論 文
56-1-30

컴퓨터 패턴 재봉기에서의 오프셋 곡선 생성 방법

An offset Curve Generation Method for the Computer Pattern Sewing Machine

吳 台 錫* · 尹 成 容* · 金 一 煥†
(Tae-Seok Oh · Sung-Yong Yun · Il-Hwan Kim)

Abstract - In this paper we propose an efficient offset curve generation algorithm for open and closed 2D point sequence curve(PS curve) with line segments in the plane. One of the most difficult problems of offset generation is the loop intersection problem caused by the interference of offset curve segments. We propose an algorithm which removes global as well as local intersection loop without making an intermediate offset curve by forward tracing of tangential circle. Experiment in computer sewing machine shows that proposed method is very useful and simple.

Key Words : offset curve, interference, intersection, sewing machine

1. 서 론

오프셋 곡선은 기본 곡선 또는 형상의 외곽선으로부터 항상 일정한 거리를 유지하도록 기본 곡선의 곡선성질을 이용해 만든 곡선을 말한다. 오프셋 곡선을 만드는 것은 컴퓨터 그래픽, NC 머신을 위한 툴패스 생성, VLSI 회로설계, robot path planning 등 다양한 적용분야에 이용되는 중요한 작업이다[1-6].

오프셋 곡선은 NURBS나 Bezier 곡선과 같은 곡선에서 오프셋 곡선을 만드는 방법[1-2][9]과 점열곡선으로부터 오프셋 곡선을 만드는 방법[3]으로 크게 나눌 수 있다.

점열곡선(polygonal chain or curve)으로부터 오프셋 곡선을 만드는 방법(점열곡선 오프셋)에는 pair-wise 오프셋, Voronoi diagram, 그리고 픽셀을 기반으로 접근한 방법 등이 있다[3][6].

곡선 오프셋이나 점열곡선 오프셋 모두 자기교차(self-interception)에 의한 국부(local)와 전역(global) 자기교차 루프 문제가 발생하고 이 두 가지 문제를 해결하기 위한 방법이 연구되어 왔다. 자기교차 문제란 기본곡선에서 일정한 오프셋 거리를 유지하면서 오프셋 곡선을 만드는 중에 오프셋 거리를 유지할 수 없는 부분에서 오프셋 곡선이 교차하게 되고 이로 인하여 잘못된 오프셋 곡선이 만들어지는 문제를 말한다. 이 문제의 해결 방법으로는 기본적인 오프셋 곡선을 먼저 만든 후에 기하학적인 루프테스트 알고리즘으로 제거하는 방법과 PWID(pair-wise interference detect) 알고리즘에 의하여 간섭이 있느냐 없느냐를 판단하여 기본적인 오프셋 곡선(raw offset curve)을 그리기 전에 먼저 제거하고 오프셋을 만드는 방법 등이 연구되어 왔다[1-3][7][8]. 기본 오프셋곡선을 만든 후 기하학적인 특성에 따라서 자기교차를 제거하는 경우에는 기본곡선과 기본 오프셋 곡선의 진행 방향에 따라 자기교차를 판단하게 되는데 기본곡선의 모양이 복잡해질수록 판단이 용이하지 않을 뿐만 아니라 잘못된 판단으로 오프셋 곡선이 제대로 만들어지지 않는 경우가 발생할 수 있다. 또한 점열곡선의 자기교차를 판단하는 sweep 이론[10]은 알고리즘이 복잡하기 때문에 오프셋 곡선을 만드는 과정보다 자기교차를 판단하는 과정이 더 복잡해지는 단점을 가지고 있다. 한편, 컴퓨터 패턴 재봉기의 곡선 편집에서 모든 곡선은 LS(line segment)가 순차적으로 연결된 곡선으로 생각할 수 있다. 재봉기의 곡선은 일반 공작기기와 달리 재봉간격(stitch)이 있기 때문에 최소의 재봉 간격으로 구성된 LS들의 연결로 구성되었다고 할 수 있다. 재봉기의 재봉간격은 1-3mm를 가장 많이 사용하고 재봉기의 이송간격 최소 단위는 일반적으로 0.1mm이다. 공작기기에서 가공 정밀도를 높이기 위해서는 많은 데이터들로 오프셋 곡선을 만들어야 하지만 재봉기에서는 마이크로프로세서에서 편집 가능한 최소의 데이터로 최소의 재봉 간격만큼의 정밀도만 유지하면 오프셋 곡선 재봉시 재봉의 질에 영향을 미치지 않는다. 따라서 본 논문에서는 LS로 연결된 열린 곡선과 닫힌 곡선에 모두 적용될 수 있고, 자기교차 문제를 보다 쉽게 해결하는 방법으로 접원의 전방향 이동방법에 의한 오프셋(forward tracing of tangential circle) 곡선 생성 방법을 제안하였다. 또한 제안된 방법은 컴퓨터 재봉기에 적용하여 그 유효성을 검증하였다.

2. 컴퓨터 패턴 재봉기에서 오프셋 곡선의 필요성

그림 1에서와 같이 육각형의 라벨을 천 또는 가죽에 부착하는 재봉을 할 경우 부착물의 외곽선을 따라 기본 패턴을 생성한 후 이를 바탕으로 그림의 오프셋 재봉 곡선과 같은 재봉 패턴을 만들 필요가 있다. 컴퓨터 패턴 재봉기에서 이러한 편집 작업은 패턴 재봉기 자체에서 편집하거나 PC에

* 學生會員 : 江原大學交 電子通信工學科
† 교신저자, 正會員 : 江原大學交 電氣電子工學部 敎授 · 工博
E-mail : ihkim@kangwon.ac.kr
接受日字 : 2006年 9月 15日
最終完了 : 2006年 11月 10日

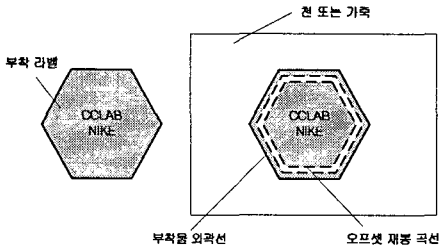


그림 1 육각형 라벨 부착
Fig. 1 hexagon label sewing

서 구동되는 CAD 또는 편집 프로그램을 이용하여 편집한다.

