

CI 방법을 이용한 2차원 곡선 내의 호 및 선분 분리

Segmentation of Arcs and Line Segments in Planar Curve Using CI Method

류 승 필*
Sung-Pil Lyu

요 약

Cone Intersection(CI) 방법에 의해서 추출된 호는 원래의 곡선으로부터 항상 허용오차 이내에 존재한다. 그러나 CI 방법에 의한 breakpoint의 위치는 허용오차에 영향을 많이 받으며 breakpoint의 위치 오차는 호의 중심 또는 반지름 추출에 나쁜 영향을 준다.

이 논문에서는 CI방법을 개선하여 중심 추출과 호 분리를 위한 방법을 제안하였다. 제안된 방법에 의해 추출된 호는 주어진 곡선으로부터 매우 안정적인 평균 오차를 가진다. 제안된 방법에 의해 추출된 breakpoint들의 위치가 다른 방법들보다 실제 위치에 더 가까운 것을 실험으로 확인하였다.

Abstract

The detected arcs by the cone intersection(CI) method exist within a tolerant error from the original curve. But the position of breakpoint by the CI method is sensitive to the tolerant error. The deviation of breakpoint position makes ill effects to the detection of arc center or radius.

In this paper a improved CI method for center detection and arc segmentation is proposed. The detected arcs from curves by the proposed method have very stable mean of deviations from the original curves. The experimental results show that the position of breakpoint by the proposed method is closer to real position than the one by other methods.

□ keyword : 호의 분리, 호의 추출, 2차원곡선 분리, arc segmentation, arc detection, segmentation of planar curve

1. 서 론

이미지의 윤곽선을 직선과 호로 근사화하는 경우 직선만으로 근사화하는 경우보다 자료 압축효과가 더 뛰어나다[1]. 호는 물체인식에 있어서 매우 중요한 요소 중에 하나이므로 이미지로부터 호를 분리하는 것은 데이터압축 뿐만 아니라 물체를 인식하는 데에 있어서도 매우 중요하다.

이미지로부터 호를 분리하는 방법으로는 원시 데이터를 그대로 입력으로 사용하는 Hough Transform을 이용한 방법[2,3,4]과 벡터를 이용한 방법[5,6]들이 있지만 많은 시간과 기억공간을 요

구하는 단점이 있다. 따라서 이런 단점을 개선하기 위해 2차원 이미지의 경계선을 디지털 곡선으로 표현하고 이 곡선으로부터 호를 분리하는 방법들이 있다[1,7,8,9,10,11,14].

호의 분리방법에는 Lim[7]의 방법과 같이 곡선 내에 있는 호의 특징 파라미터(중심 및 반지름)에 대해서만 연구하는 경우도 있으나, 디지털 곡선 내에 있는 직선과 곡선을 모두 분리 추출하는 방법에 대한 연구가 많이 이루어지고 있다.

디지털곡선으로부터 호의 분리에 있어서 가장 중요한 추출 대상은 호가 분리되는 점, 즉 breakpoint 수와 위치라고 볼 수 있다. breakpoint는 물체 윤곽선의 주요 특징 점으로 이미지 인식에 있어서 매우 중요한 역할을 한다. 만약 breakpoint의 위치 오차가 크면 호의 각도와 위치

* 정 회 원 : 세명대학교 컴퓨터학부 부교수
lsp415@semyung.ac.kr
[2007/11/16 투고 - 2007/11/26 심사 - 2007/12/31 심사완료]

등에 영향을 주게 되고, breakpoint 수가 적절한 수보다 적거나 많으면 일부의 호 또는 직선이 무시되거나 또 다른 호나 직선으로 간주되는 불필요한 분리가 일어나므로 정확한 도형 또는 물체 인식을 할 수 없게 된다.

디지털 곡선으로부터 breakpoint를 찾기 위한 방법들은 대부분 특정한 임계값을 이용한다. 임계값의 종류로는 breakpoint 수[1], 상대거리오차 [8,9,10] 및 절대거리오차[11] 등이 있다. Pei 방법[1]의 경우 breakpoint의 수를 미리 결정하면 그 수에 해당하는 최적의 breakpoint의 위치를 찾아주는 방법이다. 그러나 복잡한 이미지의 경우 그 breakpoint 수를 미리 알기 어려우며 breakpoint수는 패턴인식을 위해 자동으로 구해져야할 중요한 파라미터 중의 하나이다.

Rosin[8]과 Dosch[9]는 주어진 곡선을 직선 근사화(모든 직선에 대해 최대허용오차를 3 pixels로 제한)에 의해 디지털 곡선을 여러 개의 직선으로 근사화하고, 직선으로 근사화된 부분에 대해서 다시 호의 가능성을 검사한다. 여기서 근사화한 직선과 근사화한 호에 대해서 각각 원래의 곡선으로부터 거리오차 오차를 구하고 상대오차(최대오차/근사화 길이)가 작은 것을 선택하여 breakpoint를 찾는 방법이다. 그러나 이 방법은 근사화된 직선이 작은 오차를 가지면 이 직선이 호의 일부가 충분히 될 수 있음에도 불구하고 중요도가 낮은 호의 일부로는 합병되지 않으므로 불필요한 호의 분리가 일어난다. 또한 이 방법은 곡선의 모양에 따라 원치 않는 breakpoint (“uncontrolled location error”[9])도 발생한다.

한편, 호와 디지털곡선 간의 거리오차 제공과 호의 길이의 비율을 임계값으로 하는 Ichoku[10] 방법은 도형크기에 따른 상대적 오차를 이용하므로 큰 도형에 대해서는 더 큰 오차를, 작은 도형에 대해서는 더 작은 허용오차를 적용함으로써 직관적으로 적절한 breakpoint에 가까운 점을 찾을 수 있지만 반대로 호의 크기가 큰 경우에는 큰 오차가 생기며 호가 부드럽게 연결된, 즉, 호의

곡률변화가 작은 곳에 있는 breakpoint의 위치오차는 더 커지는 단점이 있다.

그리고 절대오차를 임계값으로 하는 류승필[11] 방법은 Cone Intersection을 이용하는 방법(이후 CI 방법이라 함)으로 일정한 오차 내에 있는 원의 중심후보 영역을 구하고 그 영역 내에서 원의 중심을 구하므로 호로부터의 주어진 2차원 곡선간의 거리가 항상 허용오차 이내에 있도록 한다. 그러나 허용오차 범위 내에서 최대한 곡선을 추적하므로 실제의 breakpoint를 지나서도 허용오차를 만족하면 계속 나아가므로 2 개의 호가 부드럽게 연결되는 경우에는 breakpoint 위치오차가 커지게 된다.

이 논문에서는 CI 방법을 개선하여(이후 이 방법을 MCI : Modified CI 방법이라 함) 입력 곡선상의 점들이 하나씩 처리 될 때마다 중심을 구하고 그 중심을 이용하여 각각의 점과 호의 거리오차의 크기를 평가하여 breakpoint를 찾는다. 제안된 방법은 기존의 방법들에 비해 breakpoint의 수와 위치 정확도가 높으며, 직선도 하나의 호로 간주하므로 직선과 곡선을 위한 별도의 처리를 거치지 않고 한 번의 처리로 이루어진다.

