임의의 축에 관한 3차원 영상의 회전 행렬 계산 속도의 개선

김 응 곤† 허 영 남†† 이 웅 기†††

요 약

컴퓨터 그래픽스의 응용 분야 특히, CAD나 애니메이션 분야에서는 하나의 물체를 서로 다른 시점에서 또는 시점은 고정하고 다른 각도로 바라본 형태를 디스플레이 할 수 있어야 한다. 그러기 위해서는 어떤 물체를 임의의 축에 관하여 임의의 각도 만큼 회전 이동시키는 회전 이동 변환이 필요하다. 3차원 그래픽스 응용에서는 많은 변환이 필요하며, 하나의 물체는 보통 많은 꼭지점들로 구성되므로 가능한한 빠른 속도로 회전 이동 좌표를 구하는 것이 중요한 문제이다. 본 논문에서는 3차원 그래픽스에서 필요한 변환중 많은 연산 횟수로 인하여 계산 시간이 문제가 되는 임의의 축에 관한 회전 이동 변환 행렬을 빠르게 계산하는 효율적인 알고리즘을 제안하고, 이를 PC-486DX2-50상에서 구현하여 기존의 계산 방법과 결과의 계산 속도를 비교함으로써 알고리즘의 유용성을 보인다.

An Improvement of Computation of Rotation Matrix for a 3D Image about an Arbitrary Axis

Eung Kon Kim† Young Nam Heo†† and Woong Ki Lee†††

ABSTRACT

One of the advantages of computer graphics is that it enables to view an object on different viewpoints and different angles. Therefore, a computer graphics system should be able to rotate an arbitrary object by an arbitrary angle about an arbitrary axis. This is usually done by rotating vertices that represent an object and connecting them. Hence an image may have many vertices, it is important to be able to rotate each of them quickly. Therefore, this paper is interested in a rotation matrix computation method that consists of the smallest number of computational steps. This paper proposes an algorithm that computes rotation matrix to rotate a 3 dimensional image about an arbitrary axis quickly.

1. 서 론

컴퓨터 그래픽스의 많은 응용에서는 물체를 표현하는 화상의 위치나 형태를 변화시켜 화면상에 물체의 연속적인 동작을 표현하거나 물체의 크기를 확대 또는 줄일 경우도 있다. 이러한 여러가지 조작은 각 화상의 좌표 점들에 적당한 기하학적 변환을 적용시킴으로써 이루어진다. 기본적인 변환에는 평행 이동(translation), 신축(scaling), 회전 이동(rotation) 등이 있다[1, 2, 3, 4,]. 이들 변환 중에서 3차원 영상의 회전 이동은 컴퓨터 그래픽스에 있어서 매우 중요하다. 컴퓨터 그래픽스의 응용 분야 특히, CAD나 애니메이션 응용 분야에서는 하나의 물체를 서로 다른 시점에서 또는 시점은 고정하고 다른 각도로 바라본 형태를 나타낼 수 있어야 한다[4, 5, 6]. 그러기 위해서는 어떤 물체를 임의의 축에 관하여 임의의 각도만큼 회전 이동시키는 회전 이동 변환이 필요하다. 물체를 회전 이동시키기 위해서는 먼저 물체를 구성하는 꼭지점들에 대한 회전 이동 좌표를 구한 다음, 이 좌표들을 화면상에 표시하고 이들을 적절히 연결시키면 된다. 그런데 3차원 그래픽스 응용에서는 많은 변환이 필요하며, 하

† 종신회원 : 순천대학교 전자계산학과 조교수
†† 종신회원 : 순천대학교 전자계산학과 교수
††† 종신회원 : 조선대학교 전산통계학과 부교수
논문접수 : 1995년 2월 24일, 심사완료 : 1995년 3월 17일

