

Ball 곡선을 이용한 곡선 Fitting 알고리즘

이 아 리[†]·최 영 근^{††}

요 약

본 논문에서는 Ball 곡선 함수식을 이용한 fitting 알고리즘을 제안한다. 제안한 방법은 기존의 Ball 함수식이 아닌 연속적인 3차 Ball 함수식에서 데이터 집합을 fitting 할 수 있는 재귀적인 방법이다. 전체적인 알고리즘은 2단계이다. 첫 번째 단계는 주어진 데이터 집합에서 점을 선택하고, 선택된 점에서 접점, 모서리 점을 분리해낸다. 두 번째 단계는 곡선 fitting 단계이다. 곡선 fitting을 하기 위한 함수식은 Ball 곡선 함수식을 이용한다. 또한 절점(knot)을 삽입하는 가중치 최소자승법은 ball 곡선의 조각곡선(Piecewise Ball Curve)를 효과적으로 만들기 위한 방법이고, 2개 이상으로 결합된 Ball 곡선 세그먼트는 모든 합쳐지는 점에서 부드럽게 연결되어진다. 제안한 알고리즘은 폰트나 GIS와 같은 이미지 표현을 위해서 사용될 수 있다.

Curve Fitting with Recursive Ball Curve

Ah-Ri Lee[†] · Young-Keun Choi^{††}

ABSTRACT

In this paper, we present a curve fitting algorithm using a ball curve. Our algorithm is recursive method for fitting, which is not a traditional ball function but a continuous ball function. This algorithm consists of two steps. The first step, it is classified the composite corner points to joint points until selected from the given data set. The second step is the curve fitting. The basis function for curve fitting is use to ball function. Also, the weighted least square method, to insert knot, is an efficient method for piecewise ball curve and ball curve segments will be smoothly connected at all composit points. The proposed algorithm will be applied to represent image representation, like fonts, digital image and GIS.

키워드 : 곡선/곡면 근사(curve/surface fitting), 보간(interpolation), approximation, Ball curve, B ezier

1. 서 론

곡선 fitting은 이미지 프로세싱, 컴퓨터 그래픽스, 패턴 인식, CAGD와 같은 응용 분야에서 사용되고, 이미지의 외곽선으로 표현되는 모든 분야에서 사용되고 응용된다.

곡선 fitting은 보간(interpolation)과 근사(approximation)을 기본으로 한다. 곡선 함수식으로 Bezier 곡선과 B-spline 곡선 함수식은 곡선/곡면 fitting에 많이 사용되어진다. 응용 분야에 있어서 특정한 데이터 점을 지나는 곡선을 생성 시, 최소의 뒤틀림(distortion)으로 이미지의 외곽선을 근사하기 위하여, 3차 조각곡선을 최소의 수로 이용하는 것이다. 제안한 방법은 모서리점과 접점에서 곡선세그먼트 fitting방법이다. 제안된 데이터 보간에 의한 날카롭게 생성된 외곽선의 기술키가 연속이지 않는 데이터 또는 이미지의 외곽선은 곡선 세그먼트의 끝점과 끝점을 연결하는 3차 곡선의 한 부분으로 구성되

어있다. 곡선의 한 세그먼트에서의 만나는 접점, 끝점, 모서리점은 절점(knot)의 위치와 이미지의 외곽선을 부드럽게 표현하는 한 방법이다. 기존의 방법은 세그먼트의 한 부분이 가능하면 길게 연결을 시켜서 전체 데이터 집합을 fitting 하는 방법이다[3]. 또한 분할되는 분할점(breaking point)을 정의 및 생성하여, 그 점을 기점으로 데이터 점 각각의 부분집합이 주어진 오차 범위하에서 곡선을 fitting하는 방법이다[4]. 그러나 이러한 방법은 기술키값이 균등하게 분포되어야 하는 단점을 가진다. 이론적으로 가장 좋은 해결 방법은 절점을 정렬하는 방법이 연구되었는데, Liu, Srimath[7]는 연결코드(chain-code)로 6개의 모서리점을 찾는 방법을 제안하였다. BeusTiu[10] 알고리즘은 freeman Davis의 알고리즘을 더욱 발전시켰다. Shao [11, 12]는 BeusTiu의 알고리즘을 응용하여 많은 이미지 상에서 모서리점을 찾는 알고리즘을 연구하였다. 그러나 이러한 알고리즘은 모서리점과 접점에서 각각 점을 찾는 알고리즘으로서, 균일하지 않는 모서리점과 접점을 찾지 못하는 단점이 있었다.

