산업기술연구(강원대학교 산업기술연구소 논문집), 제19집, 1999.
Research Report. The Institute of Industrial Technology, Kangwon Nat'l Univ., Korea, No. 19, 1999.

컴퓨터 제어 패턴 재봉기를 위한 패턴 데이타 추출 및 생성 알고리즘

Pattern Data Extraction and Generation Algorithm for A Computer Controlled Pattern Sewing Machine

윤 성 용* 백 상 현** 김 일 환***
Yun, Sung-yong Baik, Sang-hyun Kim, Il-hwan

Abstract

The computer pattern sewing machine is an automatic sewing machine that is controlled by an input pattern. Even a novice can run this machine for various tasks fast and reliably such as sewing a button, a belt ring and an airbag, etc. The pattern processing software, which is the main software of this machine, is for editing and modifying pattern data by online teaching or off-line editing, setting up parameters, and calculate a moving distance of working area on the x-y axes.

In this paper we propose an algorithm to generate pattern data for sewing by simplifying image data. The pattern data are composed of outline data like dot, line, circle, arc, curve, etc. We need converting this data into sewing data which involve sewing parameter, moving distance of working area on the x-y axes. thread, spindle speed.

키워드 : *패턴, 재봉기*
Keywords : *Pattern, Sewing Machine*

1. 서 론

컴퓨터 제어 패턴 자동재봉기는 재봉할 패턴을 입력한 후 입력된 패턴을 자동으로 재봉하는 재봉기로서 단추달이, 벨트고리, 에어백 등 다양한 응용분야에 걸쳐 사용자의 숙련도와 무관하게 고속으로 신뢰성 있는 봉제 제품을 생산할 수 있는 자동재봉기이다. 이 재봉기에서 사용하는 주요 소프트웨어인 패턴처리 소프트웨어는 재봉할 패턴 데이터의 편집 및 수정, 파라미터의 설정, X-Y축 이송 데이터의 추출 등을 위해 필요한 데이터를 처리하는 소프트웨어로써 온라인 교시(online teaching)와 오프라인 편집(offline editing)에 의해 다양하게 설정, 편집할 수 있다. 그러나 이러한 기능으로 복잡한 무늬의 패턴을 편집하는 데는 상당한 시간과 노력이 요구된다.

본 논문에서는 위의 패턴처리 소프트웨어의 구현 원리와, 스캐너를 통해 얻거나 여러 가지 응용프로그램을 통해서 얻을수 있는 여러 가지 다양한 이미지 파일을 다소의 작업 없이 재봉 데이터를 생성하여 컴퓨터 제어 패턴 자동 재봉기에서 재봉할 수 있도록 자동패턴 인식 및 생성 알고리즘을 제안한다. 그리고 이러한 이미지 없이 바로 패턴을 생성해 낼 수 있는 패턴 편집용 프로그램도 보여주었다. 이미지를 통한 패턴 데이터 추출은 기준 땀폭, 최대 땀폭, 이송 속도, 공송 등의 파라미터를 바탕으로 패턴 데이타를 추출 생성하며, 패턴처리는 사용자 정의 파라미터를 이용하여 재봉 데이터를 생성한다.

2. 이미지 간략화 알고리즘

패턴 재봉기로 표현할 수 있는 이미지는 매우 단순하다. 단지 선으로 모든 패턴을 그려야 하며 그러기 위해서는 복잡한 면이나 색상으로 되어 있는 이미지

* 강원대학교 대학원 제어계측공학과 석사과정
** 강원대학교 대학원 제어계측공학과 석사과정
*** 강원대학교 제어계측공학과 교수, 공학박사

위해서는 복잡한 면이나 색상으로 되어 있는 이미지들을 단순히 선만으로 표현해야 한다. 그러기 위해서 여기서는 이미지 처리 기법 중의 하나인 경계선 검출(edge detection) 방식을 사용하였다. 경계선은 영상의 밝기가 낮은 값에서 높은 값으로 또는 이와 반대로 변하는 지점에 존재하는 부분을 가리킨다. 그러므로 영상안에 있는 객체의 경계(Boundary)를 가리키는 것으로서, 모양(Shape), 방향성(Direction)을 탐지할 수 있는 등 여러 가지 중요한 정보가 들어 있는 것이 바로 경계선(edge)이다. 실제로 에지는 영상처리의 기본이며 가장 널리 쓰인다.

경계선 검출(edge detection)은 image segme-ntation의 첫단계이기도 하다. Image segmentation은 image분석의 한 분야로서 pixel들을 image 구성을 결정하는 지역들로 그룹화하여 나누는 것이다. 경계선을 검출하기 위해서 마스크를 사용한 컨벌루션을 하게 된다. 여기서는 컨벌루션 마스크로 라플라시안

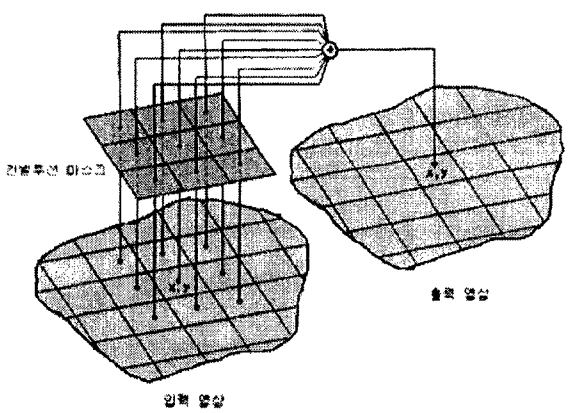


그림 1. convolution

마스크를 사용했다. 그림 1은 컨벌루션에 의한 필터링을 나타낸다.

