

## SIGHT上の3次元ベクトル演算に関する考察

成瀬 正 吉田 雅治 高橋 時市郎

NTTヒューマンインタフェース研究所

### [概要]

SIGHTのTARAI演算器は3次元のベクトル演算を並列実行するように構成されている。本稿では、その並列演算能力を集合論的立場から明らかにする。まず、3次元(定数)ベクトル、 $3 \times 3$ (定数)マトリクスから構成される集合を考え、その上に基本演算を定義する。さらに、3次元ベクトル変数を導入しこの集合を拡張する(それをGとする)。そして、集合Gが持ついくつかの基本的な性質を示す。また、G上に微分演算子を導入し、微分演算に関してGは閉じていることを示す。次に演算の並列性に関して論ずる。そしてGに属する式には本質的に逐次演算を必要とするものがあるが、その部分を除けば、TARAI演算器で完全に並列演算できることを示す。また、TARAI演算器はGの式を最小演算ステップ数で計算するという意味で最良の構成であることを示す。

### On a 3 Dimensional Vector Arithmetic on the Graphics Computer SIGHT

Tadashi Naruse Masaharu Yoshida Tokiichiro Takahashi

NTT Human Interface Laboratories

1-2356 Take, Yokosuka-shi, Kanagawa 238-03, Japan

### Abstract

TARAI-operation unit of the dedicated graphics processor SIGHT have been proposed as an architecture that executes 3 dimensional (3-D) vector operation in parallel. In this paper, we clarify a parallel operation ability of the TARAI unit from the set theory view point. First, we define the fundamental operations on the set F which consists of 3-D (constant) vectors and  $3 \times 3$ (constant) matrices. Second, we introduce a 3-D vector variable to the set F, and make an extended set G which consists of expressions including constant vectors/matrices and vector variable. Third, we show basic properties that the set G has. Moreover, we introduce a differential operator on G, and show that the set G is closed under the differential operation. Finally, we discuss on the parallelism of vector operation on the TARAI unit. We show TARAI unit can calculate the expression fully in parallel except the intrinsically sequential operation. We also show TARAI unit is the best configuration for the 3-D vector operations in the sense that it executes any expression in G in the minimal execution steps.

## 1. はじめに

極めて写実的な画像が生成できる光線追跡法の高速処理を目指して種々の計算機が提案されている [1-4]。当研究所で開発したSIGHT [4]は、光線追跡法に内在する並列性を徹底的に引き出すことによりその高速度処理を実現した画像生成計算機である。すなわち、SIGHTでは次のような並列処理を実現し、システム(16プロセッサ構成)として汎用ミニコンピュータ(VAX11-780)の約160倍の性能を達成し、スーパーコンピュータに匹敵する性能を得ている[5]。

1. マルチプロセッサ構成による画素レベルの並列処理。
2. TARA I演算器と名付けたベクトル演算器による追跡計算自体の並列処理。
3. その他アーキテクチャ上の工夫による並列処理。

SIGHTの第一の特長は2.の並列処理にある。すなわち、光線追跡計算は三次元空間のベクトル演算であり、 $x$ ,  $y$ ,  $z$  軸に関して対称な演算となることが多い。この対称性に着目して、3個の演算器とメモリを簡単なネットワークで結合したTARA I演算器を構成し、演算レベルで並列処理を行う。これにより、ベクトル演算を最大3倍高速に処理できる。

3次元のベクトル演算を主体とする応用分野は光線追跡法の他にも、音場解析、アンテナ設計など多方面に及ぶ。これらの分野にもTARA I演算器が有効性を発揮するものと思われる。そうすると、より一般的な

立場からTARA I演算器の並列演算能力を明らかにするという問題が生じる。具体的には、「3次元ベクトル演算の集合(実際には式の集合)を考え、その集合に属する式の並列実行という観点でTARA I演算器の能力を明らかにする」という問題である。

そこで、本稿では集合論的観点からSIGHTで効率良く実行できるベクトル演算を検討する。まず、(3次元)定数ベクトル、(3×3)定数マトリクスの集合を考え、その上に基本演算を定義する。更に(3次元)ベクトル変数を導入してこの集合を拡張する。(拡張した集合をGとする。)次いで、集合Gがどのような性質を持つかを明らかにする。そして、集合Gに属する式を並列実行した場合と逐次的に実行した場合との演算ステップ数の比で演算並列度を定義し、ベクトル演算の並列度を論じる。また、TARA I演算器は、集合Gに属する式を最小実行ステップ数で計算するという意味で最良であることを示す。