본 논문의 연구는 마이크로 프로세서로 구성된 컴퓨터 패턴 재봉기 자체에서 편집할 수 있도록 프로그램 용량이 작고 또한 수행시간도 빠르게 구현할 수 있는 방법을 제안한다.

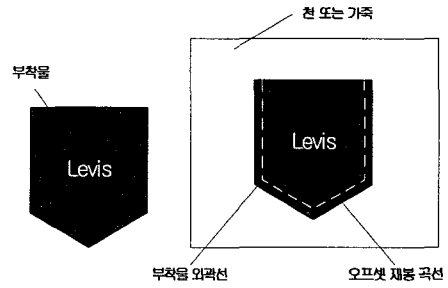


그림 2 뒷주머니 부착
Fig. 2 Hip Pocket sewing

그림 2는 패턴 재봉기에서 청바지 뒷주머니를 달기위한 오프셋 패턴 생성 방법 및 필요성을 나타낸다. 주머니의 외곽선을 따라 기본 곡선을 만들고 이를 기준으로 오프셋 패턴을 생성한다. 그리고 생성된 오프셋 패턴을 따라 재봉을 하여 주머니를 부착시킨다. 이와 같이 패턴 재봉기에서 오프셋 곡선 패턴은 여러 모양의 재봉 패턴을 손쉽게 편집할 수 있도록 하는 중요한 도구이다.

3. 기존의 오프셋 곡선을 만드는 방법

평면상의 곡선(r(t))의 오프셋은 두 개의 파라미터 s, t가 $s \neq t$ 일 때 식(1)인 곳에서 발생한다.

$$r(s)+dn(s)=r(t)+dn(t) \tag{1}$$

여기서 $0 \le s \le 1$, $0 \le t \le 1$이다. 여기서 n(t)는 r(t)의 unit normal vector, d는 offset 거리이다.

그림 3에서 자기교차가 1개가 있으면서 부적당한 루프를 갖는 부분을 국부 부적당 루프(local invalid loop)라 하고 자기교차가 2개 있으면서 부적당한 루프를 갖는 부분을 전역 부적당 루프(global invalid loop)라 한다. 따라서 이러한 부적당 루프를 제거해야 적당한(valid) 오프셋 곡선을 만들 수 있다. 그림 3은 오프셋 곡선을 만들 때 자기 교차 문제로 인하여 발생하는 부적당 루프의 생성을 보여주고 그림 4는 원하는 오프셋 곡선을 보여준다.

이러한 문제 해결 방법으로 B.K Choi[3] 등은 PWID (pair-wise interference detection)라는 테스트 방법을 통하여 간섭이 있는지 없는지를 판별하여 부적당한 루프를 기본 오프셋 곡선에 나타나지 않도록 연구하였다. Eungki Lee[6]는 어떠한 테스트도 하지 않고 기본 오프셋 곡선을 먼저 만든 후에 오프셋 곡선의 진행 방향과 루프의 생성 방향이 일치하지 않는 루프를 제거하는 방법으로 부적당한 루프를 제거하는 방법을 제안하였다.

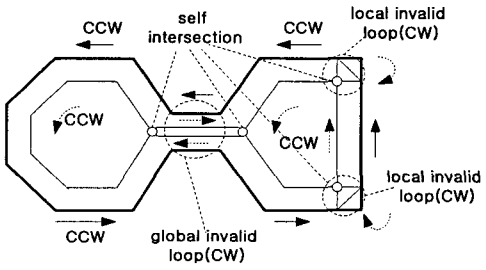


그림 3 자기교차 루프의 문제
Fig. 3 problem of self intersection

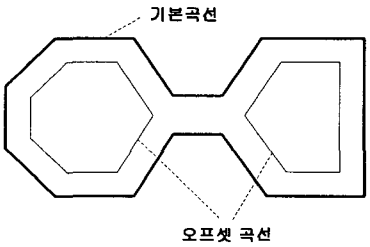


그림 4 적당한 오프셋 곡선
Fig. 4 Valid offset curve

그림 3에서 진한색의 선으로 기본곡선을 나타내었고 CCW 방향으로 오프셋 곡선을 만드는 방법을 보여준다. 기본곡선의 안쪽으로 일정한 오프셋 거리를 두고 오프셋 곡선을 만들다보면 그림 3에서처럼 네 번의 자기교차(self intersection)가 발생한다. 앞에서 설명한 것처럼 가운데 전역 부적당 루프가 생기고 오른쪽에 국부 부적당 루프가 두 개 생긴다. 이렇게 생긴 루프는 오프셋 곡선의 방향과 루프의 방향이 반대가 되고 이러한 루프의 방향을 검출하면 부적당 루프를 제거할 수 있다. 이렇게 부적당 루프를 제거하면 적당한 오프셋 곡선을 그림 4와 같이 구할 수 있다[1-6]. 그림 3에서와 같이 기본 오프셋 곡선을 만든 후에 자기 교차점을 찾는 방법으로 sweep 이론이 있다. 하지만 이러한 알고리즘들은 검증이 되어 있으나 복잡하여 오프셋 곡선을 만드는 것보다 부적당 루프를 찾는 방법이 더 복잡하게 되어 재봉기에서와 같이 제한된 연산 구조에서는 사용하기에 부적합하다.

이에 대한 해결방안으로 본 논문에서는 기본 오프셋 곡선을 만든 후에 부적당 루프를 제거하는 방법으로 오프셋곡선을 만드는 것이 아니라 접원을 이용한 접원의 전 방향 경로 이동으로 오프셋 곡선을 만들도록 하여 국부 및 전역 자기교차 문제가 기본 오프셋 곡선을 만드는 중에 발생하지 않도록 하였다.

