

〈 논문 〉

섬을 가진 2 차원 직선 폐곡선에 대한 새로운 오프셋 알고리듬

김현철*, 이성근*, 양민양†

(2005년 7월 13일 접수, 2006년 1월 13일 심사완료)

A New Offset Algorithm for Closed 2D Lines with Islands

Hyun-Chul Kim, Sung-Gun Lee and Min-Yang Yang

Key Words : Contour Parallel Tool Path(윤곽 평행 공구경로), Offset(오프셋), Bisector(이등분선), Invalid Offset Edge(무효화 오프셋 선)

Abstract

In this paper, a new offset algorithm for closed 2D lines with islands is introduced and the result is illustrated. The main point of the proposed algorithm is that every point is set to be an offset by using bisectors, and then invalid offset lines, which are not to be participated in offsets, are detected in advance and handled with an invalid offset edge handling algorithm. As a result, raw offset lines without local invalid loops are generated. The proposed offset method is proved to be robust and simple, moreover, has a near $O(n)$ time complexity, where n denotes the number of input lines. In addition, the proposed algorithm has been implemented and tested with 2D lines of various shapes.

1. 서 론

절삭가공은 전통적인 기계가공 방법으로 대부분의 기계업체에서 이용하는 모든 가공방법 중 70 ~ 80% 이상의 비중을 차지하고 있다. 이러한 기계가공 중 Harenbroek⁽¹⁾에 따르면, 기계부품의 거의 80%이상이 2.5D 밀링가공을 적용함으로써 가공할 수 있는데, 이는 많은 수의 기계부품이 2.5D이고, 더 복잡한 부품들은 2.5D 황삭 후에 3D-5D 정삭을 하기 때문이다. 그러므로 복잡한 2.5D 밀링 작업을 위한 공구경로 생성 알고리듬들은 기계가공에서 근본적으로 중요한 요소이다.

2.5D 밀링은 Z 축에 평행한 높이가 다른 각각의 가공 면들에서 독립적으로 이루어지므로, 2.5D 방식은 가공공간의 차원을 2D로 줄일 수 있다. 결과적으로 2.5D 밀링의 성과는 2D 평면가공을 위한 효율적인 알고리듬들에 기인한다. 2D 평면가

공에서 가장 많이 쓰이는 공구경로는 Fig. 1 과 같은 공구의 에어컷(air cutting)을 줄이면서 이송방향의 변화를 피할 수 있는 윤곽 평행 공구경로(contour parallel tool path)이다. 윤곽 평행 공구경로는 주어진 외곽 경계곡선의 순차적인 오프셋으로 얻어질 수 있으므로, 효율적인 2 차원 오프셋 알고리듬이 필수적이다.

또한, 최근 들어 대부분의 CAM(Computer Aided Manufacturing) 소프트웨어들은 삼각형 망(triangular mesh)으로 이루어진 다면체 모델(polyhedral model)을 이용하여 공구경로를 생성한다고 보고되고 있

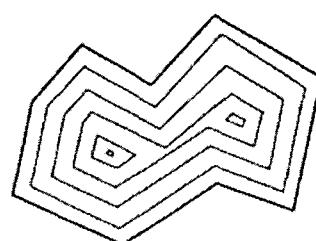


Fig. 1 Contour parallel tool path

† 책임저자, 회원, 한국과학기술원 기계공학과

E-mail : myyang@kaist.ac.kr

TEL : (042)869-3264 FAX : (042)869-5224

* 회원, 한국과학기술원 대학원 기계공학과

는데, 이는 기존의 자유 곡면 기반의 공구경로 생성에 비해서 뛰어나기 때문이다.⁽²⁾ 이러한 다면체로부터 얻어지는 폐곡선은 짧은 직선들의 집합이며, 효과적인 오프셋 알고리듬은 짧은 직선들로 이루어진 폐곡선을 효율적으로 다룰 수 있어야 한다.

2 차원 오프셋 알고리듬에 관한 기존의 연구들은 다음과 같이 크게 세 가지로 요약할 수 있다.

- 1) 상호 교선 접근법(pair-wise intersection approach)⁽³⁻⁴⁾
- 2) 보로노이 다이아그램 접근법(voronoi diagram approach)⁽⁵⁻⁷⁾
- 3) 픽셀 기반 접근법(pixel-based approach)⁽⁸⁻⁹⁾

첫 번째 방법인 상호 교선(pair-wise intersection) 접근법은 각각의 윤곽선들을 오프셋하고, 이들의 교점을 이용해 오프셋 선들을 트림(trim)하여 최종적인 오프셋을 구하는 방법이다. Bruckner⁽³⁾에 의해 제안된 이 방법은 Fig. 2에서 보는 바와 같이, 모든 유크선들에 대해 각각의 오프셋 선들을 구한 후 오프셋 한 선들 사이의 빈틈(gap)을 원호(arc)로 매꿔 폐곡선을 생성한다. 마지막으로 오프셋 한 선들의 모든 교점들로부터 자체적인 꼬임을 찾고, 곡선 내부의 불필요한 모든 루프(loop)를 제거함으로써 최종적인 오프셋 결과를 얻는다.