본 논문에선 제안한 알고리즘은 크게 2단계이다. 1단계는

※ 이 논문은 2000년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

† 준 회 원 : 광운대학교 대학원 컴퓨터학과

†† 정 회 원 : 광운대학교 전자계산학과 교수

논문접수 : 2000년 7월 24일, 심사완료 : 2001년 3월 16일

임계점 정의 단계, 2단계는 곡선 fitting 단계이다.

1단계에서는 각각의 점에서 곡선의 곡률을 계산하고, 이러한 과정에서 모서리점과 접점등을 임계점으로 정의하고, 임계점(critical point)들 사이의 점들은 Ball 곡선 fitting 의 부분집합으로 사용된다. 여기서 모서리점은 기울기 연속성 (Tangent Continuity)을 위해서 인접하는 부분에 한 쌍의 knot를 삽입한다. 임계점이 결정되면 2단계에서는 ball 함수식을 이용하여 곡선 fitting한다. 그러므로 결과적으로 2개 이상으로 결합된 ball 곡선 세그먼트는 모든 합쳐지는 점에서 부드럽게 연결되어진다. 이러한 방법은 폰트나 GIS와 같은 이미지 표현방법으로 사용될 수 있다.

2. Ball 함수식의 정의

Ball 곡선은 British Aircraft Corporation에서 로프팅 곡면 프로그램으로서 CONSUR로 정의된 계수다항식을 사용한다. Ball곡선의 특징은 차수 증가/감소의 효율적인 Ball곡선은 디자이너들에게 있어서 충분한 유연성과 형태보존성을 제공한다[6]. Ball곡선은 Casteljau 알고리즘과는 다른 계수다항식을 사용한다. 함수식은 3차 다항식으로 표현되므로 Bezier 방법에 Bernstein 다항식과는 차이가 있다. 계수다항식은 다음과 같다.

$$\beta_i^m(t) = \binom{m+i}{m} t^i (1-t)^{m+1} \quad (0 \leq i \leq m) \quad (1)$$

$$\beta_{2m+1-i}^m(t) = \beta_i^m(1-t) \quad (2)$$

식 (1)(2)는 차수가 $m+i+1$ 일 때의 생성된 기본적인 함수식 β_i^m 이다.

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 1 & -2 & 1 & 0 \\ 0 & 2 & -4 & 2 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix} \quad (3)$$

식 (3)을 이용해서, 평면 위의 점 집합 V_i 일 때, 3차 Ball 곡선의 함수식은 다음과 같다. $\beta_i(t)$: ball 곡선 계수다항식 ($0 \leq i \leq 3$)

$$B_3(t) = \sum_{i=0}^3 V_i \beta_i(t) \quad (0 \leq t \leq 1) \quad (4)$$

Ball 곡선 $B_3(t)$ 는 조정점 V_i 와 V_{i+1} 를 결합시킨 형태의 조정다각형으로 근사된다. 또한 식은 다음과 같은 조건을 이용함으로써 형태보존성의 성질을 가진다[15].

$$\beta_i(t) \geq 0, \quad i = 0, 1, 2, \dots \quad (5)$$

$$\sum_{i=0}^3 \beta_i(t) = 1 \quad (6)$$

이때, 식 (5)의 성질은 Ball 곡선 $B_3(t)$ 가 조정점 V_i 에 의하여 컨벡스 조합(convex combination)이라는 것을 의미한다. 따라서 Ball곡선은 조정다각형의 컨벡스 헐 안에서 존재한다. 식 (2)에서 조정점이 $V_1 = V_2$ 일 때, 식 (3)은 2차 곡선으로 표현된다. 3차 Ball 곡선을 bicubic곡면으로 확장된다. 이러한 방법은 Bezier 곡선으로부터 Bezier곡면으로 확장하는 방법과 유사하게 표현된다.

3. 곡선 fitting 알고리즘

3.1 임계점(critical point) 정의

본 논문에서 임계점은 모서리 점과 접점으로 정의된다. 모서리 점과 접점은 각각 연속성과 불연속성을 가지는 점이다. 또한 접 점은 큰 곡률의 변화를 가지는 점이다.

