

## 두 토러스의 교차곡선에서 이차곡선의 발견을 위한 절차적 방법

김구진\*, 김명수\*\*

### Procedural Method for Detecting Conic Sections in the Intersection of Two Tori

Kim, K. J.\* and Kim, M. S.\*\*

#### ABSTRACT

This paper presents a geometric method that can detect and compute all conic sections in the intersection of two tori. Conic sections contained in a torus must be circles. Thus, when two tori intersect in a conic section, the intersection curve must be a circle as well. Circles in a torus are classified into profile circles, cross-sectional circles, and Yvone-Villarceau circles. Based on a geometric classification of these circles, we present a procedural method that can detect and construct all intersection circles between two tori. All computations can be carried out using simple geometric operations only: e.g., circle-circle intersections, circle-line intersections, vector additions, and inner products. Consequently, this simple structure makes our algorithm robust and efficient, which is an important advantage of our geometric approach over other conventional methods of surface intersection.

**Key words :** Torus, Intersection curve, Conics, Profile circle, Cross-sectional circle, Yvone-Villarceau circle

#### 1. 서 론

CAD 및 기하 모델링 시스템에서 평면(plane), 구(sphere), 원기둥(cylinder), 원뿔(cone)과 같은 2차 곡면과 토러스(torus)는 매우 빈번히 사용되는 기본 입체이다. 사용자는 이들 기본입체에 여러 가지 연산을 적용하여 자신이 원하는 입체를 구성하는데, 연산 중에서도 합집합, 교집합, 차집합 등과 같은 Boolean 연산은 가장 기본적으로 사용되는 연산이다. 기본입체에 대한 Boolean 연산을 수행한 결과가 정확하고 빠르게 계산되는지의 여부는 시스템 자체의 성능과 robustness를 좌우하는 중요한 문제이다. 곡면에 대한 교차곡선 계산은 이러한 Boolean 연산을 구현하기 위하여 반드시 필요한 과정이므로 대단히 중요하다.

현재까지 2차 곡면 간의 교차곡선 계산에 대해 수 많은 연구가 수행되었고, 이 중에서도 특히 교차곡선에

존재하는 2차 곡선을 계산하는 방법에 대해서도 독립적인 연구들이 많이 수행되어 왔다<sup>[1,2,8-13,15-17]</sup>. 보통 교차곡선은 주어진 시작점으로부터 전체 곡선을 따라 추적하는 marching 기법, 또는 곡면들을 분할하며 bounding box test를 통해 교차곡선에 속하는 점들을 찾아내고 이 점들을 연결하여 교차곡선을 구하는 소위 곡면 분할기법 등에 의해서 구해진다.

일반적으로 두개의 2차 곡면은 4차 교차곡선에서 만나고, NURBS와 같은 스플라인 곡선으로 표현하는 것이 불가능하다. 따라서, 많은 선분들이나 3차 곡선들을 이용하여 주어진 오차 범위 내에서 근사 계산을 하여야 한다. 결과적으로 교차곡선을 정밀하게 계산하기 위해서는 많은 양의 데이터가 필요하다. 그러나, 직선(line)이나 원(circle), 포물선(parabola), 쌍곡선(hyperbola) 등과 같은 2차 곡선이 교차곡선에 포함되는 경우에는 간단한 기하학적 정보만으로 2차 교차곡선을 정확하게 표현할 수 있다. 예를 들어, 원의 경우에는 원의 중점과 원을 포함하는 평면의 법선 벡터(normal vector), 그리고 원의 반경의 정보만으로도 정확한 표현이 가능하다. 이와 같이 기하적

\*정회원, 아주대학교 정보통신전문대학원

\*\*서울대학교 컴퓨터공학부

- 심사투고일: 2000. 7. 25

- 심사완료일: 2000. 8. 29

보만으로 곡선을 표시함으로써 곡선을 더 정확하고 간략하게 표현할 수 있으므로, 교차곡선에 존재하는 2차 곡선을 발견하고 기하정보를 계산해내는 것은 중요한 문제이다.

토러스(torus)는 CAD 및 모델링 시스템에서 자주 사용되는 기본입체 중의 하나이다. 현재까지 2차 곡면 간의 교차곡선이나 여기에 존재하는 2차 교차곡선을 찾는 문제에 대한 수많은 연구가 수행된 반면<sup>[1,4,8-11,15-17]</sup>, 토러스와 2차 곡면, 또는 토러스와 토러스 간의 교차곡선에 대한 연구는 상대적으로 그 수가 적다<sup>[5,7]</sup>. 그 이유는 교차곡선의 차수에 있다. 2차 곡면 간의 교차곡선의 차수는 4차인 데 비해 토러스와 2차 곡면, 또는 토러스와 토러스 사이의 교차곡선은 일반적으로 8차로서 차수가 높으며 이에 따라 교차곡선 자체의 위상(topology)도 훨씬 더 복잡해지므로 교차곡선을 계산하기가 대단히 어려운 것으로 간주되어 왔다.

주어진 토러스와 구, 원기둥, 원뿔, 토러스 사이에 존재하는 8차 교차곡선을 모두 구하는 일반적인 방법은 본 논문의 저자들이 이전연구에서 제시한 바 있다<sup>[5,6]</sup>. 또한, 기하학적 방법을 이용함으로써 토러스와 2차 곡면(구, 원기둥, 원뿔) 간의 교차곡선에서 2차 곡선의 포함 유무를 간단하게 판정하는 알고리즘을 제시한 바 있다<sup>[7]</sup>. 본 논문에서는 이보다 어려운 문제인 토러스와 토러스 사이의 2차 교차곡선을 간단하게 발견하고 이를 구체적으로 계산하는 방법을 다룬다. 두 토러스 사이의 교차곡선은 일반적으로 8차 곡선이 되므로, 2차 교차곡선이 하나나 두 개가 포함하는 경우 나머지의 교차곡선은 6차나 4차 곡선이 된다. 이 나머지 교차곡선은 본 저자들의 이전 연구결과에 의하여 구할 수 있다<sup>[5,6]</sup>.