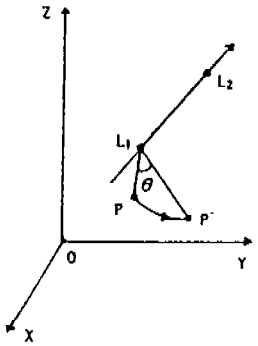
나의 물체는 보통 많은 꼭지점들로 구성되므로 가능한한 빠른 속도로 회전 이동 좌표를 구하는 것이 중요한 문제이다[2]. 회전 이동중 특히, 임의의 축에 관한 회전 이동은 좌표축에 대한 평행 이동과 회전이동을 수행하는 일련의 행렬들의 곱인 조합 행렬을 구함으로써 가능한데 다른 변환에 비하여 매우 많은 연산과 시간을 요한다.

따라서 본 논문에서는 3차원 영상을 임의의 축에 관하여 회전 이동시키기 위하여 필요한 변환 행렬을 산술 연산 횟수를 최대한으로 줄여 계산하는 알고리즘을 제시하고, 이를 PC-486DX상에서 구현하여 결과의 계산 속도를 기존의 계산 방법과 비교함으로써 알고리즘의 유용성을 보인다.

2 임의의 축에 관한 회전 이동 행렬

회전 이동 변환에는 x, y, z축 각각에 대한 회전, 임의의 점에 관한 회전 그리고 임의의 축에 관한 회전 등이 있다. 이들 회전 이동 변환 중 임의의 축에 관한 회전 이외의 변환은 계산이 간단하나, 임의의 축에 관한 회전 이동은 좌표 축에 관한 평행 이동과 좌표축에 관한 회전 이동들의 조합 변환을 통하여 수행되므로 다른 변환에 비하여 많은 산술 연산의 횟수와 계산 시간이 필요하다.

(그림 1)과 같이 3차원 물체를 구성하는 꼭지점들 중 하나의 점 P(x, y, z)를 점 $L_1(x_1, y_1, z_1)$과 $L_2(x_2, y_2, z_2)$를 잇는 직선을 회전축으로 하여



(그림 1) 점 P를 축 L_1 L_2에 관하여 θ만큼 회전시의 점 P'
(Fig. 1) Three-dimensional rotation about an arbitrary axis L_1 L_2

θ만큼 회전 이동한 후의 점의 좌표 P'(x', y', z')를 구하기 위한 회전 행렬을 구해 보자.

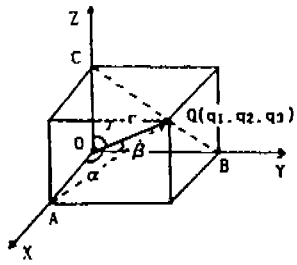
x, y, z축을 중심으로 각각 θ만큼 회전 이동할 때의 변환 행렬 Rx(θ), Ry(θ), Rz(θ)는 각각 식 1), 2), 3)과 같다[1, 2, 3, 4].

$$Rx(\theta)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & -\sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad \cdots\cdots 1)$$

$$Ry(\theta)=\begin{bmatrix}\cos\theta & 0 & -\sin\theta & 0\\ 0 & 1 & 0 & 0\\ \sin\theta & 0 & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad \cdots\cdots 2)$$

$$Rz(\theta)=\begin{bmatrix}\cos\theta & \sin\theta & 0 & 0\\ -\sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad \cdots\cdots 3)$$

x_2-x_1, y_2-y_1, z_2-z_1을 각각 q_1, q_2, q_3라 할 때 회전축의 방향을 결정하기 위한 방향 코사인 cosα, cosβ, cosγ는 (그림 2)와 같으며, 삼각형 OAQ에서 각 A가 직각이므로 cos$\alpha=q_1/r$이 되고, 같은 방법으로 모든 방향 코사인을 구하면 식 4), 5), 6)과 같다[1].