4. 제안된 방법의 오프셋 곡선 방법에 관한 내용 및 정의

4.1 열린 곡선 및 닫힌 곡선의 오프셋 곡선

4.1.1 열린 곡선에서의 오프셋 곡선

그림 5에서는 열린 곡선에서 접원을 이용한 오프셋 곡선을 만드는 방법을 보여준다. 오프셋 거리가 접원의 반지름과 같고 이 접원을 시작점부터 끝점까지 이동하면서 오프셋 점을 구하고 이 점들을 연결하여 오프셋 곡선을 그린다. 이때 접원에 간섭이 생기는 접원의 중심 점은 무시하고 간섭이 생기지 않는 접원의 중심점을 오프셋 점으로 표시하고 이렇게 생긴 오프셋 점을 연결하여 오프셋 곡선을 만든다. 그림 5 (a)처럼 CCW 방향으로 오프셋을 만드는 경우와 그림 5 (b)처럼 CW 방향으로 만드는 경우에 항상 같은 오프셋이 생성된다.

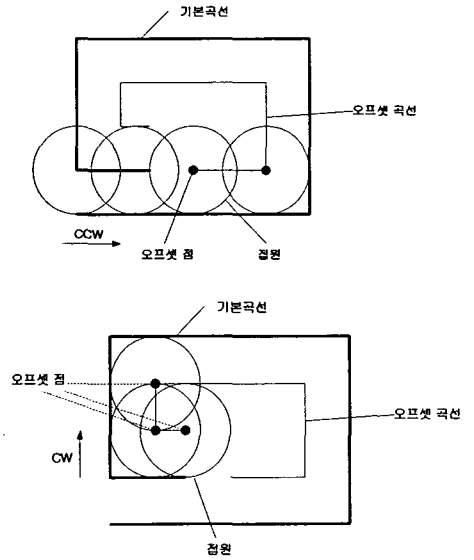


그림 5 열린 곡선의 오프셋 생성
Fig. 5 offset creation of open-curve

4.1.2 닫힌 곡선에서의 오프셋 곡선

그림 6은 닫힌 곡선에서의 오프셋 곡선 생성 방법을 나타낸다. 열린 곡선에서와 마찬가지로 간섭이 없는 접원의 중심점을 오프셋 점으로 구하고 이 점들을 연결하여 오프셋 곡선을 만든다. 그림 6에서는 다각형에 대한 설명을 하였고 LS로 연결된 곡선은 다각형으로 표현될 수 있기 때문에 점열 곡선(PS curve)인 경우는 모두 적용될 수 있다.

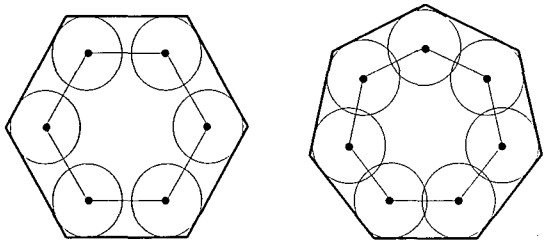


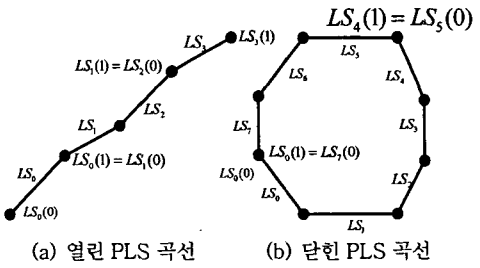
그림 6 닫힌 곡선의 오프셋 생성
Fig. 6 offset creation of closed curve

4.2 용어 및 정의

4.2.1 Poly-Line segment Sequence curve(PLS 곡선)

PLS 곡선은 평면상에서 LS(Line Segment)와 LS가 순차적으로 연결되어 있는 곡선으로 시작되는 시점과 끝나는 종점이 주어져야 한다. 즉, PLS 곡선에 속하는 n개의 연속적인 LS는 연속적인 모든 LS_i와 LS_{i+1}에서 LS_i의 종점과 LS_{i+1}의 시점은 일치한다. 열린 곡선인 경우에는 LS_0의 시점과 LS_{n-1}의 종점만 다르다.

그림 7은 열린(open) 또는 닫힌(closed) PLS 곡선의 예를 나타내었다. 그림 7 (a)는 열린 PLS 곡선이며 5개의 점과 4개의 LS로 되어있고, 그림 7 (b)는 닫힌 PLS 곡선이며 8개의 점과 8개의 LS로 되어있다.



(a) 열린 PLS 곡선 (b) 닫힌 PLS 곡선

그림 7 PLS 곡선
Fig. 7 PLS curve

4.2.2 오프셋 방향

닫힌 PLS 곡선의 오프셋은 원래의 PLS 곡선에 대해 안쪽(inner)과 바깥쪽(outer) 오프셋으로 나누어진다. 본 논문에서는 그림 8과 같이 PLS 곡선의 오프셋 방향을 안쪽을 오른쪽, 바깥쪽을 왼쪽으로 하여 설명되어진다.

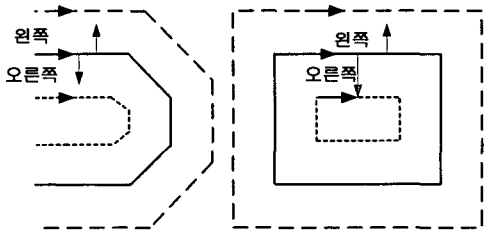
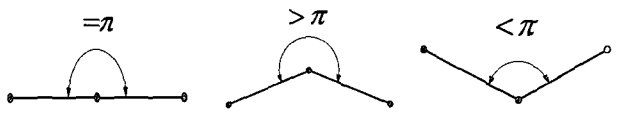


그림 8 오프셋 방향
Fig. 8 direction of offset

4.2.3 연결점의 종류(Vertex)

선분과 선분이 연결되는 각도에 따라 3가지의 용어를 사용하여 구별한다. 왼쪽 방향에 대해 두선분이 연결되는 교점이 이루는 각이 π인 경우는 직선점(colinear vertex), π보다 작은 경우는 오목점(concave vertex), π보다 큰 경우는 볼록점(convex vertex)이다. 그림 9에 3가지의 경우에 대한 그림을 나타내었다.



(a) 직선점 (b) 볼록점 (c) 오목점

그림 9 연결점의 종류

Fig. 9 kind of connection point

4.2.4 접원(Tangential circle)

곡선에 대한 오프셋 곡선은 다음과 같이 정의된다.

$$C_0(t) = C(t) \pm wN(t) \tag{2}$$

여기서 w는 오프셋 폭 또는 거리이고, +w는 왼쪽오프셋, -w는 오른쪽오프셋이 된다.

