

## 数値地図を用いた 3 次元街並み CG 映像の構築方法

藤原 士朗<sup>†</sup>      中沢 実<sup>†</sup>      服部 進実<sup>†</sup>

<sup>†</sup>金沢工業大学工学部情報工学科

〒921-8501 石川県石川郡野々市町扇が丘 7-1

Tel: 076-248-1100, E-mail: {shirou,nakazawa,hattori}@infor.kanazawa-it.ac.jp

近年、コンピュータグラフィックス(以下,CG)の技術の発展は目覚しく、我々の目にも触れる機会が多くなり身近なものとなった。CG は新たな映画やゲームなどの分野で大きく貢献している。しかし今後は、現在のような専門家が CG を作成するという状況から、写真が撮られるものから自ら撮影し楽しむものになったように、気軽に一個人が CG を作成する状況に変化していくものと考え。そのような観点から、現在の CG 作成には膨大な費用と時間、そして作成者の技術力が必要となる問題点がある。特に多くの建物が立ち並ぶ街並みを再現する場合、その問題点はさらに増大する。

本稿ではいかに手軽に素早く 3 次元街並みを構築できるかを目的に行ったものである。2 次元の数値地図から 3 次元の基本的なデータを作成し、テクスチャを動的に生成することでその要求に答える。また建物周囲の空間的スペースを認識することで、自動的に正面方向を判定する。

## The construction method of a street scene CG image using the numerical map

FUJIWARA SHIROU<sup>†</sup>    MINORU NAKAZAWA<sup>†</sup>    SHIMMI HATTORI<sup>†</sup>

<sup>†</sup>Department of Information Engineering, Kanazawa Institute of Technology

7-1 Ogigaoka, Nonoichi, Ishikawagun, Ishikawa 921-8501, Japan.

Tel: 076-248-1100, E-mail: {shirou,nakazawa,hattori}@infor.kanazawa-it.ac.jp

In recent years, technology of computer graphics is growing up quickly. Computer graphics are becoming familiar to us and also playing important role in field of movie, game and so on.

However, it will change to the situation that not only specialized creator but also an end user CG. In this case, there is a problem that huge expense and time, and technical power are needed in the present CG creation. Especially, when reproducing street scene including in many buildings, the problem increases complexity.

In this paper, it realizes the ability of 3D map from 2D numerical map and texture dynamically. Aromatic judgment system of the front has been realized by the recognizer of the spatial space of the circumference of a building.

## 1. 始めに

近年、コンピュータグラフィックス(以下,CG)の技術の発展は目覚しく、我々の目にも触れる機会が多くなり身近なものとなった。CGは新たな映画やゲームなどの分野で大きく貢献している。3次元表示による説得力については誰も疑う余地の無いものである。今後もCGの活躍分野というのはさらに増えるであろう。ここで現在のCGの世界を観察してみると、個人が自分のためにCGを作成するというはそれほど多くはない。しかし、写真が撮影してもらおうという受動的な立場から、自ら撮影し楽しむものになったように、現在、専門家がCGを作成するという状況から、気軽に一個人が自分のためにCGを作成する状況に変化していきだろ。そのような観点から考え、現在のCG作成には膨大な費用と時間、そして作成者の技術力が必要という問題がある。それを解決する技術として、個々のオブジェクトを写真から作成するイメージ・ベースド・モデリング[1]は、個人がCGを作成するには非常に有効な技術である。3次元・モデラーへの見識がそれほど無くとも、求める3次元オブジェクトが作成できる。しかし、多くの建物が立ち並ぶ街並みを再現するにはオブジェクトを個別に作成しては多くの手間・時間が必要となる。街並み作成の場合は一つ一つのオブジェクトの完成度よりも、どれだけ大量のオブジェクトを簡単に作成できるかが問題となる。

現在、数値地図から3次元の街並みを生成する技術として、数値地図と航空写真を用いてテクスチャまで動的に生成するもの[2]、数値地図のみを用いて3次元を生成するもの[3]が代表的な手法である。航空写真を用いる場合、確かにその完成度は向上するが作成にかかる時間の大幅な減少は望めず、手軽とは言い難い。また、作成を行う数値地図の場所に適合する航空写真が必要となり、データの入手方法も決して簡易とは言えない。ただ、国土地理院が平成14年4月からインターネットで空中写真の閲覧サービスを開始するため、この手法が有効になる可能性も出てきてはいる。しかしながら、手軽さ・速度を求める場合にはこの手法は適さない。またもう一つの方法である数値地図のみからの作成では、テクスチャをどのように用意するのが大きな問題となる。以上のような問題点を考慮しながら本研究では手軽さ・速度を目標に研究を行うこと

