韓國 OA學會 論文誌
第6卷 第1號, 2001. 3.

2001 6 1-1-1

# 효율적 시각화 구현을 위한 Knot 제거 알고리즘

김 혁 진*

# Knot Removal for the efficient Visualization Implementations

Hyeock-Jin Kim*


## 요 약

본 논문에서는 B-spline의 근사방법 중의 한 방법으로써 knot 제거 알고리즘을 제안한다. 이 알고리즘은 제거되는 knot의 순서에 영향을 받지 않고 항상 같은 결과를 얻을 수 있을 뿐만 아니라 수학적으로도 단순하므로 이해와 구현이 용이하다. 내부 knot들이 multiplicity를 많이 갖는 경우에 더 큰 장점을 갖는다.

## Abstract

In this paper, the problem of removing the interior knots from a B-spline is discussed. We present a new strategy for reducing the number of knots for splines. The method is the efficient for the visualization implementations and easy-to-use algorithms, and we need not to determine the knot sequence that will be removed.


---

※ 청운대학교 컴퓨터과학과 조교수
본 논문은 2000학년도 청운대학교 학술연구조성비 지원에 의하여 연구되었음
논문 접수 : 2000년 10월 3일 심사 완료 : 2001년 1월 7일

# Ⅰ. 서론

최근 컴퓨터 그래픽스(Computer Graphics)와 이미지처리(image processing) 분야, 그리고 인터넷상에서의 화상처리문제 등이 대두됨에 따라 새로운 시각화(visualization)를 위한 효율적인 접근방법이 더욱 중요시되고 있다.

Knot 삽입과 knot 제거와 같은 기하 모델링(geometric modeling)은 복잡한 자연물을 나타내는 기하학적 표현의 시각화를 위하여, 또는 B-spline을 이해·분석하고 렌더링(rendering)하는 데에 필요한 기본적인 처리과정으로써, 수치적으로도 안정(numerically stable)하기 때문에 컴퓨터 그래픽스 및 CAGD (Computer Aided Geometric Design)등에 적용되고 있으며 컴퓨터 프로그래밍이나 CAD 등에도 도입이 되고 있다.

Knot을 제거하는 과정은 knot을 삽입하는 과정의 역과정으로 진행된다. Knot 삽입 알고리즘은 본래 CAD 위원회에서 발전되었으며, 최초의 knot 삽입 처리과정은 *uniform* 2차 B-spline을 위한 Chaiken의 알고리즘으로, B-spline을 분석하는 가장 유용하고 강력한 처리 과정 중의 하나이다[1]. 일반적으로 knot 삽입은 원래의 곡선 모양과 정확히 일치하는 곡선을 얻을 수 있지만, knot 제거는 특별한 경우를 제외하고는 곡선의 모양을 변경하지 않고서는 정확하게 일치하는 곡선을 얻을 수가 없다. Knot 제거 과정은 근사(Approximation) 과정의 한 방법으로 해석된다. 또한, 더욱 함축된 표현을 할 수 있도록 자료를 압축하므로 더 빠른 알고리즘을 유도해 낸다[2].

Knot 제거 알고리즘은 자료 *fitting*, CAGD, 자료압축 등 많은 분야에서 유용하게 사용되고 있다. 이 분야들을 더 세분해 보면 다음과 같다[3]. 적응적 곡선 근사(adaptive curve approximation), 자료 *fitting*, 차수감소(degree reduction), 구간적 다항식에서 B-spline 형으로의 변환, 곡선 및 곡면의 모델링, *offset* 곡선 및 곡면의 모델링, spline 근사, 자료압축 등의 분야들이다.

본 논문에서는 기하학적 모델링 시스템에서 기본이 되는 곡선 및 곡면의 근사 개념에서 spline의 knot 제거에 대해서 소개하고, 새로운 knot 제거 알고리즘을 제시한다. 이러한 연구는 자연 현상을 효율적으로 처리하고 표현하는 연구에 기본이 되며, 인체와 같은 복잡한 형태의 물체를 자유롭게 표현할 수 있는 능력을 확보할 수 있다.

