

# 컴퓨터 자수용 박음질 알고리즘에 관한 연구

이양선, 장현익  
서경대학교 컴퓨터공학과  
e-mail:yslee@seokyeong.ac.kr  
e-mail:swansong@seokyeong.ac.kr

## A Study on Satin Algorithm for Computer Embroidery

Yang-Sun Lee, Hyun-Ik Jang  
Dept of Computer Engineering, Seokyeong University

### 요약

본 논문에서는 컴퓨터 자수 소프트웨어인 펀칭 프로그램의 개발과정에서 베지에르 곡선식을 사용하여 구현한 박음질(Satin) 알고리즘을 기술한다. 자수에서 박음질은 지그재그로 1회씩 정해진 구간의 밀도와 수폭사이를 왕복하여 작업이 이루어지며 폭이 좁거나 테두리 부분을 작업할 때 주로 사용된다. 박음질은 밀박음질(Running)과 다르게 불규칙적인 면을 채울 수 있도록 고안됐다.

### 1. 서론

70년대 우리나라의 수출 주종목 가운데 하나였던 섬유업은 수출에서 차지하는 높은 비중에도 불구하고 자수기 및 자수 프로그램은 거의 외국의 기술에 의존해 왔다. 모든 의류제품에는 브랜드 상표나 마크가 부착되어 있으며 이러한 상표나 마크에는 반드시 자수가 들어간다.

의류제품에 자수(Embroidery)를 표시하기 위해서는 자수를 만들어 내는 자수기(Embroidery Machine)가 있어야 하며 이 자수기를 구동해 주는 자수 프로그램인 펀칭 프로그램(Punching Program)이 반드시 있어야 한다. 자수기는 근간에 한국 Sunstar라는 곳에서 자체 개발하여 외국 제품과 경쟁해왔으나 자수 프로그램은 독일의 윈스(WinGs), 일본의 타지마(Tajima) 제품을 사용해 왔다. 이런 이유로 순수 국내 고급 자수 제작 기술을 바탕으로 경쟁력있는 펀칭 프로그램의 개발이 필요한 시점이 되었다.

컴퓨터 자수 프로그램인 펀칭 프로그램은 크게 밀박음질(런닝-Running), 박음질(사틴-Satin), 누비기(다다미-Tatami)의 세가지 기능으로 구성된다. 컴퓨터로 자수를 놓기 위해서는 먼저 밀박음질로 원단의 틀어짐을 방지하고 박음질을 통해 테두리부분과 거리가 짧은 부분을 디자인 내용에 따라 밀도와 수폭을 조절하여 채우고 다음에 범위가 넓은 부분을 채우기로 마무리해야 한다[10].

본 논문에서는 윈도우즈용 펀칭 프로그램의 개발과정에서 연구되어진 박음질 알고리즘을 기술한다. 펀칭 프로그램의 기능 중 기본이 되는 박음질 기능은 크게 직선 박음질과 곡선 박음질로 구분할 수 있는데 곡선 박음질은 실제 구현하는 과정에서 내부적으로 직선 박음질을 사용하고 있기 때문에 곡선 박음질에 대한 설명을 통하여 전체 자수용 박음질 알고리즘을 기술하기로 한다.

### 2. 베지에르 곡선식

자수용 프로그램에서 사용되는 박음질 및 누비기 기능은 대부분 곡선 혹은 곡면을 기준으로 작업이 이루어지기 때문에 박음질 및 누비기 기능을 구현하려면 사용자가 원하는 모양의 곡선을 자유롭게 표현할 수 있도록 하는 연구가 반드시 선행되어야 한다.

컴퓨터에서 이러한 곡선의 표현은 주로 많은 수의 작은 선분을 서로 연결하여 작성하게 되는데 이때 각각의 선분 요소를 모두 표현하는 것은 표현에 필요한 시간은 많이 소요되는 것에 비하여 표현된 모양의 자연스러움은 그에 비례하지 않는다. 때문에 단지 몇 개의 점만을 표시한 후에 매끈한 곡선을 그리는 베지에르 곡선(Bezier Curve)과 B-Spline 곡선의 두가지 방법이 널리 사용되고 있는 실정이다.

