

## 무작위 데이터 근사화를 위한 유계오차 B-스플라인 근사법

박상근<sup>†</sup>

한국교통대학교 기계공학과

### An Error-Bounded B-spline Fitting Technique to Approximate Unorganized Data

Sangkun Park<sup>†</sup>

Department of Mechanical Engineering, Korea National University of Transportation

Received 22 November 2011; received in revised form 28 June 2012; accepted 3 July 2012

#### ABSTRACT

This paper presents an error-bounded B-spline fitting technique to approximate unorganized data within a prescribed error tolerance. The proposed approach includes two main steps: least-squares minimization and error-bounded approximation. A B-spline hypervolume is first described as a data representation model, which includes its mathematical definition and the data structure for implementation. Then we present the least-squares minimization technique for the generation of an approximate B-spline model from the given data set, which provides a unique solution to the problem: overdetermined, underdetermined, or ill-conditioned problem. We also explain an algorithm for the error-bounded approximation which recursively refines the initial base model obtained from the least-squares minimization until the Euclidean distance between the model and the given data is within the given error tolerance. The proposed approach is demonstrated with some examples to show its usefulness and a good possibility for various applications.

**Key words:** B-spline hypervolume, Error-bounded approximation, Least-squares minimization

## 1. 서 론

일반적으로 데이터 근사의 궁극적 목표는 무작위로 분포된 이산 형태의 입력 데이터를 근사적으로 표현할 수 있는 효율적인 계산 모델을 생성하는데 있다. 이 데이터 근사 기술은 과학 분야를 포함하여 수많은 분야, 예를 들어 응용수학, 컴퓨터 과학, 지질학, 생물학, 공학, 심지어 경영 실무 등

의 분야에서 데이터 분석 및 가시화 등을 위한 기반 기술로서 주요한 역할을 담당해왔다. 이들 분야의 가장 대표적인 적용 사례들을 살펴보면 다음과 같다. 1) CAGD 분야에서 곡선 접합과 곡면 근사, 2) CAE 분야에서 유한요소 해석 후 계산된 응력 데이터를 기반으로 임의 위치에서의 응력 계산 및 가시화, 3) 역공학 분야에서 3차원 스캐너에서 획득한 측정 점 군을 가지고 폴리곤 곡면 재건, 4) 의료분야에서 MRI 혹은 CAT 스캐닝 장비로부터 추출한 의료 영상을 기반으로 인체 형상을 재건, 5) 지질학 분야에서 측정된 지질 정보로부터

<sup>†</sup>Corresponding Author, [skpark@ut.ac.kr](mailto:skpark@ut.ac.kr)  
©2012 Society of CAD/CAM Engineers

등고선 등의 지도 생성, 6) CFD 분야에서 계산된 유동 데이터에서 와류와 같은 특이한 유동 구조 추출, 7) 최적화 분야에서 근사 최적화 구현에 필요한 반응 표면 생성. 이러한 다양한 적용 사례들의 성공 여부는 효과적인 데이터 모델링 혹은 근사 방법에 있으며, 생성된 근사모델(기법)은 입력 데이터 내부에 숨어있는 특이 구조의 추출 및 분석, 그리고 그래픽 처리를 통한 데이터 가시화 등을 지원할 수 있어야 한다.

지난 수십 년 동안 연구 발표되어 왔던 기존의 근사화 모델(기법)들을 살펴보면 모든 응용 분야에 적용 가능한 일반화된 형태의 근사화 기법은 없다<sup>[1,2]</sup>는 것이다. 즉 해당 분야의 특성에 따라 적절한 근사화 방법을 찾아야 하며 또한 데이터의 차원, 크기, 분포, 순서 등에 의존하여 경우마다 그 결과가 다르다는 것이다. 한편, 데이터 근사화에 있어 중요한 성능 요소 중에 하나가 근사 정밀도이다. 근사모델이 얼마나 정확하게 원시 데이터를 근사하고 있는가에 관한 문제로서 근사 정밀도를 정량적으로 평가하기 위해 보통 근사모델과 입력 데이터 사이의  $L^2$ -error 혹은  $L^\infty$ -error 값을 계산한다. 여기서 근사오차인  $L^2$ -error 혹은  $L^\infty$ -error는 작을수록 반드시 최선이 되는 것은 아니다. 근사 완만성에 관한 요구 조건이 있다면 함께 고려되어야 하며, 이밖에 메모리 사용량, 계산 복잡성 및 안정성 등에 관한 요구 사항도 고려되어야 한다.

본 연구에서는 근사 완만성과 정밀도를 함께 고려하는 근사화 기법보다 근사 정밀도에 더 초점을 두어 사용자가 근사오차를 조절할 수 있도록 입력 데이터를 근사화하는 B-스플라인 기반의 유계오차 근사기법(error-bounded approximation)을 제안하고자 한다. 다차원 공간 상에 무질서하게 분포된 이산형 입력 데이터 점이 존재하고, 이 데이터 위치에 다수 개의 데이터 값이 주어졌을 때, 이들 데이터와 근사모델 간의 근사오차가 주어진 허용오차 범위 내에 존재할 수 있도록 B-스플라인 근사모델을 생성하는 근사 기법에 관한 것이다. 본 연구에서 다루는 입력 데이터는 매우 일반화된 형태의 무작위 데이터로서 적어도 데이터 특성에 제한을 받는 기존 방법들과 차별화되며, 또한 이러한 일반화 형태의 데이터에 관해 유계오차 근사화를 시도하는 선행 연구로서 기존 관련 연구에서 찾아보기 힘든 특이성을 있다.

현재까지 알려진 대표적인 데이터 근사화 모델로서 Shepard<sup>[3]</sup> 모델과 RBF(radial basis function)<sup>[4]</sup> 모델이 있다. Shepard 모델은 데이터 점과의 거리에 반비례하는 가중치를 사용하여 데이터 값을 보간하는 방식이다. 이 방식은 입력 데이터의 국부적 수정 시 이를 반영시키기 위해 모든 데이터 점에서의 가중치 값을 다시 계산해야 하는 커다란 문제점을 가지고 있다. 이러한 문제점을 해결하기 위해 Franke와 Nielson은 MQS(modified quadratic Shepard)<sup>[5]</sup>를 제시하였는데, Shepard 방식에서 가중치 계산을 국부화시켰고, 데이터 값 자리에 가중치 최소화승법에 의해 계산되는 2차 다항식을 삽입하여 사용하였다.

RBF 모델은 일종의 볼륨 스플라인 모델로서 거리 개념의 다양한 기저함수를 사용하며 주어진 입력 데이터를 RBF 모델에 대입하여 미지의 계수를 계산함으로써 데이터 근사화를 완성한다. 이 방식은 미지수 계산을 위해 유도된 선형대수 방정식의 행렬이 이론적으로 정칙 행렬이지만 수치적으로 불량조건(ill-conditioned) 문제이기에, 대용량 데이터의 경우에 많은 계산 시간을 요구하며 안정되지 못한 수치 결과를 보여준다. 이러한 단점을 해결하기 위해 CS-RBF(compactly supported RBF)<sup>[6]</sup> 모델이 제안되었다. 이 모델은 기저함수의 영향 범위를 기존 무한 영역에서 한정된 유한 영역으로 변경함으로써 국부적 성질을 가지게 하여 보다 안정된 수치 계산을 유도한 것이다. 이밖에 데이터 표현 모델로서 Hardy의 multiquadratics<sup>[7]</sup> 모델을 있고, Duchon는 곡률 에너지 최소화 개념의 thin plate splines<sup>[8]</sup>을 제안하였다.

## 2. 근사화 표현모델

무작위 랜덤 데이터의 효율적 근사화를 위해 본 연구에서 사용한 근사화 표현모델은 B-스플라인 하이퍼볼륨(B-spline hypervolume)<sup>[9]</sup>이다. 이 표현모델은 B-스플라인 곡선의 매개변수(parameter) 공간을 한 차원 확장하여 B-스플라인 곡면을 정의하듯이 매개변수 공간을 임의의  $e$  차원으로 확장한 매개변수 표현식이며, 나아가 B-스플라인 곡선 혹은 곡면의 3차원 조정점(control point) 벡터의 크기를 임의의  $s$ 개로 확장한 다속성의 벡터장을 기술할 수 있는 하이퍼볼륨 표현모델이다.