# Ⅱ. Knot 제거 방법

## 1. 관련연구

Spline에서 가장 유명한 자료압축 방법은 knot 제거 전략이다[4][5]. Le Mehaute와 Lafranche는 Bézier 근사를 이용하는 방법을 제안하였고[6], Lyche와 Morken은 *first rank knot*에 의해 하나 이상의 knot을 근사적으로 제거하고, *global approximation* 방법에 의한 작은 가중치로 모든 knot을 제거하는 방법을 제안하였다[3]. 이와 연관된 문제로써, 3차 B-spline에서 하나의 knot만을 제거하는 연구로 (Kjellader, 1983; Farin et al., 1987; Sapidis와 Farin, 1990) 등이 있다. Tiller는 모든 knot을 제거하는데 컴퓨터의 정확성 내에서 제거되어질 수 있는 알고리즘을 제시하였다[7]. Eck과 Hadenfeld는 Lyche와 Morken의 방법과 약간 다른 방법으로, B-spline에서 하나의 knot만을 제거할 수 있는 가장 간단한 경우로 제한하는 연구를 수행하였다[8]. Eck은 이 방법을 이용하여 연속적인 knot을 제거하는 만족스러운 결과를 얻는 연구도 진행되었다.

## 2. Knot을 제거하는 방법

Knot 벡터 $U$로 정의되는 곡선 $C(u)$에서 중복된 knot의 수가 $s$개인 knot $u_r$을 $t$번 제거할 경우 새로 정의되는 곡선 $\overline{C}(u)$는 다음과 같이 나타낼 수 있다[9][10].

$$\overline{C}(u) = \sum_{i=0}^{n-t} \overline{N_{i,p}}(u)\, Q_i$$

새로 정의된 곡선은 기하학적으로나 매개변수적으로 원래의 곡선 $C(u)$과 일치하므로,

$$\sum_{i=0}^{n} N_{i,p}(u)P_i = \sum_{i=0}^{n-1} \overline{N_{i,p}}(u)\,Q_i$$

이다. 이 식은 미지항의 수 보다 방정식의 수가 많기 때문에 정확한 제어점을 구하기는 힘들다. 결국엔 ( $n$-1) 개의 미지수와 $n$개의 연립 방정식이 된다. 이 때, 차수 감소에서와 같이 오차 보정법(round-off error)을 적용하여 미지수를 구하게 된다.

### 3. Eck과 Hadenfeld의 방법

Knot 제거는 knot 삽입의 역 과정이므로 하나의 knot을 삽입하는 과정을 먼저 보면,

$$C(u) = \sum_{i=0}^{n+1} N_{i,p}(u)\,Q_i \quad (2.1)$$

로 정의되는 B-spline에서 $U$에 내부 knot을 삽입한 $C(u)$가

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)\,P_i \quad (2.2)$$

이라면 새로 만들어지는 제어점은

$$P_i = \begin{cases} Q_i, & 0 \le i \le r-k+v-1, \\ (1-l_{k-r+i})Q_{i-1} + l_{k-r+i}Q_i, & r-k+v \le i \le r-1, \\ Q_{i-1}, & r \le i \le n, \end{cases}$$

이다. 여기서 $k$는 *order*, $v$는 반복 횟수($\le k$), $r$은 knot 벡터에서의 추가된 *knot index*이고,

$$l_j = \frac{t_r - t_{r-k+j}}{t_{r+j} - t_{r-k+j}}, \quad v-1 \le j \le k$$

이다. (2.3)식은 $Q_{i-1}$과 $Q_i$의 line 세그먼트를 $l_{k-r+i}$ : $(1-l_{k-r+i})$, $r-k+v \le i \le r-1$ 비율로 나눈 점으로 Boehm에 의해 유래되었다[1].

$$1 = l_{v-1} > l_v > l_{v+1} > \cdots > l_{k-1} > l_k = 0$$

B-spline을 위한 knot 제거 방법은 다음과 같다. (2.2)식으로 정의되는 B-spline에서 knot 제거로 (2.1)식을 만들 때 $Q_i$는 *extrapolation*식을 이용한다.

