論文93-30B-1-1

행렬·벡터 연산용 1-차원 시스톨릭 어레이 프로세서를 이용한 그래픽 가속기의 설계

(Design of a Graphic Accelerator using 1-Dimensional Systolic Array Processor for Matrix · Vector Opertion)

金 龍 成*, 趙 源 敬*

(Yong Sung Kim and Won Kyung Cho)

要 約

최근 컴퓨터 그래픽은 고화질의 그래픽 카드를 이용한 CAD 및 시뮬레이터등에 널리 사용되어지고 있으므로 고성능의 그래픽 처리가 요구된다. 본 논문에서는 그래픽의 기본요소(선분, 원 및 타원)의 발생 및 변환기능을 갖는 그래픽 가속기를 설계한다. 이러한 그래픽 알고리듬들은 행렬 벡터 연산에 의해 수행할 수 있으므로 본 논문에서는 1차원 시스톨릭 어레이 프로세서(1-D Systolic Array Processor)를 설계하고 이를 그래픽 가속기의 주 연산장치로 사용한다. 결과적으로 그래픽 도면상에 해상도가 800×600인 경우 33.3nsec의 연산기를 사용할 때 초당 29732개의 선분 및 초당 약 6244개의 원을 발생 시킬 수 있었다.

Abstract

In recent days high perfermance graphic operation is needed, since computer graphics is widely used for computer-aided design and simulator using high resolution graphic card. In this paper a graphic accelerator is designed with the functions of graphic primitives generation and geometrical transformations. 1-D Systolic Array Processor for Matrix · Vector operation is designed and used in main ALU of a graphic accelerator, since these graphic algorithms have common operation of Matrix · Vector. Conclusively , in case that the resolution of graphic domain is 800×600, and 33.3nsec operator is used in a graphic accelerator, 29732 lines per second and approximately 6244 circles per second is generated .

Ⅰ. 서 론

컴퓨터 그래픽은 최근 공학분야의 설계 및 모의실험, 상업용 광고 등 여러분야에 다양하게 응용되고 있다. 컴퓨터를 이용한 CAD 장비 및 윈도우즈 환경 등에서 필요한 고해상도의 그래픽 카드가 발전됨에 따라 성능의 향상을 위하여 그래픽의 기본적인 기능을 고속으로 수행하기 위한 고속의 프로세서가 요구된다. 그래픽 도형 발생의 기본요소인 선분, 원 및 타원의 발생은 다각형, 원호, 타원등의 기본적인 그래픽 도형의 형성 및 벡터 폰트 생성등 다양한 응용분야에 필수적인 요소이므로 선분 및 원 도형을 구현할 수 있는 고속의 도형 발생기가 요구된다. 기존의 선분 및 원 발생의 고속화 알고리듬은 매개변수적 알

* 正會員, 慶熙大學校 電子工學科
(Dept. of Elec. Eng., Kyunghee Univ.)
接受日字:1992年 9月 25日

고리듬과 비매개변수적인 알고리듬으로 구분되며, DDA(Digital Differential Analyser)알고리듬[1], Bresenham의 중점을 이용한 도형발생 알고리듬[2], Jordan의 위치비교 알고리듬[3], Suenage의 수정된 위치비교알고리듬[4] 등이 있다. 그러나 이러한 알고리듬은 선분(원)을 구성하는 한 점(화소점)을 발생할 때마다 일정한 판정기준에 의하여 다음 점의 증가 또는 감소를 수행하며, 이를 반복 수행하므로 고속의 전용 프로세서 설계시 규칙성 및 크립핑(Clipping)에 대한 문제점을 갖는다. 그러므로 본 논문에서는 이러한 문제점을 해결하기 위하여 기하학적인 변환 및 도형 발생에 대한 기본 알고리듬의 연산 방식이 행렬·벡터의 형태로 표현 가능함을 나타내고, 규칙성, 파이프라인 처리(Pipeline processing) 및 병렬성의 특징을 갖는 행렬·벡터 연산용 시스톨릭 어레이 프로세서를 주 연산장치로 사용하고자 한다. 또한 연속적인 도형의 발생을 위하여 초기조건부를 설계하며, 기본알고리듬의 수정된 표현 및 [M×N] · [N] ($1 \le M \le N$, $1 \le N \le 4$)과 [N/2 x N] · [N] ([N/2] ≥1)의 연산이 가능한 고속의 행렬·벡터 연산용 1차원 시스톨릭 어레이 프로세서를 설계함으로써 그래픽의 기본요소(graphic primitive)중 선분 및 원,원호,타원 발생과 그래픽의 기하학적 변환기능을 갖는 그래픽 가속기를 설계하고자 한다.

Ⅱ. 기하학적 변환

생성된 도형을 응용분야에 따라 위치의 재배치, 윈도우 내에 알맞는 도형으로 확대 및 축소가 요구되는 경우 이동(Translation), 신축(Scailing), 회전(Rotation)등의 기본적인 기하학적 변환방법[5]이 필요하며, 이러한 변환방법은 행렬·벡터의 형태로 표현 가능하다. 또한 변환된 결과를 임의의 좌표점에 위치시키거나 현재 도형의 임의의 점을 기준으로 위치시키고자 하는 경우 행렬·벡터 형태로 표현하면 두번의 변환 과정이 단일 행렬·벡터 형태로 표현되므로 첫번째 변환한 결과를 두번째 변환에 입력하는 과정이 생략되기 때문에 연산과정의 복잡도를 줄일 수 있다. 이러한 복합변환을 행렬·벡터 형태로 표현하면 다음과 같다.