## 2.1. 베지에르 곡선

그래픽에서 곡선을 처리하는 대표적인 기법인 베지에르 곡선(Bezier Curve)[14,15]은 1970년대 초반에 자동차의 자동 설계에 사용할 목적으로 프랑스의 수학자 P.Bezier라는 사람에 의해 개발되었다. 이 기법은 계산이 쉽고 안정적이기 때문에 계산시간이 많이 요구되는 그래픽 프로그램에서 많이 사용된다.

베지에르 곡선은 두 끝점을 연결하며 여러개의 컨트롤 포인트(Control Point)에 의해 결정되는 곡선을 이루는 부분을 가지고 있다. 베지에르 곡선의 모든 부분에 대해 컨트롤 포인트가 영향을 주지만 B-Spline 곡선은 곡선상의 현재 위치의 가장 가까이에 있는 네 개의 점에 의해 영향을 받는다. 베지에르 곡선은 두 점간의 매우 부드러운 곡선을 제공해주는 반면 B-Spline 곡선은 컨트롤 포인트사이에 그려진 불규칙적인 라인을 보다 가깝게 따라가는 부드러운 곡선을 제공해 준다. 때문에 본 연구에서는 베지에르 곡선을 사용하기로 결정하였다.

## 2.2. 베지에르 곡선의 방정식

베지에르 곡선의 방정식은 3좌표에서 N개의 컨트롤 포인트를 주었을 때 구해지는 베지에르 곡선의 함수식이다. 여기서, 컨트롤 포인트는 베지에르 곡선을 그리기 위해 임의로 선택해준 좌표점을 말한다.

$$P(u) = \sum_{k=0}^n p_k B_{kn}(u), \quad 0 \leq u \leq 1$$

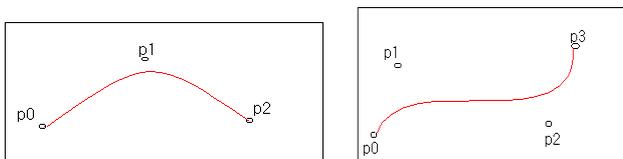
$n+1$  개의 컨트롤 포인트,  $p_k = (x_k, y_k, z_k), k=0, 1, \dots, n$

$$B_{kn}(u) = \frac{n!}{k!(n-k)!} u^k (1-u)^{n-k}$$

[그림 1] 베지에르 곡선식

3좌표  $x, y, z$ 는  $P_x(u), P_y(u), P_z(u)$  베지에르 곡선의 함수식에 의해 각각 계산된 3좌표에 의해 그려진다. 그러나 여기서는  $z$ 좌표는 고려하지 않고 단지  $x, y$ 좌표를 이용하여 곡선을 그린다. 이 함수식에서 매개변수  $u$ 는 각각의 컨트롤 포인트가 곡선을 만들 때 영향을 주는 요소이다.

다음은 위 방정식에 의해 그린 베지에르 곡선의 예이다.

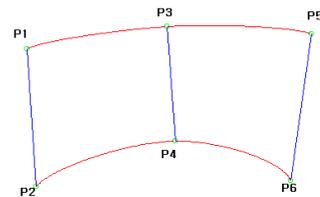


[그림 2] N=1와 N=2일 때의 베지에르 곡선

위 그림을 통해서 베지에르 곡선은 시작점과 끝점을 항상 지나지만 가운데의 컨트롤 포인트들은 지나지 않고 단지 그 위치가 곡선의 모양에 영향을 주는 것을 알 수 있다. 따라서 어떤 곡선을 표현하기 위해서는 곡선의 두 끝점을 시작점과 끝점으로 하고, 그리려는 곡선에 맞게 가운데에 컨트롤 포인트를 주면 된다.