한편, 직선도 호의 일부(반지름이 무한대인 호)로 처리하기 위해서는 호의 각도와 상관없이 구해진 호와 주어진 곡선 간의 오차가 적고 안정적인 결과를 주는 호의 중심 찾기 방법이 요구된다. 그런데 기존의 중심구하기 방법들[12-15]은 직선 또는 직선에 가까운 호의 경우 반지름이 매우 짧아져서 오차가 커지거나, 중심을 구하는 계산회수가 폭발적으로 늘어나는 단점이 있다[15]. 본 논문에서는 직선과 같이 호의 각도가 0 또는 0에 가까운 곡선에 대해서도 구해진 호와 곡선간의 거리 오차평균이 안정적인 호의 중심 찾기 방법을 제안한다.

본 논문 2장의 2.1절에서는 제안된 방법을 설명하기 위해 기존 CI방법에 대한 알고리즘 및 정의를 개략적으로 표현하고, 기존 CI의 중심구하기의 문제점을 제시한다. 2.2절에서는 개선된 중심

구하기 방법을 제시하고, 2.3절에서는 breakpoint 위치 추적 방법을 기술한다. 그리고 3장은 제안된 방법과 기존의 방법들에 대한 비교실험 결과를 보이며, 4장에서는 결론을 기술한다.

2. MCI 방법

2.1 CI 방법

2.1.1 CI 알고리즘

이절에서는 개선된 중심을 구하는 방법을 설명하기 위해서 기존 CI 방법에서 사용했던 정의와 여기서 새로 정의한 개념을 이용하여 CI 알고리즘을 기술한다.

Z를 점 P_0, P_1 및 P_n 등 $n+1$ 개의 점으로 이루어져 있는 디지털 곡선을 나타내는 점들의 집합이라 하고,

$$Z = \{P_0, P_1, P_2, \dots, P_n\}$$

로 표시할 때,

$|\overline{AB}|, |\overline{AB'}|$: 점 A에서 B에 이르는 선분 \overline{AB} 또는 벡터 \overline{AB} 의 길이

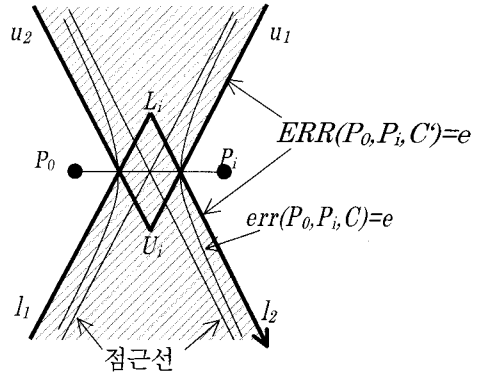
$err(A,B,C) = ||\overline{CA}| - |\overline{CB}||$: $|\overline{CA}|$ 와 $|\overline{CB}|$ 차이의 절대값

$Gr(A,C)$: C를 중심으로 점 A를 지나는 호 또는 원

로 정의하면.

$$err(P_0, P_i, C) \leq e \quad (1)$$

를 만족하는 C가 존재하는 영역은 그림 1과 같이 두 쌍곡선 사이의 영역으로 표현된다.



(그림 1) CI방법에 있어서 원의 중심후보 영역 X_i

그리고, 쌍곡선의 두 점근선을 선분 $\overline{P_0P_i}$ 에 평행한 방향으로 $\frac{e}{2}$ 만큼 좌측 및 우측으로 각각 이동하여 4 직선 사이에 만들어지는 영역(그림 1의 빗금친 영역)에 있는 임의의 점 C' 에 대해서

$$err(P_0, P_i, C') < \sqrt{2}e \quad (2)$$

이다[11]. 그림 1에서 4직선사이의 영역 경계선을 나타내는 함수를 다음과 같이 정의하며,

$$ERR(P_0, P_i, C') = e \quad (3)$$

이때, 식 (3)을 만족하는 경계선 사이에 있는 영역 내의 임의의 점 C' 는

$$ERR(P_0, P_i, C') \leq e \quad (4)$$

로 표현할 수 있다. 이후 Z 내의 모든 점 $P_i(i=0, \dots, n)$ 에 대해서 식 (4)를 적용할 때 허용오차를 'ERR=e'라 표현하고 식(3)을 만족하는 경계선 상의 임의의 점 C' 에 대해 식 (2)를 만족하는 최대값을 e_{ERR} 이라 한다. 즉, $ERR=e$ 이면 $e_{ERR} < \sqrt{2}e$ 이다. e_{ERR} 은 최대 가능 허용오차이다.

한편, 식 (4)를 만족하는 영역을 X_i 영역이라 할 때, X_i 영역은 점 U_i 와 점 L_i 을 꼭짓

점으로 하는 두개의 cone으로 나눌 수 있으며, 각각의 cone영역을 나타내는 집합 U_i 및 L_i 로 정의한다.

이 때, Q를 호 $\widehat{P_0P_i}$ 의 중간점이라 하면

$$A_{Li} = Y_0 \cap Y_1 \cap \dots \cap Y_i \quad (5)$$

여기서, if $\overrightarrow{P_0Q} \cdot \overrightarrow{QP_i} \geq 0$ then $Y_i = L_i$

else $Y_i = U_i$

그리고

$$A_{Ui} = Y_0 \cap Y_1 \cap \dots \cap Y_i \quad (6)$$

여기서, if $\overrightarrow{P_0Q} \cdot \overrightarrow{QP_i} \geq 0$, $Y_i = U_i$

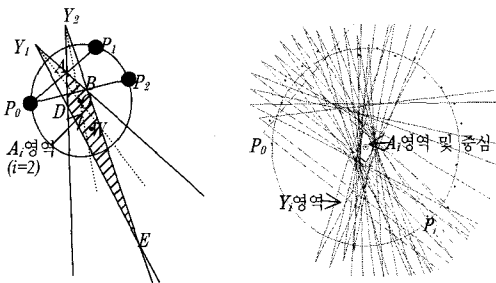
else $Y_i = L_i$

라 할 때, 중심후보영역의 집합은

$$A_i = A_{Li} \cup A_{Ui} \quad (7)$$

이며 A_i 를 'i번째 중심후보영역 A'라 부른다. 즉, $C \in A_i$ 를 만족하는 모든 C는 모든 $i=0, \dots, n$ 에 대해서 P_i 는 식(4)를 만족하며 C를 중심으로 반지름 $|\overline{CP_0}|$ 인 원으로부터 거리 허용오차 e_{ERR} 내에 존재한다.

그림 2는 식 (7)에 의해서 만들어진 중심후보 영역 A의 예이다.



(a) 호의 각도가 180도 미만인 경우

(b) 호의 각도가 180도 이상인 경우

(그림 2) 중심후보영역 A의 예

2.1.2 CI 방법에서 중심 위치의 오차 분석

CI 방법에서는 식 (7)에 의해서 만들어지는 다각형 꼭짓점들의 위치평균을 원의 중심으로 간주한다. 호의 크기(구하고자하는 중심 C에 대해, $\overrightarrow{CP_0}$ 와 $\overrightarrow{CP_i}$ 가 이루는 각도)가 180도 이상인 경우에(그림 2 (b)참조), $\overrightarrow{CP_0}$ 가 x 축상에 있다고 가정하면 꼭짓점들이 y축 방향으로 비교적 골고루 분포하지만, 호의 각도가 180도 미만인 경우 그림 2 (a)와 같이 아랫방향으로 영역이 증가한다. 이것은 식 (7)에서 180도 미만인 경우 cone의 중심선들이 이루는 각도가 모두 예각이므로 cone들의 각도가 크면 cone이 만나는 점(그림 2(a)에서 점 E 참조)이 존재하지 않을 수도 있으며, cone들이 서로 만나지 않으면 A_i 영역은 무한 크기의 영역이 된다.

