

## 구 볼록 다각형들의 최대 교차를 찾기 위한 효율적인 구 분할 방식

하 종 성\*

### Efficient Sphere Partition Method for Finding the Maximum Intersection of Spherical Convex Polygons

Ha, J. S.\*

#### ABSTRACT

The maximum intersection of spherical convex polygons are to find spherical regions owned by the maximum number of the polygons, which is applicable for determining the feasibility in manufacturing problems such mould design and numerical controlled machining. In this paper, an efficient method for partitioning a sphere with the polygons into faces is presented for the maximum intersection. The maximum intersection is determined by examining the ownerships of partitioned faces, which represent how many polygons contain the faces. We take the approach of edge-based partition, in which, rather than the ownerships of faces, those of their edges are manipulated as the sphere is partitioned incrementally by each of the polygons. Finally, gathering the split edges with the maximum number of ownerships as the form of discrete data, we approximately obtain the centroids of all solution faces without constructing their boundaries. Our approach is analyzed to have an efficient time complexity  $O(nv)$ , where  $n$  and  $v$ , respectively, are the numbers of polygons and all vertices. Furthermore, it is practical from the view of implementation since it can compute numerical values robustly and deal with all degenerate cases.

**Key words :** Visibility map, Maximum intersection, Mould design, NC machining, Spherical algorithm, Computational geometry

#### 1. 서 론

구 볼록 다각형들의 최대 교차를 찾는 문제는 컴퓨터그래픽스나 CAD/CAM 분야에서 잠재적으로 많은 응용 가치를 가지고 있다. 예를 들어 거푸집(mould)의 설계<sup>12)</sup>나 수치제어 절삭가공<sup>11,10,17)</sup>의 두 제조 방법에서도 설계된 모형의 최적의 방위(orientation)를 결정하는 문제는 그 모형의 표면(surface)의 법선벡터들로부터 유도된 가시맵(visibility map)들의 최대 교차를 찾는 문제로 변형되었다. 여기서 가시맵은 삼차원 좌표계  $E^3$ 의 원점에 중심이 위치한 단위구  $S^2 = \{p = \|p\| = 1, p = (x, y, z) \in E^3\}$ 상의 볼록한 개체이다.

가시맵과 관련된 구 볼록 다각형들의 교차를 계산하는

방법으로 크게 조각화(tessellation)<sup>16,18)</sup>와 분할(partition)<sup>11,10,17)</sup>의 두 종류로 나눌 수 있다. 조각화 방법은  $S^2$ 를 작은 삼각형 조각 집합으로 나누어 각 다각형들이 소유하는 영역을 계산하는 디지털 맵 지향 방식이다. 분할 방법은  $S^2$ 를 다각형들의 경계로 분할하여 면(face)의 집합으로 나타내고 얼마나 많은 다각형이 그 면을 포함하고 있는지 나타내는 각 면의 소유권(ownership)을 직접 계산하는 다각형 지향 방식이다. 조각화 방법은 해를 견고하게 계산할 수 있으므로 응용 분야에서 실제적으로 사용된다. 반면에 분할 방법은 모든 해의 경계를 찾는 효율적인 시간 복잡도를 가지나 수치적인 복잡성을 내포하고 있어 아직 실제적인 구현이 보고되지 않고 있다.

본 논문에서는 수치적 복잡성이 없이 견고하게 구현할 수 있는 실제적인 분할 방법을 제시한다. 분할된 면들의 소유권을 계산하는 방법을 면 기반 분할(face-based partition)이라고 하고 우리의 방법은 분할된 간선(edge)

\*정회원, 우석대학교 정보통신컴퓨터공학부  
- 논문투고일: 2000. 12. 12  
- 심사완료일: 2001. 3. 9

들의 소유권을 유지하므로 간선 기반 분할(edge-based partition)이라고 한다. 면 기반 분할은 면들을 다루기 위하여 경계표현(boundary representation)<sup>20,22</sup>과 같은 자료구조를 사용하는 반면에 간선 기반 분할은 간선들을 다루기 위하여 단방향 연결 순환 리스트<sup>7</sup>를 사용한다.

간선 기반 분할의 가장 큰 실제적인 장점은 수치 계산의 한계로 말미암아 근접의 교차점들에 발생할 수 있는 위상적 위배(topological violation)를 걱정할 필요 없이 견고하게 구현될 수 있다는 것이다. 이것은 해로서 이산적이 아닌 연결된 자료 형태로 간선을 얻을 수 있는 양방향 연결 순환 리스트를 사용하지 않은 이유이기도 하다. 다른 장점은 하나의 점 또는 간선만으로 된 가시맵과 같이 degeneracy 경우도 자연스럽게 해결할 수 있다는 것이다. 이론적인 면에서도 우리의 방법은 해의 경계를 구하는  $O(m \log n)$ 시간의 면 기반 분할보다 효율적인  $O(m)$ 의 시간 복잡도에 해의 중심을 구한다. 여기서  $n$ 과  $m$ 은 각각 구와 모든 정점의 개수이다.

2절에서는 가시맵의 교차와 관련있는 제조성 문제들과 이를 해결하기 위한 이전 연구들에서 구현의 한계를 살펴본다. 우리의 연구는 이러한 한계들을 모두 극복하기 위한 것으로 구상의 알고리즘들을 처음부터 구현하는 방법을 제시한다. 3절에서는 각 표면의 가시맵을 그 표면의 법선벡터들의 볼록형(convex hull)로부터 구성한다. 따라서 2차원 볼록형 알고리즘을 구상에서 효율적으로 구현할 수 있으며 모든 degeneracy 경우도 처리한다.

알고리즘의 주요 부분은 단방향 순환 리스트를 사용하여 분할되는 간선들의 소유권을 유지하는 4절의 간선 기반 분할이다. 한 쌍의 구 볼록 다각형을 교차시키기 위해 제시된 효율적인 알고리즘<sup>25</sup>을 바탕으로 다각형들의 모든 쌍을 점중적으로 교차시켜나가면서 분할되는 간선의 소유권을 갱신하여 나간다. 마지막에는 해의 영역중 하나의 중심을 근사적으로 얻어낸다. 5절에서는 이 연구의 이론적인 면과 실제적인 면에 대하여 토론한다.

## 2. 가시맵과 그 응용

### 2.1 가시맵의 정의

두 점을 연결하는 선분 사이에 아무런 방해가 없으면 그 두 점은 서로 가시적(visible)이라고 한다. 표면  $S$ 의 가시맵은 무한한 곳으로부터  $S$ 상의 모든 점이 가시적인 모든 방향을  $S^2$ 상에 나타낸 볼록한 영역이다.  $S$ 의 법선벡터들을  $S^2$ 상에 사상시킨 것은 가우시안맵이라 한다.