이를 기초로 점의 좌표가 6개 일 때 베지에르 곡선을 그리는 방법은 다음과 같다. 점 6개를 찍으면 곡선이므로 첫째, 우선  $p_1, p_3$ 사이의 길이와  $p_2, p_4$  사이의 길이를 구한다. 둘째, 두 직선 중에서 큰 쪽을 기준을 잡아 큰 쪽을 밀도만큼 나눈다. 셋째, 큰 직선을 나눈 수만큼 다른 직선도 나눌 수 있도록 계산해 둔다. 넷째, 큰 직선과 작은 직선 사이를 연결하는 직선을 미리 계산해 두고 베지에르 곡선을 그려 나간다. 다섯째, 두 개의 베지에르 곡선을 그려가면서 베지에르 곡선의 포인트끼리 가상적으로 연결한 다음 네번째에서 구한 두 직선의 교점을 구하여 지금 구한 두 좌표와 전에 찾았던 좌표를 읽어서 그 사이를 영문자 N 모양으로 연결해 준다.

다음은 이와같은 방법으로 6개의 점의 좌표를 가지고 그린 베지에르 곡선의 예이다.

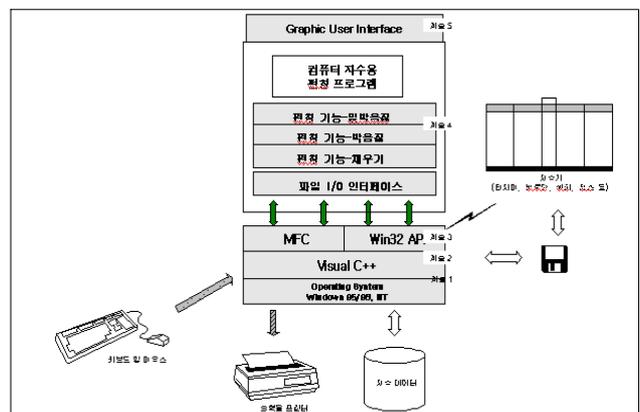


[그림 3] 베지에르 곡선의 예

## 3. 박음질 알고리즘

### 3.1 시스템 구성도

편칭 프로그램의 시스템 구성도는 [그림 4]에서 보는바와 같이 PC의 윈도우즈 환경위에서 객체지향 프로그래밍 언어인 Visual C++ 언어[1,4,13]를 사용



[그림 4] 편칭 프로그램의 시스템 구성도

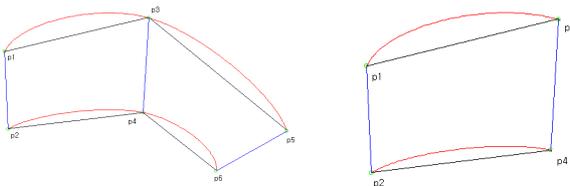
하여 개발하였으며 이 과정에서 MFC 라이브러리와 Win32 API[3,9]를 사용하였다.

### 3.2 박음질 알고리즘

박음질(Satin)은 지그재그로 1회씩 정해진 구간의 밀도와 수폭사이를 왕복하여 작업이 이루어지며 폭이 좁거나 테두리 부분을 작업할 때 주로 사용된다. 박음질은 밀 박음질(Running)과 다르게 불규칙적인 면을 채울 수 있도록 고안됐다.

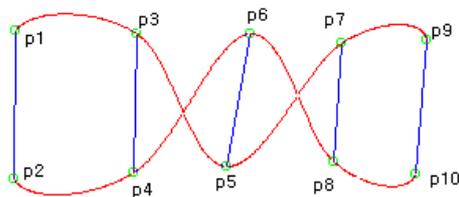
박음질은 외부로부터 밀도(density)와 기본점을 입력받아 곡선을 그리고 곡선에 사절점을 구한 다음 이 사절점들을 서로 연결하는 작업이다.

박음질은 우선 밀도와 점을 입력받는다. 그리고 짝수와 홀수로 구분하여 각각을 한 라인으로 처리하여 프로그램을 수행하고 그림과 같이 4개의 점을 하나의 블록으로 구성한 후에 블록단위로 모든 블록이 처리될 때까지 박음질 작업을 수행한다.



[그림 5] 점 6개인 누비기 [그림 6] 하나의 블록 예

만일 블록이 교차되지 않은 블록이라면 각각 4점을 입력순서로 p1, p2, p3, p4에 입력한다. 만약 그 블록이 교차된 블록이라면 p3와 p4를 교환한다. 이것은 나중에 모양이 나빠지는 것을 방지하고 다음에 임시 사절점을 구하기 위해 교차되지 않은 블록으로 고치기 위해서다.



