グラフィクスとCAD 21－7
（1986. 5. 9）

高速デジタルアフィン変換

市河 研一
NTT 電気通信研究所

2次元デジタル画像をソフトウェアで高速にアフィン変換する手法を提案する。ここでは2次元デジタル画像を同一長さ／幅を持った平行線分画像の集合とみなし、2次元画像に対するアフィン変換を1次元線分画像のアフィン変換に分解する。そしてこの線分画像のアフィン変換をDDA(Digital Differential Analyzer) のアルゴリズムを応用して、整数の加減算と符号の判定処理だけで高速に行う。

また本手法の応用例として、ラスター型グラフィックディスプレイにおけるGKSのセル配列プリミティブの出力機構への応用について述べる。

Fast Digital Affine Transformation

Ken-ichi ICHIKAWA
NTT Electric Communication Laboratories

This paper proposes a FAT (Fast Affine Transformation) algorithm for digital image data. Two-dimensional image data consists of multi-line image data which have the same length and width. An affine trnasformation of two-dimensional image data is divided into line image data transformation, and is executed quickly by DDA (Digital Differential Analyzer) algorithm, using integer addition and sign discrimination only.

It shows that the output mechanism of GKS cell array primitive is an application of this algorithm.

1．はじめに

2次元デジタル画像をソフトウェアで高速にアフィン変換する手法について提案する。

OA，CAD／CAMを始めとする各種の画像処理分野においては、画像データの編集やパターンマッチングを行うために、デジタル画像に対し拡大縮小，回転，平行移動等のアフィン変換を施す機能が必要とされており、その高速処理技術の開発が重要な課題となっている。

アフィン変換は、変換前の座標を（x，y），変換後の座標を（X，Y）とすると、3次同次座標表現を用いて、次の様な（3＊3）の変換マトリクスとして定義される。

$$\begin{vmatrix} X \\ Y \\ 1 \end{vmatrix} = \begin{vmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix} \quad \cdots\cdots\cdots (1\text{-}1)$$

$$a\,d - b\,c \neq 0 \quad \cdots\cdots\cdots (1\text{-}2)$$

一般にデジタル画像のアフィン変換では、変換後の画像データを決定するため、この変換マトリクスで定まる座標変換式に従って、各変換画素毎に原画像上の対応する位置を計算する必要があり、この全画素毎の座標変換処理が高速化のあい路となっていた。

この様な問題に対し、アフィン変換の中でも拡大縮小，回転といった基本的な変換について、問題の特殊性を利用した高速化手法が幾つか提案されている。 例えば、拡大縮小率（或いは回転に伴うx，y軸方向への変分率Δx，Δy）を原画像の座標データに逐次加算して行くことにより、画素毎の座標変換を省略する方法[2]～[4]，格子座標の周期性に着目して、変換に必要な数値を予めテーブルに格納しておくことにより、画像の拡大・縮小を高速化する方法[1] [5]，回転を2つの斜交軸変換に分解し、この斜交軸変換をテーブル参照により高速に行う手法[6] 等がある。

しかしこれらの方式では、拡大・縮小・回転等を合成した任意のアフィン変換を考慮した場合、与えられた変換を幾つかの基本変換（拡大縮小／回転，或いは2つの斜交軸変換）に分解し、原画像に対して多段階の変換処理を施さなければならない。またこれらの方式では、変換対象となる原画像が矩形領域であることを前提としているが、一般にアフィン変換後の画像領域は平行四辺形となるため、これに更に別のアフィン変換を施す場合を考えると、処理対象となる原画像の形状が制限される。

本論文の方式はこの問題を解決するものであり、2次元デジタル画像を同一長さ／幅を持った平行線分画像の集合と考え、2次元画像のアフィン変換を1次元線分画像のアフィン変換に分解し、線分画像のアフィン変換は原線分画像と出力線分画像の始点と終点の座標データに基づいて行う。そして変換対象となる線分画像の選択処理（原線分画像と出力線分画像の始点と終点座標の決定）を、原画像を定める平行四辺形の3頂点，及びそれをアフィン変換して得られる出力画像の3頂点の座標データに基づいて行う。これにより、任意の平行四辺形上に定義された画像データを、その形状，及び変換の性質に依らず、任意の変換マトリクスで定まる平行四辺形上へアフィン変換することが可能となる。

またこの時、第一列目の線分画像のアフィン変換，及び変換対象となる線分画像の選択処理はＤＤＡ(Digital Differential Analyzer) のアルゴリズムを応用して加減算のみで行い、第二列目以降の線分画像のアフィン変換は第一列目の変換処理で得られるデータに基づくテーブル参照で行うことにより、全体の処理の高速化を図っている。