곡선의 함수식 $y=f(x)$ 라 할 때, 점 (x, y) 에서의 곡률 k 는 다음과 같다.

$$k = \frac{y''}{\sqrt{(1+(y')^2)^3}} \quad (7)$$

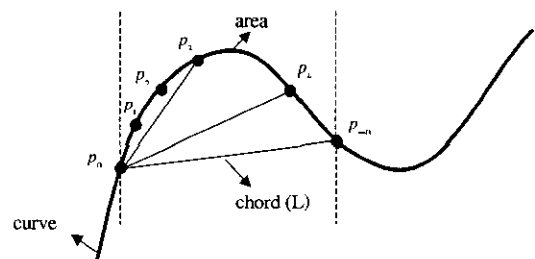
임계점은 최대의 곡률값을 갖는 점이거나, 곡률값이 급변하는 점이다.

3.1.1 영역 정의(area setting)

고정된 현의 길이(fixed-length chord)는 곡선 위의 데이터 점을 따라서 움직이고, 현(chord)과 연결된 곡선사이의 영역을 측정한다. 임계점과 근사한 위치로 계산한다.

```
do (if  $\overline{P_0P_i} \geq L$ 
then Choose  $\overline{P_0P_i}$ ;
else  $i++$ ;)
while ( $i \leq m_0$ );
```

곡선은 n 개의 점을 포함한다고 가정하고, P_0, P_1, \dots, P_{n-1} 로 표현한다. 고정된 현의 길이는 L 이고, 시작점은 P_0 에서 P_0P_1 의 현의 길이를 계산한다. 만약 P_0P_1 이 L 보다 작다면 P_0P_2 의 길이를 계산하고, P_0P_2 가 L 보다 작은 값이면 다음의 점 P_3 점으로 이동한다.



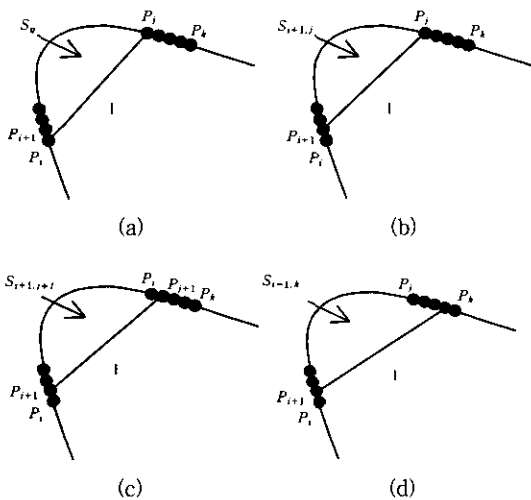
(그림 1) 고정된 현의 길이에 의한 영역 계산

이러한 과정은 P_0P_{m0} 의 현의 길이가 L 보다 크거나 같을 때까지 진행되고, P_{m0} 마지막 점일 때까지 계속 탐색한다. 현 P_0P_{m0} 와 연결된 곡선 부분(piece)사이의 영역을 계산한다. 점 P_k 의 좌표가 $P_k(x_k, y_k)$ 라면, 계산하는 영역은 다음과 같은 식이다.

$$S_{0i} = \begin{vmatrix} x_0 & x_1 & x_2 & \dots & x_i & x_0 \\ y_0 & y_1 & y_2 & \dots & y_i & y_0 \end{vmatrix} \quad (8)$$

이것은 다음과 같이 확장될 수 있다.

$$S_{0i} = \sum_{\beta=0}^i (x_{\beta}y_{\beta+1} - x_{\beta+1}y_{\beta}) + x_iy_0 - x_0y_i \quad (9)$$



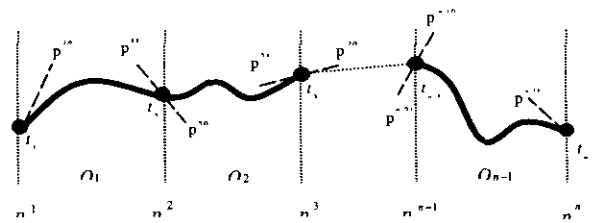
(그림 2) 현과 연결된 곡선의 영역 계산

$$\begin{aligned} \frac{\overline{P_iP_j}}{\overline{P_{i+1}P_j}} &\rightarrow S_{i,j} \\ \frac{\overline{P_{i+1}P_j}}{\overline{P_{i+1}P_{j+1}}} &\rightarrow S_{i+1,j} \\ \frac{\overline{P_{i+1}P_{j+1}}}{\overline{P_{i+1}P_k}} &\rightarrow S_{i+1,j+1} \end{aligned} \quad (10)$$