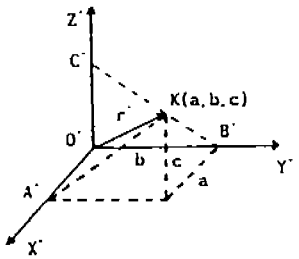
(그림 2) 방향 코사인
(Fig. 2) Direction cosines

$$\cos\alpha=\frac{q_1}{r}=\frac{q_1}{(q_1^2+q_2^2+q_3^2)^{1/2}} \quad \cdots\cdots 4)$$

$$\cos\beta=\frac{q^2}{r}=\frac{q_2}{(q_1^2+q_2^2+q_3^2)^{1/2}} \quad \cdots\cdots 5)$$

$$\cos\gamma=\frac{q_3}{r}=\frac{q_3}{(q_1^2+q_2^2+q_3^2)^{1/2}} \quad \cdots\cdots 6)$$

여기서 (그림 3)과 같이 점 K의 좌표를 a=cosα, b=cosβ, c=cosγ인 (a, b, c)로 하면 벡터 O'K의 크기는 식 7)과 같이 되어 단위 벡터가 된다.



(그림 3) 단위 벡터의 방향 코사인
(Fig. 3) Direction cosines of unit vector O'K

$$r' = (a^2+b^2+c^2)^{1/2} = (\cos^2\alpha+\cos^2\beta+\cos^2\gamma)^{1/2} = 1 \quad \cdots\cdots 7)$$

(그림 3)에서 삼각형 O'A'K는 각 A'가 직각인 직각 삼각형이므로 단위 벡터 O'K의 방향 코사인은 각 KO'A'의 크기를 α'라 할 때 $\cos\alpha' = a/1 = \cos\alpha/1 = \cos\alpha$가 되고 같은 방법으로 다른 방향 코사인은 $\cos\beta' = \cos\beta$, $\cos\gamma' = \cos\gamma$이 되어 (그림 2)의 벡터 OQ와 (그림 3)의 벡터 O'K의 방향 코사인은 같으므로, (그림 1)의 회전축 QJ의 방향 벡터를 (그림 3)의 벡터 O'K로 하여 점 P를 축 QJ에 대하여 θ만큼 회전할 때의 변환 행렬을 다음 절차에 의하여 구할 수 있다[1].

단계 1) 점 Q가 원점인 새로운 좌표계로 물체를 평행 이동시킨다. (그림 4)와 같이 새로운 좌표계를 X'Y'Z'라 할 때 평행 이동 변환 행렬 T_1은 식 8)과 같이 된다[1].

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -q_1 & -q_2 & -q_3 & 1 \end{bmatrix} \quad \cdots\cdots 8)$$

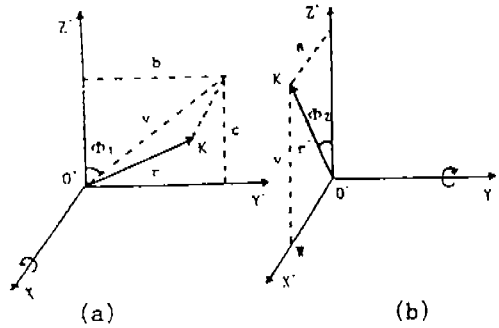
단계 2) (그림 4)의 (a)와 같이 벡터 O'K를 Y'Z' 평면에 투영시킨 선분이 Z'축과 이루는 각을 Φ_1이라 할 때 벡터 O'K를 X'축을 중심으로 반시계 방향으로 Φ_1만큼 회전시킨다. 이 때 벡터 O'K는 (그림 4)의 (b)와 같이 X'Z'평면에 놓이게 되며, 변환 행렬 $Rx(\Phi_1)$은 식 1), 9), 10), 11)에 의하여 식 12)와 같이 된다.

$$\cos\Phi_1 = \frac{c}{v} \quad \cdots\cdots 9)$$

$$\sin\Phi_1 = \frac{b}{v} \quad \cdots\cdots 10)$$

$$v = (b^2+c^2)^{1/2} \quad \cdots\cdots 11)$$

$$R_x(\Phi_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/v & b/v & 0 \\ 0 & -b/v & c/v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \cdots 12)$$