Kim and Kim<sup>[7]</sup>에서는 주어진 곡면들을 나타내는 기하정보들을 간단한 수식들에 대입하여 이 수식들이 만족되는지의 여부를 따지는 대수적 접근방법이 이용되었다. Miller and Goldman<sup>[12]</sup>에서도 이러한 대수적 접근방법을 이용하여 두개의 2차 곡면사이에 존재하는 2차 교차곡선의 유무를 판정하였다. 대수적 접근방법에서는 수식을 대수적으로 전개하여 symbolic하게 처리하면 수치오차가 전혀 없이 수식의 만족여부를 판정할 수 있다는 장점이 있다. 이에 비하여 Shene and Johnstone<sup>[16]</sup>에서 사용된 절차적 방법은 주어진 입력 데이터로부터 중간과정의 기하학적 데이터를 구하고 이들이 주어진 제한조건들을 만족하는지의 여부를 판정하는 것이다. 중간과정에서 계산되는 데이터에 수치오차가 포함되므로 완벽

하게 정확한 판정을 보장하기 힘들다. 토러스와 토러스의 경우는 그 기하학적인 상관관계가 대단히 복잡하여 2차 교차곡선의 유무를 판정하는 대수적 관계식을 찾는 것이 대단히 어렵다. 따라서, 본 논문에서는 비교적 쉬운 절차적인 접근방법을 제시한다. 그리고, 대수적인 방법을 찾는 문제는 추후 연구에서 계속 시도할 계획이다.

이 논문의 구성은 다음과 같다. 제 2절에서는 토러스 상에 존재하는 유일한 2차 곡선이 원임을 언급하고, 기하학적 정보에 따라 원들을 분류한다. 제 3절에서는 토러스 상에 존재하는 원들의 기하정보를 이용하여 두 개의 토러스 사이에 존재하는 교차원들을 발견하는 방법과 교차원을 구체적으로 계산하는 알고리즘을 제시한다. 제 4절에서는 교차원의 발견 및 계산 알고리즘을 두 개의 토러스 사이의 교차곡선 계산 알고리즘의 일부로서 사용하는 방법에 대해 설명한다. 마지막으로 제 5절에서는 본 논문에 대한 결론을 내린다.

2. 토러스 곡면 상의 원

토러스 곡면 상에 포함되는 유일한 2차 곡선은 원이며, 이 원은 profile circle, cross-sectional circle, 그리고 Yvone-Villarceau circle의 세 가지로 분류된다. 이 사실은 고전기하(classical geometry)에서 이미 잘 알려진 결과이며<sup>[14]</sup>, 초보적인 수학적 지식만을 사용한 증명은 참고문헌<sup>[5,6]</sup>에도 나와 있다. 서로 다른 종류의 원은 이 원이 가진 기하학적 성질에 의해 분류되며, 이러한 기하학적 성질을 이용하여 두 토러스 사이의 교차원을 쉽게 발견할 수 있다.

이 절에서는 3차원 공간에서 표준위치(standard position)에 놓인 minor 반경  $r$ , major 반경  $R$ 인 토러스를 예로 들어 설명한다. 그러면, 토러스는  $xy$ -평면상에서  $(R, 0, 0)$ 을 중심으로 하고 반경이  $r$ 인 원

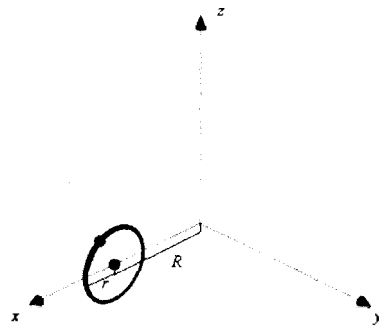


Fig. 1. Minor circle on xz-plane.

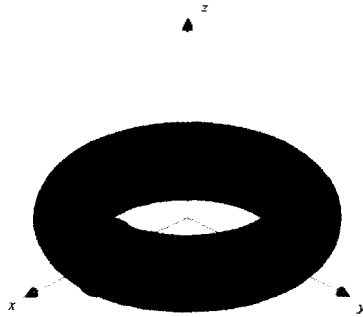


Fig. 2. Torus as a surface of revolution.

을 z축을 중심으로 회전하여 얻어지는 회전곡면으로 볼 수 있다(그림 1과 2참조). 이 때 토러스의 중심점은 원점이고, 점  $(R, 0, 0)$ 을 z축을 중심으로 회전하여 구해지는 원이 주원(main circle)이며, 주원이 존재하는 평면을 토러스의 주평면(main plane)이라 한다. 임의의 위치와 방향을 가진 토러스가 주어질 때, 이 곡면상에 포함된 원들은 표준위치의 토러스 곡면 상의 원들에 회전변환과 이동변환을 적용하여 구할 수 있다.

하나의 원을 나타내는 기하정보는 원의 중심점, 반경, 그리고, 원이 포함된 평면의 법선벡터이다. 이 논문에서는 설명을 간결하고 명확하게 하기 위해 원을 포함하는 평면의 법선벡터라는 표현 대신 원의 법선벡터라는 표현을 사용한다.

2.1 Profile Circle

Profile circle은 토러스의 주평면(main plane)과 평행한 평면이 토러스와 교차할 때 생기는 원이다. 그림 3은 표준위치에 있는 토러스를 주평면(xy-평면)과 평행한 평면으로 잘랐을 때 생기는 profile circle 두개를 보여준다. 그림 4는 이 두개의 profile circle들의 반지름을 계산하는 과정을 xz-단면을 이용하여 보여주고 있다. 점  $(0, 0, z_0)$ 를 지나고 xy-평면에 평행한 평면으로 토러스를 자를 때,  $z_0 = \pm r$ 일 경우에

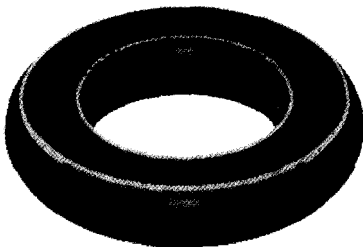


Fig. 3. Profile circles.

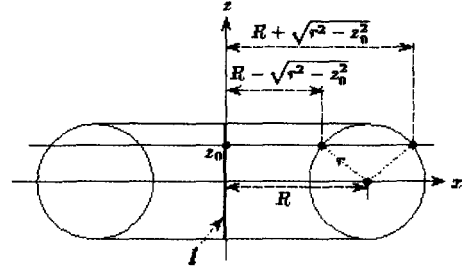


Fig. 4. Constructioin of profile circles.

는 하나의 profile circle이,  $-r < z_0 < r$ 일 경우에는 두개의 profile circle이 생긴다. Profile circle의 법선 벡터는 항상 z축과 평행이다. Profile circle의 중심점은 z축 위에 존재하는데, 이 중심점의 궤적은 다음과 같은 선분 l로 표시된다:

$$l = \{(0, 0, z_0) | -r \leq z_0 \leq r\}.$$

중심이  $(0, 0, z_0)$ 인 profile circle의 반경은  $R \pm (r^2 - z_0^2)^{1/2}$ 이다.

