4 차원 산포된 자료 선형 보간의 가시화 - 자료 값을 고려한 사면체 분할법에 의한 -

이 건†

요 약

표면 보간법을 응용하는 분야에는 모델링 자연현상 가시화 등을 비롯하여 여러 가지를 들 수 있다. 사면체 분할법은 사차원적 표면 형성을 위한 전 처리 단계 중의 하나이다. 사차원 공간상에서 피스와이즈(piecewise) 선형 보간법의 질은 삼차원에서의 자료 점의 분포에 영향을 받을 뿐 아니라 자료 값에도 영향을 받는다. 자료 값을 고려한 사면체 분할법이 추정의 질을 개선시킬 수 있음을 사차원 공간의 가시화를 통하여 보여준다. 본 논문에서는 Delaunay사면체 분할법의 구 기준(Sphere criterion)과 자료 의존형 사면체 분할법 중의 하나인 최소 제곱 근사 기준(least squares fitting criterion)을 논의하였다. 본 논문은 또한 새로운 자료 값을 고려한 기준인 gradient difference와 jump in normal direction derivatives들을 논의하였다.

Visualization of 4-Dimensional Scattered Data Linear Interpolation Based on Data Dependent Tetrahedrization

Kun Lee†

ABSTRACT

The numerous applications of surface interpolation include the modeling and visualization of physical phenomena. A tetrahedrization is one of pre-processing steps for 4-D surface interpolation. The quality of a piecewise linear interpolation in 4-D space depends not only on the distribution of the data points in R^2, but also on the data values. We show that the quality of approximation can be improved by data dependent tetraheadrization through visualization of 4-D space. This paper discusses Delaunay tetrahedrization method(sphere criterion) and one of the data dependent tetrahedrization methods(least squares fitting criterion). This paper also discusses new data dependent criteria : 1) gradient difference, and 2) jump in normal direction derivatives.

1. Introduction

The numerous applications of surface interpolation include the modeling and visualization of physical phenomena[1][2][3][4]. Earlier work in these fields related more to bivariate interpolation than to trivariate interpolation due to the complexity and difficulty of 4-D space visualization [5].

In this paper, new data dependent criteria are introduced. We visualize the 4-dimensional scattered data interpolants based on new data dependent criteria.

† 정 회 원 : 전북산업대학교 컴퓨터공학과 전임강사
논문접수 : 1996년 4월 26일, 심사완료 : 1996년 8월 2일

2. Previous Work

Scattered data is defined as a collection of observed data that have no specified relations between data points. Denote scattered data (x_i, y_i, z_i, F_i), $(i=1, ..., N)$. $v_i=(x_i, y_i, z_i)$ is called a point. F_i is called a value at v_i, $(i=1, ..., N)$. The interpolation of Scattered data in $\mathbf{R}^3$ consists of constructing a function $f=f(x, y, z)$ such as $f(x_i, y_i, z_i)=F_i$, $(i=1, ..., N)$, where $\mathbf{V}=\{v_i=(x_i, y_i, z_i)\in\mathbf{R}^3, i=1, ..., N\}$ is a set of distinct and non-collinear data points and $\mathbf{F}=(F_1, ..., F_N)$ is a real vector corresponding to points. One of volumetric interpolations of the scattered data. Thus, one of main tasks is to provide feasible 3-D domain consisting of tetrahedra. Ω is the convex hull of tetrahedra which do not intersect each other and containing the set V of scattered data. This definition is stated in detail.

The vertices and the boundary $\partial\Omega$ of Ω are in V. A set $\mathrm{TH}=\{\mathrm{TH}_i\}_i^I$ of non-degenerate, open tetrahedra is a tetrahedrization of Ω if the following conditions hold:

a) V is the set of all vertices of tetrahedra in TH,
b) every edge of a tetrahedron in TH contains only two points from V,
c) every face of a tetrahedron in TH contains only three points from V. Now, T is the set of all triangles of tetrahedra in TH,
d) $\overline{\Omega}=\cup_{i=1}^{I}\overline{\mathrm{TH}_i}$, and
e) $\mathrm{TH}_i\cap\mathrm{TH}_j=\phi(i\neq j)$.

Before data dependent triangulation was introduced, long thin triangles were treated as "bad" triangles. In other words, the triangles should be as equiangular as possible, because a "majorizing" error bound contains the factor $1/\sin^n\theta$. The denominator of error factor becomes smaller when the minimum angle of given triangles is getting smaller[6]. However, long thin triangles can be "good" triangles for approximating the function which has a preferred direction. In this case, the long thin side of the triangles should line up in the direction of a small curvature of function[7].

One single criterion on the triangulation is not adequate for all types of data sets. It is more reasonable for the user to select the appropriate criterion according to the nature of data sets. We briefly review concepts related to 2-dimensional data dependent triangulation and 3-dimensional Delaunay tetrahedrization in this section. Basic concepts of computational geometry can be founded in several texts[8][9].

2.1 Two-Dimensional Data Dependent Triangulation

In many cases, we assume that the underlying functions are "smooth" related to the data sets except special cases. Many researchers try to track this behavior by using "nearly C^1" criteria which deal with the common edge between two adjacent triangles[7].

2.1.1 Nearly C^1 Criteria

a) Angle between normals(ABN)

This criterion minimizes the angle between two normal vectors of adjacent planes. The resulting triangulation is the one in which the normal vector to the piecewise surfaces (i.e. interpolants) exhibit the least changes in its direction when passing through the edges of the triangulation.

b) Jump in normal derivatives(JND)

The resulting triangulation is the one in which directional derivatives are the least changed in the normal direction of a common edge.

2.1.2 Min Error Criterion

Schumaker proposed least squares fitting method to find interpolants such as minimizing the errors between the actual function values and the approximated values at data points. One can use a simulated annealing algorithm for (globally) optimal triangulations [10].

$$E(\Delta)=\sum_{i=1}^{n}[z_i-s(x_i, y_i)]^2 \qquad (2.1)$$

where Δ is a given triangulation.