## 2. SIGHTについて

SIGHTの1プロセッサ(PE)の構成を図1に示す。PEは、4個の主要ユニットからなる。TARA Iユニットは三次元ベクトル演算を行い、MPユニットはPEの制御、ホストとの通信等を行う。DBMユニットは、画像の定義データ等を格納する大規模メモリユニットである。MMユニットは、TARA I、MP内のレジスタファイルとDBM間のデータ転送(DMA転送)制御を行う。

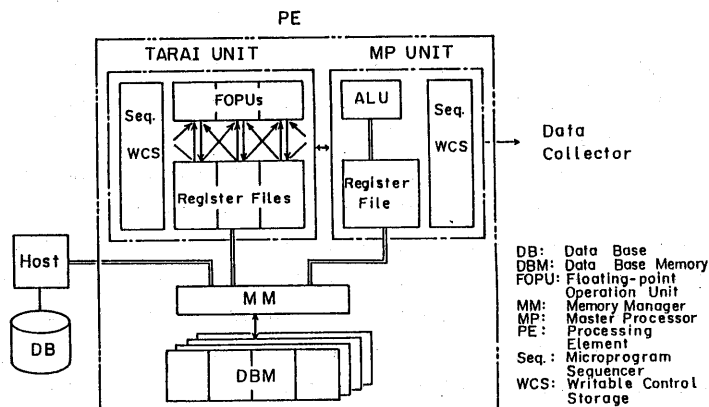


図1 PEの構成

・TARA Iユニットの構成と動作

図1のTARA Iユニットの中心部は、3組の浮動小数点演算器とレジスタファイルをネットワークで結合した構成となっている。これをTARA I演算器とよぶ。ネットワークはシンプルであり、次の4通りの結合(データ転送)が可能である。

1. レジスタファイル→真上の演算器
2. レジスタファイル→右上の演算器
3. レジスタファイル→左上の演算器
4. 演算器→真下のレジスタファイル

3個のレジスタファイルにはそれぞれx、y、z軸に関するデータを格納し、3個の演算器はそれぞれx、y、z軸に関する演算を行う。例えば、 $(a_y * a_z, a_z * a_x, a_x * a_y)$ なるベクトルを求める計算は、次のように実行される(図2参照)。但し、データ $(a_x, a_y, a_z)$ はあらかじめレジスタファイルに格納されているものとする。

- (1) データ $(a_x, a_y, a_z)$ を左上の演算器に転送する。
- (2) データ $(a_x, a_y, a_z)$ を右上の演算器に転送する。
- (3) 乗算を行う。
- (4) 結果を真下のレジスタファイルに転送する。

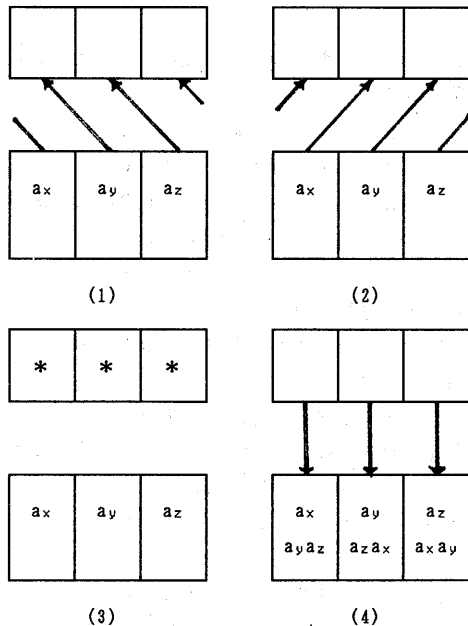


図2 TARA I演算器の動作

TARA I演算器はこの一連の動作を1マイクロ命令で実行できる。

また、マトリクス演算を効率よく行うため、レジスタファイルのアドレスは、それぞれにオフセットを付加して指定できるようになっている。

3. 定義および基本的性質

この節では、TARA I演算器上のベクトル演算を議論するために必要な基本事項の定義を行い、また、そこから導かれる基本的性質について述べる。

3次元(定数)ベクトル、 $3 \times 3$ (定数)マトリクスの集合をFとする。ただし、ベクトルおよびマトリクスの要素は実数とする。すなわち、例えばFのベクトル要素を $\mathbf{a}$ 、 $\mathbf{b}$ ・・・等と書くと、 $\mathbf{a} = (a_1, a_2, a_3)$ 、 $\mathbf{b} = (b_1, b_2, b_3)$ で $a_i, b_i \in \mathbb{R}$ である。また、マトリクスは次のようなskewed配置になっているものとする。ただし、 $m_{ij}$ はマトリクス要素であり、 $\mathbf{m}_i$ はベクトルで $(m_{i1}, m_{i2}, m_{i3})$ を意味する。また、 $\oplus$ 、 $\ominus$ は後述するシフト演算である。