2.2 Cross-Sectional Circle

z축을 포함하는 평면으로 토러스를 자를 때 생기는 두개의 원이 cross-sectional circle이다. 그림 5는 토러스에 포함된 여러 개의 cross-sectional circle들을 보여준다. 그림 6은 cross-sectional circle들의 위치와 방향을 구하는 과정을 xy-단면을 이용하여 보여준다. 이 그림에서 P는 z축을 포함하는 평면이며, 토러스와 P가 교차할 때, 두 개의 cross-sectional circle이 생긴다. Cross-sectional circle의 반경은 토러스의 minor 반경 r과 같다. 각 cross-sectional circle의 중심점은 토러스의 주원 위에 존재하는데, 점 q가 토러스 주원 상의 한 점이라고 할 때, q를 중심으로 하는 cross-sectional circle의 법선 벡터는 점 q에서 주원의 접선 벡터(tangent vector)와 같다. 일반적으로 표준위치에 있지 않는 토러스의 중심점이 p이고 법선

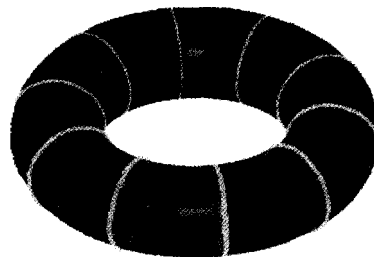


Fig. 5. Cross-sectional circles.

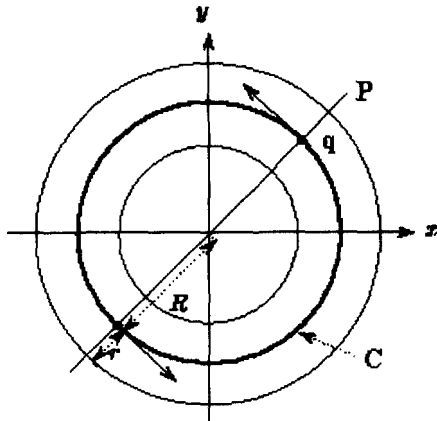


Fig. 6. Construction of cross-sectional circles.

벡터가  $N$ 이라 할 때,  $q$ 를 중점으로 하는 cross-sectional circle의 법선 벡터는  $N \times (q - p)$ 으로 주어진다.

### 2.3 Yvone-Villarceau Circle

토러스의 중심점을 지나면서 토러스와 두 점에서 접하는 평면  $P$ 가 존재함을 쉽게 보일 수 있다. Yvone-Villarceau circle은 이러한 평면  $P$ 로 토러스를 자를 때 생긴다. 그림 7은  $y$ -축을 포함하는 평면  $P$ 로 토러스를 자른  $xz$ -단면을 보여준다. 토러스를  $xz$ -평면으로 자르는 경우 반경  $r$ , 중심점  $(-R, 0, 0)$ 인 두 개의 원이 생기고,  $y$ -축을 포함하며 이 원들과 두 점에서 접하는 평면은 두 개가 존재한다. 이 평면들은 원래의 토러스와도 이 두점에서 접하며, 각각  $(\pm r/R, 0, (R^2 - r^2)^{1/2}/R)$ 을 법선벡터로 갖는다.

그림 8은  $y$ -축을 포함하고 법선벡터가  $(+r/R, 0, (R^2 - r^2)^{1/2}/R)$ 인 평면으로 토러스를 자를 때 생기는 두개의 Yvone-Villarceau circle들 중의 하나를  $xy$ -평

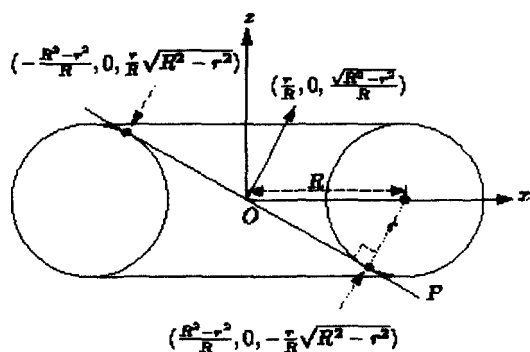


Fig. 7. Sectional view of torus and P on  $xz$ -plane.

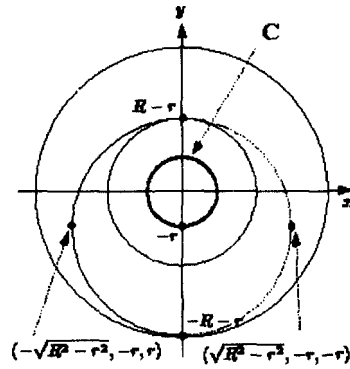


Fig. 8. Construction of yvone-villarceau circles.

면에 투영시킨 결과를 보여주고 있다. 이 Yvone-Villarceau circle은  $(0, -r, 0)$ 에 중심을 가지고 반경이  $R$ 인 원이다. 이 원은  $xy$ -평면과는 평행하지 않으므로, 투영된 결과가 타원으로 나타난다. 하지만,  $y$ -축 방향으로는 투영과정에서도 길이가 보존되므로 이 원의 반지름이  $R$ 이라는 사실을 쉽게 알 수 있다.

평면  $P$ 가  $z$ -축을 중심으로 회전하며 토러스를 자를 때, 토러스 상의 모든 Yvone-Villarceau circle들을 구할 수 있다. 따라서 Yvone-Villarceau circle의 반경은 항상 토러스 주위의 반경( $R$ )과 동일하다는 것을 알 수 있다. 토러스의 중심점을 중점으로 하고 토러스의 주평면 위에 놓인 반경  $r$ 인 원을  $C$ 라고 할 때, Yvone-Villarceau circle의 중심점은 항상 원  $C$  위에 존재한다.

일반적인 위치와 방향을 가진 토러스의 경우, 즉 토러스의 중심점이  $p$ , 법선벡터가  $N$ 인 경우, 점  $q$ 를 중점으로 갖는 Yvone-Villarceau circle의 법선벡터  $N_q$ 는 다음과 같다.

$$N_q = (\pm(q - p) \times (N + (R^2 - r^2)^{1/2}N))/R.$$

그림 9는 같은 평면에 포함되는 두개의 Yvone-Villarceau circle들을 보여준다.

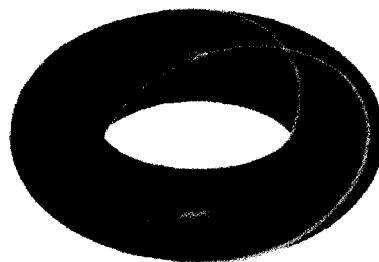


Fig. 9. Yvone-villarceau circles.