### 2.1 B-스플라인 표현모델 소개

본 연구에서 사용한 표현모델인 B-스플라인 하이퍼볼륨은 다차원의 스칼라 장(scalar field)을 격자 형태의 조정점 망(control net)에 의해 정의하는 일종의 텐서곱 스플라인(tensor product spline) 형태를 갖는다. 아래의 식 (1)은 이 볼륨 표현모델의 수학적 정의를 보여주고 있다. 즉  $e$ 차원 유클리드 공간  $\mathbf{E}^e$  상에  $s$ 개의 스칼라 장이 존재할 때, 이를 식 (1)의 B-스플라인 하이퍼볼륨 표현식에 의해 표현할 수 있다.

$$\mathbf{A}(u_1, \dots, u_e) = \sum_{i_1=0}^{n_1-1} \dots \sum_{i_e=0}^{n_e-1} \mathbf{A}_{i_1 \dots i_e} N_{i_1, \mathbf{T}_1}^{k_1}(u_1) \dots N_{i_e, \mathbf{T}_e}^{k_e}(u_e) \quad (1)$$

여기서  $u_j$ 는  $e$ 차원 유클리드 공간 상의  $j$ 방향 좌표이며,  $\mathbf{A}_{i_1 \dots i_e} \in \mathbf{R}^s$ 는 인덱스 좌표  $(i_1, \dots, i_e)$ 에 해당하는 조정점을 나타낸다.  $n_j$ 와  $k_j$ 는  $u_j$ 방향으로의 조정점 개수 및 오더(order)를 말하고,  $N_{i_j, \mathbf{T}_j}^{k_j}(u_j)$ 는 오더가  $k_j$ 인 정규화된 B-스플라인 기저함수(normalized B-spline basis function)로서 아래의 절점 벡터(knot vector) 상에서 정의된다.

$u_j$  방향 절점 벡터:

$$\mathbf{T}_j = \{t_j^{(i)}\}_{i=0}^{i_j=n_j+k_j-1} \quad (j=1, \dots, e)$$

위의 식 (1)은 앞으로 소개될 근사법 알고리즘의 간결한 수식 전개를 위하여 다음의 식 (2)와 같이 간결한 표현 형태로 다시 쓸 수 있다.

$$\mathbf{A}(\mathbf{u}) = \sum_{I=0}^{N-1} \mathbf{A}_I \mathbf{N}_I(\mathbf{u}) \quad (2)$$

여기서

$$N = n_1 \times \dots \times n_e$$

$$\mathbf{A}_I = \mathbf{A}_{i_1 \dots i_e}$$

$$\mathbf{N}_I(\mathbf{u}) = N_{i_1, \mathbf{T}_1}^{k_1}(u_1) \times \dots \times N_{i_e, \mathbf{T}_e}^{k_e}(u_e)$$

$$I = i_1 + (n_1) \cdot i_2 + (n_1 \times n_2) \cdot i_3 + \dots + (n_1 \times \dots \times n_{e-1}) \cdot i_e$$

### 2.2 자료구조 및 특징

본 연구에서 구현한 B-스플라인 하이퍼볼륨의 자료구조를 간략히 살펴보면 Table 1과 같다.

즉 자료구조의 구성요소로서 1) 유클리드 공간의 차원, 2) 스칼라 장의 개수, 3) 각 방향의 조정점 개수, 4) 각 방향의 B-스플라인 기저함수 오더(order), 5) 각 방향의 절점벡터, 6) 총  $N$ 개의 조정점(각 조정점은  $s$ 개 크기를 가진 배열임)가 있고, 이들에 의해 식 (1) 혹은 (2)가 정의된다. 그리고 각 구성요소가 요구하는 자료 개수는 Table 1과 같으며 정수형 및 실수형 자료의 크기를 각각 4 bytes라 할 때 식 (1) 혹은 (2)가 필요로 하는 메모리 크기는 아래와 같이 계산된다.

$$\left\{ 2 \times (1+e) + \sum_{j=1}^e (n_j + k_j) + s \times (n_1 \times \dots \times n_e) \right\} \times 4 \text{ bytes}$$

## 3. 최소자승 근사기법

순서 없이 무질서하게 분포된  $M$ 개의 데이터  $\mathbf{D}$ 가 식 (3)과 같이 주어졌을 때, 이를 최소자승 근사기법을 사용하여 식 (2)에 의해 근사적으로 표현하는 B-스플라인 최소자승 근사기법 알고리즘<sup>[10]</sup>을 소개하고자 한다.

$$\mathbf{D} = \{(\mathbf{x}_i, \mathbf{f}_i) | \mathbf{x}_i \in \mathbf{E}^e, \mathbf{f}_i \in \mathbf{R}^s\} \quad (i=0, \dots, M-1) \quad (3)$$

여기서  $\mathbf{x}_i$ 는 데이터 위치(data site),  $\mathbf{f}_i$ 는 데이터 값(data value)라고 부른다.

**Table 1.** Data structure of a B-spline hypervolume

Member data	Description	Data type	Number of data
$e$	Space dimension	Integer	1
$s$	Number of scalar fields or vector size of $\mathbf{A}_I$	Integer	1
$n_j (j=1, \dots, e)$	Number of control points (for the j-th direction)	Integer: 1d-array	$e$
$k_j (j=1, \dots, e)$	Order of B-spline function (for the j-th direction)	Integer: 1d-array	$e$
$\mathbf{T}_j (j=1, \dots, e)$	Knot vector (for the j-th direction)	Float: 2d-array	$\sum_{j=1}^e (n_j + k_j)$
$\mathbf{A}_I (I=0, \dots, N-1)$	Control points (cf. the symbol $N$ denotes the total number.)	Float: 2d-array	$s \times (n_1 \times \dots \times n_e)$

식 (2)의 B-스플라인 하이퍼볼륨은 Table 1에서 기술한 바와 같이 6개의 구성요소에 의해 정의된다. 즉  $e, s, n_j, k_j, \mathbf{T}_j, \mathbf{A}_I$  (여기서  $j \in \{1, \dots, e\}$ ,  $I \in \{0, \dots, N-1\}$ )의 결정에 의해 식 (3)의 입력 데이터를 근사하는 B-스플라인 하이퍼볼륨을 생성한다. 각 요소의 결정 알고리즘은 다음과 같다.

○  $e$ 와  $s$ 의 결정

식 (3)의  $M$ 개의 입력 데이터로부터  $e$ 와  $s$ 가 자동 결정된다.

○  $n_j$ 의 결정

조정점 개수  $N$ 을  $N = M/\alpha$ 에 의해 결정하고 식 (4)와 같이  $N$ 을 각 방향으로 균등하게 분배한다.

$$n_1 = \dots = n_e = \text{int}(\sqrt{M/\alpha}) \quad (4)$$

여기서  $\alpha$ 는 조정점 개수를 결정하는 파라미터로서 본 연구 예제에서  $\alpha = 2, \dots, 20$ 을 사용하였고, 조정점 개수와의 상관 관계는 4.2절에서 기술된다.

○  $k_j$ 의 결정

식 (4)에서 구한  $n_1$ 을 기준으로 식 (5)와 같이 결정한다.

$$\begin{aligned} \text{If } n_1 < 4, \text{ set } k_1 = \dots = k_e = n_1 \\ \text{Else set } k_1 = \dots = k_e = 4 \text{ (or cubic)} \end{aligned} \quad (5)$$

○  $\mathbf{T}_j$ 의 결정

3.1절에서 소개하는 절점벡터 결정 알고리즘에 의해 계산된다.

○  $\mathbf{A}_j$ 의 결정

3.2절에서 소개하는 조정점 결정 알고리즘에 의해 계산된다.

### 3.1 절점 벡터의 결정

절점 벡터  $\mathbf{T}_j$ 는 식 (3)의 입력 데이터  $\mathbf{D}$ 에서 데이터 위치(site)를 가리키는  $\{\mathbf{x}_i\} = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(e)}\}$  ( $i=0, \dots, M-1$ )와 식 (4), (5)에서 결정된  $n_j$ 와  $k_j$ 을 가지고 Fig. 1의 알고리즘<sup>[11,12]</sup>에 의해 계산된다.

알고리즘 서술 부분의 1) 첫 번째 For-루프는 입력 데이터가 존재하는 공간 영역의 최소 위치와 최대 위치의 좌표 값을 찾는다. 2) 두 번째 For-루프는 그 내부에 세 개의 For-루프를 가지고 있는데 절점벡터의 앞부분, 중간부분, 뒷부분을 각각 계산한다.