표면  $S$ 의 가우시안맵  $P = \{p_1, \dots, p_m\}$ 에 대하여  $CH(P)$ 와  $VM(P)$ 로 각각  $P$ 의 볼록형과 가시맵을 나타내자.  $HS(p_i)$ 은 반구  $\{p | p \cdot p_i \geq 0\}$ 를 나타내고  $GC(p_i)$ 은  $HS(p_i)$ 를 경계짓는 대원(great circle)을 나타낸다. 여기서  $p_i$ 를 극(pole)이라 한다. 그러면  $S$ 의 (양)가시맵  $VM(P)$ 는  $\bigcap_{i=1}^m HS(p_i)$ 로 나타내어지며 음가시맵  $-VM(P)$ 는  $\bigcap_{i=1}^m H(-p_i)$ 로 나타내어진다.

$CH(P)$ 와  $VM(P)$ 를 극점의 순서를 반시계방향으로 나타낸다고 가정하면  $CH(P)$ 와  $VM(P)$ 의 이원성(duality)<sup>8</sup>에 의하여  $VM(P)$ 의 정점들은  $CH(P)$ 의 간선을 지나는 대원의 극들과 일치하며 그 순서도 같다.

### 2.2 관련 응용

표면들의 가시맵을 사용하여 많은 제조성 문제들을 형식화할 수 있다. 설계된 다면체 모형의 거푸집의 자동적인 설계를 고려해보자. 거푸집이 분리되는데 방해가 되는 부분들을 처리하기 위하여 필요한 보조 장치의 개수를 최소화하는 모형의 방위를 찾는 것은 중요한 일이다.

어느 간선을 공유하는 두 면의 외부 각이 180 도보다 작으면 그 간선이 오목(concave)하다고 한다. 모형의 방위 문제는 오목 에지를 공유함으로써 연결된 면의 집합과 밀접한 관련이 있다. 이러한 면의 집합을 오목 연결 표면(concave-connected surface)<sup>26</sup>이라 한다. Fig. 1에서 보는 바와 같이 이전 연구<sup>12</sup>에서 사용된 하나의 포켓 표면은 연결된 볼록 간선들에 의하여 몇 개의 오목 연결 표면(concave-connected surface)으로 나뉘어질 수 있다. 모형의 최적 방위는 이 오목 연결 표면들의 양가시맵과 음가시맵의 최대 교차를 찾음으로써 결정될 수 있다.

이제 수치제어 절삭 가공에서는 설계된 다면체 모형을 한번 고정하여 절삭할 수 있는 포켓 표면의 개수를 최대화하는 방위 문제를 고려해야한다<sup>15,26</sup>. 3축 공구

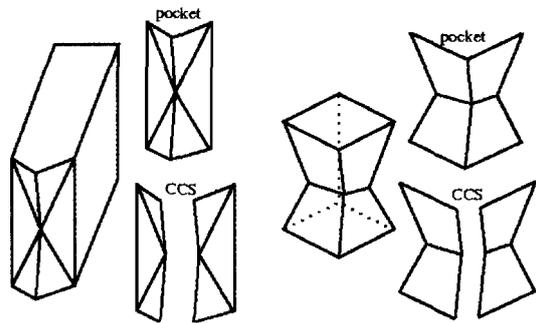


Fig. 1. Concave-connected surface and pocket surface.

의 경우에 최적의 방위는 포켓 표면의 양가시선의 최대 교차를 찾음으로써 결정할 수 있다. 다른 다축 공구에 대해서는 Chen 등<sup>[11]</sup>이 대부분의 최적 방위 문제를 가시맵으로부터 유도된 구 볼록 다각형의 최대 교차를 찾는 문제로 변형하였다. 그들은 다축 수치제어 절삭 가공의 최적 방위와 관련하여 5가지 구상의 기하적 문제를 제시하였다. 그런데 그중 가우시안맵의 densest hemisphere, separator, bisector, minimal intersection을 구하는 4가지는 구 볼록 다각형의 최대 교차 문제로, 가우시안맵의 maximal intersection을 구하는 1가지는 최소 교차 문제로 각각 해결된다. 5가지 문제의 자세한 내용은 문헌<sup>[11]</sup>을 참고하면 된다.

2.3 가시맵의 교차

가시맵으로부터 유도된 구 볼록 다각형들을 교차시키기 위하여 현재 사용되고 있는 실제적인 방법은 작은 삼각형 조각 집합으로 나누어 각 조각을 어느 다각형들이 포함하는가를 검사하는 것이다. 이 조각화 방식은 해를 견고하게 계산할 수 있지만 조각화의 정도에 따라 정확성과 효율성이 서로 반대로 달라지므로 두 측정값은 서로 교환 관계가 있다.

다른 방법은 다각형들의 경계로  $S^2$ 를 면의 집합으로 분할하여 면의 소유권을 검사하는 분할 방식이다. 가장 일반적인 분할 방법은 Chen 등<sup>[11]</sup>의 것으로서 다각형으로 분할되는 면들을 경계 표현으로 나타내고 각 면의 소유권을 완전하게 계산한다. 이 알고리즘에서는 한 면을 어느 가시맵들이 포함하는가 나타내기 위한 소유권 벡터를 각 면에 유지시킨다. 어느 면의 소유권 벡터의  $n$ 번째 비트는 그 면이  $n$ 번째 양가시맵에 포함되면 +1로,  $n$ 번째 음가시맵에 포함되면 -1로, 그렇지 않으면 0으로 세팅된다. 따라서 각 면의 소유권 벡터에서 +과 -1의 개수를 세면 최대 교차뿐만 아니라 다각형의 교차에 관련된 모든 문제를 해결할 수 있다. 그러나 이 방법은 기하적인 측면에서는 우수하지만 중심부상을 사용하지 않으므로  $z=0$ 에 가까운 점에 대한 수치적 처리 때문에 구현할 때 수치적 문제가 발생한다.

위의 면 기반 분할 방식은 효율적인 시간에 모든 해의 경계를 구할 수 있지만 Suh 등<sup>[16]</sup>이 지적하였듯이 중점투상등과 같이 수치적 복잡성을 내포하고 있는 연산들이 있어 CAD/CAM 응용에서 실제로 구현되어 사용되고 있지는 않다. 또한 Fig. 2에서와 같이 수치 계산의 한계 때문에 발생할 수 있는 근접한 교차점의 위상 오류는 분할된 면을 경계표현<sup>[20,22]</sup>과 같은 자료구조를 무효화시키므로 이를 찾아내어 교정하는 추가적인 노력이 필요하다. 더욱이 이 오류를 교정하기가 간단

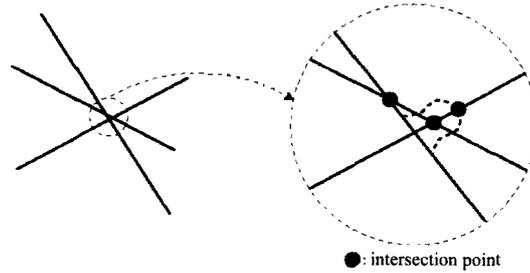


Fig. 2. Topological violation of intersection points.

하지 않아 보인다.

직교 부품을 포함하는 CAD/CAM 모형에서 자주 발생할 수 있는 하나의 점 또는 직선의 가시맵과 같은 degeneracy는 면 기반 분할 방식에서는 처리되기 어렵다. 조각화 방식에서도 이런 경우에는 해가 그 점 또는 직선상에 존재할 경우 오차가 큰 해를 얻어낼 수 있다.