### 3. 두 토러스 사이의 교차원 계산과 발견

토러스  $T$ 의 profile circle의 집합을  $PF(T)$ 로, cross-sectional circle의 집합을  $CS(T)$ 로, Yvone-Villarceau circle의 집합을  $YV(T)$ 로 나타내면, 두 토러스  $T_1, T_2$ 에 공유되는 교차원의 집합은 다음과 같은 9개의 교집합들에 대한 합집합으로 구해진다.

1.  $PF(T_1) \cap PF(T_2)$
2.  $PF(T_1) \cap CS(T_2)$
3.  $PF(T_1) \cap YV(T_2)$
4.  $CS(T_1) \cap PF(T_2)$
5.  $CS(T_1) \cap CS(T_2)$
6.  $CS(T_1) \cap YV(T_2)$
7.  $YV(T_1) \cap PF(T_2)$
8.  $YV(T_1) \cap CS(T_2)$
9.  $YV(T_1) \cap YV(T_2)$

이 절에서는  $PF(T_1) \cap PF(T_2), PF(T_1) \cap CS(T_2), PF(T_1) \cap YV(T_2), CS(T_1) \cap CS(T_2), CS(T_1) \cap YV(T_2)$ , 그리고,  $YV(T_1) \cap YV(T_2)$ 의 계산 방법을 설명하며, 설명 되지 않은 교집합은 이 절에서 설명된 내용에 의해 쉽게 추론할 수 있다.

표 1에서는 이 논문에서 사용하는 토러스의 표기 방법을 설명한다. 토러스  $T_1 = T_{r_1, R_1}(p_1, N_1)$ 과 토러스  $T_2 = T_{r_2, R_2}(p_2, N_2)$ 를 가정하여 교차원의 발견과 계산방법을 설명한다. 그리고, 설명을 단순화 시키기 위하여 일반성을 잃지 않고  $p_1 = (0, 0, 0)$ 이며  $N_1 = (0, 0, 1)$ 이라고 가정한다.

#### 3.1 $PF(T_1) \cap PF(T_2)$

이 절은  $T_1$ 의 profile circle과  $T_2$ 의 profile circle 간의 교집합을 발견하는 방법을 설명하고 이를 계산하는 알고리즘을 제시한다.  $PF(T_1)$ 과  $PF(T_2)$  간의 교집합이 존재하기 위한 조건은 다음과 같다.

$C_1, C_2$ 가 각각  $PF(T_1), PF(T_2)$ 에 속하는 임의의 원이라 하자.  $C_1$ 과  $C_2$ 가 일치하기 위해서는 동일한 중심과 반경, 서로 평행한 법선벡터를 가져야 한다. (설명을 간단하게 하기 위해서, 법선벡터가 정반대

방향인 경우도 평행한 경우에 속하는 것으로 간주한다.)  $PF(T_1)$ 에 속하는 모든 원의 법선벡터는  $(0, 0, 1)$ 이고,  $PF(T_2)$ 에 속하는 모든 원의 법선벡터는  $N_2$ 이다. 따라서  $N_2$ 는  $z$ 축과 평행해야 하며, 이 때  $C_1$ 과  $C_2$ 가 동일한 중심을 갖기 위해서는  $p_2$ 가  $z$ 축 위에 존재해야만 한다.

토러스  $T_2$ 의 법선벡터  $N_2$ 가  $z$ 축과 평행하고 중심점  $p_2$ 가  $z$ 축 위에 있을 때,  $P$ 가  $xz$ -평면이라 하자. 그러면, 평면  $P$ 는 토러스  $T_1$ 과  $T_2$ 를 각각  $z$ -축에 대칭인 두 개의 원에서 자른다. 좌우 대칭이므로  $z$ -축의 오른쪽만 고려하면 된다. 이  $xz$ -절단면의 우측에 나타난 원들의 교점을 이용하여 우리가 구하고자 하는 profile circle의 교차원들을 모두 구할 수 있다. 그림 10은 평면  $P$ 로  $T_1, T_2$ 를 자른 단면을 보이고, 그림 11은 이러한 과정을 통하여 구해진  $PF(T_1) \cap PF(T_2)$ 의 예를 보인다.

표 2는 위의 과정을 알고리즘으로 제시한다.

#### 3.2 $PF(T_1) \cap CS(T_2)$

이 절은  $T_1$ 의 profile circle과  $T_2$ 의 cross-sectional circle이 일치하여 교차원을 이루는 경우를 설명한다.  $PF(T_1)$ 의 중심점의 궤적은 선분  $l = \{(0, 0, z_0) \mid -r_1 \leq z_0 \leq r_1\}$ 이고,  $PF(T_1)$ 의 법선벡터는  $z$ 축과 평행하다.  $CS(T_2)$ 의 경우 중심점  $q$ 의 궤적은  $T_2$ 의 주원(main circle)과 같고,  $q$ 를 중심으로 하는 cross-sectional

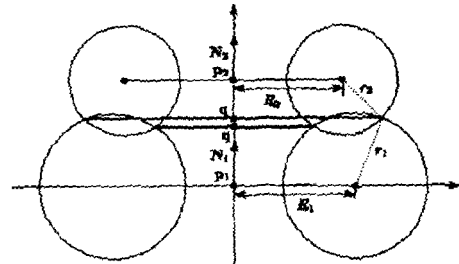


Fig. 10. Computation of  $PF(T_1) \cap PF(T_2)$ .

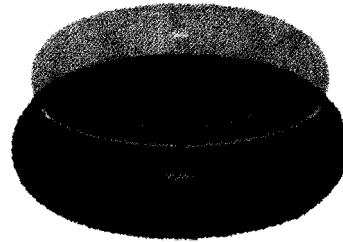


Fig. 11. Circles in  $PF(T_1) \cap PF(T_2)$

Table 1. Notation

$T_1 = T_{r_1, R_1}(p_1, N_1)$	minor 반경 $r_1$ , major 반경 $R_1$ , 중심점 $p_1$ , 법선벡터 $N_1$ 인 토러스
$T_2 = T_{r_2, R_2}(p_2, N_2)$	minor 반경 $r_2$ , major 반경 $R_2$ , 중심점 $p_2$ , 법선벡터 $N_2$ 인 토러스

Table 2. Algorithm for computing  $PF(T_1) \cap PF(T_2)$

```

Algorithm: Profile_Profile_in_TTI (T1, T2)
Begin
If N2 is parallel to the z-axis then
  If p2 is located on the z-axis then
    Begin
      P = {(x, y, z) | x > 0 and y = 0};
      C1 = T1 ∩ P;
      C2 = T2 ∩ P;
      Q = C1 ∩ C2;
      For each q=(qx, qy, qz) ∈ Q do
        Output Circle with center (0,0,qz),
          radius (qx2 + qy2)1/2,
          and normal vector N1;
      End
    End
  End
End
    
```

circle의 법선벡터는  $N_2 \times (q - p_2)$ 이다.  
 $PF(T_1)$ 과  $CS(T_2)$  간의 교집합이 존재할 필요충분 조건은 다음과 같다.