라플라시안 마스크는 2차 미분연산자이다. 2차 미분은 1차 미분에 비해 넓은 폭의 경계를 한 픽셀로 줄이는데 있어서 더욱 탁월하다. 이러한 넓은 폭의

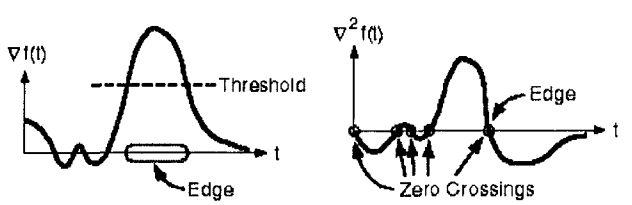


그림2. 1차 미분과 2차미분에서의 경계

0	-1	0
-1	4	-1
0	-1	0

a) 4방향 경계선 검출

-1	-1	-1
-1	8	-1
-1	-1	-1

b) 8방향 경계선 검출

그림 3. 라플라시안 마스크

경계를 한 픽셀의 경계로 축소시키는 것을 thining이라고 한다. 그림 2에서 보듯이 2차 미분보다 1차 미분에서의 경계값이 더욱 넓은 것을 알 수 있다.

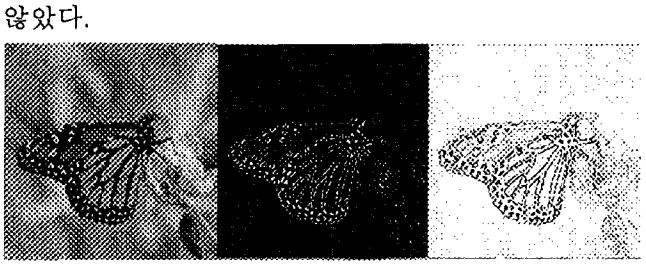
2차미분 연산자(Second order derivative operator)의 또 다른 장점은 edge 윤곽선이 폐곡선이라는 점이다. 이것은 image segmentation에서 매우 중요하다. 또 부드럽게 선형으로 변화하는 부분에서는 작용하지 않는다.

이 논문에서는 b)의 마스크를 사용한 프로그램을 작성하였다. b)의 마스크를 사용한 컨볼루션은 식(1)에서 보여준다.

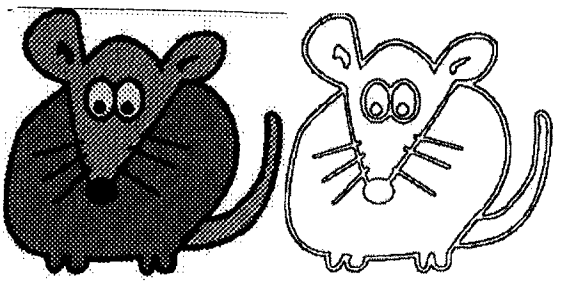
$$\nabla^2[\, f(x,y)\,] = 8f(x,y) - [\, f(x-1,y-1) + f(x-1,y) + f(x-1,y+1) + f(x,y-1) + f(x,y+1) + f(x+1,y-1) + f(x+1,y) + f(x+1,y+1)] \quad (1)$$

위의 라플라시안 마스크를 사용하여 필터링한 이미지가 그림 4에 나와있다.

그림 4는 일반적인 사진영상과 일반 응용프로그램에서 얻을 수 있는 클립아트 이미지를 간략화한 것이다. 여기서는 재봉용 패턴 데이터로 용이하게 변환시키기 위해 경계선 추출후 이미지를 반전시켜서 데이터를 흰색이 아닌 다른 색으로 나타나게 했다. 사실 이 논문의 중요한 목적 중 하나는 간략화된 이미지에서 재봉용 패턴 데이터를 추출하는 것이기 때문에 재봉기로 표현할 수 없는 a)와 같은 이미지는 다루지 않았다.



a) 나비사진 b) 경계선 추출 c) 반전



d) 쥐 이미지 e) 경계선 추출후 반전

그림4. 간략화된 이미지

3. 패턴 데이터 추출

간략화된 패턴 이미지에서 재봉용 데이터를 추출하는 방법은 다음과 같다.

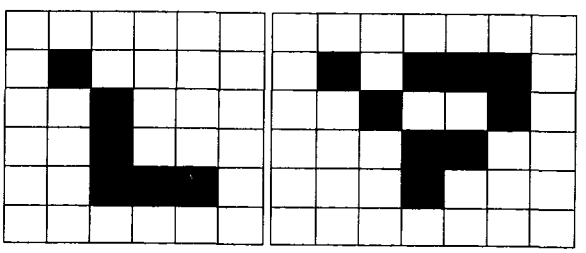


그림 5. 확대된 샘플 이미지

위의 그림을 샘플 이미지라고 했을 때 한 칸으로 표현된 부분은 이미지의 한 픽셀을 나타내고 검은 색으로 표시된 부분은 어떤 데이터가 들어가 있는 부분을 나타낸 것이다. 임의의 한 점을 시작점으로 주었을 때 그 점 주위의 8픽셀을 일정한 방향으로 검색해서 데이터가 있는 부분은 저장하고 그 저장된 픽셀로 이동하게 된다.