본 논문에서는 면 기반 분할 방식이 가지고 있는 구현상의 제약점들은 극복하면서도 효율적으로 하나의 해를 얻어내는 간선 기반 분할 방식을 제안한다. 이 방식의 견고한 구현을 위하여  $S^2$ 상의 수치값을 안전하게 계산하는 방법으로 구상의 알고리즘들을 구성한다. 또한 찾아진 하나의 해는 실제적으로 많이 요구되고 있는 해의 중심에 가깝게 한다.

3. 가시맵의 구성

먼저 주어진 표면의 가시맵을 안전한 수치적 계산으로 구성하는 방법을 설명한다.  $VM(P)$ 의 구성은 몇 단계로 나누어지는데 첫째는  $P$ 가 어느 반구안에도 포함되지 않으면  $CH(P)$ 가 존재하지 않으므로  $P$ 의 반구성(hemisphericity)을 검사한다. 즉  $VM(P)$ 는  $P$ 가 어느 반구에 포함되어야만 존재한다. 둘째  $P$ 는 반구안에 포함되지만  $CH(P)$ 가 정의되지 않는 degeneracy를 검사하는 것이다. 셋째  $CH(P)$ 를 구성하여 점과 선분과 같은 degeneracy를 검사한다. 마지막 넷째로  $CH(P)$ 로부터  $CM(P)$ 과  $-VM(P)$ 를 얻는다.

3.1 반구성 검사

$P$ 를 포함하는 모든 반구의 극 집합은  $\bigcap_{i=1}^m HS(pi)$ 로 표현된다. 즉  $VM(P)$ 이다. 반구의 교차는 모든 점  $p_i(x_i, y_i, z_i) \in P$ 에 대하여  $x_i x + y_i y + z_i z \geq 0$ 인 선형 부등식 시스템으로 표현된다. 만약 이 시스템에 선형프로그래밍(Linear Programming, LP)을 적용시키면 LP 알고리즘들<sup>[4,11,12,14]</sup>은 항상 가능한 해의 영역중 극점을 찾게되므로 trivial 해 (0,0,0)을 얻을 수 있다.

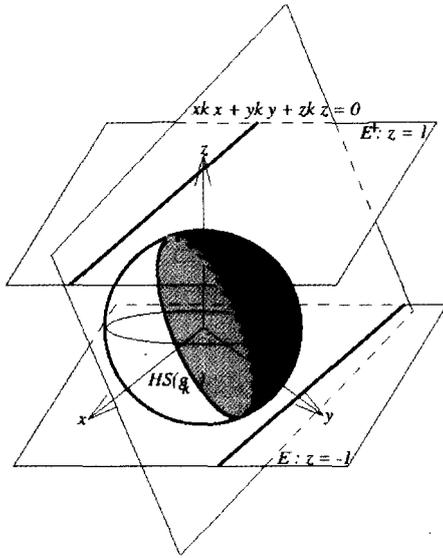


Fig. 3. Central projection of a great circle into two planes.

$P$ 의 반구성을 검사하는 문제는  $P$ 의 모든 점을  $z=1$ 인 평면에 중심투상(central projection)시켜  $z>0$ 인 점 집합과  $z<0$ 인 점 집합의 선형분리성(linear separability) 문제로 변형될 수 있다<sup>[3]</sup>. 그러나 변환된 이 문제는 LP 문제로 다시 변환하여야 하거나 복잡한 기하적 알고리즘을 사용하여 풀어야 한다.

여기에서는 위와 유사한 기하적 관찰을 통하여 중심투상의 개념을 이용하면서도 간단한 변환으로 완전한 해를 견고한 계산으로 구할 수 있는 두 개의 LP를 만든다. 반구의 경계인 대원  $GC(p_i)$ 를 Fig. 1과 같이  $E^*:z=1$ 인 평면과  $E^-:z=-1$ 인 평면에 중심투상시켜 보자. Fig. 3에서 보는 바와 같이  $z>0$ 인 대원의 일부는  $E^*$ 에서,  $z<0$ 인 대원의 일부는  $E^-$ 에서 각각 직선으로 투상된다. 즉, 반구  $HS(p_i)=x_i x + y_i y + z_i z \geq 0$ 는  $E^*$ 에서는  $x_i x + y_i y + z_i \geq 0$ 로  $E^-$ 에서는  $x_i x + y_i y - z_i \geq 0$ 로 각각 나타난다. 따라서  $\bigcap_{i=1}^m HS(p_i)$ 는 다음과 같은 두 LP로 표현할 수 있다. 기하적인 측면에서 비교한다면 점  $p_i$ 를 하나의 평면  $E^*$ 에 투상시켜 다른 기하적인 문제로 변환하지 않고 점  $p_i$ 를 극으로 하는 대원을 두 평면  $E^*$ 와  $E^-$ 에 투상시켜 곧바로 LP 부등식을 만드는 것이다.

LP 1. Minimize  $x$  subject to  $x_i x + y_i y + z_i \geq 0$  for all  $p_i \in P$ .

LP 2. Minimize  $x$  subject to  $x_i x + y_i y - z_i \geq 0$  for all  $p_i \in P$ .

LP에 대한 연구는 최근까지 많은 결과가 보고되고

있다. 차원이 고정되었을 때 LP를  $O(n)$  시간에 해결하는 알고리즘은 Megiddo<sup>[12]</sup>와 Dyer<sup>[14]</sup>에 의하여 처음 제시되었다. 그 후 Seidel<sup>[14]</sup>은 구현하기 쉬운 간단한 불규칙성 알고리즘을 제안하였다.

반구성 검사가 끝난 후 불록형을 구하기 전에 먼저 처리해야 할 degeneracy를 설명한다.  $P$ 가 반구안에 포함되면서 불록형이 정의되지 않는 경우는 두가지밖에 없다. 하나는  $p_i = -p_j$ 와 같이 서로 부호만 다른 두 점이 있는 경우이고 다른 하나는 세 점이 하나의 대원  $g$ 상에 있으며  $g$ 의 어느 반원에도 포함되지 않을 때이다. 전자의 경우에는 가시선이 하나의 간선이며 후자의 경우는 가시선이 하나의 점뿐이다. 따라서 이러한 경우에는  $P$ 의 불록형을 구성하지 않고 직접 가시선을 구성한다.

이러한 degeneracy를 찾기 위하여  $P$ 를 포함하는 최소구를 이용한다.  $(x, y, z)$ 를 구의 중심이라 하면  $P$ 를 포함하는 최소구를 찾는 문제는 다음과 같이 표현된다.