임의의 곡선을 근사화한 호 c의 높이 h가 현의 중심으로부터 곡선까지의 거리에 의해 결정되고 하면 곡선이 가지는 잡음은 그 크기만큼 높이에 직접 영향을 준다. 그러나 반지름(r)은 다음 식에서와 같이 h에 비례하지 않으며 따라서 잡음에도 비례하지 않는다.

$$r = \frac{d^2 + h^2}{2h} \quad (8)$$

여기서 d=호의 끝점들을 연결하는 직선(현) 길이의 $\frac{1}{2}$

CI 방법은 곡선상의 각 점들에 대해 ERR만큼 오차를 가지는 호의 중심후보를 만들므로 h가 ERR 근접하면 반지름이 매우 커지거나 무한대인 중심후보를 가지므로 이들의 위치 평균인 중심은 무한대 위치가 될 수 있다.

2.2 MCI 방법

2.2.1 호의 중심구하기

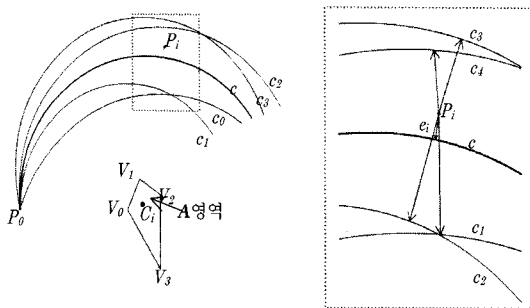
개선된 중심 구하기 방법을 적용하기위해서 m 개의 꼭짓점 V_1, \dots, V_m 이 있는 중심후보영역 A 를 가지는 디지털 곡선 Z 에 대해서 다음 3가지 조건을 가정한다.

- (1) 호 $\widehat{P_0P_i}$ 에 대해서 현 $|\overline{P_0P_i}| \gg e_{ERR}$
- (2) A 영역의 모든 꼭짓점 $V_j (j=1, \dots, m)$ 에 대해 $|\overline{V_jP_0}| \gg e_{ERR}$
- (3) A 영역의 꼭짓점인 모든 $Cir(P_0V_j)$ 에 대해 디지털곡선 Z 의 모든 점 P_i 와의 거리 오차들의 합은 최적의 중심 C 에 대해서 $Cir(P_0C)$ 모든 점 P_i 와의 거리 오차들의 합과 같다. 즉,

$$\sum_{i=1}^n \sum_{j=1}^m (|\overline{V_jP_i}| - |\overline{V_jP_0}|) = m \sum_{i=1}^n (|\overline{CP_i}| - |\overline{CP_0}|) = 0 \tag{9}$$

을 만족한다.

위 (2)의 가정에서 V_j 는 호의 중심후보중의 하나로서 (1)의 가정을 만족하는 경우 $|\overline{V_jP_0}|$ 는 작아도 반지름의 크기에 근접하는 값이라 할 수 있다. 만약 반지름이나 현의 길이가 허용오차(최대 잡음 크기)에 근접하는 경우 수학적으로나 직관적으로도 호를 판정하기가 쉽지 않으므로 (1), (2)의 가정은 실제 호를 추출할 때 타당한 가정(제약조건)이라 볼 수 있다.



(그림 3) 디지털 곡선상의 점(P_i)과 A 영역 꼭짓점을 중심으로 하는 호와의 거리 평균(e_i)

가정 (3)의 경우 A 의 모든 꼭짓점을 기준으로 P_0 를 지나는 원 $Cir(P_0V_j)$ 을 그릴 경우 Z 의 모든 점 P_i 에 대해서 $Cir(P_0V_j)$ 와의 평균오차의 합이 0에 근접함을 의미한다(그림 3 참조). 즉, 잡음이 없고 곡선상의 점들의 간격이 균등하게 분포되어 있다면(디지털 곡선의 경우 픽셀단위로 골고루 분포되어 있다고 가정) A 영역의 꼭짓점들을 중심으로 하는 원은 CI 방법의 특성상 최적의 원으로부터 모두 한계 허용오차($\pm e_{ERR}$)를 가지면서 그 중심(A 영역의 꼭짓점)들은 최적의 중심을 기준으로 대칭적으로 분포하므로 이들의 평균 위치에 존재하는 원은 최적의 원에 근접한다. 그러나 잡음이 있는 경우에는 A 영역의 일그러짐에 의하여 거리평균이 되는 원은 최적의 원으로부터 약간의 오차를 가지게 된다. 특히 잡음의 크기가 크고 호의 크기가 작을수록 가정 (3)으로부터 멀어지지만, 만약 오차가 균일하게 분포되어 A 영역의 꼭짓점이 최적의 중심을 기준으로 골고루 분포되거나 잡음의 크기가 작다면 가정 (3)과 가까워지게 된다. 제안된 방법을 적용하기위해 입력 곡선이 위 3개의 가정들을 항상 충분히 만족해야하는 것은 아니며, 3개의 가정들에 대한 만족정도는 오직 결과에 대한 정확도에만 영향을 준다.

본 방법은 가정 (3)을 적용하여 A 영역의 꼭짓점에 대해서 CI방법에 사용된 중심후보 영역의 꼭짓점 위치 평균 대신 꼭짓점을 중심으로 하는 호와 디지털곡선상의 거리오차의 평균을 이용하여 중심의 위치를 구한다.

식 (9)에서

$$e_{ij} = |\overline{V_jP_i}| - |\overline{V_jP_0}| \tag{10}$$

라두면 위 식 (10)은

$$e_{ij} = \frac{(\overline{V_jP_i} - \overline{V_jP_0}) \cdot (\overline{V_jP_i} + \overline{V_jP_0})}{|\overline{V_jP_i}| + |\overline{V_jP_0}|} \tag{11}$$

로 대신할 수 있다. 이것은

$$e_{ij} = \frac{\overrightarrow{P_0P_i} \cdot (\overrightarrow{V_jP_i} + \overrightarrow{V_jP_0})}{|\overrightarrow{V_jP_i}| + |\overrightarrow{V_jP_0}|} \quad (12)$$

가된다.

$$\overrightarrow{V_jM_i} = \frac{\overrightarrow{V_jP_i} + \overrightarrow{V_jP_0}}{2}, r_j = |\overrightarrow{V_jP_0}| \quad (13)$$

라 두면, 식 (12)는 식 (10)을 적용하면,

$$e_{ij} = \frac{\overrightarrow{P_0P_i} \cdot \overrightarrow{V_jM_i}}{r_j + \frac{e_{ij}}{2}} = \overrightarrow{P_0P_i} \cdot w_{ij} \overrightarrow{V_jM_i} \quad (14)$$

$$\text{여기서 } w_{ij} = \frac{1}{r_j + \frac{e_{ij}}{2}}$$

이다. 여기서

$$\sum_{j=1}^m e_{ij} = m \overrightarrow{P_0P_i} \cdot \overrightarrow{C_iM_i} \quad (15)$$