상호 교선 방법과 관련된 기하학적인 요소들이 Fig. 2에 보여진다. Fig. 2에 나타나 있듯이, 2 가지 종류의 무효화 루프(invalid loop)가 있는데, 첫 번째는 하나의 교점에 의해서 생기는 꼬임으로 국부적 무효화 루프(local invalid loop)라고 하고, 두 번째는 한 쌍의 교점에 의해서 생기는 꼬임으로 전역적 무효화 루프(global invalid loop)라고 한다. 상호 교선 방법의 가장 큰 문제점은 자체적인 꼬임에 의해서 생기는 국부적 무효화 루프의 제거가 $O(n^2)$ 의 시간 복잡도(time complexity)를 가지는 계산시간이 오래 걸리는 작업⁽¹⁰⁾이며, Held⁽⁶⁾가 지적했듯이 수치적인 불안정성을 야기 시킬 수 있다는 것이다.

두 번째 방법은 보로노이 다이아그램(voronoi diagram)을 이용하는 것으로 Fig. 3에 나타나 있듯이 유크곡선들을 직접 오프셋 하는 방법을 사용하지 않고, 영역을 분할하는 개념인 보로노이 다이아그램을 이용하여 꼬임과 루프 제거 등과 같은 문제를 발생시키지 않으면서 오프셋 곡선을 얻을 수 있다. 이 방법은 앞의 방법에 비해서 효율적이고 안정적이지만, 보로노이 다이아그램을 구성하는 것이 쉽지 않고, 이 역시 수치적인 불안정성을 갖는다.⁽¹¹⁾

세 번째 방법인 픽셀 기반(pixel-based) 방법은 수치적으로 안정하다는 장점은 있지만, 원하는 정

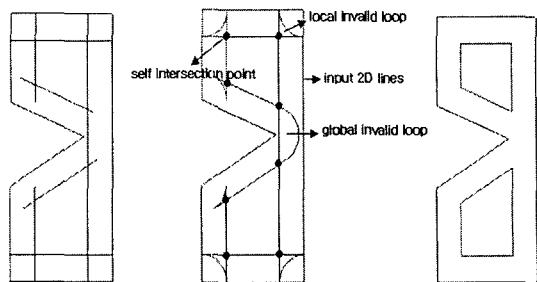


Fig. 2 Pair wise intersection approach

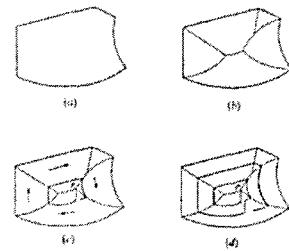


Fig. 3 Voronoi diagram approach

밀도를 이루기 위해서 엄청난 양의 메모리와 계산시간이 소요되는 단점이 있다.

위에서 언급된 문제들을 해결하기 위한 많은 알고리듬들이 그 동안 제시되어 왔지만, 다면체로부터 얻은 많고 짧은 직선들을 원호부근에서 오프셋 할 때는, 본질적으로 수치적인 불안정성을 갖는다. 다면체 모델로부터 얻어진 유크 경계곡선은 직선들로 이루어진 폐곡선이고, 특히 원호부근은 짧고 많은 직선들로 이루어진 곡선이다. Fig. 4는 다면체로부터 얻어진 유크 경계 곡선(boundary curve)과 기존의 상호 교선 오프셋 알고리듬으로부터 얻어진 원시 오프셋 곡선(raw offset curve)의 일부를 원호부근에서 확대한 그림으로 오프셋 직선들 간의 많은 교점들을 갖는 것을 볼 수 있다. 이로부터 국부적 무효화 루프를 제거하고 원시 오프셋 곡선을 얻는 것은 매우 어렵고 수치적인 불안정성을 낳는 근본적인 원인이다.

보로노이 다이아그램을 이용한 알고리듬도 역시 다면체로부터 얻어진 짧은 직선들에 대해서 적용될 때, 원호부근에서 위와 유사한 문제점을 지닌다. Fig. 5는 이 문제점을 보여주고 있으며, 보로노이 다이아그램을 만들기 위한 이등분선(bisector)들의 많은 교점들을 나타낸다. 보로노이 다이아그램을 만들기 위해서는 이등분선들을 서로 트림(trim)하는 과정이 필요하나, 원호부근에서의 특이성 때문에 이 역시 수치적인 불안정성을 가질 수 밖에 없다.

