

# 고성능/저전력 3D 기하 연산을 위한 오프라인 CORDIC 벡터회전 알고리즘

## (Off-line CORDIC Vector Rotation Algorithm for High-Performance and Low-Power 3D Geometry Operations)

김은옥<sup>†</sup> 이정근<sup>\*\*</sup>  
(Eunok Kim) (Jeong-Gun Lee)

이정아<sup>\*\*\*</sup>  
(Jeong-A Lee)

**요약** 본 논문에서는 모바일 환경에서의 3D 그래픽 처리에 효과적인 고성능/저전력의 CORDIC 구조를 구성하기 위하여 각도 기반 검색(ABS)과 스케일링 효과를 고려한 검색(SCS)과 같은 두 가지 오프라인 벡터링 방법을 제안하고 이를 통해 연산의 반복횟수를 줄이는 알고리즘을 개발한다. ABS 알고리즘은 3차원 벡터를 두 각으로 표현하고 이를 검색의 기준으로 삼았고, SCS 알고리즘은 단위 벡터를 기준으로 하여 벡터 회전 시에 최소의 반복 연산만으로도 원하는 회전을 수행할 수 있는 최적의 기본각 회전 시퀀스를 오프라인으로 미리 검색하여 적용한다. 본 논문에서 제안하는 ABS, SCS 알고리즘을 통해 지연을 각각 50% 감소시킬 수 있었으며, 이와 함께 voltage scaling 기술을 적용하여 전력 소모를 크게 감소시킬 수 있음을 논의한다.

- 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-D00014-I00277)
- 이 논문은 2008 한국컴퓨터종합학술대회에서 '고성능/저전력 3D 기하 연산을 위한 오프라인 CORDIC 벡터회전 알고리즘'의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 학생회원 : 조선대학교 컴퓨터공학과  
eunok26@gmail.com

<sup>\*\*</sup> 정회원 : 한림대학교 컴퓨터공학과 교수  
Jeonggun.Lee@hallym.ac.kr

<sup>\*\*\*</sup> 종신회원 : 조선대학교 컴퓨터공학과 교수  
jalee@chosun.ac.kr

논문접수 : 2008년 8월 26일

심사완료 : 2008년 10월 20일

Copyright©2008 한국정보과학회 : 개인 복제나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제8호(2008.11)

키워드 : 3D 벡터회전, CORDIC, 기하 연산

**Abstract** In this paper, to make a high performance and low power CORDIC architecture for 3D operations in mobile devices, we suggest two off-line vectoring algorithms named Angle Based Search (ABS) and Scaling Considered Search (SCS). The ABS algorithm represents a 3D vector with two angles and those angles are used as a condition for searching CORDIC rotation sequences. The SCS algorithm determines the best CORDIC rotation sequence in advance to eliminate extra scaling computation. Using the proposed algorithms, we can observe 50% of latency is reduced. Furthermore, we perform a simple analysis and discuss possible reduction of power consumption by applying voltage scaling method together with the proposed algorithm.

**Key words** : 3D Vector rotation, CORDIC, Geometry

## 1. 서론

3차원 컴퓨터 그래픽 연산을 위한 고성능/저전력 그래픽 프로세서의 필요성이 증가하면서 이를 만족시키기 위한 다양한 연구가 시도되어 왔으며, 현재도 많은 연구가 진행되고 있다. 과거에는 3D 그래픽스 처리의 전 과정을 CPU 프로그래밍을 기반으로 가속시켜 왔기 때문에 CPU의 부담이 가중되었다. 그러나 입체감과 사실감을 높이기 위해 더욱 향상된 그래픽 처리 능력이 요구되면서, 이를 위한 하드웨어 가속 구조가 개발되었다. 초기 PC용 GPU는 래스터화(rasterization) 단계만을 가속할 수 있었지만, 이후 기하(geometry) 연산 과정까지 포함된 GPU가 개발되었고, 최근에는 이러한 기술들이 점차 휴대용 기기에도 적용되어 가고 있다. 휴대용 단말에서는 하드웨어 면적과 전력공급이 제한적인 특성 때문에 고성능 요구조건 뿐만 아니라 저전력 및 저비용의 조건을 만족시킬 수 있는, 보다 단순하고 효율적인 구조가 필요하다.