여기서는 시작점의 좌표를 (x, y)라고 한다면 검색 순서는 (x-1, y+1)인 점부터 시작해서 (x, y+1), (x+1, y+1), ... , (x-1, y)의 시계방향이 된다. 데이터로 표시된 부분의 숫자는 각 데이터를 읽는 순서를 나타낸다. 숫자옆에 *표시가 된 것은 가지(branch)를 나타낸다. 가지점은 기준점을 중심으로 각 방향의 8픽셀중 네 개이상의 데이터를 가지면 가지점으로 그 위치를 저장하게 된다. 한 번 지나간 픽셀을 다시 반복하는 경우가 없도록 가지점에서의 모든 방향의 데이터는 저장 후 비교하게 된다.

각 픽셀을 이동할 때 마다 거리를 계산해서 더해준다. 그래서 설정된 땀폭의 거리와 크거나 같은 점이 처음으로 나온 부분에 땀폭이 표시를 해주고 그 같은 저장하게 된다. 여기서 재봉 데이타의 특성상 검색된 모든 점을 저장할 필요는 없다. 왜냐하면 재봉 데이터는 설정된 땀폭(stitch length)에 따라 재봉을 하는 것이기 때문에 만일 땀폭이 2mm라면 검색된 데이터 픽셀간의 거리가 2mm내에 있는 점들이라면 의미가 없는 값이기 때문이다.

4. 재봉용 패턴 데이터

패턴 봉제 시스템은 주어진 재봉영역 내에서 작업물에 따라 다양한 형태의 재봉 패턴이 이루어지며, 이의 구현을 위해서 설정된 땀 폭, 이송 속도, 설정 원점, 일시 정지 등 운전 매개변수에 따라 패턴 이송계가 연속적인 위치 데이터를 추종해야 한다. 본 논문에서는 패턴 데이터 처리를 위해 패턴 재봉기의 기능을 분석하여 데이터의 호환성을 높이기 위한 개방적인 패턴 데이터의 구조를 설계하고, 다양한 패턴 유형, 운전 매개변수를 편집, 수정하기 위해 오프라인으로 작동하는 패턴 입력기를 구현하였다.

4.1 패턴데이터의 특징

위에서 기술한 패턴 이미지 데이터에서 실제로 재봉할 수 있는 패턴데이터를 생성하는 알고리즘을 논하기 전에 패턴데이터의 특성을 간략히 설명한다.

(1) 패턴

패턴데이터는 점,직선,원,호,곡선 등의 모양을 나타내는 데이터와 공송,사절 등의 재봉기 운용에 관계되는 데이터로 나눌 수 있고, 공송 또는 사절로 연결되어 있는 모양들의 집합을 패턴이라고 한다.

아래 그림 6에서 ①,②,③번으로 나누어지게 되는데 점선으로 표시된 부분이 공송이다.

재봉에서의 패턴은 공송으로 나누어지고 연결되지 않은 모양들의 집합을 말한다. 패턴 ①은 두 개의 직선으로 구성되고, 패턴 ②은 4개의 직선 또는 사각형으로 구성되고 ,패턴 ③은 한 개의 원으로 구성된다.

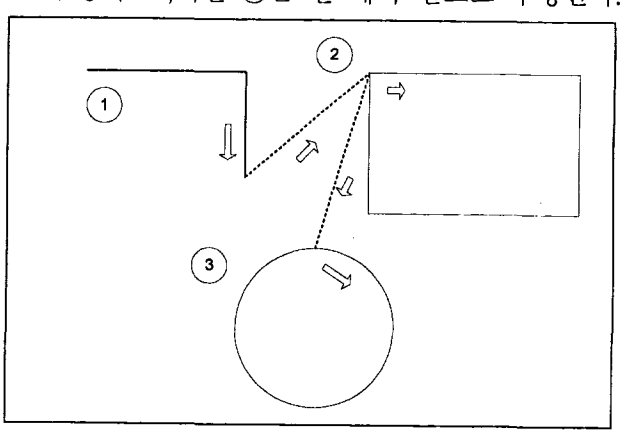


그림 6 패턴

공송은 재봉시 바느질을 하지 않고 이송계만 이동하는 부분으로 여러개의 패턴을 재봉할 때 사용되는 재봉운용데이터이다.

(2) 교시점(Teaching Point)

재봉기가 바느질하는 것을 땀(STITCH)이라고 하고, 땀하는 간격을 땀폭이라고 한다. 보통 땀폭은 2.5mm 정도가 많이 사용되며 땀을 만들기 위해 한 땀 한 땀 일일이 땀데이터를 만드는 것이 아니라 교시데이터를 만든 후 땀데이터를 만들게 되는 데 교시데이터는 직선의 경우는 양끝점이 되고 원의 경우는 땀 생성 방법에 따라 두 점 또는 세 점이 된다.

즉, 모양이 만들어지는 데 필요한 최소한의 데이터를 사용자가 on-line 또는 off-line 패턴 입력기를 이용해 교시하면 땀 생성 알고리즘에 따라 재봉시 필요한 땀데이터와 공송 등의 재봉운용데이터를 만들 수 있고, 교시데이터를 이용하면 땀폭의 변경 또는 재봉데이터의 수정 등을 용이하게 할 수 있다.