1. 신축과 이동 변환

2차원 x-y 평면에서 도형의 한점 P(x,y)에 대하여 신축요소 S_x,S_y만큼 신축한 결과를 각 좌표축에 따라 D_x, D_y만큼 이동하였을 경우 변환된 결과를 P′(x′,y′)이라 하면,

$$P=[x\ y],\ P'=[x'\ y'],\ S=[Sx\ Sy],\ T=[Dx\ Dy]$$

일때

$$P'=[x'\ y']=[x\ y]\cdot\begin{bmatrix}Sx & 0\\ 0 & Sy\end{bmatrix}+[Dx\ Dy]$$

이므로

$$P'=\begin{bmatrix}x'\\ y'\end{bmatrix}=\begin{bmatrix}Sx & 0 & Dx\\ 0 & Sy & Dy\end{bmatrix}\cdot\begin{bmatrix}x\\ y\\ 1\end{bmatrix} \quad (1)$$

로 표현된다.

2. 회전과 이동 변환

2차원 x-y 평면에서 도형의 한점 P(x,y)를 각 θ만큼 회전하고, 회전된 점을 각 좌표축에 따라 D_x,D_y만큼 이동하였을 경우, 변환된 결과를 P(x′,y′)이라 하면,

$$P=[x\ y],P'=[x'\ y'],R=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix},T=[Dx\ Dy]$$

일때

$P'=[x'\ y']=[x\ y]\cdot\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}+[Dx\ Dy]$ 은

$$P'=\begin{bmatrix}x'\\ y'\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta & Dx\\ -\sin\theta & \cos\theta & Dy\end{bmatrix}\cdot\begin{bmatrix}x\\ y\\ 1\end{bmatrix} \quad (2)$$

로 표현되므로 식 (1), (2)에 의하여 두번의 변환 과정이 각각 단일 행렬·벡터 형태로 표현 가능하다.

Ⅲ. 그래픽 기본 요소의 발생

그래픽 도면에 있어서 형성되는 모든 도형들은 점 또는 화소가 점의 연결에 의해서 이루어진다. 이점들의 자취가 직선 또는 원을 이루고 있으며, 허용된 오차범위 내에 있는 근사화된 선 또는 원인 경우 이를 그래픽 상에서의 선분 또는 원이라 할 수 있다. 그래픽 기본요소중 선분, 원의 발생법[5]은 다음과 같다.

1. 선분 발생

선분의 양 끝점을 P_1, P_2라고할 때 두 끝점에 의해 생성되는 선분의 기본식은 식(3)과 같다. 두점 P1, P2의 좌표가 (X_1,Y_1), (X_2,Y_2) 일때

$\Delta X = X_2 - X_1$, $\Delta Y = Y_2 - Y_1$

$a = dy = \Delta Y/\Delta X$, $dx_{i+1} = \Delta X/n$(n:선분 구성점의 수)

$I = f(x_1) = y_1$

$Y_i = f(x_1 + dx_i) = a \times dx_i + I \ (0 \le dx_i \le \Delta X)$ (3)

위의 직선발생식을 식 (4)와 같이 행렬·벡터 ([4×4] · [4])의 형태로 실행할 경우 선분상의 점을 3개씩 발생시킬 수 있다.

$$\begin{bmatrix} Y_i \\ Y_{i+1} \\ Y_{i+2} \\ 0 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & I \\ 0 & a & 0 & I \\ 0 & 0 & a & I \\ 0 & 0 & 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} d_{Xi} \\ d_{Xi+1} \\ d_{Xi+2} \\ 1 \end{bmatrix} \quad (4)$$

2. 원 발생

원 발생은 다음의 두가지 형태로 나눈다. 연속적인 원 도형의 자취를 생성하는 방법과 1/8원의 대칭성 이용하여 전체 원을 형성하는 방법이 있다.

원의 발생식은 다음과 같다.

$(X-C_x)^2+(Y-C_y)^2=R^2$

또는 $X=R\times\cos(\theta)+C_x$, $Y=R\times\sin(\theta)+C_y$ (5)

(R: 반지름, (C_x, C_y):원의 중점의 좌표)

1) 1/8원을 이용한 원의 발생

원을 그림 1과 같이 원의 중심각 θ를 π /2씩 분할하면 분할된 각에 따라 4개의 구간으로 분할되고 원호는 직선 X-X′, Y-Y′ 에 대하여 서로 대칭되며, 각 구간을 다시 π /4씩 분할하면 선분 O-P1, O-P2, O-P3, O-P4에 대하여 1/8원이 서로 대칭된다.

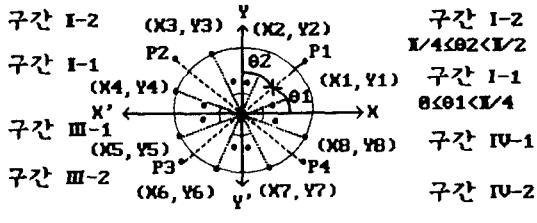


그림 1. 원의 8개 대칭점

Fig. 1. Eight symmetrical points on a circle.

즉 θ1은 양의 방향으로 θ_2는 음의 방향으로 증가할 때 $\theta=\theta_1=\theta_2$ 이면 구간 I-1에서 θ_1에 의해서 발생된 원의 자취 $X=R\times\cos(\theta_1)+C_x$, $Y_1=R\times\sin(\theta_1)+C_y$ 는 구간 I-2에서 θ_2에 의해서 발생된 원의 자취 $X_2=R\times\sin(\theta_2)+C_y$, $Y2=R\times\cos(\theta_2)+C_x$와 대응되므로 $X_1=Y_2$, $Y_1=X_2$의 관계가 성립한다. 이때 발생된 원의 자취는 다른 구간의 원의 자취와 절대값은 동일하며 부호만이 다른 대칭점이 생성된다. 식 (6)에 1/8원을 이용한 원 발생식을 행렬 벡터의 형태로 나타내었으며, 각 1/4원마다 벡터값 식 (7)이 식 (6)에 도입된다.