라하면, $\overrightarrow{C_iM_i}$ 는 $\sum_{j=1}^m w_{ij} \overrightarrow{V_jM_i}$ 의 가중치 평균 벡터이다. 즉,

$$\overrightarrow{C_iM_i} = \frac{\sum_{j=1}^m w_{ij} \overrightarrow{V_jM_i}}{\sum_{j=1}^m w_{ij}} \quad (16)$$

이다. 위 식으로부터

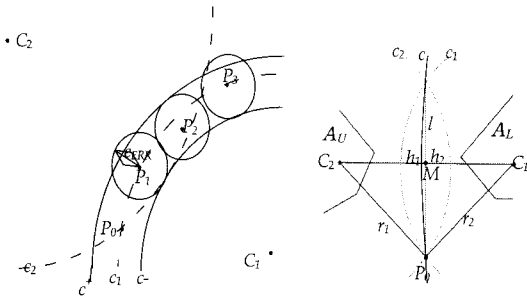
$$\overrightarrow{C_iM_0} = \overrightarrow{C_iM_i} - \overrightarrow{M_0M_i} = \frac{\sum_{j=1}^m w_{ij} \overrightarrow{V_jM_0}}{\sum_{j=1}^m w_{ij}} \quad (17)$$

이다. (여기서 M_0 는 식(13)에 의해 P_0 와 같다). 그런데 가정 (2)에 의해서 $w_{ij} \cong w_{0j} = \frac{1}{r_j}$ 라 두고 이것을 위식에 대입하면

$$\overrightarrow{C_iM_0} = \frac{\sum_{j=1}^m w_{ij} \overrightarrow{V_jM_0}}{\sum_{j=1}^m w_{ij}} \cong \frac{\sum_{j=1}^m w_{0j} \overrightarrow{V_jM_0}}{\sum_{j=1}^m w_{0j}} = \overrightarrow{C_0M_0} \quad (18)$$

이다. 즉, $w_{ij} - w_{0j} \cong \frac{e_{ij}}{2r_j}$ 에서 $e_{ij} \leq e_{ERR}$ 이므로 점 C_0 와 C_i 는 반지름의 역수에 대한 이 오차의 무게평균에 근접하여 존재하므로 최악의 경우에도 두 점사이의 거리가 $\frac{e_{ERR}}{2}$ 을 넘지 않는다. 따라서, C_0 는 임의의 C_i 에 대해서 식 (18)이 모두 성립하므로 $i=1, \dots, n$ 에 대한 모든 C_i 의 가중치 평균 위치인 C 는 C_0 로 근사화할 수 있으며 C_0 는 C 로부터 $\frac{e_{ERR}}{2}$ 이내에 존재한다. 위의 식 (18)은 A 영역의 꼭짓점(V_i)들과 곡선의 시작점 $P_0(=M_0)$ 사이의 벡터와 벡터 크기($Cir(P_0, V_i)$ 의 반지름)의 역수를 가중치로 하는 평균벡터이다.

식(18)은 오직 시작점 P_0 와 A영역의 꼭짓점만 관계되므로 꼭짓점의 갱신/삭제 시에 해당되는 값만 추가/삭제에 의해서 중심을 구할 수 있다. 즉, 디지털 곡선 Z상의 매 점에 대해서 계산복잡도의 증가 없이 중심과 반지름을 구하는 것이 가능하다.



(a) 허용오차(e_{ERR})를 만족하는 호와 그 중심들 (b) 두 호 C_1, C_2 의 평균 오차의 호 C_3

(그림 4) 호의 각도가 작은 디지털 곡선의 중심의 분포

한편 호 $\widehat{P_0P_i}$ 의 각도가 작으면 그림 4 (a)와 같이 허용오차를 만족하는 원의 중심이 호의 안쪽(그림 4 (a)의 점 C_1)과 바깥(그림 4 (a)의 점 C_2)에 동시에 존재할 수 있다. 즉 중심후보영역이 현 $\overline{P_0P_i}$ 을 기준으로 식 (7)과 같이 좌우 A_L 과 A_U 가 동시에 존재하게 된다. 그런데 이 경우에도 가정(3)과 같이 각 중심후보영역들의 꼭짓점을 중심으로 하는 호의 분포가 최적의 호를 기준으로 골고루 분포한다고 가정하면 이들의 평균 오차를 가지는 호의 중심을 구할 수 있다.

만약 그림 4 (b)에서 처럼 두 개의 분리된 중심후보영역 A_L 과 A_U 가 존재하고 이들의 평균중심은 식(18)을 이용하여 얻은 결과를 C_1, C_2 라 하자. 그리고 $Cir(P_0C_1)$ 와 $Cir(P_0C_2)$ 의 반지름을 r_1, r_2 두 원이 만나는 점을 잇는 현 l 의 중심에서 $Cir(P_0C_1)$ 와 $Cir(P_0C_2)$ 에 이르는 높이를 각각 h_1, h_2 라 하면(그림 4 (b) 참조),

$$h_1 = \frac{d^2}{(2r_1 - h_1)}, \text{ 여기서 } d = \frac{l}{2} \quad (19)$$

이다. 두 개의 중심후보영역이 생기는 경우는 각도가 작은 경우이므로 $h_1 \ll 2r_1$ 이라 하면

$$h_1 \approx \frac{d^2}{2r_1} \quad (20)$$

이다. 즉 h_1 은 반지름 r_1 에 반비례한다. 같은 방법으로

$$h_2 \approx \frac{d^2}{2r_2} \quad (21)$$

이며, 두 중심 후보영역 A_L 과 A_U 의 꼭짓점의 수가 각각 m_1, m_2 라 하면 두 영역으로부터 생성된 호의 평균이 되는 위치 h 는

$$h = \frac{(m_1 h_1 - m_2 h_2)}{m_1 + m_2} \approx \frac{d^2}{m_1 + m_2} \left(\frac{m_1}{2r_1} - \frac{m_2}{2r_2} \right) = \frac{d^2}{2r} \quad (22)$$

이고, 여기서 r 은 반지름으로

$$r = \frac{r_1 r_2 (m_1 + m_2)}{m_1 r_2 - m_2 r_1} \quad (23)$$

이다, 그리고 점 P_0 로부터 $\overrightarrow{C_1 C_2}$ 와 수직으로 만나는 점을 M 이라 하면 M 은 현 l 의 중점이다. 따라서 새로운 평균 중심 C 는 $\overrightarrow{C_1 C_2}$ 연장선 위에 존재하므로

$$\overrightarrow{MC} = (h-r)\vec{u} \quad (24)$$

여기서 \vec{u} 는 $\overrightarrow{C_1 C_2}$ 의 단위벡터

가 된다.

2.3 Breakpoint 위치추적

대부분의 디지털화에 의한 잡음은 구하고자하는 호로부터 일정한 범위 내에서 평균오차를 가지지만, 두 개의 호(직선도 무한한 반지름을 가지는 호로 간주)가 만나는 점인 breakpoint를 지나면

그 이후는 다른 호의 일부이므로 현재의 호로부터 거리오차가 점점 더 커지게 된다. 즉 양자화 오차와 같은 일반적인 잡음은 평균 크기를 전후로 커졌다 작아졌다를 반복하지만 breakpoint를 지나는 경우에는 디지털 곡선상의 점과 구해진 호와의 거리오차가 차츰 커지게 된다.

