

하우스도르프 거리를 이용한 B-spline 곡선의 낫 제거

오종석 윤승현

동국대학교 대학원 멀티미디어공학과

ziosojs@naver.com, shyun@dongguk.edu

Knot Removal of B-spline Curves using Hausdorff Distance

Jong-Seok Oh Seung-Hyun Yoon

Department of Multimedia Engineering, Dongguk University

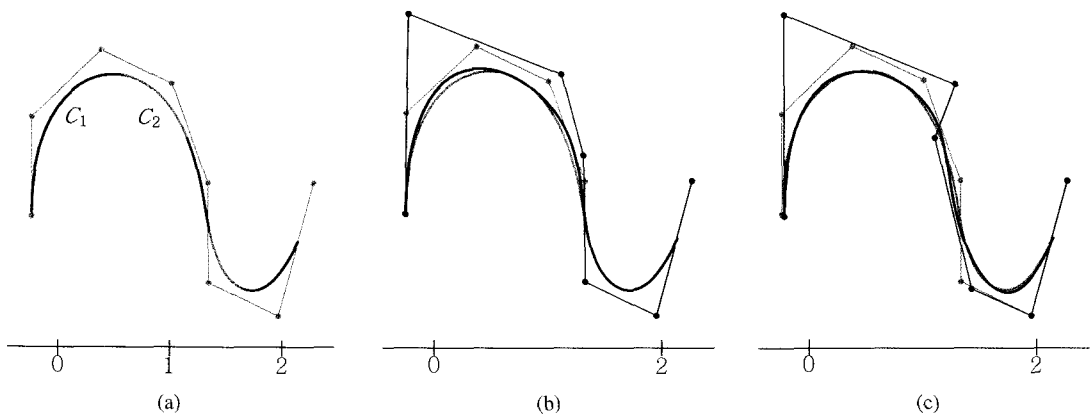


그림 1: B-spline 곡선의 낫 제거: (a) 원본곡선, (b) 연속 L_∞ 근사를 이용한 낫 제거, (c) HD를 이용한 낫 제거.

요약

본 논문에서는 B-spline 곡선의 낫 제거 (knot removal)를 위한 새로운 기법을 제안한다. 제안된 기법은 낫 제거 전후, 두 곡선의 형상의 차이를 측정하기 위해 하우스도르프 거리 (Hausdorff distance)를 이용한다. 먼저 Eck와 Hadenfeld의 연속 L_∞ 근사법 [1]을 이용하여 낫이 제거된 곡선을 생성한다. 수치적 최적화 (numerical optimization) 기법을 통해 생성된 곡선의 제어점 위치를 조정하여, 낫 제거 전 곡선과의 하우스도르프 거리가 최소화 되도록한다. 본 논문에서는 다양한 형태와 차수의 곡선들 (space curves)에 대한 낫 제거 실험을 통해 제안된 기법의 효율성과 우수성을 입증한다.

Abstract

We present a new technique for removing interior knots of parametric B-spline curves. An initial curve is constructed by continuous L_∞ approximation proposed by Eck and Hadenfeld. We employ Hausdorff distance to measure the shape difference between the original curve and the initial one. The final curve is obtained by minimizing their Hausdorff distance. We demonstrate the effectiveness of our technique with experimental results on various types of planar and spatial curves.

키워드: B-spline 곡선, 하우스도르프 거리, 연속 L_∞ 근사, Powell 방법

Keywords: Parametric B-spline curves, Hausdorff distance, continuous L_∞ approximation, Powell's method

1. 서론

B-spline을 이용한 곡선과 곡면의 모델링 기법은 컴퓨터이용 설계 (computer-aided design), 기하모델링 (geometric modeling) 그리고 컴퓨터그래픽스 등의 분야에서 3차원 자유형상 (freeform)을 표현하기 위해 널리 사용된다 [2, 3, 4]. B-spline 기저함수들 (basis functions)은 제어점들 (control points)과 선형 결합 (linear combination)을 통해 복잡한 형상의 곡선과 곡면을 쉽게 표현할 수 있다. 사용자는 제어점의 위치를 조정함으로써 모델링된 형상을 직관적이고 효율적으로 편집할 수 있다. 더욱이, B-spline 기저함수에 대한 낫 추가 (knot insertion) 기법은 모델링된 곡선과 곡면의 형상을 바꾸지 않고 새로운 제어점을 추가할 수 있도록 하는 매우 유용한 기하학적 알고리즘이다 [2, 4].

낫 제거 (knot removal) 기법은 낫 추가와는 반대되는 개념으로, 불필요한 제어점을 제거하는 기법이다 [1, 5, 6]. 원칙적으로 모든 낫이 제거가능한 것은 아니고, 곡선과 곡면의 형상을 변형시키지 않는 낫만이 제거될 수 있다. Eck와 Hadenfeld [1]는 B-spline 곡선에서 임의의 낫을 제거하는 기법을 제안하였다. 이 기법은 낫 제거 전후, 두 곡선의 형상의 차이를 연속(continuous) L_∞ 근사를 이용하여 측정하고, 이를 최소화함으로써 선택된 낫을 제거한다.

그림 1(a)는 낫 벡터 $U = \{\dots, 0, 1, 2, \dots\}$ 에서 정의된 4차 B-spline 곡선을 나타낸다. 인접한 낫 구간 $[0, 1]$ 과 $[1, 2]$ 에서 정의된 부분곡선 $C_1(u)$ 와 $C_2(u)$ 는 연결점 $C_1(1)$ 과 $C_2(1)$ 에서 C^3 연속성을 갖고 부드럽게 연결되어 있다. 낫 벡터 U 에서 낫 $u = 1$ 을 제거할 경우, 부분곡선 $C_1(u)$ 와 $C_2(u)$ 는 새로운 낫 구간 $[0, 2]$ 에서 하나의 곡선으로 근사 (approximation) 되어야 한다. 그림 1(b)의 검은색 곡선은 Eck와 Hadenfeld [1]가 제안한 기법을 통하여 낫 $u = 1$ 이 제거된 B-spline 곡선을 나타낸다.

주어진 두 개의 객체 사이의 하우스도르프 거리 (Hausdorff Distance, 이하 HD)는 하나의 객체에서 다른 객체까지의 최단 거리 (minimum distance)의 최대 편차 (maximum deviation)을 나타낸다. HD는 일반적으로 컴퓨터 비전 및 영상처리 등의 분야에서 두 이미지의 유사성 (similarity)을 측정하기 위해 사용되며 [7], 최근에는 기하모델링, 계산기하학 (computational geometry) 그리고 컴퓨터그래픽스 등의 다양한 응용 분야에서 기하학적 객체간의 유사도를 측정하는 새로운 척도 (metric)의 개념으로 활용되고 있다 [8, 9, 10].