$E(\Delta)$ is an error between the exact function value and approximated value, n is the number of data points, $z_i = f(x_i, y_i)$ is an exact and $s(x_i, y_i)$ is an approximated value at data point p_i by a interpolation function s.

2.2 Three-Dimensional Tetrahedrization

After Delaunay provided the algorithm for triangulation of a 2-dimensional plane[11], several researchers have generalized the triangulation to n -dimensional space. Watson presents the n -dimensional Delaunay tessellation with application to Voronoi polytopes[12]. Edelsbrunner has presented an algorithm that constructs a tetrahedrization of a set P of n points by the incremental refinement[13].

Joe's method presented an algorithm that can construct the three-dimensional Delaunay triangulation using local transformations by starting with a special triangulation. In Joe's paper, the sphere criterion is used for 3-D Delaunay triangulation[14][15]. In other words, no more than 4 points lie inside the circumsphere of a tetrahedron. Sphere criterion is a data independent criterion to characterize 3-D Delaunay triangulation.

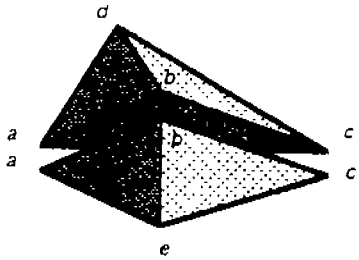
2.2.1 Joe's Algorithm

We express the algorithm describing the above processes as:

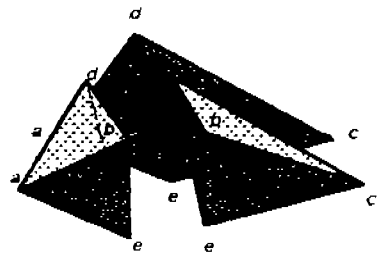
if *desirable* **then**
  if *possible* **then**
    find *case of transformation*
  endif
endif

2.2.2 Details of Transformable Cases

In 2-D triangulation, two triangles always transform into another two triangles by swapping a diagonal line in a quadrilateral. In 3-D tetrahedrization, two tetrahedra may become three tetrahedra. If *de* intersects the interior of triangle **abc**, then the initial tetrahedrization contains the two initial tetrahedrization contains the two tetrahedra **abcd** and **abce** (see Figure 2.1), and the transformed tetrahedra become three tetrahedra **abde**, **acde**, and **bcde** (see Figure 2.2). If **ac** does not intersect the interior of triangle **bde**, then one needs the third tetrahedron **acde**. Hence, these tetrahedra may be transformable.

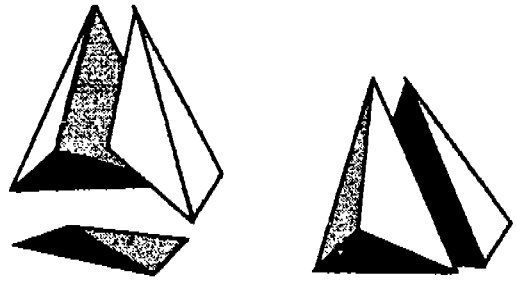


(Fig. 2.1) Horizontal transformation.
(그림 2.1) 수평적 변환

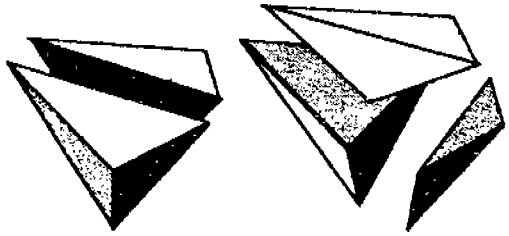


(Fig. 2.2) Vertical transformation.
(그림 2.2) 수직적 변환

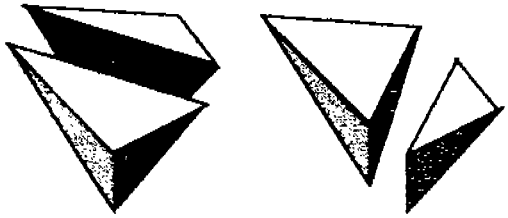
a) case 1



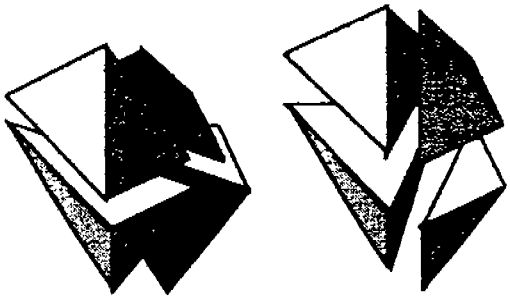
b) case 2



c) case 3



d) case 4



(Fig. 2.3) Four transformable cases.
(그림 2.3) 변환 가능한 4 가지 경우

There are 4 cases that are transformable. In the first two cases, any four of the five points are not coplanar. In the next two cases, four of five points are co-planar (see Figure 2.3.).

3. Data Dependent Tetrahedrization

We introduce several data dependent criteria of tetrahedrization. The quality of a piecewise linear interpolation in space depends not only on the distribution of the data points in $\mathbf{R}^3$, but also on the data values. We can improve the quality of an approximation by using data dependent tetrahedrization. In 2-D triangulation, normal vectors are one of the characteristics of the given surface. In 3-D tetrahedrization, gradient vectors more appropriately represents the given space. Least squares fitting method leads to feasible approximation, even if it requires the exact values which are normally not available.

3.1 Least Squares Fitting Criterion

Schumaker discusses the least squares fitting criterion for an optimal triangulation[10]. We apply least squares fitting criterion based on each tetrahedron. In our case, input data are a set of positional data points and a known test function. In a real world application, the exact function values may not be available.

3.1.1 L_2 Norm

We assume that the test function is provided for a validity of our study. We calculate a discrete norm between the actual function values and the approximated values at sampling points in a tetrahedron. A subdivision process increases the number of sampling points. The triangulation has the smallest errors in the sense of L_2 norm. An error is the difference between exact function values and approximated values at points in the given tetrahedral domain.