이러한 연구로서, Lang등[1]은 2차원 CORDIC(COordinate Rotational Digital Computer) 알고리즘을 확장하여 3차원 벡터 회전을 CORDIC형의 프리미티브(primitives)로 수행할 수 있도록 한 CORDIC-3D 알고리즘을 제안 하였다.

CORDIC-3D의 기반이 된 CORDIC[2]은 삼각함수를 비롯한 여러 가지 초월 함수의 계산을 효율적으로 할 수 있는 연산 기술로, 쉬프트와 덧셈만을 이용하여 벡터회전과 같은 복잡한 연산을 수행할 수 있다는 고유의 하드웨어 간결성으로 벡터연산을 기반으로 하는 다양한 분야에서 활용되고 있다. 그러나, 비트 폭에 비례하는 많은 반복 연산 때문에 계산 속도가 늦다는 단점이 있었다.

이러한 점을 개선하기 위해 현재까지 많은 수정 알고리즘이 제안되어 왔으며, 최근 오프라인으로 주어질 복

표각을 최소의 반복 연산만으로 구현 할 수 있는 방법인 MVR(Modified Vector Rotational)-CORDIC[3]이 제안되었다. MVR-CORDIC은 기본각(elementary angle)을 반복해서 사용하거나 생략할 수 있도록 알고리즘을 수정하여 회전 시에 필요한 연산의 반복횟수를 감소시켰다. 또 다른 개선 알고리즘인 MSR(Mixed Scaling Rotation)-CORDIC[4]은 회전식 내에 스케일링 처리를 포함하여 회전과 스케일링을 동시에 할 수 있도록 하였다. 다만, 이러한 알고리즘들은 오프라인 기법의 알고리즘이기 때문에 목표 각이 사전에 알려진 경우에 대해서만 적용할 수 있다는 한계점을 갖고 있다.

CORDIC-3D는 2차원 연산에만 사용되던 CORDIC을 3차원으로 확장하여 매우 효과적인 3D 벡터회전 연산으로 볼 수 있었다. 그러나, 기본 CORDIC을 바탕으로 하였기 때문에 CORDIC 알고리즘의 문제점을 그대로 가지게 되어, 정해진 비트 폭 만큼의 반복 연산을 모두 수행해야 하므로 지연 성능 특성이나 전력 소모 측면에서 시스템의 설계 사양을 만족시키지 못할 수 있다.

본 논문의 저자는 3D 그래픽 처리에서도 동적으로 입력되는 데이터에 의존한 응용분야가 있는 반면, 그래픽 영화와 같이 그래픽 데이터가 미리 결정되어 반복적으로 사용되는 분야가 있다는 사실에 주목하며, 이와 같이 동적인 변화가 필요없는 그래픽 응용 분야에서 CORDIC-3D의 구조를 개선하여 최소의 반복 연산으로 그래픽 처리가 가능한 방법을 제안한다.

본 연구에서 제안하는 방법은 일반적으로 MPEG 영상에서 데이터 압축을 위해 실제 데이터를 저장하기 보다는 데이터간의 관계인 모션 벡터(motion vector)를 저장하는 것과 같은 아이디어에서 출발한 것이다. 그래픽에서 회전 연산을 수행할 때, 기존의 CORDIC 방식을 이용하여 많은 수의 기본각 회전을 하는 대신 회전이 적용될 원 데이터와 회전후의 데이터간의 관계를 나타내는 회전각을 본 논문에서 제안하는 개선된 오프라인 알고리즘을 통해서 미리 계산된 최소의 기본각들로 표현하고, 실제로 회전할 때에 이 기본각들을 통해 회전을 수행할 수 있다.