P_0 에서 P_1 으로 시작점을 이동하고, 끝점 P_{m1} 을 찾는다. P_i 에서 P_j 까지의 점을 가지는 현-곡선 영역 $S_{i,j}$ 는 (그림 2) (a)와 같이 계산되기 때문에, (그림 2) (d)와 같이 P_{i+1} 에서 P_k 까지의 점인 현-곡선 영역 $S_{i+1,k}$ 를 계산한다. 시작점은 P_i 에서 P_{i+1} 으로 한 단계씩 움직임으로서 영역이 설정된다.

3.1.2 임계점 탐색

접점과 모서리점을 탐색하기 위해서, 고정된 길이의 현과 연결된 곡선사이의 영역을 계산한다. 영역의 값은 P_0P_{m0} , P_1P_{m1} , P_2P_{m2} , ..., P_nP_{mn} 의 현에 대응되는 값이다. $P_iP_{mi} - P_{i-1}P_{mi-1}$ 현의 영역차이 값이다. 영역의 값이 P_kP_{mk} 에서 최대 값이고 영역 값이 V_C 의 임계값에 의해서 미리 정의한 것보다 클 경우, 그 값이 모서리 점으로 선택되고 접점은 임의로 위치로 설정한다. 곡률 절대값 d_i 는 연속된 곡선 세그먼트 C_{i-1}, C_i, C_{i+1} 에 의해서 값이 변한다. 여기서 C_{i-1}, C_i, C_{i+1}

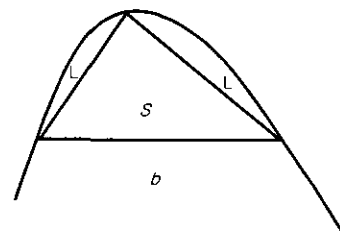


P_1, P_n : 2개의 모서리점
 $P_i (i \in [2, n-1])$: 접점
 $Q_i (i \in [1, n-1])$: Ball curve Piecewise
 $P_0, P_n (i \in [1, n-1])$: Ball curve Piecewise Q_i 2개의 내부 조정점

(그림 3) Ball곡선에서 곡부성을 이용한 곡선 fitting

는 각각의 곡선세그먼트이다. C_{i-1}, C_i, C_{i+1} 은 L_{i-1}, L_i, L_{i+1} 인 현의 길이에 따라 변한다. d_i 는 미리 정한 V_j 값보다 크다. 최대의 영역 편차(deviation)을 가지는 곡선 세그먼트에서 접점을 탐색하고, 접점의 임의의 위치에서 찾아내는 과정은 다음과 같다.

- [1단계] 영역을 계산한다.
- [2단계] 각각의 구역 내에서 가중치에 대한 최대 편차를 모두 구한다.
- [3단계] threshold 값 V_j 값보다 큰 값을 찾아내고 그것을 선택한다.
- [4단계] 곡선 내에서 joint point의 개수를 찾아낸다.



(그림 4) 한 점에서의 방향 변화를 계산

(그림 4)에서 L 은 고정된 현의 길이이고 S 는 $\triangle LBL$ 의 영역이다. E 는 $E = \frac{S}{b}$ 이다.

현의 길이 L 의 값은 선택된 곡선에서의 세그먼트 중간점이다. 곡선의 세그먼트 임계점은 곡선 세그먼트 위의 점을 선택한다. 또한 2개 현의 길이는 L 과 같이 선택되고, L 의 값에 의해서 S 는 2개의 현에서 생긴 삼각형 영역이다. E 는 2개의 현에 의해서 생성되어 그 값이 탐색되는데, E 의 최대값인 위치에서의 모서리점과 접점을 찾아낸다.