Minimize  $r$  subject to

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 \leq r^2 \text{ for all } p_i \in P$$

$(x^c, y^c, z^c)$ 를 위의 해라고 하자.  $(x^c, y^c, z^c) \neq (0, 0, 0)$ 이면 최소포함구(the smallest enclosing sphere)  $s^c$ 는  $S^2$ 보다 작으므로  $P$ 의 불록형을 구성할 수 있다. 그러나  $(x^c, y^c, z^c) = (0, 0, 0)$  이면  $s^c$ 는  $S^2$ 와 일치하고  $P$ 의 불록형은 정의되지 않는다.  $P$ 를 경계짓는 대원을  $g^c$ 라 하자. 그러면 degeneracy중 두 가지는  $g^c$ 상에 있는  $P$ 의 점들이 어느 반원에 포함되는지 아닌지를 검사하면 된다. 이 문제는 반구성 문제의 2차원 형태이다.

최소 포함구를 구하는  $O(n)$ 시간의 알고리즘은 Megiddo<sup>[11]</sup>에 의하여 처음 제안되었다. 그 후 Welzl<sup>[21]</sup>은 구현하기 간편하고 실제적으로 매우 빠른 간단한 불규칙 알고리즘을 제안하였다.

### 3.2 구불록형의 구성

구상의 알고리즘들의 시간복잡도를 이론적으로 분석하기 위하여  $P$ 를 하나의 평면에 중심투상시켜 평면 불록형 알고리즘을 적용하는 경우가 많다<sup>[1,2,3,8,9,17]</sup>. 그러나 반구성 문제에서의 마찬가지로 안전한 수치적 계산을 위하여 여기에서는 평면의 불록형 알고리즘인 Graham 스캔<sup>[9]</sup>을  $S^2$ 상에서 직접 적용시키는 구체적이고 효율적인 방법을 제시한다. 이 알고리즘은 극점 찾기, 극점에 대한 각도 정렬(angular sort)하기, 정렬된 순서로 스캔하기등 크게 3단계로 구성되어 있다.

극점은 반구성 검사에서 degeneracy를 찾아내기 위하여 계산된  $P$ 의 최소포함구의 중심  $(x^c, y^c, z^c)$ 를 이용

한다.  $P$ 에 속하는 점중에서  $(a, b, c)$ 를  $(x^c, y^c, z^c)$ 에서 가장 먼 점이라 하면 이 점이 극점중에 하나이다.

점  $(a, b, c)$ 를 중심으로  $P$ 의 나머지 점들을 각도 정렬할 차례이다. 볼록할 알고리즘을 구현할 때 가장 주의할 부분은 전체적인 시간 복잡도  $O(m \log m)$ 을 지배하는 이 각도 정렬이다.  $S^2$ 상에서의 각도 정렬은 평면보다 많은 계산이 요구된다. 그러나 다음과 기하적 관찰을 이용하면 보다 효율적인 연산으로 각도 정렬을 할 수 있다. 기준점  $(a, b, c)$ 을 접하는 평면에 모든 점을 수직으로 평행투상시켜 평면에서 각도 정렬한 순서는  $S^2$  상의 각도 정렬된 순서와 일치한다.

기준점에 접하는 평면에 모든 점을 투상시키는 계산은 그 기준점이  $z$ 축에 일치하도록 모든 점을 회전 변환시켜  $z$ 값은 버리고  $x, y$  값만 취하면 된다. 기준점  $(a, b, c)$ 를  $z$ 축과 일치하도록 회전시키기 위한 변환행렬은 다음과 같이 얻을 수 있다.

$$R_x \cdot R_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & \frac{b}{d} & 0 \\ 0 & -\frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} d & 0 & a & 0 \\ \frac{ab}{d} & \frac{c}{d} & b & 0 \\ \frac{ac}{d} & -\frac{b}{d} & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } d = \sqrt{b^2 + c^2}$$

평면상에서 각도 정렬을 수치적으로 견고하게 하기 위하여 두 점  $x_1, y_1, x_2, y_2$ 를 지나는 직선의 기울기를 대신하여 수치  $dy/dx + kd$ 를 사용하는 방법이 많이 사용되고 있다. 여기에서  $dx = x_2 - x_1$ 이고  $dy = y_2 - y_1$ 이다. 그러므로 극점  $(a, b, c)$ 와 점  $(x_i, y_i, z_i) \in P$ 에 대하여 각도 정렬에 사용되는 값은 다음과 같이 계산된다.

$$\frac{cy_i - bz_i}{|(b^2 + c^2)x_i - aby_i - acz_i| + |cy_i - bz_i|}$$

$P$ 의 볼록형을 구한 후 가시맵을 구하기 전에 처리하여야 할 degeneracy는 볼록형의 정점이 하나 또는 두 개인 경우이다. 정 점이 하나인 경우 이원 다각형은 반구가 된다. 이 경우에는 대원을 세 개의 선분으로 나누어 경계를 표시한다. 또한 정점이 두 개인 경우 이원 다각형은 두 반원으로 이루어진다. 이러한 경우 반원을 각각 두 개의 선분으로 나누어 표시한다. 일반적인 경우

에는  $CH(P)$ 와 이원성에 의하여  $VM(P)$ 를 구성한다.

지금까지의 결과를 이용하여 어느 표면의 가우시안 맵  $P$ 가 주어진 경우 견고하게 가시맵을 구성하는 알고리즘은 다음과 같다.

**Procedure VisibilityMap( $P$ )**

**Input:**  $P = \{p_1, \dots, p_m\}$  that is the Gaussian map of a surface  $S$ , where  $p_i = (x_i, y_i, z_i)$

**Output:** The visibility map (*NULL*, *POINT*, *EDGEM*, *POLYGON*) of  $S$

1. Obtain a solution  $(x^*, y^*)$  from either of the two LPs of
  - 1.1 Minimize  $x$  subject to  $x_i x + y_i y + z_i \geq 0$  for all  $p_i \in P$ .
  - 1.2 Minimize  $x$  subject to  $x_i x + y_i y - z_i \geq 0$  for all  $p_i \in P$ .
2. if  $(x^*, y^*)$  is not found then return(*NULL*).

Obtain a solution  $(x^c, y^c, z^c)$  of the smallest sphere enclosing  $P$  that inimizizes  $r$  subject to  $(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 \leq r$  for all  $p_i \in P$ .

3. if  $(x^c, y^c, z^c) = (0, 0, 0)$  then
  - $Q \leftarrow$  the points on the great circle  $GC(x^*, y^*, z^*)$ .
  - Find a feasible solution  $x^{**}$  from either of the two LPs
    - 3.1 Minimize  $x$  subject to  $x'_i x + y'_i y \geq 0$  for all  $x'_i, y'_i, z'_i \in Q$ .
    - 3.2 Minimize  $x$  subject to  $x'_i x - y'_i y \geq 0$  for all  $x'_i, y'_i, z'_i \in Q$ .
  - Compute the visibility map directly.
  - if  $x^{**}$  is found then return(*EDGE*)
  - else return(*POINT*)
  - endif

4. Find the farthest point  $(a, b, c) \in P$  from  $(x^c, y^c, z^c)$ .
5. Sort  $P$  using the value of

$$\frac{cy_i - bz_i}{|(b^2 + c^2)x_i - aby_i - acz_i| + |cy_i - bz_i|}$$

6. Apply the Graham scanning for obtaining the convex hull  $\{u_1, \dots, u_l\}$  of  $P$ .
7. if  $l=1$  then split the great circle  $GC(u_1)$  and return (*POLYGON*).
8. if  $l=2$  then split the two half-circles of  $GC(u_1), GC(u_2)$  and return(*POLYGON*).
9. Construct the visibility map  $\{v_1, \dots, v_l\}$  such that  $v_i \leftarrow u_{i-1} \times u_i$ .

