２次元Delaunay三角形分割によるＣＧ用地形モデルの生成

東急建設（株）土木設計部　正会員　○　二宮　功
東急建設（株）土木設計部　正会員　田村治幸
岡山大学　正会員　谷口健男
岡山大学　学生員　岩崎和也

1. はじめに

コンピュータグラフィックス（ＣＧ）を利用した鳥瞰図やビデオの作成は、建築分野と同様に土木分野でも盛んに行われるようになってきている。ほんの数年前まではＳＦ映画やテレビのＣＭの中で見るに過ぎなかったＣＧであるが、今や土木の設計においても強力なプレゼンテーション用ツールとして欠かせないものになりつつある。そのＣＧ利用の目的は、ゴルフ場の完成予想作成、植栽の変更による道路の景観検討、工法や工事の概要説明など多種多様で対象とする工事もバラエティに富んでいる。

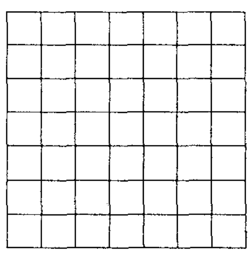
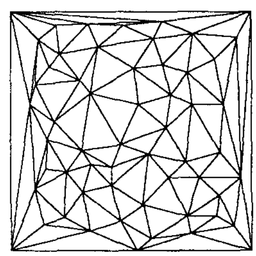
手書きのパース図や図面とＣＧを比べた場合、ＣＧを利用する長所としては、見たい角度からのパース図を素早く作ることができる、対象物の色や質感を変えて眺めの変化を見ることができる、ウォークスルーのアニメーションを作成しビデオ化することができる、設計寸法に従って計画を精確に入力することができるなどである。その反面、データの作成には多大な作業を要する。長所の方は一般にもよく知られており何ができるかということは理解されているようであるが、華やかなＣＧの影に隠れている問題点の方は、ＣＧを作成する側でないと実感できないところでありＣＧ用データの作成は大きな負担となっている。

そこで、特に土木では不可欠である地形のＣＧ化について、２次元Delaunay三角形分割を応用することで省力化を図ろうとするのが本研究の目的である。

2. 地形のモデル化[1),2)]

2.1 数値化地形モデル

造成工事の設計において、コンピュータを利用し土量を求める際メッシュ法により計算するが、その基になるデータが数値化地形モデル（DTM : Digital Terrain Model）である。DTMは、実際の測量や地形図から得られた３次元の座標値から生成されるもので、大きく分けて、図－１のように格子モデル（Grid based model）と三角形モデル（Triangulation based model）の二つがある。なお、三角形モデルは特にTIN（Triangulated Irregular Network）と呼ばれる。

図－１　格子モデルと三角形モデル

2.2 格子モデル

格子モデルは、広大な土地形状をなるべく少ないデータで表現しようとする場合に適しており、グリッドの形状は一般に正方形である。自然地形は洞窟やオーバーハングした崖など特別な状況を除くと2.5次元と見なすことができる。そこでこのモデルのデータは、DTMの範囲（グリッドの座標値の最大・最小値）とグリッドの間隔、横のグリッド数と縦のグリッド数、そしてグリッド上のｚ座標値だけで構成されている。このモデルは、コンピュータの能力がそれほど優れていなかった時代によく使われたものであるが、それは等高線の全節点数が何万点であろうと、DTMを処理するコンピュータの能力にあわせてグリッドの間隔を調整することにより、自由にデータを圧縮することができるからである。格子モデルは以下のような利点・欠点を持つ。

（利点）

・指定したグリッド上の点のｚ座標を高速に参照することができる

（欠点）

・等高線上の点などからグリッド上のｚ座標を内挿で求めるためにかなりの計算時間を要する
・グリッドは一様なものであるため、特定の場所について密なデータを作ることができない
・データがグリッドに乗っているため崖などの様子を正確に表現することができない
・サンプリングデータに関係なく、グリッド間隔によりできあがるDTMが決まる

2.3 三角形モデル[3),4)]