L_2 norm of $TH(V_a, V_b, V_b, V_d) =$

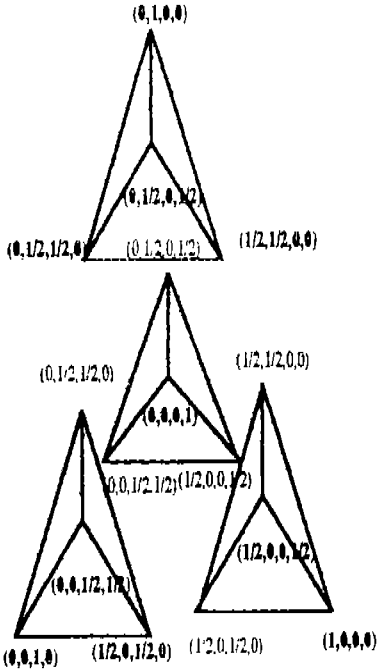
$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}[F(x_i, y_i, z_i)-f(x_i, y_i, z_i)]^2}\ , \qquad (3.1)$$

where $TH(V_a, V_b, V_c, V_d)$ is the tetrahedron that has four vertices V_a, V_b, V_c, and V_d, n is the number of sampling points,

$F_i = F(x_i, y_i, z_i)$ is at approximated function value on sampling point p_i, and $f_i = f(x_i, y_i, z_i)$ is at exact function value on sampling point p_i.

3.1.2 Sub-Division

The increasing of sampling points in a tetrahedron may improve the accuracy of calculating a discrete norm between the actual function value and the approximated one. We can increase sampling points by using a subdivision process. In the second level of subdivision, the number of sampling points is 10 points. The number of sampling points increases to 20 and 35 points at the third and forth subdivision, respectively. We present sampling points of the second level of subdivision in Figure 3.1



(Fig. 3.1) The second level sub-division of a tetrahedron (10 points bold letters only).
(그림 3.1) 한 사면체의 2 차 세분화(굵은 활자로 표기된 10 점)

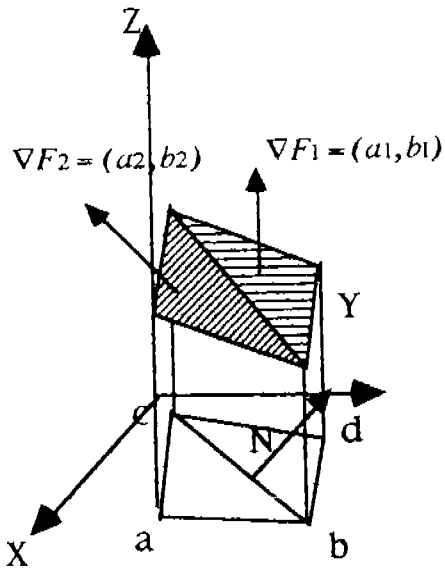
3.2 Gradient Difference Criterion

Definition 1 : if $f : U \subset \mathbf{R}^2 \rightarrow \mathbf{R}$ is differentiable, the gradient of f at (x, y, z) is the vector given by

$$\text{grad } f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right).$$

The direction of a gradient is the fastest direction of change of f. The magnitude of a gradient vector is the rate of change in that direction. We need to minimize the difference between each adjacent functional surface/space (2-D/3-D cases respectively) to satisfy the nearly C^1. As an example, we will explain this concept for a two-dimensional domain, criterion in section 2.1.1 (see Figure 3.2).

Let $F(x, y) = ax + by + c$ be the linear interpolating polynomials on a triangle. A linear function is uniquely defined by these values at the three vertices of a triangle.



▤ $= F_1(x, y) = a_1 x + b_1 y + c_1$:
▨ $= F_2(x, y) = a_2 x + b_2 y + c_2$:

∇F_1 and gradient of ∇F_2 are gradient of F_1 and F_2, respectively ;

N is a normal vector of common edge $\overline{bc}$ on 2-D domain of convex hull abcd.

(Fig. 3.2) The gradient of each adjacent surface of 3-D functional space from 2-D domain.
(그림 3.2) 2 차원적 영역으로부터 3 차원적 함수 공간상에서 각 인접표면간의 경사

To achieve a smooth transition between two adjacent surfaces, we minimize the difference of their gr-

adients. We can apply this concept to a 3-D domain through an analogous method of a 2-D domain.

Let $F(x, y, z)=ax+by+cz+d$ be linear interpolating polynomials on a tetrahedron. A linear function is uniquely defined by these values at the four vertices of a tetrahedron. ∇F is orthogonal to the isosurface in 4-D space.

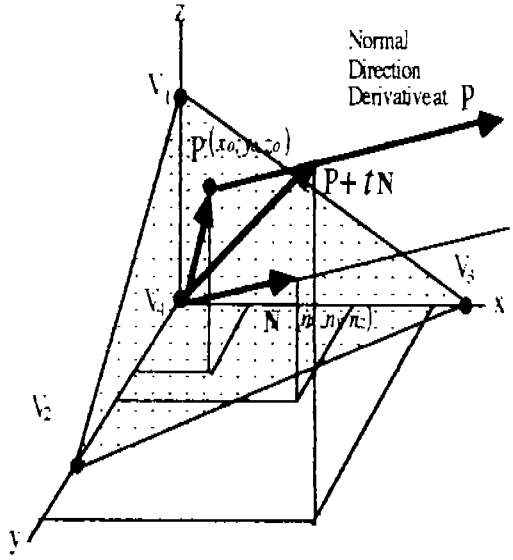
A gradient difference between two adjacent tetrahedra is

$$|\mathbf{G}_1-\mathbf{G}_2| = \sqrt{(a_1-a_2)^2+(b_1-b_2)^2+(c_1-c_2)^2} \quad (3.2)$$

where $\mathbf{G}_1$ is the gradient vector of a tetrahedron TH_1 (V_1, V_2, V_3, V_4) in 4-D functional space (i.e. $(a_1, b_1, c_1)^T$). $\mathbf{G}_2$ is the gradient vector of a tetrahedron TH_2 (V_1, V_2, V_3, V_5) in 4-D functional space (i.e. $(a_2, b_2, c_2)^T$)

3.3 Jump in Normal Direction Derivatives Criterion

Jump in normal derivatives criterion can be explained as minimizing the difference of normal derivatives across the common edge (i.e. edge $\overline{bc}$ in Figure 3.2)[7]. In our case, we minimize the difference of directional derivatives along the direction of N. N is the normal vector of the common face.