$$Q_i = \begin{cases} P_i, & 0 \le i \le r-k+v-2, \\ (1-\mu_{k-r+i})Q_i^L + \mu_{k-r+i}Q_i^R, & r-k+v-1 \le i \le r-1, \\ P_{i+1}, & r \le i \le n-1. \end{cases}$$

$Q_i^L$과 $Q_i^R$간의 line 세그먼트를 $\mu_{k-r+i}$ : (1-$\mu_{k-r+i}$)의 비율로 나눈 점을 새 제어점으로 정하였다. 이때, 실수 $\mu_j, v-1 \le j \le k-1$을 다음과 같이 정의하였다[8].

$$0 \le \mu_{v-1} \le \mu_v \le \cdots \le \mu_{k-2} \le \mu_{k-1} \le 1.$$

### 4. Tiller의 방법

(2.2)식에서 $U$에 $s$번 반복 횟수의 내부 knot $u_r$을 갖는다고 하자. $C(u)$가 다음과 같이 정확한 표현을 갖는다고 한다면 $u_r$을 $t$번 제거 가능하다고 한다.

$$C(u) = \sum_{i=0}^{n-t} \overline{N_{i,p}}(u)\,Q_i$$

이때 (2.1)식과 (2.2)식은 기하학적 매개변수적으로 같은 곡선이다.

Knot 제거 알고리즘은 다음의 2가지를 해야한다.

- knot가 제거 가능한지, 몇 번이나 제거 할 수 있는지를 결정
- 새 제어점 $Q_i$를 계산

Knot 제거 과정의 한 예제로, 3차 B-spline으로부터 하나의 knot을 제거한다고 할 때, 이 알고리즘은 다음의 과정을 수행한다.

- spline을 Bézier 형태로 변환한다.
- 모든 내부 knot들이 차수만큼 multiplicity를 갖는 B-spline 형태를 취한다.
- 한번에 불필요한 여러개의 knot과 제어점을 제거한다.

Tiller가 제안한 knot 제거 방법에서 제거할 때의 오차는 다음과 같이 계산한다.

$$TOL \ \ge \ \frac{d \cdot w_{\min}}{1 + |P|_{\max}}$$

여기서, $w_{\min}$는 처음 곡선의 최소 가중치, $|P|_{\max}$는 원점에서 처음 곡선까지의 최대 거리, $d$는 하나의 제어점을 기준으로 변화한 곡선의 편차이다[7].

## Ⅲ. 새로운 Knot 제거 알고리즘

Knot 벡터 $U$로 정의되는 $p$차 B-spline $C(u)$는

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)P_i$$

으로 정의하고, knot벡터 $U$에서 내부 knot 중 1개를 제거할 경우에 새로 정의되는 곡선 $\overline{C}(u)$는 다음과 같다.

$$\overline{C}(u) = \sum_{i=0}^{n-1} \overline{N_{i,p}}(u) Q_i$$

Knot 삽입은 렌더링과 교차 알고리즘에 구간적 선형 근사를 제공할 수 있어 추가적인 knot들을 가지고 정확히 주어진 곡선이나 곡면을 정교하게 표현할 수 있지만, knot 제거는 몇몇 knot들을 가지고 주어진 곡선이나 곡면을 어떻게 근사화 시킬 수 있는가에 대한 방법으로써 이 두 방법은 많은 응용에 이용되는 중요한 이론적인 도구들이다.

B-spline을 정의할 때 knot의 수는 (차수+제어점수+1)개가 되므로 차수와 제어점의 수에 영향을 받는다. 예를 들어, 12개의 제어점으로 3차 B-spline을 정의한다면 knot의 수는 3+12+1 즉 16개가 된다. 따라서, 차수가 고정되어 있을 때 하나의 knot을 제거한다는 것은 하나의 제어점을 함께 제거한다는 것을 의미한다.

본 논문에서 제안하는 알고리즘은 다음과 같다.