三角形モデルでは、入力された座標点全てを地形データとして反映することができるので細かなデータを作成することができる。できあがるモデルは不定形／不規則な三角形の集合であるので、格子モデルでは表現することのできない詳細な状況をうまく再現できるという利点を持っている。また一般的にTINでは、各三角形が座標値へのポインタ、隣り合う三角形へのポインタをそれぞれ持っており、このためデータが大きくなる傾向にあるが、各種計算で高速な処理ができるような工夫がされている。このモデルの特徴は以下のとおりである。

・三角形分割には膨大な計算を要する
・地形の形状を正確に再現できる
・局所的に三角形の大きさを変えることにより緻密な表現が可能である
・サンプリングされたデータによりそのTINの性格が決まる

三角形モデルは格子モデルと特徴を比較した場合に、TIN作成のためにより計算時間を必要とするが、コンピュータの処理能力が向上した現在では、三角形モデルの方がより詳細なモデルを作成することができるので主流となりつつある。ＣＧデータの種類には多角形から自由曲面まで幾つかあるが、三角形ポリゴンは最もプリミティブなものであり、TINはそのままＣＧデータの三角形ポリゴンとして使える。以上により地形のＣＧ化のためにも三角形モデルが適していると考えられる。

2.4 地形モデルＣＧ化の問題点

ＣＧシステムでは、地形モデルをどのくらいの処理時間でレンダリングできるかということがまず問題となる。後述する二次元Delaunay三角形分割を用いると、生成されるTINに含まれる三角形ポリゴンの総数は、座標点の数をnとすると2n+1個になるので、仮に座標点が2万点とすると40,001個の三角形ができあがることになる。ＣＧの表示を高速で処理するGWS (Graphics Work Station) は、我々の使用しているもので1秒間に30万三角形ポリゴンをレンダリングできる性能を持っているので、数万個の三角形で形作られる地形モデルであればこの問題を意識する必要はない。

各種設計計算に用いるTINをそのままＣＧ用データとして使うためには、TINを生成する段階の処理の方が問題となる。格子モデルと三角形モデルのところで述べたように、TINで緻密なモデルを生成しようとするとデータサイズが大きくなる。データ量はそのままコンピュータへ負荷となってかかってくるから、ゴルフ場のように数万点にのぼる座標点を処理してTINを生成するのは容易なことではない。コンピュータの急激な高性能化に伴いこれらの問題は表面にでなくなってきたものの膨大な計算を必要とすることに変わりはない。この問題を如何に解決していくかが最も重要である。

ところで、ＣＧのアプリケーションでは通常色分けのために個々のデータをそれぞれの色毎にグループ化しなければならない。道路表面のデータは道路表面のグループに、舗道のデータは舗道のグループへという具合であるが、この作業はことのほか手間のかかるものとなっている。今までのTINではこの点が考慮されていな

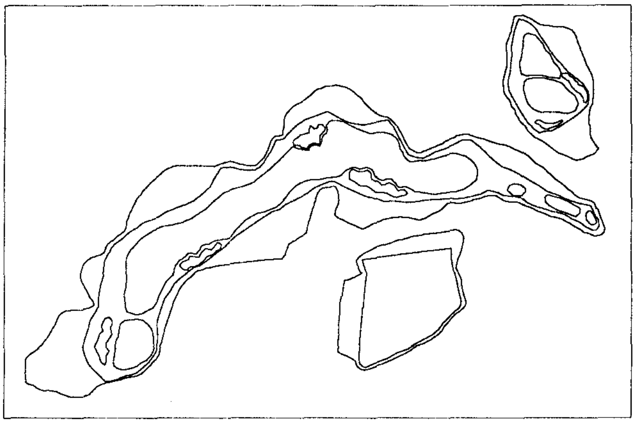
い。つまり、地形モデルＣＧ化のためには、TIN生成の高速化と三角形ポリゴンのグループ分けを簡便にすることが要求される。

以上まとめると地形モデルＣＧ化の問題点は、

・TIN生成法の高速化

・生成された三角形のグループ化

の２点であると考えられ、次節においてこれらの問題の解決策を提案する。



図ー２　ゴルフ場の境界線

3. 三角形分割

3.1 三角形分割のための基本データ

地形の形状は、地表上に散らばった座標点の集まりで表現することができるが、より正確に自然の状況を伝えるためには、さらに等高線や地性線が必要である。これらは幾何図形としてみると点そして線（任意個の節点を有する連続線）ということになる。点は単純に３次元空間内の任意の座標を表すものであるが、線はその線上に無限個の点を有するものということになる。しかしそれは抽象的な定義にすぎないので実際には節点の座標を取って座標点の集合として扱う。また線には閉じた線と開いた線がある。前者は対象とする２次元空間の一部を囲むようなものであって、その部分領域の境界をなすもの、後者は単に空間の一部を分断するような線である。以後では、前者を境界線、後者をブレークラインと呼ぶことにする。