1.  $T_2$ 의 주원(main circle)이 선분  $l$ 과 교차한다.
2. 조건 1에서 구한 교점을  $q$ 라 할 때,  $N_2 \times (q - p_2)$ 가  $z$ 축과 평행하다.
3. 점  $q$ 를 중심으로 하는  $T_1$ 의 profile circle의 반경이  $T_2$ 의 minor 반경  $r_2$ 와 같다.

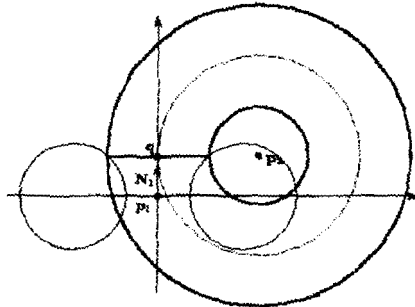


Fig. 12. Computation of  $PF(T_1) \cap CS(T_2)$ .

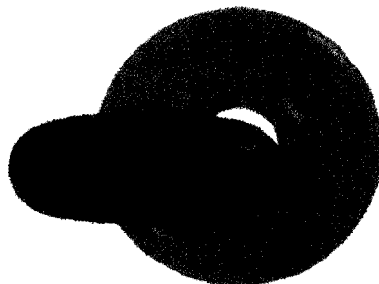


Fig. 13. Circle in  $PF(T_1) \cap CS(T_2)$ .

Table 3. Algorithm for Computing  $PF(T_1) \cap CS(T_2)$

```

Algorithm: Profile_CrossSection_in_TTI (T1, T2)
Begin
C = main circle of T2;
l = {(0, 0, z0) | -r1 ≤ z0 ≤ r1};
For each q=(qx, qy, qz) ∈ C ∩ l do
  Begin
    If N2 × (q - p2) // N1 then
      If R1 ± (r12 - qx2 - qy2)1/2 = r2 then
        Output Circle with center q, radius r2,
          and normal vector N1;
      End
    End
  End
End
    
```

그림 12는 위 세 가지 조건들을 모두 만족하는 경우의 예를 토러스의 단면으로 보이며, 그림 13은 profile circle과 cross-sectional circle이 일치하여 교차원을 이루는 예를 보인다. 이 그림에서 교차원은 2개의 교차곡선 중 위쪽에 있는 밝은 색의 곡선이다. 아래쪽의 어두운 색의 교차곡선은 6차 곡선이다. 표 3은 위의 조건들을 검증하는 과정을 알고리즘으로 제시한다.

### 3.3 $PF(T_1) \cap YV(T_2)$

이 절에서는  $T_1$ 의 profile circle과  $T_2$ 의 Yvone-Villarceau circle 간의 교집합의 발견에 대해 설명한다.  $PF(T_1)$ 의 중심점의 제적은 선분  $l = \{(0, 0, z_0) | -r_1 \leq z_0 \leq r_1\}$ 이고,  $PF(T_1)$ 의 법선벡터는  $z$ 축과 평행하다. 제 2.3절에서 설명된 바와 같이  $YV(T_2)$ 의 중심점의 제적을  $C^{YV}$ 라 하면,  $C^{YV}$ 는 중점  $p_2$ , 법선벡터  $N_2$ , 반경  $r_2$ 인 원이다. 토러스  $T_2$ 상의 모든 Yvone-Villarceau circle의 반경은  $R_2$ 이며,  $T_2$ 상의 임의의 Yvone-Villarceau circle의 중점이  $q$ 일 때, 법선벡터는  $(\pm(q - p_2) \times N_2 + (R_2^2 - r_2^2)^{1/2} N_2) / R_2$ 이다<sup>[5]</sup>.

$PF(T_1)$ 과  $YV(T_2)$  간의 교집합이 존재하기 위한 필요충분조건은 다음과 같다.

1.  $l$ 과  $C^{YV}$ 가 교차한다.
2. 조건 1에서 구한 교점을  $q$ 라고 할 때  $q$ 를 중심으로 하는  $T_1$ 의 profile circle 반경이  $R_2$ 이다.
3. 점  $q$ 를 중심으로 하는  $T_2$ 의 Yvone-Villarceau circle의 법선벡터가  $z$ 축 방향과 평행하다.

그림 14는 이 조건이 만족되는 경우의 토러스  $T_1$ ,  $T_2$ 를  $xy$ -평면에 투영하여 보이며, 이 그림에서 굵은 선으로 표시된 원이 교차원이다. 그림 15는 두 개의 토러스에서 profile circle과 Yvone-Villarceau circle이 일치하여 생기는 교차원을 갖는 경우를 보여준다.

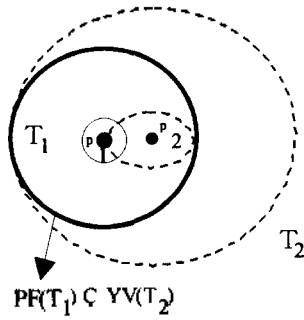


Fig. 14. Computation of  $PF(T_1) \cap YV(T_2)$ .

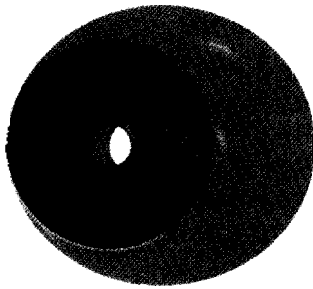


Fig. 15. Circle in  $PF(T_1) \cap YV(T_2)$ .

Table 4. Algorithm for computing  $PF(T_1) \cap YV(T_2)$

```

Algorithm: Profile_YvonneVillarceau_in_TTI (T1, T2)
Begin
  CYV = Circle with center p2, normal vector N2,
    and radius r2;
  l = {(0, 0, z0) | -r1 ≤ z0 ≤ r1};
  For each q = (qx, qy, qz) ∈ CYV ∩ l do
  Begin
    If R1 ± (r12 - qz2)1/2 = R2 then
      If (±(q - p2) × N2 + (R22 - r22)1/2 N2) // N1 then
        Output Circle with center q, radius R2,
          and normal vector N1;
  End
End
    
```

이 그림에는 밝은 색과 어두운 색의 두 개의 교차곡선이 있는데, 교차원은 밝은 색의 곡선으로서 토러스 T<sub>2</sub>를 바깥쪽에서 둘러싸고 있다. 어두운 색의 교차곡선은 6차 곡선이다. 표 4는 위의 조건에 따라 교차원을 구하는 알고리즘을 보인다.