(Fig. 3.3) Directional derivative along the normal of a common face $\Delta V_1V_2V_3$

(그림 3.3) 공통 면인 삼각형 $v_1v_2v_3$의 정규 방향에 따른 방향성 미분

We need to define a directional derivative at $\mathbf{P}=(x_o, y_o, z_o)^T$ in the direction of normal vector $\mathbf{N}=(n_x, n_y, n_z)^T$ which is the normal vector of common face $\Delta V_1 V_2 V_3$ (see Figure 3.3).

Definition 2: if $f: \mathbf{R}^3 \rightarrow \mathbf{R}$, the directional derivative of f at $\mathbf{P}=(x_o, y_o, z_o)$ in the direction of a unit vector N is given by

$$\frac{d}{dt} f(\mathbf{P}+t\mathbf{N})|_{t=0}.$$

The directional derivative can also be defined by the formula:

$$\lim_{t\rightarrow o} \frac{f(\mathbf{P}+t\mathbf{N})-f(\mathbf{P})}{t},$$

where $f(\mathbf{P}+t\mathbf{N})=a(x_o+n_xt)+b(y_o+n_yt)+c(z_o+n_zt)+d$, and $f(\mathbf{P})=a(x_o)+b(y_o)+c(z_o)+d$.

Let $D_1=(a_1n_x+b_1n_y+c_1n_z)$ and $D_2=(a_2n_x+b_2n_y+c_2n_z)$ be the limit directional derivative at the common face of tetrahedron TH_1 and TH_2, respectively. A difference between two directional derivatives along the normal vector of common face $\Delta V_1V_2V_3$ is

$$|D_1-D_2| = |(a_1-a_2)n_x+(b_1-b_2)n_y+c_1-c_2)n_z|. \quad (3.3)$$

4. Implementation

In this section, GEOMPACK is introduced. We use this package for constructing initial tetrahedrization (i.e. Delaunay tetrahedrization). Also, a simulated annealing algorithm is introduced. It is used for achieving global-like optimum.

4.1 GEOMPACK

GEOMPACK is a mathematical software package, written in standard FORTRAN 77 for GEneration

Of Meshes using GEOMetric algorithms. B. Joe provides all routines including an introduction, and the explanation of each routine[16]. We use GEOMPACK to generate convex tetrahedral domains in 3-D space to deal with large data sets (i.e. 1000 data points).

Data dependent tetrahedrization routines are written in the C language for small data sets (i.e. 20 and 50 data points) by author. We employ the GEOMPACK to achieve the fast construction of initial tetrahedrization for large data set (i.e. 1000 data points). To implement data dependent tetrahedrization, we modify GEOMPACK and combine simulated annealing routines in FORTRAN 77. Visualization routines are written in the C language by author. We produce the list of tetrahedra using a DEC 3000 Alpha-AXP and transmit them to a Silicon Graphics IRIS Crimson for visualization routines.

4.2 Simulated Annealing

Certain researchers have drawn an analogy between the annealing of metals and the solution of combinatorial optimization problems [17], finding relationship between the molecular vibration of molecules through many states and the simulated vibration of a stochastic system through its more artificial states.

Simulated annealing is a global optimization method that distinguishes between different local optima. From an initial state, the algorithm takes a step and the cost function is evaluated. With the minimum of cost function, any downhill step is accepted, and the process repeats from this new, down-step point. In a similar manner an uphill step may be accepted. Thus, the algorithm can escape from local optima. This uphill decision is made by comparison between a random number and the Boltzmann probability distribution shown in Equation (4.2) (metropolis criterion) [18]. As the optimization process proceeds, the number of uphill steps decreases and the state closes to a global optimum.

We can use a simulated annealing algorithm to approximate the global optimum with many local optima. In our context, "swap" means to change the way of tetrahedrization (see section 2.2.2). In other words, the common face of adjacent tetrahedra are transformed from vertical to horizontal, or vice versa. A good swap improves some measure of quality; a bad swap does not. One can reduce the probability of a bad swap according to an annealing schedule. We can determine the occurrence of a bad swap by comparing a random number with the Boltzmann probability distribution. Even a bad swap might lead to a good swap eventually, in the view of a global optimum.

The *ntemps* controls the stages in the annealing algorithm. The number of stages depends on the stage of convergence. The *nlimit* controls the number of swaps to be attempted at each stage of the algorithm, while *glimit* controls the number of good swaps allowed at each temperature. We choose the number *nlimt* and *glimit* based on the number of faces of tetrahedrization. In our case, we multiply the total number of faces by 5 (see Table 4.1). The initial tetrahedrization(Delaunay) provides the total number of faces. These two values allow the exploration of a wider class of possible tetrahedrizations, increasing the chance of finding the global optimum.

We should limit the number of good swaps. The idea is that if one makes too many good swaps too soon, the algorithm will not find its way out of a false valley and into the region of the global optimum. We can describe our simulated annealing algorithm as:

do initial tetrahedrization
do $k=1$ to *ntemps*
$t_k = t_{k-1} * r_k$
  do $l=1$ to *nlimit*
  while
  the number of good swaps $\leq$ *glimit*
    choose a random face of existing
    tetrahedra
    if *possible* (see 2.2.2)
    if *desirable* (see 2.2.2)
      make good swap

else

choose a random number θ

where $0 \leq \theta \leq 1$

if $\theta \leq e^{-d/t_k}$ accept bad swap

endif

endif.

