

## B-spline 곡선을 power 기저형태의 구간별 다항식으로 바꾸는 Direct Expansion 알고리즘

김덕수\*, 류종현\*\*, 이현찬\*\*\*, 신하용\*\*\*\*, 장태범\*\*

### A Direct Expansion Algorithm for Transforming B-spline Curve into a Piecewise Polynomial Curve in a Power Form.

Kim, D.-S.\*, Ryu, J.\*\*, Lee, H.\*\*\*, Shin, H.\*\*\*\* and Jang, T.\*\*

#### ABSTRACT

Usual practice of the transformation of a B-spline curve into a set of piecewise polynomial curves in a power form is done by either a knot refinement followed by basis conversions or applying a Taylor expansion on the B-spline curve for each knot span. Presented in this paper is a new algorithm, called a *direct expansion algorithm*, for the problem. The algorithm first locates the coefficients of all the linear terms that make up the basis functions in a knot span, and then the algorithm directly obtains the power form representation of basis functions by expanding the summation of products of appropriate linear terms. Then, a polynomial segment of a knot span can be easily obtained by the summation of products of the basis functions within the knot span with corresponding control points. Repeating this operation for each knot span, all of the polynomials of the B-spline curve can be transformed into a power form. The algorithm has been applied to both static and dynamic curves. It turns out that the proposed algorithm outperforms the existing algorithms for the conversion for both types of curves. Especially, the proposed algorithm shows significantly fast performance for the dynamic curves.

**Key words** : dynamic curve, B-spline, polynomial, power form, knot refinement, graph, basis conversion, Taylor expansion

#### 1. 서 론

컴퓨터 그래픽이나 기하 모델링에서, B-spline 곡선이나 곡면을 power기저형태로 기저변환해서 사용할 필요성이 종종 생긴다. 곡선의 power 기저형태를 이용하게 되면 변곡점(inflexion point)이나 첨점(cusp) 같은 곡선의 특성점들을 빨리 계산해 낼 수가 있으며, 곡선을 이런 특성점들에서 분할(subdivision)함으로써 곡선간의 교점을 좀더 빨리 계산할 수도 있게 된다<sup>[1]</sup>. 또한, 실시간으로 모양이 변화하는 곡선과 곡면에 대해서 유리할 것이다.

일단 곡선이 power 기저형태로 변환이 되고 난 후

에는 곡선 위의 점들을 계산하는 과정도 Horner's rule을 이용해서 짧은 시간 내에 행할 수 있다<sup>[2]</sup>. 뿐만 아니라 IGES에서는 B-spline곡선과 곡면을 Type113으로 지정하여 지원하고 있다<sup>[3]</sup>.

곡선이나 곡면에 임의의 점이 놓여 있는지 여부를 판단하는 문제(point inclusion문제)의 경우, 음함수형태의 곡선이나 곡면이 매개변수형태보다 계산상으로 유리하다. 이 때문에 매개변수형태의 곡선이나 곡면을 종결식(resultant)을 이용해서 음함수화(implicitization)할 필요성이 종종 생긴다<sup>[4]</sup>. 종결식을 사용하기 위해서는 대상이 되는 곡선이나 곡면의 power기저형태가 필요하게 되며<sup>[5]</sup>, 이 과정도 곡선이나 곡면의 차수가 커지게 되면 계산시간이 상당한 부담이 된다.

이 논문에서는 B-spline곡선을 power기저형태의 다항식 곡선들로 바꾸는 방법에 대해서 논의한다. 이 논문에서는 대상이 되는 곡선을 정적인 곡선(static

\*종신회원, 한양대학교 시스템융합공학부  
\*\*학생회원, 한양대학교 시스템융합공학부  
\*\*\*종신회원, 홍익대학교 산업공학과  
\*\*\*\*종신회원, Chrysler Corporation

curve)과 동적인 곡선(dynamic curve) 등 두 가지로 나누어 다룬다. 이 논문에서 정의하는 동적인 곡선 이라함은 곡선의 조정점들 중 한 개 이상이 시간에 따라서 다른 값들로 변화하는 곡선을 의미하며, 정적인 곡선은 그렇지 않은 곡선을 의미한다.

정적인 B-spline곡선에 대해서는 power기저형태의 곡선으로 바꾸는 몇 가지 방법이 문헌에 소개되어 있다. 먼저, knot refinement를 이용해서 B-spline곡선을 구간별 Bezier곡선들로 바꾼 뒤<sup>10)</sup> 기저변환을 통해서 각 Bezier곡선을 power기저형태의 다항식 곡선으로 변환하는 방법이 있다. 이 논문에서는 이 방법을 KR 알고리즘이라 부르겠다. 또 다른 방법으로는, 각 knot 구간에 해당하는 곡선상의 한 점에서 Taylor 전개(Taylor expansion)를 통해서 각 knot 구간별로 B-spline곡선을 직접 power기저형태의 다항식 곡선으로 변환하는 방법으로서 이 방법을 TE 알고리즘이라 부르겠다.<sup>10)</sup> 또 다른 방법으로는 NURBS 곡선을 행렬형태로 표현하는 방법이 있다<sup>11,12)</sup>. 그러나, 동적인 B-spline곡선에 대해서 이와 같은 문제들 논의한 것은 문헌에 보이지 않고 있다.

이 논문에서 제시하는 알고리즘의 아이디어는 필요한 정보를 재귀적이 아닌 방법으로 직접전개 (direct expansion) 하는 것으로서 우리는 이를 DE 알고리즘이라 부르겠다. DE알고리즘의 중요한 아이디어는 다음과 같다. 먼저, 각 knot 구간의 B-spline 기저함수를 구성하는 모든 1차식들을 찾아낸 후 이 1차식들을 직접 곱하고 더해서 전개함으로써 B-spline 기저함수들의 power기저형태를 구한 후, 다음단계에서 각 knot구간의 power형태의 기저함수 들과 관련된 조정점들을 적절히 곱하고 더함으로써, power기저형태의 B-spline 곡선을 구하게 된다<sup>13)</sup>.