3.4 CS(T<sub>1</sub>) ∩ CS(T<sub>2</sub>)

이 절에서는 T<sub>1</sub>의 cross-sectional circle과 T<sub>2</sub>의 cross-sectional circle 간의 교집합의 발견에 대해 설명한다. CS(T<sub>1</sub>)에 속하는 원의 반경은 r<sub>1</sub>이며 중심점의 궤적은

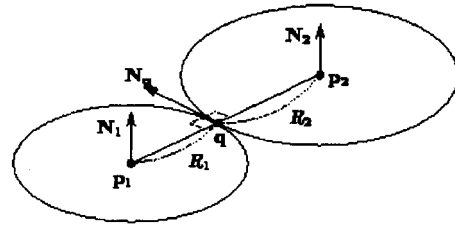


Fig. 16. Computation of  $CS(T_1) \cap CS(T_2)$ .

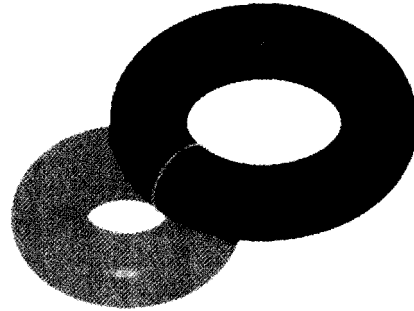


Fig. 17. Circle in  $CS(T_1) \cap CS(T_2)$ .

T<sub>1</sub>의 주원(main circle)과 같다. CS(T<sub>2</sub>)에 속하는 원의 반경은 r<sub>2</sub>이며, 중심점의 궤적은 T<sub>2</sub>의 주원과 같다.

CS(T<sub>1</sub>)과 CS(T<sub>2</sub>) 간의 교집합이 존재하기 위한 필요충분조건은 다음과 같다.

1. r<sub>1</sub> = r<sub>2</sub>
2. T<sub>1</sub>의 주원과 T<sub>2</sub>의 주원이 교차한다.
3. 조건 2에서 구한 교점이 q일 때, N<sub>1</sub> × (q - p<sub>1</sub>)와 N<sub>2</sub> × (q - p<sub>2</sub>)가 평행이다.

그림 16은 이 조건이 만족되는 경우의 주원들의

Table 5. Algorithm for Computing  $CS(T_1) \cap CS(T_2)$

```

Algorithm:
CrossSection_CrossSection_in_TTI (T1, T2)
Begin
  C1 = main circle of T1;
  C2 = main circle of T2;
  If r1 = r2 then
  Begin
    For each q ∈ C1 ∩ C2 do
    Begin
      If (q - p1) × N1 // (q - p2) × N2 then
        Output Circle with center q, radius r2,
          and normal vector (q - p1) × N1;
    End
  End
End
    
```

상호관계를 보인다. 그림 17은 이 조건이 만족되어 교차원이 생기는 예를 제시하는데, 그림에서 밝은 색과 어두운 색의 두 개의 교차곡선을 발견할 수 있다. 이중 밝은 색의 교차곡선이 교차원이며, 어두운 색의 교차곡선은 4차 곡선이다. 표 5는 위의 조건을 만족하는 원을 계산하는 알고리즘을 보인다.

3.5  $CS(T_1) \cap YV(T_2)$

이 절에서는  $T_1$ 의 cross-sectional circle과  $T_2$ 의 Yvone-Villarceau circle이 일치하는 경우를 발견하는 과정을 설명한다.  $CS(T_1)$ 에 속하는 원의 반경은  $r_1$ 이며, 중심점의 궤적은  $T_1$ 의 주원과 같다.  $YV(T_2)$ 의 중심점의 궤적은  $p_2$ 를 중심으로 하고 법선벡터  $N_2$ , 반경이  $r_2$ 인 원이며, 이 원을  $C_2^{YV}$ 라고 하자.  $CS(T_1)$ 과  $YV(T_2)$  간의 교집합이 존재하기 위한 필요충분 조건은 다음과 같다.

1.  $r_1 = R_2$
2.  $T_1$ 의 주원과  $C_2^{YV}$ 가 교차한다.
3. 조건 2에서 구한 교점이  $q$ 라고 할 때,  $q$ 를 중심으로 하는  $T_1$ 의 Yvone-Villarceau circle의 법선벡터가  $N_1 \times (q - p_1)$ 와 평행이다.

그림 18(a)는 이 조건을 만족하는 두 토러스의 xz-단면도를 보이고, 그림 18(b)는 동일한 예를 토러스

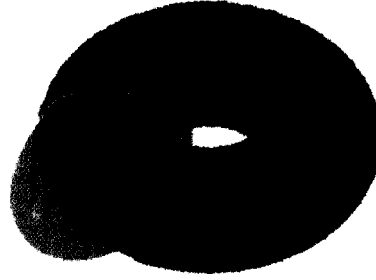


Fig. 19. Circle in  $CS(T_1) \cap YV(T_2)$ .

Table 6. Algorithm for Computing  $CS(T_1) \cap YV(T_2)$

<p>Algorithm:                  CrossSection_YvoneVillarceau_in_TT1 (<math>T_1, T_2</math>)                  Begin  <math>C_1 =</math> main circle of <math>T_1</math>;  <math>C_2^{YV} =</math> Circle with center <math>p_2</math>, radius <math>r_2</math>,                  and normal vector <math>N_2</math>;                  If <math>r_1 = R_2</math> then                  For each <math>q \in C_1 \cap C_2^{YV}</math> do                  Begin                  If <math>(q - p_1) \times N_1</math>                  // <math>(\pm (q - p_2) \times N_2 + (R_2^2 - r_2^2)^{1/2} N_2)</math> then                  Output Circle with center <math>q</math>, radius <math>r_1</math>,                  and normal vector <math>(q - p_1) \times N_1</math>;                  End                  End</p>
--

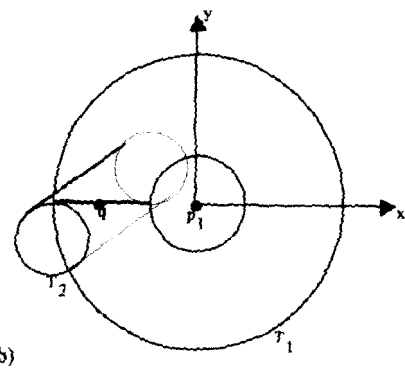
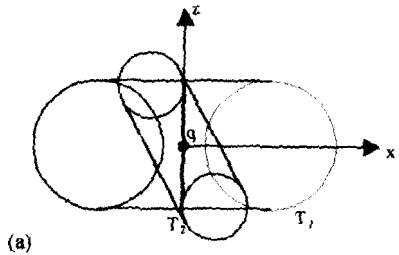


Fig. 18. Construction of  $CS(T_1) \cap YV(T_2)$ .