(그림 4) 점 K를 Z' 축에 일치시키기 위한 과정
(Fig. 4) Two rotations to align the point K(a, b, c) with the Z' axis

단계 3) (그림 4)의 (b)에서 벡터 O'K가 Z'축과 이루는 각 Φ_2만큼 Y'축을 중심으로 벡터 O'K를 회전 이동시키면 벡터 O'K는 Z'축과 일치하게 되며, 이 때의 변환 행렬 $R_y(\Phi_2)$는 식 2), 13), 14)에 의하여 식 15)와 같이 된다.

$$\cos(-\Phi_2) = \cos(\Phi_2) = v \quad \cdots\cdots 13)$$

$$\sin(-\Phi_2) = -\sin(\Phi_2) = -a \quad \cdots\cdots 14)$$

$$R_y(\Phi_2) = \begin{bmatrix} v & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \cdots 15)$$

단계 4) Z'축을 중심으로 θ만큼 회전 이동한다. 이 때의 변환 행렬 $R_z(\theta)$는 식 16)과 같다.

$$R_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \cdots 16)$$

단계 5) 물체를 원래의 좌표계(XYZ)로 환원시킨다. 그러기 위해서 단계 3), 단계 2), 단계 1)을 거꾸로 변환하는 행렬 $R_y(-\Phi_2)$, $R_X(-\Phi_1)$, T_1^{-1}을 각각 적용시키면 된다. 따라서 최종적인 변환 행렬 T는 식 20)과 같이 된다.

$$R_y(-\Phi_2)=\begin{bmatrix} v & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \cdots\cdots 17)$$

$$R_X(-\Phi_1)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/v & -b/v & 0 \\ 0 & b/v & c/v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \cdots 18)$$

$$T_1^{-1}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ q_1 & q_2 & q_3 & 1 \end{bmatrix} \quad \cdots 19)$$

$$T=T_1\cdot R_X(\Phi_1)\cdot R_y(\Phi_2)\cdot R_Z(\theta)\cdot Ry(-\Phi_2)\cdot R_X(-\Phi_1)\cdot T_1^{-1} \quad \cdots\cdots 20)$$

3. 회전 이동 변환 행렬 계산 알고리즘

2장의 식 20)과 같은 회전 이동 변환 행렬을 구하기 위한 기존의 일반적인 알고리즘은 다음과 같다[2].

단계 1) $\sin\theta$, $\cos\theta$, $q_1=x_2-x_1$, $q_2=y_2-y_1$, $q_3=z_2-z_1$, $r=(q_1^2+q_2^2+q_3^2)^{1/2}$, $a=\cos\alpha=q_1/r$, $b=\cos\beta=q_2/r$, $c=\cos\gamma=q_3/r$를 계산한다.

단계 2) $v=(b^2+c^2)$을 계산한다.

단계 3) 행렬 T_1과 $R_X(\Phi_1)$의 곱을 계산한다.

단계 4) 단계 3의 결과 행렬과 $R_y(\Phi_2)$의 곱을 계산한다.

단계 5) 단계 4의 결과 행렬과 $R_Z(\theta)$의 곱을 계산한다.

단계 6) 단계 5의 결과 행렬과 $R_y(-\Phi_2)$의 곱을 계산한다.

단계 7) 단계 6의 결과 행렬과 $R_X(-\Phi_1)$의 곱을 계산한다.

단계 8) 단계 7의 결과 행렬과 T_1^{-1}의 곱을 계산한다.

이 알고리즘의 계산 복잡도(computational complexity)[7]를 구해 보면 다음과 같다.