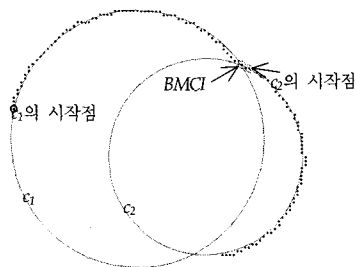
그림 5 (a)는 디지털 카메라로 입력한 동전 두 개 이미지 일부의 윤곽선이다. 그리고 그림 5 (b)는 두 동전의 윤곽선으로부터 앞의 방법을 이용하여 호 c_1 , c_2 의 중심과 반지름을 구하고 두 호로부터 각 점(i 는 점의 위치)의 거리오차를 나타내는 그래프이다. 그림 5 (b)로부터 중심이 다른 두호의 경우 B_{avg} 를 전후로 호와 디지털 곡선상의 점간 오차가 연속적으로 증가하는 것을 알 수 있다.

CI 방법의 경우 오차가 ERR 직전의 점, 즉 $ERR=2$ 인 경우 BCI_1 의 위치에서 breakpoint가 발생하고 $ERR=3$ 인 경우는 BCI_2 의 위치에서 breakpoint가 만들어진다. 그러나 실제 $BMCI$ 는 두 개의 호가 근접하는 위치이며 이 위치는 ERR 의 크기와 거의 무관하다.

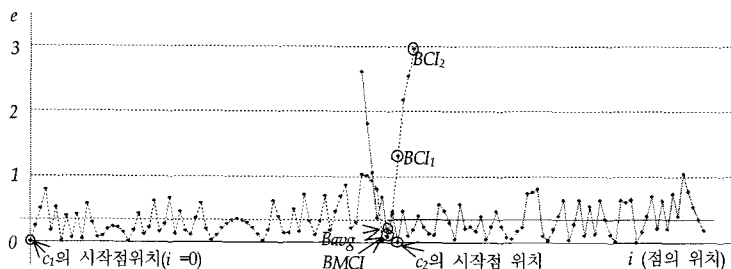
여기서 $BMCI$ 는 두 개의 호가 만나는 점으로

두 개의 호로부터 거리의 합이 가장 짧은 위치이다. 정확한 $BMCI$ 는 보다 정확한 중심과 반지름으로부터 구할 수 있는데, 호의 일부로 간주되는 디지털 곡선상의 점들 중에 오차가 크거나 다른 호의 일부가 포함되는 경우 정확한 중심과 반지름을 구하는 데에 부정적인 영향을 준다. 특히 breakpoint 근처의 점들 중 다른 호의 일부가 편입되는 경우 많은 왜곡을 일으킬 수 있다.

이를 피하기 위해서 제안된 방법에서는 디지털 곡선상의 각 점에 대해서 현재까지 진행된 호(중심과 반지름)를 계산하고, 그 점이 호로부터 거리오차가 평균 이내에 있는 점(이후 B_{avg} 라 함)인지 검사한다. 계속해서 CI의 방법과 같이 breakpoint(이후 BCI 라 함)를 찾으면 마지막 B_{avg} 를 호의 끝점으로 간주한다. 이때 B_{avg} 는 구하고자하는 호에 작은 오차를 가지고 근접해 있는 점이다. 이 이전의 점들이 호의 윤곽선으로부터 일정한 잡음만 가지고 있다면 평균오차 내외에서 호의 근처에 대체로 균등하게 분포되므로 B_{avg} 까지 중심후보 영역은 BCI 까지 중심후보영역에 비해 일그러짐이 덜하게 되어 앞 절의 가정 (3)에 보다 가까워진다.



(a) 두 개의 호와 breakpoint



(b) 거리오차 변화와 주요 특징점

(그림 5) 두 개의 호와 점들 간의 거리오차 변화

다음은 호의 중심과 $Bavg$ 을 찾는 알고리즘이다.

```

GetBavgAndCenter(i)
{
    Ps=Pi; //Ps는 호의 시작점
    Ai = {실수평면 전체영역} //중심후보
    영역 초기화
    while((i≤n) and (Ai≠∅)) { //Pi가 끝점
    (Pn)이 아니고 중심후보영역이 ∅이 아
    니면
        if (|PsPi|>4e) then // 가정 (I) 적용
        {
            중심 C를 구한다; //식 (18) 및
            (24)참조
            e=err(Ps,Pi,C);
            if(e≤평균오차) then
            {
                pBavg=i; //Bavg의 위치
                CMCI=C; //Bavg까지 호의 중심
            }
            오차(e)의 합과 평균오차를 구한다;
        }
        Ai영역을 구한다; //CI 방법 이용
        i=i+1; //다음 점으로 이동
    } //반복 종료
    r=|CMCI PBavg|; //호의 반지름
}
    
```

한편 $Bavg$ 는 c_1 상에 있지만 반드시 c_2 에 포함된다고는 볼 수 없으므로 보다 확실한 c_2 의 시작점을 찾아야한다. 만약 c_2 의 시작점을 breakpoint 후 보라고 하면 이 점은 c_1 과 c_2 에 반드시 포함되어야 한다. 따라서 c_2 에 포함될 가능성이 크면서 c_1 에도 포함될 수 있는 점은 c_1 으로부터 거리가 ERR 이 되기 직전의 점을 c_2 의 시작점으로 한다. 그림 5 (b)의 c_2 의 시작점은 $ERR=2$ 일 때 다음 호의 시작점에 대한 예이다.

두 개의 호가 부드럽게 연결되는 경우 디지털

곡선 상의 점들과 약간의 거리오차에도 많은 breakpoint 위치 차이를 만들 수 있다. 이러한 위치 차이를 완화하기 위해서 두 호 c_1 과 c_2 로부터 오차의 합이 가장 작은 곳 즉, $err(P_0, P_i, C_1) + err(Bavg, P_i, C_2)$ 의 값이 최소가 되는 점 P_i 를 $BMCI$ (그림 5 (b)의 $BMCI$ 참조)로 한다.

다음은 디지털 곡선 Z 로부터 호를 추출하는 알고리즘이다.

```

ArcSegmentation{
    {
        GetBavgAndCenter(0)로 부터 첫 번째 호
        의 중심  $C_1=C_{MCI}$ , 반지름  $r_1=r$  및  $pBavg$ 
        를 구한다;
        while(pBavg<n) //끝점 Pn을 처리할 때까지
        {
             $\overrightarrow{C_1 P_{i-r_1}} \geq ERR$ 인  $P_i(i \geq pBavg)$ 를 찾
            고, 다음 호의 시작점  $P_S=P_{i-1}$ 로 둔다;
            GetBavgAndCenter( $P_S$ )로 부터 다음 호
            의  $C_2=C_{MCI}$ ,  $r_2=r$  및  $pBavg$ 를 구한다;
             $err(P_0, P_i, C_1) + err(P_S, P_i, C_2)$ 의 값이 최소
            가 되는 점을  $BMCI$ 로 한다;
             $c_2$ 의 중심과  $BMCI$ 를 출력한다;
             $C_1=C_2$ ;
             $r_1=r_2$ ;
        }
    }
}
    
```