**Algorithm: Knot Removal**

/* Notation

$C(u, n)$ : $n$+1개의 knot로 정의되는 원래의 spline

*TOL* : 허용오차 범위(tolerance)

*p* : 곡선의 차수

*e*[ ]: knot span 각각의 오차

$T(u, n)$ : $n$개로 정의되는 임시 spline

*error*(a,b) : a와 b간의 최대오차를 구하는 함수

*MIN_ERR* : 최소 오차 값

*k* : 최소 오차 값을 갖을 때의 *k*번째 knot

$\overline{C(u, r)}$ : *k*번째 knot이 제거된 *r*개의 knot로 정의되는 spline

*/

1. ***for*** *i*=1 ***to*** (*the number of interior knot*)
   1) $T(u, n) = C(u, n)$
   2) Remove one knot of $T(u, n)$
   3) $e[p+i]$ = *error* $(C(u, n), T(u, n-1))$
2. Find the *MIN_ERR* and it's index *k* of *e*[ ]
3. ***if*** (*MIN_ERR* $\le$ *TOL*) ***then***

   Remove the $k^{th}$ knot of $T(u, n)$

   ***else*** stop

위의 알고리즘에서는 제1, 제2 그리고 제3의 모든 단계를 통하여 하나의 knot와 하나의 제어점만을 제거한다. 만일, 세 단계를 반복한다면 반복하는 수만큼 knot와 제어점의 수를 줄일 수 있다. 제3 단계에서 *if* 조건이 참이면 하나의 knot와 하나의 제어점이 제거된 $\overline{C(u, r)}$을 얻고, 조건이 거짓이면 주어진 허용오차 범위에서는 knot 제거가 불가능하여 처리과정은 종료하게 된다.

Knot 제거는 knot들 중에 B-spline을 구성하는데 불필요한 knot을 제거하는 처리과정으로써 이 과정의 수행에 의하여 자료의 양을 줄일 수 있다. 이때 원래의 spline과 knot이 제거된 spline과의 오차가 발생한다. 이때 발생되는 최대오차 *error*( )를 계산하는 방법과 알고리즘은 다음과 같다[11].

- 임의의 위치 *t*를 정한다.
- *t*에 대한 a곡선과 b곡선의 샘플 점을 구한다.
- 이 두 점간의 거리 *d*를 계산한다.
- *d*의 최대 값을 최대 오차 값으로 정한다.

**Algorithm: Error Estimation**

```
/*
n : degree
A[ ] : Spline control points
B[ ] : Knot-removed control points
sample( ) : compute the sample point
*/
double error(A[ ], B[ ])
max=0.;
for i=1 to (i<=2* n)
    t= i/( n*2.+1.);
    a=sample( n,A,t);
    b=sample( n-1,B,t);
    dist=Distance(a, b);
    if (dist > max) then max=dist;
return(max);
```

이때 등간격 샘플점들을 많이 취하면 취할 수록 더 정확한 최대 오차를 구할 수 있다. 일반적으로 샘플점들의 개

수는 (2*차수)를 취하는 발견적 수(heuristic number)를 쓰게 된다.

기존의 knot 제거 알고리즘들은 새 제어점이나 rank를 구해야 하지만 제시하는 알고리즘은 이러한 요소들을 계산하지 않기 때문에 알고리즘의 속도가 빠르다. 또한, 기존의 알고리즘들에서는 제거하는 knot의 순서에 따라 결과가 달리 나타남으로 knot의 순서를 효율적으로 정하는 알고리즘도 필요하지만 제시하는 알고리즘은 제거하는 knot의 순서에도 상관이 없기 때문에 어떠한 경우에도 항상 같은 결과를 얻을 수 있다. 뿐만 아니라 이 방법은 수학적으로도 단순하므로 이해와 구현이 용이한 특징이 있다. 특히 내부 knot들이 multiplicity를 많이 갖는 경우에 더 효율적인 결과를 얻는다.