1) 단계 1)의 $\sin\theta$, $\cos\theta$를 계산하는데 2개의 연산과 q_1, q_2, q_3를 계산하는데 3개의 뺄셈과 r을 계산하는데 3개의 곱셈과 2개의 덧셈 그리고 1개의 비산술 연산인 제곱근 연산이 필요하다. 또한 a, b, c를 계산하는데 3개의 나눗셈이 필요하다.
2) 단계 2)의 v를 계산하는데 2개의 곱셈과 1개의 덧셈 그리고 1개의 제곱근 연산이 필요하다.
3) 행렬 T_1의 원소 중 $-q_1$, $-q_2$, $-q_3$를 구하는데 3개의 −연산이 필요하다.
4) 행렬 $R_X(\Phi_1)$의 원소 중 b/v, c/v, -b/v를 구하는데 2개의 나눗셈과 1개의 −연산이 필요하다.
5) 행렬 $R_y(\Phi_2)$의 원소 중 -a를 구하는데 1개의 −연산이 필요하다.
6) 행렬 $R_Z(\theta)$의 원소 중 $-\sin\theta$를 구하는데 1개의 −연산이 필요하다.
7) 단계 3)에서 단계 8)까지 변환 행렬 T를 구하는데 4×4 행렬들의 곱셈이 6개 필요하다. 이를 위해서 우선 2개의 4×4행렬(A,B)의 곱(C)을 구하는데 필요한 연산의 수를 계산해 본다. C행렬의 원소 하나를 구하는데 4개의 곱셈과 3개의 덧셈이 필요하므로 C행렬의 16개 원소를 구하는데는 64개의 곱셈과 48개의 덧셈이 필요하다. 결국 행렬의 곱을 6회 수행하여야 하므로 384개의 곱셈과 288개의 덧셈이 필요하다.

임의의 축에 대한 회전 이동 변환 행렬 T를 구하기 위해서는 1)에서 7)까지 291개의 덧셈, 3개의 뺄셈, 389개의 곱셈, 5개의 나눗셈, 10개의 비산술 연산 등 모두 698개의 연산이 필요하다.

4. 본 논문의 알고리즘

3차원 영상을 구성하는 하나의 점의 좌표(x, y, z)를 임의의 축에 대하여 회전 이동하였을 때의 새로운 좌표(x′, y′, z′)를 구하기 위해서는 앞에서 구한 변환 행렬 T를 먼저 구한 후, 좌표(x,

y,z)를 포함하는 벡터[x, y, z, 1]와 행렬 T를 곱하면 된다[1, 2].

따라서 3차원 영상을 임의의 축에 관하여 회전시켰을 때의 새로운 좌표를 구하기 위해서는 변환 행렬 T를 빨리 구하는 것이 가장 중요한 문제이다.

3장에서 회전 이동 변환 행렬을 구하는 기존의 일반적인 알고리즘은 매우 많은 연산이 요구되므로 비효율적이다. 이 방법에서는 조합 행렬의 곱 $(T_1 \cdot R_X(\Phi_1) \cdot R_Y(\Phi_2) \cdot R_Z(\theta) \cdot R_Y(-\Phi_2) \cdot R_X(-\Phi_1) \cdot T_1^{-1})$을 개별적으로 구해 나가므로 연산의 수가 많다.

조합 행렬들의 곱의 결과에서 나오는 각 원소로부터 동일한 요소를 유도해 내면 연산의 수를 줄일 수 있다. 본 논문에서는 식 20)의 조합 행렬의 곱의 결과 나오는 각 원소를 다음과 같이 변형하여 동일한 요소를 유도함으로써 연산의 수를 줄였다.