Initial temperature t_0 is fixed as twice the maximum difference of cost between one way of swapping and another. In our context, cost means the difference between the actual function values and the approximated values at sampling points. If the initial temperature is too high, the space is searched inefficiently (virtually every bad step is accepted). If the initial temperature is too low, too many function evaluations are required to escape from a local optimum. The t_k is decreased by factor r_k in Equation (4.1) at each stage. In our case, r_k is 0.95.

$$t_k = t_{k-1} * r_k \tag{4.1}$$

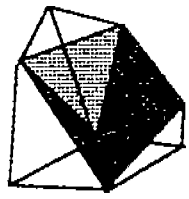
The probability of a bad swap is slowly reduced according to a prescribed annealing schedule. The probability of making a bad swap at temperature t_k for a given value d is

$$Pr\left(\theta \leq e^{-d/t_k}\right) = e^{-d/t_k} \tag{4.2}$$

where d is the difference of cost between one way of triangulation and another, and θ is a random number such that $0 \leq \theta \leq 1$.

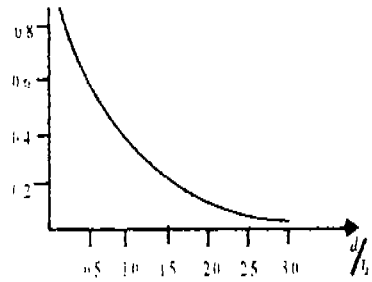
The t_k is analogous to temperature in the annealing of metal. As the t_k gets smaller, a bad swap is less likely to occur. The initial temperature can be almost twice the greatest cost (see Table 4.2). If one selects the small initial temperature, the algorithm will converge quickly, but often it will converge to a local optimum instead of global one.

The total cost is calculated by summing cost values at each common face based on criteria defined in Chapter 4 such as least squares fitting, gradient difference, and jump in normal direction derivatives. It is very inefficient to determine the acceptance of each swapping by calculating the total cost. In our case, local swapping is restricted in its convex hull. In Figure 4.1, the shaded tetrahedra represent the convex hull for the local swapping which is restricted. In other words, the cost of a locally swapped common face is changed, while others remain the same. When determining the acceptance of each swap, we can save the time required to evaluate the total cost by calculating only the cost of the changed common face.



(Fig. 4.1) Convex hull(shaded) for the local swapping which is restricted.
(그림 4.1) 제한된 지역적 교환을 위한 convex hull

The probability of accepting bad swaps can be determined by an annealing scheduling (see Figure 4.2). Our annealing scheduling has the characteristics of the Boltzmann probability distribution. An annealing scheduling consists of two stages. In the first stage, the system is optimized at a high, effective temperature. In the second stage, the temperature is cooled by slow stages until the system freezes and no further



(Fig. 4.2) Probability to making bad swaps.
(그림 4.2) 좋지 못한 교환이 일어날 수 있는 확률

changes occur. What might be considered a bad swap for a local optimum, might be a good swap for a global optimum, or vice versa. Also, what appears as a bad swap originally for a local optimum, might be a good swap in the long run.

To search for an optimal tetrahedrization at each stage, one may choose at random, from all the tetrahedra and four faces, one tetrahedron and one face. The algorithm improves continuously by swapping interior faces until it converges.

〈Table 4.1〉 Values of Parameters nlimit and glimit
〈표 4.1〉 매개변수 nlimit와 glimit의 값

number of data points	5×total number of faces
1000	5×12697
5000	5×66190
10000	5×132892

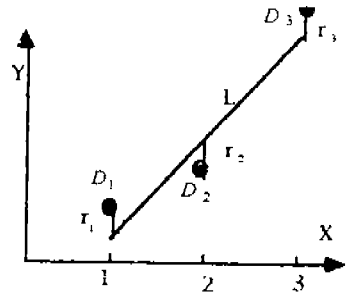
〈Table 4.2〉 Values of Parameter t_0
〈표 4.2〉 매개변수 t_0의 값

criterion	test function 2	test function 8
least sq.	0.1	0.6
grad. diff.	6.6	6.6
dir. deriv.	6.2	6.2

4.3 Averaging Technique

One must compare the case in which two tetrahedra become three tetrahedra in an objective manner. Two tetrahedra have one common face; however, three tetrahedra have three common faces. One needs to employ a reasonable averaging method for comparison with one common face and three common faces. We use the least squares averaging technique.

For example, we need to find the average of three cost values (i.e. D1, D2 and D3). We need to sort three cost values in ascending order. We try to fit three cost values with a straight line L, which has a minimum of R (see Figure 4.3). After we determine coefficients a and b of the straight line L, we calculate the polynomial value of its midpoint(i.e. $x=2$, $y=ax+b$).



$R = r_1^2 + r_2^2 + r_3^2$;
L is the straight line which is a linear function $y = ax + b$;
D_i is three cost values ;
r_i is the difference between polynomial value and the observed data value.

(Fig. 4.3) Least squares averaging technique.
(그림 4.3) 최소 제곱 평균 기법

5. Validation Study

We introduce 2 test functions based on the 3-D version of Franke's test function, and explain visualization techniques for 4-D functional space including iso-surface and color contouring methods, while presenting numerical and graphical results.

5.1 Test Functions

$$f_2 = \frac{tanh(9y-9x-9z)+1}{9} \tag{5.1}$$

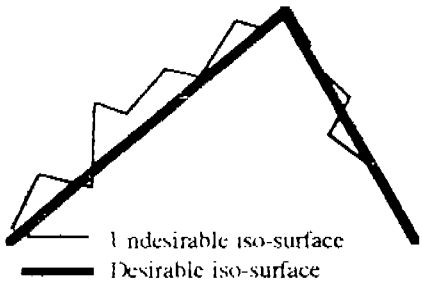
$$f_8 = tanh(-3\times(0.595576\times z+3.79762)^2 - x - y - 10)) + 1.0 \tag{5.2}$$

The above two test functions can be considered by referring to Franke's test functions[19]. To make the 4-D version of Franke's test functions, we just add another parameter 'z' to each 2-dimensional original test function.