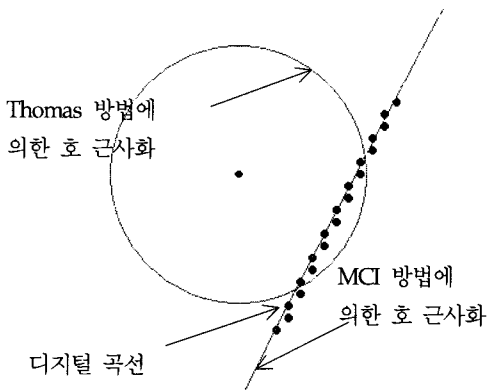
3. 실험 및 결과

3.1 호의 중심에 대한 비교

제안된 breakpoint 위치 추적방법을 적용하기 위해서는 직선을 포함한 모든 범위의 각도에 대해서도 안정적인 오차를 제공하는 호의 추출방법이 필요하다. Chernov[15]는 각도가 작은 호에 대해서 안정적인 결과를 주는 여러 가지 방법들에 대한 비교분석을 하고, 각도가 작아지는 경우 폭

발적으로 계산회수가 증가하는 Landau 방법[12]에 비해 빠르고 안정적인 LSF(Least Square Fitting) 방법을 제시하였다. 그러나 Chernov가 제시한 방법 또한 호의 각도가 0에 가까워지면 반복회수도 급격하게 증가하므로 화소수가 많은 이미지에 적용하기에는 어려운 점이 있다.

한편, LSF 방법을 단순화시켜 빠른 속도를 보이는 Thomas[13] 또는 Wu[14]의 방법들이 있는데 이 방법들은 각도가 작은 경우 추출되는 반지름이 지나치게 짧아지는 것으로 잘 알려져 있다 [15]. 즉, 그림 6과 같이 직선에 가까운 디지털 곡선에 Thomas[13] 방법을 적용하는 경우 거리오차가 매우 크게 나타나는데 이는 오차의 누적을 이용하는 본 방법의 breakpoint추적에는 적절하지 못한 방법이라 할 수 있다.



(그림 6) 디지털 직선에 대한 호 근사화의 예

그러나 CI방법과 제안된 MCI방법에 의한 호는 주어진 디지털 곡선과의 거리오차의 크기가 항상 허용오차 이내에 있고 호의 각도와 무관하게 모든 점에 대해 한번 처리로 중심을 구할 수 있으므로 제안된 breakpoint 추적방법에 적절하다고 할 수 있다. 따라서 이 실험에서는 본 논문에서 제안하는 breakpoint 추출에 적절한 CI 방법과 MCI 방법에 대해서만 비교한다.

MCI 방법에 의한 중심 구하기와 CI 방법을 비교하기 위해 중심의 위치를 (0,0)로 하는 목표 원 (잡음이 없는 원) c 에 대해, 잡음(random noise)의 최대크기 1 (평균오차 0.250, 표준편차 0.286)인 잡음을 가지는 디지털 곡선 Z 에 대해서 breakpoint 위치 결정에 영향이 큰 다음의 파라미터들을 측정하였다(사용 컴퓨터의 CPU Intel pentium IV /133MHZ, 메인메모리 512MB).

- $|e|_{avg}$: 구해진 호와 디지털 곡선상의 점들과의 거리오차 절대값 평균
- $|e|_{max}$: 구해진 호와 디지털 곡선상의 점들과의 최대 거리오차의 절대값
- σ : 거리오차 절대값의 표준편차
- $|\Delta C|_{avg}$: 원 c 의 중심과 구해진 호의 중심과의 거리오차 절대값의 평균

여기서 $ERR=1.5$ 로 하였는데 이것은 추출된 호를 기준으로 $\pm e_{ERR} (<1.5\sqrt{2})$, 즉, 오차 변동폭

(표 1) 호의 각도 크기에 따른 주요 파라미터의 변화 ($r=10$)

| n | θ | CI 방법 | | | | MCI 방법 | | | |
|-----|----------|-------------|----------|-------------|--------------------|-------------|----------|-------------|--------------------|
| | | $ e _{avg}$ | σ | $ e _{max}$ | $ \Delta C _{avg}$ | $ e _{avg}$ | σ | $ e _{max}$ | $ \Delta C _{avg}$ |
| 5 | 0.1π | 0.252 | 0.322 | 1.137 | ∞ | 0.255 | 0.331 | 1.197 | ∞ |
| 10 | 0.2π | 0.328 | 0.410 | 1.249 | ∞ | 0.281 | 0.348 | 1.143 | ∞ |
| 15 | 0.3π | 0.522 | 0.618 | 1.416 | ∞ | 0.310 | 0.380 | 1.147 | 9.599 |
| 20 | 0.4π | 0.652 | 0.762 | 1.497 | ∞ | 0.312 | 0.380 | 1.282 | 3.318 |
| 25 | 0.5π | 0.380 | 0.402 | 1.316 | 3.016 | 0.299 | 0.363 | 1.239 | 0.944 |
| 50 | 1.0π | 0.297 | 0.360 | 1.114 | 0.309 | 0.300 | 0.363 | 1.138 | 0.301 |
| 100 | 2.0π | 0.301 | 0.364 | 1.287 | 0.208 | 0.303 | 0.367 | 1.290 | 0.211 |

최대 $3\sqrt{2}$ 픽셀 미만의 잡음을 허용하는 값으로 디지털화에 의한 오차 및 윤곽선의 애매함에 의한 작은 왜곡 등을 처리할 수 있는 충분한 크기이며, 이후 실험에 여러 가지 크기의 이미지에 대해서 다 같이 적용한 값이다.

표 1은 위에서 언급한 조건을 만족하는 디지털 곡선 1,000개에 대해서 구해진 호와 주어진 디지털 곡선간의 거리오차의 평균, 표준편차, 최대오차 및 중심의 오차를 나타낸다.

표 1에서 CI방법은 호의 각도가 매우 작으면 꼭짓점의 위치가 무한 값을 가지므로 구해진 호가 직선 근사화된다. 따라서 호의 각도가 작을수록 직선 근사화에 의해 오차가 작아지고, 중심후보영역 A의 크기가 무한인 상태에서 호가 커지면 오차가 오히려 증가한다. 그 후 호의 각도가 더 커져서 중심후보영역 A가 유한 크기로 줄어들면 그 이후 오차는 다시 줄어든다.

반면에, MCI방법은 호의 크기(각도)와 상관없이 거의 일정한 평균과 표준오차를 나타내면서 최적의 평균(0.25)에 0.06이내에 근접하는 매우 안정적인 오차를 보여준다. 이 결과는 오차의 누적 및 평균을 이용하는 breakpoint 추적에 MCI 방법이 CI 방법보다 더 유리한 것을 보여준다.

표 1에서 중심이 무한위치가 아닌 유한 범위 내에서 안정적으로 추출되는 각도는 CI 방법의 경우 0.46π 부터, MCI 방법은 0.28π 부터 이루어졌

으며 그 이후는 급속도로 안정되는 것을 알 수 있다.

표 2는 디지털 곡선의 길이와 점의 수를 일정하게 하고 반지름(곡률)을 변화시켰을 때의 결과를 보여준다. 여기서는 대부분 Z가 직선에 가까운 호들로서 중심의 위치의 편차가 워낙 크므로 중심오차의 평균은 의미가 없어 생략하였다.