이러한 원호 특이성에 기인한 수치적인 불안정

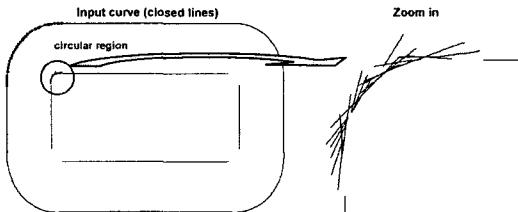


Fig. 4 Circular singularity in pair wise offset

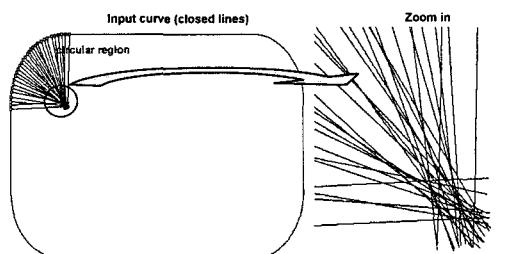


Fig. 5 Circular singularity in voronoi diagram construction

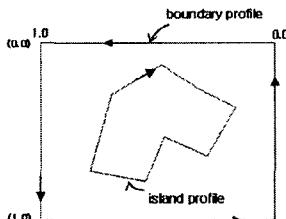


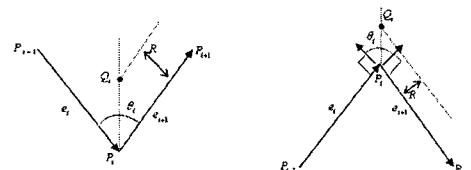
Fig. 6 Input profile

성은 다면체 모델 원호부근의 많고 짧은 직선들을 오프셋 하는 과정에서 국부적 무효화 루프를 제거하는데 발생하고, 원시 오프셋 곡선을 얻을 때 국부적 무효화 루프를 제거할 수 있다면 오프셋 알고리듬의 수치적인 안정성을 기대할 수 있다. 본 논문에서 제안하는 새로운 오프셋 알고리듬은 다면체로부터 얻어진 많고 짧은 2 차원 직선들에 대해서 낮은 시간 복잡도(low time complexity)를 갖고, 수치적으로 안정한 오프셋을 구현하는 것이다.

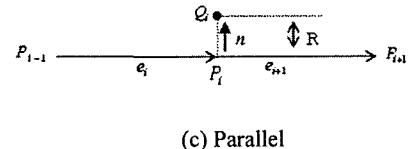
2. 새로운 오프셋 알고리듬

2.1 이등분선을 이용한 오프셋 점 계산

오프셋 알고리듬의 적용을 위해서, 입력 파일의 방향성을 정의하는데, Fig. 6에서 나타나 있듯이 외곽경계는 반 시계(counterclockwise) 방향으로 섬은 시계(clockwise) 방향으로 정의한다. 또한 각 직선들의 파라미터(parameter)도 $t \in [0,1]$ 와 같이 0에서 1의 범위를 갖는 값으로 정의한다. 임의의 꼭지점의 오프셋 점은 꼭지점의 유형에



(a) Concave (b) Convex



(c) Parallel

Fig. 7 Offset points according to vertex types

따라 다르게 계산되며, 모든 경우에 오프셋 점들은 Fig. 7에서 점선으로 표시된 이등분선 위에 놓이게 된다. Fig. 7에서 $P_i (i=0,1,\dots)$ 은 꼭지점들이고 $Q_i (i=0,1,\dots)$ 은 P_i 의 오프셋 점들이다. Fig. 7(a)의 경우처럼 꼭지점 P_i 가 오목할 때, 오프셋 점 Q_i 는 식 (1)과 같이 계산된다.

$$Q_i = P_i + \frac{e_{i+1} - e_i}{|e_{i+1} - e_i|} \cdot \frac{R}{\sin(\theta_i/2)} \quad (1)$$

여기서, $e_i = \frac{P_i - P_{i-1}}{|P_i - P_{i-1}|}$, R 은 오프셋 거리, θ_i 는 $P_{i-1}P_i$ 와 P_iP_{i+1} 사이의 각도이다.

Fig. 7(b) 경우는 꼭지점 P_i 가 불록한 경우이고, 이때 오프셋 점 Q_i 는 식 (2)와 같이 계산된다.

$$Q_i = P_i + \frac{e_i - e_{i+1}}{|e_i - e_{i+1}|} \cdot \frac{R}{\cos(\pi - \theta_i/2)} \quad (2)$$

여기서, $e_i = \frac{P_i - P_{i-1}}{|P_i - P_{i-1}|}$, R 은 오프셋 거리, θ_i 는 $P_{i-1}P_i$ 와 P_iP_{i+1} 사이의 각도이다.

마지막으로, Fig. 7(c)의 경우처럼 꼭지점 P_i 가 평행일 때는 오프셋 점 Q_i 는 식 (3)과 같이 계산된다.

$$Q_i = P_i + n \cdot R \quad (3)$$

여기서, $n = \frac{Q_i - P_i}{|Q_i - P_i|}$ 이다.

