A_k, B_k は $\{h_{ij}(p,q)\}$ から求められる N 次の行列である。また、上式における $A_k F B_k$ の項数、つまり K の値は次式によって与えられる。

$$K = \text{rank } \tilde{H} \quad (\leq N^2) \quad (3.4)$$

ここで、 \tilde{H} は

$$\tilde{H} = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1N} \\ H_{21} & H_{22} & \cdots & H_{2N} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NN} \end{bmatrix} \quad (3.5.a)$$

ただし

$H_{ij} = [h_{ij}(p,q)]$ ((2P+1) × (2Q+1) 行列) (3.5.b)
 によって与えられる (2P+1)N × (2Q+1)N 行列である. 特
 に, K=1 のとき, すなわち T の係数が

$$h_{ij}(p,q) = a_i(p) b_j(q) \quad (3.6)$$

のように分離可能なとき, T を分離可能な演算子と呼ぶ.

演算子 T の係数 $\{h_{ij}(p,q)\}$ が 2 次元平面上の位置に依
 存しないとき, すなわち, 式(3.2)が

$$g(i,j) = \sum_{p=-P}^P \sum_{q=-Q}^Q h(p,q) f(i-p, j-q) \quad (3.7)$$

と書けるとき, T を位置不変演算子と呼ぶ. 位置不変演
 算子の場合, 式(3.3)の行列表現において, A_k および B_k
 は Toeplitz 行列となり,

$$K = \text{rank } H H_t \quad (3.8.a)$$

ただし

$$H = [h(p,q)] \quad ((2P+1) \times (2Q+1) \text{ 行列}) \quad (3.8.b)$$

であることが示される. ここで, 下付き添字 t は行列の転
 置を表す.

2 次元信号処理モデルを表す演算子 T に対して, T の
 逆演算子 T^{-1} は

$$T^{-1} : \mathcal{D}_2 \rightarrow \mathcal{D}_1 \quad (3.9.a)$$

あるいは, $\forall G = [g(i,j)] \in \mathcal{D}_2$ に対して,

$$T^{-1}(G) = F = [f(i,j)] \in \mathcal{D}_1 \quad (3.9.b)$$

によって定義される. 文献[8]では, T の係数 $\{h_{ij}(p,q)\}$ が
 条件

$$\rho = \max_{i,j} \frac{\sum_{(p,q) \neq (0,0)} |h_{ij}(p,q)|}{|h_{ij}(0,0)|} < 1 \quad (3.10)$$

を満たすとき, T^{-1} が存在し, 式(3.3)と同じ形の行列表現

$$F = T^{-1}(G) = \sum_{l=1}^L P_l G Q_l \quad (3.11)$$

をもつことを示した.

式(3.3)および式(3.11)は, T および T^{-1} が分離可能な演
 算子の和に分解されることを意味しており, 分解可能な
 演算子を行列の積と演算を効率的に実行するシストリッ
 クアレイなどのアレイプロセッサにより構成すれば, T お
 よび T^{-1} はこれらの回路の並列接続により実現することが
 できる (図 1).

上記の手法の応用例として, 画像の輪郭線抽出フィル
 タの行列表現を与える.

【例 1】 2 次元画像の縦および横の輪郭線を抽出する
 演算子をそれぞれ T_1 および T_2 とすると, 輪郭線を抽出
 する演算子 T は T_1 と T_2 の和 $T_1 + T_2$ によって表される.

また, T_1, T_2 の係数はそれぞれ次式によって与えられ
 る[2].

$$H_1 = [h^{(1)}(p,q)] = \begin{bmatrix} -1/4 & 0 & 1/4 \\ -1/2 & 0 & 1/2 \\ -1/4 & 0 & 1/4 \end{bmatrix} \quad (3.12.a)$$

$$H_2 = [h^{(2)}(p,q)] = \begin{bmatrix} -1/4 & -1/2 & -1/4 \\ 0 & 0 & 0 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} \quad (3.12.b)$$

このとき, T の表現式が次のように得られる.

