

## 不規則形状生成のためのモデリング手法

藤本 忠博<sup>†</sup>      大野 義夫

慶應義塾大学 理工学部

エンジニアリングやデザインなどの分野では、物体の表面を構成するデータ点群の座標値から表面格子網(メッシュ)を生成することが必要となる場合が多い。このとき重要なのは、各データ点どうしの連結に対して、対象となる物体形状を表現するための正しい位相的および幾何的關係をもたせることである。

本研究では、データ点群に対してDelaunay 3角網を形成しHamilton閉面を抽出することで、最適な表面格子網の生成を行なう手法を提案する。本手法には、

- 各データ点の順序や位置関係は、いかなる規則性をも必要としない。
- 生成される物体の形状には制約がない。

という特徴があり、非常に自由度の高い形状モデリングを実現することができる。

## A modeling technique for generating irregular mesh

Tadahiro Fujimoto<sup>†</sup>      Yoshio Ohno

Keio University, Faculty of Science and Technology

In CAE and CAD, when the data points on the surface of an object are given, we need a surface mesh which is generated by connecting them suitably. Then, it is important to connect the data points so that the topological and geometric feature of the object is properly expressed.

The technique proposed in this paper uses Delaunay triangulation and Hamiltonian sphere to create a suitable surface mesh. This technique has the following advantages:

- No regularity is necessary in the order of data points.
- There is no constraint in the shape of object.

Therefore, this technique realizes a very flexible modeling.

---

<sup>†</sup>現在: 三菱総合研究所 (Mitsubishi Research Institute)

## 1 はじめに

エンジニアリングやデザインなどの分野では、対象となる物体の形状をコンピュータ内部に構築する。そして、物体の表面を構成するデータ点群の座標値の情報のみから最適な表面格子網(メッシュ)を生成することが必要となる場合が多い。

ここで問題となるのは、同じデータ点群であっても、その連結の仕方によって全く異なる格子網が生成されうることである。したがって、各データ点どうしの連結に対して、対象となる物体形状を表現するための正しい位相的および幾何的關係をもたせることが重要となる。

これを実現するため、一般には、あらかじめデータ点を一定の順序で並べ、規則的な格子網の生成を行なうことが多い。ところが、この方法では、データ点のサンプリングに対する制約が大きく、デジタイザによって入力を行なう場合などには非常に手間がかかる。また、格子網の規則性のため、複雑な物体を扱うことは困難である。

これらの問題を解決するため、これまでに、不規則に並べられたデータ点群から適切な格子網を生成する方法がいくつか提案されている(図1)。しかし、どの方法も、あらゆる場合への適用は不可能であり、複雑な物体形状に対して十分に対応できないのが現状である。

そこで、本研究では、3次元空間上に分布するデータ点群の座標値の情報のみから、それらの位相的ならびに幾何的關係を決定し、それらによって構成されるべき最適な表面格子網を生成する手法を提案する。本研究で提案する手法には、

- 各データ点の順序や位置関係は、いかなる規則性をも必要としない。
- 生成される物体の表面形状には制約がない。

という特徴があり、非常に自由度の高い形状モデリングを実現することができる。

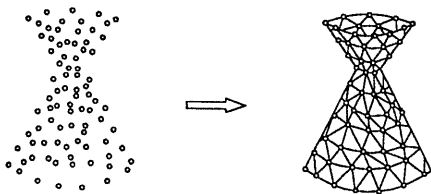


図1: 不規則な格子網生成

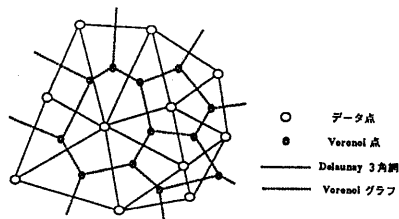


図2: Delaunay 3角網とVoronoi グラフ

## 2 本研究の概要

3次元物体の表面格子網は、一般に2次元多様体となる。そのため、これまでに提案されてきた方法は、はじめから各データ点間に2次元のな關係を構築してしまうものであった。しかし、データ点群は3次元空間上に分布するため、このようなアプローチのしかたでは、すべての場合において各データ点どうしの最適な位相關係を決定することはできない。