3.2 가중치 최소 자승법(Weighted Least Square method)

fitting을 더욱 효과적으로 하기 위해서, 곡률 값은 데이터 집합에서 기존의 최소자승법이 아닌 가중치를 적용한 최소 자승법을 제안한다. 가중치 최소자승법은 비선형데이터 문제와 곡선 세그먼트의 날카로운 부분과 편평한 부분의

문제점을 해결하기 한 방법이다. 제안한 방법은 가중치 w_{ij} 를 각각 데이터 점 $C_i(j)$ 에 삽입한다. 방정식은 다음과 같다.

$$S = \sum_{i=1}^{n-1} \sum_{j=1}^{m_i} [Q_i(t_{ij}) - w_{ij}C_i(j)]^2 \quad (11)$$

$$\alpha_{i,0}, \alpha_{i,1} \quad (i \in [1, n-1])$$

3.3 Ball 함수의 곡선 fitting

조각 Ball 곡선의 끝점 P_i 에서 단위 기울기 벡터(Unit Tangent Vector)일 때, $(n-1)$ 개의 조각 Ball 곡선 Q_i 는 $P_i (i \in [2, n-1])$ 에서의 기울기는 연속이다.

다음과 같은 최소의 값 S에서 Q_i 를 찾아낸다.

$$S = \sum_{i=1}^{n-1} \sum_{j=1}^{m_i} [Q_i(t_{ij}) - C_i(j)]^2 \quad (12)$$

m_i 는 곡선의 i 번째 부분에 의해서 fitting되는 점의 개수이다. $C_i(j)$ 는 Ball 함수식에서 i 번째 부분에서 fitting되는 j 번째 데이터 점이다. 3차 Ball 함수식에서의 i 번째 부분 세 그먼트의 수식은 다음과 같다.

$$Q_i(t_{ij}) = P_i B_{13}(t_{ij}) + P_{i0} B_{23}(t_{ij}) + P_{i1} B_{33}(t_{ij}) + P_{i+1} B_{43}(t_{ij})$$

$$B_{13} = (1 - t_{ij})^3$$

$$B_{23} = 3(1 - t_{ij})^2 t_{ij} \quad t_{ij} \in [0, 1]$$

$$B_{33} = 3(1 - t_{ij}) t_{ij}^2$$

$$B_{43} = t_{ij}^3 \quad (13)$$

곡선 $Q_i(t_{ij})$ 의 내부조정점은 다음에 의해서 정의된다.

$$P_{i0} = P_i + \alpha_{i0} \hat{t}_i$$

$$P_{i1} = P_{i+1} - \alpha_{i1} \hat{t}_{i+1} \quad (14)$$

α_{i0}, α_{i1} 는 임의의 상수이다(scalar value). 이것은 조정점 P_{i0}, P_{i1} 의 위치를 명시하기 위해서 사용되는데, 각각의 \hat{t}_i, \hat{t}_{i+1} 에 의해서 정의된다. 식 (13)에서 $Q_i(t_{ij}) - C_i(j)$ 를 확장시키면 다음과 같다.

$$A_i(t_{ij}) = P_i B_{13}(t_{ij}) + P_{i0} B_{23}(t_{ij}) + P_{i1} B_{33}(t_{ij}) + P_{i+1} B_{43}(t_{ij}) - C_i(j) \quad (15)$$

그리고 함수식 S는 다음과 같다.

$$S = \sum_{i=1}^{n-1} \sum_{j=1}^{m_i} [A_i(t_{ij}) + \alpha_{i0} \hat{t}_i B_{23}(t_{ij}) - \alpha_{i1} \hat{t}_{i+1} B_{33}(t_{ij})]^2$$

$$i \in [1, n-1]$$

$$\alpha_{i0}, \alpha_{i1} \quad (i \in [1, n-1]) \quad (16)$$

$$\hat{t}_i \quad (i \in [1, n]) : \text{최소 S를 구하는 기울기 벡터}$$

각각의 기울기 벡터 \hat{t}_i 를 이용하여 선형(linear) 형태로 변화시키고 최소자승법을 이용한다. 그리고 기울기 벡터 $\hat{t}_i (i \in [1, n])$ 를 초기값으로 하고, $\alpha_{i0}, \alpha_{i1} (i \in [1, n])$ 을 이용

해서 최소자승법을 반복적으로 계산한다. 그러므로 각각의 점과 모서리점 $\hat{t}_i (i \in [1, n])$ 에서 최소의 기울기 벡터와 최적의 $\alpha_{i0}, \alpha_{i1} (i \in [1, n])$ 는 최적의 스칼라 값이 모두 구해질 수 있다.