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

あるいは

$$\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix}$$

通常配置

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{23} & m_{21} & m_{22} \\ m_{32} & m_{33} & m_{31} \end{pmatrix}$$

あるいは

$$\begin{pmatrix} \mathbf{m}_1 \\ \ominus \mathbf{m}_2 \\ \ominus \mathbf{m}_3 \end{pmatrix}$$

skewed配置

まず、F上に基本演算を定義する。次の4通りの基本演算クラスが考えられる。

- ベクトル-ベクトル演算
- ベクトル-マトリクス演算
- マトリクス-ベクトル演算
- マトリクス-マトリクス演算

ベクトル-ベクトル演算クラス

① ベクトル要素間の二項演算  $\odot$

$$\mathbf{a} \odot \mathbf{b} = (a_1 \odot b_1, a_2 \odot b_2, a_3 \odot b_3)$$

$\odot$ は+、-、 $\times$ 、 $\div$ のいずれかである。ただし、割算では、 $\mathbf{b}$ はstrictly-nonzero ( $b_1, b_2, b_3 \neq 0$ )でなけ

ればならない。

②ベクトル要素の単項演算 -

$$-a = (-a_1, -a_2, -a_3)$$

③シフト演算  $\ominus, \oplus$

$$\oplus a = (a_3, a_1, a_2)$$

$$\ominus a = (a_2, a_3, a_1)$$

以下の議論でわかるように、シフト演算はS I G H T上のベクトル代数において中心的役割を果たす。

### ベクトル-マトリクス演算クラス

この演算は考えない。

### マトリクス-ベクトル演算クラス

積を考える。

### マトリクス-マトリクス演算クラス

①マトリクス要素間の二項演算  $\odot$

$\odot$ は+、-、 $\times$ 、 $\div$ のいずれかである。割算では、除数はstrictly-nonzeroでなければならない。

②マトリクス要素の単項演算 -

ベクトル演算と同様。

③シフト演算  $\ominus, \oplus, \textcircled{1}, \textcircled{2}$

$$\ominus M = \begin{pmatrix} \ominus m_1 \\ \ominus m_2 \\ \ominus m_3 \end{pmatrix} \quad \textcircled{1} M = \begin{pmatrix} \ominus m_2 \\ \ominus m_3 \\ m_1 \end{pmatrix}$$

である。 $\ominus, \textcircled{1}$ も同様。

④対角要素の抽出diag, diag<sup>r</sup>, diag<sup>l</sup>

通常配置の場合

$$\text{diag}(M) = (m_{11}, m_{22}, m_{33})$$

$$\text{diag}^r(M) = (m_{31}, m_{12}, m_{23})$$

$$\text{diag}^l(M) = (m_{21}, m_{32}, m_{13})$$

skewed配置の場合

$$\text{diag}(M) = (m_{11}, m_{21}, m_{31})$$

$$\text{diag}^r(M) = (m_{32}, m_{12}, m_{22})$$

$$\text{diag}^l(M) = (m_{23}, m_{33}, m_{13})$$

ただし、 $m_{ij}$ はマトリクス要素

⑤マトリクス-マトリクス積

通常のマトリクス積。

集合Fに関して次の性質が成り立つ。

[性質1]

Fは上記演算に関して閉じている。

次に、ベクトル変数xを導入する。(x=(x1, x2, x3)あるいはx=(x, y, z)と記す。) 集合G<sub>0</sub>(=F)の要

素とxに上記の演算を行って作られる式をG<sub>0</sub>に付加した集合をG<sub>1</sub>[x]と書く(注)。さらに、その推移的閉包G[x]を次のように定義する。

$$G[x] = \bigcup_n G_n[x]$$

ここで、

$$G_n[x] = G_{n-1}[x] \cup \{e \odot x : e \in G_{n-1}[x], \odot \text{は上記演算}\}$$

作り方から明らかなように、

[性質2]

G[x]は上記演算に関して閉じている。

G[x]は定数(ベクトル、マトリクス)と式から成る集合であり、式はベクトル値をとる。

次に、G[x]上に微分演算子を導入する。

すなわち、E(x) ∈ G[x]に対して、微分演算子を、

$$\frac{dE(x)}{dx} = \left( \frac{\partial E_x}{\partial x}, \frac{\partial E_y}{\partial y}, \frac{\partial E_z}{\partial z} \right)$$