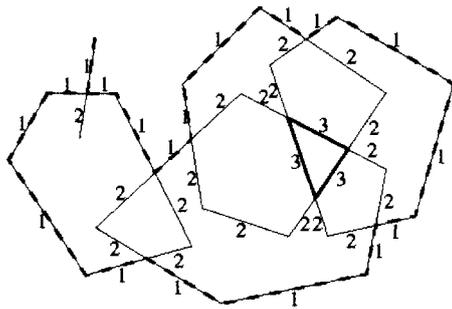


Fig. 4. Ownerships of split edges.

10. Return(POLYGON).  
endProcedure VisibilityMap

### 4. 간선 기반 분할

#### 4.1 최대 교차 및 간선의 교차

Fig. 4에서 보는 바와 같이  $n$ 개의 구 볼록 다각형이 있을 때 분할된 각 간선에는 그 간선에는 그 간선을 포함(소유)하는 다각형의 개수가 표시되었다고 하자. 앞으로  $k$ 개의 다각형에 소유되는 간선이나 면등을  $k$ -교차한다라고 말한다. 만약 최대 최소의 교차수를 각각  $min$ 과  $max$ 라 하면  $k$ -교차 면은 다음과 같이 특성화된다.

- $k = min$  :  $min$ -교차 경계의 외부
- $min < k < max$  :  $k$ -교차의 내부 또는  $(k+1)$ -교차의 외부
- $k = max$  :  $max$ -교차 경계의 내부

일반적으로  $k$ -교차 면의 경계는  $k$ -교차와  $(k+1)$ -교차의 간선으로 구성되어 있다. 그러나 모든 최적화 문제는  $k$ 를 최소화 또는 최대화하는 것이므로 해가 되는 면의 경계는 항상  $min$ -교차 또는  $max$ -교차의 간선로만 구성된다. 간선 기반 분할 방식은 바로 이 관점에서부터 출발한다. 특히 구 볼록 다각형의 교차로 형식화되는 대부분의 문제는 교차되는 다각형의 개수를 최대화하는 것이다. 이러한 성질을 이용하기 위하여 임의의 개수의 다각형의 교차를 찾을 수 있는 면 기반 분할과 달리 우리의 알고리즘은 다각형의 최대 교차를 계산한다. 최소 교차의 경우에도 임의의 개수보다는 더 효율적으로 해결될 수 있다.

최대 교차의 해가 되는 각 면들은 항상 볼록하고 홀(hole)이 없다는 또 하나의 성질이 있다. 따라서 최적화 문제를 해결하기 위하여 면 기반 분할에서 사용하는 면들간의 포함 관계 그래프 자료구조가 필요하지 않다. 또한 면들을 다루는 자료구조보다 더 간단한 간선을

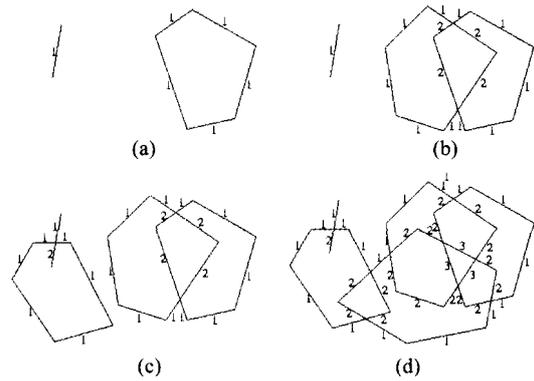


Fig. 5. Splitting edges and updating ownerships.

다루기 위한 자료구조를 사용할 수 있다.

구 볼록 다각형의 최대 교차를 계산하기 위하여 각 간선과 그 소유권을 노드에 나타내는 단방향 순환 리스트를 사용하는 간선 기반 분할 방식을 사용한다. 초기에는 각 다각형을 나타내는 하나의 순환 리스트로 나타낸다. 그리고 Fig. 5에서 보는 바와 같이 한 쌍의 다각형들이 하나씩 증분적으로 교차되면서 분할되는 간선을 나타내는 노드를 각 순환 리스트에 삽입하고 모든 간선의 소유권을 갱신한다. 이 때 최종 결과는 다각형을 교차시켜나가는 순서에 관계없이 동일하다. 또한 하나의 점 또는 간선의 degeneracy도 처리될 수 있다.

다각형들이 차례대로 하나씩 교차시켜나감으로 한 쌍의 다각형을 교차시키는 것은 중요한 기초 연산이다. 평면에서는 주어진 두 다각형의 교차에 대해서는 많은 연구가 이루어져 두 볼록 다각형의 교차는 선형 시간에 계산할 수 있다<sup>[13,15,19]</sup>. 이 중에서 가장 우아하고 구현하기 간단한 것은 간선을 전진시켜나가면서 교차를 계산하는 O'Rourke 등의 알고리즘<sup>[13]</sup>이다. 그러나 3절에서와 같이 수치적으로 안전한 계산을 위하여는 구상에서 간선을 전진시키며 교차를 계산하기 위하여 제시된 새로운 알고리즘<sup>[25]</sup>을 사용하여야 한다.

구상의 간선을 전진시키는 이 알고리즘은 두 간선간의 직면(facing) 관계와 상대적 위치를 기초로 하고 있다. 다각형  $P = \{p_1, \dots, p_m\}$ 와  $Q = \{q_1, \dots, p_n\}$ 의 정점은 반시계 방향으로 되어있다고 가정하고 간선  $\overline{p_{i-1}p_i}$ 는 간단히  $\overline{p_i}$ 로 나타낸다. 또한 간선  $\overline{p_i}$ 를 지나는 대원으로 나뉘어지는 왼쪽 및 오른쪽 반구를 각각  $LHS(\overline{p_i})$ 와  $RHS(\overline{p_i})$ 로 나타낸다. 이러한 표현을 사용하여 다음과 같이 두 간선간의 직면 관계와 두 다각형간의 맞물림 상태를 먼저 정의한다.

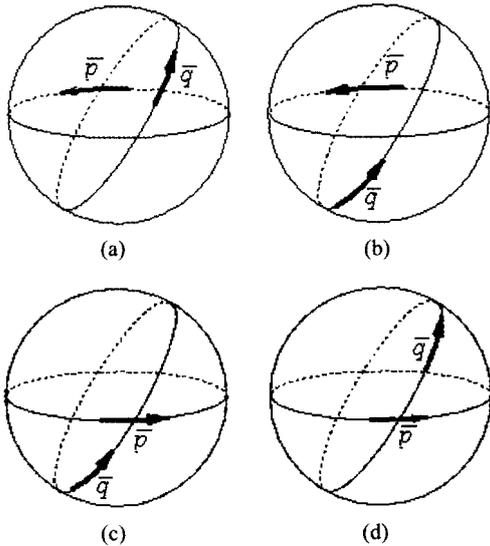


Fig. 6. Facing and non-facing of edges on  $S^2$ .