$$G = T(F) = A_1 F B_1 + A_2 F B_2 \quad (3.13)$$

ただし

$$A_1 = L_1 + 2L_0 + L_{-1} \quad (3.14.a)$$

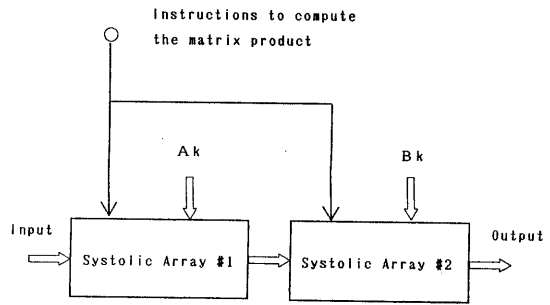
$$B_1 = (-L_1 + L_{-1}) / 4 \quad (3.14.b)$$

$$A_2 = -L_1 + L_{-1} \quad (3.14.c)$$

$$B_2 = (L_1 + 2L_0 + L_{-1}) / 4 \quad (3.14.d)$$

ここで, L_n は次式によって定義される N 次の行列であ
 る.

$$L_n = [l_{ij}^{(n)}], \quad l_{ij}^{(n)} = \begin{cases} 1 & (j = i + n) \\ 0 & (\text{それ以外}) \end{cases} \quad (3.15)$$



(a) シストリックアレイによる $A_k F B_k$ の実現

```
input := F; /* Set input to F */
for k := 1 to K do
begin
  SEND ( A_k, #(2k-1) )
  /* Send the code A_k to the (2k-1)-th array processor */
  SEND ( B_k, #2k )
  /* Send the code B_k to the 2k-th array processor */
  output(k) := GO ( A_k, B_k, input )
  /* Compute the matrix product */
end;
G := SUM ( output(k), k = 1, 2, ..., K );
/* Get the sum of output(k)'s and set G to the sum */
```

(b) シストリックアレイによる信号処理アルゴリズム

図 1 演算子 T のシストリックアレイによる実現

2次元信号処理における演算子 T および逆演算子 T^{-1} の表現とこれらの基本的性質については文献[6]-[8]を参照されたい。

4. 2次元信号処理における演算子のテンソル積による表現

まず、分離可能な演算子

$$G = T(F) = AFB \quad (4.1)$$

のテンソル積による表現を求める。このために、 N 次の行列 F, G から $N^2 (= M)$ 次元ベクトル

$$\tilde{F} = [\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N]_t \quad (4.2.a)$$

$$\tilde{g} = [\tilde{g}_1, \tilde{g}_2, \dots, \tilde{g}_N]_t \quad (4.2.b)$$

をつくる。ここで、 $\tilde{f}_i, \tilde{g}_i (i = 1, 2, \dots, N)$ はそれぞれ F, G の行ベクトル (N 次元ベクトル) を表す。

$$F = \begin{bmatrix} -\tilde{f}_1- \\ -\tilde{f}_2- \\ \vdots \\ -\tilde{f}_N- \end{bmatrix} \quad G = \begin{bmatrix} -\tilde{g}_1- \\ -\tilde{g}_2- \\ \vdots \\ -\tilde{g}_N- \end{bmatrix} \quad (4.2.c)$$

このとき、式(4.1)は

$$\tilde{g} = T(\tilde{F}) = (A \otimes B) \tilde{F} \quad (4.3)$$

と書くことができる。ここで、 $(A \otimes B)$ は N 次の行列 $A = [a_{ij}]$ と $B = [b_{ij}]$ のテンソル積 ($M (= N^2)$ 次の行列) を表す。

$$(A \otimes B) = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1N}B \\ a_{21}B & a_{22}B & \dots & a_{2N}B \\ \vdots & \vdots & & \vdots \\ a_{N1}B & a_{N2}B & \dots & a_{NN}B \end{bmatrix} \quad (4.4)$$

一方、 F および G の列ベクトルから M 次元ベクトル

$$\tilde{F} = \begin{bmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_N \end{bmatrix} \quad \tilde{g} = \begin{bmatrix} \tilde{g}_1 \\ \tilde{g}_2 \\ \vdots \\ \tilde{g}_N \end{bmatrix} \quad (4.5.a)$$

ただし

$$F = \begin{bmatrix} | & | & & | \\ \tilde{f}_1 & \tilde{f}_2 & \dots & \tilde{f}_N \\ | & | & & | \end{bmatrix} \quad G = \begin{bmatrix} | & | & & | \\ \tilde{g}_1 & \tilde{g}_2 & \dots & \tilde{g}_N \\ | & | & & | \end{bmatrix} \quad (4.5.b)$$