そこで、本研究では、データ点群の個々の位置情報から全体的な位相關係を決定するため、各データ点間にいったん3次元のな關係を構築し、そこから2次元のな表面格子網を抽出するという方針をとる。

本研究で提案する方法の概略は、次のようになる。

1. 3次元 Delaunay 3角網の形成 ...  
データ点群から3次元の4面体格子網を生成する。  
↓
2. 3次元 Hamilton 閉面の抽出 ...  
生成された4面体格子網から、物体形状を表現する最適な3角形表面格子網(2次元多様体)を抽出する。

## 3 Delaunay 3角網

### 3.1 Delaunay 3角網の定義

Delaunay 3角網とは、データ点群中の各データ点を連結して生成される3角形格子網の代表的なものである。

2次元の場合には、Delaunay 3角網は、各3角形(要素3角形)の外接円内に他のデータ点が入らないという外接円原理、あるいは、各要素3角形の最小角の大きさが最大になるといふ最小角最大原理に基づいて形成される(図2)。この2つの原理は互いに等価であることが証明されている。

また、これに関連して、Voronoi グラフが定義できる。Voronoi グラフとは、Delaunay 3 角網の各要素 3 角形について、互いに辺 (edge) を共有するものどうしの外接円の中心 (Voronoi 点) を連結してできるグラフである (図 2)。

Voronoi グラフの枝 (branch) で囲まれた Voronoi 領域は凸形の多角形となり、その内部にはデータ点が 1 つ含まれる。各データ点が自分の領域内の任意の位置から最も近いものとなることから、Voronoi グラフは勢力圏図とも呼ばれる。

3 次元の場合には、Delaunay 3 角網は 4 面体 (要素 4 面体) を単位とする 4 面体格子網となる。これも 2 次元の場合と同様に、外接球原理または最小角最大原理に基づいて生成され、各要素 4 面体の外接球の中心を結ぶことで Voronoi グラフが形成される。

なお、データ点の総数  $n$  と Voronoi 点の総数  $m$  の関係は、2 次元の場合には  $m = O(n)$ 、3 次元の場合には  $m = O(n^2)$  となる [1]。

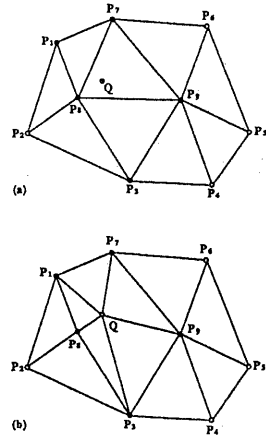


図 3: 1 ステップでの 3 角網の更新

### 3.2 Delaunay 3 角網の生成アルゴリズム

Delaunay 3 角網の生成アルゴリズムは、その方針によって大きく 2 種類に分けることができる [2]。逐次型アルゴリズムは、初期の 3 角形あるいは 4 面体からスタートして、徐々にデータ点を付加しながら 3 角網を生成していくというものである。それに対して、分割統治型アルゴリズムは、最終的な 3 角網が得られるまで再帰的に領域を分割していく方法である。

本研究では、A. Bowyer[3] および D. F. Watson[4] の方法に類似した逐次型アルゴリズムを提案する。

この方法では、データ点が追加されるごとに 1 ステップで 3 角網を更新する。その例を図 3 に示す。ある段階までで生成された 3 角網に新たなデータ点が追加された場合、そのデータ点を外接円内に含む要素 3 角形は、そのデータ点を 1 つの頂点とする新たな要素 3 角形に置き換えることになる。そこで、追加されたデータ点を外接円内に含む要素 3 角形を探索し、それらを更新していくことが、この方法の特徴である。