이 논문의 구성은 다음과 같다. 먼저 2절에서는 B-spline 기저함수에 대한 기본적인 개념과 DE알고리즘을 만들어 내기 위해서 필요한 중요한 개념들을 소개한다. 3절에서는 각 구간의 기저함수 들을 구성하는 모든 1차식을 알아내기 위해서 필요한 경로 (path)들을 열거하는 알고리즘을 소개한다. 4절과 5 절에서는 정적인 곡선에 대한 DE알고리즘과 실험결과를 보이고, 6절과 7절에서는 동적인 곡선에 대한 경우를 보인 후 마지막으로 결론을 맺는다.

## 2. B-spline기저함수의 다른 표현방법

### 2.1 B-spline기저함수의 정의와 절단기저다항식(truncated basis polynomial)

(m+1)개의 knot을 갖는 p차의 B-spline 곡선은 다음과 같이 정의된다.

$$C(t) = \sum_{i=0}^{m-p+1} P_i N_{i,p}(t) \text{ for } 0 \leq t \leq 1 \text{ for} \quad (1)$$

P<sub>i</sub>와 N<sub>i,p</sub>(t)는 각각 조정점과 knot vector U={0, ..., 0, t<sub>p+1</sub>, ..., t<sub>m-p+1</sub>, 1, ..., 1}에서의 p차의 B-spline기저함수를 나타낸다<sup>14)</sup>. 그리고, 는 다음과 같은 재귀식으로 표현된다.

$$N_{i,p}(t) = \frac{t-t_i}{t_{i+p}-t_i} N_{i,p-1}(t) + \frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}} N_{i+1,p-1}(t) \quad (2)$$

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

(2)는 p차의 B-spline 기저함수가 p-1차의 B-spline 기저함수들의 선형결합으로 이루어져 있고, 각 계수는 매개변수 t에 관한 1차식임을 보여 준다.

Fig. 1은 3차인 경우 B-spline 기저함수의 계산을 그래프(graph)의 형태로 보여준다. 이 그래프는 directed graph로서 임의의 차수의 기저함수에 대해서 유사하게 정의될 수 있다. 식 (2)를 다음과 같이 바꾸어 보자.

$$N_{i,p}(t) = e_b(t;i,p)N_{i,p-1}(t) + e_v(t;i,p)N_{i+1,p-1}(t) \quad (3)$$

$$e_b(t;i,p) = \frac{t-t_i}{t_{i+p}-t_i}, \quad e_v(t;i,p) = \frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}} \quad (4)$$

**정의 1.** 변이호(transition edge)는 Fig. 1의 그래프의 에지(edge)이다. 만약 N<sub>i,p</sub>에서 N<sub>k</sub>로 향하는 변

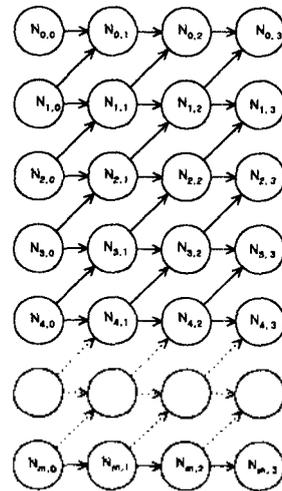


Fig. 1. A graph representation of calculation for cubic B-spline basis functions.

이호의 첨자들의 관계가  $k=i$ 이고,  $l=j+1$  라면 이를 수평변이호(horizontal transition edge)  $e_h(t;k,l)$ 라 하고,  $k=i-1$ 이고  $l=j+1$  라면 수직변이호(vertical transition edge)  $e_v(t;k,l)$ 라 한다.

일단 변이호가 수평 혹은 수직인지 여부가 결정되면, 해당 변이호의 값은 변이호의 도착 노드(node)의 아래 첨자로부터 쉽게 구할 수 있다. 예를 들어  $N_{3,2}$ 에서  $N_{2,3}$ 로의 변이호는 수직이고 값은  $e_v(t;2,3) = (t_{2+3+1}-t)/(t_{2+3+1}-t_{2,1}) = (t_6-t)/(t_6-t_3)$ 를 갖는다. 마찬가지로,  $N_{3,2}$ 에서  $N_{3,3}$ 으로 향하는 수평 변이호의 값은  $e_h(t;3,3) = (t-t_3)/(t_{3,3}-t_3) = (t-t_3)/(t_6-t_3)$ 이다.

**정의 2.** 절단기저다항식(truncated basis polynomial)  $h_{i,w}(t)$ 는  $N_{i,p}$ ,  $i=w-p, w-p+1, \dots, w$ ,  $w=p, p+1, \dots, m-p-1$ , 중  $[t_w, t_{w+1})$ 에서 0가 아닌 다항식이다.

예를 들면, Fig. 2(a)는  $m=11$ 인 경우의 3차 B-spline 들을 보여주고 있고, Fig. 2(b)는 그것들의 절단기저다항식을 보여준다.

$p$ 차의 B-spline 곡선의 경우 각각의 knot구간  $[t_i, t_{i+1})$ ,  $t_i < t_{i+1}$ , 에 대해서,  $p+1$ 개의 절단기저다항식이 존재한다. 예를 들어 Fig. 2(b)의 경우,  $[t_3, t_4)$ 에서 0이 아닌 절단기저다항식은 4개이다.

또한 다음과 같은 식이 성립하는데,

$$N_{i,p}(t) = \sum_{w=i}^{i+p} h_{i,w}(t)N_{w,0}(t)$$

즉 이는 같은  $i$  값의 절단기저다항식을 순서대로 적절히 합치면 B-spline이 된다는 것을 Fig. 2b에서 알 수 있다.

본 논문에서 제시하는 알고리즘은 각 knot구간의 절단기저다항식을 구성하는 모든  $e_h(t;k,l)$ 과  $e_v(t;k,l)$

들을 구한 후, 이들 간의 적절한 곱셈연산과 합을 통하여 power형태의 절단기저다항식을 구한다. 그 후 각 knot구간에서  $p+1$ 개의 power기저형태의 절단기저다항식들과 관련된 조정점들을 곱한 후 각 항들을 더해주면 그 구간에서의 power기저형태의 B-spline곡선이 구해지게 된다. 이러한 연산을 구간의 길이가 0이 아닌 모든 knot구간에 대해서 수행하면 주어진 B-spline 곡선을 power기저형태로 변환할 수 있게 된다.