以下、本論文で提案する高速アフィン変換の処理方式と、ＧＫＳのセル配列プリミティブ出力機構への応用例について報告する。

2．高速アフィン変換

2．1 処理概要

2次元デジタル空間において、同一直線上に存在しない3点(P_{oo},P_{on},P_{mo}) が、変換マトリクスＡで定まるアフィン変換により、これも同一直線上に存在しない3点(Q_{oo},Q_{ol},Q_{ko}) に変換されるものとする。（図1）
この時変換マトリクスＡは逆に、対応する3点の組合せ(P_{oo},P_{on},P_{mo}) (Q_{oo},Q_{ol},Q_{ko}) によって一意に定まるともいえる。

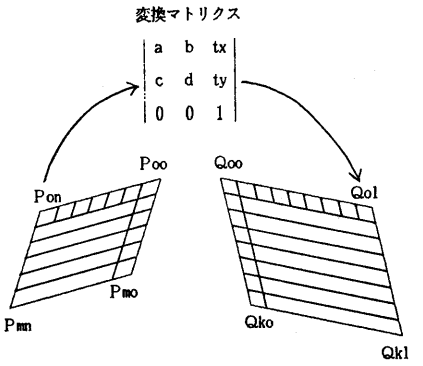

図1 デジタル画像のアフィン変換

これにより P_{mn}, Q_{kl} をそれぞれ P_{oo}, Q_{oo} に対する平行四辺形の対角点とすれば、２次元画像($P_{oo}, P_{on}, P_{mo}, P_{mn}$)は、２次元画像($Q_{oo}, Q_{ol}, Q_{ko}, Q_{kl}$)に写像される。

ところで２次元デジタル空間上に定義された平行四辺形の画像データは、図１に示す様に同一の長さ，及び同一の幅（１ピクセル）を持った１次元平行線分画像の集合と見做すことができる。

またアフィン変換には次の様な性質が在ることが分かっている[7]。

(a)線分（直線）は線分（直線）に変換される。

(b)２線分（直線）間の平行性は保存される。

(c)線分（直線）上の点の比は保存される。

そこでいまデジタル画像のアフィン変換を、出力画素に対応する原画素を求め、その画像データを出力画素に設定する処理と考える。すると２次元デジタル空間上で原画像($P_{oo}, P_{on}, P_{mo}, P_{mn}$)，とアフィン変換マトリクスＡが定義された場合、そのデジタルアフィン変換は以下の様な処理で行うことができる。

ｉ）変換マトリクスＡに基づき、アフィン変換後の($P_{oo}, P_{on}, P_{mo}, P_{mn}$)の座標($Q_{oo}, Q_{ol}, Q_{ko}, Q_{kl}$)を求める。 ……〔平行四辺形のアフィン変換〕

ii）原画像，出力画像とも、それぞれベクトル$\overrightarrow{P_{oo}P_{on}}$，$\overrightarrow{Q_{oo}Q_{ol}}$に平行な１次元線分画像の集合と考え、出力画像$\overline{Q_{Jo}\ Q_{J1}}$に対応する原画像$\overline{P_{io}\ P_{in}}$を求める。

……〔線分画像の対応付け〕

iii）対応する線分画像毎に、線分の長さの比に基づいて各画素毎の対応関係を求め、原画素の画像データを出力画素に転送する。

……〔線分画像のアフィン変換〕

そこで次に、１次元線分画像のアフィン変換と、対応する線分画像の選択処理に基づいた２次元アフィン変換について述べる。

２．２　線分画像のアフィン変換

始点Po終点Pmで定義される線分画像を、これまた始点Qo終点Qnで定義される線分画像へ写像するアフィン変換について考察する。

いま図２に示す様に、Poの中心がQoの中心へ，Pmの中心がQnの中心へ写像されるものとして、各々の座標データをPo(Xpo, Ypo), Pm(Xpm, Ypm)，及びQo(Xqo, Yqo), Qn(Xpn, Ypn) とする。（図2.太線□）

また線分画像Ｐ上の画素Piは線分画像Ｑ上の画素Qjに変換されるものとして各々の座標データをPi(Xpi, Ypi), Qj(Xpj, Ypj) とする。（図2.網）

ここで線分画像のアフィン変換を、Qjに対応するPiの画像データをQjに設定する処理と考えれば、問題は各線分画像の始点・終点の座標データに基づいてj=０～ｎに対するPi(Xpi, Ypi), Qj(Xpj, Ypj) を求めることに帰着される。