2.2 타당성 검사 및 무효화 오프셋 선 처리

원시 오프셋 선(raw offset lines)은 오프셋 점들

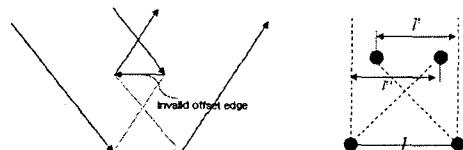


Fig. 8 Invalid offset edge

을 순서대로 연결하면 얻을 수 있지만, 그 오프셋 선들 가운데 오프셋 후에 선분의 방향이 바뀌는 선들을 무효화 오프셋 선(invalid offset edge)이라고 하고, 무효화 오프셋 선들은 오프셋이 정상적으로 이루어지지 않은 선들로 원시 오프셋 선에 포함되지 않아야 한다. Fig. 8은 오프셋 후에 선분의 방향이 바뀐 무효화 오프셋 선을 보여준다. 즉, $l - (l' + l'') \leq 0$ 이 만족하면, 그 선분은 무효화 오프셋 선이다. Fig. 9는 무효화 오프셋 선들을 제거한 원시 오프셋 선을 보여준다. 이 경우 오프셋 선들은 임의의 무효화 오프셋 선의 앞의 선과 뒤의 선이 만나지 않기 때문에, 폐곡선을 이루지 못한다. 이를 해결하기 위해서 즉 임의의 무효화 오프셋 선의 앞의 선과 뒤의 선을 연결하기 위해서, 무효화 오프셋 선 처리(Invalid offset edge handling: IOEH) 알고리듬이 필요하다. 원시 오프셋 선은 Fig. 8에서 살펴본 바와 같이 무효화 오프셋 선, 즉 오프셋 선의 방향이 바뀌는 것 때문에 꼬임이 발생한다. 이러한 꼬임이 발생한 경우는 단순히 앞의 선과 뒤의 선을 연결해 주면 뒤에 설명할 루프 처리(loop-processing) 알고리듬에 의해서 처리가 되지만, 무효화 오프셋 선에 의해서 꼬임이 발생하지 않는 특별한 경우가 존재하며, 이 경우에 무효화 오프셋 선 처리 알고리듬이 적용된다. 무효화 오프셋 선 처리 알고리듬은 다음의 두 가지 경우에 각각 다르게 적용된다.

Case (I): 무효화 오프셋 선의 앞의 선과 뒤의 선 사이의 무효화 오프셋 선의 개수가 하나인 경우

Case (II): 무효화 오프셋 선의 앞의 선과 뒤의 선 사이의 무효화 오프셋 선의 개수가 둘 이상인 경우

Case I 알고리듬은 임의의 무효화 오프셋 선의 앞의 선과 뒤의 선 사이의 무효화 오프셋 선의 개수가 하나인 경우, 즉 무효화 오프셋 선들이 연속적이지 않을 경우에 적용된다. 이 알고리듬의 순서는 다음과 같다.

1. 앞의 선과 뒤의 선을 연장하여 두 선의 교점을 찾는다.
2. 두 선의 교점으로 앞의 선의 시작점과 뒤의 선의 끝점을 갱신한다.

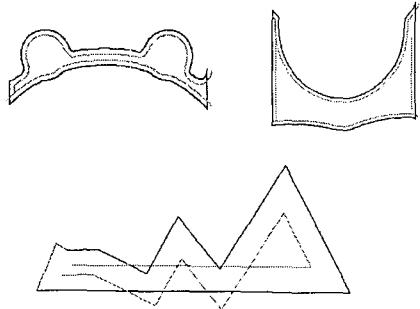
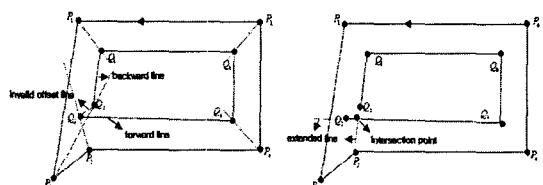
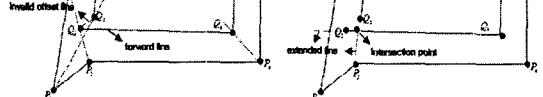


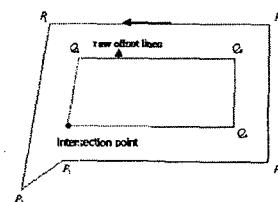
Fig. 9 Raw offset lines without invalid offset edges



(a)



(b)



(c)

Fig. 10 Invalid offset edge handing algorithm of Case I

Fig. 10(a)에서 오프셋 선 Q_2Q_3 는 선분 P_2P_3 의 오프셋 선으로 오프셋 후에 방향이 바뀌었으므로, 무효화 오프셋 선이다. 이 무효화 오프셋 선은 오프셋이 되지 않아야 하는 선으로 원시 오프셋 선들에 속하지 않아야 한다. 무효화 오프셋 선이 연속적이지 않으므로 Case I에 해당하는 알고리듬이 적용되며 그 순서는 Fig. 10에 나타나 있다.