문제는 어떻게 각 knot 구간의 절단기저다항식들을 구할 것인가인데, 이 문제는 다시 어떻게 구성 수평변이호와 수직변이호들을 구하느냐하는 문제가 된다. 이 논문에서 제시하는 알고리즘은 식 (2)와 같이 주어진 B-spline의 재귀 형태대신에 반복형태(repeated form)를 이용하게 된다.

예를 들어, knot구간  $[t_3, t_4)$ 에서의 절단기저다항식의 계산에 대해서 생각해 보자. Fig. 3은 Fig. 1에 있는 그래프의 부분그래프(subgraph)인데,  $[t_3, t_4)$ 의 절단기저다항식들을 계산하기 위한 triangular scheme을 보여준다. Fig. 3의 각 변이호는 식 (4)에 의해서 정의되는 1차 선형다항식으로서, 구간의 4개의 절단기저다항식을 형성하는 수평변이호와 수직변이호들이 된다. Fig. 3의 root노드와 leaf 노드는 각각 knot 구간과  $p$ 차의 절단기저다항식( $p$ 차의 B-spline곡선일 경우이며, 본 예에서는  $p=3$ )에 해당된다고 볼 수 있다.

**2.2 절단기저다항식의 계산을 위한 경로(path)**

**정의 3.**  $N_{i,j}$ 에서  $N_{k,i}$ 로의 경로(path)  $\pi$ 는  $N_{i,j}$ 에서  $N_{k,i}$ 로의 경로를 구성하는 순서가 있는 변이호(ordered transition edge)들의 집합이고, 경로집합(path set)은  $\Pi = \{\pi_a, a=1, 2, \dots, n\}$ 이다. 여기서  $n$ 은 가능한 모든

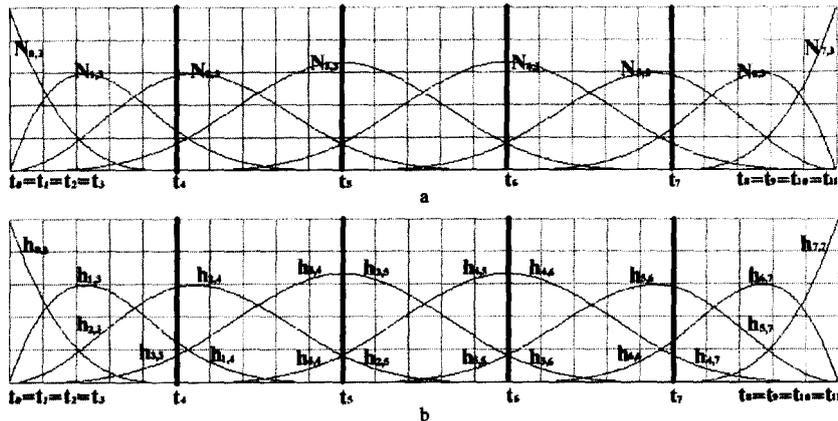


Fig. 2. An example of basis splines and truncated basis polynomials( $p=3, m=11$ ).

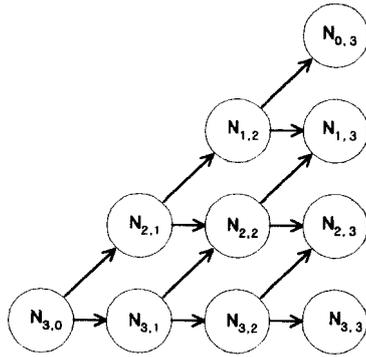


Fig. 3. Triangular scheme to get the truncated basis polynomials in  $[t_3, t_4]$ .

경로의 수이다.

**정의 4.**  $N_{i,p}$ 와  $N_{j,q}$  그리고  $\Pi$ 가 주어졌을 때,  $\cup \pi$ 는 그래프 프리미티브(graph primitive)라 부른다.

앞의 정의에서 합집합연산자(union operator)는 연산결과가 위상적으로 합쳐진 그래프(topologically merged graph)가 되도록 하는 가능한 모든 경로의 합(superposition)을 의미한다. 따라서, 하나의 절단기저다항식은 하나의 그래프 프리미티브와 일대일로 대응된다고 볼 수 있다. 예를 들어,  $[t_3, t_4]$ 에는 4개의 절단기저다항식이 존재하고, 각 절단기저다항식  $h_{i,j}(t)$ 는  $N_{3,0}$ 에서  $N_{i,j}$ ,  $i=0,1,2,3$ , 로의 경로들 중에 있는 모든 선형 다항식들을 곱해서 더해준 결과이다. 예를 들면,  $N_{3,0}$ 에서  $N_{1,3}$ 로 다음과 같은 세 개의 가능한 경로가 존재한다.

$$\begin{aligned} N_{3,0} &\rightarrow N_{2,1} \rightarrow N_{1,2} \rightarrow N_{1,3} \\ N_{3,0} &\rightarrow N_{2,1} \rightarrow N_{2,2} \rightarrow N_{1,3} \\ N_{3,0} &\rightarrow N_{3,1} \rightarrow N_{2,2} \rightarrow N_{1,3} \end{aligned}$$

그러므로, 절단기저다항식  $h_{1,3}(t)$ 는 다음과 같이 구할 수 있으며, 이는 Fig. 4에서도 쉽게 확인할 수 있다.