をつくると、式(4.1)は行列 $B = [b_{ij}]$ と $A = [a_{ij}]$ のテンソル積 $(B \otimes A)$ を用いて

$$\tilde{g} = T(\tilde{F}) = (B \otimes A) \tilde{F} \quad (4.6.a)$$

ただし

$$(B \otimes A) = \begin{bmatrix} b_{11}A & b_{12}A & \dots & b_{1N}A \\ b_{21}A & b_{22}A & \dots & b_{2N}A \\ \vdots & \vdots & & \vdots \\ b_{N1}A & b_{N2}A & \dots & b_{NN}A \end{bmatrix} \quad (4.6.b)$$

と書くことができる。

注意： 行列のためのテンソル積は一般に非可換

$$(A \otimes B) \neq (B \otimes A)$$

である。したがって、分離可能な演算子 T は2次元信号 F, G からつくられる M 次元ベクトル \tilde{F}, \tilde{g} の選び方により二つのテンソル積表現をもつことになる。しかし、どちらの表現式を用いても同様に取り扱いすることができるので、以下の議論では特に断らない限り式(4.3)の表現式を用いることにする。

前節で示したように、一般の演算子 T は

$$G = T(F) = \sum_{k=1}^K A_k F B_k \quad (4.7)$$

のように分離可能な演算子の和として表現される。したがって、上記の議論から、 T は行列のテンソル積を用いて次式のように表される。

$$\tilde{g} = T(\tilde{F}) = \sum_{k=1}^K (A_k \otimes B_k) \tilde{F} \quad (4.8)$$

次に、議論のための準備として、行列のテンソル積 \otimes に関する幾つかの基本的性質を示す。

1° 結合則

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C \quad (4.9)$$

2° 行列の積に関する分配則

$$(A \otimes B) (C \otimes D) = (AC) \otimes (BD) \quad (4.10)$$

3° 逆行列

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \quad (4.11)$$

4° 転置

$$(A \otimes B)_t = A_t \otimes B_t \quad (4.12)$$

5° 順序行列によるテンソル積の交換

行列 A, B をそれぞれ s 次、 r 次の行列とし、 $n = sr$ とする。このとき、

$$(A \otimes B) = P_{n,s} (B \otimes A) P_{n,r} \quad (4.13)$$

ただし、 $P_{n,s}$ は $(1, 2, \dots, n)$ の順列

$$\begin{aligned} & (i_1, i_2, \dots, i_r, i_{r+1}, \dots, i_{sr}) \\ & = (1, s+1, 2s+1, \dots, (r-1)s+1, \\ & \quad \dots, \dots, \dots, \dots, \dots \\ & \quad s-1, 2s-1, 3s-1, \dots, rs-1, \\ & \quad s, 2s, 3s, \dots, rs, \dots) \end{aligned}$$

に於ける順序行列 (permutation matrix) である。つまり、 $P_{n,s}$ は $(1, i_1), (2, i_2), \dots, (n, i_n)$ 要素だけ1で、他の要素はすべて0である n 次の行列である。

【例2】 $n = 6$ として、順列 $(1, 3, 5, 2, 4, 6)$ に応ずる順序行列 $P_{6,2}$ によるベクトルの変換を示す。

$$P_{6,2} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ x_2 \\ x_4 \\ x_6 \end{bmatrix} \quad (4.14)$$

注意：順序行列によるベクトルの変換はベクトルの各要素の並べ換えを行うシャッフル変換である。これは、データが格納されているメモリのアドレスを順序行列の順列に応じて i 番地から i_l 番地へ換えることに相当する。この変換は一段スイッチネットワークと呼ばれるスイッチ回路（シャッフル交換スイッチ）により実現することができる（図2）。