<p><b>PROBLEM</b> Find the knot vectors <math>\{\mathbf{T}\}</math> of <math>\mathbf{A}(\mathbf{u})</math> where <math>\{\mathbf{T}_j\} = \{\mathbf{T}_j   j = 1, \dots, e\}</math> and <math>e</math> denotes the space dimension.</p> <p><b>INPUT</b> 1) <math>\{\mathbf{x}_i\} = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(e)}\}</math> (<math>i=0, \dots, M-1</math>) 2) <math>n_j</math> and <math>k_j</math> (<math>j = 1, \dots, e</math>)</p> <p><b>OUTPUT</b> <math>\mathbf{T}_j = \{t_i^{(j)}\}_{i=0}^{n_j+k_j-1}</math> (<math>j = 1, \dots, e</math>)</p> <p><b>ALGORITHM</b> <b>For each</b> <math>j = 1, \dots, e</math>, <b>do</b> {   Find <math>x_{\min}^{(j)}</math> and <math>x_{\max}^{(j)}</math>   where <math>x_{\min}^{(j)} \leq x_i^{(j)} \leq x_{\max}^{(j)}</math> (<math>i = 0, \dots, M-1</math>) } <b>For each</b> <math>j = 1, \dots, e</math>, <b>do</b> {   <b>For each</b> <math>i = 0, \dots, k_j-1</math>, <b>do</b> {     <math>t_0^{(j)} = \dots = t_{k_j-1}^{(j)} = x_{\min}^{(j)}</math>   }   <b>For each</b> <math>i = k_j, \dots, n_j-1</math>, <b>do</b> {     <math>d = (M-1) / (n_j - k_j + 1)</math>     <math>l = \text{int}((k_j-1+i) \times d)</math>     <math>\alpha = (k_j-1+i) \times d - l</math>     <math>t_i^{(j)} = (1-\alpha) \cdot x_l^{(j)} + \alpha \cdot x_{l+1}^{(j)}</math>   }   <b>For each</b> <math>i = n_j, \dots, n_j+k_j-1</math>, <b>do</b> {     <math>t_{n_j}^{(j)} = \dots = t_{n_j+k_j-1}^{(j)} = x_{\max}^{(j)}</math>   } }</p>
--

Fig. 1 An algorithm for determining the knot vectors

### 3.2 조정점의 결정

조정점  $\mathbf{A}_j$ 는 다음의 식 (6)과 같은 최소자승 최소화 문제의 해를 구함으로써 계산된다.

$$\text{Minimize}_{\mathbf{A}_I} \sum_{i=0}^{M-1} \left\| \mathbf{f}_i - \sum_{J=0}^{N-1} \mathbf{A}_J \mathbf{N}_J(\mathbf{x}_i) \right\|^2 \quad (6)$$

여기서 식 (6)은 선형대수 방정식 형태인 식 (7)과 같이 다시 작성할 수 있다.

$$\text{Minimize}_{\{\mathbf{A}\}} \|\{\mathbf{f}\} - [\mathbf{N}]\{\mathbf{A}\}\|^2 \quad (7)$$

여기서

$$\{\mathbf{f}\} = \{\mathbf{f}_0, \dots, \mathbf{f}_{M-1}\}^T \in \mathbf{R}^{M \times s}$$

$$\{\mathbf{N}\} = [(\mathbf{N}_I(\mathbf{x}_i))_{i,I}] \in \mathbf{R}^{M \times N}$$

$$\{\mathbf{A}\} = \{\mathbf{A}_0, \dots, \mathbf{A}_{N-1}\}^T \in \mathbf{R}^{N \times s}$$

<p><b>PROBLEM</b> Find the control points <math>\{\mathbf{A}\}</math> of <math>\mathbf{A}(\mathbf{u})</math> where <math>\{\mathbf{A}_i   i = 0, \dots, N-1\}</math> and <math>N</math> denotes the total number of control points.</p> <p><b>INPUT</b> 1) <math>\{\mathbf{D}_i\} = \{(\mathbf{x}_i, \mathbf{f}_i)   \mathbf{x}_i \in \mathbf{E}^s, \mathbf{f}_i \in \mathbf{R}^s\} (i = 0, \dots, M-1)</math> 2) <math>n_j</math> and <math>k_j (j = 1, \dots, e)</math></p> <p><b>OUTPUT</b> <math>\{\mathbf{A}\} = \{\mathbf{A}_0, \dots, \mathbf{A}_{N-1}\}^T \in \mathbf{R}^{N \times s}</math></p> <p><b>ALGORITHM</b> Build <math>[\mathbf{N}] = [(\mathbf{N}_i(\mathbf{x}_j))_{i,j}] \in \mathbf{R}^{M \times N}</math> Build <math>\{\mathbf{f}\} = \{\mathbf{f}_0, \dots, \mathbf{f}_{M-1}\}^T \in \mathbf{R}^{M \times s}</math></p> <p>If <math>\text{rank}[\mathbf{N}] = \min(N, M)</math> Then If <math>M \geq N</math>, then compute <math>\{\mathbf{A}\} = ([\mathbf{N}]^T [\mathbf{N}])^{-1} [\mathbf{N}]^T \{\mathbf{f}\}</math> If <math>M &lt; N</math>, then compute <math>\{\mathbf{A}\} = [\mathbf{N}]^T ([\mathbf{N}] [\mathbf{N}]^T)^{-1} \{\mathbf{f}\}</math></p> <p>If <math>\text{rank}[\mathbf{N}] &lt; \min(N, M)</math> Then compute <math>\{\mathbf{A}\}</math> by using <i>Pseudo-inverse</i> technique and <i>singular value decomposition</i> algorithm.</p>
--

Fig. 2 An algorithm for determining the control points

식 (7)의 미지수  $\{\mathbf{A}\}$ 는 식 (2)의 조정점  $\mathbf{A}_i$ 로서 다음의 3가지 문제, 즉 과결정 문제, 미결정 문제, 불량조건 문제로 분류하여 결정한다. 각 문제의 해는 앞으로 소개될 방법에 의해 계산되며, 결국 Fig. 2의 조정점 결정 알고리즘에 의해 식 (7)의 최소화  $\{\mathbf{A}\}$ 를 얻는다.

#### ○ 과결정 문제의 최소화

식 (7)에서 행렬  $[\mathbf{N}]$ 의 계수(rank)가  $\text{rank}[\mathbf{N}] = N$ 이고 동시에  $M \geq N$ 일 때 이를 과결정 문제라 한다. 과결정 문제는 유일 해가 존재하며 이를 최소화 해라 부른다. 식 (7)의 목적함수가 양수이고 2차이므로 목적함수가 최소가 될 필요충분 조건은 식 (8)과 같이 편미분 식이 0일 때이다.

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{A}} \|\{\mathbf{f}\} - [\mathbf{N}]\{\mathbf{A}\}\|^2 \\ &= 2 \frac{\partial}{\partial \mathbf{A}} (\{\mathbf{f}\} - [\mathbf{N}]\{\mathbf{A}\}) \cdot (\{\mathbf{f}\} - [\mathbf{N}]\{\mathbf{A}\}) \\ &= 2[\mathbf{N}]^T (\{\mathbf{f}\} - [\mathbf{N}]\{\mathbf{A}\}) = 0 \end{aligned} \quad (8)$$

식 (8)의 미지수  $\{\mathbf{A}\}$ 에 관해 정리하면 식 (9)을

구할 수 있고 이로부터  $N$ 개의 조정점을 계산한다.

$$\{\mathbf{A}\} = ([\mathbf{N}]^T [\mathbf{N}])^{-1} [\mathbf{N}]^T \{\mathbf{f}\} \quad (9)$$

#### ○ 미결정 문제의 최소화

식 (7)에서 행렬  $[\mathbf{N}]$ 의 계수(rank)가  $\text{rank}[\mathbf{N}] = M$ 이고 동시에  $M < N$ 일 때 이를 미결정 문제라 한다. 미결정 문제는 목적함수 값이 0이 되게 하는 최소 해가 무한 개 존재하는데 이것들 중에서 유클리드 노름(Euclidean norm)이 최소인 해를 구한다. 이 해는 유일 해로서 최소노름 해라 부른다. 수치적으로 Pseudo-inverse 기법을 적용하면 식 (10)을 구할 수 있으며 이로부터  $N$ 개의 조정점을 계산한다.

$$\{\mathbf{A}\} = [\mathbf{N}]^T ([\mathbf{N}] [\mathbf{N}]^T)^{-1} \{\mathbf{f}\} \quad (10)$$

위에서 소개한 과결정 문제와 미결정 문제는 모두 행렬  $[\mathbf{N}]$ 이 완전계수(full rank)일 때의 경우로서 수치적으로 양호조건(well-conditioned)에 해당하는 문제이다. 경험적으로  $M \gg N$ 인 경우 또는  $M \ll N$ 인 경우에 양호조건 문제가 된다. 그러나 일반적인 경우에 행렬  $[\mathbf{N}]$ 의 계수는 부족하다.