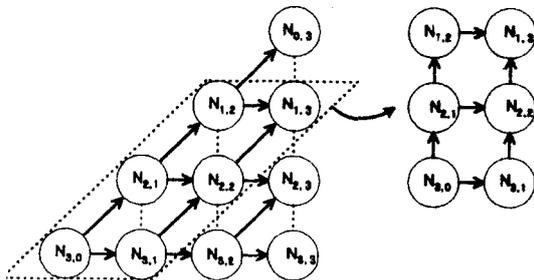


Fig. 4. A graph primitive corresponding to  $h_{1,3}(t)$ .

$$\begin{aligned} h_{1,3}(t) = &e_r(t;2,1)e_v(t;1,2)e_h(t;1,3)+ \\ &e_r(t;2,1)e_h(t;2,2)e_v(t;1,3)+ \\ &e_h(t;3,1)e_v(t;2,2)e_r(t;1,3)+ \end{aligned} \quad (5)$$

따라서 다음과 같은 정리가 성립한다.

**정리 1.** 구간  $[t_w, t_{w+1}]$ 에서의 절단기저다항식  $h_{i,w}(t)$ ,  $i=w-p, w-p+1, \dots, w$ ,  $w=p, p+1, \dots, m-p-1$ , 는 다음과 같이 표현되어 질 수 있다.

$$h_{i,w}(t) = \sum_{k=1}^{n_p} P_k(t) \quad (6)$$

$$P_k(t) = \prod_{i_1=1}^{n_h} e_h^{i_1}(t; p_h, q_h) \prod_{i_2=1}^{n_v} e_v^{i_2}(t; p_v, q_v)$$

여기서  $n_p$ 는  $N_{w,0}$ 에서  $N_{i,w}$ ,  $i=w-p, \dots, w$ , 로의 가능한 모든 경로들의 개수이고,  $n_h$ 과  $n_v$ 는 각 경로에 있는 수평과 수직 변이호의 개수를 의미하며,  $n_h+n_v=p$ 이 성립한다.  $p_h, q_h, p_v, q_v$ 는 각 경로내의 적절한 수평과 수직 변이호를 생성하기 위한 정수들이다.

**따름정리 1.** 앞의 정리에서 가능한 모든 경로의 개수  $n_p$ 는  $p!/(w-i)!(p-w+i)$ 이고, 하나의 knot 구간 존재하는 그래프 프리미티브의 개수는  $p$ 차의 곡선에 대해서  $p+1$ 이고, 이는 그 구간의 절단기저다항식의 개수가 된다.

Fig. 5는 그림 4에 존재하는 4개의 그래프 프리미티브들을 보여주며, 각 그래프 프리미티브 a, b, c, d는 knot 구간  $[t_3, t_4]$ 의 4개의 절단기저다항식  $h_{0,3}(t)$ ,  $h_{1,3}(t)$ ,  $h_{2,3}(t)$ ,  $h_{3,3}(t)$ 에 해당된다.

정리 1에 따르면, 만약 한 개의 knot 구간에 있는 모든 그래프프리미티브의 가능한 모든 경로만 알 수

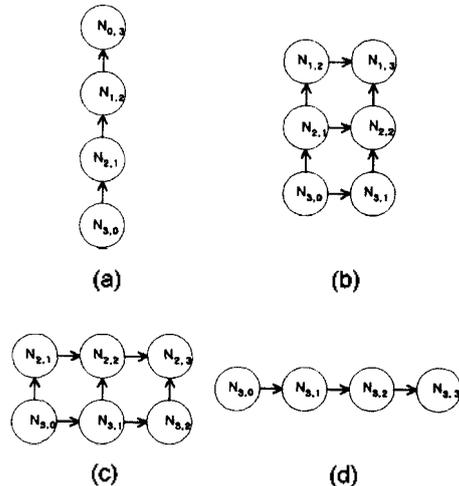


Fig. 5. Four graph primitives for Fig. 4.

있다면, 그 구간의 모든 절단기저다항식은 경로들에 있는 변이호들에 해당되는 1차식들의 곱을 더해줌으로써 구할 수 있다. 또한, 곡선의 차수가 정해지면 이 경로들은 모든 knot구간에 대해서 동일하므로 경로를 열거하는 과정은 전체 알고리즘에서 단 한번의 수행으로 충분하다.

### 3. 경로 열거(path enumeration)

하나의 그래프 프리미티브에 존재하는 모든 가능 경로들을 열거하는 문제는 그래프 프리미티브의 수평과 수직 변이호들의 개수만큼의 0과 1로 이루어진 크기  $p$ 의 모든 가능한 순열을 열거하는 문제로 바꿀 수 있다<sup>13)</sup>. 여기서  $n_h+n_v=p$ 이다.

하나의 knot구간에 존재하는 모든 가능경로를 열거하는 문제는 0과 1의 모든 가능한 조합으로 이루어진 크기  $p$ 의 순열을 열거하는 문제가 된다. 예를 들면, 3차 B-spline곡선의 하나의 knot구간에 대한 모든 가능경로는  $\{1, 1, 1\}, \{1, 1, 0\}, \{1, 0, 1\}, \{0, 1, 1\}, \{1, 0, 0\}, \{0, 1, 0\}, \{0, 0, 1\}, \{0, 0, 0\}$ 이다.

$p$ 차의 B-spline곡선에 대해서는 하나의 knot구간에 있는 가능경로의 개수가 다음과 같다.

$$\sum_{j=i-p}^i \frac{p!}{(i-j)!(p-i+j)!} = \sum_{k=0}^p \frac{p!}{k!(p-k)!} = \sum_{k=0}^p \binom{p}{k} = 2^p$$

3차 곡선에 대해서는  $2^3=8$ 로 위에서 구한 결과를 확인할 수 있다. 이전 절에서도 언급되었듯이, 곡선의 차수만 결정된다면, knot vector 크기에 관계없이 경로열거과정은 단 한번으로 충분하다. 이러한 성질은 정적인 곡선뿐만 아니라 동적인 곡선에 대해서도 동일하게 성립한다.