변환 행렬 T의 각 원소를 식 21)과 같이 A~I라 할 때 각 원소는 식 20)에 식 8), 12), 15), 16), 17), 18), 19)를 대입하면 식 22)~30)과 같이 된다.

$$T = \begin{bmatrix} A & B & C & 0 \\ D & E & F & 0 \\ G & H & I & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \cdots\cdots 21)$$

$$A = a^2+(1-a^2)\cos\theta \quad \cdots\cdots 22)$$
$$B = ab(1-\cos\theta)+c\sin\theta \quad \cdots\cdots 23)$$
$$C = ac(1-\cos\theta)-b\sin\theta \quad \cdots\cdots 24)$$
$$D = ab(1-\cos\theta)-c\sin\theta \quad \cdots\cdots 25)$$
$$E = b^2+(1-b^2)\cos\theta \quad \cdots\cdots 26)$$
$$F = bc(1-\cos\theta)+a\sin\theta \quad \cdots\cdots 27)$$
$$G = ac(1-\cos\theta)+b\sin\theta \quad \cdots\cdots 28)$$
$$H = bc(1-\cos\theta)-a\sin\theta \quad \cdots\cdots 29)$$
$$I = c^2+(1-c^2)\cos\theta \quad \cdots\cdots 30)$$

여기에 (그림 2)와 (그림 3)에 의해 $a=q_1/r$, $b=q_2/r$, $c=q_3/r$를 각각 대입하면 식 31)~39)와 같이 된다.

$$A = \frac{(1-\cos\theta)}{r^2}\cdot q_1^2+\cos\theta \quad \cdots\cdots 31)$$

$$B = \frac{(1-\cos\theta)}{r^2}\cdot q_1\cdot q_2-\frac{\sin\theta}{r}\cdot q_3 \quad \cdots\cdots 32)$$

$$C = \frac{(1-\cos\theta)}{r^2}\cdot q_1\cdot q_3-\frac{\sin\theta}{r}\cdot q_2 \quad \cdots\cdots 33)$$

$$D = \frac{(1-\cos\theta)}{r^2}\cdot q_1\cdot q_2-\frac{\sin}{r}\cdot q_3 \quad \cdots\cdots 34)$$

$$E = \frac{(1-\cos\theta)}{r^2}\cdot q_2^2+\cos\theta \quad \cdots\cdots 35)$$

$$F = \frac{(1-\cos\theta)}{r^2}\cdot q_2\cdot q_3-\frac{\sin\theta}{r}\cdot q_1 \quad \cdots\cdots 36)$$

$$G = \frac{(1-\cos\theta)}{r^2}\cdot q_1\cdot q_3-\frac{\sin\theta}{r}\cdot q_2 \quad \cdots\cdots 37)$$

$$H = \frac{(1-\cos\theta)}{r^2}\cdot q_2\cdot q_3-\frac{\sin\theta}{r}\cdot q_1 \quad \cdots\cdots 38)$$

$$I = \frac{(1-\cos\theta)}{r^2}\cdot q_3^2+\cos\theta \quad \cdots\cdots 39)$$

$\frac{(1-\cos\theta)}{r^2}$를 P로, $\frac{\sin}{r}$를 Q로 각각 치환하면 다음과 같다.

$$A = P\cdot q_1^2+\cos\theta \quad \cdots\cdots 40)$$
$$B = P\cdot q_1\cdot q_2+Q\cdot q_3 \quad \cdots\cdots 41)$$
$$C = P\cdot q_1\cdot q_3-Q\cdot q_2 \quad \cdots\cdots 42)$$
$$D = P\cdot q_1\cdot q_2-Q\cdot q_3 \quad \cdots\cdots 43)$$
$$E = P\cdot q_2^2+\cos\theta \quad \cdots\cdots 44)$$
$$F = P\cdot q_2\cdot q_3+Q\cdot q_1 \quad \cdots\cdots 45)$$
$$G = P\cdot q_1\cdot q_3+Q\cdot q_2 \quad \cdots\cdots 46)$$
$$H = P\cdot q_2\cdot q_3-Q\cdot q_1 \quad \cdots\cdots 47)$$
$$I = P\cdot q_3^2+\cos\theta \quad \cdots\cdots 48)$$

따라서 변환 행렬 T의 원소 A~I를 구하는 알고리즘은 다음과 같다.

단계 1) $q_1=x_2-x_1$, $q_2=y_2-y_1$, $q_3=z_2-z_1$를 계산한다.