このアルゴリズムの 3 次元の場合の概略は次のようになる。

1. 初期 4 面体の形成
2. Delaunay 3 角網の生成

すべてのデータ点について以下の作業を繰り返す。

- (a) データ点を内部に含む要素 4 面体の探索

4 面体の 番号	4 面体を構成するデータ点の番号				隣接する 4 面体の番号			
	①	②	③	④	①	②	③	④
⋮								
⋮								

図 4: データ構造

- (b) データ点を外接球内部に含むすべての要素 4 面体の探索

- (c) 3 角網の更新

### 3. 初期 4 面体の削除

また、データ構造としては、操作上の効率のよさのため、図 4 のようなテーブル (Delaunay テーブル) を採用する。

このアルゴリズムによる時間計算量は、3 次元の場合、データ点数  $n$  に対して  $O(n^{5/3})$  となる。

## 4 Hamilton 閉面

### 4.1 Hamilton 閉面の定義

Hamilton 閉路とは、与えられた 2 次元グラフ中のすべての頂点を 1 度ずつ通過する閉路であり、このような閉路が存在するグラフを Hamilton グラフという。

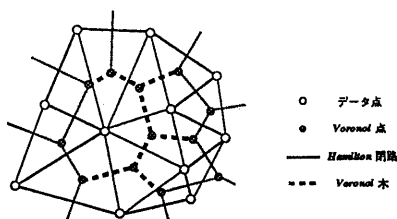


図 5: Voronoi 木

Hamilton 閉面とは、この Hamilton 閉路を 3 次元に拡張したものであり、3 次元グラフ中のすべての頂点を 1 度ずつ通過する閉多面体である。

Hamilton グラフに関しては、これまでのところ厳密な特徴づけがなされていない。あるグラフに Hamilton 閉路が存在するかどうかという問題は NP 完全問題の 1 つであり、これを効率よく解くのは困難であることが知られている。

#### 4.2 問題の変換

ここでの問題は、Delaunay 3 角網から Hamilton 閉面を求めることである。

これを解決するため、本研究では、直接 Hamilton 閉面を求めるのではなく、Voronoi グラフ (Delaunay 3 角網に対応) から Voronoi 木 (Hamilton 閉面に対応) を求めるという間接的なアプローチをとる。

Voronoi 木とは、Hamilton 閉面の内部に含まれる要素 4 面体の Voronoi 点を連結した木のことをいう。これはちょうど Hamilton 閉面に対する骨格のようなものとなる。2 次元の場合の例を図 5 に示す。3 次元の場合にはループができることもあり、厳密な意味で“木”であるとはいえないが、便宜上“Voronoi 木”と呼ぶことにする。

Voronoi 木をもちいることによってデータ点群を最適に連結する手法は、これまでにも、J. O'Rourke らによって提案されている [5]。J. O'Rourke らは、2 次元の場合について、Voronoi 木のうちで長さが最小となる最小 Voronoi 木が最適な形状をつくる Hamilton 閉路に対応するとし、いくつかの例によってその妥当性を検証している。

しかし、その手法に関する理論的な考察は行なわれておらず、ヒューリスティックな方法とされたままである。そして、最小 Voronoi 木を求めるための効率のよいアルゴリズムも提

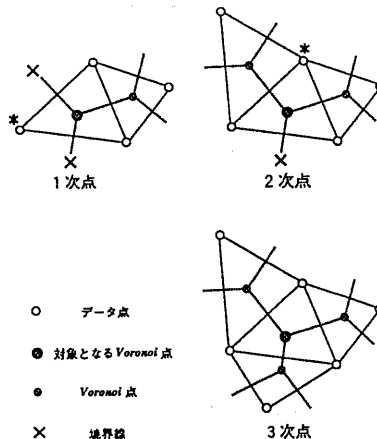


図 6: Voronoi 点の分類 (2 次元)

案されていない。また、3 次元の場合への適用についてもわずかに触れてはいるが、具体的な検証はされておらず、その可能性を述べている程度である。

そこで、本研究では、これを 3 次元に応用し実現していくために、Voronoi 木の性質を解析し、さらに最小 Voronoi 木を求めるための効率のよいアルゴリズムを提案することにする。

#### 4.3 Voronoi 木の性質

ここで、Voronoi 木の性質に関する検討を行なう。

まず、2 次元の場合について次のことがいえる。

##### 定義 1.1 (Voronoi 木の定義)

Voronoi 木とは、Hamilton 多角形 (Hamilton 閉路によって囲まれる多角形) を構成する各要素 3 角形の Voronoi 点を連結してできる木である (図 5)。