### 4. 정적인 곡선에 대한 DE 알고리즘

일단 모든 knot구간에 대해서 절단기저다항식을 구하면, 주어진 B-spline곡선을 power기저형태의 다항식 곡선으로 바꾸는 일은 각 knot 구간에 대해서 절단기저다항식과 대응되는 조정점들을 곱해서 더해줌으로써 가능해진다. 구간  $[t_i, t_{i+1})$ 에서 power기저형태의 곡선으로의 변환은 식 (7)로 표현할 수 있다.

$$C_i(t) = \sum_{j=i-p}^i h_{j,w}(t)P_j \tag{7}$$

여기서  $P_j$ 는  $h_{j,w}(t)$ 가 한 부분이 되는 기저함수  $N_{j,p}(t)$

에 대응되는 조정점이다. 또한,  $w=p, p+1, \dots, m-p-1$ 로서  $w=i$ 이다. 예를 들면, 구간에서 정의되는 3차 B-spline 곡선 조각은  $C_3(t)$ 이며,  $C_3(t)=h_{0,3}(t)P_0+h_{1,3}(t)P_1+h_{2,3}(t)P_2+h_{3,3}(t)P_3$ 이 된다. 여기서 각 절단기저다항식들은 power 기저형태로 이미 구해져 있다.

따라서, 각각의 knot구간에 대해서 위와 같은 과정을 반복한다면 B-spline곡선  $C(t)$ 전체의 구간별 power 기저형태의 다항식을 구할 수 있으며, 이는 다음과 같이 표현될 수 있다.

$$C(t) = \sum_{i=p}^{m-p-1} C_i(t)N_{i,0}(t) \tag{8}$$

### 5. 정적인 곡선에 대한 DE알고리즘의 수행도 분석

정적인 B-spline곡선에 대한 DE알고리즘의 성능을 평가하기 위해서 KR알고리즘 및 TE알고리즘과 비교하는 실험을 시행하였다. Fig. 6은 knot vector 크기가 50인 곡선들의 차수에 따른 계산시간을 세가지 알고리즘에 대해 비교한 결과를 보여주고 있다. 이 그림을 통해서 DE알고리즘은 KR알고리즘에 비해서 저차의 곡선의 경우 수행시간이 더 짧다는 것을 알 수 있으며, DE의 계산시간은 차수가 올라감에 따라서 급격하게 증가하는 것을 볼 수 있다.

DE알고리즘의 계산시간이 매우 빠르게 증가하는 패턴을 보이는 것에 반해 KR알고리즘과 TE알고리즘이 선형적으로 증가하는 패턴을 보이는 이유는 곡선의 차수가  $p$ 일 때 각 knot구간에 대한 경로의 개수가  $2^p$ 이므로, 차수가 하나 증가할 때 마다 절단기저다항식을 계산하기 위한 시간이 배로 증가하기 때문이다.

Fig. 7은 knot vector 크기에 관계없이 3차와 4차의 곡선에 대해서 DE알고리즘이 KR 알고리즘에 비해

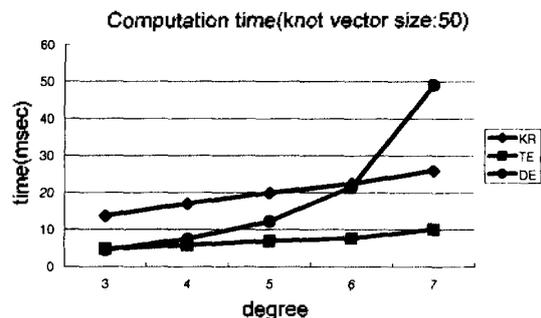


Fig. 6. Time vs. degrees (knot vector size: 50)

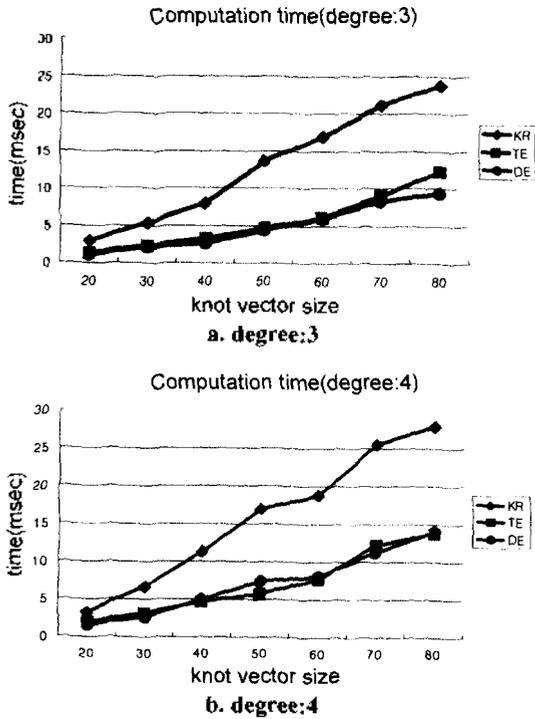


Fig. 7. Time vs. knot vector sizes.

매우 우수함을 보여 주고 있다.

### 6. 동적인 곡선에 대한 DE 알고리즘

**정의 5.** 동적인 곡선(dynamic curve),  $C_d(t)$ 는 하나 이상의 조정점이 시간에 따라서 변화하는 B-spline 곡선이다.

즉, 동적인 B-spline 곡선  $C_d(t)$ 는 아래와 같이 표현될 수 있다.

$$C_d(t) = \sum_{i \in I} B_i(t)P_i + \sum_{j \in J} B_j(t)\tilde{P}_j \quad (9)$$

여기서  $I$ 는 변화가 없는 조정점의 첨자집합이며  $J$ 는 변화하는 조정점의 첨자집합을 의미한다. 그리고,  $B_i(t)$ 와  $P_i$  및  $\tilde{P}_j$ 는 기저함수 및 이와 대응되는 조정점을 나타낸다.

식 (9)를 power기저형태로 변환하는 방법에는 여러가지가 있을 수 있다. 예를 들면 조정점들이 변화할 때 마다 전체 knot 구간에 대해서 곡선을 다시 power기저 형태로 변환할 수도 있을 것이나 이는 매우 비 효율적이다. 따라서 변화하는 조정점이 영향을 주는 곡선의 knot구간만을 찾아서, 그 구간만 변

화된 조정점만을 고려하여 power기저형태로 바꾸는 방법이 더 효율적일 것이다.