$T_1$ 의 주평면인  $xy$ -평면에 투영하여 보인다. 각각의 그림에서 굵은 선분은 투영된 교차원을 나타낸다. 그림 19는 실제 두 개의 토러스가 교차원을 갖는 예를 보여주는데, 그림에서 밝은 색과 어두운 색의 두 개의 교차곡선을 발견할 수 있다. 밝은 색의 교차곡선이 교차원이며, 어두운 색의 교차곡선은 6차곡선이다. 표 6은 위의 조건을 만족하는 교차원을 구하는 알고리즘을 제시한다.

3.6  $YV(T_1) \cap YV(T_2)$

이 절에서는  $T_1$ 의 Yvone-Villarceau circle과  $T_2$ 의 Yvone-Villarceau circle이 일치하는 경우의 발견에 대해 설명한다.  $YV(T_1)$ 에 속한 원들의 중심점이 이루는 궤적을  $C_1^{YV}$ 이라 하면,  $C_1^{YV}$ 은 중점  $p_1$ , 법선벡터  $N_1$ , 그리고 반경이  $r_1$ 인 원이다. 마찬가지로,  $YV(T_2)$ 에 속한 원들의 중심점이 이루는 궤적을  $C_2^{YV}$ 라 하면,  $C_2^{YV}$ 은 중점  $p_2$ , 법선 벡터  $N_2$ , 그리고 반경이  $r_2$ 인 원이다.

$YV(T_1)$ 과  $YV(T_2)$  사이에 공유되는 원이 존재하기 위한 필요충분조건은 다음과 같다.



1.  $R_1 = R_2$ .
2.  $C_1^{YV}$ 과  $C_2^{YV}$ 가 교차한다.
3. 조건 2에서 구한 교점이  $q$ 라고 할 때,  $q$ 를 중심으로 하는  $T_1$ 의 Yvone-Villarceau circle과  $T_2$ 의 Yvone-Villarceau circle의 법선벡터가 평행이다.

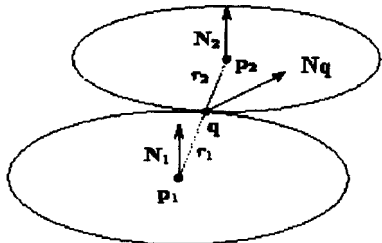


Fig. 20. Computation of  $YV(T_1) \cap YV(T_2)$ .

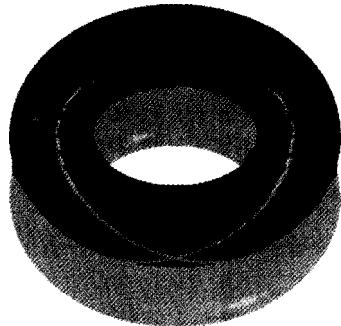


Fig. 21. Circles in  $YV(T_1) \cap YV(T_2)$ .

Table 7. Algorithm for Computing  $YV(T_1) \cap YV(T_2)$

```

Algorithm:
YvoneVillarceau_YvoneVillarceau_in_TTI (T1, T2)
Begin
  C1YV = Circle with center p1, normal vector N1,
  and radius r1;
  C2YV = Circle with center p2, normal vector N2,
  and radius r2;
  If R1 = R2 then
    For each q ∈ C1YV ∩ C2YV do begin
      If ((q - p1) × N1 + (R12 - r12)1/2 N1)
      // (±(q - p2) × N2 + (R22 - r22)1/2 N2) then
        Output Circle with center q,
        normal vector (q-p1)×N1+ (R12-r12)1/2 N1,
        and radius R2;
      If (-(q - p1) × N1 + (R12 - r12)1/2 N1)
      // (±(q - p2) × N2 + (R22 - r22)1/2 N2) then
        Output Circle with center q,
        normal vector -(q-p1)×N1+ (R12-r12)1/2 N1,
        and radius R2;
    End
  End
End
    
```

그림 20은 이 조건이 만족되는 경우에 있어서  $C_1^{YV}$ 과  $C_2^{YV}$  사이의 상호관계를 보여준다. 그림 21은 실제 두 토러스가 교차원을 가지는 경우의 예를 제시하는데, 이 그림에서는 밝은 색과 어두운 색의 교차곡선이 각각 두 개씩 존재한다. 두 개의 밝은 색의 교차곡선은 교차원들이며, 두 개의 어두운 색의 교차곡선은 하나의 4차 평면곡선이다. 표 7은 위의 조건을 만족하는 교차원을 구하는 알고리즘을 제시한다.

#### 4. 두 토러스 사이의 교차곡선 계산

이 절에서는 두 토러스 사이의 교차곡선을 구하는 전체 알고리즘 안에서 교차원을 발견하고 이를 구체적으로 계산하는 각각의 부분적인 알고리즘을 사용하는 방법을 표 8에서 간단한 의사코드를 이용하여 설명한다. 전체 교차곡선을 계산하기 위해서는 우선 교차곡선에 존재하는 모든 2차 곡선, 즉 원을 발견하고 계산한 뒤, 원 이외의 곡선이 교차곡선에 포함되어 있으면 이들을 이전의 연구결과<sup>15)</sup>에서 개발된 방법을 이용하여 마저 계산한다.

#### 5. 결 론

본 논문에서는 기하학적 접근방법을 이용하여 두 토러스 간의 교차곡선 상에 존재하는 2차 곡선을 발견하는 방법과 발견된 2차 곡선을 구체적으로 계산하는 알고리즘을 제시하였다. 토러스 곡면 상에 존재하는 유일한 2차 곡선은 원이며 이들은 profile circle, cross-sectional circle, Yvone-Villarceau circle의 세

Table 8. Algorithm for computing intersection curves between two tori

```

Algorithm: Torus_Torus_Intersection (T1, T2)
Begin
  Profile_Profile_in_TTI (T1, T2);
  Profile_CrossSection_in_TTI (T1, T2);
  Profile_CrossSection_in_TTI (T2, T1);
  Profile_YvoneVillarceau_in_TTI (T1, T2);
  Profile_YvoneVillarceau_in_TTI (T2, T1);
  CrossSection_CrossSection_in_TTI (T1, T2);
  CrossSection_YvoneVillarceau_in_TTI (T1, T2);
  CrossSection_YvoneVillarceau_in_TTI (T2, T1);
  YvoneVillarceau_YvoneVillarceau_in_TTI(T1, T2);
  If T1 ∩ T2 contains some curves other than conic
  sections then
    Compute_non_ConicSection (T1, T2);
  End
End
    
```

가지로 분류된다. 세 종류의 원이 갖는 기하학적 정보를 바탕으로 두 개의 토러스 사이의 교차곡선에 존재하는 모든 원을 발견하고 이를 구체적으로 계산하는 알고리즘을 제시하였다.