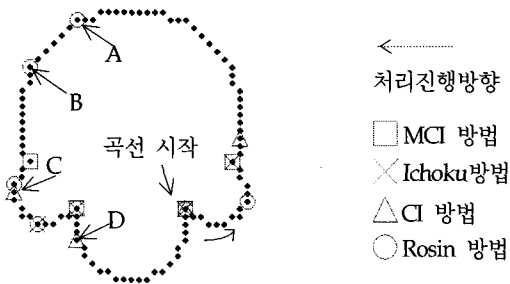
표 2에서 CI 방법에 비해 MCI 방법은 거의 변화 없이 잡음 최대 크기(=1)의 5% 이내의 안정적인 값을 보인다. breakpoint 위치추적에 있어서 B_{avg} 는 평균과의 비교에 의해서 결정되므로 평균 오차의 크기보다 안정성이 더 중요하다고 볼 수 있다.

3.2 breakpoint 추출 실험

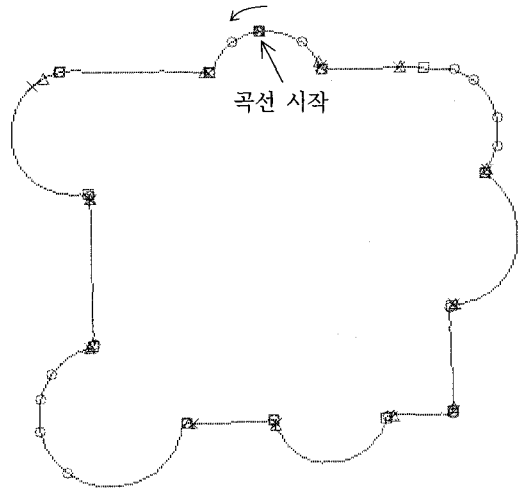
그림 7은 호의 분리 실험을 위해서 직선 또는 곡선의 분리에 많이 인용되는 Teh의 곡선[16](그림 7(a)), 기계도면(7 (b)) 및 동전과 지폐(7 (d)) 등의 이미지로 4~180 픽셀 크기의 반지름을 가지는 호와 직선을 포함한 윤곽선을 가지는 그림이다. 그리고 그림 7 (a), (c) 및 (e)는 각 이미지들에 대한 여러 가지 방법들의 곡선분리 실험결과이다. 실험에 사용된 방법은 Rosin 방법, Ichoku 방법, CI 및 MCI방법 등이며, breakpoint 위치나 수를 추출하지 않거나[1,7], Rosin 방법과 유사한 Dosch 방법[9]은 비교대상에서 제외하였다.

(표 2) 호의 반지름 크기에 따른 주요 파라미터의 변화($n=20$)

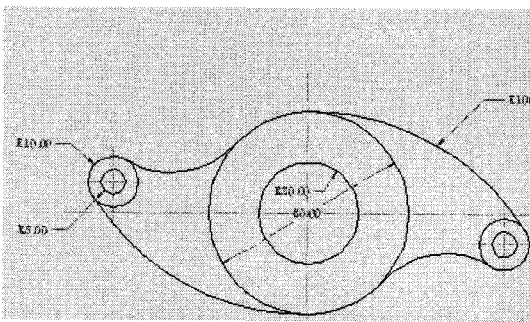
| r | θ | CI 방법 | | | MCI방법 | | |
|----------|-----------|-------------|----------|-------------|-------------|----------|-------------|
| | | $ e _{avg}$ | σ | $ e _{max}$ | $ e _{avg}$ | σ | $ e _{max}$ |
| 10 | 0.4π | 0.652 | 0.762 | 1.497 | 0.312 | 0.379 | 1.282 |
| 20 | 0.2π | 0.488 | 0.579 | 1.393 | 0.286 | 0.349 | 1.057 |
| 30 | 0.16π | 0.380 | 0.462 | 1.261 | 0.277 | 0.337 | 1.055 |
| 40 | 0.1π | 0.338 | 0.413 | 1.185 | 0.273 | 0.331 | 0.960 |
| 50 | 0.08π | 0.302 | 0.372 | 1.131 | 0.268 | 0.327 | 1.026 |
| 100 | 0.04π | 0.270 | 0.331 | 1.014 | 0.268 | 0.326 | 0.956 |
| ∞ | 0.0π | 0.258 | 0.313 | 0.942 | 0.265 | 0.322 | 0.967 |



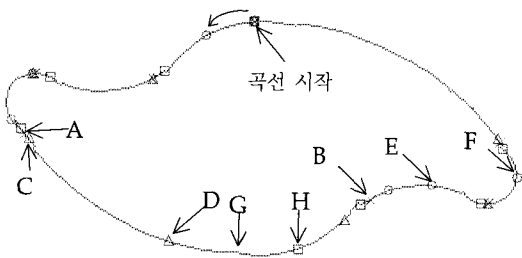
(a) Teh 곡선의 분리결과



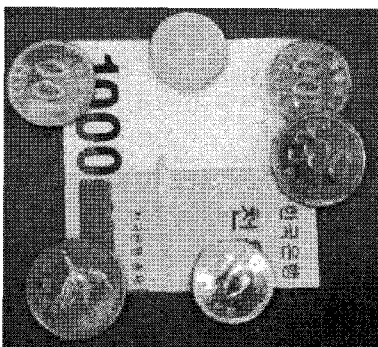
(e) 동전과 지폐의 윤곽선 분리
그림 7 디지털 곡선 분리의 예



(b) 기계도면



(c) 기계도면의 윤곽선 분리



(d) 동전과 지폐

이 실험에서 사용한 임계값은 Ichoku 방법[10]의 경우 저자가 추천한 임계값 범위(Normalized Maximum Square Deviation : NMSD 10-25%, Slope of Regression Line : SRL 1-5%)중에 위의 실험에 가장 좋은 결과를 주는 임계값 NMSD=10%와 SRL=2%를 각각 적용하였고, CI와 MCI 방법은 $ERR=1.5$ 로, Rosin방법[8]은 직선근사화시 최대 거리허용오차는 저자가 지정한 3픽셀이며 곡선의 합병에는 별도의 임계값이 없다.

그림 7 (a)의 Teh의 곡선에 있어서 점 A와 B사이의 직선은 Rosin의 방법에 의해 추출된 직선으로 디지털 곡선과의 거리오차가 0(중요도가 가장 높은 값)이므로 이것은 다른 직선과 합쳐져 호가 될 수 없다. 이와 같이 중요도가 높은 직선의 일부는 입력 곡선 내에 흔히 존재할 수 있고(그림 7 (e) 참조) 이것에 의하여 Rosin의 방법은 많은 breakpoint를 발생시킨다. 그리고 그림 7 (c)에서 점 E와 F사이의 곡선은 볼록 곡선과 오목곡선이 합쳐지는 부분이 직선 처리되면서 곡률의 왜곡으로 호가 잘못 추출된 경우를 보여준다.

한편 CI 방법은 허용오차 이내에 있는 모든 점들을 같은 호의 일부로 간주하므로 그림 7 (a)의

점 C와 D사이의 호와 같이 실제 breakpoint를 지나서 잘못 처리된 호의 경우를 보이는데, 구해진 breakpoint가 대부분 실제 breakpoint를 지나서 존재하는 것을 실험결과로 확인할 수 있다. 그런데 그림 7 (c)의 점 C는 호 \widehat{CD} 의 시작점인데 디지털 곡선 상에서 오차가 큰 지점이므로 D지점에서 미리 breakpoint가 발생하였다.