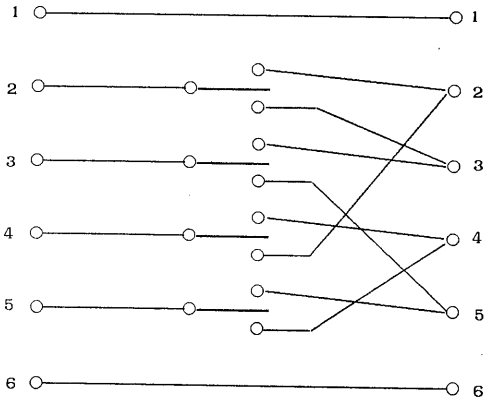


図2 順序行列 $P_{6,2}$ のシャッフル交換スイッチによる実現

テンソル積 $(A \otimes B)$ は、 2° および 5° から

$$\begin{aligned} (A \otimes B) &= (A I) \otimes (I B) \\ &= (A \otimes I) (I \otimes B) \\ &= (A \otimes I) [P_{M,N} (B \otimes I) P_{M,N}] \\ &= [(A \otimes I) P_{M,N}] [(B \otimes I) P_{M,N}] \quad (4.15) \end{aligned}$$

あるいは

$$\begin{aligned} (A \otimes B) &= [P_{M,N} (I \otimes A) P_{M,N}] (I \otimes B) \\ &= [P_{M,N} (I \otimes A)] [P_{M,N} (I \otimes B)] \quad (4.16) \end{aligned}$$

と表すことができる。ここで、 I は N 次の単位行列、 $P_{M,N}$ は M 次の順序行列である。

テンソル積 $(A \otimes I)$ による M 次元ベクトルの変換

$$\begin{aligned} \tilde{g} &= [\tilde{g}_1, \tilde{g}_2, \dots, \tilde{g}_N]_t \\ &= (A \otimes I) \tilde{f} \\ &= (A \otimes I) [\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N]_t \quad (4.17) \end{aligned}$$

は、 N 次元ベクトル間の線形演算（スカラ倍および加算）によって次式のように表される。

$$\tilde{g}_i = a_{i1} \tilde{f}_1 + a_{i2} \tilde{f}_2 + \dots + a_{iN} \tilde{f}_N \quad (i = 1, 2, \dots, N) \quad (4.18)$$

したがって、式(4.15)から、演算子 T を

$$\begin{aligned} \tilde{g} &= T(\tilde{f}) \\ &= \sum_{k=1}^K [(A_k \otimes I) P_{M,N}] [(B_k \otimes I) P_{M,N}] \tilde{f} \quad (4.19) \end{aligned}$$

のように表現することにより、ベクトル計算機システムに適した2次元信号処理アルゴリズムを構成することができる。

一方、テンソル行列 $(I \otimes A)$ はブロック対角行列

$$(I \otimes A) = \begin{bmatrix} A & & & & & \\ & A & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & A \end{bmatrix} \quad (4.20)$$

であるから、 $(I \otimes A)$ による M 次元ベクトルの変換

$$\begin{aligned} \tilde{g} &= [\tilde{g}_1, \tilde{g}_2, \dots, \tilde{g}_N]_t \\ &= (I \otimes A) \tilde{f} \\ &= (I \otimes A) [\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N]_t \quad (4.21) \end{aligned}$$

は、 N 次行列 A による N 次元ベクトルの線形変換

$$\tilde{g}_i = A \tilde{f}_i \quad (i = 1, 2, \dots, N) \quad (4.22)$$

によって表される。したがって、式(4.16)から、演算子 T を

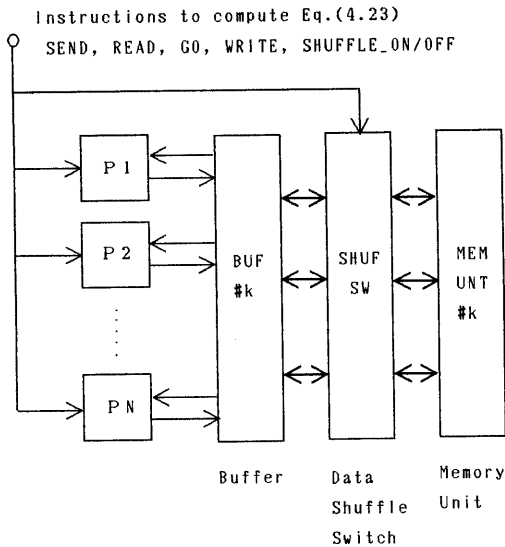
$$\begin{aligned} \tilde{g} &= T(\tilde{f}) \\ &= \sum_{k=1}^K [P_{M,N} (I \otimes A_k)] [P_{M,N} (I \otimes B_k)] \tilde{f} \quad (4.23) \end{aligned}$$