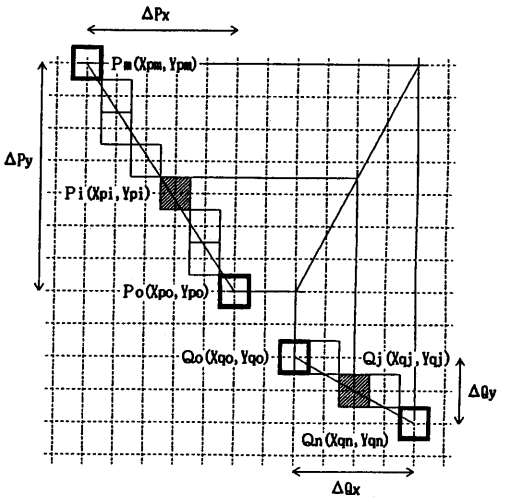

図２　線分画像のアフィン変換

……問　題……

ｊ＝0,1,・・ｎに対して、関数：

Xpi(j), Ypi(j), Xq(j), Yq(j)　　…… (2-1)

を求める。但し、

Xpi(0)＝Xpo　　Xpi(n)＝Xpm　　…… (2-2)

Ypi(0)＝Ypo　　Ypi(n)＝Ypm

Xq(0) ＝Xqo　　Xq(n) ＝Xqn

Yq(0) ＝Yqo　　Yq(n) ＝Yqn

とする。

本論文では(2-1) 式の値をＤＤＡを応用した線分生成アルゴリズムを用いて算出する。

図２において、出力画像の始点・終点間の距離のｘ及びｙ座標のうち大きい方が、出力画像の画素数(-1)、つまりポインタｊの移動回数に等しい。いまこの値をΔｊとして、Δｊ対する各線分画像のｘ，ｙ座標の距離の比を、始点座標に逐次加えることにより、j=0,1,・・ｎに対する(2-1) の値を順次計算する。

図２の例に於いて、Po～Pm，Qo～Qnの中心間の距離（但し、始点から終点の向きを正とする符号つき距離）をそれぞれΔPx，ΔPy，ΔQx，ΔQyとする。

ΔPx＝Xpm －Xpo　　… (2-3)

ΔPy＝Ypm －Ypo

ΔQx＝Xqn －Xqo

ΔQy＝Yqn －Yqo

またΔｊは出力画素数(-1)に等しいことから

$$\Delta j = \max\{|\Delta Qx|, |\Delta Qy|\} = n \quad \cdots\cdots (2\text{-}4)$$

である。

ここで画素Pi,Qj は格子点以外には存在しないことから、Xpi(j), Ypi(j), Xq(j), Yq(j)は整数値であることが必要である。よって(2-1) 式は以下の様になる。

$$Xq(j) = Xqo + \mathrm{sign}(\Delta Qx) \cdot \left(\frac{|\Delta Qx|}{\Delta j} j\right) \cdots (2\text{-}5)$$

$$Yq(j) = Yqo + \mathrm{sign}(\Delta Qy) \cdot \left(\frac{|\Delta Qy|}{\Delta j} j\right) \cdots (2\text{-}6)$$

$$Xpi(j) = Xpo + \mathrm{sign}(\Delta Px) \cdot \left(\frac{|\Delta Px|}{\Delta j} j\right) \cdots (2\text{-}7)$$

$$Ypi(j) = Ypo + \mathrm{sign}(\Delta Py) \cdot \left(\frac{|\Delta Py|}{\Delta j} j\right) \cdots (2\text{-}8)$$

※ j =0,1,・・n;

※ sign(α) はαの符号を表す。

※〔β〕はβを越えない整数とし、負数の場合は絶対値が小さい方へ丸めるものとする。

ところで上の(2-5) ～(2-8) 式の〔式〕の部分は、Δｆ＞0として次の形に一般化できる。

$$f(j) = \left(\frac{\Delta f}{\Delta j} j\right) \quad \cdots\cdots (2\text{-}9)$$

j =0,1,・・n

ここで f(0)=0, f(n)=fnとすれば、(2-10)は格子座標(j,f)上の2点(0,0),(n,fn)を結ぶデジタル線分を考えたとき、この線分上で整数ｊに対するｆの値を求めることに他ならない。そこでDDA の一手法であるBresenham の線分生成アルゴリズムを利用して(2-10)を計算することが考えられる。