##### 事実 1.1 (Voronoi 点の分類)

Voronoi 点は 3 つの枝をもち、周囲の Voronoi 点との連結状態によって 1 次点、2 次点、3 次点に分類される (図 6)。

ここでは、Voronoi グラフから Voronoi 点を削除していくことで Voronoi 木が得られるという観点で検討を行なう (図 7)。すると、図 6 より次のことがいえる。

##### 規則 1.1 (Voronoi 点の削除)

Voronoi グラフから削除可能な Voronoi 点は 2 次点だけである。

枝によって閉じている各 Voronoi 領域内部には、それぞれ

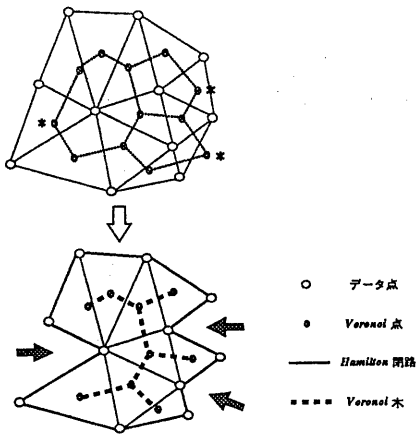


図 7: Voronoi グラフからの Voronoi 点の削除

1 個ずつデータ点が存在する (図 7). Hamilton 多角形をつくるためには, この Voronoi 領域内部に埋もれたデータ点を外部との境界線に出せばよい.

**規則 1.2 (局所性)**

閉鎖状態の Voronoi 領域内部のデータ点を境界線に出すためには, その領域に関して, その領域を構成する Voronoi 点を 1 つだけ削除して開放状態にすればよい.

証明 図 8 について考える. (a) 図において, Voronoi 領域は枝で閉じられ, \* 印のデータ点はその内部に埋もれた状態である. このような Voronoi 領域の状態を閉鎖状態と呼ぶことにする. このデータ点を境界線に出すためには, この領域を開放しなければならない. そこで, Voronoi 点を 1 個だけ削除して領域を開放したのが (b) 図である. このような Voronoi 領域を開放状態と呼ぶことにする. ところが, これからさらに Voronoi 点の削除を進めようとする, (c) 図の \*\* 点および (d) 図の \* 点のように不都合が生じる. このことから, 1 つの Voronoi 領域について 2 個以上の Voronoi 点は削除できないことがわかる. よって, 削除しなければならない Voronoi 点は 1 個のみでなければならない. □

このことを Voronoi グラフ全体で考えると, 次のことがいえる.

**規則 1.3 (大域性)**

全データ点を境界線に出すためには, 全 Voronoi 領域を開放状態にすればよい.

また, 次の事実によって, Voronoi グラフから Voronoi 点の

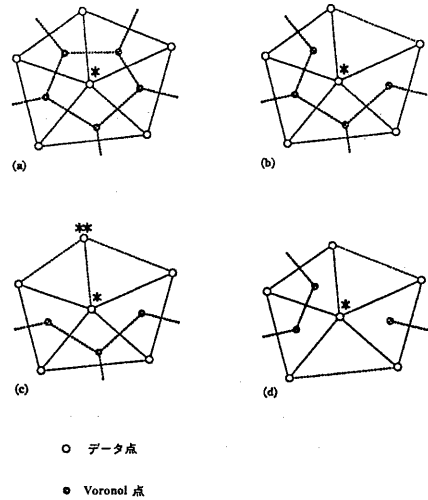


図 8: Voronoi 領域の開放

削除によって得られるものが Voronoi 木として妥当なものであることがわかる.

**事実 1.2 (妥当性)**

2 次点の削除による Voronoi 領域の開放によって, 木が分断されることはない.

以上のことから, 次の定理が導かれる.

**定理 1.1 (Voronoi 木の十分条件)**

Voronoi グラフの全 Voronoi 領域について, 2 次点を 1 つだけ削除して開放状態にすることで, Voronoi 木が得られる.