본 논문에서도 패턴이미지로부터 교시데이터를 먼저 생성한 후 재봉데이터를 생성하고 있다. 교시점에 대한 예로 그림 7을 보면 패턴①에서 두 개의 직선에 3개의 교시점이 표시되고, 패턴②에서는 4개의 직선에 4개의 교시점이 표시되고, 패턴③에서는 한 개의 원에 3개의 교시점이 표시된다.

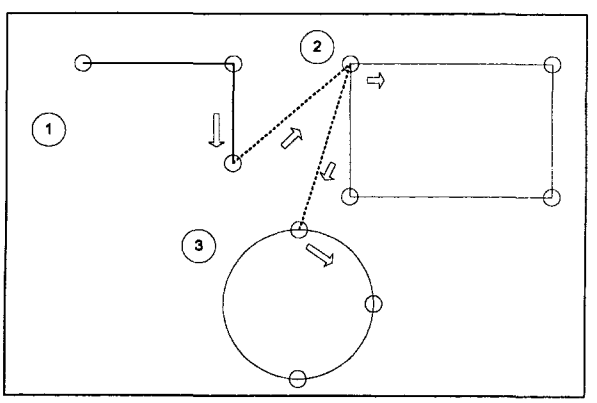


그림 7 패턴의 교시점

4.2 재봉 데이터의 생성

교시데이터로부터 재봉데이터를 생성하는 방법에 대해 간략히 설명한다. 그림 8은 그림 7의 패턴에서 재봉데이터를 생성한 예를 보인다.

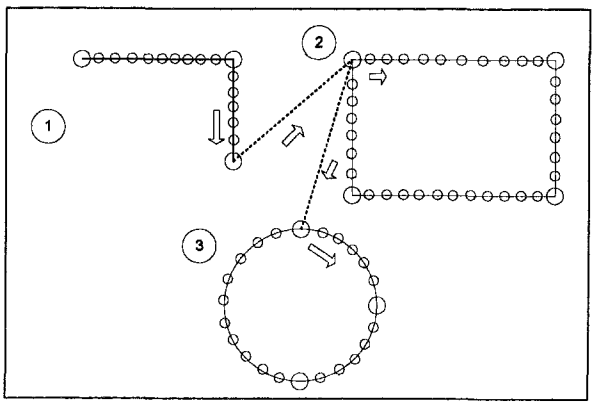


그림 8 재봉데이터 생성

(1) 직선(Line)

직선으로 교시된 처음과 끝점의 좌표가 (x1, y1) (x2, y2)일 때 전 길이는

$$l=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

이고, 땀의 갯수 n은

$$n=(int)\frac{l}{l_s}$$

이다. 여기서 ls는 땀폭이다.

만약 n*lm < 1 이면 n=n+1로 수정한다. i 번째 가야할 땀의 좌표는 다음과 같다.

$$x_i=x_i+(int)(l_x i+0.5)$$
$$y_i=y_i+(int)(l_y i+0.5)$$

로 계산되며 이때 lx, ly 는 다음과 같다.

$$l_x=\frac{l_{x2}-l_{x1}}{n}$$
$$l_y=\frac{l_{y2}-l_{y1}}{n}$$

(2) 호(Arc)

호는 (x1, y1), (x2, y2), (x3, y3)의 교시 데이터가 필요하다. 이들 3개의 점이 주어졌을 때 호 중심의 위치와 반경은 원의 방정식으로부터 각각 다음과 같이 계산된다.

$$x_c=\frac{1}{2}\frac{(x_1^2-x_2^2+y_1^2-y_2^2)(y_2-y_3)-(x_2^2-x_3^2+y_2^2-y_3^2)(y_1-y_2)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)}$$
$$y_c=\frac{1}{2}\frac{(x_1^2-x_2^2+y_1^2-y_2^2)(x_2-x_3)-(x_2^2-x_3^2+y_2^2-y_3^2)(x_1-x_2)}{(y_1-y_2)(x_2-x_3)-(y_2-y_3)(x_1-x_2)}$$

$$r=\sqrt{(x_1-x_c)^2+(y_1-y_c)^2}$$

이 원의 식을 이용하여 각 교시 점의 각 1, 2, 3을 구하고, 이들의 방향을 구한다. 호 상에서 땀폭에 상당하는 증분각은 다음과 같이 계산된다.

$$\beta=\cos^{-1}(1-\frac{l_s^2}{2r^2})$$

이때 호의 각이 α라면 각 땀의 갯수 n은

$$n=(int)(\alpha/\beta)$$

이다. 이들의 식을 이용하여 n-1 개까지의 땀을 계산한다.

(3) 원(Circle)

원의 경우는 호의 경우와 같이 반경과 원의 중심을 구하며, 이 때 땀의 갯수는 다음과 같다.

$$n=(int)(\frac{2\pi}{\beta})$$

이들의 식을 이용하여 n-1개까지의 땀을 계산한다.

(4) 곡선(Spline)

여러 개의 교시 점들 사이를 부드러운 곡선으로 연결하는 것으로 Ferguson 보간법을 이용하였다. 땀의 위치는 교시 점들 사이를 보간하면서 다음과 같은 방법으로 구하였다. i-1과 i번째 땀의 위치 사이의 거리를 li라 했을 때 0.95 ls <= li <= 1.05 ls인 점을 땀의 위치로 결정하였다.