$$\begin{bmatrix} X1 \\ Y1 \\ X2 \\ Y2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & Cx & 0 \\ 0 & \sin\theta & 0 & Cy \\ \sin\theta & 0 & Cx & 0 \\ 0 & \cos\theta & 0 & Cy \end{bmatrix} \bullet I \quad (6)$$

$$I = \begin{bmatrix} R \\ R \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -R \\ R \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -R \\ -R \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} R \\ -R \\ 1 \\ 1 \end{bmatrix} \quad (7)$$

구간Ⅰ 구간Ⅱ 구간Ⅲ 구간Ⅳ

Ⅳ 그래픽 가속기의 구성

기하학적인 변환 및 그래픽 기본요소의 Ⅲ발생시의 식 (4), (6)의 행열·벡터의 연산을 수행하므로 기본적인 연산 프로세서는 동일한 구조의 행렬 벡터 연산용 시스톨릭 어레이 프로세서로 설계가 가능하다. 또한 각각의 초기조건이 공통성이 있으므로 효율성있는 가속기를 설계하기 위하여 초기조건을 동일구조로 수행할 수 있는 하드웨어로 설계되어야 한다. 그러므로 이절에서는 위와 같은 구조로 수행할 수 있는 그래픽 가속기의 설계, 시스톨릭 어레이 프로세서의 수정된 설계 및 초기조건 생성부를 설계 한다.

1. 그래픽 가속기의 전체 구성도

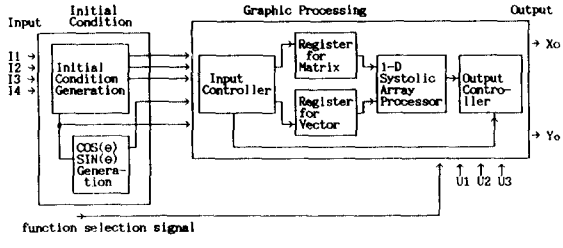


그림 2. 그래픽 가속기의 구조

Fig. 2. Structure of Graphic Accelerator.

본 논문에서 설계한 그래픽 가속기의 전체구조를 그림 2에 나타내었으며 2개의 구성요소로 구분된다. 초기조건 발생부는 처리될 요소에 따른 초기조건 생성부와 삼각함수 처리기로 구성되고, 그래픽 처리부는 각각의 처리요소에 따른 입출력 데이타의 제어부와 주 연산장치인 1차원 시스톨릭 어레이 프로세서로 구성된다. 각각의 기능은 기능선택신호에 따라 선택되며, U1,…,U3는 각 그래픽 기능에 따라 고정된 값이 입력된다. (예: 선분의 경우 U1=a, U2=I, 원의 경우 U1=R, U2=C_x, U3=C_y)

2. 시스톨릭 어레이 프로세서의 구조

시스톨릭 어레이 프로세서는 파이프라인 처리(Pipeline Processing)와 병렬연산처리의 특성을 가

지고 있으며,구조가 단순하고 규칙적인 설계가 가능하다. 일반적인 행렬 · 벡터 연산인 경우 4×4행렬 A와 4×1 열벡터 B의 곱(C=A · B) 연산용 1차원 시스톨릭 어레이 프로세서[6]의 구조는 그림 3과 같다.

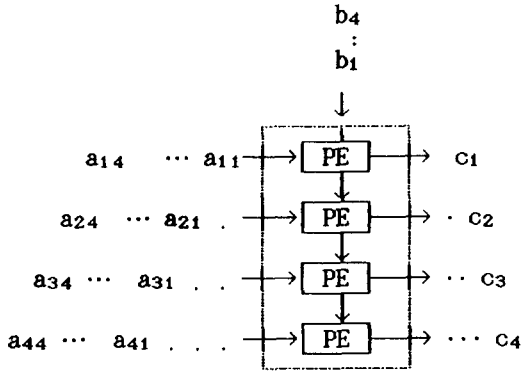


그림 3. 1차원 시스톨릭 어레이 프로세서 (PE=4)

Fig. 3. 1-D Systolic Array Processor (PE=4).

3. 그래픽 기본요소의 연속적인 발생을 위한 시스톨릭 어레이 프로세서의 설계

그림 3의 시스톨릭 어레이 프로세서는 [N×N] · [N] (N=4)의 행렬 벡터 연산만을 수행할 수 있으며, 각 PE마다 출력이 생성되므로 출력된 데이타의 전송시 부가적인 회로가 요구된다. 그러므로 본 논문에서는 기하학적인 변환과정과 기본도형 발생시 [M×N] · [N] (1≤M≤N)의 연산 및 [N/2×N] · [N] ([N/2] ≥1)의 연산이 연속적으로 출력되게 하기 위하여 N = 4인 경우 시스톨릭 어레이 프로세서를 그림 4와 같이 설계한다. 그림 4에 나타낸 연산부는 시스톨릭 어레이 프로세서, 행렬 Rg (Register), 벡터 Rg로 구성되며, 어레이 프로세서에서는 [N×M] · [N] (1≤M≤4,1≤N≤4)의 연산이 가능하도록 하기 위하여 제어신호에 의해 출력이 조정된다. 또한 [N/2×N] · [N] 의 행렬 · 벡터 연산결과가 연속해서 출력되게 하기위하여 MUX를 이용하여 PE의 벡터입력을 분리시켜 두개의 벡터입력을 선택할 수 있게 함으로써 두개의 [2×4] · [4] (N=4)연산이 가능한 구조로 설계하였다. 행렬 입력으로는 레지스터 (Rg0~Rg3)를 사용하고 벡터입력으로는 RgH와 RgL을 사용한다. 예를 들면 [4×4] · [4] 의 연산인 경우 RgH만을 벡터입력으로 사용하고, 두개의 [2×4] · [4] 의 연산인 경우 RgH, RgL을 벡터입력으로 사용한다.