境界線は通常図－２に一例を示すように、階層構造をなす。同図に示すように、ラフのなかにフェアウェイやグリーンやバンカーがあり、さらにフェアウェイのなかにバンカーがある場合がある。それぞれの境の線、つまり境界線は図－２のように階層構造になっていることが分かる。以上のことより境界については、外部境界、内部境界、第ｎ内部境界というように分けることにする。なお外部境界とは、三角形分割を行う領域を示し、三角形分割は必ずこの内部で行われる。内部境界とは、全ての第ｎ内部境界を包含する領域であり、いわば階層構造をなしている境界の最上位に位置する境界である。そして第ｎ内部境界で、その他の細かな境界を表現する。ここでｎは1以上で階層の深さを表す。

つまり、地形モデル化の基本データは、点（ここでは内部固定点と呼ぶ）、ブレークライン上の点、境界線上の点に限定し、これら諸点を用いて三角形分割を行う。なお、ブレークラインおよび境界線上の点を用いた三角形分割では、ブレークラインと境界線を構成する線分（その上の相隣り合う２点間を結ぶ線分）は必ず生成されなければならない。言い換えると、生成された三角形の辺には、これらブレークラインおよび境界線を構成する線分が含まれていなければならない。

3.2 三角形分割法とその問題点

多くのTIN作成のアルゴリズムは1930年代Delaunayによって提案された三角形分割（Delaunay triangulation）の理論に基づいている。このDelaunay三角形分割は２次元平面上に任意に設定された節点群を対象として、できあがる三角形のそれぞれの内角がなるべく等しくなるようにそれら節点の支配する凸領域を三角形分割するものである。この結果、得られる三角形は正三角形に近いものとなる。

前述したように、節点数が増加するにつれて三角形分割に要する演算時間が問題となる。S.W.Sloanはこの問題を解決するため、Watsonの提案した超三角形（Super triangle）、三角形分割を行う節点順序を最適化するためのBin Sorting、そしてDelaunayの意味での三角形生成を高速化できるスワッピング・アルゴリズム、さら

に空間内に新しく設定された節点の位置を捜し出す手法の導入を図り、高速のDelaunay三角形分割アルゴリズムの提案を行った[5]。しかし、彼の方法にも次に示すような大きな欠点を見つけることができる。

・幾何学的に複雑な凹凸のある境界に囲まれた任意２次元領域に対しては適用できない。すなわち、境界の凹部といった領域外部についても三角形分割を行ってしまう。
・複雑な境界を形作る境界上の辺を生成できない場合が発生する。
・多連結領域（穴のような内部境界を有する）の場合、三角形分割の不要な穴の内部についても三角形に分割してしまう。

以上示した欠点を取り除き、任意２次元領域内部だけをDelaunayの意味での三角形分割できるのが修正Delaunay三角形分割である[6,7]。この方法はその基本をSloanの提案した高速Delaunay三角形分割に置き、それに境界生成のアルゴリズムを追加したものであって、本研究の目的にあった方法と思われる。

今、修正Delaunay三角形分割を基にして本研究にあったアルゴリズムを作ろうとした場合、次のような問題点があることに気づく。以下においてこれらの問題点の解決策を考える。

(1) 問題点１：ブレークラインの生成

ブレークラインと境界との差異を調べると、後者ではその線が閉じるのに対し、前者ではそれが開いているだけである。よって、ブレークラインの生成には修正Delaunay法の境界生成のサブルーチンを修正すればよい。

(2) 問題点２：階層構造をなしている境界の生成と三角形分割

階層構造の境界とは境界が多重になっていることであって、それぞれの境界は閉じた線である。修正Delaunay法に含まれるよう境界生成法は、閉じた線を一個ずつ取り扱いそれを生成する。従って、境界が階層構造であっても既に提案されている境界生成法はそのままで適用できる。