〔0≦Δｆ≦Δｊ の場合〕

Bresenham のアルゴリズムはΔｊ とΔｆ のうち絶対値が大きい方の座標を一つづつ変化させ、もう一方の座標は真値に最も近いデジタル値を選ぶというやり方である。

(図3)

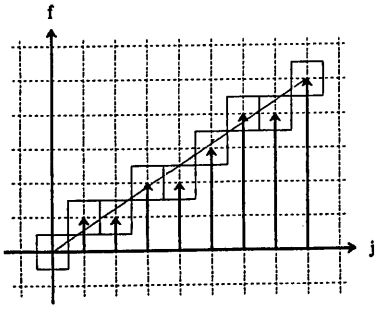

図3 Bresenham の方法 (Δf<Δj)

いま、

$$0 \leq \Delta f \leq \Delta j \quad (\Delta j \neq 0) \quad \cdots\cdots (2\text{-}10)$$

とすれば、ｊの方を１づつ増やすことになる。

ここで、

$$f'(j) = \frac{\Delta f}{\Delta j} j \quad \cdots\cdots (2\text{-}11)$$

として、j-1 におけるf'(j-1) の真値とf(j-1)との差をR'(j-1) とすれば、f(j)における差R'(j) は

$$R'(j) = R'(j-1) + \frac{\Delta f}{\Delta j} j \quad \cdots\cdots (2\text{-}12)$$

となり、

R'(j) ≦1/2 なら f(j)=f(j-1) ……(2-13)

R'(j) >1/2 なら f(j)=f(j-1)+1

のようにf(j)が決定される。

ところでこの時(2-13)式においてｆが１増やされるた場合、R'(j) は１だけ減ずる必要がある。

つまり(2-12)(2-13)は次式のようになる。

R'(0) =0 として、 ……(2-14)

$R'(j-1) + \frac{\Delta f}{\Delta j} \leq 1/2$ ならば、

$f(j) = f(j-1)$

$R'(j) = R'(j-1) + \frac{\Delta f}{\Delta j} j$

$R'(j-1) + \frac{\Delta f}{\Delta j} > 1/2$ ならば、

$f(j) = f(j-1) + 1$

$R'(j) = R'(j-1) + \frac{\Delta f}{\Delta j} j - 1$

ここで

$$R''(j) = R'(j) - 1/2 \quad \cdots\cdots (2\text{-}15)$$

と置き換えると(2-14)式中の判別処理，及び差分R"(j)の計算は次のようになる。

R"(0) =-1/2 として ……(2-16)

$R''(j-1) + \frac{\Delta f}{\Delta j} \leq 0$ ならば、

$R''(j) = R''(j-1) + \frac{\Delta f}{\Delta j} j$

$R''(j-1) + \frac{\Delta f}{\Delta j} > 0$ ならば、

$R''(j) = R''(j-1) + \frac{\Delta f}{\Delta j} j - 1$

また（Δｆ／Δｊ）が小数であることから、R"(j) の計算処理は実数演算となっているが、

$$R(j) = 2\Delta j * R''(j) \quad \cdots\cdots (2\text{-}17)$$

とおけば(2-16)式は、

R(0)＝－Δｊ として ……(2-18)

$R(j-1) + 2\Delta f \leq 0$ ならば、

$R(j) = R(j-1) + 2\Delta f$

R(j-1)＋2Δf＞0ならば、

$$R(j)=R(j-1)+2\Delta f-2\Delta j$$

となり、全ての計算処理を整数の加減算のみで行なうことができる。

以上をまとめると、$0\leqq\Delta f\leqq\Delta j$ の場合、(2-9) は次のような漸化式により整数の加減算と正負の判別処理のみで計算できる。

$$\Delta f=fn \quad : \quad \Delta j=n \qquad \cdots\cdots(2\text{-}19)$$

$$f(0)=0 \quad : \quad R(0)=-\Delta j$$

として、j＝0,1,・・nについて、

R(j-1)＋2Δf≦0ならば、 ……(2-20)

$$f(j)=f(j-1)$$

$$R(j)=R(j-1)+2\Delta f$$

R(j-1)＋2Δf＞0ならば、

$$f(j)=f(j-1)+1$$

$$R(j)=R(j-1)+2\Delta f-2\Delta j$$

〔$0<\Delta j<\Delta f$ の場合〕

ところで線分画像のアフィン変換Pi→Qjにおいて、線分の長さを縮める縮小変換を考えた場合、(2-7)(2-8)式においてΔPx，ΔPyの絶対値がΔjより大きくなる，すなわち(2-10)式においてΔf＞Δj となる場合があり、Bresenham の手法をそのまま用いると $\Delta f>\Delta j$ の場合一つのjに対して複数のfの値が存在することになる。(図4(a)) しかし、いま求めるものはjの各整数値に対するf'の真値に最も近いデジタル値fである (図4(b))。