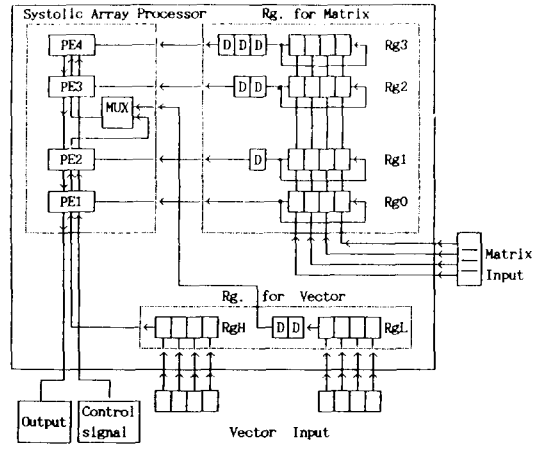


그림 4 .그래픽 기본요소의 발생을 위한 행렬 · 벡터 연산용 시스톨릭 어레이 프로세서의 구조 (N = 4)

Fig. 4. Structure of 1-D Systolic Array Processor for Matrix · Vector operation to generate graphic primitive (N=4).

4. 행렬 · 벡터 연산용 시스톨릭 어레이 프로세서의 PE설계

그림 3에서 설계한 시스톨릭 어레이 프로세서의 k번째 PE를 그림 5와 같이 구성한다.

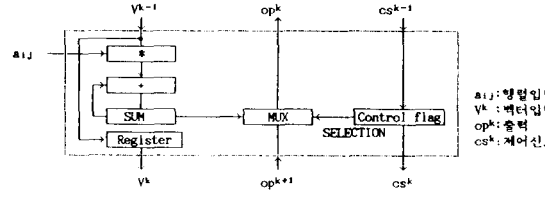


그림 5. 행렬 벡터 연산용 시스톨릭 어레이 프로세서의 PE설계

Fig. 5. PE design of 1-D Systolic Array Processor for Matrix · Vector operation.

그림 3의 구조에서 각 PE마다 출력이 생성되는 문제를 해결하기 위하여 MUX 및 N 클럭마다 발생되는 제어신호를 사용하여 각 PE의 출력 결과가 동일 버스를 통하여 전달되도록 하며, [N×M] · [N] (1≤M≤4,1≤N≤4)의 행렬 벡터 연산을 가능하도록

설계한다.

5. 그래픽 가속기의 초기조건 생성부 설계

초기조건 생성부에서는 도형발생시 초기 입력에 따라 도형발생에 필요한 초기값 및 연속적인 입력 값을 생성한다. 본연구에서 설계한 그래픽 가속기의 초기조건 생성부를 그림 6과 같이 설계한다.

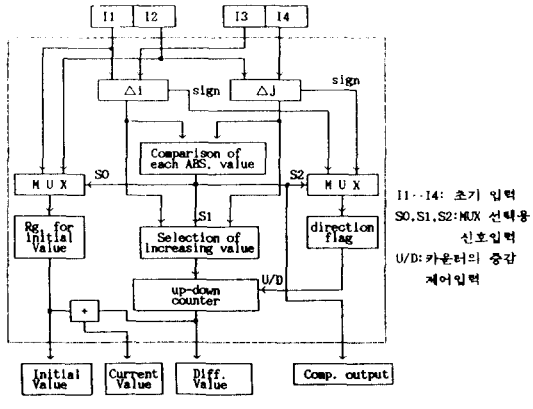


그림 6. 그래픽 기본요소 발생기의 초기조건 생성부

Fig. 6. Initial Condition Generation of graphic primitive.

초기 입력 I1, I2, I3, I4에 따라 Δi=I1-I2와 Δj=I3-I4를 구하며,비교기에 의해 두값의 절대치를 비교한 결과는 다음 세경우의 선택신호로 사용한다. Δi, Δj중 절대값이 큰쪽을 카운터에 입력하게 하며, I1,I2입력중에서 초기값이 선택되게 한다. 또한 Δi, Δj의 부호를 선택하여 카운터의 증감을 결정하게 한다. 이에 따라 초기조건 생성부에서는 초기값, 증감값, 초기값과 증감값이 가산된 현재값 및 비교기의 결과가 출력된다.

Ⅴ. 설계된 프로세서를 이용한 그래픽 기본연산 알고리듬

그래픽 기본요소 및 기하학적 변환시 [4×4] · [4] 행렬·벡터용 시스톨릭 어레이 프로세서를 사용하는 경우, 연속적인 도형발생 및 변환이 이루어지려면 출력에서 발생되는 널(Null) 데이타 및 행렬, 벡터용 데이타를 입력할때 마다 발생되는 소요시간이 감소되어야 하므로 본 논문에서는 다음과 같이 기본연산 알고리듬을 변환한다.

1. 기하학적 변환

Ⅲ의 식 (1)과 (2)에서와 같이 복합변환을 수행하는 경우 그림 3의 행렬 벡터([4×4] · [4])의 시스톨릭 어레이 프로세서를 사용하면 1개 점 변환시 2사이클(cycle)의 널(Null) 데이타를 출력하게 된다. 그러므로 식 (1), (2)를 2개의 변환점이 연산되도록 2개의 [2×4] · [4] 의 행렬 벡터 연산으로 나타내면 신축과 이동 변환은 식 (8)로 ,회전과 이동 변환은 식 (9)로 표현된다.

$$\begin{bmatrix} x1' \\ y1' \end{bmatrix} = \begin{bmatrix} Sx & 0 & Dx & 0 \\ 0 & Sy & 0 & Dy \end{bmatrix} \bullet \begin{bmatrix} x1 \\ y1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x2' \\ y2' \end{bmatrix} = \begin{bmatrix} Sx & 0 & Dx & 0 \\ 0 & Sy & 0 & Dy \end{bmatrix} \bullet \begin{bmatrix} x2 \\ y2 \\ 1 \\ 1 \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} x1' \\ y1' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & Dx & 0 \\ sin\theta & cos\theta & 0 & Dy \end{bmatrix} \bullet \begin{bmatrix} x1 \\ y1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x2' \\ y2' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & Dx & 0 \\ sin\theta & cos\theta & 0 & Dy \end{bmatrix} \bullet \begin{bmatrix} x2 \\ y2 \\ 1 \\ 1 \end{bmatrix} \quad (9)$$

식 (8),(9)에서 그림 4의 수정된 시스톨릭 어레이 프로세서에 적용시 두개의 행렬은 Rg0~Rg3에 입력하고 두개의 벡터는 RgH와 RgL에 입력함으로써 연속적인 변환 점이 발생되도록 할 수 있다.