무효화 오프셋 선들이 연속적일 경우에는 Case II의 무효화 오프셋 선 처리 알고리듬이 적용되어 그 순서는 다음과 같다.

1. 무효화 오프셋 선에 해당하는 오프셋 전의 선분을 선택하고, 선택된 선분이 타당한 오프셋 선들과 교차하는지를 검사한다. 선택된 선분 중 교차하는 선분의 순서를 기록한다.
2. 기록된 순서에 해당하는 오프셋 전의 선분을 기준의 상호 교선 오프셋 방법에 의해서 오프셋 한다.

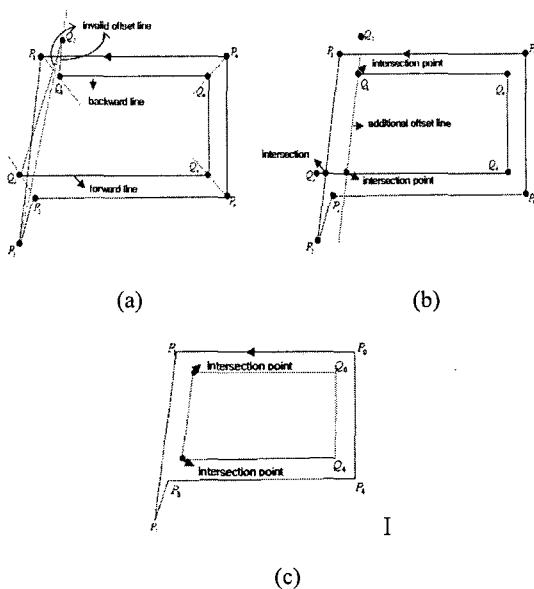


Fig. 11 Invalid offset edge handing algorithm of Case II

3. 상호 교선 오프셋 방법에 의해서 추가된 오프셋 선들과 앞의 선과 뒤의 선의 교점을 찾고, 이 교점들의 정보를 이용해 앞의 선과 뒤의 선을 생성한다.

Fig. 11(a)에서 오프셋 선 Q_1Q_2 , Q_2Q_3 는 무효화 오프셋 선들이고, 무효화 오프셋 선들이 연속이므로 Case II에 해당하는 무효화 오프셋 선 처리 알고리듬이 적용된다. 무효화 오프셋 선 Q_1Q_2 , Q_2Q_3 에 해당하는 오프셋 전 선분 P_1P_2 , P_2P_3 가운데 P_1P_2 는 타당한 오프셋 선 Q_3Q_4 와 교차하므로, Fig. 11(b)에서 보이는 것처럼 P_1P_2 의 추가적인 오프셋 선이 기준의 상호 교선 방법에 의해서 얻어진다. 이렇게 얻어진 추가적인 오프셋 선과 앞의 선과 뒤의 선의 교점을 찾고, 이 교점들로부터 앞의 선과 뒤의 선을 생성한다. Case II 알고리듬의 전체적인 절차는 Fig. 11에 나타나 있다.

Case I과 Case II에 의해서 앞의 선과 뒤의 선이 생성되지 않는 경우는 앞에서 설명한 꼬임이 발생한 경우로 단순히 두 선 사이를 임의의 선으로 연결해 주면, 뒤에서 설명할 루프 처리 알고리듬에 의해서 꼬임이 제거된다.

위에서 설명한 것처럼, 이동분선을 이용하여 오프셋 점들을 계산하고 오프셋 되지 않아야 하는 선들을 미리 찾아내서 무효화 오프셋 선 처리 알고리듬을 적용하여 처리함으로써 국부적 무효화 루프가 제거된 원시 오프셋 선을 얻을 수 있다.

Fig. 12는 Fig. 9의 무효화 오프셋 선들이 없는 원시 오프셋 선에 무효화 오프셋 선 처리 알고리듬

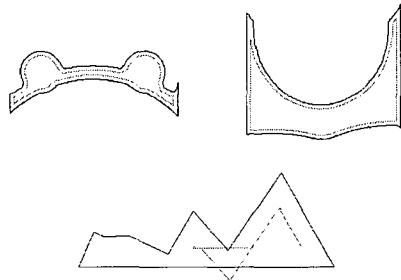


Fig. 12 Raw offset lines after an invalid offset edge handling algorithm is applied

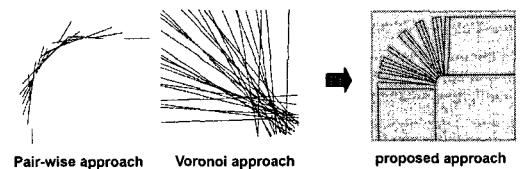


Fig. 13 Offset results by a new offset approach in circular region

을 적용하여 국부적 무효화 루프가 제거된 원시 오프셋 선을 얻은 결과이다. 이 과정이 기준의 오프셋 방법이 갖고 있는 미세한 직선으로 이루어진 곡선을 처리하는데 많은 시간이 소요되며, 수치적인 불안정성을 갖는 문제점을 해결하는 가장 중요한 부분으로, Fig. 13에서 보는 바와 같이 기준의 방법들이 가지고 있는 원호 부근에서의 수치적인 불안정성을 해결할 수 있다.