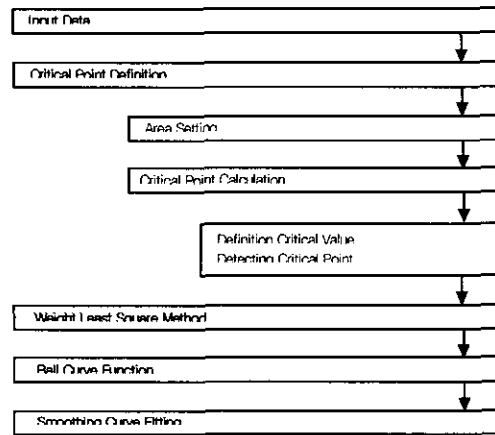
$$\frac{\partial S}{\partial \alpha_{i0}} = 0, \frac{\partial S}{\partial \alpha_{i1}} = 0, \quad i \in [1, n-1] \quad (17)$$

수식은 식 (16)(17)을 이용한다.

$$2 \sum_{j=1}^{m_i} \hat{t}_i B_{23}(t_{ij}) [A_i(t_{ij}) + \alpha_{i0} \hat{t}_i B_{23}(t_{ij}) - \alpha_{i1} \hat{t}_{i+1} B_{33}(t_{ij})] = 0, \quad i \in [1, n-1] \quad (18)$$

3.4 전체적인 알고리즘

곡선 fitting에 관한 알고리즘의 구조는 다음과 같다.



(그림 5) 알고리즘 구조

Input : Corner Point, Joint Point of Data Point Set
MaxPointError : error margin
MaxNumberLoop : maximum loop count
FitError : critical value error
Output : Smoothing Piecewise Cubic ball Curve by selected Control Points

Begin
 Break the data set in each segments and based on the corner point, joint point ;
For each segment Do
Begin
 Compute initial \hat{t}_i values for each corner and joint points
 Compute initial t_{ij} values for all the data points
Do
 Compute α_{i0}, α_{i1} for each ball curve piece Q_i
 Compute t_{ij} values for each data points
 Compute \hat{t}_i values for each corner point and joint point
 Compute maximum fitting error and joint point
 Compute maximum fitting error

```

FitError between  $C_i(j)$  and  $Q_i$ ;
    Increase the number of iteration step
NumberLoop :
    Until (FitError < MaxPointError)
    or (NumberLoop > MaxNumberLoop)
End ;
End ;
    
```

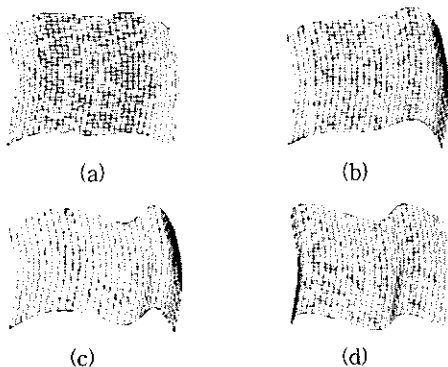
(알고리즘 : Recursive Ball Curve Fitting)

4. 실행 결과 및 분석

제안한 알고리즘은 크게 2가지 방법으로 나누어진다. 임계 점 정의단계와 곡선 fitting 단계로 나누어진다. 임계 점 정의 단계는 각각의 점에서 곡선의 곡률을 계산하고, 이러한 과정에서 모서리 점과 접 점등을 임계 점으로 정의하고, 임계 점 (critical point)들 사이의 점들은 재귀적인 Ball 곡선 fitting의 부분집합으로 사용된다. 여기서 모서리 점은 기울기 연속성을 위해서 인접하는 부분에 한 쌍의 점점을 삽입한다. 임계점으로 데이터 집합이 설정되면, 기존의 곡선 함수식인 Bezier, B-Spline 함수식과 다른 형태보존성을 가지는 Ball 함수식을 이용하여 곡선 fitting한다. 곡선 fitting은 2개 이상으로 결합된 Ball 곡선 세그먼트는 모든 합쳐지는 점에서 부드럽게 연결되어진다.