のように定義する。(これをE'(x)とも記す。)

微分演算がG[x]で閉じていることを以下に示す。最初に、ベクトルのシフト演算に関していくつかの性質を述べておく。

[性質3]  $\ominus$ に関する二、三の公式( $\oplus$ についても同様の関係が成立する。)

f(x), g(x) ∈ G[x], M ∈ G[x]のとき、

$$1. \ominus(f(x) \odot g(x)) = (\ominus f(x)) \odot (\ominus g(x)) \quad ; \odot \text{は二項演算}$$

$$2. \ominus(\ominus f(x)) = \oplus f(x) \quad ; \oplus \text{は単項演算}$$

$$3. \ominus \oplus f(x) = \ominus f(x)$$

$$4. \ominus(M f(x)) = (\textcircled{1}M) (\ominus f(x))$$

[証明]

いずれも定義にもどって考えれば明かである。たとえば4.は、M=(m<sub>ij</sub>)、f(x)=(f<sub>i</sub>)とすると

$$M f(x) = (\sum m_{1i} f_i, \sum m_{2i} f_i, \sum m_{3i} f_i)$$

$$(\textcircled{1}M) (\ominus f(x)) =$$

(注)意味のある演算だけを考える。たとえば、a+xは意味があるが、M+xは意味を持たない。

$$(\Sigma m_2 i f_i, \Sigma m_3 i f_i, \Sigma m_1 i f_i)$$

より成立することがわかる。

[性質4] (シフト還元)

$\ominus f(x)$  なる形の式は中間のレベルにシフト演算を含まない式に還元できる。すなわち、還元した式ではシフト演算は $\ominus x$ 、 $\ominus x$ なる形でのみ現れる。

[証明]

付録1に示す。

次に、微分演算に関して二、三の公式を述べる。

[性質5]

$$(\overset{\sim}{f}(x) \pm \overset{\sim}{g}(x))' = \overset{\sim}{f}'(x) \pm \overset{\sim}{g}'(x)$$

$$(\overset{\sim}{f}(x) \times \overset{\sim}{g}(x))' = \overset{\sim}{f}'(x) \times \overset{\sim}{g}(x) + \overset{\sim}{f}(x) \times \overset{\sim}{g}'(x)$$

$$(\ominus x)' = (\ominus x)' = 0$$

[証明] 定義にもどって考えれば明か。

性質4、性質5を使うと、次の性質が証明できる。

[性質6]

$G[x]$  は、上記微分演算に関して閉じている。

[証明]

付録2に示す。

性質6から、曲面の式が $G[x]$ に属すなら、その法線ベクトルを表す式は、 $G[x]$ に属することがわかる。

次に、TARA I演算器で上記の演算を行うのに必要なステップ数(コスト)について述べる。

1. ベクトル-ベクトル演算のコストは1である。ただし、シフト演算のコストは、他の演算と組み合わせて使用した場合は0とする。たとえば、 $a + b$ のコストは1、 $a + \ominus b$ のコストも1である。

2. マトリクス-マトリクス演算の①~③の演算のコストは3、④の演算のコストは1である。ただし、シフト演算および対角要素の抽出演算のコストは、他の演算と組み合わせて使用した場合は0とする。

3. マトリクス・ベクトル積  $Ma$

$$Z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} m_1 \times a \\ \ominus \begin{pmatrix} m_2 \\ m_3 \end{pmatrix} \times a \end{pmatrix}$$

とおく。(Zは通常の配置になっている。)

$$Ma = \text{diag}(Z) + \ominus \text{diag}^r(Z) + \ominus \text{diag}^l(Z)$$

となる。従ってコストは5である。( $\ominus$ 、 $\text{diag}$ 等の演算のコストは0である。)

4. マトリクス・マトリクス積  $L = MN$

$$Z^i = \begin{pmatrix} m_i \times \text{diag}(N) \\ m_i \times \ominus \text{diag}^r(N) \\ m_i \times \ominus \text{diag}^l(N) \end{pmatrix} \quad i=1,2,3$$

と置くと、

$$l_1 = \text{diag}(Z^1) + \ominus \text{diag}^r(Z^1) + \ominus \text{diag}^l(Z^1)$$

$$\ominus l_2 = \ominus \text{diag}(Z^2) + \text{diag}^r(Z^2) + \text{diag}^l(Z^2)$$

$$\ominus l_3 = \ominus \text{diag}(Z^3) + \ominus \text{diag}^r(Z^3) + \text{diag}^l(Z^3)$$