이와 같이 곡선의 변한 부분과 변화지 않은 부분을 나누어서 접근할 때도 앞서 정적인 곡선에서 사용했던 KR과 TE 알고리즘을 그대로 적용할 수 있다. 만약 KR 알고리즘을 이용한다면, 곡선의 모양이 변화한 모든 knot 구간에 대해서 knot refinement와 기저변환을 다시 수행해야 하며, TE 알고리즘을 적용할 경우에도 모양이 변화한 모든 knot 구간의 다항식 곡선의 계수를 다시 구해야 되는데 이때 비분정보가 필요하게 된다. 그러나, DE 알고리즘의 경우에 절단기저다항식이 모든 knot구간에서 고정되어 있으므로 동적인 곡선의 경우에 있어서 다른 두 알고리즘에 비해 더 큰 계산상의 이득이 있을 것으로 기대할 수 있다.

먼저 초기에 주어진 곡선을 생각하자. 이때의 곡선은 정적인 상태의 곡선이다. 이를 앞에서 설명한 방법에 따라서 DE 알고리즘을 통하여 power기저를 갖는 구간별 다항식곡선으로 변환하여 식 (8)과 같이 구하였다고 가정하자. 즉, 이 과정은 지금부터 설명할 동적인 곡선에 대한 DE 알고리즘의 preprocessing 이 된다.

주어진 곡선 가 시간에 따라 변화하고 있는 동적인 곡선을 라 하자. 식 (9)와 같이 주어진 는 원래의 조정점으로부터 변화된 조정점까지의 차벡터(difference vector)를 이용해서 다음과 같이 다시 표현될 수 있다.

$$C_d(t) = \sum_{k \in K} B_k(t)P_k + \sum_{j \in J} B_j(t)D_j \quad (10)$$

여기서  $K=I \cup J$ , 즉,  $P_k, k \in K$ , 는 원래의 곡선  $C(t)$ 의 모든 조정점들을 의미하며,  $D_j = (\tilde{P}_j - P_j)$ 는 변화된 조정점의 변화량을 의미하는 차벡터이다. 즉, 식 (10)은 원래의 곡선  $C(t)$ 에 변화된 조정점들의 변화량을 관련된 기저함수들과 곱해서 다 더해준 것이라 의미이다.

따라서,  $C_d(t)$ 는 변화된 조정점에 의해 영향을 받는 knot 구간과 받지 않는 knot 구간으로 나누어 질 수 있으며, 이를 식으로 표현하면 다음과 같다.

$$C_d(t) = \sum_{m \in M} C_m(t)N_{m,0}(t) + \sum_{n \in N} \tilde{C}_n(t)N_{n,0}(t) \quad (11)$$

여기서  $M$ 과  $N$ 은 각각 변화된 조정점에 의해 영향을 받지 않는 knot 구간과 영향을 받는 knot구간의 첨자 집합이다.

절단기저다항식을 이용하면  $\tilde{C}_n(t)$ 는 다음과 같이

표현할 수 있다.

$$\tilde{C}_n(t) = \sum_{q \in Q} h_{q,n}(t) P_q + \sum_{r \in R} h_{r,n}(t) \tilde{P}_r \quad (12)$$

여기서  $Q$ 와  $R$ 은 각각  $\tilde{C}_n(t)$ 의 조정점중 변화하지 않은 조정점과 변화한 조정점을 나타내는 점자의 집합이다. 즉,  $\tilde{C}_n(t)$ 는 각각 관련된  $p+1$ 개 ( $p$ 는 곡선의 차수의 조정점들과 역시  $p+1$ 개의 절단기저다항식의 곱들의 합인데, 조정점들 중 값이 변화한 것과 변화하지 않은 것들을 분리하여 정리한 것이다. 식 (12)은 다시

$$\tilde{C}_n(t) = \sum_{s \in S} h_{s,n}(t) P_s + \sum_{r \in R} h_{r,n}(t) (\tilde{P}_r - P_r) \quad (13)$$

로 재정리 될 수 있다. 여기서  $S \equiv QU \cup R$ 이며,  $|S|=p+1$ 이다. 즉,  $P_s, s \in S$ ,는 곡선이 변화하기 전의 이 구간의 곡선조각  $C_n(t)$ 를 정의하는 모든 조정점들을 의미한다. 따라서, 식 (13)는 다음과 같이 재정리 될 수 있다.

$$\tilde{C}_n(t) = C_n(t) + \sum_{r \in R} h_{r,n}(t) D_r \quad (14)$$

$D_r = \tilde{P}_r - P_r$ 은 변화한 조정점의 원래의 위치로부터의 변위를 의미하는 차벡터이다. 즉, 식 (14)은, 특정 knot 구간에서 변화된 곡선조각의 power 기저형태의 다항식  $\tilde{C}_n(t)$ 은 변화한 조정점의 변위만 그 조정점에 관련된 절단기저다항식에 곱해서 주어진  $C_n(t)$ 에 더해지면 된다는 것을 의미한다. 이때  $C_n(t)$ 는 preprocessing 단계에서 주어짐을 유의할 필요가 있다.

따라서,  $C_d(t)$ 는 식 (11)에서 의미하는 바와 같이 원래의 B-spline 곡선에서 곡선의 모양이 변화하는 모든 구간을 찾아서 그 각각에 대해서 위의 작업을 해 주면 된다. 그리고, 곡선의 모양이 변화하는 구간은 변위가 일어난 조정점들의 점자로부터 직접 구해 질 수 있다. 이 과정은 프로그램화하기도 매우 간단하다.

## 7. 동적인 곡선에 대한 DE 알고리즘의 수행도 분석

동적인 B-spline곡선을 DE알고리즘을 통해서 knot 구간별로 power기저형태의 다항식 곡선으로 바꾸는 작업은

- preprocessing 과정
- 식 (14)

와 같이 두 부분으로 나누어 진다.