N(t)는 기본곡선 C(t)의 unit normal로 아래와 같다.

$$N(t) = \frac{(y'(t), x'(t))}{\sqrt{x'(t)^2 + y'(t)^2}} \tag{3}$$

반지름 w인 원이 LS 위에 있는 한점에서 접하는 원을 접원이라고 하고 둘 이상의 LS와 접하는 원을 공통접원이라고 한다. LS에 접하는 접원의 중심은 오프셋 곡선식과 선분방정식에서 구할 수 있다. 평면상의 두 점 $A(x_1, y_1)$, $B(x_2, y_2)$를 지나는 선분을 $LS(t) = (x(t), y(t))$라 할 때 매개변수 t를 $0 \le t \le 1$이라고 하면, 식(4)로 나타낼 수 있다.

$$\begin{aligned} x(t) &= (1-t)x_1 + tx_2 \\ y(t) &= (1-t)y_1 + ty_2 \\ x'(t) &= (x_1 + (x_2 - x_1)t)' = x_2 - x_1 \\ y'(t) &= (y_1 + (y_2 - y_1)t)' = y_2 - y_1 \\ N(t) &= \frac{(y_2 - y_1, x_1 - x_2)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ C(t) &= LS(t) \pm d \cdot N(t) \end{aligned} \tag{4}$$

4.3 연결점에 따른 오프셋 점

오프셋 곡선은 기본곡선과 오프셋 거리를 유지하면서 존재해야 한다. 따라서 모든 기본곡선의 PLS와 OPLS(offset PLS)는 오프셋 거리를 유지해야한다. 주어진 PLS에서 접원을 이동해서 얻은 접원의 중심점 경로가 OPLS가 되는 것이다. LS는 양끝점으로 구성되므로 OPLS는 PLS와 마찬가지로 LS의 양끝점으로 구성된다. n개의 PLS 곡선이 주어졌을 때 시작점 $LS_0(0)$로부터 접원을 이동해서 m개의 OPLS를 만들게 된다. 그림 10 (a)는 열린 PLS에서 시점 $LS_0(0)$에 접하는 원의 중심을 구하고 그 중심을 OPLS 점으로 저장하게 된다. 종점 또한 마찬가지의 방법으로 $LS_{n-1}(1)$에서 접원의 중점을 구해 OPLS 점으로 만들면 된다.

인접한 PLS간의 오프셋 방향에 대한 사이각이 π인 직선점에서는 그림 11처럼 양끝점만 알면 된다는 직선의 성질에 따라 OPLS를 하나만 만든다.

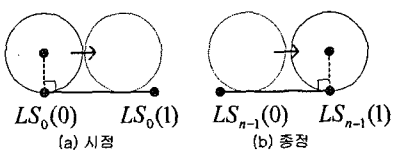


그림 10 시점과 종점

Fig. 10 start point and end point

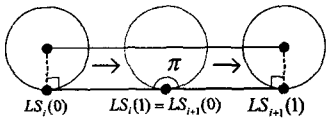


그림 11 직선점의 오프셋

Fig. 11 offset of line point

그림 12의 볼록점에서는 두개의 인접한 PLS에 대해 OPLS를 만든 후 두개의 OPLS가 만나는 점을 새로운 OPLS의 한쪽 점으로 설정한다. NC 머신의 pocketing과 다른 점인데 이런 교점에서 NC pocketing에서는 호로 설정하지만 여기서는 LS로 설정한다[5].

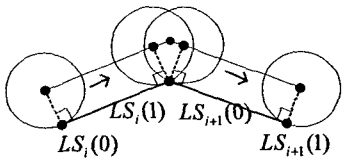


그림 12 볼록점의 오프셋

Fig. 12 offset of convex

그림 13의 오목점에서는 LS_i와 LS_{i+1}의 공통접원을 구하고 공통접원의 중점을 OPLS의 한쪽 끝점으로 설정한다.

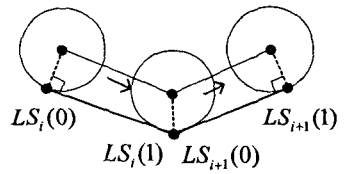


그림 13 오목점의 오프셋

Fig. 13 offset of concave

가지점(branch offset point)은 그림 14에 보인 것처럼 두 가지 있는 데 그 첫 번째로 그림 14 (a)는 두 개의 LS사이에 존재하고 두 번째는 한 LS와 다른 LS의 볼록점이 만날 때 발생한다. 이런 가지점에서는 가지점을 이루는 두 LS 사이에 존재하는 LS를 검사하여 오프셋점이 존재하는 지를 판단 한 후 오프셋점이 존재하면 그 가지점을 저장하고, LS의 큰 쪽을 먼저 경로이동하여 OPLS 점을 만든 후 더 이상 OPLS 점을 만들 수 없을 때 즉 예를 들어 종점을 만나면 저장된 가지점을 가져와 다시 경로이동으로 OPLS 점을 만

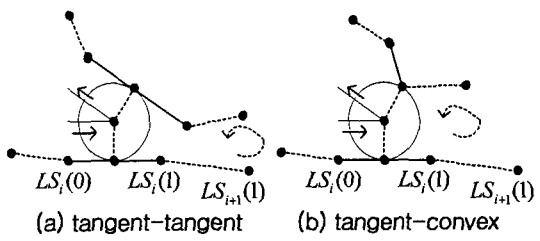


그림 14 가지점의 오프셋

Fig. 14 offset of branch

들게 된다. 그리고 오프셋점이 존재하지 않는 경우는 그 부분을 플래그로 표시하고 다시 경로이동을 하지 않도록 한다. 이런 영역은 국부자기교차가 일어나는 부분으로 국부자기교차가 자동으로 삭제되게 된다.

5. 접원의 전방향 경로이동에 의한 오프셋 생성 방법

기존의 오프셋 곡선을 만드는 방법들은 고정밀도의 CAD 시스템에서 사용하기 위한 방법으로서 NC머신 등과 같은 공작기기의 가공정밀도를 위해서는 많은 데이터들로 구성되어진다. 이러한 알고리즘을 마이크로프로세서 기반인 컴퓨터 재봉기의 알고리즘으로 사용하기에는 제약이 있다.