정의 1.  $\text{Face}(\bar{q}_j, \bar{p}_i)$ :

$$p_{i-1} \in \text{LHS}(\bar{q}_j) \wedge p_i \in \text{RHS}(\bar{q}_j) \wedge p_j \in \text{RHS}(\bar{p}_i)$$

$$\vee p_{i-1} \in \text{RHS}(\bar{q}_j) \wedge p_i \in \text{LHS}(\bar{q}_j) \wedge q_j \in \text{LHS}(\bar{p}_i)$$

정의 2. 교차하는 두 다각형  $P$ 와  $Q$ 상에서 진행중인 각각의 간선  $\bar{p}$ 와  $\bar{q}$ 가 있을 때  $\bar{q} \in \text{RHS}(\bar{p}) \vee \text{Face}(\bar{p}, \bar{q}) \vee \bar{p} \in \text{RHS}(\bar{q}) \vee \text{Face}(\bar{q}, \bar{p})$ 라면 맞물림 상태 (geared state)에 있다고 한다.

$S^2$ 상의 어느 반구안에서 간선  $\bar{p}$ 를 향하고 있는 모든 간선  $\bar{q}$ 는  $\bar{p}$ 를 직면하고 있다고 정의된다. Fig. 6(a)와 (c)의 경우  $\bar{q}$ 가  $\bar{p}$ 를 직면하고 있지만 (b)와 (d)의 경우는 그렇지 않다.

위 두 정의를 사용하여 구상에서 간선의 전진을 위한 직면 규칙을 다음과 같이 표현할 수 있다. 간선과 정점이 만나거나 정점과 정점이 교차되는 degeneracy의 경우 참고문헌<sup>[9]</sup>에서와 동일한 방법으로 처리될 수

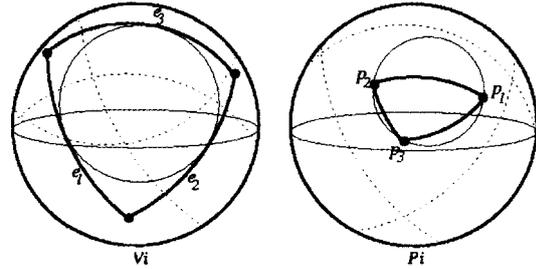


Fig. 7. Inscribing and circumscribing circles of a polygon.

있으며 보다 자세한 내용은 문헌<sup>[25]</sup>을 참고하면 된다.

### 4.2 중심 해의 계산

다각형들의 모든 쌍을 교차시킨 후에 모든 순환 리스트를 순환하여 가장 큰 소유권 값을 찾아낸다. 그리고 다시 모든 순환 리스트를 순환하여 그 값과 동일한 소유권을 가지는 간선들을 모은다. 이 간선들은 이산적인 형태이므로 연결하면 해의 경계를 구성할 수 있다. 그러나 대부분의 경우 하나의 해가 필요하며 경계의 중심에 있는 해가 선호된다. 따라서 여기에서는 해의 경계를 구성하지 않고 근사적으로 중심에 있는 해를 효율적으로 계산한다.

블록 다각형을 내접원, 즉 그 다각형 내부에서 가장 큰 원의 중심은 그 다각형의 중심에 좋은 근사치가 된다. 그러나 이 원이 접하는 간선이 될 수 있는 후보는 조합적 (combinatorial)이기 때문에 그 원의 중심을 정하기에 복잡하다. 그런데 Fig. 7에서 보는 바와 같이 그 다각형의 내접원의 중심은 이원 다각형의 외접원의 중심과 같다는 성질<sup>[8]</sup>을 이용하면 효율적으로 계산할 수 있다. 이 외접원은 2절에서 설명된 최소포함구를 구함으로써 가능하다.

여기에서 수치적 한계로 발생하는 교차점들의 위상 위배에도 이 방법이 견고함을 설명한다. Fig. 2에서와 같이 위상 위배가 생긴 경우에 최대 소유권을 가지는 간선을 모으면 하나의 면이 구성되지는 않는다. 그러

Table 1. Edge advancing rules based on facing on  $S^2$

	$\text{Face}(\bar{p}, \bar{q})$	$\neg \text{Face}(\bar{p}, \bar{q})$
$\text{Face}(\bar{p}, \bar{q})$	(규칙 0) advance the one that has not recently advanced, if geared advance $\bar{p}$ , if non-geared	(규칙 1) advance $\bar{q}$
$\neg \text{Face}(\bar{p}, \bar{q})$	(규칙 2) advance $\bar{p}$	(규칙 3) (1) advance $\bar{p}$ , if $p \in \text{RHS}(\bar{q})$ , (2) advance $\bar{q}$ , else if $p \in \text{RHS}(\bar{p})$ , (3) advance $\bar{p}$ , otherwise

나 위의 내접원을 구하는 방식의 계산은 이러한 경우에도 전혀 문제가 되지 않는다.

지금까지 설명한 간선 기반 분할 방식의 알고리즘을 표현하면 다음과 같다.

#### Procedure MaximumIntersection(VMS)

**Input:** A set of visibility maps  $VMS = \{VM_1, \dots, VM_n\}$

**Output:** An approximating centroid of a maximum intersection face

1. foreach a pair of  $(VM_i, VM_j) \in VMS$  do
  - Compute the intersection points of  $VM_i$  and  $VM_j$ .
  - Split the edges of  $VM_i$  and  $VM_j$  with the intersection points.
  - Update the ownerships of all edges of  $VM_i$  and  $VM_j$ .
- endforeach
2. Find an edge  $e$  that has the maximum number of ownerships by traversing VMS.
3.  $E = \{e_1, \dots, e_L\} \leftarrow$  edges with the same ownership with  $e$ , which are in VMS.
4. if  $L=1$  then return(the center of  $e_1$ ).
5.  $V \leftarrow$  the poles of great circles passing the edges in  $E$  such that  $v_i \leftarrow e_{i,1} \times e_i$ .
6. Find the smallest sphere enclosing  $P$ .
7. Return(the center of  $s$ ).

#### endProcedure MaximumIntersection

$VM_i$ 의 정점의 개수를  $v_i$ 라 하고  $v = \sum_{i=1}^n v_i$ 라 하자. 그러면 위 알고리즘에서 한 쌍의 구 볼록 다각형을 교차시키는 선형시간 알고리즘<sup>[23]</sup>을 이용하므로 모든 교차점을 찾아내기 위한 foreach 루프는  $\sum_{i=1}^{n-1} \sum_{j=i+1}^n O(v_i + v_j) = O(nv)$  시간에 수행된다. 최대 교차수를 결정하고 그 수를 소유권으로 가지는 간선을 모으기 위하여 각각 순환 리스트를 한번씩 순회하며 이것은 모든 점의 개수에 선형적이며  $O(m)$ 이다. 모아진 간선을 이용하여 해의 이원다각형을 구하는 시간과 최소포함구를 이용하여 근사적인 중심 해를 구하는 시간도 역시 간선의 개수에 선형적이므로 다음과 같은 결과를 얻을 수 있다.