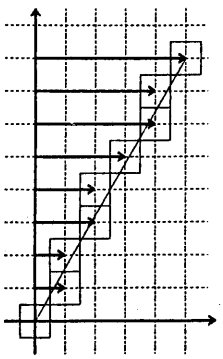

図4(a) Bresenham の方法 (Δf<Δj)

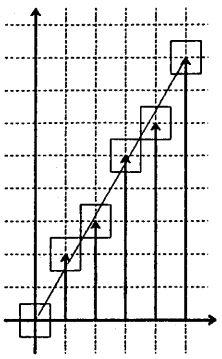

図4(b) 整数jに対するfのデジタル値

そこで本論文では、以下に示すBresenham の手法を拡張したアルゴリズムにより(2-9) 式を計算する。

原理を図5に示す。

いま、

$$\Delta f=\Delta j\cdot\Delta s+\Delta t \qquad \cdots\cdots(2\text{-}21)$$

（Δs，Δtは整数で $0\leqq\Delta t<\Delta j$）

とすると、

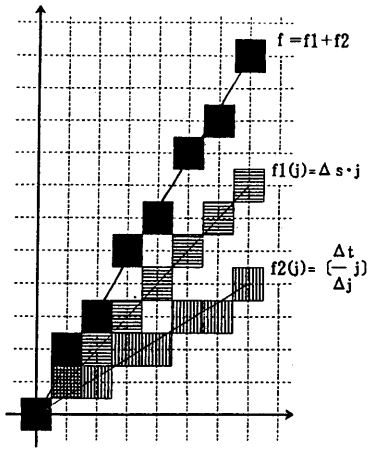

図5 Bresenham の拡張法

$$f(j)=\left[\Delta s\cdot j+\frac{\Delta t}{\Delta j}j\right]$$

$$=\Delta s\cdot j+\left[\frac{\Delta t}{\Delta j}j\right] \qquad \cdots\cdots(2\text{-}22)$$

である。ここで、

$$f1(j)=\Delta s\cdot j \qquad \cdots\cdots(2\text{-}23)$$

$$f2(j)=\left[\frac{\Delta t}{\Delta j}j\right] \qquad \cdots\cdots(2\text{-}24)$$

とおけば、(2-22)式は

$$f(j)=f1(j)+f2(j) \qquad \cdots\cdots(2\text{-}25)$$

となる。

いま、f1はj＝1,2,・・nに対して、

$$f1(0)=0 \qquad \cdots\cdots(2\text{-}26)$$

$$f1(j)=f1(j-1)+\Delta s$$

であり，またf2においては $0\leqq\Delta t<\Delta j$ であることから、これは(2-20)(2-21)式で $\Delta t\rightarrow\Delta f$ とした場合に相当する。よって $0<\Delta j<\Delta f$ の場合、(2-10)式は以下のように計算される。

Δs，Δtを

$$\Delta f=\Delta j\cdot\Delta s+\Delta t: \qquad \cdots\cdots(2\text{-}27)$$

$$0\leqq\Delta t<\Delta j \quad :$$

なる整数として、

$$f(0)=0 \quad : \quad R(0)=-\Delta j \qquad \cdots\cdots(2\text{-}28)$$

j＝0,1,・・nについて、

R(j-1)＋2Δt≦0ならば、

$$f(j)=f(j-1)+\Delta s$$

$$R(j)=R(j-1)+2\Delta t$$

R(j-1)＋2Δt＞0ならば、

$$f(j)=f(j-1)+\Delta s+1$$

$$R(j)=R(j-1)+2\Delta t-2\Delta j$$

ここで$0 \leq \Delta f \leq \Delta j$ の場合；即ち(2-19)(2-20)式は(2-27)(2-28)式で$\Delta s = 0$ (即ち$\Delta t = \Delta f$) とした場合に相当する。

よって (2-5)～(2-8) 式は(2-9) 式を

$$f(\Delta f, \Delta j, j) = \left[\frac{\Delta f}{\Delta j} j\right] \quad \cdots\cdots (2\text{-}29)$$

とすれば、

$$Xq(j) = Xqo \pm f(|\Delta Qx|, \Delta j, j) \quad \cdots\cdots (2\text{-}30)$$
$$Yq(j) = Yqo \pm f(|\Delta Qy|, \Delta j, j)$$
$$Xpi(j) = Xpo \pm f(|\Delta Px|, \Delta j, j)$$
$$Ypi(j) = Ypo \pm f(|\Delta Py|, \Delta j, j)$$