논문의 구성은 다음과 같다. 2장에서는 본 논문의 배경이 되는 관련연구로 CORDIC 알고리즘과 CORDIC-3D 회전 알고리즘에 대해 살펴본다. 3장에서는 제안하는 두 가지 오프라인 벡터링 알고리즘과 회전과정을 설명하고, 4장에서 이에 대한 실험결과를 통해 기존의 CORDIC-3D와 성능을 비교한다. 5장에서는 결론 및 향후 과제에 대해 기술한다.

## 2. 기존연구

### 2.1 CORDIC 알고리즘

일반적인 2차원 회전에서 벡터  $[x \ y]^T$ 를 각  $\phi$ 만큼 회전한 벡터  $[x' \ y']^T$ 는 식 (1)과 같이 나타낼 수 있다[2].

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

식 (1)의 회전 행렬을 다음과 같이 변형할 수 있다.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \cos \phi \begin{bmatrix} 1 & -\tan \phi \\ \tan \phi & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

CORDIC 알고리즘은 하드웨어 구조를 단순화 하기위해 식 (2)의  $\tan \phi$ 를 2의 제승 형태로 정의하여 쉬프트 연산으로 처리해보려는 시도에서 출발하였다. 식으로 표현하면 식 (3)과 같다.(여기서,  $i=0,1,2,\dots,N-1$ )

$$\tan \phi = 2^{-i} \quad (3)$$

이를 만족하는 각  $\phi$ 가 CORDIC의 기본각으로 정의되며, 식 (3)에 의해 식 (2)는 식 (4)와 같이 다시 정리할 수 있다.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \cos \phi \begin{bmatrix} 1 & -2^{-i} \\ 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4)$$

식 (4)에서  $\cos \phi$ 를 제외하면 각의 회전을 VLSI 회로로 구현하기 쉬운 쉬프트와 덧셈 연산만을 이용하여 구성할 수 있음을 알 수 있다.

CORDIC 알고리즘에서는  $\tan \phi$ 로 이루어지는 의사 회전(pseudo-rotation)을 통해 목표 각 만큼의 회전을 수행한 후에 일괄적으로  $\cos \phi$ 의 효과를 적용하며, 이를 스케일 팩터(scale factor) 보상 또는 스케일링 이라 한다. 일반적으로 스케일 팩터는  $K$ 로 나타낸다.

회전 목표각이  $\theta$ 일 때, 기본 CORDIC 알고리즘에서  $\theta$ 는 식 (5)과 같이 기본각의 조합으로 표현할 수 있다.

$$\theta = \sum_{i=0}^{N-1} \mu(i)e(i) + \xi_n \quad (5)$$

$N$ 은 반복횟수를 나타내며, 이는 기본각인  $e(i)$ 의 개수와 동일하다.  $\mu(i)$ 는 각 반복에서 현재 위치와 목표각의 위치를 비교하여 +1 또는 -1로 결정된다.  $\xi_n$ 은 누적된 부분각과 실제 목표각  $\theta$ 와의 차이인 오차값이다.

(알고리즘 1)은 기본 CORDIC 알고리즘에서 원형 모드에 대한 기본각 회전과 스케일링을 요약한 것이다.

1) 초기화 :  
 $x(0), y(0), z(0)$ .

2) 반복식 :  
 For  $i = 0$  to  $N - 1$   
 $\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & -\mu(i)2^{-i} \\ \mu(i)2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}$   
 $a(i+1) = a(i) - \mu(i)e(i), e(i) = \tan^{-1}(2^{-i})$   
 End

3) 스케일 보상:  
 $\begin{bmatrix} x_f \\ y_f \end{bmatrix} = K \begin{bmatrix} x(N) \\ y(N) \end{bmatrix} = \left( \prod_{i=0}^{N-1} \sqrt{1+2^{-2i}} \right)^{-1} \begin{bmatrix} x(N) \\ y(N) \end{bmatrix}$

(알고리즘 1) 기본 CORDIC의 회전식 요약

2.2 CORDIC-3D 회전 알고리즘

[1]에서 제안한 CORDIC-3D 연산은 2차원 벡터회전을 주로 수행하던 기본 CORDIC을 3차원 처리를 위하여 확장한 것이다.