예를 들어  $p$ 차의 B-spline곡선에서 한 개의 조정점만을 변화 시켰다면, 해당 조정점은 최대  $(p+1)$ 개의 knot구간에 영향을 준다. 각 knot 구간에서  $(p+1)$ 개의 절단기저다항식은 고정되어 있기 때문에, 먼저 원래의 조정점에서 변화한 조정점까지의 차벡터와 절단기저다항식을 곱하여 이것을 기존의 power기저형태의 다항식곡선에 더해줌으로써 그 구간의 변화된 B-spline곡선에 대한 power기저형태의 다항식 곡선을 구할 수 있게 된다. 이러한 연산을  $(p+1)$ 개의 knot구간에 대해서 모두 수행해 주면 원하는 power형태의 곡선을 구할 수 있다.

이때, 계산을 위해서 고려해야 될 knot 구간은  $(p+1)$ 개이며, 각 구간에서는  $(p+1)$ 번의 곱셈연산과 역시  $(p+1)$ 번의 덧셈연산만이 필요하다. 그러므로, 한 개의 조정점이 변화한  $p$ 차의 B-spline 곡선의 경우 새로운 구간별 power 기저형태의 다항식 곡선을 얻기 위해서는  $O(p^2)$ 만큼의 연산만이 필요하다. (차벡터를 구하는 것은  $O(1)$ 의 작업이다.)

만약 이 문제에 KR 알고리즘을 이용한다면  $(p+1)$ 개의 구간에 대해서 knot refinement와 기저변환을 모두 다시 수행해야 되므로 DE의 경우에 비해서 많은 계산시간이 필요하다.

TE 알고리즘의 경우도 마찬가지로  $(p+1)$ 개의 구간에 대해서 다항식 곡선을 다시 계산해야 된다. 하나의 knot 구간에 대해서 재 계산이 필요한  $(p+1)$ 개의 계수가 있고, 각 계수의 계산에는 미분 값 계산과 factorial함수의 계산이 필요하게 된다. B-spline 곡선의 미분 값 정보를 얻기 위해서는 주어진 매개변수 값에서 호도그래프를 평가함으로써 가능한데, Horner's rule을 이용한다면  $O(p)$ 만큼의 연산이 필요하다. factorial함수의 계산 역시  $O(p)$ 의 계산을 필요로 하는 작업이다. 그러나, 이 경우 호도그래프 역시 다항식형태로 변환되어 있음을 전제로 하는 작업임을 유의할 필요가 있다. 따라서, TE 알고리즘을 이용할 경우 빨라야  $O(p^3)$ 의 연산이 필요하다.

Fig. 8은 3,4,5,6차의 곡선에 대해서 변화하는 조정점의 개수와 각 알고리즘의 계산시간의 관계를 실험적으로 보여준다. 이 실험은 각 알고리즘을 이용하여 변화된 조정점에 의해 영향받는 최소한의 knot구간만을 power기저형태로 변환하도록 설계되었다.

그림에서 알 수 있듯이, DE 알고리즘의 계산시간은 다른 알고리즘의 계산시간에 비해서 무시할 만큼 아주 작아서 마치 계산이 수행되지 않는 것처럼 보일 정도이다. 그리고, DE알고리즘과 다른 알고리즘

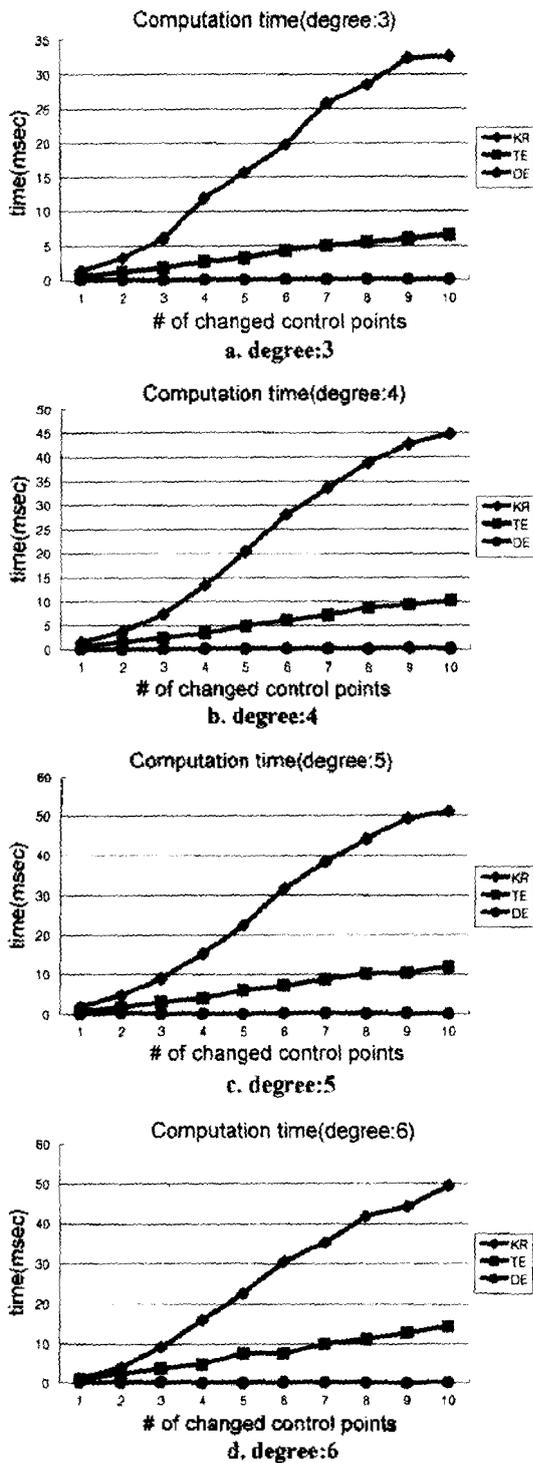


Fig. 8. Computation time vs. the number of changed control points(The number of all control points of given B-spline curve is 10.)

의 수행도의 차이는 변화된 조정점의 개수가 증가하면 할수록 매우 급격하게 커짐을 알 수 있다. 이런 현상은 곡선의 차수가 증가하면서 더욱 확실히 나타나서 DE 알고리즘이 상대적으로 매우 유리하다는 것을 알 수 있다.