[그림 7] 교차된 박음질 예

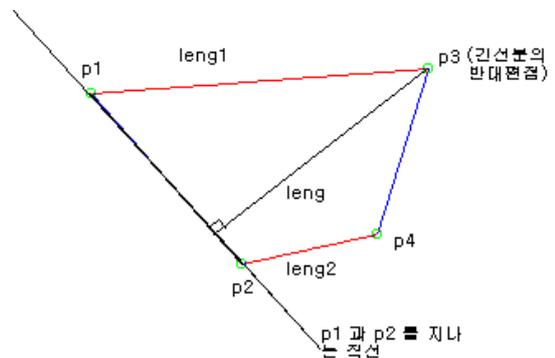
다음에 잘라주는 간격을 정하기 위해 길이를 구한다. 이것은 각각의 길이를 밀도로만 나누었을 때 생기는 문제 즉, 짧은 길이 쪽이 점이 작아서 점을 블록단위로 수행할 수 없는 문제를 해결할 뿐 아니라 [그림 8]과 같이 어느 하나의 길이를 중심으로 사절점을 나누었을 때 한쪽이 일방적으로 넓어지거나 일방적으로 좁아지는 현상을 해결

해 줄 수 있다.



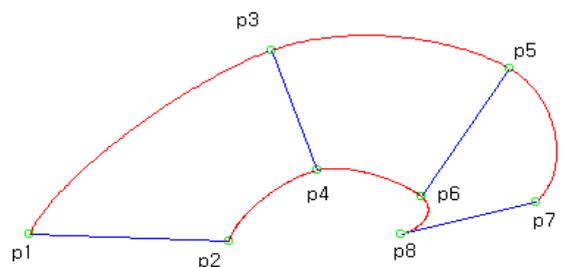
[그림 8] 짧은쪽과 긴쪽을 기준으로 한 누비기

따라서, 이때 길이는 [그림 9]와 같이 선분 (p1,p3)의 길이와 선분 (p2,p4)의 길이를 구한 후 긴 선분의 끝점에서 p1과 p2 직선에 내린 수선을 길이(leng)로 설정한다.



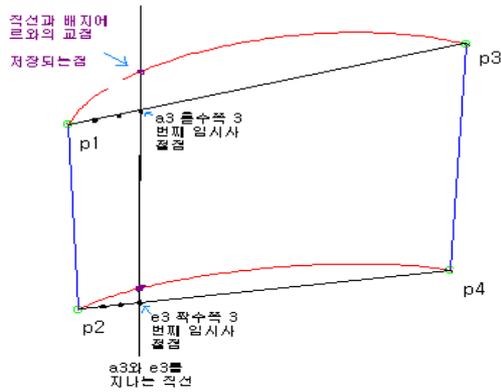
[그림 9] 길이를 구하는 작업

그리고 길이를 밀도로 나누어 사절점의 개수를 구한다. 점의 개수를 구한 다음에는 실제 곡선을 구하기 위해 앞에서 바꾼 p3와 p4를 다시 원상태로 만든다. 그 이유는 실제로 구하는 베지에르 곡선은 교차된 곡선이기 때문이다. 베지에르 곡선에서 실제 길이를 구해서 사절점을 구하는 것은 너무나 어렵다. 또한, 직접 베지에르 곡선에서 사절점을 구한다 하여도 반드시 좋은 모양이 나오는 것은 아니다. 오히려 곡선이 복잡한 경우 [그림 10]의 p8 부근과 같이 블록의 시작점 또는 끝점에서 촘촘해지는 경우가 발생할 수 있다.