ところが, ここで次の問題が生じる.

**問題 1.1**

全 Voronoi 領域が開放可能であるのか. すなわち, 全 Voronoi 領域に 2 次点が存在するのか.

つまり, 図 9 のような場合には開放できない Voronoi 領域が残ってしまう.

しかし, 次の事実によってこの問題は解決可能である.

**事実 1.3 (2 次点の生成)**

各 Voronoi 領域に対する逐次的な 2 次点の削除で, 他の Voronoi 領域に新たな 2 次点が生じる.

すなわち, 各 Voronoi 領域に対する逐次的な 2 次点の削除によって, それまでに 2 次点をもたなかった Voronoi 領域にも 2 次点が発生するのである. しかし, 次の問題が残る.

**問題 1.2**

任意の Voronoi グラフに対して、全 Voronoi 領域に 2 次点が生じるような逐次的な開放のしかたが存在するのか。

このような開放のしかたが存在するとすれば、それは任意の Delaunay 3 角網に Hamilton 閉路が存在するというものである。前に述べたように、これは NP 完全問題である。そこで、本研究では、これに関して次のことを仮定する。

**仮定 1.1**

任意の Voronoi グラフ中に Voronoi 木が存在する。

ここでの結論として、2 次元の Voronoi 木について次の定理が導かれたことになる。

**定理 1.2 (Voronoi 木の条件)**

Voronoi グラフの全 Voronoi 領域について、逐次的に 2 次点を 1 つだけ削除して開放状態にすることで、Voronoi 木が得られる。

続いて、3 次元の場合については次のことがいえる。

**定義 2.1 (Voronoi 木の定義)**

Voronoi 木とは、Hamilton 多面体を構成する各要素 4 面体の Voronoi 点を連結してできる木である。

**事実 2.1 (Voronoi 点の分類)**

Voronoi 点は 4 つの枝をもち、周囲の Voronoi 点との連結状態によって 1 次点から 4 次点に分類される (図 10)。

**規則 2.1 (Voronoi 点の削除)**

Voronoi グラフから削除可能な Voronoi 点は 2 次点と 3 次点である (図 10)。

2 次点に対応する要素 4 面体上のデータ点はすべて境界面上に出ているため、内部のデータ点を境界面上に出すための 2 次点の削除は不要であるが、その削除によって Hamilton 多面体の形状が変更できる。しかし、3 次点に関しては、内部のデータ点を境界面上に出すために削除が必要である。つまり、2 次点は幾何的な影響を、3 次点は位相的な影響をもつと考え

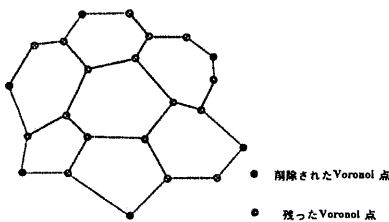


図 9: 開放不可能な Voronoi 領域

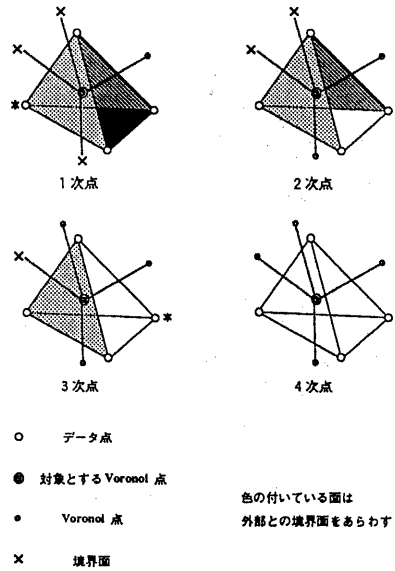


図 10: Voronoi 点の分類 (3 次元)