**정리 1.** 구 볼록 다각형의 각 최대 교차 영역의 근사적인 중심 해는  $O(m)$  시간에 찾아진다. 여기서  $n$ 과  $v$ 는 각각 다각형과 모든 정점의 개수이다.

#### 4.3 적용된 예

가시맵을 구성하고 구 볼록 다각형들의 최대 교차를 찾는 알고리즘은 윈도우즈98 환경에서 C++ 프로그래밍

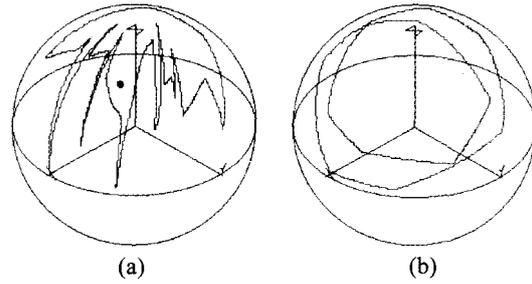


Fig. 8. Computation of a visibility map.

언어의 클래스로 구현되었다. 본 논문에서 사용하는 기본 알고리즘중에 LP와 최소포함구의 두 부분은 Seidel<sup>[14]</sup>과 Welzl<sup>[21]</sup>의 효율적인 불규칙 알고리즘을 각각 구현한 Hohmeyer<sup>[6]</sup>와 Erickson & Honda<sup>[5]</sup>의 공개 코드를 이용하였다. 주어진 표면의 수직벡터를 입력으로 받아 가우시안맵의 볼록형과 그 가시맵을 구하고 여러 가시맵으로 구를 분할하는 본 논문의 나머지 주요 알고리즘은 처음부터 구현하였으며 degeneracy 처리 부분을 제외하면 1000여 줄에 이른다.

Fig. 8은 구현된 코드에 의하여 주어진 하나의 가우시안맵에 대하여 그 가시맵을 구하는 과정을 출력한 것이다. (a)는 주어진 가우시안맵에서 찾아진 극점을 기준으로 각도정렬한 후 그 순서로 단순 다각형을 생성한 것이다. 이 그림에서 검은 점은 최소포함구의 단위중심벡터이다. (b)는 구하여진 가우시안맵의 볼록형과 그 이원다각형인 가시맵의 볼록형을 보여주고 있다.

Fig. 9은 구현된 코드에 의하여 여러 가시맵의 최대 교차 영역을 구하여 출력할 예이다. (a)는 주어진 가우시안맵의 볼록형들을 나타내고 (b)는 (a)의 양가시맵에서의 최대 교차면과 그 중심해를 (c)는 양가시맵 및 음가시맵에서의 최대 교차면과 중심 해를 각각 구한 결과이다.

양가시맵의 최대교차 (a)는 수치제어 절삭가공에서 3축 공구가 주어진 표면을 가공할 수 있는 방향을 결정하는데 사용될 수 있으며 양가시맵 및 음가시맵의 최대교차 (b)는 주어진 모델의 두 개의 거푸집을 설계할 때 두 거푸집이 분리될 수 있는 방향을 찾는 데 사용될 수 있다.

이 밖에도 구현된 코드는 여러 수치제어 절삭가공 환경에서 공구의 표면에 접근할 수 있는 방향을 찾는 데 이용될 수 있으며<sup>[11]</sup> 방향을 찾는 여러 계산 문제에 응용될 수 있는 잠재력이 있다.

Table 2는 불규칙적으로 생성된 자료로부터 사전에 구성된 가시맵 볼록다각형의 개수  $n$ 과 모든 정점의 개수  $v$ 가 변함에 따라 양가시맵의 최대 교차를 계산하는

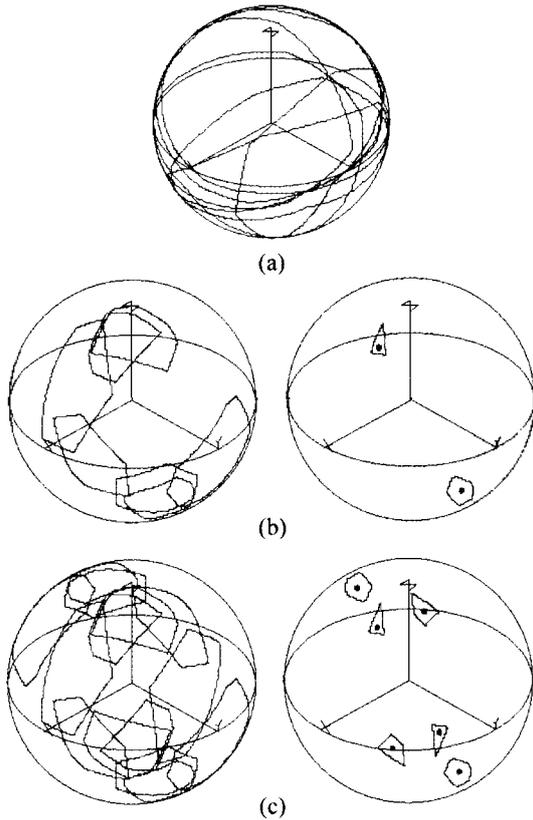
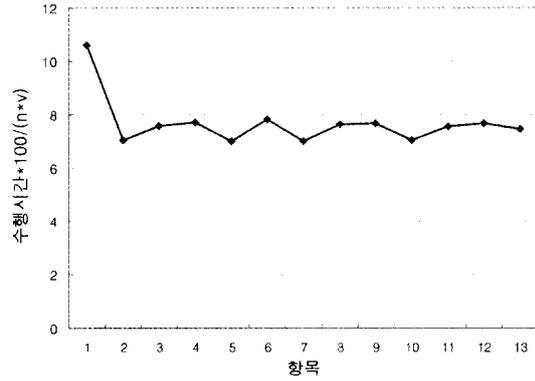


Fig. 9. Centroids of solution faces for the maximum intersection.

시간이 변하는 것을 여러번 측정 한 결과이다. 수행시간을  $mv$ 로 나눈 값이 거의 일정하므로 실제 구현된 코드에서도 수행 시간이 이론적으로 분석된 시간복잡도  $O(mv)$ 에 근사함을 확인시켜주고 있다. 프로그램이 개발된 환경은 64 메가 메모리의 펜티엄 컴퓨터이다.

5. 결 론

본 논문에서는 구 볼록 다각형의 최대 교차를 구하



기 위하여 구를 분할하는 견고한 방법을 제시하였다. 모든 해 영역의 중심은  $O(mv)$ 의 효율적인 시간에 구해진다. 여기에서  $n$ 과  $v$ 는 각각 다각형과 모든 정점의 개수이다. 또한 우리의 알고리즘은 간단한 순환리스트를 사용하여 안전한 수치 계산으로 구현될 수 있었고 모든 degeneracy를 처리할 수 있으므로 실제적이다. 찾아진 하나의 해는 해 영역의 근사적인 중심이므로 실제 CAD/CAM 응용에서 선호되는 해이기도 하다.