컴퓨터 재봉기는 재봉 간격이 있기 때문에 고정밀도의 많은 데이터를 요구하는 것이 아니고 재봉의 질에 영향을 주지 않는 정밀도의 범위에서 알고리즘이 고속으로 구현될 수 있도록 단순화 될 필요가 있다.

본 논문에서 제안한 방법으로 열린 PLS와 닫힌 PLS에 대해 OPLS를 만든 예가 그림 15와 그림 16에 나타낸다.

PLS 곡선의 시점에서부터 접원의 오프셋 경로이동을 시작해서 모든 LS에 대해 오프셋 경로이동을 하게 되면 OPLS(offset PLS)가 만들어 진다. PLS는 n개가 주어지고 각 LS는 순차적으로 주어진다고 보고, 접원의 경로이동을 하면서 OPLS을 만든다. 접원의 경로이동 중에 가지 구간을 만나면 n이 현재 값보다 큰 쪽으로 이동하고, 두 개의 갈래가 나오는 구간에서는 두 갈래 중 n이 큰 쪽으로 이동 한다 이렇게 큰 쪽으로 이동하면서 전진 경로 이동을 하게 되면 앞서 말한 전역 자기교차가 발생하지 않고 OPLS가 만들어지게 된다.

기존에 제안된 방법에서와는 달리 본 논문에서의 방법은 오프셋 경로이동시에 국부자기교차와 전역자기교차들이 동시에 제거되므로 기본 오프셋곡선을 만든 후에 자기교차를 검사하는 루틴들이 필요 없어진다. 또한 블록점에서의 오프셋 처리를 단순화하였다. 그림 17은 본 논문에서 제안한 접원의 전방향 경로이동에 의한 오프셋 생성 방법의 순서도를 나타낸다.

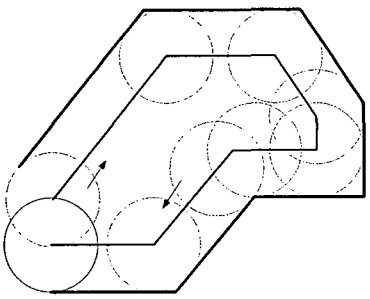


그림 15 열린 PLS 곡선의 오프셋
Fig. 15 offset of open-PLS curve

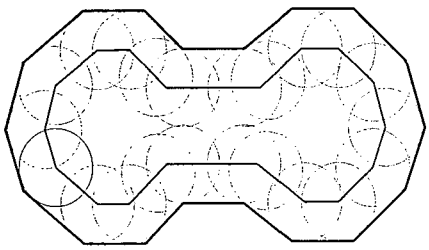


그림 16 닫힌 PLS 곡선의 오프셋
Fig. 16 offset of closed-PLS curve

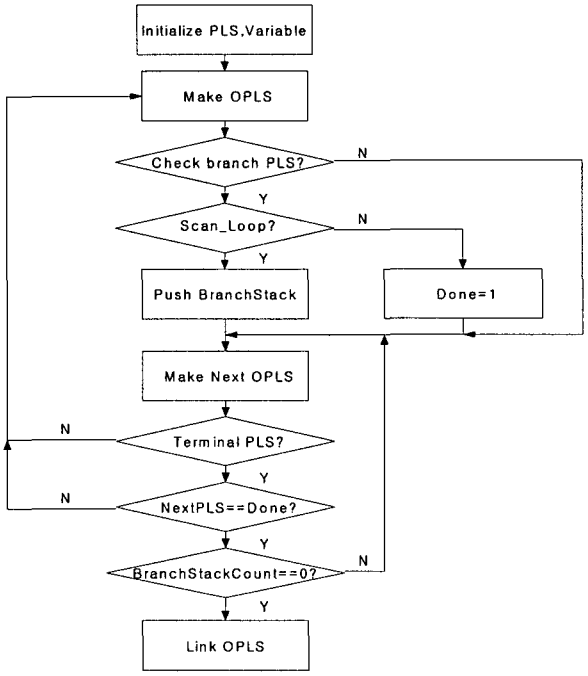


그림 17 접원을 이용한 오프셋 곡선 생성 방법 순서도
Fig. 17 flow chart of offset creation method using tangential circle

-*PLS* : PLS는 배열 형식으로 n개의 순차적으로 연결된 LS가 저장되고 아래와 같은 변수들이 사용된다.
-*Done* : Done 플래그는 오프셋 생성과 관련된 경로 이동을 이미 한 경우에는 1(True), 아닌 경우에는 0(False)가 된다.
-t : 경로 이동한 t값($0.0 \le t \le 1.0$)
-*dir* : 오프셋 방향 (+ or -)
-w : 오프셋 폭 또는 오프셋 거리.
-τ : 톨러런스로 경로이동 단위.
-*OPLS* : 오프셋 PLS.
-*CurPLS* : 현재 경로이동 중인 PLS.
-*NextPLS* : 다음 이동할 PLS.
-*BranchPLS* : 가지구간에서 만나는 PLS.
-*BranchStack* : 가지구간에서 가지 오프셋 점을 만드는 PLS를 저장하는 스택.

최초 초기화 부분은 모든 변수 및 조건을 초기화한다. Done, OPLS, BranchStack, CountStack등의 변수들을 초기화 한다. OPLS점을 만들기 위하여 PLS_0인 세그먼트의 시작점부터 t=0.0에서부터 1.0까지 톨러런스 τ만큼 증가 시키면서 오프셋 점이 생성될 때까지 증가 시키고 오프셋 점을 생성할 수 없을 때는 현재의 PLS의 Done 플래그를 1로 설정(Set)한다. 이렇게 Done 플래그를 1로 하면 다시 오프셋 점을 찾지 않게 된다. 그리고 다시 다음 PLS를 가져와서 위의