5.패턴데이터 추출 및 생성 알고리즘

패턴데이터는 상기 기술한 패턴이미지의 특성과 재봉데이터의 특성을 고려해서 추출하고 생성해야 한다. 패턴데이터의 추출 및 생성은 첫째 패턴검색, 둘째 시작점 검색 ,셋째 교시점 추출, 넷째 재봉데이터를 생성하는 4단계로 이루어진다.

5.1.패턴 검색

재봉에서의 패턴은 상기 기술한바와 같이 여러 패턴으로 나뉘어질 수 있는 데 패턴을 어떤 순서로 재봉할 것이냐에 따라 재봉 순서가 달라지므로 패턴검색의 순서는 중요하다.

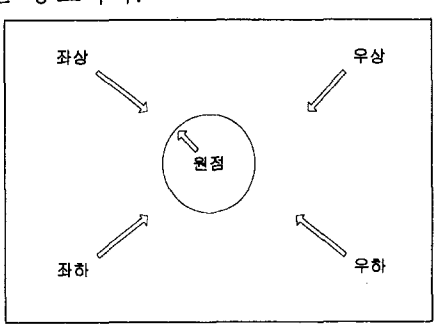


그림 9 패턴검색 시작위치

패턴검색의 시작 위치는 사용자의 옵션 지정으로 정해질 수 있고, 그림 9와 같이 좌상, 좌하, 우상, 우하, 원점 등으로 나누어진다.

지정된 위치에서 x,y를 증가,감소할 때 P(x,y)가 0이 아니면 그 점이 패턴의 검색 시작위치가 된다. 이는 검색상의 방법이고 사용자는 패턴편집기에서 패턴의 재봉 순서를 조정할 수 있다.

5.2. 시작점 검색

패턴의 검색 시작위치가 선정되고 그 패턴의 어디에서부터 재봉 데이터를 추출하는 가는 그 점이 패턴의 공송 위치가 되므로 재봉의 질과도 관계되고, 다음 패턴과의 연계성에서 중요하다.

시작점은 끊긴점이 있는 경우와 끊긴점이 없는 경우로 나눌 수 있는 데 끊긴점이 있는 경우는 그림 10에서 보는 바와 같이 그 점에서 연결성이 하나인 경우로 끊긴점이 여러개 있는 경우에는 시작위치에서 가장 가까운 끊긴점을 시작점으로 본다.

그리고, 끊긴점이 없는 경우는 시작위치에서 가장 가까운 격인점을 시작점으로 보고, 격인점이 없으면 패턴의 시작위치를 시작점으로 본다.

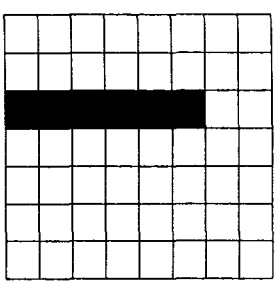


그림 10 끊긴점

시작점 탐색 알고리즘은 다음과 같다

1) 현재의 점이 끊긴점이면
 ①끊긴점 리스트에 현재의 좌표를 저장한다.
 ②끊긴점 카운트를 증가시킨다.
2) 현재의 점이 격인점이면
 ①격인점 리스트에 현재의 좌표를 저장한다.
 ②격인점 카운트를 증가시킨다.
3) 연결되어 있고 미지의 점이 있으면
 ①선분 탐색 우선 순위에 따라 1점 진행
 ②가지점이 있으면 검색 리스트에 저장
4) 연결되어 있으나 이미 가본 점이면
 ①검색 리스트가 비어있지 않으면
  - 검색 리스트에서 가지점 좌표를 현재의 좌표로 설정한다.
  - 검색 리스트에서 가지점 좌표를 제거한다.
  - 1)로 간다
 ②검색 리스트가 비어있으면 6으로 간다
5) 1로 간다
6) 끊긴점 리스트가 비어있지 않으면
 ①시작위치에서 가장 가까운 점의 좌표를 시작점으로 하고 9)로 간다
7) 격인점 리스트가 비어있지 않으면
 ②시작위치에서 가장 가까운 점의 좌표를 시작점

으로 하고 9)로 간다

8) 패턴의 시작위치를 시작점으로 한다.

9) 끝

5.3 교시점 추출 및 생성

상기 기술한 바와 같이 교시점은 패턴데이터를 생성하는 데 기본 골격이되는 데이터이다. 패턴이미지 데이터로부터 교시점을 어떻게 추출,생성하느냐에 따라 의도한 패턴과 다른 패턴데이터가 생성될 수 있다.

교시점은 선분 탐색 우선순위에 따라 패턴의 선분을 탐색하며 끊긴점 판단, 꺽인점 판단, 교차점 처리, 겹침점에서의 소멸된 연결점 복원, 모양에 따른 교시점 생성 등에 의해 추출하게 된다.

(1) 선분 검색 우선순위

선분 검색 우선순위란 2이상의 갈림길이 있을 때 어느 방향을 먼저 검색할 것인 가를 결정하는 방법이다. 직진성에 의해 검색하던 방향을 우선하여 탐색하며 식(3.1)에 의해 연결성이 2이면 갈 수 있는 방향은 한 방향이므로 그 방향으로 진행하고, 3이상이면 선분이동 각도를 식()로 계산하여 가장 작은 각도를 가진 방향을 우선적으로 진행하게 되고, 가지 않은 방향은 검색 리스트에 저장하고, 더 이상 갈 방향이 없을 경우에 검색 리스트 중에서 현재 점에서 가장 가까운 점으로 이동한다.