となる。従って、コストは15である。

$G[x]$ の式Eに対し、Eを逐次計算機で計算した場合の演算ステップ数を $N_s(E)$ とし、TARA I演算器で計算した場合の演算ステップ数を $N_p(E)$ とする。また、これらの比、

$$r = N_p(E) / N_s(E)$$

を演算並列度とよぶ。

このようにrを定義するとrは、

$$1 \leq r \leq 3$$

の範囲の値をとる。

上記の各基本演算に対して、それを逐次的に計算した場合の演算ステップを数えて比較すれば、

[性質7]

上記の各基本演算に対してrの値は3である。

という結果を得る。

#### 4. 演算の拡張

本節では、ベクトル演算やマトリクス演算で使用される内積、外積、逆行列等の種々の演算が3節で定義した基本演算の組合せで記述できることを示しその演算並列度を検討する。(線形代数[6]で使用される主要な演算について検討する。)

まず、以下の演算に有用な2つの演算を定義する。

$$\begin{aligned} \text{ベクトル要素の総和: } \Sigma x &= x + \ominus x + \ominus x \\ &= (\Sigma x_i, \Sigma x_i, \Sigma x_i) \end{aligned}$$

$$\begin{aligned} \text{ベクトル要素の連乗: } \Pi x &= x \times \ominus x \times \ominus x \\ &= (\Pi x_i, \Pi x_i, \Pi x_i) \end{aligned}$$

これらの演算は、本質的に逐次演算であり、並列演算はできない。これらの演算のコストは2である。また、 $r$ の値は1である。

### ベクトル演算

①内積：ここでは内積を次のように定義する。

$$(a, b) = \sum (a \times b)$$

内積演算の結果はスカラーであるが、ここでは、ベクトルとして扱う。そうすることによる計算上の不都合は何もない。

$$N_p = 3, N_s = 5 \quad \therefore r = 5/3$$

②外積： $[a, b] = \ominus a \times \ominus b - \ominus a \times \ominus b$

$$N_p = 3, N_s = 9 \quad \therefore r = 3$$

③rot：回転は次のように記述できる。

$$\text{rot}(E) = \ominus \frac{d(\ominus E)}{dx} - \ominus \frac{d(\ominus E)}{dx}$$

従って、 $E \in G[x]$ なら性質6により $\text{rot}(E) \in G[x]$ である。

④div：発散は次のように定義する。

$$\text{div}(E) = \sum \frac{dE}{dx}$$

演算結果は、ベクトルと考える。回転と同様 $E \in G[x]$ なら $\text{div}(E) \in G[x]$ である。

### マトリクス演算 (skewed配置行列)

①転置行列：転置行列は次のように記述できる。

$${}^t M = \begin{pmatrix} \text{diag}(M) \\ \text{diag}(M) \\ \text{diag}(M) \end{pmatrix}$$

$$N_p = 3, N_s = 9 \quad \therefore r = 3$$

②行列式：行列式の値は次のように記述できる。(ただし、演算結果はベクトル値とする。)

$$\det(M) = \sum (m_1 \times \ominus(\ominus m_2) \times \ominus(\ominus m_3) -$$

$$m_1 \times \ominus(\ominus m_2) \times \ominus(\ominus m_3))$$

$$N_p = 7, N_s = 17 \quad \therefore r = 2, 4, 3$$

③逆行列：逆行列は、以下のように定義される。

$$M^{-1} = \frac{1}{\Delta} \begin{pmatrix} \Delta_{11} & \Delta_{21} & \Delta_{31} \\ \Delta_{12} & \Delta_{22} & \Delta_{32} \\ \Delta_{13} & \Delta_{23} & \Delta_{33} \end{pmatrix}$$

ここで、 $\Delta = \det(M)$ 、 $\Delta_{ij}$ は $(i, j)$ 余因子[6]である。

ここでは、skewed配置を前提としているから、上式は次のようになる。

$$M^{-1} = \frac{1}{\Delta} \begin{pmatrix} \Delta_{11} & \Delta_{21} & \Delta_{31} \\ \Delta_{32} & \Delta_{12} & \Delta_{22} \\ \Delta_{23} & \Delta_{33} & \Delta_{13} \end{pmatrix}$$