(3) 問題点３：各領域毎の認識

この領域毎の認識は個々の領域（閉じた線で囲まれる領域）を異なった色に区分するためのものである。今、一つの境界を生成すると、既に生成されている領域の内部に新しい領域が発生する。もし、その新しい領域の内外が区別できれば領域の認識ができたことになる。この区別法は次節において述べるDelaunay三角形分割で生成された三角形が領域の内部に位置するか、それとも外部に位置するかを判断する三角形の分類法と同じである。よって、この問題も解決できる。

3.3 アルゴリズム

ここでは前節で述べた修正Delaunay三角形分割を基にし、さらに問題点１から３についての解決策を導入して、TIN作成アルゴリズムを示す。その流れの中心は、境界上とブレークライン上の節点、そして必要な領域内部の固定点を用いた三角形分割であって、対象領域の生成と同時にその領域の三角形分割を行う。まず、その流れを示す。

step1　データの入力
step2　節点の支配する空間のスケーリング
step3　超三角形の設定
step4　境界上の節点を用いた三角形分割（境界数繰り返す）
step5　ブレークライン上の節点を用いた三角形分割（ブレークライン数繰り返す）
step6　内部固定点のソーティング
step7　ソートされた内部固定点を用いた三角形分割（内部固定点数繰り返す）
step8　領域外部に位置する三角形の除去
step9　TINデータの出力

以下において上の流れのいくつかについて説明を行う。

(1) 入力データについて

入力データは次のようなものである。境界数、ブレークライン数、個々の境界を構成する節点数、個々のブレークラインを構成する節点数、内部固定点数（領域内に設定される他の点）、節点座標値および内部固定点座標値。

(2) 領域のスケーリングについて

設定された節点群の分布する範囲を標準化して超三角形の設定やBin sortingの処理を容易にするため、一辺の長さが1.0の正方形の内部に納まるように、スケーリングを行う。

(3) 超三角形の設定について

設定される節点群を包括するように超三角形を設定する。この超三角形の設定により、アルゴリズムが簡単になるという長所もある。

(4) Delaunay三角形分割の高速化について

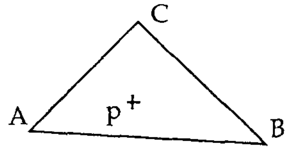
Delaunay三角形分割法は節点を順次空間に設定し、その点を用いて新しいDelaunay三角形分割を完成させる。節点数が増加するにつれ、この三角形分割に要する演算時間が増加するが、それを防止し演算時間の節約を図るため、次の諸技法を用いる。

i) Bin Sorting法

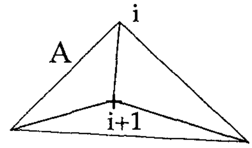
内部固定点や自動的に発生された節点を対象としてDelaunay分割の順番を決定するのがこのソーティングである。これは２次元空間を小さな碁盤目に分割して、(1,1)、(1,2)、...、(1,n)、(2,n)、(2,n-1)、...、(2,1)、(3,1)、(3,2)、...の順の正方形内部に位置する節点の順に三角形分割を行う。この順番はDelaunay分割に必要な演算時間を大幅に節約する。

ii) 点を含む三角形の探査法

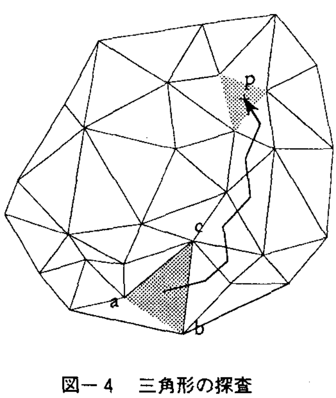
今、三角形ABCが点pをその内部に包含する場合、点pは三角形の三点の構成するベクトル$\overrightarrow{AB}$、$\overrightarrow{BC}$、$\overrightarrow{CA}$に対して常に左側に位置する（図－３参照）。この性質を利用して最も最近（第p-1番目の点を作成した段階で作られた三角形の一つ）作られた三角形abcから出発して、点pを内部に含む三角形を捜し出す方法である。この方法を用いると、ほとんど直線的に目的とする三角形に到達する（図－ 4）。