2.3 루프 처리 알고리듬

주어진 입력 프로파일이 위에서 설명한 알고리듬들에 의해서 오프셋이 되면, 국부적 무효화 루프는 제거되는 반면 Fig. 14(a)에서처럼 전역적 무효화 루프(global invalid loop)는 남게 된다. 이 경우 타당한 루프(valid loop)는 남기고 무효화 루프는 모두 제거해야 한다. 루프 처리(loop-processing) 알고리듬은 오프셋 하기 전 입력 프로파일의 외곽 경계는 반시계 방향이고, 섬은 시계 방향으로 정의된 사실에 기반을 둔다. 또한 원시 오프셋 선들에 대해서 그리드 타입 헤시 테이블을 이용하여 오프셋 선들 간의 교차하는 정보를 모두 저장한다. 어떤 오프셋 선이 몇 번째 오프셋 선과 교차하는지와 어떤 파라미터 값을 갖는지의 정보를 모두 저장하고 무효화 루프를 제거하는 과정에 사용된다. 루프 처리 알고리듬은 다음의 추적 법칙(tracing rule)과 4 가지 절차에 의해 타당한 루프들을 얻을 수 있다.

- 추적 법칙(tracing rule): 어떤 시작점에서부터 선들에 부여된 방향을 따라 선들을 이동하면

서 교차점이 있는가를 검사한다. 교차점이 없는 경우는 방향을 따라 다음 선으로 이동하고, 교차점이 있는 경우는 그 교차점을 전에 방문한 적이 있나 검사한다. 방문한 적이 없는 교차점일 경우 그 교차점을 다음 루프를 찾는 시작점의 정보로 스택(stack)에 저장하고 그 선과 교차한 다른 선으로 이동 방향을 옮긴다.

1. 시작점을 저장한다.
2. 시작점에서부터 위에서 언급한 추적 법칙에 따라 원시 오프셋 선들을 이동한다. 추적 법칙에 따라 이동하면서 방문한 선들의 끝점이 시작점과 일치하는지를 검사하고, 시작점과 같게 될 때 지금까지 이동한 선들을 반환한다. 이렇게 반환된 선들은 하나의 루프를 형성한다.
3. 이렇게 얻어진 루프의 방향성을 조사한다. 방향이 오프셋하기 전의 방향과 같으면 타당한 루프이고 방향이 반대인 루프는 무효한 루프로 제거되어야 한다.
4. 스택(stack)에 시작점이 있으면, 스택에서 시작점을 꺼내어 위의 1-3 의 과정을 반복한다. 스택에 더 이상 시작점이 없으면, 모든 오프셋 선들을 방문했으므로 알고리듬을 종료한다.

루프의 방향 검사는 폐 경로 정리(closed-path theorem)⁽¹²⁾에 의해서 가능하다. 모든 도형의 외곽의 합은 360° 이고 시계 방향 혹은 반시계 방향에 따라서 부호가 바뀌게 된다. 이를 이용하여 오프셋하기 전과 오프셋 한 후, 외곽의 합의 부호가 반대인 루프를 제거해 주면 된다. 지금까지의 과정을 통해서 전역적 무효화 루프들이 제거된 결과는 Fig. 14 (b)와 같다.

2.4 외곽경계와 섬의 병합

마지막으로 앞의 과정들을 통해 구해진 모든 무효화 루프들이 제거된 외곽경계들과 섬들을 병합(merging)하는 과정이 필요하다. 이를 구현하는 방법은 루프 처리 알고리듬과 유사하다. 우선 모든 선들에 대해서 교차점의 정보를 찾아서 저장하고, 임의의 시작점으로부터 주어진 방향을 따라

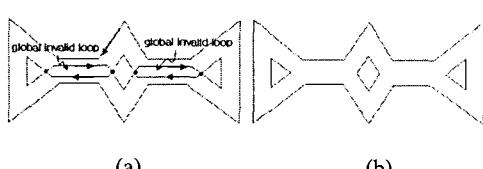


Fig. 14 Global invalid loops removal

교차점과 만날 때까지 선들을 방문한다. 그리고 교차점에서 앞에서 설명한 추적 법칙에 따라서 이동하는 방향을 변경한다. 모든 외곽경계들과 섬들을 방문할 때까지 위의 과정을 반복한다. 이렇게 찾아낸 모든 루프들은 각각을 독립적으로 다루고, 각 루프가 외곽경계가 될 것인지 섬이 될 것인지를 판단하는 방법은 다음과 같다. 우선 방문된 선들 중에서 외곽경계가 포함되어 있다면, 이 루프는 외곽경계가 되고 외곽경계가 하나도 포함되지 않은 경우는 섬이 된다. 그리고 섬은 반드시 외곽경계 안에 포함이 되므로 찾아진 섬이 어떤 외곽경계에 포함되는가에 대한 판단은 바운딩 박스(bounding box)를 이용한 포함관계로부터 구한다. 이렇게 병합하는 과정은 Fig. 15 에 나타나 있고, 섬들을 포함한 각각의 외곽경계들은 독립적인 입력 프로파일로 다음 오프셋을 구하는데 사용된다.