そうすると、余因子行列の各行は、次のように記述で

きる。

$$(\Delta_{11} \ \Delta_{21} \ \Delta_{31}) = \ominus \text{diag}^r(M) \times \ominus \text{diag}^l(M) - \text{diag}^r(M) \times \text{diag}^l(M)$$

$$(\Delta_{32} \ \Delta_{12} \ \Delta_{22}) = \ominus \text{diag}(M) \times \ominus \text{diag}^l(M) - \text{diag}(M) \times \text{diag}^l(M)$$

$$(\Delta_{23} \ \Delta_{33} \ \Delta_{13}) = \ominus \text{diag}(M) \times \ominus \text{diag}^r(M) - \text{diag}(M) \times \text{diag}^r(M)$$

したがって、逆行列を求める演算ステップ数は、

$$N_p = 19, N_s = 53 \quad \therefore r = 2, 79.$$

尚、 $3 \times 3$ 行列の場合、直接計算のほうが掃き出し法で計算するより演算回数は少ない。

### 二次形式

二次形式は、 $(x, Mx)$ で与えられる。ここで、 $x$ はベクトル、 $M$ はマトリクスである。したがって、以下のように記述できる。

$$(x, Mx) = \sum (x \times Mx)$$

$$N_p = 9, N_s = 23 \quad \therefore r = 2, 56$$

前節および本節の議論からわかるように、ここで検討しているベクトル演算において並列性を阻害する要因は、「演算の本質的逐次性」だけである。したがって、次の性質が成り立つ。

[性質8] 与えられた式が上記の要因を含まないならば、すなわち、 $\sum$ 、 $\ominus$ を含まなければ、与式は演算並列度3で計算できる。

## 5. TARA I 演算器の最良性

今、3個の2入力1出力の演算器を用いて演算システムを構築する時、いかなる構成にしたら、①上で議論した演算が最小ステップで計算できるか、②構成要素を最小にできるか、ということを考える。①、②を満たす構成を最良の構成とよぶ。答えは、TARA I 演算器の構成である。以下にそれを示す。

[最小ステップであること]

3個の演算器を用いれば演算並列度は高々3である。逆に、 $G[x]$ の式が演算並列度3で計算できるならそれはその式を最小ステップ数で計算したことになる。性質8から $\sum$ 、 $\ominus$ を含まない $G[x]$ の式の演算並列

度は3であるからTARA Iはその式を最小ステップで計算する。また、 $\Sigma$ 、 $\Pi$ の計算は本質的に逐次であるからいかなる構成にしても2ステップを要する。よって、TARA I演算器は $G[x]$ の任意の式を最小ステップ数で計算する。

[最小構成要素であること]

TARA I演算器の構成要素は、

- ・演算器とメモリ：3個ずつ必要。
  - ・ネットワーク：上下の結合の他に、シフト演算に対応する右上、左上演算器との結合が必要。
  - ・アドレスオフセット：対角要素のアクセスに必要。であり、これで上記の演算はすべて実行できる。
- 逆に、これらのどの要素が欠けても上記の演算を並列度を落とすことなく実行することはできない。すなわち、最小構成要素である。

よって、TARA I演算器は $G[x]$ の式の計算をするのに最良の構成であることがわかる。

## 6. おわりに

SIGHTのTARA I演算器上で効率よく計算できる式を明らかにするため、3次元ベクトル、 $3 \times 3$ マトリクスから構成される式の集合を考えその性質を調べた。線形代数に現れる基本的なベクトル演算がこの集合に含まれることを示した。したがって、この集合は3次元ベクトル演算にあらわれるほとんど全ての式を含む。

ついで、TARA I演算器でこの集合に属す式を計算したときの演算並列度について検討した。残念ながらこの集合には本質的に逐次的な演算を必要とする式を含む。そのような式に対しては、TARA I演算器の演算並列度は低下する。しかし、それを含まない式に対しては、TARA I演算器は演算並列度3で式の計算を行える。また、TARA I演算器は3次元ベクトル演算を実行する最良の構成法であることを示した。

曲面と光線の交点計算をこの集合に属する式で記述できるかどうかについてはここでは検討しなかった。しかし、平面、二次曲面との交点計算は大部分がこの集合に属する式で記述できることがわかっている。その他の曲面については、今後検討を進める。また、光線追

跡法以外のベクトル演算を主体とする応用についても検討を行い、TARA I演算器の応用範囲を広げていく。

[謝辞] 日頃ご指導頂く知能ロボット部滝川啓主幹員に感謝する。金子博主幹員、橋本秋彦研究主任、斎藤隆文研究主任には、いくつかの有益な助言を頂いた。感謝する。

[参考文献]