실행 결과는 VC++과 OpenGL 함수를 이용하였다. (그림 6)은 곡선 Fitting 초기 곡선과 결과곡선이 존재한다. 즉, 병 모양의 형태는 곡선이 fitting 되는 과정에서 다양하게 표현된다. (그림 6) (a)는 초기의 곡선으로 (d)로서 곡선이 fitting 되는 동안 (b)(c)의 과정이 표현된다. (그림 7) (b)(c)의 그림

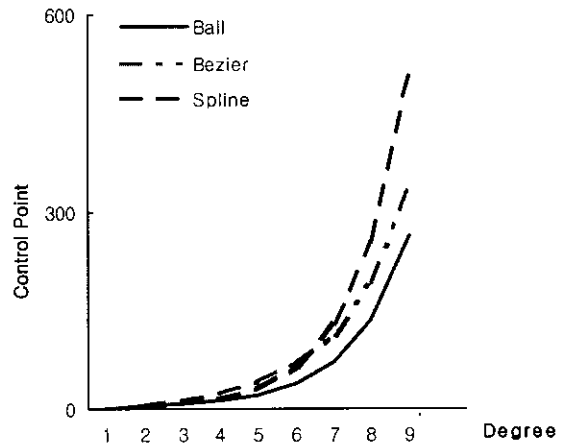
(a) (b) (c) (d)
(그림 6 (a)(b)(c)(d)) 곡선 fitting 과정



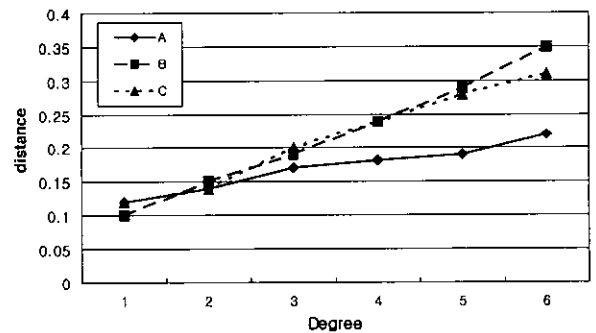
(그림 7 (a)(b)(c)(d)) 곡면 fitting 과정

또한 곡면의 fitting 과정으로 중간과정의 곡면이 생성된다. 그러므로 곡선/곡면 fitting 과정에서 중간단계의 곡선/곡면이 생성됨으로서 다양하게 곡선/곡면을 표현할 수 있다.

(그림 8)은 제안한 알고리즘의 Ball 함수와 기존의 곡선 함수식인 Bezier, Spline, Ball 함수식의 조정점을 비교한다. 제안한 fitting 방법은 임계값과 임계점을 이용하고 재귀적인 Ball 함수식으로서 적은 데이터로서 곡선 fitting된다. 각각의 조정점은 $(2^n + n)$ 개로 감소된다. 특히 차수가 3차 이상 일 때, 제안한 알고리즘에서 적은 데이터로 곡선 fitting이 가능하다. (그림 9)는 결과곡선에 더욱 근사하도록 곡선/

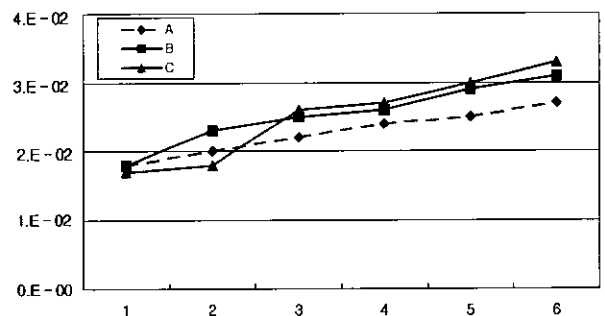


(그림 8) 곡선 함수식에서의 차수와 조정점



(A : 제안된 fitting algorithm, B : Said method C : Xuefu method)

(그림 9) 기존의 Fitting 방법과의 거리 비교



(A : 제안된 fitting algorithm, B : Said method C : Xuefu method)

(그림 10) 기존의 Fitting 방법과의 오차율 비교

곡면이 fitting된다. 기존의 방법과 제안한 방법을 비교하면 차수가 3차 이하일때는 기존의 방법과 차이가 없거나, 또는 오차의 간격이 크게 나타나고, 변형된 거리의 간격 또한 크게 나타난다. 그러나 3차 이상일 때의 곡선 함수식에서는 효과적으로 표현된다. 기존의 방법과 비교한 (그림 10)은 곡선의 차수에 따라서 오차율에 관한 비교이다. 제안된 방법, Said[15], Xuefu[16]방법을 비교했을 때, 차수가 2차인 경우 2.E-02, 3.E-02, 3.E-02로 각각 표현된다. 여기서 제안된 방법은 Ball 함수의 속성만을 이용한 방법은 제안한 방법보다 오차율이 크고, Xuefu방법에서는 근사/보간 방법을 이용함으로써 오차율의 간격이 작아진다. 그러나 3차식 또는 그 이상의 차수를 비교 분석하였을 때, 제안된 방법은 기존의 방법보다 오차의 간격이 감소하게 표현된다.