2. 그래픽 기본 요소의 초기조건 및 연속된 도형 발생을 위한 기본식의 수정

1) 선분 및 다각형 발생

Ⅲ의 식 (3)의 경우 선분발생시 기울기에 따라 선분밀도의 문제점을 갖는다. 예를들면 I=0일 때 선분과 X축이 이루는 양의 각이 π /8인 선분과 3π /8인 선분은 동일한 길이를 갖는 선분이지만, X축과 π /8를 이루는 선분이 밀도가 높게 그래픽 도면에 나타난다. 이를 보완하기 위하여 다음과 같이 선분을 발생한다.

그림 7에서 선분 P1-P2의 기울기가 a≥0인 경우

X축과 평행인 반직선 O′-A와 선분 O′-P2가 이루는 양의 각 θ가 $0\leq\theta<\pi/4$, $\pi/4\leq\theta<\pi/2$인가에 따라 구간 Ⅰ과 구간 Ⅱ로 분할하며, 선분 P1-P2가 구간 Ⅰ에 속하면 시작점 P1에서 P2까지 X값의 증가에 따라 선분 O′-P2의 Y좌표점을 구하고, 구간 Ⅱ에 속하면 시작점 P1에서 P2까지 Y값의 증가에 따라 선분 O′-P2의 X좌표점을 구하여 선분을 발생한다. 기울기가 a<0인 경우 반직선 O′-B′와 선분 O′-P3가 이루는 양의 각을 θ라할 때에도 기울기가 a≥0인 경우와 동일하게 적용된다.

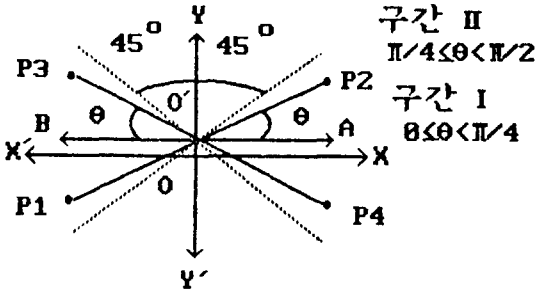


그림 7. P1-P2, P3-P4의 선분발생
Fig. 7. Line drawing of P1-P2, P3-P4.

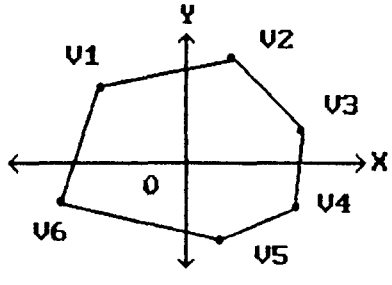


그림 8. 6개의 정점을 갖는 다각형
Fig. 8. Polygon with 6 vertics.

즉 $|\Delta X|\geq|\Delta Y|$일 때는 X값의 변화에 따라 Y좌표점을 구하고, $|\Delta X|<|\Delta Y|$인 경우 선분은 Y값의 변화에 따라 X좌표점을 구하여 선분을 발생한다. 그러므로 본 논문에서는 ΔX와 ΔY 의 대소비교에 따라 초기치 및 기울기의 판정을 하여 선분을 발생시키고자 한다.

또한 다각형 도형의 발생은 선분발생의 확대된 개념이므로 그림 8과 같이 6개의 정점(Vi: i=1,…,6)으로 구성되는 6각형의 경우 선분발생기의 두 입력점을 Pm, Pm+1이라 하면 Pm=Vi, Pm+1=Vi+1(i=1,…,6 and if i=6 then i+1=0)와 같이 연속적으로 정점값을 입력하여 다각형을 발생시킬 수 있다.

(1) 선분 발생시 초기조건

선분의 양끝점 P1, P2의 좌표를 (X1, Y1), (X2, Y2)라할 때 선분발생을 위한 초기 조건은 다음과 같다.

① ΔX=X2-X1, ΔY=Y2-Y1, comp=|ΔX|-|ΔY| 이며,

② 입력 초기치 I는 if comp≥0 then I=X1 else I=Y1이다.

③ 기울기 판정은 if comp≥0 then a= ΔY/ΔX else a = ΔX/ΔY 이다.

④ 선분발생 정지기준은 다음과 같다.

입력의 증감치 D는 if comp≥0 then D=|ΔX| else D=|ΔY|

증감의 조건 d는 if comp≥0 then d= sign of ΔX else d = sign of ΔY

그림 6의 초기조건 생성부를 사용하는 경우 I1=X1, I2=Y1, I3=X3, I4=X4이며, 증감조건 d=0일 때는 카운터를 감소하며, d=1일 때는 카운터를 증가시켜 증감치가 0일때까지 현재 값이 발생되며, 이 값에 따라 어레이 프로세서의 행렬 및 벡터값을 입력한다.