본 논문에서는 B-spline 곡선에 대한 새로운 낫 제거 기법을 제안한다. 제안된 기법은 Eck와 Hadenfeld [1]의 기법을 확장하여, 낫 제거 전후의 두 곡선의 차이를 HD를 이용하여 측정하고, 이를 최소화함으로써 낫이 제거된 곡선을 생성한다. HD를 이용하여 측정된 두 곡선간의 거리는 연속 L_∞ 근사를 이용한 방법에 비하여, 두 곡선간의 기하학적 형상의 차이를 보다 정확하게 측정한다. 결과적으로 낫 제거 후에도 제거 전의 곡선과 매우 유사한 형상을 갖는 곡선을 생성하게 된다. 그림 1(c)

의 검은색 곡선은 본 논문에서 제안된 기법을 통하여 낫을 제거한 결과를 나타낸다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2절에서는 낫 제거 및 HD 계산에 관련된 이전 연구들을 소개하고, 3절에서는 B-spline 곡선의 낫 제거에 대한 일반적인 접근법을 소개한다. 본 논문에서 제안하는 새로운 낫 제거 방법은 4절에서 상세히 설명하고, 이러한 기법을 이용한 다양한 실험결과는 5절에서 제시된다. 끝으로 6절에서는 본 논문의 결론을 맺으며 향후연구방향을 제시한다.

2. 관련연구

본 절에서는 B-spline 곡선과 곡면의 낫 제거 기법과 HD 계산에 관련된 이전 연구결과들을 소개한다.

Lyche와 Morken [11]은 이산길이 (discrete norm)를 이용하여 B-spline 곡선과 곡면의 낫 제거 기법을 제안하였다. 이 방법은 낫 제거 전후, 두 곡선들 또는 곡면들 사이의 최대 이산길이로써 형상의 차이를 측정하고, 이를 최소화한다. 이산길이를 이용한 낫 제거 방법은 곡선의 근사, 차수 감소 등에 쓰일 수 있지만 낫 제거 전후, 두 곡선의 형상의 차이를 제어점을 통해 비교하기 때문에 근사가 정확하지 않은 단점이 있다. Eck와 Hadenfeld [1]는 이 방법 [11]을 확장하여 낫 제거 전후, 두 곡선의 형상의 차이를 측정하기 위해서 레메즈 (Remez) 알고리즘 [12]을 이용한 연속 L_∞ 근사법을 제시하였다. 그러나, 이 기법은 두 곡선간의 최단거리를 동일한 매개변수값에 대응하는 곡선위의 점들간의 거리로 근사하는 문제점을 갖는다. 본 논문에서는 Eck와 Hadenfeld [1]의 기법을 더욱 확장하여, HD를 이용한 새로운 낫 제거기법을 제안한다.

다양한 형태의 낫 제거 기법들 [6, 13, 14, 15]이 곡선과 곡면의 모델링 후, 형상에 대한 간결한 표현 (compact representation)을 얻기 위해 제안되었다. Mehaute와 Lafranche [6]는 흩어진 점군 (scattered point cloud)을 B-spline 곡면으로 복원한 후, 주어진 오차의 범위내에서 내부 (interior) 낫의 위치와 갯수를 자동으로 계산하여 불필요한 낫을 제거하는 기법을 제안하였다. Jo와 Han [13]은 단면곡선 (cross-sectional curve)을 이용한 인체형상의 모델링 기법을 제안하였다. 이 기법은 인체형상의 단면형상을 보간하여 B-spline 곡선으로 근사한 후, 주어진 오차범위에서 불필요한 제어점을 제거하여 인체형상을 효율적으로 모델링한다. Wang 등 [14]은 객체 데이터의 점들을 보간하여 B-spline 곡선을 만든 후, 곡면으로 만드는 과정에서 기존 곡선의 형상을 보장하는 객체 데이터의 점들을 제외한 불필요한 점들을 제거하는 에너지 최소화 (energy minimization) 기법을 제시하였다. Ibrahim 등 [15]는 입자 무리 최적화 (particle swarm optimization)를 통해 자동적으로 오차의 한계를 결정하고 낫을 제거하는 향상된 기법을 제시하였다.

최근 HD는 다양한 기하학적 객체간의 유사성을 측정하는 도구로써 활용되고 있다. Tang 등 [9]은 BVH (Bounding Vol-

ume Hierarchy)와 cross-culling 기법을 이용하여 일반적인 다각형 메쉬 (polygonal mesh)간의 HD를 근사적으로 계산하는 효율적인 알고리즘을 제안하였으며, Bartoň 등 [16]은 원뿔과 등거리 이등분선의 교차점을 구하여 정확한 HD를 계산하는 기법을 제시하였다. Klein 등 [17]은 3차원 메쉬 간소화 (mesh simplification) 기법에서 메쉬간의 형상의 차이를 측정하기 위해서 HD를 사용하였다.

다각형 모델에 비해 자유형상 (freeform) 곡선과 곡면의 HD 계산은 어려운 문제로 인식되어왔다. 최근 Elber와 Grandine [18]은 자유형상 곡선간 또는 곡면간의 HD가 생기는 모든 경우를 조사하고, 이를 계산하는 기하학적 알고리즘을 제시하였다. Jüttler [19]는 평면 상의 음함수 (implicit) 곡선에 대한 HD 계산 방법을 제안하였고, Kim 등 [10]은 평면 곡선의 HD를 실시간으로 계산할 수 있는 방법을 제안하였다. 이 기법은 주어진 곡선을 G^1 연속성을 갖는 이중원호 (bi-arc)로 근사하고 그래픽스하드웨어의 깊이버퍼 (depth buffer)를 이용한 거리 맵 (distance map)을 생성하여, 정확한 HD를 실시간에 계산한다. 본 논문에서는 낫 제거 전후의 곡선의 HD를 계산하기 위해서 Kim 등 [10]이 제시한 기법을 활용한다.

3. B-spline 곡선의 낫 제거

본 절에서는 B-spline 곡선의 낫 제거기법을 위한 접근방법을 설명한다.

차수(degree)가 p 인 B-spline 곡선 $C(u)$ 는 다음과 같이 정의된다:

$$C(u) = \sum_{i=0}^n N_i^p(u) \mathbf{p}_i \quad u_p \leq u \leq u_{m-p},$$

여기서 $\{\mathbf{p}_i\}$ 는 곡선의 제어점을 나타내고, $N_i^p(u)$ 는 낫 벡터 $U = \{u_0, u_1, \dots, u_p, \dots, u_{m-p}, \dots, u_m\}$ ($u_i \leq u_{i+1}, i = 0, 1, \dots, m-1$)에서 정의되는 p 차 i 번째 B-spline의 기저함수이다 [2].