※±はそれぞれΔQx, ΔQy, ΔPx, ΔPyの符号((2-3)式) による。

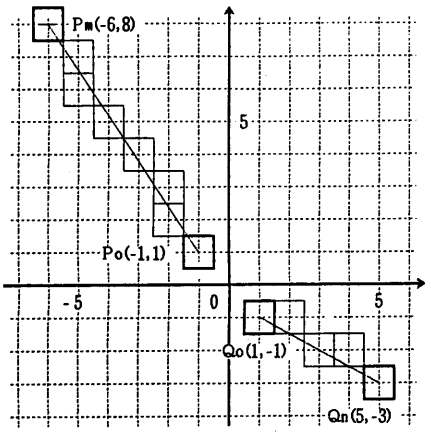

図6 線分画像のアフィン変換 — (例)

となり、(2-27)(2-28)式に基づき、整数の加減算と正負の判別処理のみで計算できる。

このとき、出力画像については$\Delta Qx = \Delta j$ ならば(2-5)式は単なるインクリメント処理となり、ΔQyについては$\Delta Qy \leq \Delta j$ となるから、(2-28)式ではなく(2-20)式によって計算処理を簡略化できる。

図6に、Po(x,y) =(-1,1), Pm(x,y) =(-6,8)
Qo(x,y) =(1,-1), Qn(x,y) =(5,-3)
とした場合について、線分画像のアフィン変換の例を示す。また、表1にj = 0～4に対するXpi(j),Ypi(j), Xq(j),Yq(j) の各パラメータの値，及びj-1 の値に対する相対移動量$\Delta Xq(j)$ ～$\Delta Ypi(j)$の値を示す。

表1 線分画像のアフィン変換

J	Px				Py			
	fpx	Rpx	Xp	ΔXp	fpy	Rpy	Yp	ΔYp
0	0	-4	-1	—	0	-4	1	—
1	1	-2	-2	-1	2	-6	3	2
2	2	0	-3	-1	3	0	4	1
3	4	-6	-5	-2	5	-2	6	2
4	5	-4	-6	-1	7	-4	8	2
J	Qx				Qy			
	fqx	Rqx	Xq	ΔXq	fqy	Rqy	Yq	ΔYq
0	0	-4	1	—	0	-4	-1	—
1	1	-4	2	1	0	0	-1	0
2	2	-4	3	2	1	-4	-2	1
3	3	-4	4	3	1	0	-2	0
4	4	-4	5	4	2	-4	-3	1

2．3 2次元デジタルアフィン変換

(1)線分画像の対応付け

図7において、原画像，出力画像ともに各々ベクトル$\overrightarrow{P_{oo}P_{on}}$, $\overrightarrow{Q_{oo}Q_{ol}}$に平行な線分画像の集合と見なした時の出力線分画像$|Q_{jo}\ Q_{jl}|$と原線分画像$|P_{io}\ P_{im}|$との対応付けについて考える。

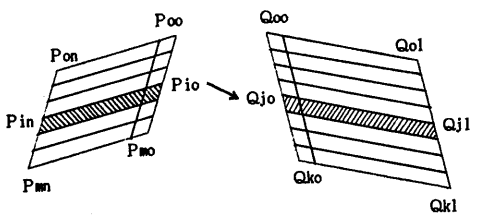

図7 線分画像の対応付け

図7の線分画像の対応付けにおいて、出力線分画像の始点が原線分画像の始点に対応することに着目すれば、アフィン変換の性質に基づき、線分画像の対応付けは各線分画像の始点の対応付け，即ち、始点の集合である線分画像$|P_{oo}\ P_{mo}|$ から$|Q_{oo}\ Q_{kd}|$へのアフィン変換により処理することができる。つまり2.2 章の方式により線分画像の対応付けもまた、全て整数の加減算と正負の判別処理のみで計算できることになる。

(2)第2列目以降の線分画像のアフィン変換

平行四辺形で定義されたデジタル画像を構成する線分画像は、原画像・出力画像ともに平行で且つ長さが等しいことから、これらは始点座標の位置を別にすれば、全て同一長さ・傾きの線分画像と見做すことができる。

そこで第一列目の線分画像のアフィン変換を行った時に、表1に示すようにXq(j),Yq(j),Xpi(j),Ypi(j) のj-1の値に対する相対移動量をテーブル情報として格納しておけば、第2列目以降のアフィン変換は、2.3 (1)で求めた各始点座標の値に、このテーブル情報の値を順次加えることにより行うことができる。