#### ○ 불량조건 문제의 최소화

행렬  $[\mathbf{N}]$ 의 계수가 부족할 경우, 즉  $\text{rank}[\mathbf{N}] < \min(M, N)$ 일 경우, 일반적으로 식 (7)의 목적함수 값과 미지수의 유클리드 노름이 동시에 최소인 해를 찾는데 이를 불량조건(ill-conditioned) 문제라 부른다. 본 연구에서는 SVD(singular value decomposition) 알고리즘과 Pseudo-inverse 기법을 사용하여 수치 해를 계산한다.

대개 불량조건은 행렬  $[\mathbf{N}]$ 의 대각선 주변의 요소 크기가 비대각선 요소의 크기에 비해 작을 때 흔히 나타나는데, 본 연구의 경우에  $M \approx N$ 일 때 혹은 입력 데이터  $\{\mathbf{x}_i\}$ 가 비균일하게 분포되어 있을 때(예: 데이터 공간 내부의 특정 영역에  $\{\mathbf{x}_i\}$ 가 집중) 자주 나타난다.

## 4. 유계오차 근사기법

### 4.1 유계오차 B-스플라인 근사기법

본 연구에서 해결하고자 하는 유계오차 데이터 근사화 문제는 식 (11)과 같이 기술할 수 있다. 즉

식 (3)의 무작위 데이터  $\mathbf{D} = \{\mathbf{D}_i | i = 0, \dots, M-1\}$ 가 주어졌을 때, 각 데이터 위치  $\mathbf{x}_i$ 에서의 데이터 값  $\mathbf{f}_i = \mathbf{f}(\mathbf{x}_i)$ 과 그 위치에서의 식 (2)의 함수 값  $\mathbf{A}(\mathbf{x}_i)$  사이의 유클리드 거리(오차)  $\delta_i = |\mathbf{f}(\mathbf{x}_i) - \mathbf{A}(\mathbf{x}_i)|$ 가 모든  $i$ 에 대해 주어진 허용오차  $\delta_{TOL}$  내에 존재하도록  $\mathbf{A}(\mathbf{u})$ 을 결정하는 데이터 근사화 문제이다.

$$|\mathbf{f}(\mathbf{x}_i) - \mathbf{A}(\mathbf{x}_i)| < \delta_{TOL} \quad (i=0, \dots, M-1)$$

$$\text{(or } |\mathbf{f}(\mathbf{x}_i) - \mathbf{A}(\mathbf{x}_i)| < \varepsilon_{TOL} \times |\mathbf{f}_{\min} - \mathbf{f}_{\max}| \text{)} \quad (11)$$

여기서  $\delta_{TOL}$ 는 절대오차 개념의 주어진 허용오차를 나타내며,  $\varepsilon_{TOL}$ 는 데이터 값의 범위  $|\mathbf{f}_{\min} - \mathbf{f}_{\max}|$ 을 기준으로 정의한 상대오차 개념의 허용오차를 나타낸다.

본 연구에서 제시하는 유계오차 B-스플라인 근사법은 아래와 같이 4가지 과정에 의해 간략히 요약된다.

- Step (1): 초기 근사모델의 생성
- Step (2): 최대오차 발생 부위에 절점 삽입
- Step (3): 최대오차 감소를 위해 조정점 수정
- Step (4): 최대오차가 허용오차보다 작을 때까지 Step 2)~4)의 반복 수행

각 과정의 상세한 내용을 살펴보면 다음과 같다.

○ Step (1): 3절에서 기술한 최소자승 근사기법을 사용하여 초기 근사모델  $\mathbf{A}(\mathbf{u})$ 를 생성한다.

○ Step (2): 오차를 감소시키기 위해 절점을 삽입한다. 절점이 삽입되면 조정점 개수가 증가하고 그 만큼  $\mathbf{A}(\mathbf{u})$ 의 표현 능력이 향상되므로 오차를 감소시킬 수 있다. 절점 삽입 위치는 Fig. 3과 같이 최대오차 발생 지점인  $\xi_j$ 가 절점 벡터  $\mathbf{T}_j$ 에서

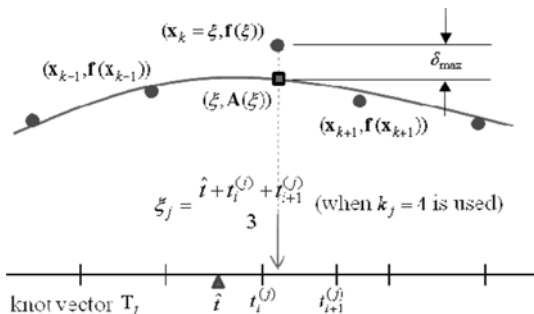


Fig. 3 Knot insertion for the maximal error's data site  $\xi_j \in [t_i^{(j)}, t_{i+1}^{(j)}]$  to be Greville abscissa. The symbol  $\hat{t}$  denotes the inserted knot

새로운 절점 노드(Greville abscissa)<sup>[13]</sup>가 되도록 결정한다. 여기서  $j$ 는  $j = 1, \dots, e$ 로서 각 방향의 절점 벡터에서 절점 삽입이 수행된다. Fig. 3에서  $k_j = 4$ 일 때 삽입할 절점  $\hat{t}$ 는  $\hat{t} = 3\xi_j - (t_i^{(j)} + t_{i+1}^{(j)})$  이고, 절점 삽입 후 절점 벡터  $\mathbf{T}_j$ 는 아래의 세 가지 경우 중에 하나가 된다.

$$\mathbf{T}_j = \{\dots, \hat{t}, t_i^{(j)}, t_{i+1}^{(j)}, \dots\}$$

$$\mathbf{T}_j = \{\dots, t_i^{(j)}, \hat{t}, t_{i+1}^{(j)}, \dots\} \quad (12)$$

$$\mathbf{T}_j = \{\dots, t_i^{(j)}, t_{i+1}^{(j)}, \hat{t}, \dots\}$$

만약  $\hat{t}$ 가  $\hat{t} = t_i^{(j)}$  또는  $\hat{t} = t_{i+1}^{(j)}$  라면 절점 삽입 위치에 이미 절점이 존재하므로 절점 삽입은 필요 없다. 참고로 절점 노드란 조정점 근방에 존재하는 매개변수로서 조정점  $\mathbf{A}$ 의 절점 노드  $\xi_j^{A_i}$ 는 식 (13)과 같이  $k_j - 1$ 개의 절점들을 평균하여 구한다.

$$\xi_j^{A_i} = \frac{t_{i+1}^{(j)} + \dots + t_{i+k_j-1}^{(j)}}{k_j - 1} \quad (13)$$

○ Step (3): 각 절점 벡터에서 절점 삽입 후, 최대오차  $\delta_{\max} = |\mathbf{f}(\xi) - \mathbf{A}(\xi)|$ 를 감소시키기 위해 절점 노드  $\xi$ 에 해당하는 조정점  $\mathbf{A}_m$ 를  $\Delta \mathbf{A}_m$ 만큼 변경한다. 여기서 조정점 인덱스  $m$ 과  $\Delta \mathbf{A}_m$ 은 각각 식 (14)와 (15)로부터 구한다.

$$m = m_1 + (n_1)m_2 + (n_1n_2)m_3 + \dots + (n_1 \dots n_{e-1})m_e \quad (14)$$

$$\Delta \mathbf{A}_m = \frac{\mathbf{f}(\xi) - \mathbf{A}(\xi)}{\mathbf{N}_m^k(\xi)} = \frac{\mathbf{f}(\xi) - \mathbf{A}(\xi)}{N_{m_1, T_1}^{k_1}(\xi_1) \times \dots \times N_{m_e, T_e}^{k_e}(\xi_e)} \quad (15)$$

○ Step (4): 수정된  $\mathbf{A}(\mathbf{u})$ 에서 다시 최대오차  $\delta_{\max}$ 을 찾고 허용오차  $\delta_{TOL}$ 와 비교한다. 만약  $\delta_{\max} < \delta_{TOL}$  라면 모든 근사법 과정은 종료되고, 그렇지 않으면 반복 수행을 위해 Step (2)로 간다.