낫 벡터 U 에 새로운 낫, $\hat{u} \in [u_k, u_{k+1}]$ 을 추가하면, 곡선 $C(u)$ 는 제어점의 갯수가 하나 늘어난 곡선 $\hat{C}(u)$ 가 되며 새로운 제어점 $\hat{\mathbf{p}}_i$ 은 다음과 같이 선형보간 (linear interpolation)에 의해 계산된다:

$$\hat{\mathbf{p}}_i = (1 - \alpha_i) \mathbf{p}_{i-1} + \alpha_i \mathbf{p}_i, \quad (1)$$

여기서 α_i 는 다음과 같이 정의된다:

$$\alpha_i = \begin{cases} 1 & 0 \leq i \leq k-p, \\ \frac{\hat{u} - u_{k-p+1}}{u_{k+1} - u_{k-p+1}} & k-p < i < k+1, \\ 0 & k+1 \leq i \leq n+1. \end{cases}$$

곡선 $\hat{C}(u)$ 의 변경된 제어점 $\hat{\mathbf{p}}_i$, ($k-p \leq i \leq k+1$)에 대하여

식 (1)는 다음과 같이 행렬의 곱으로 표현될 수 있다:

$$\begin{bmatrix} \hat{\mathbf{p}}_{k-p} \\ \hat{\mathbf{p}}_{k-p+1} \\ \vdots \\ \hat{\mathbf{p}}_{k-1} \\ \hat{\mathbf{p}}_k \\ \hat{\mathbf{p}}_{k+1} \end{bmatrix} = A \begin{bmatrix} \mathbf{p}_{k-p} \\ \mathbf{p}_{k-p+1} \\ \vdots \\ \mathbf{p}_{k-1} \\ \mathbf{p}_k \end{bmatrix},$$

여기서 A 는 크기가 $(p+2) \times (p+1)$ 인 행렬로 다음과 같이 식 (1)의 선형보간비율을 원소로 갖는다:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ (1 - \alpha_{k-p+1}) & \alpha_{k-p+1} & 0 & 0 & \cdots & 0 \\ 0 & (1 - \alpha_{k-p+2}) & \alpha_{k-p+2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & (1 - \alpha_k) & \alpha_k \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

낫 제거기법은 낫 추가의 역과정이므로 행렬 A 의 역행렬을 이용하여 제어점 $\hat{\mathbf{p}}_i$ 으로부터 \mathbf{p}_i 를 계산할 수 있다. 그러나 행렬 A 는 정방행렬이 아니므로 역행렬이 존재하지 않는다. 일반적으로 최소자승법 (least-squares)을 이용하여 \mathbf{p}_i 를 구할 수 있지만, 이러한 방법은 낫 제거 전후, 두 곡선의 형상의 차이를 최소화 시키지 못한다.

행렬 A 의 첫번째 행과 마지막 행을 제거한 정방행렬을 각각 A_R 과 A_L 이라고 하면, 제어점 \mathbf{p}_i 는 각각 다음과 같이 계산될 수 있다:

$$\begin{bmatrix} \mathbf{p}_{k-p}^R \\ \mathbf{p}_{k-p+1}^R \\ \vdots \\ \mathbf{p}_{k-1}^R \\ \mathbf{p}_k^R \end{bmatrix} = (A_R)^{-1} \begin{bmatrix} \hat{\mathbf{p}}_{k-p+1} \\ \vdots \\ \hat{\mathbf{p}}_{k-1} \\ \hat{\mathbf{p}}_k \\ \hat{\mathbf{p}}_{k+1} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \mathbf{p}_{k-p}^L \\ \mathbf{p}_{k-p+1}^L \\ \vdots \\ \mathbf{p}_{k-1}^L \\ \mathbf{p}_k^L \end{bmatrix} = (A_L)^{-1} \begin{bmatrix} \hat{\mathbf{p}}_{k-p} \\ \vdots \\ \hat{\mathbf{p}}_{k-1} \\ \hat{\mathbf{p}}_k \end{bmatrix}$$

그림 2(a)와 (b)는 제어점 6개로 구성된 4차 B-spline 곡선의 낫 벡터 $U = \{0, 0, 0, 0, 0, 0.5, 1, 1, 1, 1, 1\}$ 에서 낫 $u = 0.5$ 을 제거한 뒤에 각각 A_R 과 A_L 을 이용하여 제어점을 구한 결과이다. 결과적으로 A_R 은 오른쪽 제어점으로부터 시작하여 외분 (extrapolation)을 이용하여 \mathbf{p}_i^R 를 구하고, A_L 은 왼쪽 제어점으로부터 외분을 이용하여 \mathbf{p}_i^L 를 구하게 된다. Eck와 Hadenfeld [1]는 연속 L_∞ 근사를 이용하여 낫 제거 전후, 두 곡선의 형상의 차이를 측정하고, 이를 최소화 하기 위하여 그림 2(c)와 같이 \mathbf{p}_i^L 과 \mathbf{p}_i^R 의 선형보간비율 β_i , ($k-p \leq i \leq k$)를 찾는 방법을 제시하였다. 본 논문에서는 Eck와 Hadenfeld [1]의 방법을 확장하여 HD를 이용하여 \mathbf{p}_i^L 과 \mathbf{p}_i^R 의 선형보간비율 β_i , ($k-p \leq i \leq k$)를 찾는 방법을 제시한다.

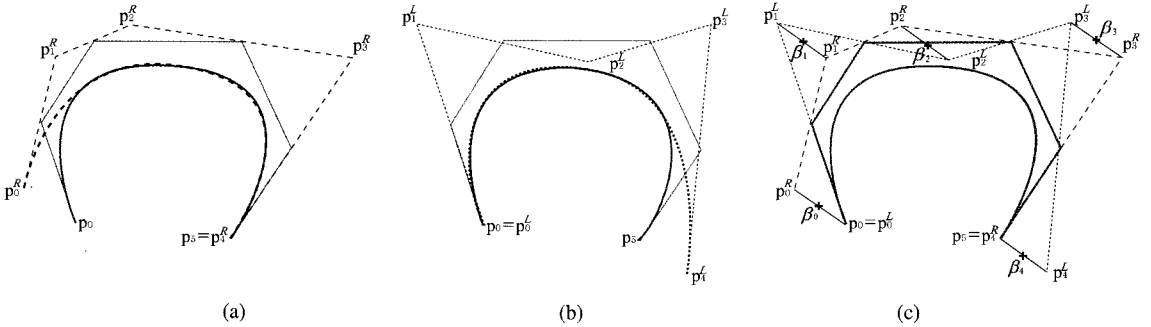


그림 2: 4차 B-spline 곡선의 낫 제어: (a) A_R 을 이용한 제어점 p_i^R 의 생성, (b) A_L 을 이용한 제어점 p_i^L 의 생성, (c) p_i^R 과 p_i^L 의 선형보간을 통한 제어점의 생성.