- [1] H. Nishimura, et. al.: "LINKS-1: a parallel pipelined multicomputer system for image creation", Proc. 10th Annu. Int'l Symp. on Comput. Arch., pp. 387-394, 1984
- [2] 村上、他: "セルラアレイプロセッサCAPによるレイトレーシング"、情報処理学会、グラフィックスとCAD研究会、CAD 22-2、1986
- [3] 日高、他: "マルチコンピュータ画像生成システムMC-1"、情報処理学会、計算機アーキテクチャ研究会、CA 58-5、1985
- [4] T. Naruse, M. Yoshida, T. Takahashi: "SIGHT - A Dedicated Computer Graphics Machine", Computer Graphics Forum, Vol. 6, No. 4, pp. 327-334, 1987
- [5] 吉田、成瀬、高橋: "画像生成計算機SIGHT-2"、情報処理学会、計算機アーキテクチャ研究会、CA 77-6、1989
- [6] 佐竹: "行列と行列式"、裳華房、昭和46年

## 付録1 性質4の証明

⊙について示す。(⊕についても同様)

⊙  $k(x) \in G[x]$  なる式を考えたとき、 $k(x)$  は次のいずれかの形をしていなければならない。(ここで $k(x)$ はベクトルであることに注意。)

- ①  $f(x) \odot g(x)$  ; ⊙は二項演算  
 $f(x), g(x) \in G[x]$
- ②  $\odot f(x)$  ; ⊙は単項演算  
 $f(x) \in G[x]$
- ③  $\odot f(x), \ominus f(x)$  ;  $f(x) \in G[x]$
- ④  $Mf(x)$  ; Mは定数マトリクス  
 $f(x) \in G[x]$
- ⑤  $a, x$  ; 定数/変数ベクトル

上記のうち①~④にシフト演算を適用すると、性質3により部分式のシフト演算に還元することができる。

また、⑤の場合はシフト還元された形になっている。  
この還元手続きを繰り返し行うことにより $\ominus \mathbf{k}(\mathbf{x})$ は中間レベルにシフト演算を含まない式に還元できることをいう。そのためには、有限回の繰り返しで還元手続きが終了することを言えばよい。

いま、式に含まれる基本演算の数を $\text{Nop}(\cdot)$ で表す。

$$\text{Nop}(\ominus(\mathbf{f}(\mathbf{x}) \odot \mathbf{g}(\mathbf{x}))) > \text{Nop}(\ominus \mathbf{f}(\mathbf{x}))$$

$$\text{Nop}(\ominus(\mathbf{f}(\mathbf{x}) \odot \mathbf{g}(\mathbf{x}))) > \text{Nop}(\ominus \mathbf{g}(\mathbf{x}))$$

$$\text{Nop}(\ominus(\odot \mathbf{f}(\mathbf{x}))) > \text{Nop}(\odot \mathbf{f}(\mathbf{x}))$$

$$\text{Nop}(\ominus(\ominus \mathbf{f}(\mathbf{x}))) > \text{Nop}(\ominus \mathbf{f}(\mathbf{x}))$$

$$\text{Nop}(\ominus(\ominus \mathbf{f}(\mathbf{x}))) > \text{Nop}(\mathbf{f}(\mathbf{x}))$$

$$\text{Nop}(\ominus(\mathbf{M} \mathbf{f}(\mathbf{x}))) =$$

$$\text{Nop}(\oplus \mathbf{M}(\ominus \mathbf{f}(\mathbf{x}))) > \text{Nop}(\ominus \mathbf{f}(\mathbf{x}))$$

であるから、部分式の基本演算の数は元の式の基本演算の数より必ず小さい。 $\text{Nop}(\ominus \mathbf{k}(\mathbf{x}))$ は有限であるから、シフト演算の入った部分式はいつかは、 $\ominus \mathbf{x}$ 、 $\ominus \mathbf{a}$ 等の形になる。[証明終]