CORDIC-3D는 벡터링 모드와 벡터링 후 회전의 두 가지 모드로 구성된다. 벡터링 연산은 하나의 3차원 상의 벡터를 3D 기본 회전을 이용하여 두 개의 회전축을 기준으로 나머지 하나의 축으로 접근하도록 회전한다. 벡터링 후 회전 연산은 벡터링 과정에서 사용된 회전 방향과 같은 3D 기본 회전의 내용을 다른 3차원 벡터들에 적용할 수 있다. CORDIC-3D의 알고리즘은 다음과 같이 나타낼 수 있다.

```

1) 초기화 :  $(x[1], y[1], z[1]) = (x_0, y_0, z_0)$ 
2) 반복식 :
For j = 1 to N
  if ( 벡터링 모드 )
    then // [벡터링] 연산인 경우
       $(s_{1,j}, s_{2,j}) = (\text{sign}(x[j])\text{sign}(y[j]), \text{sign}(x[i])\text{sign}(z[i]))$ 
    else // [벡터링 후 회전] 연산인 경우
      precomputed  $s_1, s_2$ 
    Endif
     $x[j+1] = x[j] - 2^{-2j+1}x[j] + s_{1,j}2^{-j+1}y[j]$ 
       $+ s_{2,j}2^{-j+1}z[j]$ 
     $y[j+1] = y[j] - s_{1,j}2^{-j+1}x[j] - s_{1,j}s_{2,j}2^{-2j+1}z[j]$ 
     $z[j+1] = z[j] - s_{2,j}2^{-j+1}x[j] - s_{1,j}s_{2,j}2^{-2j+1}y[j]$ 
  Endfor
3) 스케일 보상 :
 $x_f = K_{3D}x[N+1]$ 
 $(K_{3D} = \prod_{i=1}^N 1/(1+2^{-2i+1}))$ 
    
```

(알고리즘 2) CORDIC-3D 회전 알고리즘

초기 위치에서 시작하여 위의 반복 식에 의해 전체 반복 횟수인 N만큼 회전을 수행하게 되는데, 벡터링에서 회전의 방향  $s_{1,j}, s_{2,j}$ 는 각각 x와 y좌표의 곱의 부호, x와 z좌표의 곱의 부호로 결정된다. 2D CORDIC에서와 마찬가지로 회전 후에 변경된 벡터의 크기를 복구시켜 주기 위해 스케일링 단계를 거친다. 여기에서는 스케일 팩터를  $K_{3D}$ 로 나타내었다. 벡터링 단계의 결과로 시퀀스  $s_1, s_2$ 와  $x_f$ 가 결정되고, 벡터링 후 회전 과정에서 미리 계산된 시퀀스를 이용하여 실제 회전을 수행하게 된다.

3. 제안하는 CORDIC 기반 회전 알고리즘

앞서 살펴본 CORDIC-3D는 기본 CORDIC 알고리즘을 확장하였기 때문에 데이터 패스의 비트 폭 'N'만큼의 반복을 수행해야 했다. 이로 인한 오버헤드를 감소시키기 위해 본 논문에서는 2차원 CORDIC 알고리즘의 개선을 위해 제안되었던 MVR-CORDIC, MSR-CORDIC과 같은 맥락으로 오프라인 상에서 주어진 입력에 대한 최적의 시퀀스를 검색하고 이를 통해 회전함으로써 반복횟수를 줄이는 방법을 제안한다. 제안 알고리즘은 오

프라인 벡터링 단계와 검색한 시퀀스로 실제 회전을 수행하는 벡터링 후 회전단계로 구성된다. 오프라인 벡터링에서는 각도 기반 검색과 단위벡터 기반으로 스케일링을 고려한 검색의 두 가지를 제시하고 이를 통해 최적의 시퀀스를 결정하여 반복횟수를 줄이고자 하였다.