5.2 Visualization of Iso-surfaces

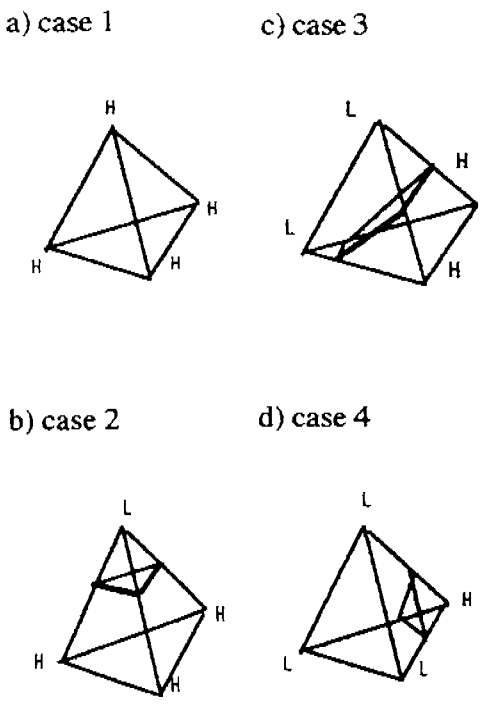
One can use the iso-surfaces visualization to investigate the quality of 4-D space through trivariate in-

terpolation. In Figure 5.1, the comparison between the desirable surface and the undesirable surface is shown. We assume that an actual iso-surface is flat.



(Fig. 5.1) Validation study based on iso-surface method.
(그림 5.1) 등면법에 기초를 둔 타당성 검사 연구

One can classify all possible cases of iso-surface in a tetrahedron as sixteen cases. One can reduce the number of cases to five by eliminating geometrically identical cases (see Figure 5.2).



(Fig. 5.2) Five cases of intersection between a tetrahedron and iso-surface.
(그림 5.2) 사면체와 등면과 교차할 때 5 가지 경우

5.3 Visualizing 4-D Space Using Color Contouring

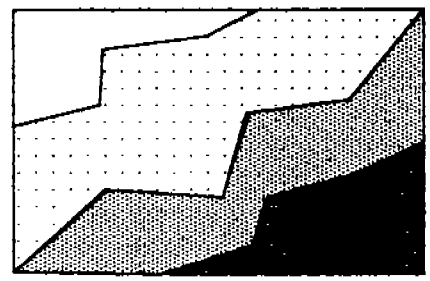
We restrict ourselves to interpolations from the space $S_1^0(TH)$ of piecewise linear functions defined by its values at the four vertices of each tetrahedron TH_i of (Ω). Mathematically,

$$S_1^0(TH) = \{ g \in C^0(\Omega) : g|_{TH_i} \in \Pi_1 \}, \quad (i = 1, \ldots, N) \quad (5.3)$$

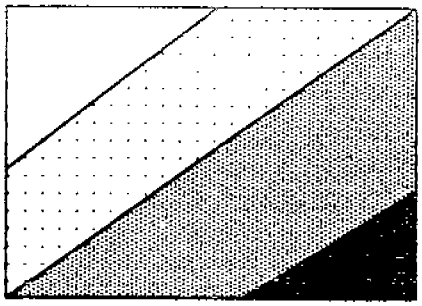
where Π_1 is the space of linear polynomials of (x, y, z).

We assign a different color for each functional bandwidth to visualize 4-D function space through color contouring method.

In Figure 5.3, the comparison between the desirable functional contour and undesirable functional contour is shown. We assume that an actual functional contour is linear.



Undesirable functional contour



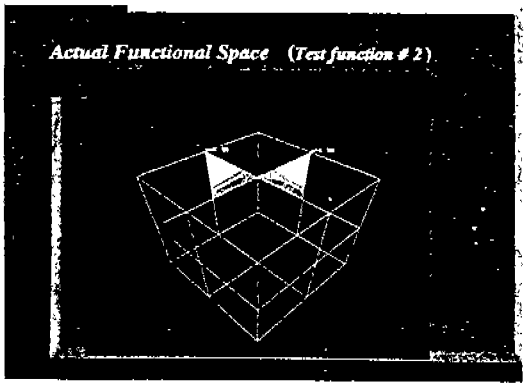
(Fig. 5.3) Validation study based on color contouring method.
(그림 5.3) 색채 윤곽법에 기초를 둔타당성 검사 연구

5.4 RMS Error

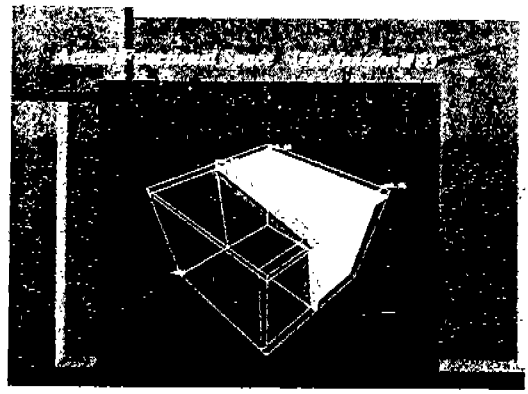
One can explain the method of calculating RMS error with L_2 norm (see section 3.1.1).

One samples a unit cube rectilinearly through 8000 points (i.e. $20 \times 20 \times 20$). Some parts of sampling points are inside of the convex tetrahedra and have approximated values. We use two test functions, proposed in the Dyn's paper [9] We make 4-D test functions from 3-D test functions.

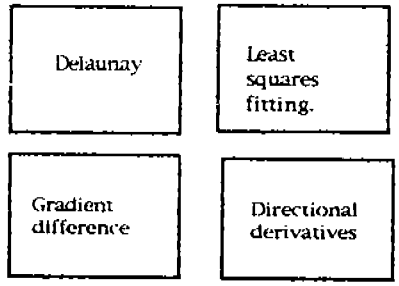
The test function $f_2(x, y, z)$ simulates a sharp rise, making cliff-like iso-surfaces in a unit cube. The test function $f_8(x, y, z)$ resembles $f_2(x, y, z)$. Their iso-surfaces are also cliff-like. Their actual function spaces of test function 2 and 8 are shown in Figure 5.4 and Figure 5.5, respectively. Data dependent criteria work more effectively on a function that has a preferred direction. We test five data set (i.e. 1000 data points). Data dependent criterion, least squares fit criterion, leads to a smaller RMS error than Delaunay tetrahedrization (i.e. Sphere criterion).