のように表現することにより、SIMD制御方式の計算機システムに適した2次元信号処理アルゴリズムを構成することができる（図3）。

5. むすび

本稿では、2次元信号処理における演算子の表現問題を、筆者が従来展開している2次元信号の取り扱いに対する作用素代数的アプローチに基づいて取り扱い、演算子が行列のテンソル積によって表現されることを示した。さらに、その結果を応用することにより、SIMD（単一命令複数データ流）制御方式のアーキテクチャをもつ計算機システムに適した2次元信号処理アルゴリズムの構成を行った。

筆者は、このような作用素代数的アプローチは種々の2次元画像信号処理問題に対して有効であると考えており、これらに関連した諸問題については別の機会に報告したい。



$P_i (i=1,2,\dots,N)$: Processing Element

(a) SIMDマルチプロセッサによる

$$\tilde{g} = (A_k \otimes B_k) \tilde{f} \text{ の実現}$$

```

for k := 1 to K do
begin
  SHUFFLE ( #k ) OFF /* Shuffle switch off */
  MEM ( #k ) :=  $\tilde{f}$ 
  /* Set the k-th memory unit to  $\tilde{f}$  */
end ;
for k := 1 to K do
begin
  SEND (  $B_k$  , #k )
  /* Send the code  $B_k$  to the k-th SIMD multiprocessor */
  READ ( MEM ( #k ) , buffer )
  /* Read the data from the k-th memory unit to buffer */
  GO (  $B_k$  , buffer )
  /* Compute  $(I \otimes B_k) \times \text{buffer}$  */
  SHUFFLE ( #k ) ON /* Shuffle switch on */
  WRITE ( buffer , MEM ( #k ) )
  /* Write the data from buffer to the k-th memory unit */
  SHUFFLE ( #k ) OFF
  SEND (  $A_k$  , #k )
  READ ( MEM ( #k ) , buffer )
  GO (  $A_k$  , buffer )
  SHUFFLE ( #k ) ON
  WRITE ( buffer , MEM ( #k ) )
  SHUFFLE ( #k ) OFF
end ;
 $\tilde{g} := \text{SUM} ( \text{MEM} ( \#k ) , k = 1, 2, \dots, K ) ;$ 
  /* Get the sum of MEM ( #k )'s and set  $\tilde{g}$  to the sum */

```

(b) SIMDマルチプロセッサによる信号処理アルゴリズム

図3 演算子 T の SIMDマルチプロセッサによる実現

文献

- [1] 小川英光, “信号と画像の復元 [I]-[IV],” 信学誌, Vol. 71, No. 5-8, 1988.
- [2] 鳥脇純一郎, 画像理解のためのデジタル画像処理 (I),(II), 昭晃堂, 1988.
- [3] J. S. Lim, TWO-DIMENSIONAL SIGNAL AND IMAGE PROCESSING, Prentice-Hall, p.694, 1990.
- [4] H. T. Kung, et. al., “A Two Level Pipelined Systolic Array for Convolutions,” VLSI Systems and Computations, pp.255-264, Computer Science Dept., Carnegie Mellon Univ., 1981.
- [5] J. Granata, et. al., “The Tensor Product : A Mathematical Programming Language for FFTs and other Fast DSP Operations,” IEEE Signal Processing Magazine, Vol. 9, No. 1, pp.40-48, Jan. 1992.
- [6] A. Miyazaki, “Matrix Representation of Operators in Two-Dimensional Signal Processing and its Applications to Picture Processing,” Proc. IEEE Int. Symp. Circuits and Syst., Vol. 5, pp. 2912-2915, June 1991.
- [7] A. Miyazaki, “Matrix Representation of Operators in Two-Dimensional Signal Processing,” IEICE Trans., Vol. E-74, No. 10, pp. 3098-3101, Oct. 1991.
- [8] 宮崎, 中田, 森田, “二次元信号処理における演算子の表現とその画像信号のモデリングへの応用,” 第14回情報理論とその応用シンポジウム予稿集, Vol. 1, pp. 229-232, Dec. 1991.