3.1 오프라인에서의 시퀀스 검색(오프라인 벡터링)

3.1.1 각도 기반 검색(Angle Based Search:ABS)

3차원 벡터 표현에서 축과 각 표현법에 따르면 3차원 상의 벡터는 두 개의 각으로 나타낼 수 있다. 이에 착안하여 그림 1과 같이 벡터 P를  $\theta$ 와  $\phi$ 의 두 각으로 표현하고 이를 검색의 기준으로 활용하였다.

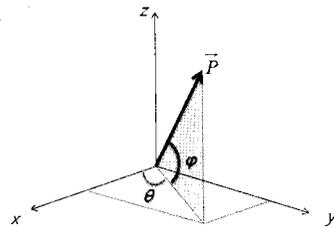


그림 1 두 각에 의한 3차원 벡터의 표현

벡터링의 결과로 벡터 P가 x축에 접해야 하므로, 두 각  $\theta$ 와  $\phi$ 가 얼마나 0에 수렴하는지를 시퀀스 결정의 기준으로 삼았다. 오차를 측정하는 지표로서  $e_\theta, e_\phi$ 를 각각  $\theta$ 와  $\phi$ 의 절대값인  $|\theta|, |\phi|$ 로 정의하였다.

본 논문에서는 시퀀스 결정을 위한 최적화 알고리즘으로 그리디(greedy) 탐색법을 적용하였다. ABS 알고리즘의 흐름은 (알고리즘 3)과 같다.

```

1) 초기화:  $i = 1, e_{\min} = \infty$ 
2) 입력좌표 U, 워드길이 W, 반복횟수 N이 주어짐
3)  $d_1(i), d_2(i) \in \{-1, 0, 1\}$ ,  $s(i) \in \{1, 2, 3, \dots, W\}$ 일 때, 각 경우에 회전 후의  $U'(x(i+1), y(i+1), z(i+1))$ 와  $e_\theta(i), e_\phi(i)$ 를 계산
4)  $e_\theta(i), e_\phi(i)$ 를 전 단계의 오차 최소값  $e_{\min}$ 과 비교하여 값이 더 작으면  $d_1(i), d_2(i), s(i)$ 를 저장하고,  $e_{\min}$ 을 갱신
5)  $i > N$ 이면 수행 종료. 그 외의 경우,  $i$ 를 1만큼 증가하고 3)단계부터 재 수행
    
```

(알고리즘 3) ABS 알고리즘

$N'$ 은 반복횟수, UI는 입력좌표를 나타낸다. CORDIC-3D 알고리즘이 방향만을 변경하며 회전하였던 것과 달리 ABS는 시퀀스 검색을 통해 각 반복마다 오차값  $e_\theta, e_\phi$ 를 최소화하는 방향시퀀스  $d_1(i), d_2(i)$ 와 회전시퀀스  $s(i)$ 를 찾게 된다. 그림 2는 ABS 알고리즘의 순서도이다.

3.1.2 스케일링을 고려한 검색(Scaling Considered Search: SCS)

두 번째 검색 방법은 스케일 효과를 고려한 시퀀스 검색으로써 입력벡터가 벡터링 수행 후 단위벡터  $v(1,0,0)$ 에 수렴하도록 검색 조건을 준다. 이 방법은 벡터링 과

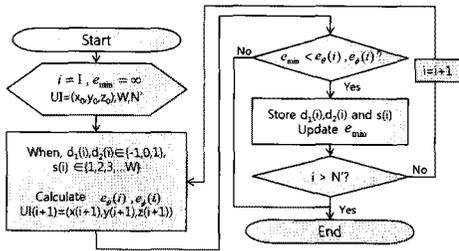


그림 2 ABS의 그리다 탐색 순서도

정에서 결과 벡터의 크기를 최대한 단위 길이로 유지할 수 있는 시퀀스들을 선택하도록 함으로써 스케일링의 오버헤드를 줄일 수 있다. 각도 기반 검색과 마찬가지로 그리다 탐색법을 이용하였고, 검색의 지표로 사용된  $e_n$  는 다음과 같이 계산된다.

$$e_n = n_d - n_f \quad (6)$$

$$n_d = \sqrt{x_d^2 + y_d^2 + z_d^2}, \quad n_f = x(i+1) \quad (7)$$

$n_d$ 는 회전 후 벡터의 이상적인 크기를 나타내며, 본 알고리즘의 벡터링 단계에서는 단위벡터에 접근하도록 하였으므로  $n_d$ 는 1이 된다.