3. 예제 및 고찰

직선들로 이루어진 폐곡선을 오프셋 하는 전체적인 절차는 Fig. 16 과 같이 표현 할 수 있으며, 알고리듬들은 C++언어로 PC에서 구현되었다. Fig. 17 은 섬이 있는 경우와 없는 경우에 대한 오프셋 결과들을 보여준다. Fig. 18 은 삼각형 망 모델인 스피커로부터 얻은 입력 프로파일을 오프셋 한 결과를 나타낸다. Fig. 18(a)는 스피커 모델을 나타내고, Fig. 18(b)는 임의의 높이의 절단평면으로 가공 영역을 추출한 결과로 2069 개의 선분을 얻었다. 오프셋 거리(offset distance) 3mm로 오프셋 한 결과는 Fig. 18(c)에 보여진다.

지금까지 섬을 가지는 2 차원 직선들을 하나의 입력 프로파일로 정의하고 이에 적합한 새로운 오프셋 알고리듬을 제안했다. 이등분선을 이용하여 오프셋에 포함되지 않아야 하는 선들을 미리 찾아내서, 무효화 오프셋 선 처리 알고리듬을 적용하여 국부적 무효화 루프가 없는 원시 오프셋 선을 얻었다. 원시 오프셋 선을 얻는데 드는 시간 복잡도(time complexity)는 앞의 알고리듬 표현에서 볼 수 있듯이 모든 선분들이 이등분선을 이용해 오프셋 점을 계산할 때 오직 한번만 방문되고, 무효화 오프셋 선을 처리하는 데는 상수 시간 복잡도(constant time complexity)를 가지므로 거의 $O(n)$ 이다. 여기서, n 은 선분의 개수를 나타낸다. 이 과

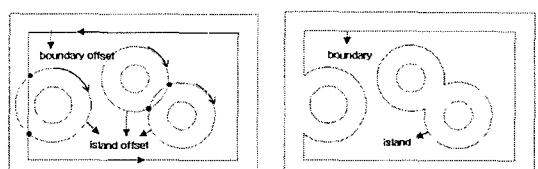


Fig. 15 Merging of a boundary and islands

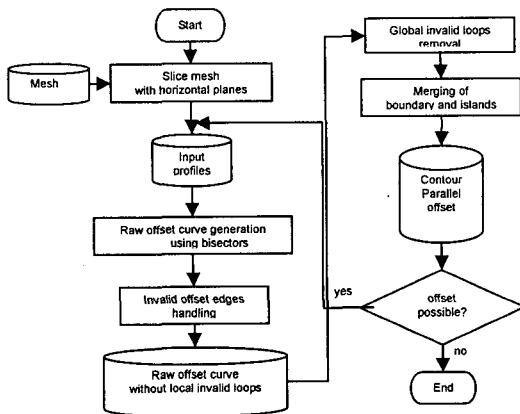


Fig. 16 Flow chart of a new offset approach

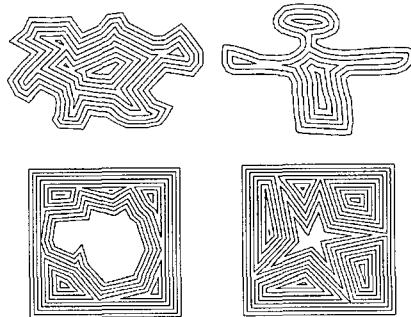


Fig. 17 Offset examples

정이 기존의 상호 교선 오프셋 방법이 가지고 있던, 국부적 무효화 루프들을 찾아내서 제거하는데 발생하는 많은 시간 소요 문제와 수치적인 불안정성 문제에 대한 해답을 준다.

제안된 루프 처리 알고리듬과 병합 알고리듬은 단지 모든 선들을 한번씩 방문하면 되므로, 원시 오프셋 선들 사이의 모든 교차점을 미리 찾을 수 있다면 $O(n)$ 의 시간 복잡도를 갖는다. 원시 오프셋 선들 사이의 교차점은 그리드 타입 헤시 테이블을 이용하여 구하며, $O(n \log n)$ 의 시간 복잡도를 갖는다.⁽¹³⁾

보로노이 다이아그램을 이용한 오프셋 방법은 상호 교선 방법에 비해 안정적이고 널리 사용되지만, 보로노이 다이아그램을 만드는 일이 쉽지 않으며 많은 시간을 요하고 앞에서 언급한 것처럼 짧은 직선들로 이루어진 곡선 부위에서 수치적인 불안정성을 갖는다. 이에 반해 제안된 새로운 오프셋 방법은 수치적으로 안정하며 간단하다. 더불어 새로운 오프셋 방법은 보로노이 다이아그램을 만드는 부수적인 작업 또한 필요하지 않다.