이상의 유계오차 B-스플라인 근사법은 Fig. 4와 같은 반복순환 알고리즘에 의해 구현된다. 위에서 설명한 4가지 과정의 알고리즘 내의 해당 위치는 다음과 같다. Step (1)은 **Construct A(u)**, Step (2)는 **Repeat** 루프 내의 **For** 루프, Step (3)은 **Compute & Set**, 그리고 Step (4)는 **Repeat** 루프

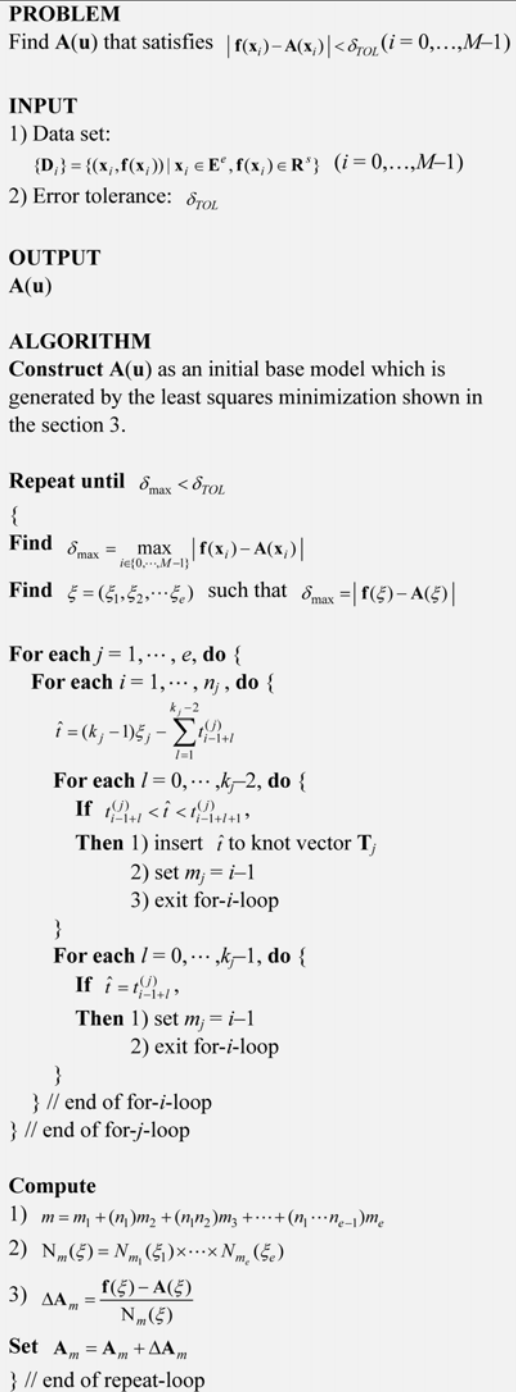


Fig. 4 An algorithm for error-bounded approximation

내의 종료조건  $\delta_{\max} < \delta_{TOL}$  및 **Find**문에 해당된다.

#### 4.2 제시 알고리즘의 성능 분석

본 연구에서 제시하는 유계오차 근사법은 최소

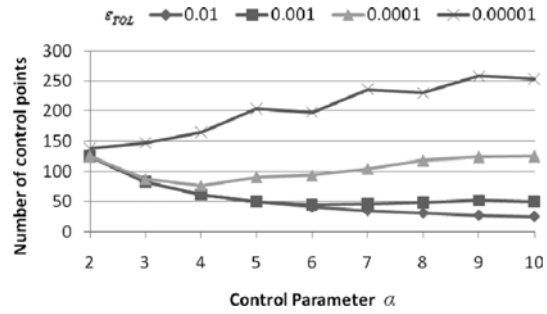


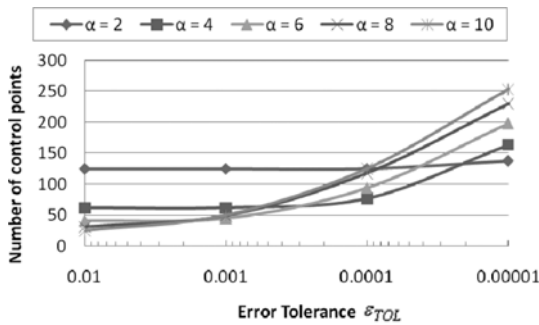
Fig. 5 Control parameter  $\alpha$  affecting the number of control points

자승 근사기법<sup>[10]</sup>에 의해 생성한 초기 근사모델을 기반으로 절점 삽입 및 조정점 수정의 반복 과정을 통하여 주어진 허용오차를 만족하는 최종의 근사모델을 얻는다. 여기서 최종 근사모델의 조정점 개수는 본 연구 알고리즘의 성능을 측정할 수 있는 주요 요소로서 초기 근사모델의 조정점 개수와 관련이 많다. 즉 초기 근사모델의 조정점 개수를 결정할 때 사용하는 식 (4)의  $\alpha$  값 설정에 의해 최종 근사모델의 조정점 개수  $N$ 이 결정된다. 적절한  $\alpha$  값의 선택은 허용오차를 만족하는 효율적인 최종 근사모델의 생성을 위하여 중요하며 본 연구 알고리즘의 성능에도 영향을 준다. Fig. 5는 5.1절의 적용 예제인 face모델에 관하여 주어진 허용오차 별로  $\alpha$  값 변화 ( $\alpha = 2 \sim 10$ )에 따른 최종 근사모델의 조정점 개수  $N$ 의 변화를 보여주고 있고,

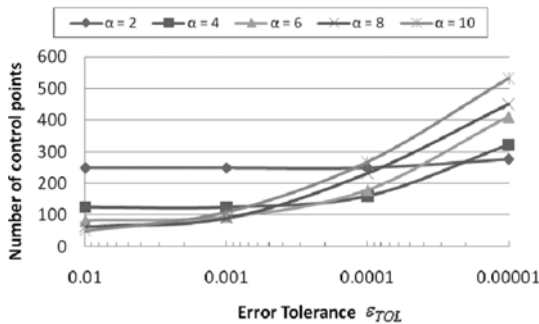
Fig. 6(a)는 같은 모델에 대하여 각  $\alpha$  값 별로 허용오차 변화에 따른  $N$ 의 변화를 보여주고 있다. 이 그림으로부터  $\alpha$  값과  $N$  사이의 상관 관계를 분석해 보면 다음과 같다.

1)  $\alpha$  값이 작을수록 초기모델의  $N$ 은 커지며 그만큼 표현능력이 커져서 절점 삽입 없이도 허용오차를 만족시켜 줄 가능성이 커진다. 여기서 허용오차가 클수록 작은  $N$ 으로도 오차를 만족시켜 줄 수 있는데 큰  $N$ 을 쓸 가능성이 많아져서 메모리 낭비가 가능하다. 반대로 허용오차가 작을수록  $N$ 의 활용도가 높아지며 절점 삽입 없이 초기모델만 가지고 오차를 만족시켜 줄 가능성이 높아진다.