표 1은 입력값  $UI=(0.8232, 0.4194, 0.3827)$ , 반복횟수  $N=4$ 인 경우, 그리다 알고리즘을 이용한 오프라인 검색으로 결정되는 방향과 회전 시퀀스 집합을 각 단계별로 정리한 것이다.

- 1) 초기화:  $i = 1, e_{min} = \infty$
- 2) 입력자료 UI, 워드길이 W, 반복횟수 N이 주어짐
- 3)  $d_1(i), d_2(i) \in \{-1, 0, 1\}, s(i) \in \{1, 2, 3, \dots, W\}$ 일 때, 각 경우에 회전 후의  $UI(x(i+1), y(i+1), z(i+1))$ 와  $e_n$ 을 계산
- 4)  $e_n$ 를 전 단계의 오차 최소값  $e_{min}$ 과 비교하여 값이 더 작으면  $d_1(i), d_2(i), s(i)$ 를 저장하고,  $e_{min}$ 을 갱신
- 5)  $i > N$ 이면 수행 종료. 그 외의 경우,  $i$ 를 1만큼 증가하고 3)단계부터 재 수행

(알고리즘 4) SCS 알고리즘

표 1 UI=(0.8232,0.4194,0.3827)일 때, ABS와 SCS의 시퀀스

시퀀스	반복단계(i)							
	ABS				SCS			
	1	2	3	4	1	2	3	4
$d_1(i)$	1	-1	1	-1	1	-1	1	1
$d_2(i)$	1	-1	0	-1	1	-1	-1	1
$s(i)$	2	5	6	8	2	5	7	8

### 3.2 벡터링 후 회전단계

본 단계에서는 오프라인 단계에서 벡터링을 통해 얻은 방향시퀀스  $d_1, d_2$ 와 회전시퀀스  $s$ 를 이용하여 3차원 벡터 회전을 수행한다.  $d_1, d_2, s$ 는 각 반복단계에서 사용될 방향, 회전 정보를 가진 시퀀스 집합이다. 회전 알고리즘은 (알고리즘 5)와 같이 정리될 수 있다.

```

1) 초기화 : (x[1],y[1],z[1])=(x0,y0,z0)
2) 반복식
For j = 1 to N'
x[j+1] = x[j] - 2^{-2s(j)+1}x[j] + d_1(j)2^{-s(j)+1}y[j]
          + d_2(j)2^{-s(j)+1}z[j]
y[j+1] = y[j] - d_1(j)2^{-s(j)+1}x[j] - d_1(j)d_2(j)2^{-2s(j)+1}z[j]
z[j+1] = z[j] - d_2(j)2^{-s(j)+1}x[j] - d_1(j)d_2(j)2^{-2s(j)+1}y[j]
End
    
```

(알고리즘 5) 벡터링 후 회전 알고리즘

## 4. 실험결과 및 고찰

본 절에서는 앞서 제안한 ABS, SCS의 두 오프라인 알고리즘과 기존의 CORDIC-3D 알고리즘의 성능을 비교한다. 그림 3은 ABS 방식을 적용한 경우의 반복횟수에 따른 오차값의 변화를 나타낸 것이다.  $e_\theta, e_\phi$ 의 두 오차값에 동일하게 비중을 두기 위해 두 값의 평균치를 계산한  $e_{avg}$ 를 기준으로 나타내었다. 또한 비교를 위해 기존 CORDIC-3D 알고리즘에 대한 변화 그래프도 함께 표시하였다. 그림을 통해 알 수 있듯이, 기존의 CORDIC-3D가 N번의 반복연산으로 달성할 수 있는 오차값을 ABS 알고리즘은 그 절반 수준인 N/2 정도의 반복만으로 달성할 수 있음을 알 수 있다.