[그림 10] 블록의 끝점에서 좁아지는 경우

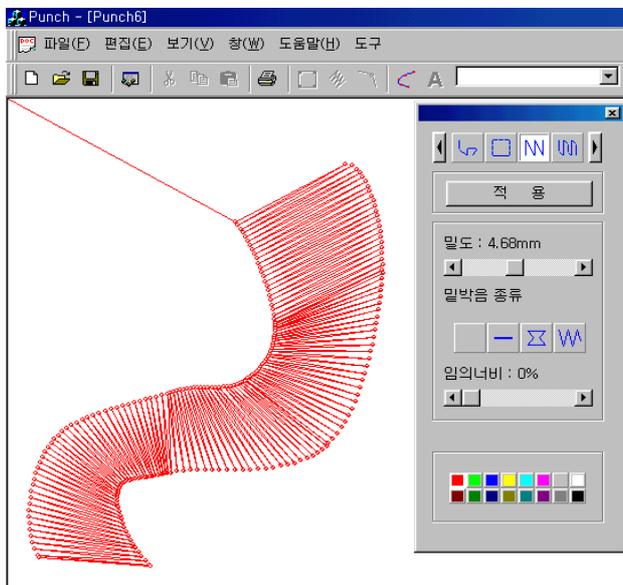
따라서, [그림 11]과 같이 (p1,p3) 직선과 (p2,p4) 직선을 각각 사절점의 개수로 나누어 임시 사절점을 구한 다음 (p1,p3) 직선의 임시 사절점과 (p2,p4) 직선의 임시 사절점을 연결한 직선을 구하고 각각의 직선에 관하여 (p1,p3) 곡선과 (p2,p4) 곡선의 교점을 구한다.



[그림 11] 교점 구하기

앞에서 구한 각각의 사절점을 순서에 맞춰 리스트에 저장하고 임시 사절점들의 직선과 두 곡선의 교점을 구해 저장하는 작업을 블록이 끝날때까지 반복한다.

[그림 12]는 위 알고리즘에 따라 박음질되어야 할 곳의 외곽선을 입력시킨 후에 그 사이점들을 주어진 값인 밀도로 분할하여 면을 채운 모습이다.



[그림 12] 박음질-외곽선 입력 및 면채우기

#### 4. 결론

의류제품에 자수를 표시하기 위해서는 자수 프로그램인 펀칭 프로그램이 반드시 있어야 한다. 그러나 섬유산

업의 사양화로 국내 자수 프로그램 업체가 2-3군데 밖에 안되고 원천기술이 부족하여 국내에서는 거의 독일의 wings(Wings)와 일본의 타지마(Tajima) 제품을 사용해 왔다.

이런 이유로 본 논문에서는 순수 국내 고급 자수 제작 기술을 바탕으로 윈도우즈용 펀칭 프로그램의 개발을 통해 원천기술을 확보하고 외국 제품의 국내시장 잠식을 막기 위해 펀칭 프로그램의 기본형을 개발하였다. 본 논문에서는 이 펀칭 프로그램의 중요한 기능중 하나인 박음질 알고리즘을 기술하였다. 앞으로 매끄럽고 섬세한 고급 자수걸을 제작하기 위한 기능 향상이 필요하다.

#### 참고문헌

- [1] B.Stroustrup, The C++ Programming Language, Prentice-Hall, 1996.
- [2] E.Horowitz,S.Sahni,D.Mehta, Fundamentals of Data Structure in C++,CSP,1992.
- [3] J.Conger, Windows API Bible, The Waite Group, 1992.
- [4] J.Martin & J.Odell, Object-oriented Analysis & Design, Prentice-Hall, 1992.
- [5] L.Heiny, Windows Graphics with Borland C++, John Wiley & Sons, 1994.
- [6] Pohl, Object-oriented Programming using C++, Addison-Wesley, 1996.
- [7] P.S.Wang, C++ with Object-oriented Programming, PWS, 1994.
- [8] Viking Sewing Machines AB, Professional Embroidery System User's Guide, 1998.
- [9] R.Simon, WIN 32 API 슈퍼 바이블 상,하, 정보문화사, 1998.
- [10] Tajima Industires Ltd., DG/ML For Windows User Manual, 1999.
- [11] V.B.Anand, Computer Graphics & Geometric Modeling for Engineers, John Wiley & Sons, 1992.
- [12] 사재학, 시스템 분석과 설계, 옹보, 2000.
- [13] 송호중, MFC 4 바이블, 대림, 1997.
- [14] 유희용, 베지에르 곡선의 컨트롤포인트 찾기, 프로그램 세계 4월호, 1997.
- [15] 임용식, C 그래픽 연습, 크라운 출판사, 1993.