(2) 연속적인 도형발생을 위한 기본식의 수정

Ⅲ의 식(4)에서 선분발생시 행렬·벡터([4×4] · [4])의 형태로 실행할 경우 선분의 3점 발생 후 1 사이클의 널 데이타를 출력하고 다음의 선분점을 생성하므로 연속적으로 선분이 발생되지 않는 단점이 있다. 그러므로 다음의 식(10)과 같이 4개의 선분 구성점이 발생되도록 2개의 [2×4] · [4] 의 행렬 벡터 연산으로 분리한다.

$$\begin{bmatrix} x1' \\ y1' \end{bmatrix} = \begin{bmatrix} Sx & 0 & Dx & 0 \\ 0 & Sy & 0 & Dy \end{bmatrix} \cdot \begin{bmatrix} x1 \\ y1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x2' \\ y2' \end{bmatrix} = \begin{bmatrix} Sx & 0 & Dx & 0 \\ 0 & Sy & 0 & Dy \end{bmatrix} \cdot \begin{bmatrix} x2 \\ y2 \\ 1 \\ 1 \end{bmatrix} \qquad (10)$$

식 (9) 경우와 같이 식 (10)을 그림 4에 설계된 시스톨릭 어레이 프로세서에 적용시 두개의 행렬은 Rg0~Rg3에 입력하고 두개의 벡터는 RgH와 RgL에 입력함으로써 연속적으로 선분의 점을 발생할 수 있다.

2) 원 발생

식 (6)에 의한 방법은 θ값이 증가할 때마다 새로운 행렬과 벡터 값을 입력하여 연산하여야 하므로 1차원 시스톨릭 어레이 프로세서를 이용한 발생기 설계시 발생속도가 저하된다. 그러므로 본 논문에서는 다음의 3가지 방식으로 원도형을 발생한다.

방식 Ⅰ.(원의 발생)

행렬 값으로 고정된 값인 R과 Cx를 사용하며, 각 구간에 따른 행렬값 식 (11)과 θ의 증가에 따른 sin(θ), cos(θ)를 벡터 값을 입력하여 원을 발생한다.

$$\begin{bmatrix} x1 \\ y1 \\ x2 \\ y2 \end{bmatrix} = M \bullet \begin{bmatrix} \cos\theta \\ \sin\theta \\ 1 \\ 1 \end{bmatrix} \tag{11}$$

(θ는 각 구간마다 0~π/4만큼 변화함)

$$M = \begin{bmatrix} R & 0 & Cx & 0 \\ 0 & R & 0 & Cy \\ 0 & R & Cx & 0 \\ R & 0 & 0 & Cy \end{bmatrix}, \begin{bmatrix} 0 & R & Cx & 0 \\ R & 0 & 0 & Cy \\ R & 0 & Cx & 0 \\ 0 & R & 0 & Cy \end{bmatrix}, \begin{bmatrix} R & 0 & Cx & 0 \\ 0 & R & 0 & Cy \\ 0 & R & Cx & 0 \\ R & 0 & 0 & Cy \end{bmatrix}, \begin{bmatrix} 0 & R & Cx & 0 \\ R & 0 & 0 & Cy \\ R & 0 & Cx & 0 \\ 0 & R & 0 & Cy \end{bmatrix} \tag{12}$$

$0 \le \theta < \pi$ $\quad \pi/2 \le \theta < \pi$ $\quad \pi \le \theta < 3\pi/2$ $\quad 3\pi/2 \le \theta < 2\pi$

방식 Ⅱ.(원,원호,타원의 발생)

식 (11)에 의한 방법은 θ=0~2π 의 원만을 발생시 유용하지만 그림 9와 같은 원호의 발생시에는 문제점을 갖는다. 그러므로 원 및 원호의 발생을 동시에 고려하는 경우 다음 식 (13)과 같이 표현할 수 있다.

$$\begin{bmatrix} X1 \\ Y1 \\ - \\ - \end{bmatrix} = \begin{bmatrix} R & 0 & Cx & 0 \\ 0 & R & 0 & Cy \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \bullet \begin{bmatrix} cos(\theta) \\ sin(\theta) \\ 1 \\ 1 \end{bmatrix} \quad 0 \le \theta < 2\pi \tag{13}$$

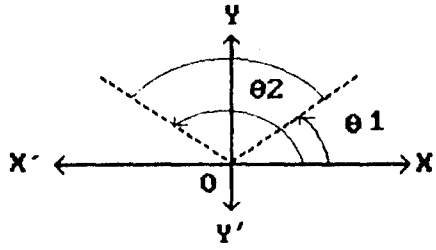


그림 9. θ1과 θ2 사이의 원호의 발생

Fig. 9. ARC generation from θ1 to θ2

식 (13)의 경우 $0 \le \theta < \pi/4$, $\pi/4 \le \theta < \pi/2$ 등의 구간 구분 없이 θ의 값을 연속적으로 변환시키며 원을 발생할 수 있다. θ가 $\theta_1 \le \theta < \theta_2$ 인 원호의 발생시 선분 발생과 같이 동일한 구조로 수행할 수 있으므로 하드웨어 설계시 공통된 설계방식 및 크기의 감소를 갖을 수 있으나, 행렬·벡터 연산시 사용하지 않는 부분(행)에 대하여 연산기 설계시 연속적으로 도형 생성점이 발생할 수 있도록 효율적인 설계가 되어야 한다. 또한 원 발생과 동일한 발생식을 사용하여 타원의 발생에도 적용할 수 있다.

타원의 방정식은

$$X = R1 \times \cos(\theta),\ Y = R2\ \sin(\theta) \tag{14}$$

이므로 식 (13)의 행렬 값을 식 (15)로 전환하면 타원을 발생할 수있다.

$$\begin{bmatrix} R_1 & 0 & Cx & 0 \\ 0 & R_2 & 0 & Cy \\ - & - & - & - \\ - & - & - & - \end{bmatrix} (0 \le \theta < 2\pi) \tag{15}$$

방식 Ⅲ.(연속적인 도형발생을 위한 기본식의 수정)

원, 원호, 타원의 발생식 식 (13)의 경우 식 (4)의 경우와 같이 널 데이타가 발생되므로 2개의 원 구성점이 발생되도록 식 (16)과 같이 2개의 [2×4]·[4] 행렬·벡터 연산으로 분리하면 다음 식과 같다.