2)  $\alpha$  값이 클수록 초기모델의  $N$ 은 작아지며 표현능력이 작아져서 그만큼 절점 삽입을 해야 허용오차를 만족시켜 줄 가능성이 커진다. 절점 삽입을 통해 생성된 조정점으로 최대오차 발생 부위의 근사오차 값은 줄일 수 있으나 허용오차보다 큰 오차를 갖는 여타 데이터 점의 오차는 줄이지



(a) face model



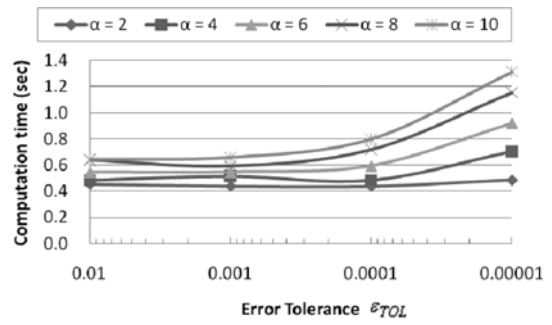
(b) hand model

Fig. 6 Number of control points required for the given error tolerance: (a)  $M = 251$ ; (b)  $M = 501$

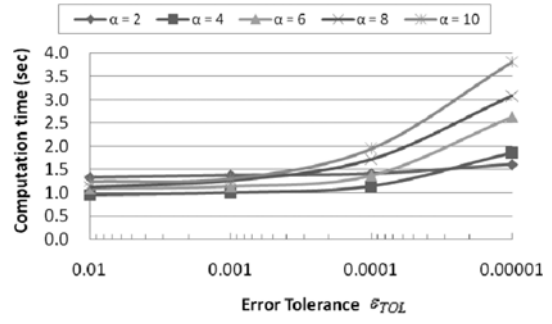
못하기 때문에 이들의 오차를 줄이기 위해선 최악의 경우 이들의 개수 규모만큼 절점을 삽입해야 한다. 여기서 허용오차가 작을수록 이들 개수가 증가하므로 최종 조정점의 개수  $N$ 이 증가한다. 반대로 허용오차가 클수록 작은  $N$ 만으로도 충분히 오차를 만족시켜 줄 가능성이 높아진다.

3) 결국  $\alpha$  값은 주어진 허용오차의 크기에 의존하여 결정되어야 한다. 즉 허용오차가 크면  $\alpha$  값을 크게, 허용오차가 작으면  $\alpha$  값을 작게 설정해 주어야 한다. Fig. 6에서 허용오차가  $\epsilon_{TOL} = 0.01 \sim 0.001$ 일 때 허용오차가 크기 때문에  $N$ 이  $\alpha$  값에 둔감하여 크게 증가하지 않으나, 반대로 허용오차  $\epsilon_{TOL} = 0.001 \sim 0.00001$ 일 때 허용오차가 작으므로  $N$ 은  $\alpha$  값에 민감하여  $\alpha$  값이 클수록 크게 증가한다.

한편, 본 연구 알고리즘의 계산 성능을 메모리 사용량 측면과 계산 시간 측면에서 살펴보면 다음과 같다. 메모리 사용량은 최종 모델이 요구하는 메모리 크기로 2.2절의 메모리 크기 계산 식에 의해 측정할 수 있는데 가장 큰 부분을 차지하고 있



(a) face model



(b) hand model

Fig. 7 Computation time (seconds) required for the given error tolerance: (a)  $M = 251$ ; (b)  $M = 501$

는 조정점 개수  $N$ 에 의해 그 크기를 간접적으로 확인할 수 있다. Fig. 6은 5.1절의 적용 예제인 face 모델과 hand 모델에 관해 주어진 허용오차가 0.00001 ~ 0.01일 때 사용된 조정점 개수  $N$ 을 보여주고 있다. Fig. 6과 같이 허용오차가 작을수록  $N$  값이 기하 급수적으로 증가함을 알 수 있다.

계산 시간은 초기 근사모델 및 반복 수행에 의한 최종 근사모델 생성 시까지의 총 소요 시간을 초 단위로 측정된 것으로 Fig. 6의 두 모델에 관하여 측정하였고 그 결과는 Fig. 7과 같다. 허용오차가 작을수록 요구되는 계산 시간은 증가하며 또한  $\alpha$ 가 클수록 계산 시간이 증가함을 확인할 수 있다. 참고로 본 연구의 모든 계산 작업은 3.06GHz Intel Xeon CPU에서 수행되었다.

### 5. 적용예제 및 토의

본 연구에서 개발한 유계오차 근사화 기법의 적합성, 유용성 등을 확인하고, 다분야로의 확장 가능성 등을 보이기 위해, 본 연구 방식을 다음의 3가지 예제 문제에 적용해 보았다.



- 1D 문제: 곡선 접합 문제( $e = 1, s = 2$ 인 경우)
- 2D 문제: 2차원 수학 함수의 근사 문제( $e = 2, s = 1$ 인 경우)
- 3D 문제: 폴리곤 곡면의 재건 문제( $e = 3, s = 1$ 인 경우)

**5.1 곡선 접합(curve fitting)**

본 연구에서 다루는 곡선 접합 문제의 입력 데이터는 아래와 같다.

$$D = \{ (u_i, x_i, y_i) \mid u_i, x_i, y_i \in \mathbf{R}^1, i = 0, \dots, M-1 \}$$

여기서  $u_i$ 는 독립변수로서 데이터 위치를 나타내고,  $(x_i, y_i)$ 는 종속변수로서 데이터 값을 의미한다. 평면 상에 존재하는 B-스플라인 곡선의 경우와 같다.

첫 번째 예제는 Fig. 8과 같이 face 모델로서 입력 데이터 개수가  $M = 251$ 이고, B-스플라인 근사

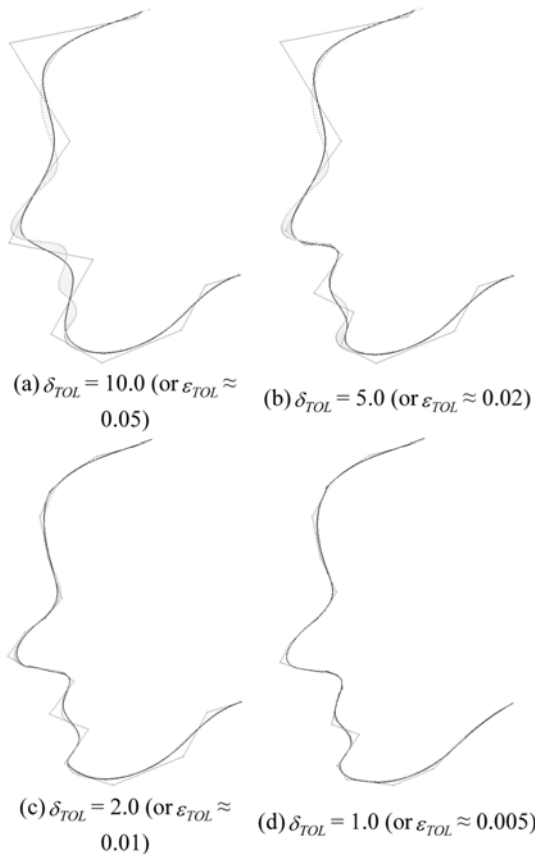
모델 생성을 위해  $k = 4, \alpha = 20$ 을 사용하였다. Fig. 8에서 빨간색 점은 입력 데이터 점이고, 검은색 곡선은 최종 근사모델이며, 회색 폴리라인은 근사모델의 조정점 다각형이다. 그리고 빨간 점과 검은색 곡선 사이의 녹색 선분이 근사오차를 가시화한 결과이다.

Fig. 8에서 허용오차가 작을수록 조정점 개수  $N$ 은 증가하며 그만큼 근사오차를 표시하는 녹색 선분이 점차 사라짐을 확인할 수 있다.

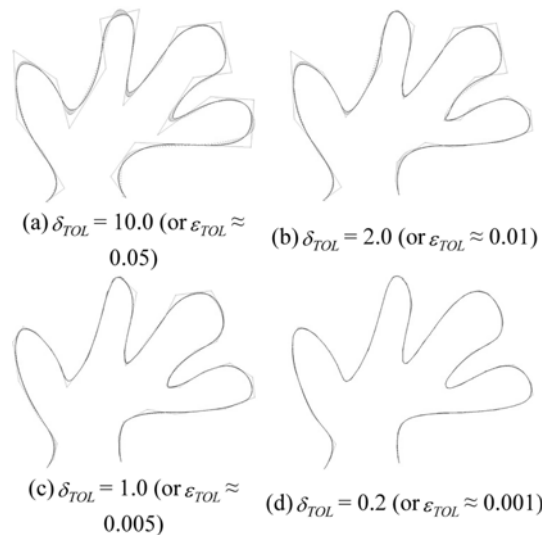
두 번째 예제는 Fig. 9에서 보는 바와 같이 hand 모델로서  $M = 501$ 이고,  $k = 4, \alpha = 20$ 을 사용하였다. 첫 번째 예제와 유사하게 허용오차가 작아질수록 근사오차의 크기를 보여주는 녹색 선분이 점차 사라짐을 확인할 수 있다.