단계 2) $\sin\theta$,$\cos\theta$를 계산한다.

단계 3) q_1^2, q_2^2, q_3^2, $r^2=q_1^2+q_2^2+q_3^2$, $r=(q_1^2+q_2^2+q_3^2)^{1/2}$을 계산한다.

단계 4) $P=(1-\cos\theta)/r^2$, $Q=\sin\theta/r$를 계산한다.

단계 5) $P\cdot q_1^2$, $P\cdot q_2^2$, $P\cdot q_3^2$, $P\cdot q_1$, $(P\cdot q_1)\cdot q_2$, $(P\cdot q_1)\cdot q_3$, $P\cdot q_2$, $(P\cdot q_2)\cdot q_3$를 계산한다.

단계 6) $Q\cdot q_1$, $Q\cdot q_2$, $Q\cdot q_3$를 계산한다.

단계 7) 식 40)~48)을 이용하여 A~I를 계산한다.

이 알고리즘에 대한 계산 복잡도는 다음과 같다.

1) 단계 1)에서 3개의 뺄셈이 필요하다.
2) 단계 2)에서 2개의 비산술 연산이 필요하다.
3) 단계 3)에서 3개의 곱셈, 2개의 덧셈, 1개의 제곱근 계산이 필요하다.
4) 단계 4)에서 1개의 뺄셈과 2개의 나눗셈이 필요하다.
5) 단계 5)에서 8개의 곱셈이 필요하다.
6) 단계 6)에서 3개의 곱셈이 필요하다.
7) 단계 7)에서 6개의 덧셈과 3개의 뺄셈이 필요하다.

5. 실증 및 고찰

3차원 영상을 구성하는 하나의 점에 대한 좌표 (x, y, z)를 임의의 축에 관하여 회전 이동하였을 때의 새로운 좌표(x′, y′, z′)는 앞에서 구한 회전 이동 변환 행렬 T를 이용하면 식 49)와 같이 된다.

$$[x', y', z'] = [x\ y\ z] \cdot \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \quad \cdots 49)$$

따라서 (x′, y′, z′)를 구하기 위해서 필요한 연산의 수는 변환 행렬 T의 원소 A~I를 구하는데 필요한 연산의 수와 식 49)를 계산하는데 필요한 9개의 곱셈과 6개의 덧셈의 수를 합한 것이 된다.

회전 변환 행렬 T를 구하기 위하여 3장에서와 같이 조합 행렬의 곱을 차례로 구해 나가는 일반적인 방법을 이용하면 모두 698개의 연산이 필요하며, 회전 이동 후의 좌표를 계산하기 위해서는 4개의 원소로 된 벡터와 4×4 행렬의 곱을 계산하여 세 좌표를 구해야 하므로 12개의 곱셈과 9개의 덧셈이 추가로 소요된다.

반면에 본 논문에서는 조합 행렬의 곱의 결과 나오는 각 원소를 변형하여 동일한 요소를 유도함으로써 연산의 수를 줄이는 방법을 이용하였다. 그 결과 34개의 연산만으로도 계산이 가능하여 연산 횟수를 크게 줄임으로써 계산 시간을 빠르게 함을 알 수 있다.

〈표 1〉은 두 가지 방법을 비교한 것으로 괄호 안의 수는 회전 변환 행렬 T를 구하는데 필요한 연산의 수와 식 49)의 회전 이동 좌표를 구할 때의 연산 수를 합한 것이다. 다음은 본 알고리즘을 이용하여 회전 변환 행렬의 원소를 1,000회 반복하여 계산하였을 때 소요되는 시간을 측정하기 위한 C코드를 나타낸 것이다.

연산의 종류와 횟수는 계산 시간과 직접적인 관계가 있다. 본 알고리즘의 연산 횟수는 일반적인 알고리즘에 비하여 1/20 정도이며, 연산의 종류 중 계산 시간에 큰 영향을 주는 산술 연산의 수를 크게 줄였다.