にする。よって航空写真は用いず3次元の街並み生成を行い、テクスチャは動的に生成することで対応する。

## 2. システム構成・処理の流れ

本研究でのシステム構成と処理の流れを図1に示す。本研究では数値地図データを扱うために、GeoBaseと呼ばれるGISに必要な各種の機能が凝縮された32ビットWINDOWS DLLのツールライブラリを用いる。その概要図は次項にて簡単に説明する。

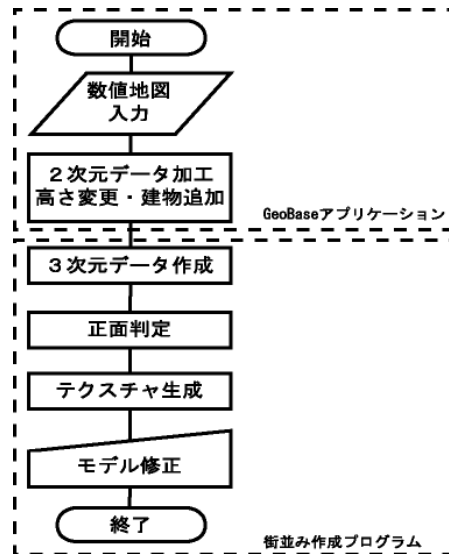


図1: システム構成・処理の流れ

始めに2次元数値地図からデータを読みこむ。この数値地図はこの数値地図はGeoBaseが読み込める形式のもので無ければならない。その後、読み込んだ図形群に対し高さ変更及び新たなオブジェクトの追加を行う。この段階でもGeoBaseの機能を活かして仮の3次元表示が可能である。その後高さ情報を持った2次元データを書き出し、別プログラムへ処理を移す。

街並み作成プログラムでは、高さ情報と2次元データから3次元ポリゴンデータを生成し、そのオブジェクト間の距離を計算することで正面・側面を決定する。そして自動的にテクスチャを生成・貼り付けを行う。その後個々のオブジェクトに対し手入力による修正を施し、終了となる。また本研究では表示にOpenGLを用いている。

## 3. GeoBase アプリケーション

このアプリケーションに用いたGeoBaseのイメージを図2に示す。このアプリケーションでは数値地図の読み込み、建物オブジェクトの追加・削除、及び高

さの変更が行える。また、3次元に加工するとどのような概観になるのかを、プリミティブなオブジェクトの状態でのプレビューできる。その後、範囲を指定し2次元のベクトルデータを書き出す。

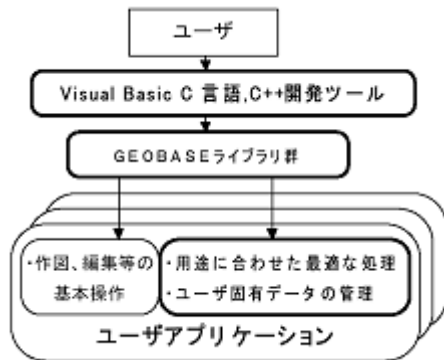


図 2 : GeoBase イメージ図

#### 4. 2次元データから3次元データ生成

GeoBase アプリケーションによって吐き出された2次元データから3次元データを生成する手法を述べる。

##### 4.1 側面ポリゴンの生成

本研究では建物のオブジェクトの側面の形状は平らであると限定しているため、側面ポリゴンの生成は上面に比べ難易度は低く非常に容易で、面を上につ引っ張るようなイメージそのままで作成できる。

ある一辺の2つの頂点座標を  $(x1, y1)$   $(x2, y2)$ 、高さを  $H$  とすると、その側面の三角ポリゴンは次の図のようになる。

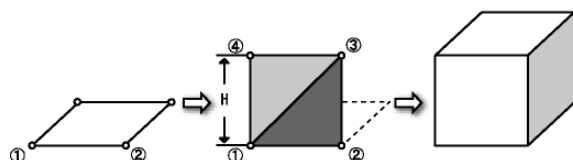


図 3 : 立ち上げによる面の作成

、以外の座標は  $(x2, y2 + H)$   $(x1, y2 + H)$  となる。つまり側面の三角ポリゴンは「 $(x1, y1)$ 、 $(x2, y1)$ 、 $(x2, y2 + H)$ 」の頂点を結ぶことで作成できる。この方法を他の3辺にも適用することで3次元な立方体の座標が手に入れられる。