SCS 알고리즘에 대해서도 앞에서 정의한 오차지표  $e_n$ 을 통해 반복횟수에 따른 오차값의 변화를 그림 4로 나타내었다. 이 경우에는 CORDIC-3D와 비교하여 약 2/3의 반복만으로 동일한 성능을 내는 것으로 나타났다.

SCS 알고리즘의 경우, 검색과정 자체에 스케일링 효과가 적용되어 있기 때문에 적절한 비교를 위해서는 벡터 회전 단계의 반복횟수 N과 함께 스케일링에 소요되는 반복횟수까지 함께 고려하여야 한다.

CORDIC-3D에서 회전 단계와 스케일링 단계에서 소요되는 반복횟수의 비율을 CORDIC에서와 비슷한 수준으로 예상하고 회전 단계의 반복횟수가 N일 때 스케일링 단계에서는 약 N/3이 소요된다고 하면[5], CORDIC-3D와 ABS 알고리즘의 전체 반복연산 횟수는 아래의

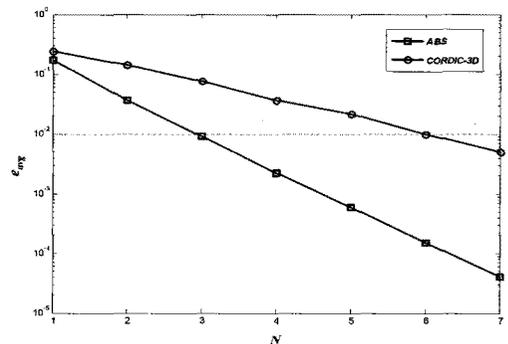


그림 3 CORDIC-3D와 ABS 알고리즘의 반복횟수에 따른  $e_{avg}$ 의 변화

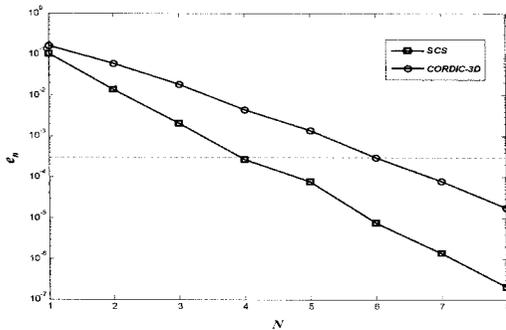


그림 4 CORDIC-3D와 SCS 알고리즘의 반복횟수에 따른  $\epsilon_n$ 의 변화

표 2와 같이 회전 단계와 스케일링 단계의 횟수를 합하여 각각  $4N/3$ ,  $2N/3$ 으로 계산된다.

즉, 본 논문에서 제안하는 알고리즘을 적용하였을 경우에는,  $2N/3$  정도의 반복만으로도 기존의 CORDIC-3D와 동일한 오차 수준을 갖도록 3D 회전연산을 수행시킬 수 있다. 따라서, 전체 회전에 소요되는 시간에서 50%의 감소 효과를 보임으로써 계산 속도를 크게 향상시킬 수 있다.

• **전력 소모 감소에 대한 논의:** 디지털 회로의 전력 소모  $P$ 는 “ $C(V_{DD})^2f$ ”에 비례한다. 회로의 지연시간  $T$ 는  $V_{DD}$ 에 의존하며 다음 식으로 기술된다[6].

표 2 CORDIC-3D와 ABS, SCS 알고리즘의 반복연산 횟수 비교

	CORDIC-3D[2]	ABS	SCS
반복횟수	$4N/3$ = $N/3(\text{rotation})$ + $N/3(\text{scaling})$	$2N/3$ = $N/2(\text{rotation})$ + $N/6(\text{scaling})$	$2N/3$