即ち、第2列目以降のアフィン変換はテーブルの参照と整数の加算のみで行うことができ、全体の処理を更に高速化することが可能となる。

図8に本手法による画像データの変換出力の処理結果を示す。

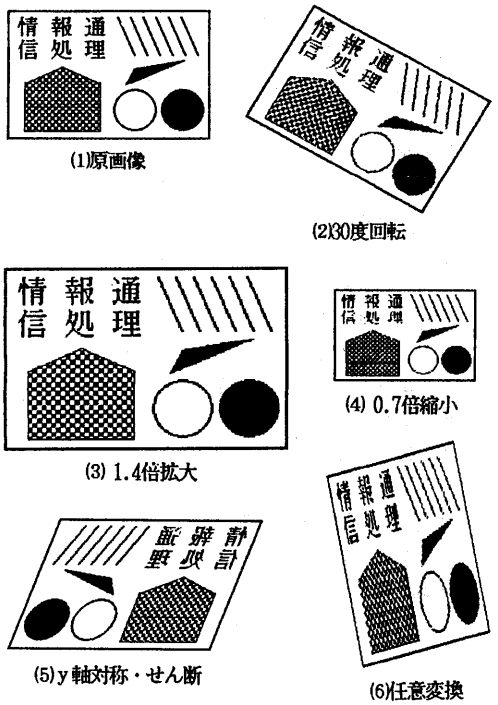

図8　画像データの高速アフィン変換

3．GKSセル配列プリミティブ

GKSのセル配列プリミティブを、配列の形で定義された2次元画像データとみなせば、そのラスター型グラフィックディスプレイへの出力は、ピクセルで定義されたデジタル空間上へのアフィン変換と考えることができる。そこでここでは、高速デジタルアフィン変換の一つの応用例として、GKSのセル配列プリミティブの出力処理について述べる。

3．1　セル配列のディスプレイへのマッピング

図形処理の国際標準仕様であるGKS(Graphical Kernel System)では、図形の基本出力要素（プリミティブ）として、線分列，多角形とともに、色インデックスをその要素とする2次元配列データによって、画像データをセル配列プリミティブという形で規定している[8]。図9にセル配列のマッピングルールを示す。

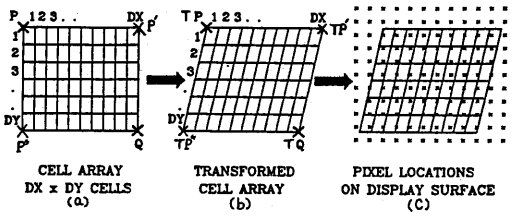

図9　セル配列のマッピング

GKSのセル配列プリミティブは、図9(a)に示すように、プリミティブの形状を定める長方形の左上と右下の座標点P／Q，画像データの横及び縦方向の画素数を定めるDX／DY，そして各画素の表示色を定めるカラーインデックスの2次元配列によってワールド座標上に定義される。ワールド座標上のセル配列はGKSの規定する正規化変換／セグメント変換／ワークステーション変換を経て、ディスプレイ固有のデバイス座標上の図形へと変換される（図9(b)）。デバイス座標上で定義された各セルは、図9(c)に示すようにディスプレイの各ピクセルの位置と大きさの関係から、次に示すルールに従って表示される。

(a)プリミティブの形状を定める長方形が変換されてできた平行四辺形の内側にピクセルの中心がある場合、当該ピクセルがセル配列を表示するものとして表示色が設定される。

(b)表示対象となる各ピクセルは、そのピクセルの中心点が含まれるセルの色とする。この時、ピクセルの色はあくまでも各セルの色をピクセルの中心座標によりサンプリングするのであって、ピクセル及びセルの大きさに基づく領域単位のサンプリングやフィルタリング等の処理は行わない。

(c)セル配列プリミティブの出力にあたっては、最低レベルのシミュレーションとして、各ディスプレイに依存した色，線幅，線種の線分列により、セル配列の表示領域を示すこと。

GKSでは上記の様に各セルのピクセルへのマッピングルールを定めているが、これをそのまま実際のディスプレイにインプリメントすると、各ピクセル単位にその

中心がどのセルに含まれるかを判定するために、精密な座標変換とそのデータに基づく判別処理が必要となり、ＣＡＤ等のインタラクティブな図形処理で求められる高速処理を実現するのが困難となる。