연결성은 어느 한 pixel의 주변 8 piexel를 기준으로 단순히 패턴이 연결되었는지 만을 판별할 수 있으므로 본 논문에서는 선분의 이동각도를 이용해 우선순위를 결정하는 방법을 제시한다. 선분의 이동각도는 어느 한 pixel의 전후로 1 pixel과 2 pixel의 각도를 계산하는 것으로 꺽인점 판단, 교차점에서의 직진성 판단 , 선분탐색우선순위 등 본 논문에서 중요한 판단기준이다.

선분의 이동각도는 다음과 같이 계산된다.

$\theta n(x,y) = [\alpha n2(x,y) + \alpha n1(x,y)]/2$

여기서, $\theta n(x,y)$는 P(x,y)의 n번째 선분이동 각도, n=0,1,3,...,7

$\alpha n2(x,y)$는 P(x,y)에서 2 pixel 이전의 좌표와 2 pixel 앞의 점과의 각도.

$\alpha n1(x,y)$는 P(x,y)에서 1 pixel 이전의 좌표와 1 pixel 앞의 점과의 각도.

예를 들어 설명하면 다음과 같다

	P(x-1,y-2)	P(x,y-2)	P(x+1,y-2)	
P(x-2,y-1)		P(x,y-1)		P(x+2,y-1)
P(x-2,y)	P(x-1,y)		P(x+1,y)	P(x+2,y)
P(x-2,y+1)	P(x-1,y+1)	P(x,y+1)		P(x+2,y+1)
P(x-2,y+2)	P(x-1,y+2)	P(x,y+2)	P(x+1,y+2)	

그림 11 선분검색 우선순위

그림 11에서 2개이상의 갈 수 있는 방향이 있는 경우 선분검색 우선순위를 판단하기 위해 현재의 점 P(x,y)이라고 하면 P(x,y)에서 P(x-2,y-2)와 P(x+2,y+2)의 선분 이동각도를 구하면

$\alpha 12(x,y)$ = angle(P(x-2,y-2),P(x+2,y+2)) = 0

$\alpha 11(x,y)$ = angle(P(x-1,y-1),P(x+1,y+1)) = 0

$\theta 1(x,y) = [\alpha 12(x,y) + \alpha 11(x,y)]/2 = 0$

이고, P(x,y)에서 P(x-2,y-2)와 P(x+2,y+2)의 선분 이동각도를 구하면

$\alpha 22(x,y)$ = angle(P(x-2,y-2),P(x+2,y-2)) = 45

$\alpha 21(x,y)$ = angle(P(x-1,y-1),P(x+1,y-1)) = 45

$\theta 2(x,y) = [\alpha 22(x,y) + \alpha 21(x,y)]/2 = 45$

이고, $\theta 1(x,y) < \theta 2(x,y)$이므로 P(x+1,y+1)으로 진행한다.

(2) 끊긴점 처리

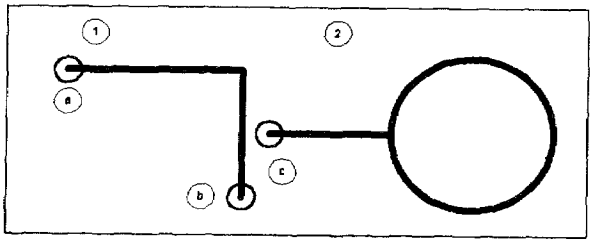


그림 12 패턴의 끊긴점

끊긴점은 연결성이 1인 경우로 패턴의 시작이나 패턴의 끝이므로 교시점이 된다. 그림 1.2에서 ①번 패턴과 ②번패턴의 ⓐ,ⓑ,ⓒ가 여기에 해당된다.

(3) 꺽인점 처리

꺽인점은 주로 선분이 직선으로 된 경우에 다른 직선으로 이분되거나 원 , 호, 곡선 등으로 바뀌는 경우에 나타난다. 예로 들면 그림 5.3.3의 ①번 이미지를 보면 ⓐ에서 ⓑ로 이동시 꺽인점이 나타나고 이점이 교시점이 된다. ②번 이미지는 직선에서 곡선으로 바뀌는 경우로 ⓒ에서 ⓓ로 이동시 꺽인점이 나타나고 이점이 교시점이 된다. 꺽인점은 선분의 이동각도를 계산하고 이동각도가 어느 정도 커지면 꺽인점으로 판단한다. 꺽인점은 교시점이 된다.

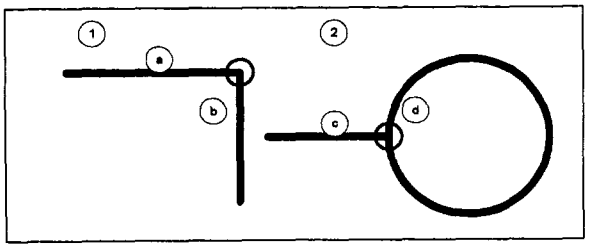


그림 13 꺽인점의 예

(4) 교차점 처리

교차한다는 것은 한 모양 이상이 서로 만나는 경우를 말할 수 있으나 본 논문에서는 한 모양 이상이 서로 만나고 최소한 한 개의 선분은 직진하는 점을 교차점으로 정의한다.