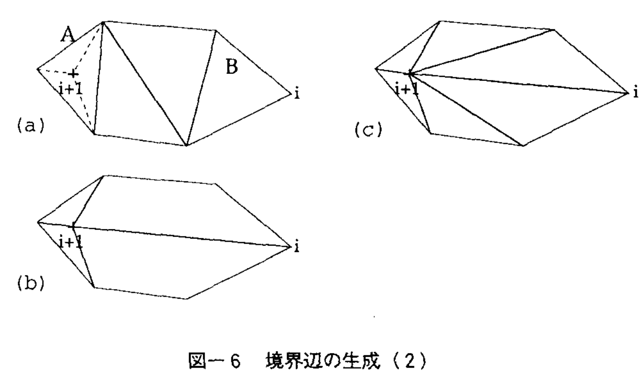
図－3　点pと辺の関係



図－5　境界辺の生成（1）



図－4　三角形の探査



図－6　境界辺の生成（2）

iii) スワッピング・アルゴリズム

Delaunay三角形分割を行うには、三角形の各内角がなるべく等しくなるように三角形を作らなければならない。この部分を局所的に行おうとするのがスワッピング・アルゴリズムである。

(5) 境界辺上の節点を用いた三角形分割について

一つ一つの境界をその上に位置する節点を空間内に設定し、修正Delaunay法を用いて三角形分割し、その辺を用いて表現する。境界上の節点の順序は時計回りに入力され、その順番に従って取り入れられ、三角形分割される。

図－５に示すように、n個の節点を有する境界辺上のi点までの境界辺が生成され、i+1番目の節点を空間に置いた場合を考える。もしi+1番目の節点を内部に含む三角形（Aで示す）の頂点の一つがiであれば、Aをi+1点を用いて3個の小三角形に分割すればiとi+1が結ばれ、境界辺の一部が生成できたことになる。

もしAの頂点がiでない場合（図－６参照）、三角形Aをまず3個の小三角形に分割し、さらにiを頂点とする三角形Bとの間に位置する全ての三角形を捜し出す。次に点iとi+1をつなぎ、多角形を2個の部分に分割し、最後にこれら2個の部分を三角形に分割する。この三角形分割では辺i-(i+1)を壊さないかぎりにおいてDelaunay三角形分割を行う。一つの境界生成の最後において、最後の点と第一点をつなぐように上と同様の操作を行う。

以上の操作を全ての境界上の節点について繰り返す。

(6) ブレークライン上の節点を用いた三角形分割

上に示した境界上の節点に対する分割と全く同様にしてこの三角形分割を行う。ただし、一つのブレークライン上の最後の節点と第一番目の節点をつなぐ操作は行わない。

(7)領域外部の三角形の除去について

修正Delaunay分割によって生成された三角形は次の6種類である。

a.超三角形の頂点の一つを含む

b.外部境界上の節点だけで作られる

c.少なくとも一個の内部点を含む

d.外部・内部境界上の節点を両方含む

e.いくつかの内部境界上の節点を含む

f.一個の内部境界上の点だけで構成される

ここで、領域外部に位置する三角形はa,b,fであり、領域内部に位置する三角形はb,c,d,e,fであるので、bとfに入る三角形で領域外部に位置する三角形を内部に位置するものから区別しなければならない。

いま、境界上の節点を時計回りに入力したとする。また、生成された三角形は反時計回り方向に格納された3個の節点で認識されるものとする。そうすると領域外部の三角形を構成する3個の節点と内部の三角形の3個の節点はその格納順序が逆であるので、この差から外部と内部の区別をすることができる。