$$\begin{bmatrix} X1 \\ Y1 \end{bmatrix} = \begin{bmatrix} R & 0 & Cx & 0 \\ 0 & R & 0 & Cy \end{bmatrix} \bullet \begin{bmatrix} cos(\theta 1) \\ sin(\theta 1) \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X2 \\ Y2 \end{bmatrix} = \begin{bmatrix} R & 0 & Cx & 0 \\ 0 & R & 0 & Cy \end{bmatrix} \bullet \begin{bmatrix} cos(\theta 2) \\ sin(\theta 2) \\ 1 \\ 1 \end{bmatrix} \tag{16}$$

식 (16)의 두 행렬 값을 식 (17)으로 전환하면 타원을 발생할 수있다.

$$\begin{bmatrix} x1 \\ y1 \\ x2 \\ y2 \end{bmatrix} = M \bullet \begin{bmatrix} cos\theta \\ sin\theta \\ 1 \\ 1 \end{bmatrix} \tag{17}$$

식 (9) 경우와 같이 식 (16),(17)을 그림 4의 수정된 시스톨릭 어레이 프로세서에 적용시 두개의 행렬은 Rg0~Rg3에 입력하고 두개의 벡터는 RgH와 RgL에 입력함으로써 연속적으로 원, 원호 및 타원의

점이 발생되도록 할 수 있다.

(1) 원 및 원호,타원 발생의 초기조건

원의 중점을(C_x, C_y), 반지름을 R이라할 때 식 (16)에 의한 원 발생 초기조건은 θ의 증감치만 요구된다. 식 (6), (11)에 의한 원호의 발생은 그림 9와 같이 θ1에서 θ2까지 원호를 생성할때 θ에 따라 1/8원씩 구간분할을 수행하여야 하므로 비효율적이다. 그러므로 본 논문에서는 원과 원호의 발생이 동시에 가능한 식 (16), (17)에 의해서 원, 원호 및 타원의 발생을 수행한다. 이 경우의 초기 조건은 다음과 같다.

① $\Delta\theta=\theta_2-\theta_1$

② 입력의 초기치 θI는 if $\Delta\theta\geq 0$ then $\theta I=\theta 1$ else $\theta I=\theta 2$ 이다.

③ 발생 정지기준은 다음과 같다.

입력의 증감치 D는 $D=|\Delta\theta|$이며

증감의 조건 d는 d=sign of $\Delta\theta$이다.

원, 원호, 타원의 발생시 그림 6의 초기조건 생성부를 사용하는 경우 $I1=\theta 1, I2=\theta 1, I3=\theta 2, I4=\theta 2$ 이며, 증감조건 d = 0일 때는 카운터가 감소하며, d = 1일 때는 카운터가 증가하여 증감치가 0일 때까지 증감값 θ가 발생한다. 이 값과 초기 값을 합한 현재 값에 따라 cos(θ),sin(θ) 값을 생성하며, 식 (16)의 형태로 행렬 및 벡터의 값을 생성하여 행렬 벡터 연산용 1차원 시스톨릭 어레이 프로세서에 적용함으로써 도형을 발생한다.

Ⅵ. 그래픽 가속기의 입력 데이타의 분류 및 연산시간

표 1. 그래픽 기능에 따른 입력 데이타의 분류

Table1. Classification of input data in each graphic function.

기능 \ 입력	초기조건 생성부	벡 터	행렬
선 분	X1, Y1, X2, Y2	d_{x1}	a, I
원	θ1, θ2	cosθ,sinθ	R, Cx, Cy
원 호	θ1, θ2	cosθ,sinθ	R, Cx, Cy
타 원	θ1, θ2	cosθ,sinθ	R1, R2, Cx, Cy
회전및 이동	-	x, y	cosθ,sinθ Dx, Dy
신축및 이동	-	x, y	Sx, Sy Dx, Dy

본 논문에서 제안한 그래픽 가속기의 각 그래픽 기능에 따른 입력 데이타의 분류를 표 1에 표시하였으며, 초기조건 생성부는 그래픽 기본요소 발생시에만 사용한다.

제안된 그래픽 가속기의 연산시간은 다음과 같다. 행렬, 벡터용 레지스터에 데이타 저장시간을 t_{dr}, 곱셈기의 연산시간을 t_{mul}, 가산기의 연산시간을 t_{add}라 할 때, PE당 소요시간은 $t_{pe}=t_{mul}+t_{add}$ 이며, 대소 비교기에 소요되는 시간 t_{comp}, 증감치 연산시 소요되는 시간 t_{dif}이라 하면 $t_{dif}\cong t_{comp}$이다. 그래픽 기본요소 및 기하학적인 변환에 필요한 처리시간은 도형생성시 발생되는 점의 수를 m이라할 때 처리시간을 표 2에 나타내었으며, 선분 발생시는 식 (10)이 적합하며, 원 발생시는 식 (16)이 적합함을 알 수 있다.

표 2. 그래픽 기본요소 및 기하학적 변환시 처리시간

Table 2. Processing time of graphic primitive and geometrical transforamtions.

기능 \ 처리시간	초기조건 생성 및 그래픽 처리
선분:식(4)	$2t_{df}+4t_{dr}+(m+[m/3]+3)t_{pe}$
선분:식(10)	$2t_{df}+4t_{dr}+(m+3)t_{pe}$
원:식(6),(7)	$2t_{df}+(4t_{dr}+19t_{pe})/8$
원:식(11),(12)	$4(2t_{df}+4t_{dr}+(3+m/2)t_{pe})$
원:식(16)	$2t_{df}+4t_{dr}+(3+2m)t_{pe}$
원호:식(16)	$2t_{df}+4t_{dr}+(3+2m)t_{pe}$
타원:식(16)	$2t_{df}+4t_{dr}+(3+2m)t_{pe}$
회전 및 이동	$4t_{dr}+(2m+3)t_{pe}$
신축 및 이동	$4t_{dr}+(2m+3)t_{pe}$

(m : 발생된 화소수)

또한 설계된 그래픽 가속기는 파이프라인의 특성을 가지고 있으므로 연산속도는 데이타의 저장 및 연산시 처리속도가 가장 늦은 부분에 의해 결정된다. 그러므로 식 (18)과 같이 처리시간들 중에서 최대치를 그래픽 가속기의 기준 클럭 (t_{clock})으로 사용한다.

$$t_{clock}=\mathrm{Max}\{t_{dr}, t_{pe}, t_{comp}, t_{dif}\} \quad (18)$$

이때 제안된 그래픽 가속기와 동일한 속도(t_{clock})의 연산기를 사용하는 경우 다른 그래픽 프로세서 (HD66400 [7], TMS34020 [8])의 처리시간 비교를 표 3에 나타내었다.