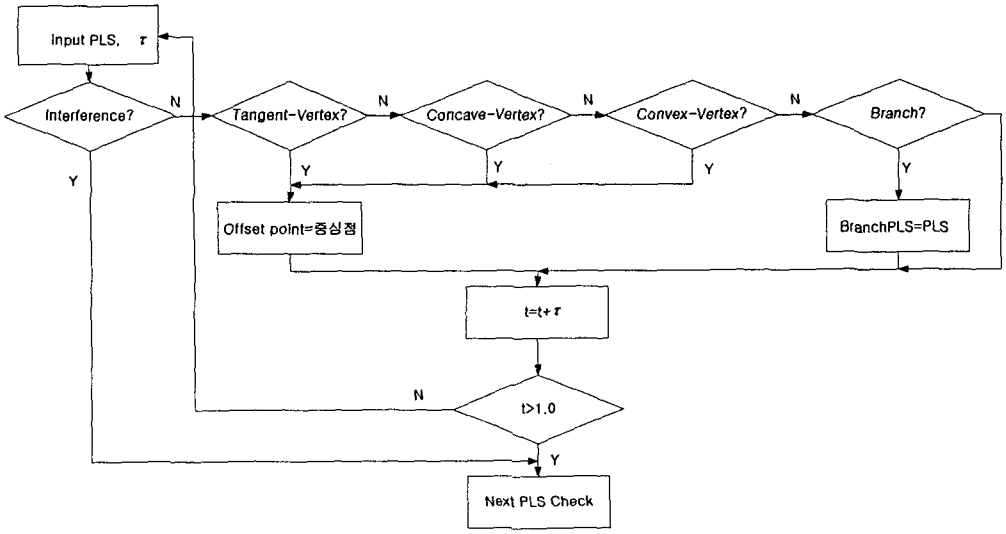
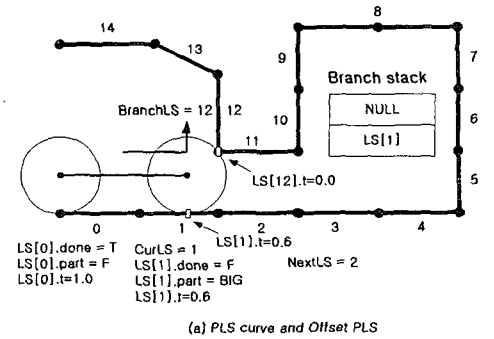
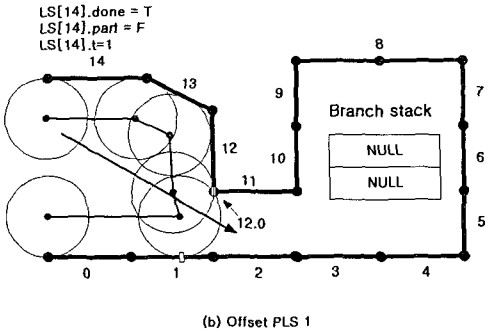


그림 18 오프셋 점 찾는 방법의 순서도

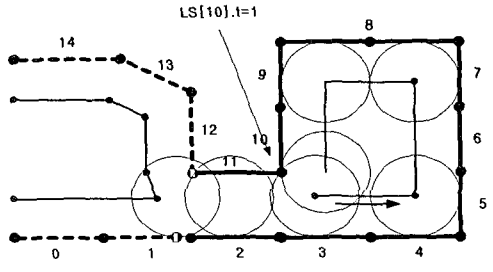
Fig. 18 flow chart of method to find offset point



(a) PLS curve and Offset PLS



(b) Offset PLS 1



(c) Offset PLS 2

그림 19 접원의 전방향 경로이동에 의한 오프셋 곡선

Fig. 19 offset curve using forward tracing of tangential circle

작업을 반복한다. 오프셋 점을 찾는 방법을 다시 자세히 설명하면, 간섭이 있으면 오프셋 점이 될 수 없고 간섭이 없는 공통 접원의 경우 그림 14와 같은 경우에 가지점과 오프셋 점으로 표시하고 가지점이 아닌 경우는 오프셋 점으로만 선택한다. 그림 18에 오프셋 점을 찾는 방법의 순서도를 보여준다.

그림 19 (a)는 *CurPLS=0*에서 시작하여 PLS[0]와 PLS[1]에 대한 오프셋 점이 만들어지는 과정이다.

그림 19 (b)에서 가지점으로 판단되면 BranchStack에 다음 PLS(NextPLS)를 저장을 한다. 그림 19 (a)를 보면 CurPLS=1지점이 가지 구간이고 NextPLS=2이다. 접원에 접한 BranchPLS=12가 된다. 따라서 큰 쪽인 12를 다음 NextPLS로 설정한다. 따라서 BranchStack에는 2가 저장되고 PLS_{12}로 이동하여 오프셋 점을 찾고 검색이 다되면 BranchStack에 저장된 2를 가져와 LS_2에서 오프셋 점 검색을 다시 시작한다. 가지점이 판단되면 NextPLS까지 오프셋 점이 생성될 수 있는지를 계속 판단하게 되는데 오프셋 점이 생성될 수 없는 점들은 무시하게 되고 생성될 수 있는 PLS는 BranchStack에 저장하여 나중에 다시 검색하게 된다. 그림 19 (a)에서 2번째 접원의 중심점은 가지점으로 tangent-convex 오프셋 점이다. 남은 가지 점이 있는 지의 여부는 BranchStackCnt가 0인지 아닌지를 검사하게 되는데 0이면 BranchStack에 저장되어 있는 가지 점이 없으므로 더 이상 검색할 PLS가 없는 것이고 마지막 단계인 그림 19 (c)와 같이 오프셋 점들을 순서대로 연결하여 오프셋 곡선을 만든다.

6. 모의실험 결과

본 논문에서 제안한 오프셋 곡선 생성방법에 대한 모의실험을 여러 가지 열린 또는 닫힌 PLS 곡선에 대하여 수행하였다. 그림 20은 닫힌 PLS 곡선에 대해 CCW 방향에 대해 왼쪽(안쪽)과 오른쪽(바깥쪽) 방향으로 제안된 방법에 의하여 만든 오프셋 곡선이다. 주어진 기본 PLS 곡선은 시점과 종

점이 연결되어 있고 시점과 종점은 가장 왼쪽 모서리로 CCW 방향으로 연결되어있다. 105개의 LS와 104개 연결점으로 구성되어 있고, 이중 오목점 12개, 볼록점 12개 직선점 80개이다. PLS 곡선의 왼쪽방향(안쪽)에 대해 오프셋 거리를 5.0mm, 톨러런스(τ)를 0.001mm로 설정하여 생성하였다.