(Fig. 5.4) An actual functional space of test function #2.
(그림 5.4) 2 번째 시험 함수의 실제적 함수 공간



(Fig. 5.5) An actual functional space of test function #8.
(그림 5.5) 8 번째 시험 함수의 실제적 함수 공간

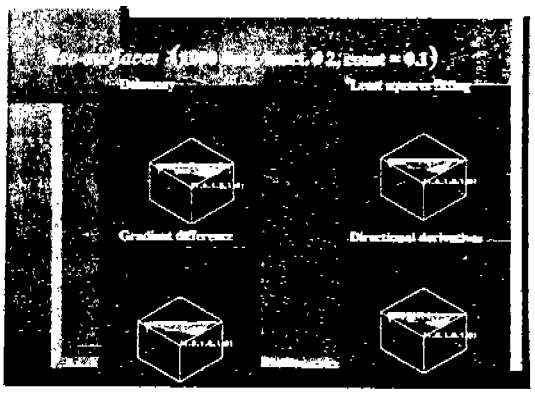


(Fig. 5.6) Layout of Figures 5.7 through 5.10
(그림 5.6) 그림 5.7에서 그림 5.10 까지 내용 배치도

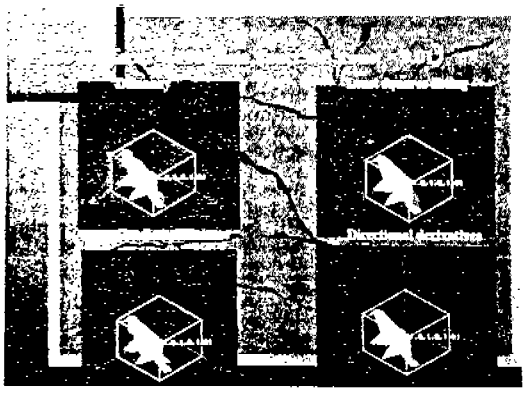
All figures, 5.7 through 5.10, have the same layout as shown in Figure 5.6. Delaunay tetrahedrization is based on sphere criterion. It is a data independent criterion. Least squares fitting, gradient difference, and jump in normal direction derivatives are data dependent criteria. We will show that tetrahedrization based on data dependent criteria provide a better approximation than the tetrahedrization based on data inde-

pendent criterion (Delaunay tetrahedrization). In Figure 5.7 and 5.8, the iso-surface (i.e. constant function value is 0.1) and the functional space are shown respectively (the test function #2 and 1000 data points). In Figure 5.9 and 5.10, the iso-surface (i.e. constant function value is 0.01) and the functional space are shown respectively (the test function #8 and 1000 data points).

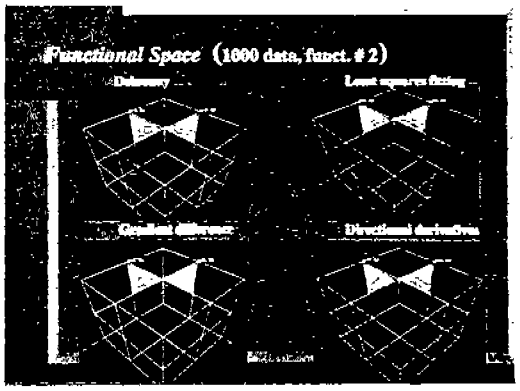
We can see that data dependent criteria (i.e. least squares fitting, gradient difference, and jump in directional derivatives) provide a better approximation than sphere criterion (i.e. Delaunay tetrahedrization). Also, least square fitting criterion leads to a better approximation than other data dependent criteria. However, least squares criterion assumes that exact function values are already known.



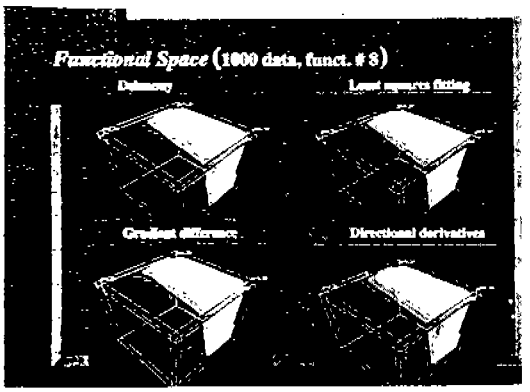
(Fig. 5.7) Visualization of iso-surface(const=0.1, test function #2, 1000 data).
(그림 5.7) 등면의 가시화(상수 = 0.1, 2 번째 시험 함수, 자료 1000 개)



(Fig. 5.9) Visualization of iso-surface(const=0.01, test function #8, 1000 data).
(그림 5.9) 등면의 가시화(상수 = 0.01, 8 번째 시험 함수, 자료 1000 개)



(Fig. 5.8) Visualization of functional space (test function #2, 1000 data points).
(그림 5.8) 함수 공간의 가시화 (2 번째 시험 함수, 자료 1000 개)



(Fig. 5.10) Visualization of functional space (test function #8, 1000 data points).
(그림 5.10) 함수 공간의 가시화 (8 번째 시험 함수, 자료 1000 개)

a) Test function:

$$f_2(x, y, z) = \frac{\tanh(9y - 9x - 9z) + 1}{9} .$$

Resolution 6787/8000 points and 1000 data points

* Error measurement based on L_2 norm	
1) Delaunay:	0.007475
2) Least squares fitting:	0.001570
3) Gradient difference:	0.005445
4) Directional derivatives:	0.004361

b) Test function: $f_8(x, y, z) = \tanh(-3g(x, y, z)) + 1$,
where $g(x, y, z) = 0.595576(z + 3.79762)^2 - x - y - 10$.