그리고 Ichoku 방법은 CI방법과 마찬가지로 허용오차 NMSD의 크기에 따라 breakpoint를 지나치는 경우를 볼 수 있는데 다만 이 방법은 CI방법과 달리 곡선의 길이에 따라 신축적으로 적용된다. 즉 길이가 짧은 경우에는 허용 거리오차의 크기가 작아지므로 대부분 CI 방법보다 이상적인 breakpoint에 가까운 점을 찾을 수 있다. 그러나 곡선의 길이가 긴 경우에는 허용 거리오차가 커지므로 그림 7 (c)의 점A, B간의 곡선처럼 실제 breakpoint를 훨씬 지나서 곡선을 분리하게 된다.

끝으로 MCI 방법의 경우 그림 7 (c)에서 실제 breakpoint인 점 G에서 만나는 두 개의 곡선의 곡률 변화가 거의 없으므로 G를 지나서 breakpoint (점 H)가 추출되었다. 이 경우를 제외한 다른 그림에서는 대부분 실제 breakpoint 위치와 구해진 위치가 거의 일치하는 것을 볼 수 있다.

표 3. 그림 7의 디지털 곡선과 추출된 호 또는 직선간의 거리오차평균

| | MCI 방법 | Ichoku 방법 | Rosin 방법 | CI 방법 |
|----------|--------|-----------|----------|-------|
| 그림 7 (a) | 0.364 | 0.573 | 0.654 | 0.562 |
| 그림 7 (c) | 0.536 | 1.030 | 1.717 | 0.781 |
| 그림 7 (e) | 0.413 | 0.969 | 0.497 | 0.723 |

표 3은 각 방법들을 적용하여 구해진 호와 주어진 디지털 곡선간의 평균거리오차로서 MCI방법이 비교적 정확한 breakpoint를 추적함으로써 다른 방법들 보다 적은 오차를 가짐을 확인할 수 있다.

4. 결론

데이터 압축을 위해서 곡선을 근사화하는 데에는 breakpoint 위치가 크게 문제되지 않으며 추출된 호와 주어진 곡선간의 거리 오차와 breakpoint의 수가 중요하다고 볼 수 있다. 그러나 도형이나 물체인식에 있어서 breakpoint의 위치는 모양을 결정짓는데 매우 중요하며 이들의 위치는 물체의 위치와 크기, 즉, 호의 중심, 반지름 및 호의 각도를 결정하는 중요한 파라미터다.

본 방법에서는 CI방법을 개선하여 LSF[15]를 이용하지 않으면서 반지름의 크기에 상관없이 안정적인 오차를 가지는 중심구하기 방법을 제시하고, 이를 이용하여 직선을 포함하는 곡선에 대해서도 직선을 위한 별도의 프로세스가 없이 곡선을 분리할 수 있도록 하는 방법을 제시하였다.

제안된 방법은 절대거리오차에 대한 임계값을 사용하고, breakpoint를 기준으로 연결된 호가 만나는 점에 가까운 breakpoint를 찾음으로써 기존의 방법들보다 breakpoint의 위치가 정확하고, 항상 구해진 호와 주어진 곡선간의 거리가 허용오차 이내를 만족하면서 거리오차 평균이 다른 방법들에 비해 적음을 많은 실험으로 확인하였다.

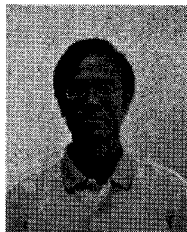
본 방법은 breakpoint의 위치와 수 그리고 주어진 곡선과 추출된 호 간의 거리오차 등에 있어서 좋은 결과를 보이므로 곡선의 근사화나 물체의 인식에 있어서 유용하게 사용될 수 있을 것으로 기대된다.

참고문헌

- [1] Pei, S. C. and Horng, J. H., "Circular arc detection based on Hough transform," Pattern Recognition Letters 16, pp.615-625, 1995.
- [2] Ho, C. H. and Chen, L. H., "A Fast ellipse /circle detector using geometric symmetry" Pattern Recognition, Vol.28, No.1, pp.117-124, 1995.

- [3] Kim, H. S. and Kim, J. H., "A two-step circle detection algorithm from the intersecting chords," Pattern Recognition Letter 22, pp.627-636, 2001.
- [4] Chju, S.H. and Liaw J.J., "An effective voting method for circle detection," Pattern Recognition Letter 26, pp.121-133, 2005.
- [5] Dori, D. and Liu, W., "Stepwise recovery of arc segmentation in complex line environments" International Journal on Document Analysis and Recognition, pp.62-71, 1998.
- [6] Song, J., Lyu, M. R. and Shijie C., "Effective multiresolution arc segmentation: algorithms and performance evaluation" Pattern Analysis and Machine Intelligence, IEEE Transaction on, Vol.26, No.11, 2004.
- [7] Lim K. B., Xin K. and Hong G. S., "Detection and estimation of circular arc segments," Pattern Recognition Letters 16, pp.627-636, 1995.
- [8] Rosin, P. and West, G., "Segmentation of edges into lines and arcs" Image and Vision Computing Vol.7, No.2, pp.109-114, 1989.
- [9] Dosch, P., Masini, G. and Tombre, K., "Improving arc detection in graphics recognition," 15th International Conference on Pattern Recognition Vol.2, pp.2243-2246, 2000.
- [10] Ichoku, C., Defontaine, B. and Chorowicz, J., "Segmentation of digital plane curves: a dynamic focusing approaching", Recognition Letter 26, pp.741-750, 1995.
- [11] 류승필, "허용오차를 만족하는 호의 추출," 정보과학회논문지 : 소프트웨어 및 응용, 제 32권 9호, 2005.
- [12] Landau, U. M., "Estimation of a circular arc center and its radius" Computer Vision, Graphics, and Image processing 38, pp.317-326, 1987.
- [13] Thomas, S. M. and Chan, Y. T., "A simple approach for the estimation of circular arc center and its radius" Computer Vision, Graphics, and Image processing 45, pp.362-370, 1989.
- [14] Wu, Z., Wu, L. and Wu, A., "The robust algorithm finding the center of an arc," Computer Vision and Image Understanding, Vol. 62, No. 3, PP269-278, 1995.
- [15] Chernov, N. and Lesort, C., "Least square fitting of circles," Journal of Mathematical Imaging and Vision 23, pp239-252, 2005
- [16] Teh, C. and Chin, R., "On the detection of dominant points on digital curves," IEEE Trans. PAMI., 11(8), pp859-872, 1989.

○ 저 자 소 개 ○



류 승 필(Sung-Pil Lyu)

1979년 서울대학교 전자공학과 졸업 (학사)
 1980~1993 한국원자력연구소 계측제어연구실 선임 연구원
 1987년 충남대학교 대학원 전자공학과 졸업 (석사)
 1991년 충남대학교 대학원 전자공학과 졸업 (박사)
 1993~현재 세명대학교 컴퓨터학부 부교수
 관심분야 : 패턴인식, 인공지능, 시뮬레이션
 E-mail : lsp415@semyung.ac.kr