**5.2 2차원 수학 함수의 근사**

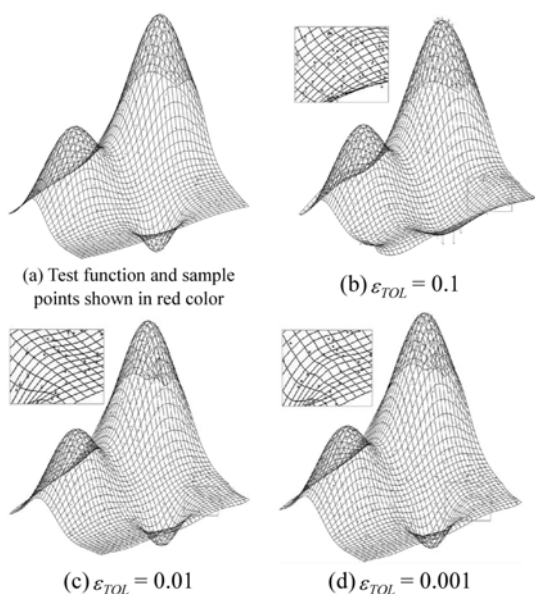
2차원 유클리드 공간 상에 산만하게 분포된  $M$ 개의 데이터 샘플 점  $(x_i, y_i)$ 와 그 위치에서의 데이터 값  $f(x_i, y_i)$ 이 주어졌을 때, 이것을 본 연구 방식에 의해 근사화 하였을 때 얻은 결과를 사용한 허용오차와 함께 살펴보면 Fig. 10과 같다. 여기서 데이터 값은 아래와 같은 Franke<sup>[2]</sup>의 2차원 테스트 함수를 사용하였다. 이 함수는 분산(scattered) 데이터 모델링<sup>[2]</sup> 분야에서 테스트 함수로서 자주 언급되는 함수 중에 하나이다.



**Fig. 8** A curve fitting example of a face model ( $k = 4, M = 251$  used): (a)  $N = 12$ ; (b)  $N = 16$ ; (c)  $N = 26$ ; (d)  $N = 36$



**Fig. 9** A curve fitting example of a hand model ( $k = 4, M = 501$  used): (a)  $N = 25$ ; (b)  $N = 46$ ; (c)  $N = 59$ ; (d)  $N = 122$



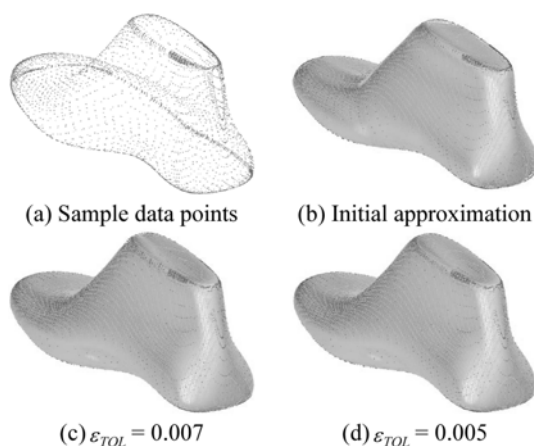
**Fig. 10** A fitting example of a Franke test function where the function range is  $[0.0, 1.3]$  ( $k_1 = k_2 = 4$  used): (a)  $M = 500$ ; (b)  $N = 7 \times 7$ ; (c)  $N = 16 \times 16$ ; (d)  $N = 25 \times 25$

$$f(x,y) = 0.75 \exp\left[-\frac{(9x-2)^2 + (9y-2)^2}{4}\right] + 0.75 \exp\left[-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}\right] + 0.5 \exp\left[-\frac{(9x-7)^2 + (9y-3)^2}{4}\right] - 0.2 \exp[-(9x-4)^2 - (9y-7)^2]$$

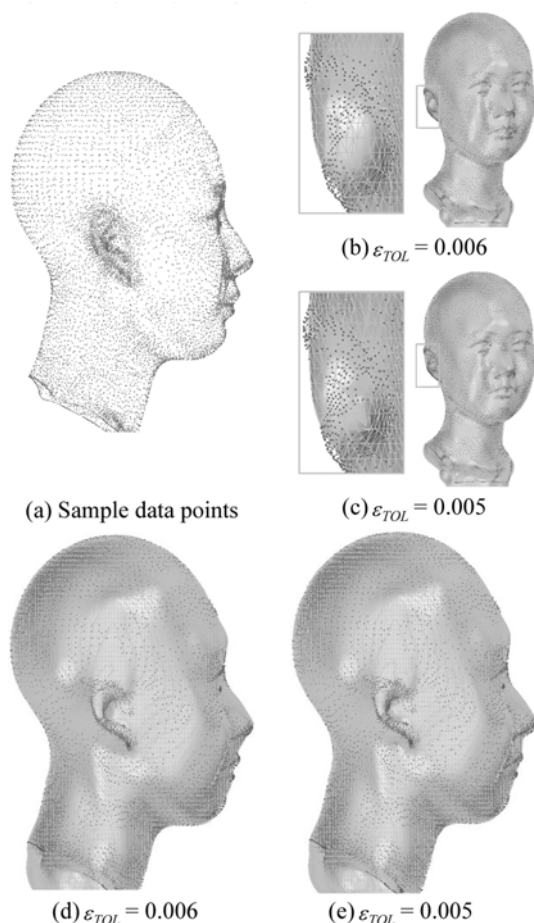
Fig. 10에서  $M = 500$ 이고 B-스플라인 근사모델 생성을 위해  $k_1 = 4, k_2 = 4$ 을 사용하였다. Fig. 10 (a)는 테스트 함수와 입력 데이터를 보여주고 있고, (b)는  $\epsilon_{TOL} = 0.1, \alpha = 10$ 을 사용했을 때의 근사 결과를, (c)는  $\epsilon_{TOL} = 0.01, \alpha = 5$ 을 사용했을 때, (d)는  $\epsilon_{TOL} = 0.001, \alpha = 2$ 을 사용했을 때의 결과를 보여주고 있다. 각 결과 그림에서 녹색 선분은 근사오차의 크기를 가시화한 결과인데 허용오차가 작아질수록 조정점 개수는 증가하나 그만큼 근사오차를 의미하는 녹색 선분의 크기가 작아짐을 확인할 수 있다.

**5.3 폴리곤 곡면의 재건(surface reconstruction)**

3차원 스캐너로부터 획득한 점 군 데이터로부터 삼각형 메쉬 형태의 폴리곤 곡면을 재건하는 데



**Fig. 11** Surface reconstruction of a shoelast model ( $k_1 = k_2 = k_3 = 4$  used): (a)  $M = 5,318$ ; (b)  $N = 9 \times 9 \times 9$ ; (c)  $N = 12 \times 12 \times 12$ ; (d)  $N = 22 \times 22 \times 22$



**Fig. 12** Surface reconstruction of a head model ( $k_1 = k_2 = k_3 = 5$  used): (a)  $M = 11,486$ ; (b) & (d)  $N = 20 \times 20 \times 20$ ; (c) & (e)  $N = 41 \times 41 \times 41$

본 연구 방식이 활용될 수 있음을 Fig. 11과 12의 적용 예제를 통하여 확인할 수 있다. 먼저 본 연구에서 수행한 곡면 재건 절차를 간략히 기술하면 아래와 같다.

○ 폴리곤 곡면 재건 절차

1) 측정된 각 점에서의 데이터 값을 모두 특정한 상수 값으로 설정하여 아래의 입력 데이터를 구성한다.

$$\mathbf{D} = \{(x_i, y_i, z_i, f_i = C) \mid i = 0, \dots, M-1\}$$

여기서  $(x_i, y_i, z_i)$ 는 독립변수로서 측정 점의 좌표 값이고,  $f_i = C$ 는 종속변수로서 데이터 값을 나타낸다. 여기서  $C$ 는 특정한 상수 값을 의미한다.

2) 측정 점 근 내부 혹은 외부 영역에 일정 개수의 점을 추가하고 그 데이터 값이  $f_i = C + dist_i$ 가 되도록 설정한 후 입력 데이터에 추가한다. 여기서  $dist_i$ 는 추가 점이 내부 영역에 존재할 때 양의 값을, 외부 영역에 존재할 때 음의 값을 설정한다.