이러한 결과는 인체와 같은 복잡한 형태의 물체를 자유롭게 표현할 수 있는 능력을 확보할 수 있고, 수행 결과를 인체 모델러 등에 적용할 수 있으며, 모델러의 핵심 기술로 자리를 잡을 수 있다. 또한 실시간을 요구하는 기술 등과 결합하여 가상공간을 구축하는 기술 등에도 적용을 할 수 있어 인터넷 및 컴퓨터 그래픽스계의 발전에도 큰 기여를 할 수 있을 것으로 사료된다.

## Ⅳ. 결론

Knot 제거 알고리즘은 많은 분야에서 유용하게 사용되고 있고 특히, CAD 시스템이나 컴퓨터 그래픽스 분야에서 널리 사용되고 있다.

본 논문에서는 복잡한 자연물을 표현하는데 기본이 되는 기하 모델링 중에서 B-spline의 knot 제거에 의한 기하학적 시각화를 위한 효율적인 접근방법을 제시하였다.

기존의 방법들은 새 제어점이나 *rank*를 구해야 하지만 제시하는 알고리즘은 이러한 요소들을 계산할 필요가 없다. 제거하는 knot의 순서에도 상관없기 때문에 제거되는 knot의 순서에 영향을 받지 않고 항상 같은 결과를 얻을 수 있다.

뿐만 아니라 수학적으로도 단순하므로 이해와 구현이 용이하다. 특히 내부 knot들이 multiplicity를 많이 갖는 경우에 더 많은 knot을 제거할 수 있다. 그러나 multiplicity가 적은 경우에는 기존의 알고리즘보다 다소 적게 줄일 수도 있다.

앞으로, 기존의 knot 제거 알고리즘들과 본 논문에서 제시하는 알고리즘에 대한 성능 분석과 곡면으로의 확대, 그리고 가상공간 구축과 같은 대량의 자료를 요구하는 분야나 실시간을 요구하는 기술 등과 결합하는 연구가 요구된다.

## 참고문헌


[1] Boehm, W., Inserting new knots into B-spline curves, CAD, Vol.12, No.4, pp.199-201, 1980

[2] (Ed.)Goldman, R. and Lyche, T., Knot insertion and deletion algorithms for B-spline curves and surfaces, SIAM, the Society for Ind. and Appl. Math., 1993

[3] Lyche, T. and Morken, K., Knot removal for parametric B-spline curves and surfaces, Comput. Aid. Geom. Des., Vol.4, pp.217-230, 1987

[4] Farin, G.E., Curves and surfaces for Computer Aided Geometric Design - A Practical Guide, 3rd ed., Boston: Academic Press, 1993.

[5] Saux, E. and Daniel, M., Data reduction of polygonal curves using B-splines, CAD, Vol.31, pp.507-515, 1999

[6] Le Mehaute, A.J.Y. and Lafranche, Y., A knot removal strategy for scattered data in R2, in Mathematical methods in computer aided geometric design, Ed. Lyche, T. and Schumaker, L.L., Boston, Academic Press, pp.419-426, 1989

[7] Tiller, W., Knot-removal algorithms for NURBS curves and surfaces, CAD, Vol.24, No.8, pp.445-453, 1992

[8] Eck, M. and Hadenfeld, J., Knot removal for B-spline curves, Comput. Aid. Geom. Des., Vol.12, pp.259-282, 1995
[9] Kim, Hyeock-Jin et al., The degree reduction of B-splines using Bézier methods, Jour. of the KISS, Vol.26, No.8, pp.875-883, 1999
[10] Piegl, L. and Tiller, W., The NURBS book, 2nd ed., Springer, 1997
[11] Kim, Hyeock-Jin, Degree reduction and computation of the maximum error of Bézier curves, Jour. of the KI of OA, Vol.4, No.4, pp.25-33, 1999

## 저자소개

**김 혁 진**
아주대학교 대학원 컴퓨터공학과 공학박사
1992-1997년 김천대학 사무자동화과 조교수
1997년-현 청운대학교 컴퓨터과학과 조교수
관심분야 : Computer Graphics, CAGD, 웹 기술 등