5. 결 론

본 논문에서는 곡선 fitting 알고리즘을 제안하였다. 제안한 방법은 기존의 문제점인 데이터의 비선형의 곡선 fitting이 표현된다. 또한 각각의 기울기 벡터와 가중치 최소자승법(Weight Least Square)으로 기존의 곡선 표현의 문제점을 해결하였다. 제안한 방법은 형태보존성을 가지는 ball 함수를 이용하여 적은 데이터로서 곡선 fitting 방법이다. 그리고 임계점인 데이터 점 집합으로부터 모서리 점과 점점을 임계값을 설정하는 방법과 정확한 표현을 위하여 가중치 최소자승법으로 표현되는 곡선 fitting 방법이다. 제안한 방법은 폰트, GIS와 디지털 이미지 표현을 위해서 사용될 수 있다.

참 고 문 헌

[1] Buxton. W. A. S. There's more to intersection than meets the eye : Some issues in manual input, In User Centered System Design : New Perspectives on Human-Computer Interaction, pp.219-337, 1986.
 [2] Byoung K. Choi, Surface Modeling for CAD/CAM, Elsevier, 1991.
 [3] D. Ansari and E. J Delp, On detecting Dominant Points, Pattern Recognition 24(5), pp.441-451, 1991.
 [4] D. Ansari and K. W. Huang, Non-parametric dominant point detection, Pattern Recognition 24(9), pp.849-862, 1991.
 [5] Gerald. Farin. Curves and surfaces for CAGD. Academic Press, INC, 1990.
 [6] H. B. SAID, A generalized Ball curve and its recursive algorithm. ACM Transactions on Graphics, Vol.8, No.4, pp. 360-371, October, 1989.

[7] H. C. Liu and M. D. Srimath, Corner detection from chain-code, Pattern Recognition, 3(12), pp.51-68, 1990.
 [8] H. Prautzsch, Charles A, Computing curves invariant under halving. CAGD, Vol.4, pp.133-140. 1987.
 [9] J. C. Ferguson and S. Pruess, "Shape-preserving interpolation by parametric piecewise cubic polynomial," CAD, Vol.23, No.7, pp.498-504, September 1991.
 [10] L. Beus and S. S. H. Tiu, An improved corner detection algorithm based on chain-code plane curves, Pattern Recognition 20(3), pp.291-296, 1987.
 [11] L. Shao and Hao Zhou, Curve fitting with Bezier cubics, Graphical Models and Image Processing, Vol.58, No.3, pp. 223-232, May, 1996.
 [12] L. Shao, C. L. Kai, and T. M. Shyan, On corner detection from chain code, in Proc. of 5th. System Applications, pp. 15-17, Mar, 1995.
 [13] Michael E. Mortenson, Geometric Modeling, Wiley, 1997
 [14] Shin-Min Hu, Guo-Zhao Wang and Tong-Guang Jin, Properties of two types of generalized Ball curves, CAD, Vol.28, No.2, pp.125-133, 1996.
 [15] T. N. T. Goodman and H. B. Said, Properties of generalized Ball curves and Surfaces, CAD, Vol.23, No.8, pp.554-560, October 1991.
 [16] Xuefu Wang, Fuhua (Frank) Cheng and Brian A Barsky, Energy and B-spline interproximation, CAD, Vol.29, No.7, pp.485-496, 1997.



이 아 리

e-mail : ahri@cs.kwangwoon.ac.kr
 1993년 광운대학교 전자계산학과(이학사)
 1996년 광운대학교 대학원 전자계 산학과
 (이학석사)
 1997년~현재 광운대학교 대학원 컴퓨터과학
 과 박사과정
 관심분야 : 컴퓨터그래픽스, 알고리즘



최 영 근

e-mail : ygchoi@cs.kwangwoon.ac.kr
 1980년 서울대학교 사범대학 수학 교육과
 (이학사)
 1982년 서울대학교 계산 통계학과(이학 석사)
 1989년 서울대학교 계산 통계학과(이학 박사)
 1998년~1999년 광운대학교 정보전산원 원장
 1983년~2000년 광운대학교 이과대학 전자계산학과 교수
 관심분야 : 병렬 컴파일러, 병렬 프로그래밍 언어, 분산컴퓨팅, 이
 동에이전트등