##### 4.1 側面ポリゴンの生成

上面の形状は、単純で無い多角形の場合が多いため、側面に比べ三角形分割が複雑になる。多角形の分割アルゴリズムには「ポロノイ多角形とドロネー三角形分割」[4]、「デローニ三角形分割」[5]などがあるが、今

回は描画プログラムに OpenGL を使用しているということからも、OpenGL のライブラリの一つである GLU ライブラリのタイリングルーチン[6]を利用した。

#### 5. 正面判定

本研究では自動的に正面を判定することで様々な自動処理の実現を図る。その判定には周囲の建物との距離、つまり距離が遠い面を正面とする条件と、面の大きさを用いる。距離計算には直方体のバウンディングボックスを用いて行う。作成したバウンディングボックスの4辺それぞれに対し、縦の長さが一定の長方形を作成し、その長方形に対して他辺との交点・内側に含まれる点を他の全ての建物オブジェクトに対し計算する。その値を元に各オブジェクトとの距離を算出し、正面判定に利用する。次節においてバウンディングボックスについて述べる。

##### 5.1 バウンディングボックス

バウンディングボックスとはオブジェクトを囲む衝突判定領域のことである。最も単純な作成方法は、あるオブジェクトの最大・最小の X, Y 座標を用いるもので、非常に簡単に作成できる反面、判定が曖昧になるという欠点がある。例えば図4左では右下に大きな空

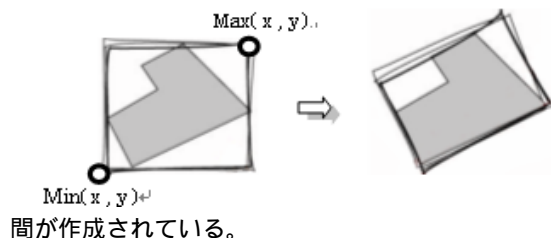


図 4 : バウンディングボックス

よりオブジェクトにフィットした図4右の様なバウンディングボックス作成のアルゴリズムを図5に示す。

この手法の特徴は、先に述べた最大・最小値から求める手法では X・Y 軸に沿った辺を作成されるのに対し、始めの一辺をそのオブジェクトを象徴するような最も大きな辺と同じ傾きで作成し、それを基準に次からの辺を作成することにある。次に、作成された辺が縦・横の方向について調べる。そして縦であるなら上向き・下向きを、横であるなら右向き・左向きかをチェックする。これは次に作成する辺がどの点を通れば、オブジェクトを包む辺ができるのかを判断する為のも

のである。このアルゴリズムを適用することにより  
図4右のようなバウンディングボックスが作成できる。

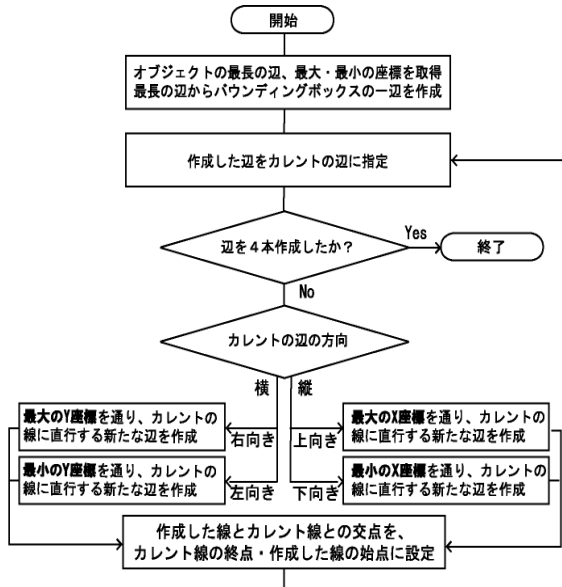


図 5：バウンディングボックス作成アルゴリズム

## 5.2 建物周囲のスペースの判定

スペース判定のために前述したバウンディングボックスに密着した長方形を作成する。その長方形と、他の建物オブジェクトのバウンディングボックスとの交点、及び長方形内側にあるバウンディングボックスの頂点（以下内包点）を求め、その求められた点までの距離が、あるオブジェクトの面から他のオブジェクトまでの距離となる。図6が実際の画面での交点・内包点を求めた結果である。