교차원이 존재하기 위한 조건을 따지는 과정에서는 원과 선분의 교차점 계산, 그리고 원과 원의 교차점의 계산, 그리고 벡터 연산(vector operation) 등과 같은 비교적 간단한 계산들만이 이용되었다. 따라서, 본 논문에서 제시된 알고리즘은 대단히 효율적이고 수치계산에 수반되는 오차가 적다. Heo *et al.*<sup>12,13</sup>에서는 공간상에서 주어지는 반지름이 서로 다른 두 원의 교차점 계산을 세 평면 사이의 교차점을 계산하는 문제로 변환하는 기법을 소개하고 있다. 유틸리티 연산을 이용하면 세 평면의 교점을 오차가 없이 계산할 수 있으므로, 두 원의 교점을 아주 정밀하게 계산할 수 있다. 공간상에서 주어진 선분과 원의 교점 또한 비슷한 방법으로 정밀하게 계산할 수 있다.

### 감사의 글

본 연구는 과학기술부 국가지정연구실 과제에 의하여 지원되었고, 정보통신부 대학S/W연구센터 지원사업과 정보통신부 대학기초연구 지원사업에 의해 지원되었습니다. 논문을 위한 도움 말씀을 주신 오스트리아 비엔나공과대학 기하학연구소 Helmut Pottmann 교수님과 공동연구에 참여해 주신 미국 미주리대학교 수학과 오경호 교수님께 감사 드립니다. 그리고, 본 논문의 수정을 위하여 많은 조언을 해주신 심사위원 두 분께도 깊이 감사 드립니다.

### 참고문헌

1. Farouki, R., Neff, C. and O'Connor, M., "Automatic parsing of degenerate quadric-surface intersections", *ACM Trans. on Graphics*, Vol. 8, No. 3, pp. 174-203, 1989.
2. Heo, H.-S., *The Intersection of Ruled and Ringed Surfaces*, Ph.D. Thesis, Dept. of Computer Science, POSTECH, February, 2000.
3. Heo, H.-S., Hong, S.J., Kim, M.-S. and Elber, G., "The Intersection of Two Ringed Surfaces", *Proc. of Pacific Graphics 2000*, Hong Kong, October 3-5, 2000, pp. 146-153.
4. Johnstone, J. and Shene, C.-K., "Computing the intersection of a plane and a natural quadric", *Computers and Graphics*, Vol. 16, No. 2, pp. 179-186, 1992.
5. Kim, K.-J., Kim, M.-S. and K. Oh., "Torus/Sphere Intersection Based on a Configuration Space Approach", *Graphical Models and Image Processing*, Vol. 60, No. 1, pp. 77-92, January, 1998.
6. Kim, K.-J., *Torus and Simple Surface Intersection Based on a Configuration Space Approach*, Ph.D. Thesis, Dept. of Computer Science and Engineering, POSTECH, February, 1998.
7. Kim, K.-J. and Kim, M.-S., "Computing All Conic Sections in Torus and Natural Quadric Intersections", *Proc. of Israel-Korea Bi-National Conference on New Themes in Computerized Geometrical Modeling*, Tel-Aviv, Israel, February 18-19, 1998, pp. 11-20.
8. Levin, J., "A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces", *Commun. ACM*, Vol. 19, No. 10, pp. 555-563, 1976.
9. Levin, J., "Mathematical models for determining the intersections of quadric surfaces", *Comput. Graphics Image Process*, Vol. 11, pp. 73-87, 1979.
10. Miller, J., "Geometric approaches to nonplanar quadric surface intersection curves", *ACM Trans. on Graphics*, Vol. 6, No. 4, pp. 274-307, 1987.
11. Miller, J. and Goldman, R., "Using Tangent Balls to Find Plane Sections of Natural Quadrics", *IEEE Computer Graphics and Applications*, pp. 68-82, March, 1992.
12. Miller, J. and Goldman, R., "Geometric Algorithms for Detecting and Calculating All Conic Sections in the Intersection of Any Two Natural Quadric Surfaces", *Graphical Models and Image Processing*, Vol. 57, No. 1, pp. 55-66, 1995.
13. Piegl, L., "Geometric method of intersecting natural quadrics represented in trimmed surface form", *Computer Aided Design*, Vol. 21, No. 4, pp. 201-212, 1989.
14. Pottmann, H., Private Communications.
15. Sarraga, R., "Algebraic methods for intersections of quadric surfaces in GMSOLID", *Comput. Vision Graphics Image Process*, Vol. 22, No. 2, pp. 222-238, 1983.
16. Shene, C.-K. and Johnstone, J., "On the Lower Degree Intersections of Two Natural Quadrics", *ACM Trans. on Graphics*, Vol. 13, No. 4, pp. 400-424, 1994.
17. Wilf, I. and Manor, Y., "Quadric-surface intersection curves: Shape and structure", *Computer Aided Design*, Vol. 25, No. 10, pp. 633-643, 1993.



### 김 구 진

1990년 이화여자대학교 전산학 학사  
 1992년 한국과학기술원 전산학 석사  
 1998년 포항공과대학교 컴퓨터공학 박사  
 1998.3~1998.11 포항공과대학교 컴퓨터공학과 박사후 연수과정  
 1998.12~2000.5 Purdue University 전산학과 박사후 연수과정  
 2000.6~2000.8 서울대학교 컴퓨터공학부 방문연구원  
 2000.9~현재 아주대학교 정보통신전문대학원 조교수 대우



### 김 명 수

1980 서울대학교 수학교육학 학사  
 1982 서울대학교 수학 석사  
 1985 Purdue University 응용수학 석사  
 1987 Purdue University 전산학 석사  
 1988 Purdue University 전산학 박사  
 1988~1998 : 포항공과대학교 전산학과 조교수, 부교수  
 1995 : Purdue University 전산학과 객원부교수  
 1999~현재 : 서울대학교 컴퓨터공학부 부교수  
 국제학술지 편집위원 : Computer-Aided Design, Computer Graphics Forum, Int'l J. of Shape Modeling