한 모양 이상이 서로 만나는 경우라도 한 선분이 직진선분이면 교시점이 생성되지 않을 수 있는 데 이 점이 꺽인점과 다르므로 교차점을 다시 정의했다. 직진여부는 선분의 이동각도를 계산하고 각도가 5도 이내일 때로 가정한다.

교차점에서는 먼저 직진성에 의해 다음 검색점으로 이동하고 현재의 점에서 검색하지 않은 점을 가지점이라 하는 데 , 이 가지점은 좌표와 가지점의 방향을 검색리스트에 저장한 후 더 이상 갈 지점이 없으면 검색리스트에서 읽어와 상기 선분 탐색 우선순위에 따라 검색하며 교차점에서의 가지점이 직진이면 모양별로 교시점이 생성되거나 또는 생성되지 않는 다.

그림 14에서 ①번 패턴은 두 개의 직선이 교차하는 경우로 교시점은 직선 ⓐ->ⓑ와 직선 ⓒ->ⓓ 재봉데이터가 생성된다.

②번 패턴에서는 직선과 원이 교차하는 경우로 ⓐ->ⓑ를 검색 후 ⓑ에서 가까운 ⓒ에서 교시점이 생성되고 다시 ⓒ가 나오면 원상의 가장 거리가 먼점이 원의 양끝점이므로 ⓓ점을 교시점으로 생성한다.

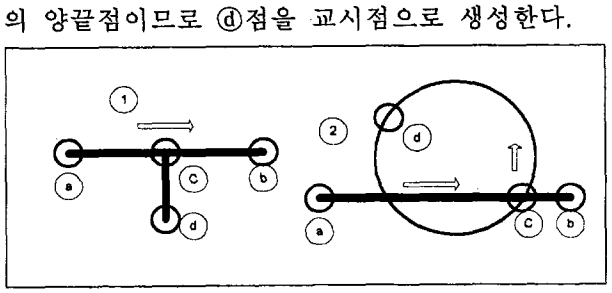


그림 14 교차점에서의 교시점

(5) 겹침점 처리

겹침점은 그림 15에서 선분의 진행방향에 대해 선분이동 각도가 좁아지는 들어가는 점과 각이 넓어지는 나가는 점으로 일단 선분 탐색이 완료된 후 ②처럼 소멸된 점을 복원한 후 들어가는 점과 나가는 점의 중간을 교시점으로 잡고 나머지 교시점을 생성한다.

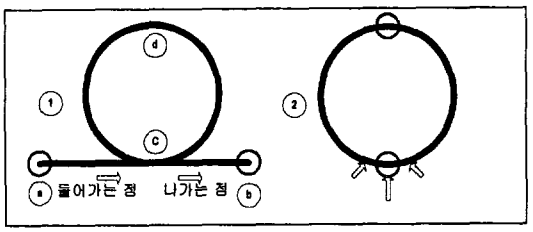


그림 15 겹침점처리

직선,원,호 또는 곡선은 교시점이 생성되는 방법이 다르므로 검색되는 모양에 따라 각기 다른 교시점 생성 방법으로 교시점이 생성되어야 한다.

직선은 두 개의 양 끝점과 직선내의 점은 일직선이라는 성질이 있으므로 현재의 교시점에서 다음 교시점까지 현재의 교시점을 시작점으로 직선성을 잃지 않으면 직선으로 판단한다. 직선성의 판단은 현재의 교시점에서 검색점의 각도가 $-\alpha^\circ \sim \alpha^\circ$ 사이면 직선이라고 본다. 본 논문에서는 pixel 좌표와 실제 좌표 상의 오차를 고려해서 α는 5로 설정했다

원은 현재의 교시점과 검색점이 우선 직선성이 없고 , 검색해서 다시 현재의 교시점으 로 돌아오면 현재의 교시점, 현재의 교시점과 가장 먼 점, 두 점의 중간점으로 원의 방 정식을 만든 후 선분 위에 점들과 비교한다. 검색결과 주어진 땀폭으로 임시 땀데이터를 생성했을 때 오차가 1 pixel이내라면 원으로 판단하고, 세점을 교시점으로 추출한다.

호는 직선 또는 원이 아닌 경우로 양쪽 교시점이 주어지고 이들 두 점의 중간에 한 개의 점을 선택해 호 방정식에 따라 땀데이터를 생성했을 때 1 pixel의 오차라면 나머지 한점 을 교시점으로 생성한다

직선도 원도 호도 아닌 경우는 곡선으로 판단하고 교시점은 매땀폭마다 생성한다. 곡선에서 매 땀폭마다 교시점을 생성시키는 이유는 곡선은 교시점이 많을수록 원래의 이미지에 가까워지기 때문이다

(6) 교시점 추출 알고리즘

상기 기술한 교시점 추출 방법에 따라 알고리즘을 정리하면 다음과 같다

1) 미지의 점이 없으면 5로 간다.

2) 선분탐색 우선 순위에 따라 다음점을 찾는다.

3) 모양별 교시점 추출방법에 따라 교시점을 추출한다.

4) 가지점이 있으면

① 꺽인점이면 속성을 꺽인점으로 표시하고 좌표와 방향을 검색리스트에 저장한다.

② 교차점이면 속성을 교차점으로 표시하고 좌

표와 방향을 검색리스트에 저장한다.

③ 겹침점이면 속성을 들어가는 점 또는 나가는 점으로 표시하고 좌표와 방향을 검색 리스트에 저장한다.

5) 갈방향이 없으면

① 검색리스트에서 좌표, 방향, 속성을 가져온다.