이렇게 하여 가장 먼저 만들어진 OPLS는 그림 20에서 가장 처음 넓어 보이는 OPLS로 직선점 0개, 오목점 12개, 볼록점 12개로 구성된 오프셋 곡선이 생성된다. 네 번째로 만드는 OPLS는 tangent-covex 관계인 가지점 4개를 가지고 있어서 4개의 섬(island)으로 만들어진다. 안쪽 오프셋을 만든 후 PLS 곡선의 오른쪽방향(바깥쪽)에 대해 오프셋 거리를 3.0mm, 톨러런스(τ)를 0.001mm로 설정하여 오프셋 곡선을 만들었다. 그림 20에서 가장 처음 좁아 보이는 오프셋 곡선으로 직선점 0개, 오목점 12개, 볼록점 12개로 구성된 오프셋 곡선이 만들어진다.

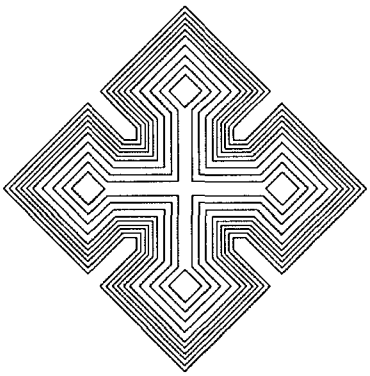


그림 20 닫힌 PLS와 OPLS
Fig. 20 closed PLS and OPLS

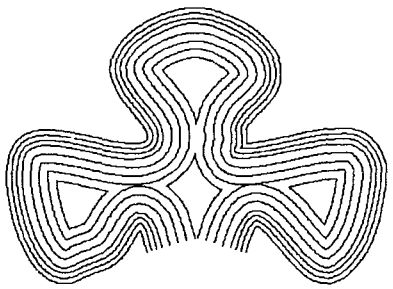


그림 21 열린 PLS와 OPLS
Fig. 21 open PLS and OPLS

그림 21은 열린 PLS 곡선에 대하여 만들어진 오프셋 곡선이다. 215개의 LS와 216개 연결점으로 구성되어 있고, 이중 오목점 99개, 볼록점 85개, 직선점 32개이다. 닫힌 곡선에서와 마찬가지로 왼쪽방향(안쪽)에 대해 오프셋 거리를 5.0mm, 톨러런스(τ)를 0.001mm로 설정하여 생성하였다. 이렇게 하여 가장 먼저 만들어진 OPLS는 그림 21에서 가장 처음 넓어 보이는 OPLS로 직선점 0개, 오목점 99개, 볼록점 85개로 구성된 오프셋 곡선이 생성된다. 열린 PLS 곡선의 오른쪽방향(바깥쪽)에 대해 오프셋 거리를 3.0mm로 하여 생성한 오프셋 곡선은 그림 21에서 가장 처음을 좁아 보이는 OPLS로 직선점 0개, 오목점 122개, 볼록점 81개로 구성되어진다.

7. 컴퓨터 재봉기에서의 적용 실험

패턴 재봉기는 PC에서 Auto CAD로 디자인된 재봉 패턴을 시리얼 통신포트를 통하여 재봉기 컨트롤러로 다운로드 받을 수도 있고 재봉기의 컨트롤러에서 직접 편집을 할 수도 있다. 실험은 DSP로 구현된 재봉기의 메인컨트롤러에서 본 논문에서 제안한 방법으로 오프셋 곡선을 편집하고 이 패턴에 따라 재봉을 할 수 있도록 하였다.

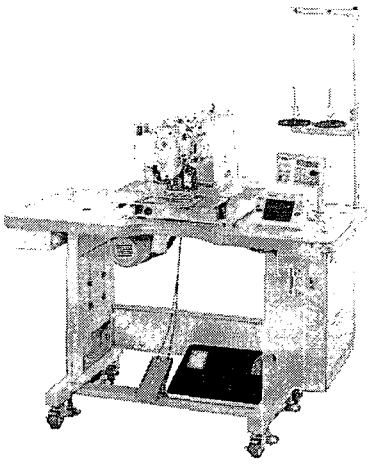


그림 22 컴퓨터 패턴 재봉기
Fig. 22 computer pattern sewing machine

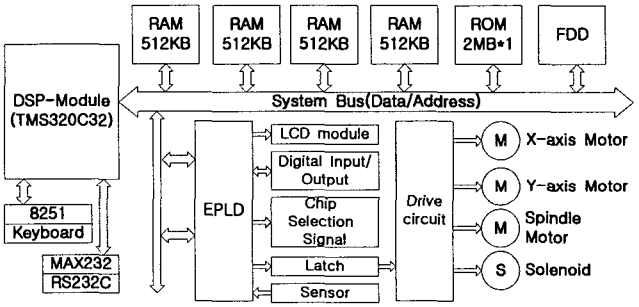


그림 23 컴퓨터 패턴 재봉기의 시스템 구성도
Fig. 23 system structure of computer sewing machine

패턴 재봉기는 기구부와 전장부로 크게 나눌 수 있고 기구부는 스핀들이 1회전 하면 한 땀이 재봉되도록 만든 재봉 기구와 패턴에 따라 X-Y축으로 움직이도록 만들어진 이송부로 나누어진다. 전장부는 스핀들 제어, 이송부제어, 편집기, 기타 주변기기 제어 등을 할 수 있도록 구성되어 있다. 본 논문은 전장부의 편집기 부분에 대한 연구로서 DSP320C32를 기반으로 구성된 컴퓨터 시스템에서 재봉 패턴 편집을 쉽고 용이하도록 하는데 목적이 있다. 그림 22는 전체적인 재봉기의 모습이다. 그림 23은 제어기의 블록선도를 나타낸다. 그림 24는 제안된 방법으로 편집기에서 편집된 오프셋 곡선 패턴을 보여준다. 그림 24는 재봉기에 붙은 320*240의 해상도를 갖는 LCD화면에 나타낸 패턴으로 해상도 때문에 약간 원 또는 호 부분에서 찌그러짐을 볼 수 있다. 그림 25는 그림 24에서 편집된 패턴을 컴퓨터 재봉기

에서 재봉한 재봉 결과물을 보여준다. 재봉 간격은 2.5mm, 오프셋 거리는 U자 모양은 3mm, 사각형 모양은 2mm이다. 톨러런스는 0.01mm로 설정하였다.