ることができる。そして、2 次元の場合と比較してみると、2 次元の場合における 2 次点と 3 次元の場合における 3 次点とが類似した特徴をもつことがわかる。

ここで、3 次元の場合の閉鎖状態の Voronoi 領域について次のことがいえる。

**事実 2.2 (閉鎖状態の Voronoi 領域)**

閉鎖状態の Voronoi 領域は、3 次点および 4 次点のみから構成される。

つまり、3 次元の Voronoi 領域を開放して内部のデータ点を境界面上に出すためには、3 次点を削除しなければならないことになる。

以上のことをもとに、2 次元の場合と 3 次元の場合との類似性を考えていくと、3 次元の Voronoi 木の条件として次の定理に達する。

**定理 2.1 (Voronoi 木の条件)**

Voronoi グラフの全 Voronoi 領域について、逐次的に 3 次点を 1 つだけ削除して開放状態にすることで、Voronoi 木が得られる。

また、2 次点の有無によって Voronoi 木を次のように分類する。

**定理 2.2 (Voronoi 木の分類)**

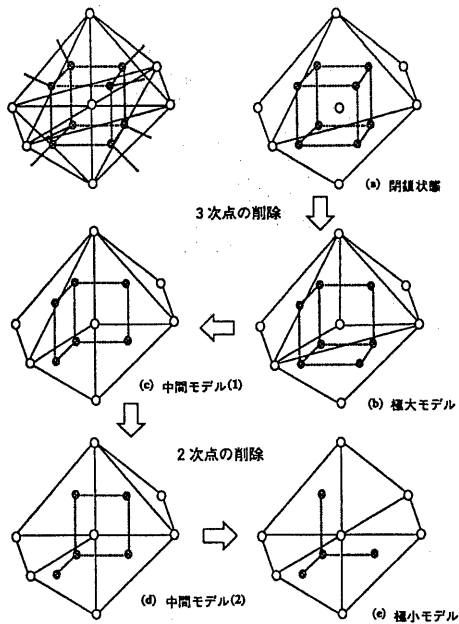


図 11: Voronoi 木の分類

Voronoi 木のうち、3 次点のみを削除して得られたものを極大モデル、極大モデル中のループを 2 次点の削除によりすべてカットしたものを極小モデル、その中間、すなわちいくつかのループが残されているものを中間モデルと呼ぶ。

1 つの例を図 11 に示す。(a) 図は閉鎖状態であり、中心のデータ点が内部に埋もれている。これから 3 次点を 1 つ削除することで Voronoi 領域を開放したものが (b) 図であり、これが極大モデルである。この状態で中心のデータ点は境界面上に出ることになる。(b) 図の状態からさらに 2 次点の削除を進めたのが、(c) 図、(d) 図および (e) 図である。(e) 図ではそれ以上カットできるループが存在しない。つまり、これが極小モデルである。(c) 図と (d) 図は中間モデルということになる。

このように、3 次元の場合には Voronoi 木および Hamilton 多面体があいまいさをもつ。よって、最適な形状を求めるためには、そのことを十分に考慮する必要がある。

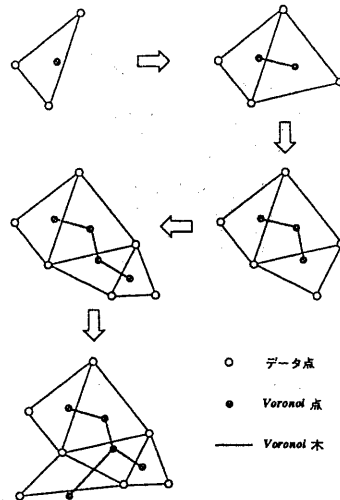


図 12: Voronoi 木の成長 (2 次元)

#### 4.4 最小 Voronoi 木問題の解法

ここで、最小 Voronoi 木を求めるためのアルゴリズムについて検討してみたい。

Voronoi 木を求める方法として、直感的に次の 2 つのアプローチが考えられる。1 つは、Voronoi 木の性質を考えるうえで方針としたように、Voronoi グラフの枝を削除していくことで Voronoi 木を得る方法であり、もう 1 つは、J. O'Rourke らも提案しているが、Voronoi 木の枝を成長させていく方法である。前者を shrink アルゴリズム、後者を grow アルゴリズムと呼ぶことにする。