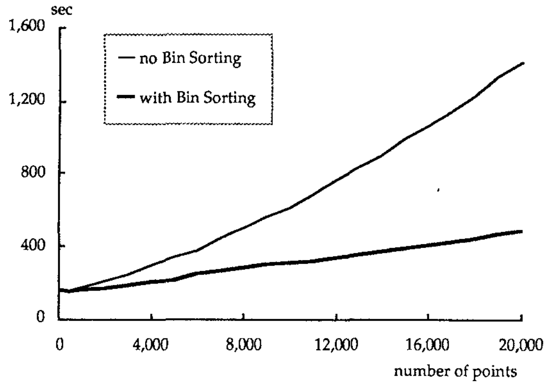
4. 適用例

4.1 処理時間の測定

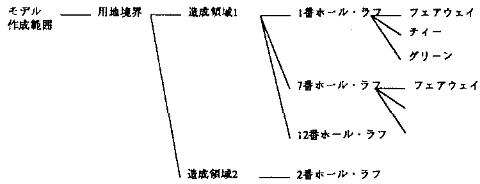
(1) 内部固定点による測定

座標値が(0,0)から(100,100)の範囲で乱数を発生させ、全て内部固定点として（ブレークライン、境界線はなし）点の数を変えて演算時間を測った結果は、図－７の通りである。このグラフよりBin sortingであらかじめソートしておくと大変効果的であり、演算時間は点の総数にほぼ比例していることが分かる。なお、使用したコンピュータはApollo DN4000である。

またSPARCstation II GX上で稼働している市販の３次元ＣＡＤでは、1万点のデータから約2分でTIN作成したが、今回開発したプログラムでは、同様な条件で1分以下で処理することができた。



図－７　内部固定点数と演算時間の関係



図－８　境界の設定（ゴルフ場）

(2) 境界線を設定したゴルフ場の例

処理時間に大きく影響を与えるのは、まず全ての座標値の数であるが、プログラムでは境界毎に三角形分割を行っているので、ゴルフ場のデータを元に境界線の状況を変えて実行してみた。座標値の総数は約1万5千点であり、その半分は境界上の点、残りの半分が内部固定点であった。なお、(1)の測定とはプログラムのコンパイルの条件が異なる。

a) ゴルフ場として全ての境界を適切に設定した場合241秒

b) 全てを固定点とした場合222秒

c) 27個のラフ線のみを境界線とした場合212秒

d) ゴルフ場用地の境界線をただ一つの境界線とした場合178秒

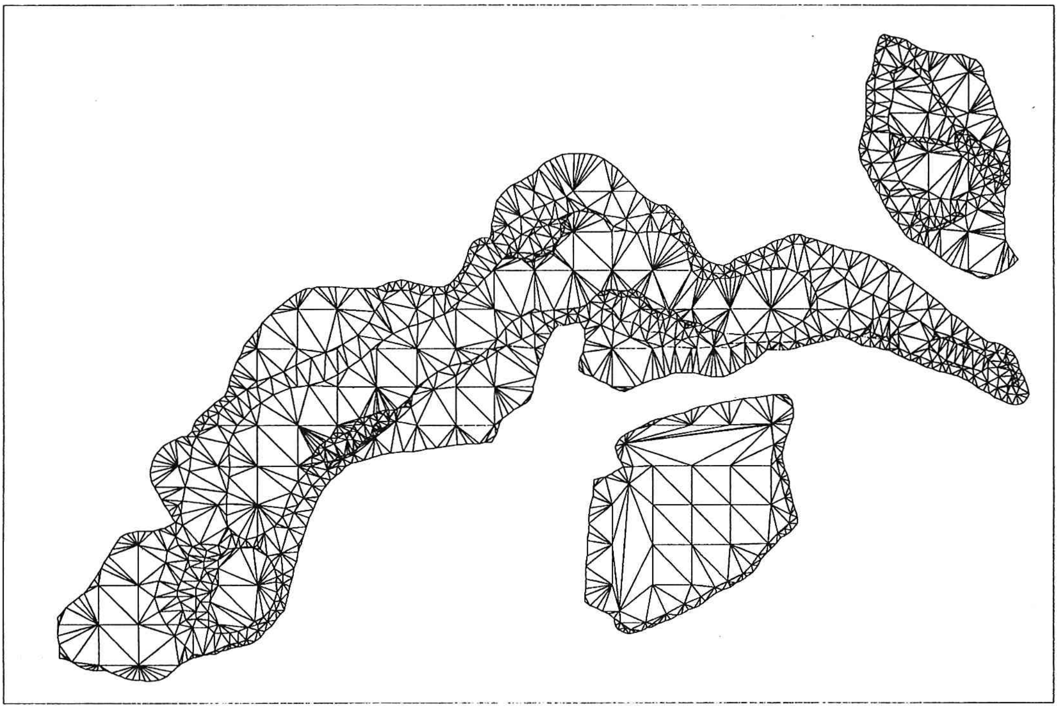
境界の大きさやその階層構造の状況が演算時間に影響しているのは確かであろうが、この結果を見たかぎり明確な傾向を見いだすことは困難である。ただし境界線上の節点は時計回りに次々に処理されるのでソーティングのような効果を出すであろうと考えられる。