### 8. 결 론

이 논문에서는 B-spline곡선을 구간별로 power기저형태의 다항식 곡선으로 변환하는 새로운 알고리즘을 제시했으며, 절단기저함수나 변이호화 같은 몇 가지 새로운 개념들을 정의함으로써 이론적인 기반도 제시하고 있다.

제안하는 알고리즘은 정적인 B-spline곡선의 경우 저차의 곡선에 대해서 기존의 KR알고리즘에 비해 수행속도가 빠르며 TE알고리즘과는 대등한 수행속도를 보인다. 특히, 곡선의 모양이 동적으로 변화하는 경우에는 제안된 DE알고리즘의 수행속도는 기존의 타 알고리즘에 비해서 월등한 수행속도를 보이고 있다. 이 논문에서는 이론적인 분석과 실험을 통해서 DE 알고리즘이 다른 알고리즘에 비해 계산시간상의 이득이 있다는 것을 보여 주었다.

DE 알고리즘의 특징은 계산 방법상으로는 B-spline기저함수를 계산하기 위한 기존의 재귀식대신에 반복형태를 이용한다는 점과 데이터 저장방법상으로는 절단기저다항식을 이용한다는 점이다. 이러한 이유 때문에 DE 알고리즘을 B-spline곡면으로 확장한다면 더 큰 이득이 있으리라 예상된다.

DE 알고리즘을 B-spline곡면과 유리 B-spline곡선으로 확장하는 일은 어렵지 않은 일이며, 차수가 증가함에 따라서 계산시간이 급격히 증가하는 DE 알고리즘의 약점을 보완하는 일과 power기저형태로 변환시 생길 수도 있는 수치에러에 대한 고찰이 차후의 연구과제가 될 것이다.

### 감사의 글

이 연구는 한국과학재단 지정 최적설계기술연구센터의 연구비 지원으로 수행되었습니다.

### 참고문헌

1. Kim, D.-S., Lee, S.-W., and Shin, H., "A cocktail algorithm for planar Bezier curve intersections," *Computer Aided Design*, Vol. 30, No. 13, pp. 1047-1051, 1998.

2. Farin, G., *Curves and Surfaces for Computer-Aided Geometric Design*, 3<sup>rd</sup> Ed., Academic Press, 1997.
3. *The Initial Graphics Exchange Specification (IGES)*, Version 5.2, ANSI Y14.26M, 1993.
4. Bloomenthal, J., *Introduction to implicit surfaces*, Morgan Kaufmann Publishers, Inc., 1997.
5. Sederberg, T. W., *Implicit and parametric curves and surfaces for computer aided geometric design*, Ph. D. Thesis, Purdue University, 1983.
6. Boehm, W. and Prautzsch, H., "The insertion algorithm," *Computer-Aided Design*, Vol. 12, No. 4, July, pp. 58-59, 1985.
7. Boehm, W., "On the efficiency of knot insertion algorithms," *Computer Aided Geometric Design*, Vol. 2, Nos. 1-3, July, pp.141-143, 1985.
8. Cohen, E., Lyche, T. and Riesenfeld, R., "Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics," *Computer Graphics and Image Processing* Vol. 14, No. 2, pp. 87-111, 1980.
9. Goldman, R.N., "Blossoming and knot insertion algorithm for B-spline curves," *Computer Aided Geometric Design*, Vol. 7, pp. 69-81, 1990.
10. Lasser, D. and Hoschek, J., *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, 1993.
11. Choi, B. K., Yoo, W. S. and Lee, C. S., "Matrix representation for NURB curves and surfaces," *Computer-Aided Design*, Vol. 22, No. 4, May, pp.235-240, 1990.
12. Yamaguchi, F., *Curves and Surfaces in Computer Aided Geometric Design*, Springer-Verlag, 1988.
13. Kim, D.-S., Ryu, J., Jang, T., Lee, H., and Shin, H., "Conversion of a B-spline curve into a set of piecewise polynomial curves in a power form," *Korea Israel Bi-National Conference on Geometric Modeling and Computer Graphics in the World Wide Web Era*, pp.195-201, Seoul, Korea, 1999.
14. Piegl, L. and Tiller, W., *The NURBS Book*, Springer, 1995.



**류 중 현**

1997년 한양대학교 산업공학과 학사  
 1999년 한양대학교 산업공학과 석사  
 1999년-현재 한양대학교 산업공학과 박사 과정  
 관심분야: geometric modeling, optimization, computational geometry, computer graphics



**이 현 찬**

1978년 서울대학교 산업공학과 학사  
 1980년 한국과학기술원 산업공학과 석사  
 1988년 The University of Michigan 산업공학과 박사  
 1980년-1983년 세아제강 기획실  
 1988년-1991년 전자통신연구원 자동설계연구실 실장  
 1991년-현재 홍익대학교 정보산업공학과 부교수  
 관심분야: geometric modeling, computer graphics, engineering database, information systems



**신 하 용**

1985년 서울대학교 산업공학과 학사  
 1987년 한국과학기술원 산업공학과 석사  
 1991년 한국과학기술원 산업공학과 박사  
 1991년-1993년 LG 생산기술원 선임연구원  
 1993년-1997년 (주) 큐빅테크 연구소 부소장  
 1997년-현재 DaimlerChrysler senior software specialist  
 관심분야: 형상모델링, CAD/CAM, PDM, 시스템 모델링



**김 덕 수**

1982년 한양대학교 산업공학과 학사  
 1983년 New Jersey Institute of Technology 산업공학과 석사  
 1990년 The University of Michigan 산업공학과 박사  
 1989년-1991년 Schlumberger Technology CAD/CAM Co. Senior Software Engineer  
 1991-1995년 삼성 종합 기술원 선임 연구원  
 1995년-현재 한양대학교 산업공학과 부교수  
 관심분야: geometric modeling, computational geometry, STEP and Internet application



**장 태 범**

1997년 한양대학교 산업공학과 학사  
 1999년 한양대학교 산업공학과 석사  
 1999년-현재 한양대학교 산업공학과 박사 과정  
 관심분야: geometric modeling, optimization, computer graphics, computational geometry