4. 하우스도르프거리를 이용한 낫 제어

본 절에서는 두 곡선간의 형상의 차이를 측정하기 위해서 HD를 계산하는 방법을 설명하고, 이를 이용하여 낫 제어 전후, 두 곡선간의 형상의 차이를 최소화하는 새로운 낫 제어 기법을 제시한다.

$\hat{C}(u)$ 를 낫 벡터 $\hat{U} = \{\dots, u_k, \hat{u}, u_{k+1}, \dots\}$ 에서 정의된 B-spline 곡선이라고 하고, $C(u)$ 를 낫 \hat{u} 를 제거하여 얻어진 곡선이라고 하자. 두 곡선 $\hat{C}(u)$ 와 $C(u)$ 사이의 하우스도르프거리 $H(\hat{C}, C)$ 는 다음과 같이 정의된다:

$$H(\hat{C}, C) = \max \left(h(\hat{C}, C), h(C, \hat{C}) \right),$$

$$h(\hat{C}, C) = \max_{p \in \hat{C}} \left(\min_{q \in C} \|p - q\| \right).$$

$H(\hat{C}, C)$ 는 두 곡선간의 최단 거리중에서 가장 긴 거리를 나타내며, 결국 낫 제어 전후, 두 곡선의 형상의 차이를 표현하기 위한 기준값으로 활용될 수 있다.

최근 Elber와 Grandine [18]은 3차원 공간의 두 곡선 C_1 과 C_2 에 대하여 HD가 측정될 수 있는 모든 경우를 검사하여 두 가지 경우로 구분하였다. 첫번째 경우는 HD가 두 곡선의 이중법선 (bi-normal)에서 생기는 안티포달 (antipodal)의 경우이며, C_1 과 C_2 사이의 모든 안티포달 점들은 다음 방정식의 해를 구함으로써 찾을 수 있다 [10, 18]:

$$\begin{aligned} \langle C_1(r) - C_2(t), C_1'(r) \rangle &= 0, \\ \langle C_1(r) - C_2(t), C_2'(t) \rangle &= 0, \end{aligned} \quad (2)$$

여기서 \langle, \rangle 는 내적 (inner product)을 나타낸다. 두 번째 경우는 한 곡선의 이등분선과 다른 곡선의 교차점에서 HD가 측정되는 바이섹터 (bisector) 경우이다. C_1 과 C_2 사이의 모든 바이섹터 경우는 다음 방정식의 해를 구함으로써 찾을 수 있다 [10,

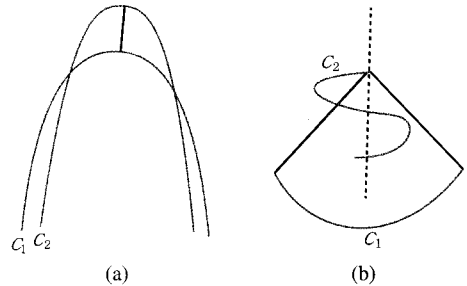


그림 3: 두 곡선 사이의 HD (검은색 실선): (a) antipodal 경우, (b) bisector 경우 (점선은 C_1 의 이등분선).

18]:

$$\begin{aligned} \langle C_1(r) - C_2(s), C_2'(s) \rangle &= 0, \\ \langle C_1(r) - C_2(t), C_2'(t) \rangle &= 0, \\ \langle C_2(t) - C_2(s), C_2(t) + C_2(s) - 2C_1(r) \rangle &= 0. \end{aligned} \quad (3)$$

두 곡선 C_1 과 C_2 사이의 HD는 식 (2)와 (3)를 만족하는 모든 점들사이의 최대거리가 된다. 그림 3은 HD가 생기는 두 가지 경우를 보여준다.

본 논문에서 제안하는 낫 제어기법은 Eck와 Hadenfeld [1]의 방법과 유사하게 p_i^L 와 p_i^R 의 선형보간비율을 찾는다. 그러나 Eck와 Hadenfeld [1]의 방법은 낫 제어 전후, 두 곡선간의 형상의 차이를 동일한 매개변수값에 대응하는 곡선위의 점들간의 거리로 측정하기 때문에, 두 곡선의 국소적인 차이만을 고려하는 문제점을 갖는다. 본 논문에서는 HD를 이용하여 이러한 문제점을 해결하고, 수치적 최적화 (numerical optimization) 기법을 통해 이를 최소화하는 선형보간비율 β_i , ($k - p \leq i \leq k$)를 정한다.

표 1은 본 논문에서 제시된 낫 제어 기법의 절차를 나타낸다. 각 행에 대한 세부적인 설명은 다음과 같다.

- 1행: A^R 의 역행렬을 이용하여 p_i^R 를 구한다.

표 1: HD를 이용한 낫 제거.

| | |
|---------|--|
| Input: | $\hat{C}(u)$: 입력곡선, $\hat{u} \in \hat{U} = \{\dots, u_k, \hat{u}, u_{k+1}, \dots\}$: 제거할 낫. |
| Output: | $C(u)$: 낫 \hat{u} 이 제거된 근사곡선, $U = \{\dots, u_k, u_{k+1}, \dots\}$: 새로운 낫 벡터. |
| 1: | $\mathbf{p}_i^R := (A^R)^{-1} [\mathbf{p}_i], (k-p \leq i \leq k)$; |
| 2: | $\mathbf{p}_i^L := (A^L)^{-1} [\mathbf{p}_i], (k-p \leq i \leq k)$; |
| 3: | $\tilde{\beta}^0 := (\beta_{k-p}, \beta_{k-p+1}, \dots, \beta_k)$; |
| 4: | $\tilde{\beta}^* := \arg \min_{\tilde{\beta}} f(\tilde{\beta})$; |
| 5: | $\mathbf{p}_i := (1 - \beta_i^*) \mathbf{p}_i^R + \beta_i^* \mathbf{p}_i^L, (k-p \leq i \leq k)$; |
| 6: | $C(u) := \sum_{i=0}^n N_i^p(u) \mathbf{p}_i$; |

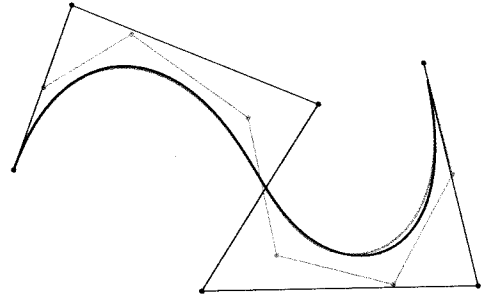
- 2행: A^L 의 역행렬을 이용하여 \mathbf{p}_i^L 를 구한다.
- 3행: 초기 선형보간비율 $\tilde{\beta}^0$ 를 정한다. 본 논문에서는 Eck와 Hadenfeld [1]의 방법을 이용하였다.
- 4행: 수치적 최적화 (numerical optimization) 기법을 통하여 목적함수 (objective function) $f()$ 를 최소화 하는 선형보간비율 $\tilde{\beta}^*$ 을 구한다.
- 5행: 구해진 선형보간비율 $\tilde{\beta}^*$ 를 이용하여, 낫 제거 후 곡선의 제어점을 계산한다.
- 6행: 새로운 제어점으로 낫이 제거된 곡선 $C(u)$ 를 생성하여 반환한다.