$$T = \frac{C \times V_{DD}}{\mu C_{ox} (W/L) (V_{DD} - V_{th})^2} \quad (8)$$

최신 45nm CMOS 공정에서 1V의  $V_{DD}$ 와 0.18V의  $V_{th}$ 를 고려하자[7]. 본 논문에서는 제안한 오프라인 3D CORDIC 알고리즘인 MSR-3D를 적용하여, 기존의 성능 지연  $T$ 를 반으로 줄여  $T/2$ 로 만들었기 때문에, 향상된 성능을 전력소모의 감소로 변환시키기 위하여  $V_{DD}$ 를 감소시켜 성능은 기존의 방식에 맞추면서 전력소모는 크게 감소시킬 수 있다. 45nm CMOS의 경우,  $V_{DD}$ 를 1V에서 0.646V까지 감소시킬 수 있으며, 이 경우, 기존 CORDIC-3D 알고리즘과 유사한 성능을 갖게 되며, 이때의 전력 소모는 “ $C(0.646 \cdot V_{DD})^2f = 0.4173 \cdot C(V_{DD})^2f$ ”로 기술된다. 이는 기존 CORDIC-3D 알고리즘에 비해 같은 성능을 유지하면서, 기존 방법의 전력 소모의 약 42%만이 필요하다는 것을 의미한다. 설계자는 전압을 적절히 조절하여 필요한 성능과 전력소모 요건에 맞는 설계를 구성할 수 있다.

### 5. 결론 및 향후 과제

본 논문에서는 모바일 기기에 적합한 고성능 3D 기하 연산을 위해, 2차원 CORDIC을 위해 제안된 오프라인 기법을 바탕으로 3차원 벡터회전을 개선할 수 있는 오프라인 알고리즘인 ABS, SCS를 제안하고 성능을 평가하였다. 본 논문에서 제안한 두 알고리즘은 방향시퀀스만을 사용하던 기존 방식에서, 방향과 회전 시퀀스를 이용하여 회전할 수 있도록 알고리즘을 확장하였다. 오프라인 응용에만 적용된다는 단점을 갖고 있으나, 제안된 알고리즘은 기존의 CORDIC-3D 알고리즘에 비해 전체 회전 소요시간에서 각각 50%의 감소 효과를 보였다. 더불어 향상된 성능 향상은 전력 소모 감소로 이루어 질 수 있으며, 최신 45nm CMOS 공정의 파라미터 값을 고려했을 때 최대 기존 전력 소모의 42%만을 사용함을 토의를 통해 살펴보았다.

본 논문의 제안 내용을 토대로 3D 그래픽 처리를 필요로 하는 휴대용 기기에 적용하여 고성능 및 저전력의 특성을 만족시키는 효과적인 기하 연산구조를 구성할 수 있을 것으로 기대된다. 향후, 특정 3D 응용에 해당하는 벤치마크들을 구성하고, 실제 3D 응용에서 수행되는 전체 연산 중 3D rotation의 비중을 분석하는 일과 Real 3D application의 구현을 통한 전체 시스템의 성능향상 폭을 평가하는 작업은 앞으로 수행해야 할 과제이다.

### 참고 문헌

- [1] Tomás Lang and Elisardo Antelo, "High-Throughput CORDIC-based Geometry Operations for 3D Computer Graphics," IEEE Trans. on Computers, Vol.54, No.3, pp.347-361, Mar. 2005.
- [2] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Computers, Vol.8, pp.330-334, Sept. 1959.
- [3] C. S. Wu and A. Y. Wu, "Modified vector rotational CORDIC(MVR-CORDIC) algorithm and architecture," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Processing, Vol. 48, No. 6, pp.548- 561, Jun. 2001.
- [4] C. H. Lin, A. Y. Wu, "Mixed-scaling-rotation CORDIC(MSR-CORDIC) algorithm and architecture for high performance vector rotational DSP applications," IEEE Trans. Circuits and Systems Part-I: Fundamental Theory and Applications, Vol. 52, No. 11, pp.2385-2396, Nov. 2005.
- [5] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," IEEE Signal Processing Mag., pp.16-35, July 1992.
- [6] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, "Low Power CMOS Digital Design," IEEE Journal of Solid State Circuits, Vol. 27, pp 473-484, 1992.
- [7] "Predictive Technology Model (PTM), <http://www.eas.asu.edu/~ptm>"