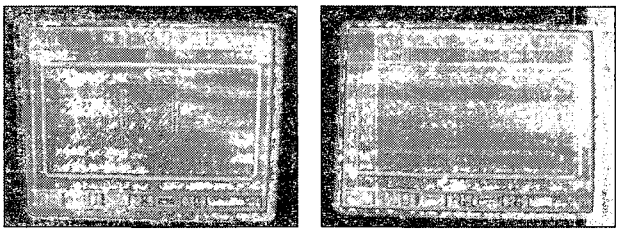


그림 24 컨트롤러 보드에서 편집된 오프셋 곡선의 화면
Fig. 24 display of edited offset curve in controller board

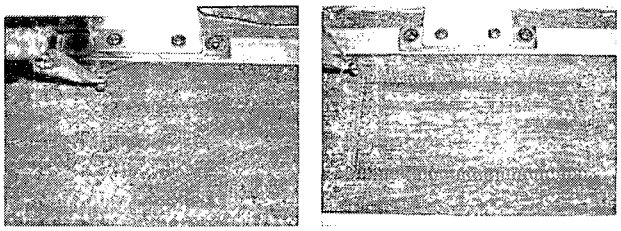


그림 25 편집된 오프셋 곡선의 실제 재봉된 모습
Fig. 25 sewn shape of edited offset curve

8. 결 론

본 논문에서 제안한 접원의 전방향 이동방법에 의한 오프셋 곡선 생성 방법은 열린곡선과 닫힌곡선에 모두 적용될 수 있는 방법이다. 기존에 제안된 방법은 주로 공작기계 및 전문 CAD시스템에 적용할 수 있는 방법으로 프로그램 및 방법이 복잡하여 마이크로 프로세서 기반의 패턴 재봉기의 편집 방법으로 사용하기에는 부적합하였다. 오프셋 생성에서 문제가 되는 자기교차 문제를 접원을 이용한 방법으로 쉽게 해결할 수 있음을 실험을 통해 보여 주었다. 또한 볼록점에서의 오프셋 점을 NC머신에서는 호(arc)로 처리하지만 본 논문에서는 접원의 교차점으로 처리하여 방법을 단순화하였다. 실제 패턴 재봉기 시스템에 적용하여 그 유효성을 검증하였다.

본 논문에서 구현한 접원의 전방향 이동방법은 주어진 PLS 곡선이 LS가 겹치지 않는 단순곡선(simple curve)이어야 하고 어떠한 교점도 반복되지 않는 곡선에 제한을 두고 있으므로 이런 곡선에 대한 연구가 앞으로의 과제이다.

감사의 글

이 논문은 강원대학교 BK21사업 및 정보통신연구소 지원에 의하여 이루어진 연구로서, 관계부처에 감사를 드립니다.

참 고 문 헌

[1] V.D.Holla, K.G.Shastry, B.G.Prakash, "Offset of curves on tessellated surfaces", Computer-Aided Design, Vol. 35, Issue 12, pp. 1099-1108, 2003

[2] G.V.V. Ravi Kumar, K.G. Shastry, B.G. Prakash, "Computing constant offsets of a NURBS B-Rep", Computer-Aided Design, Vol. 35, Issue 10, pp. 935 - 944, 2003

[3] B.K. Choi and S.C. Park, "A pair-wise offset algorithm for 2D point-sequence curve", Computer -Aided Design, Vol. 31, Issue 12, pp. 735-745, 1999

[4] I.K.Lee, M.S.Kim and G.Elber, "Planar curve offset based on circle approximations", Computer-Aided Design, Vol. 28, Issue 8, pp. 617-630, 1996

[5] Hansen.A, Arbab F, "An algorithm for generating NC tool paths for arbitrarily shaped pockets with islands", ACM Transactions on Graphics, Vol. 11, Issue. 2, pp. 152-182, 1992

[6] Eungki Lee, "Contour offset approach to spiral tool path generation with constant scallop height", Computer-Aided Design, Vol. 35, Issue 6, pp. 511 - 518, 2003

[7] Shi-Nine Yang,Ming-Liang Huang, "A new offsetting algorithm based on tracing technique", ACM Symposium on Solid Modeling and Applications, pp. 201 - 210, 1993

[8] G.Elber, I.K.Lee, M.S.Kim, "comparing offset curve approximation methods", IEEE Computer Graphics and Applications, Vol. 17, Issue 3, pp. 62-71, 1997

[9] L.A. Piegl, W.Tiller, "Computing offsets of NURBS curves and surfaces", Computer-Aided Design, Vol. 31, Issue. 2, pp. 147 - 156, 1999

[10] Bentley J.L, Ottmann T.A, "Algorithm for reporting and counting geometric intersection", Computers, IEEE Transactions on, Vol. C-28, Issue. 9, pp. 643 - 647, 1979

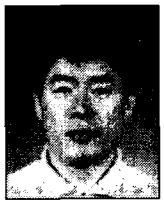
저 자 소 개

오 태 석 (吳 台 錫)

강원대학교에서 제어계측 학사, 석사,박사학위를 각각 1998년, 2001년, 2005년에 받았으며, 현재 강원대학교 BK21 사업단에서 박사후 연구원으로 있다. 관심 연구 분야는 제어 시스템, 지능 제어, 모터 드라이브이다.

윤 성 용 (尹 成 容)

강원대학교에서 제어계측 학사, 석사학위를 각각 1997년, 2001년 받았으며, 현재 동 대학원에서 박사과정 중에 있다. 관심 연구 분야는 지능 제어, 이동로봇 제어이다.



김 일 환 (金 一 煥)

서울대학교에서 제어계측 학사, 석사 학위를 각각 1982년과 1988년에 받았으며, 1993년에 일본 토호쿠대학에서 공학 박사 학위를 받았다. 1995년 강원대학교 전기전자공학부 교수로 임용되어 현재 동 학부 교수로 재직 중이다. 관심 연구 분야는 제어, 메카트로닉스 및 휴먼 인터페이스이다.