본 논문에서는 최소화-목적함수 $f(\tilde{\beta}) : \mathbb{R}^{p+1} \rightarrow \mathbb{R}$ 를 다음과 같이 낫 제거 전후, 두 곡선의 HD로 정의하고 비선형최적화 (nonlinear optimization) 기법 [20]을 이용하여 이를 최소화한다:

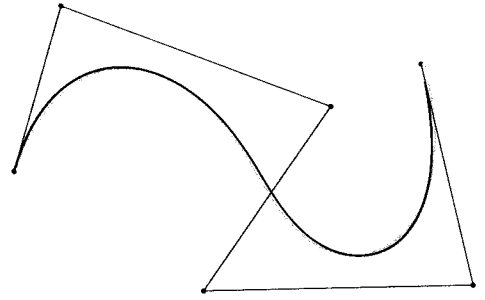
$$f(\tilde{\beta}^j) = H(\hat{C}(u), C^j(u)),$$

여기서 $C^j(u)$ 는 최소화를 위한 j 번째 반복과정에서 구해진 $\tilde{\beta}^j$ 로 \mathbf{p}_i^R 과 \mathbf{p}_i^L 을 선형보간하여 생성되는 곡선이다.

그림 4는 낫 벡터 $U = \{0, 0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4, 4\}$ 에서 정의된 4차 B-spline 곡선에서 낫 $u = 1$ 과 $u = 3$ 을 제거하는 과정을 나타낸다. 그림 4(a)는 연속 L_∞ 근사법 [1]을 이용하여 초기 선형보간비율 $\tilde{\beta}^0$ 을 구하여 낫을 제거한 결과이다 (알고리즘의 3행). 이러한 기법은 국소적인 차이만을 고려하기 때문에 곡선의 가장자리 부분에서 형상의 차이를 유발하고, 두 곡선간의 형상의 차이를 동일한 매개변수값에 대응하는 곡선위의 점들간의 거리로 측정하였기 때문에 곡선간의 차이를 효과적으로 줄이지 못한다. 그림 4(b)는 초기 선형보간비율 $\tilde{\beta}^0$ 로부터 수치적 최소화 과정을 통해 두 곡선간의 HD를 최소화 하는 선형보간비율 $\tilde{\beta}^*$ 을 찾고, 이를 이용하여 낫이 제거된 곡선을 생성한 결과이다. 그림 4(b)에서 볼 수 있듯이, HD를 최소화 하여 낫을 제거한 곡선은 낫 제거전 곡선의 형상을 보다 정확하게 근사하는 것을 볼 수 있다.



(a)



(b)

그림 4: B-spline 곡선의 낫 제거 (회색: 낫 제거 전, 검은색: 낫 제거 후): (a) 초기 선형보간비율 $\tilde{\beta}^0$ 을 이용한 결과, (b) HD를 최소화 하는 선형보간비율 $\tilde{\beta}^*$ 을 이용한 결과.

5. 실험결과

본 논문에서 제안된 HD를 이용한 낫 제거 기법은 Intel(R) Core(TM) i7 2.80 GHz CPU와 4.0GB의 메모리 그리고 NVIDIA GeForce GTS 250 VGA 카드를 탑재한 시스템에서 C++ 언어로 구현되었다. 두 곡선 사이의 HD를 계산하기 위해서 IRIT [21]을 사용하였으며, 제안된 기법은 다양한 형태의 평면 곡선 (planar curve)과 공간 곡선 (space curve)의 낫 제거 실험에 사용되었다. 낫 제거 전후, 두 곡선의 형상의 차이는 HD로 측정하였다. 본 실험에서 연속 L_∞ 근사법을 이용한 낫 제거의 처리 시간은 수초에서 수십초가 걸렸으며, HD를 이용한 낫 제거의 처리 시간은 수분에서 수십분이 걸렸다. 연속 L_∞ 근사에 비해 HD를 이용한 낫 제거 기법의 처리 시간이 늘어나게 된 이유는 낫 제거 전후, 두 곡선의 형상의 차이를 측정하기 위한 하우스도르프거리의 계산이 수치적 최적화 기법에 의해 수십에서 수백번 이상 반복되기 때문이다.

그림 5는 4차 평면 곡선의 낫 제거의 결과이다. 그림 5(a)와 (b)에서는 낫 $u = 0.5$ 를 제거하였고, 그림 5(c)와 (d)에서는 두 개의 낫 $u = 1$ 와 $u = 4$ 를 순차적으로 제거한 결과를 보여준다.