## 付録2 性質6の証明

$\mathbf{k}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$ に対して、 $\mathbf{k}(\mathbf{x})$ をシフト還元する。それをあらためて $\mathbf{k}(\mathbf{x})$ とする。そうすると、 $\mathbf{G}[\mathbf{x}]$ の作り方から $\mathbf{k}(\mathbf{x})$ は次のいずれかの形にかける。

$$\textcircled{1} \mathbf{f}(\mathbf{x}) \odot \mathbf{g}(\mathbf{x}) ; \odot \text{は二項演算} ; \mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$$

$$\textcircled{2} \odot \mathbf{f}(\mathbf{x}) ; \odot \text{は単項演算} ; \mathbf{f}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$$

$$\textcircled{3} \ominus \mathbf{x}, \ominus \mathbf{x}$$

$$\textcircled{4} \mathbf{M} \mathbf{f}(\mathbf{x}) ; \mathbf{M} \text{は定数マトリクス} ; \mathbf{f}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$$

$$\textcircled{5} \mathbf{a}, \mathbf{x} ; \text{定数/変数ベクトル}$$

各場合につきその微分が $\mathbf{G}[\mathbf{x}]$ に入るかをチェックする。

$$\textcircled{1} \mathbf{f}(\mathbf{x}) \odot \mathbf{g}(\mathbf{x}) : \odot \text{は} +, -, \times, / \text{のいずれかである。} \mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}] \text{ならば性質5により} (\mathbf{f}(\mathbf{x}) \odot \mathbf{g}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$$

$$\textcircled{2} \odot \mathbf{f}(\mathbf{x}) : \mathbf{f}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}] \text{ならば} (\odot \mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$$

③、⑤は明か。

$$\textcircled{4} (\mathbf{M} \mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$$

$$\left( \frac{\partial}{\partial \mathbf{x}} - (m_{11} f_1 + m_{12} f_2 + m_{13} f_3), \right.$$

$$\left. \frac{\partial}{\partial \mathbf{y}} - (m_{21} f_1 + m_{22} f_2 + m_{23} f_3), \right.$$

$$\left. \frac{\partial}{\partial \mathbf{z}} - (m_{31} f_1 + m_{32} f_2 + m_{33} f_3) \right)$$

ここで、

$$\mathbf{V} = \begin{pmatrix} \mathbf{f}(\mathbf{x}) \\ \odot \mathbf{f}(\mathbf{x}) \\ \ominus \mathbf{f}(\mathbf{x}) \end{pmatrix}$$

$$\mathbf{W} = \begin{pmatrix} \text{diag}(\mathbf{V}) \times \text{diag}(\mathbf{M}) \\ \odot \text{diag}'(\mathbf{V}) \times \text{diag}'(\mathbf{M}) \\ \ominus \text{diag}'(\mathbf{V}) \times \text{diag}'(\mathbf{M}) \end{pmatrix}$$

と置くと上式は、

$$\text{diag}(\mathbf{W}) + \odot \text{diag}'(\mathbf{W}) + \ominus \text{diag}'(\mathbf{W})$$

とかける。

従って、 $(\mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$ 、 $(\odot \mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$ 、 $(\ominus \mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$ なら $(\mathbf{M} \mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$ である。

以上から $\mathbf{f}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$ 、 $\mathbf{g}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$ 、 $(\odot \mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$ 、 $(\ominus \mathbf{f}(\mathbf{x})) \in \mathbf{G}[\mathbf{x}]$ について、すなわち、部分式について上記の①～⑤のチェックを行うことにより、元の $\mathbf{k}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$ にはいるかどうか分かる。(ただし、 $\odot \mathbf{f}(\mathbf{x})$ 、 $\ominus \mathbf{f}(\mathbf{x})$ についてはシフト還元を行った式をチェックする。)よって、残るは有限回で部分式のチェックが終了することを言えばよい。

①、②、④について、シフト演算を除く基本演算の数を $\text{Nop}(\cdot)$ とすると、

$$\text{Nop}(\mathbf{k}(\mathbf{x})) > \text{Nop}(\mathbf{f}(\mathbf{x}))$$

$$\text{Nop}(\mathbf{k}(\mathbf{x})) > \text{Nop}(\mathbf{g}(\mathbf{x}))$$

$$\text{Nop}(\mathbf{k}(\mathbf{x})) > \text{Nop}(\odot \mathbf{f}(\mathbf{x}))$$

$$\text{Nop}(\mathbf{k}(\mathbf{x})) > \text{Nop}(\ominus \mathbf{f}(\mathbf{x}))$$

である。なぜなら、シフト還元により、シフト演算以外の基本演算の数は変わらないからである。また、シフト還元を行った式では、シフト演算は $\odot \mathbf{x}$ 、 $\ominus \mathbf{x}$ の形であらわれる。この式はこれ以上分解の必要がないので、部分式の分解においてシフト演算の数をカウントする必要はない。

よって、部分式は有限回で③、⑤の形に帰着する。故に、 $\mathbf{k}(\mathbf{x}) \in \mathbf{G}[\mathbf{x}]$ がいえろ。[証明終]