계산시간을 비교하기 위하여 PC-486DX2-50상에서 C언어로 구현하여 실증 분석한 결과 〈표 1〉과 같이 회전 이동 변환 행렬을 1,000회 반복하여 계산하는데 소요 시간이 크게 감소되어 본 알고리즘이 효율적임을 알 수 있다.

〈표 1〉 알고리즘 비교
〈Table 1〉 Comparison of algorithms

알고리즘	산술 연산의 수				기타 연산수	총 연산 수	계산시간 (1,000회 반복)
	덧셈	뺄셈	곱셈	나눗셈			
일반적인 방법	291 (300)	3	389 (401)	5	10	698 (719)	3.56초.
본 논문의 방법	8 (14)	7	14 (23)	2	3	34 (49)	0.38초

6. 결 론

3차원 그래픽스에서 필요한 변환 중 많은 연산 횟수로 인하여 계산 시간이 문제가 되는 임의의 축에 관한 회전 이동 변환 행렬을 계산하는 효율적인 알고리즘을 제안하였다.

기존의 일반적인 방법은 행렬의 크기가 4×4인 평행 이동 행렬과 좌표 축에 관한 회전 이동 행렬의 일련의 곱을 단계적으로 수행하므로 많은 연산이 필요하다. 본 알고리즘에서는 조합 행렬의 곱의 결과에서 나오는 각 원소로부터 동일한 요소를 유도해 내어 불필요한 계산을 없앰으로써 연산 횟수를 줄이도록 하였다.

본 연구 결과는 3차원 컴퓨터 그래픽스의 응용, 특히 CAD나 애니메이션 분야 등에 적용 가능하다.

참 고 문 헌

[1] Chan S. Park, Interactive Microcomputer Graphics, Addison-Wesley Publishing Company Inc., 1985.

[2] Ladik Kreinovich and Juli Pena, Fast Rotation of a 3D Image, Computer Graphics Vol. 11, No.2, pp. 121-126, 1990.

[3] Walter F. Taylor, The Geometry of Computer Graphics, Wadsworth & Brooks, Cole Advanced Books & Software, 1992.

[4] Donald Hearn and M. Pauline Baker, Computer Graphics, Prentice Hall Inc., 1988.

[5] Mikell P. Groover and Emony W. Zimmers, CAD/CAM:Computer Aided Design and Manufacturing, Prentice Hall Inc., 1984.

[6] Michael F. Hordeski, CAD/CAM Techiques, Reston Publishing Company Inc., 1988.

[7] Ellis Horowitz and Sartaj Sahni, Fundmentals of omputer Algorithms, Compuer Science Press, 1978.

김 응 곤

1980년 조선대학교 전자공학과 졸업(학사)
1987년 한양대학교 대학원 전자공학과(공학석사)
1992년 조선대학교 대학원 전자공학과 전산전공(공학박사)
1984년~85년 금성반도체(주) 연구소 연구원
1987년~91년 국방과학연구소 선임연구원
1991년~93년 여수수산대학교 전임강사
1993년~현재 순천대학교 전자계산학과 조교수
관심분야 : 컴퓨터그래픽스, CAD.

허 영 남

1967년 공주사범대학 졸업(학사)
1982년 조선대학교 대학원 전자공학과(공학석사)
1992년~현재 조선대학교 대학원 전산통계학과 박사과정수료
1983년~86년 순천대학교 전자계산소장
1983년~현재 순천대학교 전자계산학과 교수
관심분야 : 영상처리, 병렬처리.

이 웅 기

1975년 조선대학교 전자공학과 졸업(학사)
1981년 명지대학교 대학원 전자공학과(공학석사)
1980년~88년 조선대학교 컴퓨터공학과 조교수
1985년~86년 미국 Oregon Graduate Center 객원 교수
1988년~현재 조선대학교 전자통계학과 부교수
관심분야 : 영상처리, 계산기 구조.