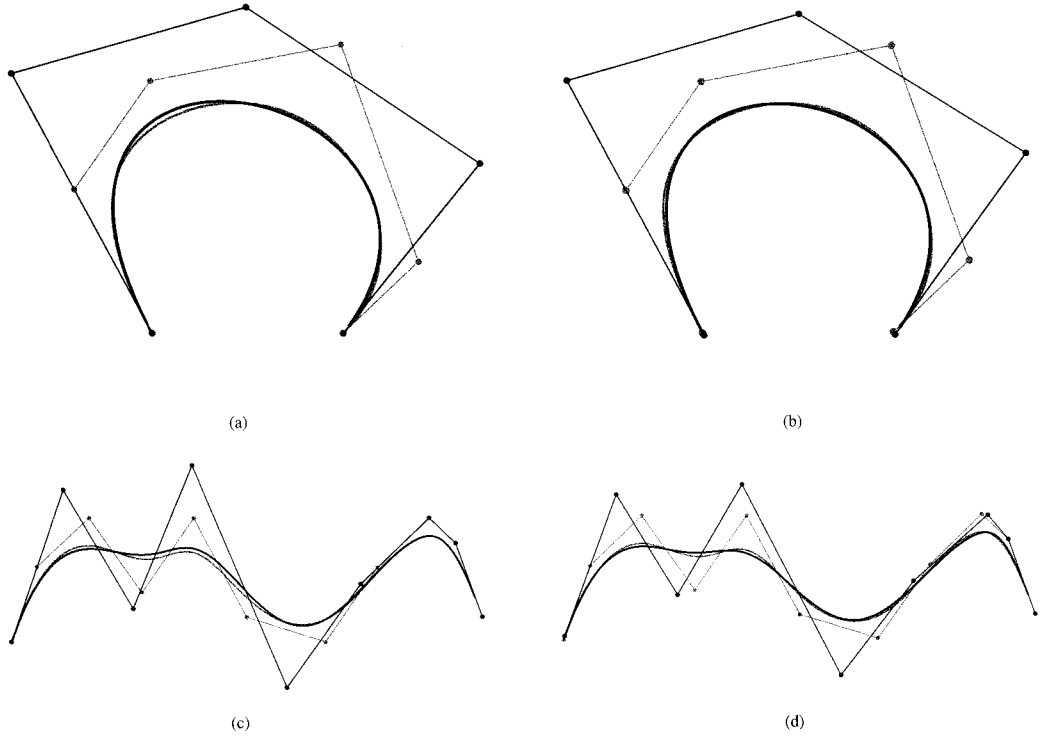


그림 5: 평면 곡선의 낫 제거 (회색: 낫 제거 전, 검은색: 낫 제거 후): (a)와 (c) 연속 L_∞ 근사를 이용한 낫 제거, (b)와 (d) HD를 이용한 낫 제거.

그림 5의 낫 제거 전후, 연속 L_∞ 근사의 결과는 그림 5(a)와 (c)에서 보여주고 있다. 본 논문에서 제안한 HD를 이용한 낫 제거의 결과는 그림 5(b)와 (d)에서 각각 보여준다. 그림 5의 평면 곡선에 대하여 연속 L_∞ 와 HD를 이용한 낫 제거 후, 원본 곡선과의 HD는 각각 표 2의 4열과 5열에서 비교하였다. 표 2의 6열에서 HD를 이용한 낫 제거 기법은 연속 L_∞ 근사를 이용한 기법에 비해 낫 제거 전후, 두 평면 곡선의 형상의 차이를 더 최소화 시키는 것을 확인할 수 있다.

그림 6은 4차 공간 곡선의 낫 제거의 결과이다. 그림 6(a)와 (b)에서는 낫 $u = 0.2$ 를 제거하였다. 그림 6(c)와 (d)에서는 연속된 두 개의 낫 $u = 0.2$ 와 $u = 0.4$ 를 순차적으로 제거하였다. 그림 6(e)와 6(f)에서는 떨어진 두 개의 낫 $u = 0.16$ 과 $u = 0.83$ 을 순차적으로 제거하였다. 그림 6의 낫 제거 전후, 연속 L_∞ 근사의 결과는 그림 6(a), (c) 그리고 (e)에서 보여주고 있다. 본 논문에서 제안한 HD를 이용한 낫 제거의 결과는 그림 6(b), (d) 그리고 (f)에서 각각 보여준다. 그림 6의 곡선에 대한 낫 제거 전후의 HD 측정 결과는 표 3에서 비교하였다. 평면 곡선 (표 2)와 마찬가지로 표 3은 일반적으로 연속 L_∞ 근사를 이용한 낫

제거보다 본 논문에서 제안한 HD를 이용한 낫 제거가 현저하게 성능이 향상되는 것을 보여준다. 그러나 그림 6(e)와 (f)와 같이 두 개의 낫을 순차적으로 제거한 경우, 성능이 떨어지는 경우가 있다. 이 현상은 적은 수의 제어점으로 곡선을 생성할 경우, HD의 측정 결과는 하나의 객체에서 다른 객체까지의 최단거리의 최대편차이기 때문에 나타날 수 있다. 그림 6에서 보듯이 낫 제거 전후, 두 곡선의 형상의 차이가 그림 6(e)에서 전체적으로 나타나는 반면, 그림 6(f)에서는 부분적으로 일어나는 것을 볼 수 있다.

그림 7은 차수가 4인 공간 곡선에서 세 개의 낫 $u = 1$ 와 $u = 6$ 과 $u = 5$ 를 순차적으로 제거한 결과이다. 그림 7(a)는 연속된 낫 $u = 6$ 과 $u = 5$ 를 제거할 경우, 연속 L_∞ 근사 기법은 낫 제거가 일어난 곡선의 국소적인 부분에서 낫 제거가 이루어진 후, 다시 그 과정을 반복함에 따라 두 곡선의 형상의 차이가 커지는 것을 보여준다. 반면에 그림 7(b)는 HD를 이용한 기법이 낫 제거 전후, 두 곡선 전체의 HD를 최소화시키기 때문에 연속된 낫을 제거하더라도 곡선의 전체적 모양을 조정하여 형상의 차이를 줄여주는 것을 볼 수 있다.

표 2: 평면 곡선에 대한 낫 제거 전후의 HD 측정 결과

| 실험결과 | 낫 벡터 | 낫 삭제 | 연속 L_∞ 를 이용한 근사: ω | HD를 이용한 근사: δ | 근사성능(배): ω/δ |
|--------------|---|-----------|-------------------------------------|-------------------------|-----------------------------|
| 그림 5(a), (b) | $U = \{0,0,0,0,0,0.5,1,1,1,1\}$ | $u = 0.5$ | 0.83 | 0.15 | 5.45 |
| 그림 5(c), (d) | $U = \{0,0,0,0,0,1,2,3,4,5,6,7,7,7,7\}$ | $u = 1$ | 0.98 | 0.71 | 1.38 |
| | | $u = 4$ | 0.88 | 0.40 | 2.19 |

표 3: 공간 곡선에 대한 낫 제거 전후의 HD 측정 결과

| 실험결과 | 낫 벡터 | 낫 삭제 | 연속 L_∞ 를 이용한 근사: ω | HD를 이용한 근사: δ | 근사성능(배): ω/δ |
|--------------|---|------------|-------------------------------------|-------------------------|-----------------------------|
| 그림 6(a), (b) | $U = \{0,0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1\}$ | $u = 0.2$ | 2.38 | 1.02 | 2.34 |
| 그림 6(c), (d) | $U = \{0,0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1\}$ | $u = 0.2$ | 0.65 | 0.41 | 1.61 |
| | | $u = 0.4$ | 2.85 | 1.40 | 2.03 |
| 그림 6(e), (f) | $U = \{0,0,0,0,0,0.16,0.33,0.5,0.66,0.83,1,1,1,1\}$ | $u = 0.16$ | 1.89 | 0.83 | 2.27 |
| | | $u = 0.83$ | 1.90 | 1.73 | 1.1 |
| 그림 7(a), (b) | $U = \{0,0,0,0,0,1,2,3,4,5,6,7,7,7,7\}$ | $u = 1$ | 1.02 | 0.71 | 1.45 |
| | | $u = 6$ | 1.20 | 0.65 | 1.85 |
| | | $u = 5$ | 2.84 | 1.18 | 2.4 |