도형발생시 그래픽 도면의 해상도가 800×600인 경우 최장의 선분은 대각선이고 1000 dots로 구성되며, 원은 최대 반지름 R = 300이고 원둘레는 약 2400 dots로 구성된다. 이 경우 처리 시간을 표 3의 '( )'에 나타내었으며, 다른 그래픽 프로세와 비교시 제안된 그래픽 가속기가 선분의 경우는 4배 이상, 원 도형의 경우는 40배 정도 빠름을 알 수 있다. 예를

들어 식 (10), (16)을 제안된 그래픽 가속기에 적용시 $t_{pc} \cong t_{clock} \cong 33.3$ nsec인 연산기를 사용하였을 때 선분인 경우는 최장의 선분을 초당 약 29732개 발생할 수 있으며, 원도형의 경우 최대의 원을 초당 약 6244개 발생할 수 있다.

표 3. 제안된 그래픽 가속기와 타 그래픽 프로세서의 처리시간 비교

Table 3. Comparison of the proposed graphic accelerator with other grahpic processor in processing speed.

(단위 : t_{clock})

처리시간 / 기능	초기조건 생성 및 그래픽 처리		
	제안된 그래픽 가속기	HD64400 GDP	TMS34020
선분 : 식(10)	m+9.(1009)	120+4m.(4120)	5+3m.(5004)
원 : 식(16)	2m+9.(4809)	691+170×2R.(204691)	—
원호 : 식(16)	2m+9.(4809)	605+170×2R.(204691)	—
타원 : 식(16)	2m+9.(4809)	691+170×2R.(204691)	—

('—'는 기능이 없음을 표시함, m : 발생된 화소수 , R : 반지름 '( )'는 해상도 800x600인 경우, 선분의 대각선 최대길이 m=1000, 원의 최대 반지름 R=300이고 m 2400일 때 실행 클럭수를 표시함.)

Ⅶ. 결 론

컴퓨터 그래픽의 응용분야가 발전함에 따라 고속의 그래픽 연산이 요구된다. 기하학적 변환 및 도형발생의 기본요소인 선분, 원, 원호, 타원 등의 도형발생시 행렬·벡터의 연산 구조를 갖는다. 그러므로 본 논문에서는 기하학적 변환 및 도형발생시 연속적으로 결과가 발생되게 하기 위하여 $[N \times M] \cdot [N]$ $(1 \le M \le N,\ 1 \le N \le 4)$ 및 $[N/2 \times N] \cdot [N]$ ($[N/2] \ge 1$)의 행렬·벡터 연산이 가능한 1차원 시스톨릭 어레이 프로세서를 설계하고 이를 주 연산장치로 사용하며, 연속적인 입력값이 발생하도록 초기조건 생성부를 설계하여 고해상도 도면에서 그래픽 기능을 고속으로 처리기 위한 그래픽 가속기를 설계하였다. 33.3 nsec의 연산기를 사용하는 경우 선분인 경우는 초당 29732개, 원의 경우는 초당 6244개의 도형이 발생될 수 있으므로 고속의 그래픽 처리가 가능하다. 앞으로 기존의 그래픽 장치와의 호환성에 대한 연구가 필요하며, 삼각함수 발생기 및 보다 다양한 응용적 기능에 대한 연구가 계속되어야할 것이다.

參 考 文 獻

[1] P .E.Danielson, "Incremental Curve Generation," *IEEE Trans.*on Computer, vol.C-19, pp. 783-793, Sept.1970.

[2] Foley, Van Dam, Feiner, and Hugles, Computer Graphic Principle and Practice, Addison-Wesley, pp. 202-210, 1990.

[3] B.W.Jordan, "An Improved Algorithm for The Generation of Nonparametric Curves," *IEEE Trans.* on Computer, vol C-22, pp. 1052-1060, Dec. 1979.

[4] Y.Suenaga, T.Kamae, and T. Kobayashi, "A High-speed Algorithm for the generation of straight Line and Circular Arcs," *IEEE Trans.* on Computer, vol. C-28, pp. 728-736, No. 10, Oct. 1979.

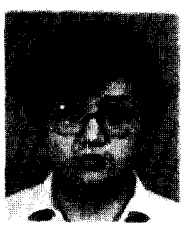
[5] W.M.Neuman, Principle of Interactive Computer Graphics, McGraw-hill Inc, pp. 22-28, 1979.

[6] S.Y. Kung, VLSI Array Processors, Prentice-Hall, pp. 11-145, 1988.

[7] TMS34010 User's Manual, Taxas Instrument, 1986

[8] HD64400 GDP User's Manual, Hitachi, 1990

著 者 紹 介



金 龍 成(正會員)

1959年 9月 10日生. 1983年 경희대학교 전자공학과 졸업(공학학사) 1985年 경희대학교 대학원 전자공학과(공학석사) 1989年 현재 경희대학교 전자공학과 박사과정 재학중 주관심분야는 고속 그래픽처리를 위한 VLSI설계, 고속 연산기 설계 및 고속 디지탈 신호처리용 VLSI설계 등임.

•

趙 源 敬(正會員) 第28卷 B編 第10號 參照

현재 경희대학교 전자공학과 교수