Resolution 6787/8000 and 1000 data points

* Error measurement based on L_2 norm	
1) Delaunay:	0.091196
2) Least squares fitting:	0.036494
3) Gradient difference:	0.073935
4) Directional derivatives:	0.059416

5.5 Time Analysis

To do time analysis, we measure CPU times on a DEC 3000 (Alpha-AXP processor). The time unit is a second. Least squares fitting criterion is the most time consuming method. Delaunay tetrahedrization is extremely faster than any other data dependent tetrahedrizations. The most time would be taken by a simulated annealing to achieve the global-like optimum. If accuracy is extremely critical, the cost of computation time can be compensated through super computing machine.

a) $f_2(x, y, z) = \frac{\tanh(9y - 9x - 9z) + 1}{9}$.

i) 1000 data points	
1) Delaunay:	1.9
2) Least Squares:	15241.7
3) Gradient Difference:	7087.8
4) Directional Derivative:	6319.8

b) $f_8(x, y, z) = \tanh(-3g(x, y, z)) + 1$,
where $g(x, y, z) = 0.595576(z + 3.79762)^2 - x - y - 10$.

i) 1000 data points	
1) Delaunay:	1.9
2) Least Squares:	9269.6
3) Gradient Difference:	4899.0
4) Directional Derivative:	3548.9

6. Conclusions

This paper discussed new data dependent criteria including gradient difference and jump in normal direction derivatives. The extension of Schumaker's least squares fitting criterion is defined on 3-D space. This paper established several methods of a validation study including the visualization of iso-surfaces, and the color contouring of a functional space.

A variety of test cases verified that data dependent tetrahedrization improves the quality of approximation and that long and thin tetrahedra, which are avoided in other criterion (i.e. sphere criterion), may be suitable. This statement is especially true when the test function has a strong preferred direction.

The shape of tetrahedra in domain becomes more sensitive to RMS error when we perform a linear interpolation based on the given domain. In other words, the shape of tetrahedra in domain becoms more important in case of linear interpolation than higher degree of interpolation than higher degree of interpolation. In this paper, the visulaization of 4-dimensional scattered data interpolants from tetrahedral domain is focused. The visualization of tetrahedral domain will be presented in a separated paper.

Data dependent tetrahedrization may not be better than Delaunay tetrahedrization for a certain type of data (i.e. round-like function). We assume that our data are randomly scattered. However, this may not be true in a real world application (i.e. medical im-

age). We need to find more desirable criteria which can be useful for more general types of data. We strongly believe that our visualization routines are useful for verification of new criteria.

Applications of 4-dimensional scattered data visualization can be enumerated, including climate data, medical image data, and air pollution data. For further research, high degree interpolation can be attempted instead of linear interpolation, even high complexity of computation for gradient estimation is expected. In this case, we need to consider a trade-off between accuracy and cost computing.

REFERENCES

[1] Nielson, G. M. "Modeling and visualizing volumetric and surface-on-surface data," In Focus on Scientific Visualization, H. Hagen, H. Mueller and G. Nielson, Eds. Springer, pp. 219-274, 1992.

[2] Nielson, G. M. "Scattered data modeling," Computer Graphics and Applications, vol. 13, no. 1, January, pp. 60-70, 1993.

[3] Nielson, G. M. and Franke, R. "Scattered data interpolation and applications: A tutorial and survey," In Geometric Modeling: Methods and Their Applications, H. Hagen & D. Roller, Eds. Springer, pp. 131-160, 1993.

[4] Nielson, G. M. and Tvedt J. "Comparing metods of interpolation for scattered volumetric data," In State of the Art in Computer Graphics-Aspects of Visualization, D. Rogers and R. A. Earnshaw Eds. Springer-Verlag, pp. 67-86, 1993.

[5] Nielson, G. M. and Franke, R. "Surface construction based upon triangulations," In Surfaces In Computer Aided Geometric Design, R. E. Barnhill and W. Boehm, Eds. North-Holland Pub. Co., pp. 163-179, 1983.

[6] Barnhill, R. E. and Little, F. F. 1984. "Three-and four-dimensional surfaces," The Rocky Mountain Journal of Mathematics, vol. 14, no. 1, pp. 77-102, 1984.

[7] Dyn, N., Levin, D., and Rippa, S. "Data dependent triangulations for piecewise linear interpolation," IMA Journal of Numerical Analysis, vol. 10, pp. 137-154, 1990.

[8] Preparata, F. P., Shamos, M. I. 'Computational Geometry', Springer-Verlag, New York, 1985.

[9] Edelsbrunner, H. 'Algorithms in Computational Geometry,' Springer-Verlag Berlin, 1987.

[10] Schumaker, L. L. "Computing optimal triangulations using simulated annealing," Computer Aided Geometric Design 10, pp. 329-345, 1993.

[11] Delaunay, B. "Sur la Sphere Vide," Bull. Acad. Sci. USSR (VII), Classe Sci. Mat. Nat., pp. 793-800, 1934.

[12] Watson, D. F. "Computing the n-dimensional Delaunay tessellation with applications to Voronoi polytopes," The Computer Journal 24 (2), pp. 167-172, 1981.

[13] Edelsbrunner, H., Preparata, F. P. and Wast, D. B. "Tetrahedrizing point sets in Three Dimensions," J. Symbolic Computation, vol. 10, pp. 335-347, 1990.

[14] Joe, B. "Three-dimensional triangulations from local transformations," SIAM J. Sci. Stat. Comput. 10, pp. 718-741, 1989.

[15] Joe, B. "Construction of three-dimensional Delaunay triangulations using local transformations," Computer Aided Geometric Design 8, pp. 123-142, 1991.

[16] Joe, B. "GEOMPACK users' guide," Department of Computing Science, University of Alberta, Canada, 1993.

[17] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller A., and Teller J. "Equation of state calculations by fast computing machines," Journal of Chemical Physics 21, 1087-1091, 1953.

[18] Kirkpatrick, S., Gelatt, C., and Vechni, M. "Optimization by simulated annealing," Science 220, 671-680, 1983.

[19] Franke, R. "A critical comparison of some methods for interpolation of scattered data," Report

NPS-53-79-003, Naval Postgraduate School, 1979.

이 건

1978년 고려대학교 전자공학과 졸업(학사)

1983년 고려대학교 대학원 전자공학과(공학석사)

1985년 아리조나 주립대학교 대학원 전기공학과(공학석사)

1995년 아리조나 주립대학교 대학원 전산과학과(공학박사)

1995년~현재 전북산업대학교 컴퓨터공학과 전임강사

관심분야 : 컴퓨터그래픽스(scientific visualization), 멀티메디아.