3) 본 연구 방식을 사용하여 입력 데이터로부터 B-스플라인 근사모형을 생성한다. 참고로 생성된 근사모형은 등위(level set) 곡면<sup>[14]</sup>의 성질을 갖는다.

4) Marching Cubes<sup>[15]</sup> 방법을 사용하여 생성된 근사모형에서 폴리곤 곡면을 추출한다.

첫 번째 적용 예제는 Fig. 11과 같이 구두골 (shoelast) 모델로서  $M = 5,318$ 이고  $k_1 = k_2 = k_3 = 4$ 를 사용하였다. Fig. 11(a)는 입력 데이터인 측정 점 근의 모습이고, (b)는 3절에서 소개한 최소자승 근사법에 의해 생성한 초기 근사모형의 결과이다. 그리고 (c)와 (d)는 (b)의 초기모형에 각각  $\epsilon_{TOL} = 0.007$ 과  $\epsilon_{TOL} = 0.005$ 를 적용하였을 때의 근사 결과를 보여주고 있다. 그림에서 녹색 선분은 근사오차의 크기를 의미하며 허용오차가 작아질수록 작아짐을 확인할 수 있다.

두 번째 적용 예제는 Fig. 12와 같이 두상 모델로서  $M = 11,486$ 이고  $k_1 = k_2 = k_3 = 5$ 를 사용하였다. Fig. 12(a)는 입력 데이터 점을, (b)와 (d)는  $\epsilon_{TOL} = 0.006$ 일 때의 근사 결과를, (c)와 (e)는  $\epsilon_{TOL} = 0.005$ 일 때의 근사 결과를 보여주고 있다. 그림에서 녹색 선분은 근사오차의 크기를 나타낸다. Fig.

12(b)와 (c)에서 귀 부분의 차이를 비교할 수 있고, (d)와 (e)에서 인종 부분의 차이를 비교할 수 있다. 즉 허용오차가 작을수록 근사모형이 입력 데이터에 근접하고 있음을 확인할 수 있다.

## 6. 결 론

본 연구에서는 무작위로 산만하게 분포된  $M$ 개의 입력 데이터를 주어진 허용오차 범위 내에서 근사화하는 유계오차 B-스플라인 근사법에 대해 소개하였다. 먼저 근사화 표현 모델로서 B-스플라인 하이퍼볼륨에 대해 소개하였고, 이에 기반을 둔 근사화 기법으로서 최적화 형태의 최소자승 근사 기법에 대해 기술하였다. 그리고 입력 데이터와 근사모형 간의 최대 오차가 주어진 허용오차를 넘지 못하도록 최소자승법에 의해 생성된 초기 근사모형을 수정하는 유계오차 B-스플라인 근사기법에 대해 설명하였다. 또한 적용 예제로서 곡선 접합 문제, 2차원 수학 함수의 근사 문제, 그리고 폴리곤 곡면 재건 문제에 대해 본 연구 유계오차 기법의 수행 결과를 제시하였다.

본 연구에서 제안하는 유계오차 B-스플라인 접근 방식의 주요 특징을 살펴보면 다음과 같이 요약된다.

○ 데이터 복잡성에 무관: 본 연구 유계오차 B-스플라인 근사법은 입력 데이터의 순서, 개수, 그리고 분포 형태에 제한을 받지 않는다.

○ 계산 복잡성 및 안정성 제공: 과결정 혹은 미결정 문제에 무관하며, 불량 조건의 문제라도 안정된 근사 결과를 제공한다.

○ 국부수정 가능: 기존 데이터가 일부 변경되거나 새로운 데이터가 추가되었을 때, 기존 근사모형의 재생성 없이, 해당 조정점의 변경 혹은 이것에 의해 해결되지 않을 경우 점점 삽입에 의한 조정점의 추가 및 수정 등을 통하여 효율적으로 처리할 수 있다.

○ 다양한 활용 분야로의 유용성: 본 연구 적용 예제가 비록 1차원~3차원의 경우만을 다루었지만 공간 차원 수( $e$ )에 무관하며 나아가 스칼라 장의 개수( $s$ )에도 무관하다. 또한  $e$ 개의 독립변수,  $s$ 개의 종속변수 각각은 적용 분야에 의존하여 그 물리적 의미가 다를 수 있지만 수학적 관점에서 제안 기법의 적용에 있어 특별한 제한이 없다. 즉 해당 분야에서 독립변수와 종속변수를 구별하여 사용한

다면 적용 분야의 특성에 무관하게 본 연구 기법을 활용할 수 있다.

한편 향후 연구 과제로서 아래의 문제 해결이 필요하다.

○ 최소의 조정점 개수: 부적절한  $\alpha$ 의 선정으로 과대한 조정점 개수를 갖는 근사모델이 생성될 수 있다. 경험에 의존하지 않는 적절한  $\alpha$  선정에 관한 추가 연구가 필요하다.

○ 근사 정밀도와 근사 완만성: 근사 정밀도와 완만성은 상호 배타적 모순 관계에 있다. 적절한 초기 근사모델의 생성을 통하여 근사 완만성을 달성할 수 있고, 본 연구 유계오차 방식에 의해 근사 정밀도를 달성할 수 있다. 그러나 완전한 초기 모델은 유계오차 방식에 의해 변경되어 최종 모델의 완만성을 보장받지 못한다. 결국 근사 완만성과 정밀도 사이의 충돌 문제를 해결하기 위해선 초기 모델과 최종 모델 간의 적절한 가중치 결합이 필요하다.

## 감사의 글

이 논문은 2012년도 한국교통대학교 교내학술연구비의 지원을 받아 수행한 연구임.

## 참고문헌

1. Haber, J., Zeilfelder, F., Davydov, O. and Seidel, H.-P., 2001, Smooth Approximation and Rendering of Large Scattered Data Sets, *12th IEEE Visualization 2001*, pp. 341-571.
2. Nielson, G.M., Scattered Data Modeling, 1993, *IEEE Computer Graphics and Applications*, 13(1), pp. 60-70.
3. Shepard, D., 1968, A Two Dimensional Interpolation Function for Irregularly Spaced Data, *Proceedings of ACM 23rd National Conference*, pp. 517-524.
4. Schaback, R., 1995, Multivariate Interpolation and Approximation by Translates of a Basis Function in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, World Scientific Publishing, Singapore, pp. 491-514.
5. Franke, R. and Nielson, G.M., 1980, Smooth Interpolation of Large Sets of Scattered Data, *International Journal of Numerical Methods in Engineering*, 15, pp. 1,691-1,704.

6. Wendland, H., 1995, Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree, *Advances in Computational Mathematics*, 4, pp. 389-396.
7. Hardy, R., 1971, Multiquadric Equations of Topography and Other Irregular Surfaces, *J. Geophysical Research*, 76(8), pp. 1,905-1,915.
8. Duchon, J., 1975, Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces in *Multivariate Approximation Theory*, Basel, Switzerland: Birkhauser, pp. 85-100.
9. Park, S., 2009, A Rational B-spline Hypervolume for Multidimensional Multivariate Modeling, *Journal of Mechanical Science and Technology*, 23, pp. 1967-1981.
10. Park, S., 2011, On B-spline Approximation for Representing Scattered Multivariate Data, *Transactions of the KSME A*, 35(8), pp. 921-931.
11. Piegl, L. and Tiller, W., 1995, *The NURBS Book*, Springer-Verlag.
12. De Boor, C., 1978, *A Practical Guide to Splines*, New York, Springer-Verlag.
13. Farin, G., 1990, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, San Diego.
14. Sethian, J.A., 1999, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, UK.
15. Lorensen, W.E. and Cline, H.E., 1987, Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer Graphics (Proceedings of SIGGRAPH 87)*, 21(4), pp. 163-169.



## 박 상 군

1991년 포항공과대학교 학사  
 1993년 서울대학교 기계설계학과 석사  
 1997년 서울대학교 기계설계학과 박사  
 1997년~1999년 삼성SDS 정보기술연구소 책임연구원  
 2000년 서울대 BK21 기계분야사업단 계약교수  
 2000년~2002년 (주)K&I 테크놀로지 책임연구원  
 2003년~현재 국립한국교통대학교 기계공학과 교수  
 관심분야: Computational Geometry, CAD/CAM/CAE, Design Optimization, Scientific Visualization