6. 결론

본 논문에서는 HD를 이용한 B-spline 곡선의 새로운 낫 제거 기법을 제안하였다. 낫 제거 전후, 두 곡선간의 형상의 차이를 최소화 하기 위해 HD를 사용하였으며, 최적화 기법을 통해 이를 최소화하는 제어점의 선형보간비율을 찾는 방법을 제시하였다. 제안된 기법은 다양한 형태의 평면 및 공간 곡선에서 기존의 연속 L_∞ 근사법 [1]에 비해 낫 제거 후에도 원본 곡선의 형상을 보다 정확하게 근사하는 결과를 보였다. 또한 여러 개의 낫을 연속적으로 제거할 경우, 원본 곡선과의 형상의 차이가 연속 L_∞ 근사법 [1] 보다 현저하게 줄어드는 것을 확인하였다.

본 논문에서 제시한 기법은 수치적 최적화 과정을 거쳐야 하기 때문에 낫이 제거된 곡선을 실시간으로 생성하지 못하는 한계를 갖는다. 이러한 문제점은 최적화 과정에서 반복적으로 수행되는 HD의 계산시간에서 기인한다. 향후 연구계획으로는 두 곡선간의 HD를 실시간으로 계산하는 방법을 개발하고, 이를 활용하여 낫 제거 기법의 계산시간을 단축할 계획이다. 또한, HD를 이용한 곡선의 낫 제거 기법을 확장하여 곡면의 낫 제거를 위한 새로운 기법을 제시할 것이다.

감사의 글

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2011-0003474) 지원을 받아 수행되었으며, 문화체육관광부 및 한국콘텐츠진흥원의 2011년도 문화콘텐츠산업기술지원사업(지정공모)의 연구결과로 수행되었음.

참고 문헌

- [1] M. Eck and J. Hadenfeld, "Knot removal for B-spline curves," *Computer Aided Geometric Design*, vol. 12, no. 3, pp. 259–282, 1995.
- [2] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. Springer-Verlag, 1997.
- [3] D. Salomon, *Curves and Surfaces for Computer Graphics*. Springer, 2006.
- [4] G. Farin, *Curves and Surfaces for CAGD*, 5th ed. Academic Press, 2002.
- [5] T. Lyche and K. Morken, "Knot removal for parametric B-spline curves and surfaces," *Computer Aided Geometric Design*, vol. 4, no. 3, pp. 217–230, November 1987.
- [6] A. L. Méhauté and Y. Lafranche, "Knot removal for scattered data," in *Proceedings of SPIE*, 1991, pp. 161–164.
- [7] W. Rucklidge, *Efficient Visual Recognition Using the Hausdorff Distance*. Springer, 1996.
- [8] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbury, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," in *SIGGRAPH'95 Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 1995, pp. 173–182.

- [9] M. Tang, M. Lee, and Y.-J. Kim, "Interactive Hausdorff distance computation for general polygonal models," *ACM Transactions on Graphics*, vol. 28, no. 3, August 2009.
- [10] Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, and G. Elber, "Precise Hausdorff distance computation for planar freeform curves using biarcs and depth buffer," *The Visual Computer*, vol. 26, no. 6–8, pp. 1007–1016, 2010.
- [11] T. Lyche and K. Morken, "Knot removal for parametric B-spline curves and surfaces," *Computer Aided Geometric Design*, vol. 4, no. 3, pp. 217–230, 1987.
- [12] S. A. Tawfik, "Minimax approximation and Remez algorithm," *Computer Aided Design*, vol. 40, pp. 999–1008, October 2008.
- [13] J.-W. Jo and S.-S. Han, "The NURBS human body modeling using local knot removal," *Fibers and Polymers*, vol. 6, no. 4, pp. 348–354, 2005.
- [14] W.-K. Wang, H. Zhang, H. Park, J.-H. Yong, J.-C. Paul, and J.-G. Sun, "Reducing control points in lofted B-spline surface interpolation using common knot vector determination," *Computer Aided Design*, vol. 40, no. 10-11, pp. 999–1008, October 2008.
- [15] A. R. Ibrahim, S. M. Shamsuddin, and A. Ali, "Improving Non-Uniform Rational B-splines' knot removal with Particle Swarm Optimization," in *Proceedings of the 2009 6th International Conference on Computer Graphics, Imaging and Visualization*, 2009, pp. 24–27.
- [16] M. Bartoň, I. Hanniel, G. Elber, and M.-S. Kim, "Precise Hausdorff distance computation between polygonal meshes," *Computer Aided Geometric Design*, vol. 27, no. 8, pp. 580–591, 2010.
- [17] R. Klein, G. Liebich, and W. Straßer, "Mesh reduction with error control," in *Proceedings of the 7th Conference on Visualization '96*. IEEE Computer Society Press, 1996, pp. 311–318.
- [18] G. Elber and T. Grandine, "Hausdorff and minimal distances between parametric freeforms in \mathbb{R}^2 and \mathbb{R}^3 ," in *Proceedings of the 5th International Conference on Advances in Geometric Modeling and Processing*, 2008, pp. 191–204.
- [19] B. Jüttler, "Bounding the Hausdorff distance between implicitly defined and/or parametric curves," in *Mathematical Methods for Curves and Surfaces*. Vanderbilt University, 2001, pp. 223–232.
- [20] W.-H. Press, S.-A. Teukolsky, W.-T. Vetterling, and B.-P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 1992.
- [21] *IRIT 9.0 User's Manual*. Technion, October (2002). [Online]. Available: <http://www.cs.technion.ac.il/~irit/>

〈저자 소개〉



오종석

- 2007년 관동대학교 경영학과 학사
- 2011년 동국대학교 멀티미디어공학과 석사
- 관심분야 : 컴퓨터그래픽스, 기하모델링, 멀티미디어응용시스템



윤승현

- 2001년 한양대학교 수학과 학사
- 2007년 서울대학교 컴퓨터공학과 박사
- 2007년~현재 동국대학교 멀티미디어공학과 조교수
- 관심분야 : 컴퓨터그래픽스, 기하모델링, 멀티미디어응용시스템