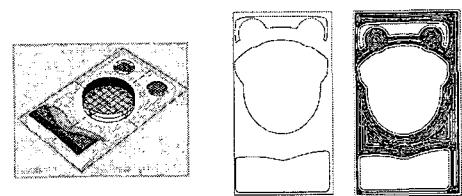


Fig. 18 Offset of a speaker

4. 결 론

본 논문에서는 기계가공에서 가장 많이 사용되는 윤곽 평행 공구경로 생성을 위한 핵심 기하 알고리듬인 2 차원 오프셋에 관한 새로운 오프셋 방법을 개발하여 공구경로 생성의 효율화를 꾀하였다. 다면체 모델로부터 얻어진 직선으로 이루어진 커브 오프셋에 효율적인 새로운 알고리듬을 개발하였는데, 이 알고리듬은 여러 개의 섬들을 포함하고 있는 경우라도 문제없이 오프셋을 할 수 있다. 새로운 오프셋 알고리듬의 가장 큰 특징은 이등분선과 무효화 오프셋 선 처리 알고리듬을 이용하여 국부적 꼬임을 완전히 제거함으로써, 초기 오프셋 후에 발생하는 자체 꼬임의 개수를 격감시킨다는데 있다. 이 특징은 기존 오프셋 알고리듬들의 가장 큰 문제인 원호 특이성을 극복하게 할뿐만 아니라 계산 속도의 향상을 가져온다. 뿐만 아니라 보로노이 다이아그램을 만드는 부수적인 작업이 필요가 없으며 빠르다. 마지막으로 새로운 오프셋 방법은 다양한 예제로부터 검증되었다.

참고문헌

- (1) Harenbrock, D., 1980, "The Connection of CAD and CAM by Means of the Program Package PROREN1/NC," Technical Report, Inst. fur Konstruktionstechnik, Ruhr-Universitat Bochum, Bochum, FRG
- (2) Jun, C. S., Kim, D. S. and Park, S. H., 2002, "A New Curve-Based Approach to Polyhedral Machining," Computer-Aided Design, Vol. 34, No. 5, pp. 379~389.
- (3) Bruckner, L. K., 1982, "Geometric Algorithms for 2.5D Roughing Process of Sculptured Surfaces," In Proc. Joint Anglo-Hungarian Seminar on Computer-Aided Geometric Design, Budapest, Hungary.
- (4) Hansen, A. and Arbab, F., 1992, "An Algorithm for Generating NC Tool Path for Arbitrary Shaped Pockets with Islands," ACM Transaction on Graphics, Vol. 11, No. 2, pp. 152~182.
- (5) Persson, H., 1978, "NC Machining of Arbitrary Shaped Pockets," Computer-Aided Design, Vol. 10, No. 3, pp. 169~174.

- (6) Held, M., Lukacs, G and Andor, L., 1994, "Pocket Machining Based on Contour-Parallel Tool Paths Generated by Means of Proximity Maps," *Computer-Aided Design*, Vol. 26, No. 3, pp. 189~203.
- (7) Jeong, J. and Kim, K., 1999, "Generating Tool Paths for Free-Form Pocket Machining Using z-Buffer-Based Voronoi Diagrams," *International Journal of Advanced Manufacturing Technology*, Vol. 15, No. 3, pp. 182~187.
- (8) Choi, B. K. and Kim, B. H., 1997, "Die-Cavity Pocketing via Cutting Simulation," *Computer-Aided Design*, Vol. 29, No. 12, pp. 837~846.
- (9) Jeong, J. and Kim, K., 1999, "Generation of Tool Paths for Machining Free-From Pockets with Islands Using Distance Maps," *International Journal of Advanced Manufacturing Technology*, Vol. 15, No. 5, pp. 311~316.
- (10) Lambregts, C., Delbressine, F., de Vries, W. and van der Wolf, A., 1996, "An Efficient Automatic Tool Path Generator for 2/2D Free-Form Pockets," *Computers in Industry*, Vol. 29, No. 3, pp. 151~157.
- (11) Kokichi, S., 1998, "Degeneracy and Instability in Geometric Computation," Proceedings of IFIP WG5.2 GEO-6 Conference in Torkyo University, pp. 5~15.
- (12) Mortenson, M. E., 1985, "Geometric Modeling," John Wiley & Sons.
- (13) Lee, D. Y., Kim, S. J., Lee, S. G and Yang, M. Y., 2003, "Incomplete Mesh Based Tool Path Generation," *Proceeding of the SMPE Spring Conference 2003*, pp. 844~847.