② 속성이 꺽인점이면 꺽인점 처리를 하고 1로 간다.

③ 속성이 교차점이면 교차점 처리를 하고 1로 간다.

④ 속성이 겹침점이면 겹침점 처리를 하고 1로 간다.

5) 끝.

5.4 재봉데이터 생성

상기의 5.3의 교시점 추출 방법으로 그 패턴에서 모든 교시점을 추출하면 그 패턴의 모든 점은 이미지에서 삭제한다. 나머지 패턴도 5.1, 5.2, 5.3의 방법으로 교시점을 찾은 후 모든 패턴의 검색이 완료되었으면 상기 기술한 재봉데이터 생성알고리즘에 따라 재봉데이터를 생성한다

5.5 패턴데이터 추출 및 패턴데이터 생성 알고리즘

패턴이미지에서 패턴데이터를 생성하는 알고리즘을 정리하면 다음과 같다

1) 미지의 패턴이 없으면 4)로 간다.

2) 패턴의 시작위치를 찾고 시작위치가 없으면 3)으로 간다.

① 시작점 탐색 알고리즘에 따라 시작점을 찾는다.

② 교시점 추출방법에 따라 교시점을 추출한다.

③ 검색된 패턴내의 모든 점을 삭제한 후 1로 간다.

3) 교시점데이터로부터 재봉데이터를 생성한다.

4) 끝

6.예 제

그림 16은 그림판에서 그린 모신발회사의 마크를 그린 것이고, 그림 17는 상기 기술된 패턴 인식 및 패턴데이터 생성알고리즘에 따라 패턴데이터를 생성한 것이다. 그림 16을 실제크기로 패턴데이터를 생성하기 위해 1pixel에 0.5mm를 설정했고, 그림 17는 땀폭 3mm로 패턴데이터가 생성되었다.

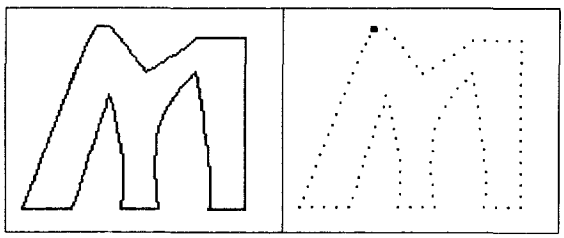


그림 16 예제 패턴이미지 그림 17 생성된 패턴데이터

두 그림을 비교하면 거의 같음을 알 수 있다. 그림 17에서 사각형의 검은 점은 공송을 표시하는 데이는 이 패턴의 시작점으로 패턴 검색 시작위치를 좌상으로 했기 때문이다. 시작점은 5.2의 시작점 검색 알고리즘에 따라 끊긴점이 없으므로 꺽인점이 되고, 패턴 검색 시작위치에서 가장 가까운 꺽인점이 되므로 공송위치는 맞게 설정되었음을 알 수 있다.

7. 결 론

본 논문에서는 재봉용 데이터를 생성하기 위한 이미지 간략화로 라플라시안 마스크를 이용한 컨볼루션을 이용하였고, 간략화된 이미지로 재봉용 패턴데이터를 추출하기 위해 패턴 이미지데이터와 재봉을 위한 패턴데이터의 특성을 설명했고, 패턴 인식과 패턴데이터 추출 및 생성 알고리즘을 제시하였다.

그러나, 이 알고리즘은 앞의 예제처럼 비교적 단순한 패턴은 무리없이 추출해 낼 수 있지만 복잡하거나 한 픽셀로 경계선 추출이 정확히 되지 않은 경우에는 패턴데이터를 사용자가 본래 의도한 대로 재봉패턴을 생성하기가 어려웠다. 특히 겹침점에서 들어가는 점과 나가는 점이 불명확한 경우는 다른 패턴데이터가 생성되는 경우가 있다.

재봉은 그 재봉하는 순서에 따라 재봉의 질이 달라지는 바 일괄적인 알고리즘으로는 전부 해결될 수 있는 문제가 아니고, 겹침점에서의 처리와 패턴의 이미지의 오차의 범위 등은 사용자의 옵션설정 및 수동지시, 이를 테면 재봉순서를 정해준다거나 불명확한 지점은 사용자의 설정을 우선적으로 고려하는 방법 등이 있을 수 있는 데 이러한 점을 충분히 고려하는 부분이 앞으로 과제이다.

참고문헌

[1] J.R. Parker, "Algorithms for Image Processing and Computer Vision", *WILEY*

[2] *JSN International*, 1996, No. 96-5, pp.9-32

[3] *JSN International*, 1996, No. 96-4, pp.5-20
[4] TICE Engineering & Sales Inc., 1995, *Introducing the JL5000 Beltloop Machine*
[5] 박신웅, 공석붕, 1991, *봉제과학*, 교문사, pp.228-236
[6] Brother Industries, LTD, 1991, BAS-311 : *Maintenance Book*, pp.1-35
[7] Brother Industries, LTD, 1991, BAS-311 : *Parts Book*, pp.5-6, 21-22
[8] Norton, R. L., 1992, *Design of Machinery : An Introduction to the Synthesis and Analysis of Mechanisms and Machines*, McGraw-Hill Inc., pp.117-124, 216-220
[9] Kuo, B. C., *Incremental Motion Control Vol. II : Step Motors and Control Systems*, SRL Pub. Co., pp.391-402