Shrink アルゴリズムでは、定理 2.1 に基づいて 3 次点を削除していくことで Voronoi 木を得る。しかし、厳密に最小 Voronoi 木を得るためには、すべての Voronoi 木を全探索しなければならず、膨大な計算時間を要する。そこで、本研究では、3 次点の削除の過程で、その 3 次点に連結している 3 本の枝の長さの和が大きいものから順に削除を進め、はじめに得られる Voronoi 木を近似的に最小 Voronoi 木とみなすことにする。なお、この操作には、適切なバックトラック法の適用が必要となる。

Grow アルゴリズムでは、ある Voronoi 点から木の成長をスタートさせ、それまでに生成された木に連結している枝のうちで最短のものを木に加えていくことで最小 Voronoi 木を

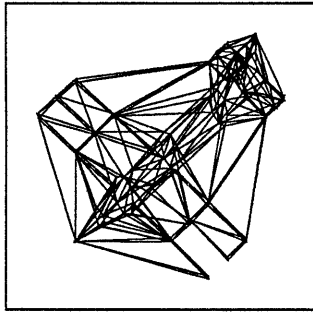


図 13: データ点数 ... 114 個

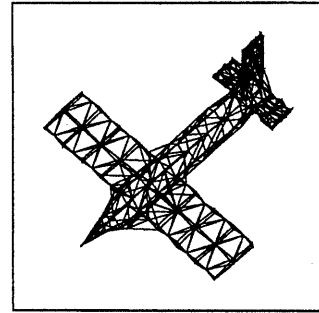


図 14: データ点数 ... 325 個

得る (図 12). すべての Voronoi 点をスタート点とし, 得られた Voronoi 木のうちで長さが最短のものを最小 Voronoi 木とする. また, 成長の停止条件によって, 定理の条件を完全に満たすまで成長を続ける極大 grow アルゴリズムと, すべてのデータ点が含まれた時点で成長を停止する最適 grow アルゴリズムが提案される.

## 5 評価

本研究で提案した技法について, 次の評価を行なった.

1. shrink アルゴリズムと grow アルゴリズムの比較
2. 極大 grow アルゴリズムと最適 grow アルゴリズムの比較
3. データ点数と生成された 3 角形格子網の形状の関係

1. に関しては, 実験の結果, grow アルゴリズムの有効性が明らかとなった. shrink アルゴリズムでは, Voronoi 点の削除によってはじめに得られたものを最小 Voronoi 木とみなしてしまうため, 真の最小 Voronoi 木とはかけ離れたものに陥ってしまうことが多いことがわかった.

2. に関しては, 最適 grow アルゴリズムの有効性が確かめられた. 極大 grow アルゴリズムによってできる極大モデルには余分な枝が多く, 結果としてできる Hamilton 閉面は最適な形状であるとはいえないものが多かった.

最後に 3. に関してであるが, データ点数の増加にともなって, 生成される 3 角形格子網の形状はより最適なものに近づくことが示された. その例を図 13 および図 14 に示す.

## 6 おわりに

本研究で提案したアルゴリズムによって, 3 次元空間上のデータ点群から, それらによって構成されるべき最適な 3 角形格子網を生成することができた. また, すべてのアルゴリズムのうち, 最適 grow アルゴリズムが最も有効であるという結論が得られた.

## 参考文献

- [1] 伊理 正夫, 腰塚 武志 “計算幾何学と地理情報処理 bit 別冊,” 共立出版, September 1986, pp.57-148.
- [2] L. D. Floriani, B. Falcidieno, C. Pienovi “Delaunay-based Representation of Surfaces Defined over Arbitrarily Shaped Domains,” Computer Vision Graphics and Image Processing vol.32, 1985, pp.127-140.
- [3] A. Bowyer “Computing Dirichlet tessellations,” The Computer Journal vol.24 no.2, 1981, pp.162-166.
- [4] D. F. Watson “Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes,” The Computer Journal vol.24 no.2, 1981, pp.167-172.
- [5] J. O'Rourke, H. Booth, R. Washington “Connect-the-Dots: A New Heuristic,” Computer Vision Graphics and Image Processing vol.39, 1987, pp.258-266.