4.2 ゴルフ場

ゴルフ場に見られる階層構造をなす各種の境界は図－８のように設定した。第一内部境界は工事によって造成が行われる領域を、第二内部境界は各ホールのラフの領域を、その他のティーやグリーンやバンカーあるいはフェアウェイの領域を第三内部境界に設定している。ここでテストデータとして用いているゴルフ場では、第一内部境界が17個、第二内部境界が27個、第三内部境界が130個となっている。なお現在使用しているＣＡＤでは地形モデルは格子モデルで行っているため、固定点のデータは計画地形DTMのグリッド上の点を用いており、約1万点であった。この三角形分割結果を図－９、１０に示す。

5. まとめ

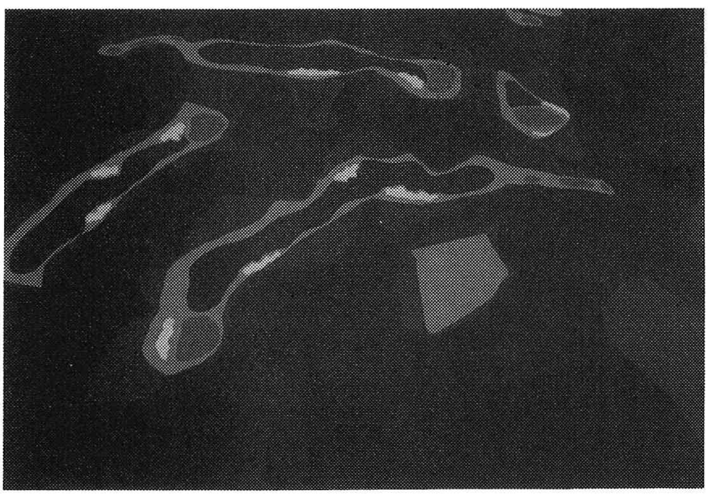
本研究では、２次元Delaunay三角形分割を拡張し様々な地形に対応して高速に、そして境界により作成される三角形のグループ分けを行う三角形分割を提案した。この方法により、点と線そして階層構造を持つ境界線のデータを基にＣＧ用地形モデルの作成が可能であることが確認できた。現在我々の使用している市販の３次元ＣＡＤシステムでは、TIN処理用のソフトウェアが付属しているものの、凹形境界の外部に三角形を生成する、ブレークラインの数が増えると計算時間が飛躍的にかかるなどいくつかの欠点を持っているため、ゴルフ場データのＣＧ化のために１週間以上の作業を必要としていた。しかし今回開発したプログラムを利用することで、即座にＣＧデータを生成することが可能となった。



図－9　ゴルフ場三角形分割結果（部分）

参考文献

1) Petrie, G and Kennie, T.J .M.: Terrain modelling in surveying and civil engineering, Vol. 19, pp. 171-187, 1987.5.
2) McCullagh, M.J.: Terrain and surface modelling systems: theory and practice, Photogrammetric Records, pp. 747-779, 1988.10.
3) De Floroani, Leila : Data structure for encoding triangulated irregular networks, Advances in Engineering Software, Vol. 9, No. 3, pp. 122-128, 1987.7.
4) Fowler, R.J. and Little, J.J. : Automatic extraction of irregular network digital terrain models, Proc. of SIGGRAPH '79, pp. 199-207, 1979.9.
5) Sloan, S.W.: A fast algorithm for computing Delaunay triangulations in the plane, Advances in Engineering Software, Vol.9, No.1, pp.34-55, 1987.



図－１０　レンダリング後の鳥瞰図

6) 谷口健男、太田親: 直線辺で構成される任意二次元領域へのデラウニー三角分割の適用, 土木学会論文集 , No. 432 / I-16, pp. 66-77, 1991.7.
7) Baker, T.J.: Tetrahedral mesh generations for the calculation of flows around complex configurations, Manuscript for the Second Nobeyama Workshop on Fluid Dynamics and Supercomputers, 1987.