제안된 간선 기반 분할 방식은 다각형의 최대 교차만을 찾을 수 있으나 임의의 개수의 다각형의 교차를 찾을 수 있도록 확장하는 것이 가능하다. 즉 해 영역 경계의 해로 나오는 간선을 정렬만 시킨다면 연결된 경계를 구성할 수 있다. 이 때의 시간복잡도는 해로 나오는 간선의 개수  $L$ 을 정렬시키는 시간을 포함한  $O(vn+L\log L)$ 이 된다.  $L \leq n$ 인 경우에 해 영역의 경계를 구하는 Chen 등의  $O(vn\log n)$ 시간 일반적 알고리즘에 비하여 우수하다. 또한  $L \gg n$ 이라도 실제 구현 시에는 이산적인 간선을 병합하는 시간  $O(L\log L)$ 은 순환리스트의 단순한 자료구조보다 경계표현의 복잡한 자료구조를 사용하는 비용보다는 미미할 것으로 여겨지며 특히 교차점의 위상 위배를 극복할 수 있으므로 간선에 소유권을 유지하는 우리의 방법이 일반적인 교차를 구하는데도 효율적으로 사용될 수 있을 것으로 기대된다.

Table 2. Experiments of running time

항 목	1	2	3	4	5	6	7	8	9	10	11	12	13
다각형개수(n)	5	8	10	15	15	20	20	23	25	25	27	29	29
모든정점개수(v)	45	71	90	103	149	163	187	179	203	233	228	234	272
$n*v$	225	568	900	1545	2235	3260	3740	4117	5075	5825	6156	6786	7888
수행시간(1/1000)	24	40	68	119	156	255	262	314	388	410	465	520	589
수행시간*100/( $n*v$ )	10.6	7.04	7.58	7.7	7	7.82	7.01	7.64	7.66	7.04	7.55	7.67	7.46

## 감사의 글

이 논문은 우석대학교 교내 학술연구비지원에 의하여 연구되었음.

## 참고문헌

- Chen, L. L., Chou, S. Y. and Woo, T. C., "Separating and intersecting spherical polygons: computing machinability on three- four- and five-axis numerically controlled machines," *ACM Tr. on Graphics*, Vol. 12, No. 14, pp. 305-326, 1993.
- Chen, L. L., Chou, S. Y. and Woo, T. C., "Paring directions for mould and die design," *Computer-Aided Design*, Vol. 25, No. 12, pp. 762-768, 1993.
- Chen, L. L. and Woo, T. C., "Computational geometry on the sphere with application to automated machining," *Tr. ASME*, Vol. 114, pp. 288-295, 1992.
- Dyer, M. E., "Linear time algorithms for two- and three-variable linear programs," *SIAM J. Computing*, Vol. 13, No. 1, pp. 31-45, 1984.
- Erickson, J. and Honda, H., *Implementation code of the smallest enclosing sphere*.
- Hohmeyer, M. E., *Implementation code of linear programming*.
- Horowitz, E. and Sahni, S., *Fundamentals of Data Structure*, Computer Science Press, 1973.
- Gan, J. G., Woo, T. C. and Tang, K., "Spherical maps: their construction, properties, and approximation," *ASME J. Mechanical Design*, Vol. 116, pp. 357-363, 1994.
- Graham, R. L., "An efficient algorithm for determining the convex hull of finite planar set," *Information Processing Letters*, Vol. 1, pp. 132-133, 1972.
- Gupta, P. et al., "Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC machining," *Computer-Aided Design*, Vol. 28, No. 8, pp. 577-587, 1996.
- Megiddo, N., "Linear-time algorithms for linear programming in and related problems," In *Proc. 23rd Annual IEEE Sym. Found. Comput. Sci.*, pp 329-338, 1982.
- Megiddo, N., "Linear programming in linear time when the dimension is fixed," *J. ACM*, Vol. 31, No. 1, pp. 114-127, 1984.
- O'Rourke, J., Chien, C. B., Olson, T. and Naddor, D., "A new linear algorithm for intersecting convex polygons," *Computer Graphics and Image Processing*, Vol. 19, pp. 384-391, 1982.
- Seidel, R., "Small-dimensional linear programming and convex hulls made easy," *Discrete and Computational Geometry*, Vol. 6, pp. 423-434, 1991.
- Shamos, M. I. and Hoey, D., "Geometric intersection problems," *Seventeenth Annual IEEE Symposium on Foundations of Computer Science*, pp. 208-215, 1976.
- Suh, S. H. and Kang, J. K., "Process planning for multi-axis NC machining of free surfaces," *Int. J. Prod. Res.* Vol. 33, No. 10, pp. 2723-2738, 1995.
- Tang, K., Woo, T. and Gan, J., "Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining," *ASME J. Mechanical Design*, Vol. 114, pp. 477-485, 1992.
- Tangelder, J. W. H., *Automated fabrication of shape models of free-form objects with a sculpturing robot* Ph. D. Thesis, Industrial Design Engineering of Delft Univ. of Technology, 1998.
- Toussaint, G. T., "A simple linear algorithm for intersecting convex polygons," *The Visual Computer*, Vol. 1, pp. 118-123, 1985.
- Weiler, K., "Edge-based data structures for solid modeling in curved-surface environments", *IEEE Computer Graphics and Application*, Vol. 5, No. 1, pp. 21-40, 1985.
- Welzl, E., *Smallest enclosing disks (balls and ellipsoids)* New Results and New Trends in Computer Science, Springer Lecture Notes in Computer Science 555, pp. 359-370, 1991.
- Woo, T. C., "A combinatorial analysis of boundary data structure schema," *IEEE Comput. Graph. Appl.*, Vol. 5, No. 3, pp. 19-27, 1985.
- 김병국, 박준영, 위남숙, "Determination of cutting orientation in zigzag milling operations: a geometrical approach," 한국CAD/CAM학회논문집, 제2권, 제3호, pp. 186-194, 1997.
- 전차수, 차경덕, "복합곡면의 5축 NC가공을 위한 공구자세 최적화," 한국CAD/CAM학회논문집, 제5권, 제4호, pp. 393-402, 2000.
- 하종성, "구상의 볼록 다각형의 교차 계산을 위한 선형 시간 알고리즘," 한국정보과학회논문지: 시스템 및 이론, 제28권, 제2호, pp. 58-63, 2001.
- 하종성, 신성용, "삼차원 제조성에 응용할 수 있는 다면체 단조성의 특성화," 한국정보과학회논문지(A), 제24권, 제11호, pp 1051-1058, 1997.



## 하 종 성

1984년 서울대학교 컴퓨터공학과 학사  
 1986년 한국과학기술원 전산학과 석사  
 1996년 한국과학기술원 전산학과 박사  
 1986년~1989년 (주)현대전자산업 근무  
 1990년~ 현재 우석대학교 정보통신컴퓨터공학부 부교수  
 관심분야: 응용계산기하학, 컴퓨터그래픽스, CAD/CAM 등임