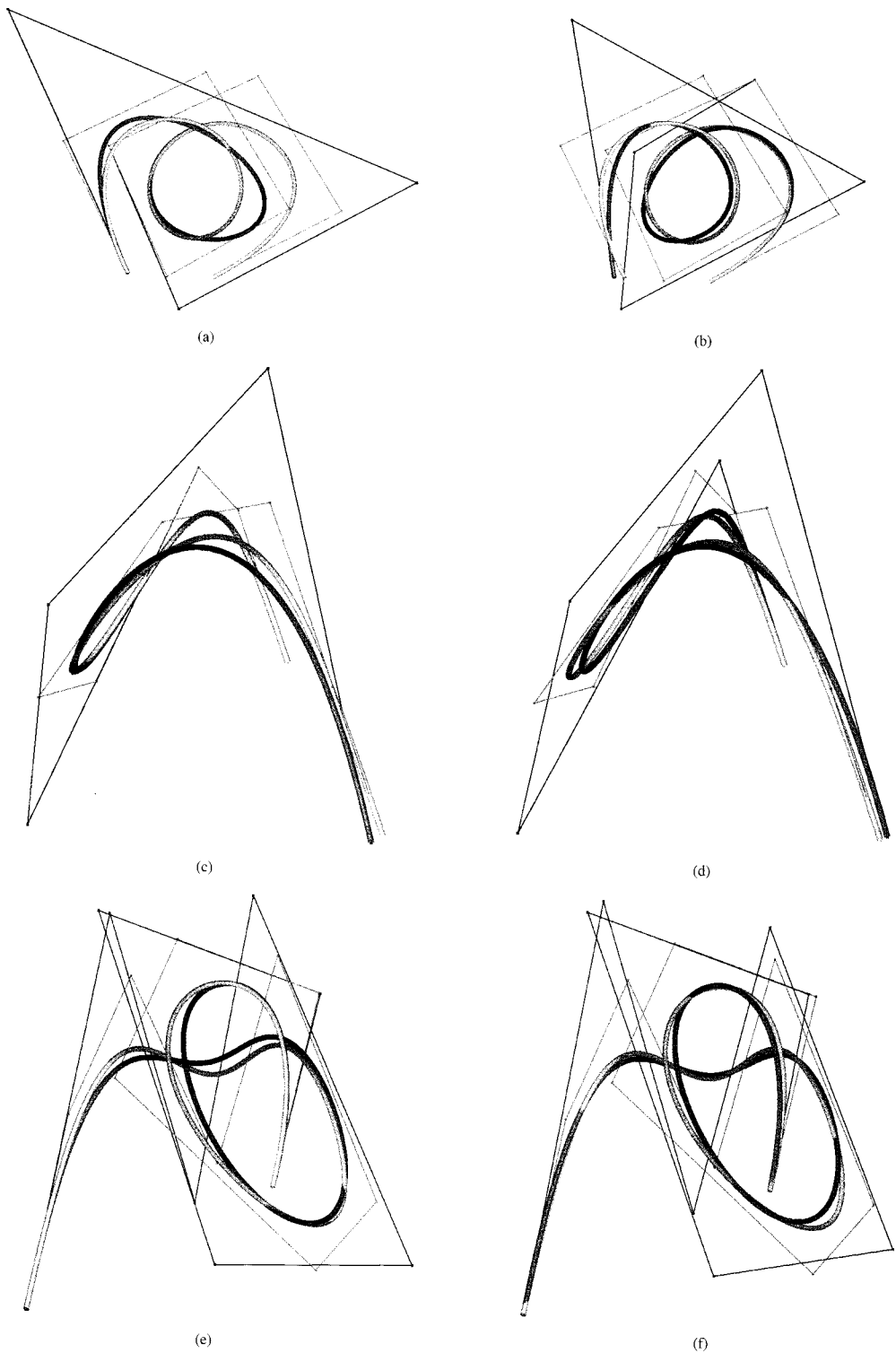
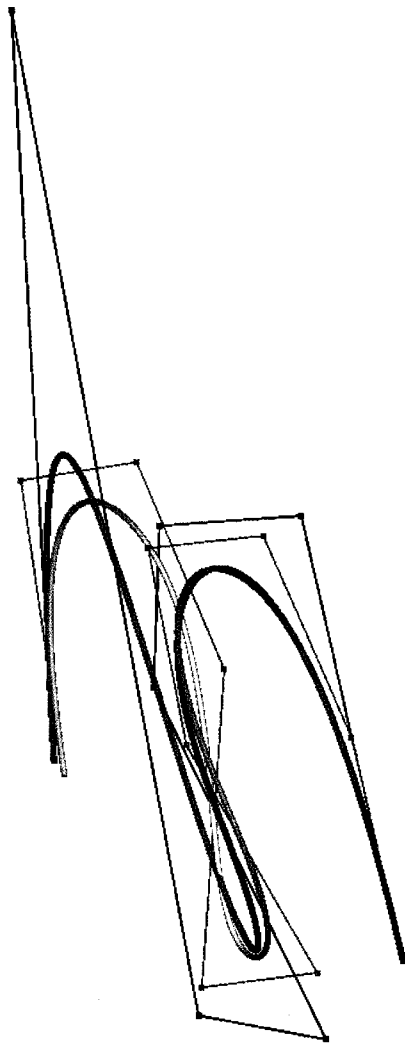
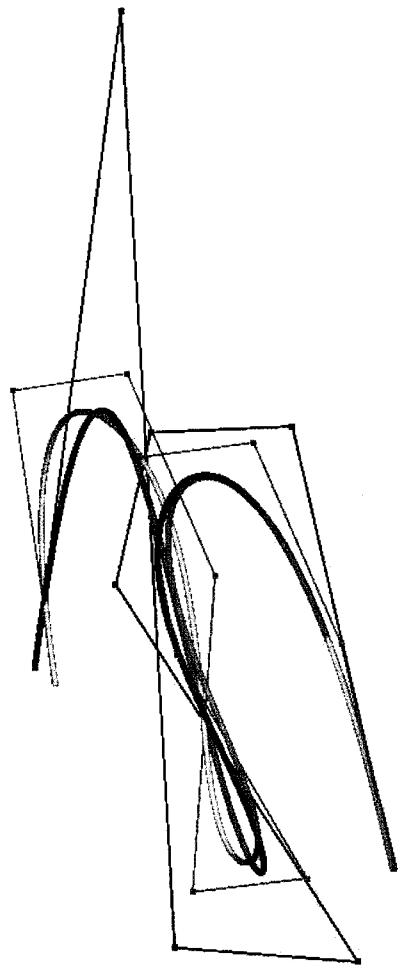


그림 6: 공간 곡선의 뉘 제거 (붉은색: 뉘 제거 전, 검은색: 뉘 제거 후): 연속 L_∞ 근사를 이용한 뉘 제거 (a, c, e), HD를 이용한 뉘 제거 (b, d, f).



(a)



(b)

그림 7: 공간 곡선의 낫 제거 (붉은색: 낫 제거 전, 검은색: 낫 제거 후): (a) 연속 L_∞ 근사를 이용한 낫 제거, (b) HD를 이용한 낫 제거.