そこで、ここでは先に提案した高速デジタルアフィン変換の手法を用いて、セル配列プリミティブの出力を高速に処理する。

3．2　高速アフィン変換の応用

先ずＧＫＳが規定する座標変換が、いずれも(1-1) 式で定義されるアフィン変換であることに着目して、ワールド座標からデバイス座標への変換を定める合成変換マトリクスＡを求める（図10）。

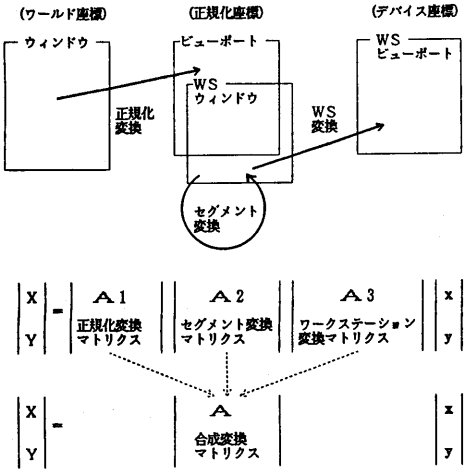

図１０　ＧＫＳにおける座標変換

いま図９において、プリミティブを定義する長方形のP,Q 以外の対角点をP'P"とし、P,P',P" の３点を合成変換マトリクスＡにより変換したデバイス座標上の点をTP,TP',TP"とする。ここでセル配列プリミティブは、各セルを画素とした２次元デジタル画像と考えることができる。またこの時、縦横の画素数DX,DY が十分大きいときP,P',P" 及びQ の座標点をプリミティブの四隅のセルの中心座標点と見なせる。これより、セル配列プリミティブの出力を、P,P',P" で定まる平行四辺形から、デバイス座標上のTP,TP',TP"で定まる平行四辺形へのデジタル画像のアフィン変換として処理する。

これにより、セル配列プリミティブの出力を整数の加減算とテーブル参照だけで高速に処理することが可能となる。特に、原画像の形状が長方形であることに着目すれば、第一列目のアフィン変換，及び線分列の選択処理において、(2-30)式の第一式または二式を省略（定数となる）することが可能であるから、なお一層の高速化を図ることができる。

4．おわりに

２次元デジタル画像を１次元平行線分画像の集合と見なし、これに対するアフィン変換を、ＤＤＡのアルゴリズムを応用して整数の加減算とテーブル参照により、高速に行う手法を提案した。本手法は次の様な特徴を有する。

（ｉ）変換マトリクスに従って、予め原画像を定める平行四辺形の３頂点をアフィン変換し、以後の処理はこの原画像，及び出力画像を定める３頂点の座標データだけに基づいて行う。これにより、拡大縮小・回転・平行移動といった変換の性質，及び変換対象となる画像データの形状を限定することなく、２次元デジタル画像をアフィン変換できる。

（ii）画素ごとに座標変換を行う従来方式では、各画素毎に実数演算でそれぞれ４回の加算と乗算を必要としたのに対し、本方式における各画素毎の処理としては、テーブル参照と４回の整数加算で済むため、全体として処理の高速化が可能となる。この時、全体の処理時間は、出力画像の大きさ、即ち画素数に比例する。

また応用例としてＧＫＳのセル配列プリミティブの出力処理を取り上げたが、本方式はＯＡにおけるイメージデータの編集処理等，各種の画像処理分野に適用することができる。

今後は、画素毎に座標変換を行う方式を基準として、本手法を用いて変換した場合の各画素の位置ずれ（誤差）の評価，及び出力処理における性能評価等の検討を行う予定である。

〔参考文献〕

(1)正嶋他：二値画像の各種拡大／縮小方式の性能評価および処理速度改良方式，情処学論,1985 Vol.26 No.5
(2)宮田他：２値画像の拡大縮小方式，信学技報（画像工学),IE84-98
(3)田畑他：格子座標の周期性を利用した画像拡大縮小の高速処理方式，情処学論,1983 Vol.24 No.5
(4)田畑他：ラスタ走査とテーブル参照による画像回転の高速処理，電子通信学会論文誌,1986/1Vol.J69
(5)川上他：画像プロセッサMN8614の処理方式，情報処理学会計算機アーキテクチャ研究会資料，昭和58-11，51-4
(6)川上他：事務処理分野をねらった２値画像プロセサＬＳＩ，NIKKEI ELECTRONICS 1983.12.19
(7)山口：コンピュータディスプレイによる図形処理工学，昭和56年，日刊工業新聞社
(8)"Graphical Kernel System(GKS) Functional Description",ISO 7942,Aug.1985