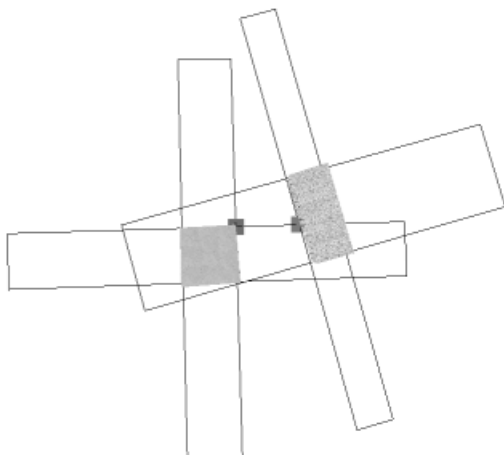


図 6：密着した長方形による距離計算

図6は二つの建物オブジェクトについて計算してい

る。中央にある四角いオブジェクトの周囲にある長方形と、他のオブジェクトとの交点・内包点が求められているのわかる。図6の建物オブジェクトはバウンディングボックスとほぼ同じ形状であるため、点は建物オブジェクトの辺上にポイントされている。右側の建物オブジェクトから左に伸びている長方形では内包点がポイントされている。ある面で複数の交点・内包点がある場合、もっとも距離が近い点を採用する。その点がある面の正面スペースとして用いられる。この処理をバウンディングボックス4面全てに施す。求めた各面の交点に対する正面決定条件を以下に示す。

- 交点が存在しない場合  
面の大きさにより決定する。
- 交点がある場合  
交点がある面を正面とする。
- 交点がある場合  
複数の交点のうち、交点までの距離が最も遠い面を正面とする。

続いて、交点を求めるために必要な周囲に伸びた長方形の作成方法を図7を参考にしながら説明する。

建物オブジェクトの辺から伸びる長方形は一定の長さ・オブジェクトに適した傾きでなければならない。その作成方法であるが、まずあらかじめ決めておいた長さの正方形を作成する。次にその正方形の2辺を、対象となる建物オブジェクトの面の長さに変形する。その後、面とX軸との角度を内積を利用して求め、その角度だけ変形した長方形を回転させる。そして作成された長方形を密着させたい面の座標に移動させる。以上の処理により、建物オブジェクトのある面に密着した一定の距離の長方形が作成できる。

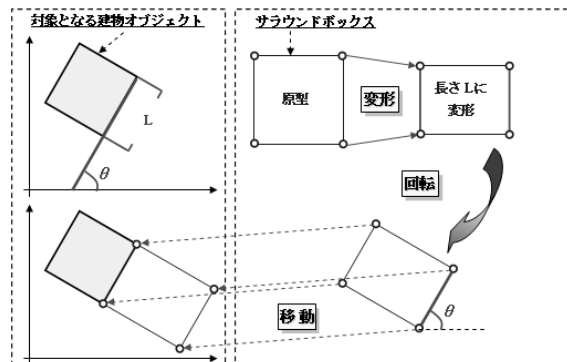


図 7：密着した長方形作成概要図

作成した長方形と、他の全ての建物オブジェクトに対する交点・内包点を求めることで正面判定が行える。

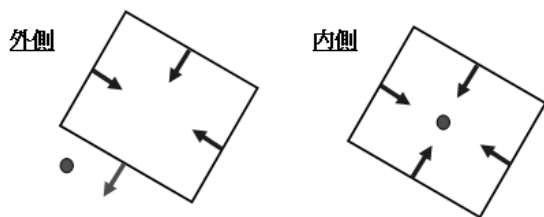


図 8 : 点と辺の位置関係による判定例

ある四角形の中に点が内包されるかどうかを判定するために、外積を用いて点が四角形の全ての辺に対し同じ方向（今回の長方形は左回りに定義されているので左側）に位置するかを調べる。全てが同じ方向であれば内包され、一つでも異なる方向があれば外側にあるということになる。

### 5.3 テクスチャ生成

数値地図では様々な大きさ・高さの建物オブジェクトが存在し、かつ動的に高さの変更も行えるため、あらかじめテクスチャを用意することができない。そこで、多様な建物オブジェクトに対応する為に数多くのテクスチャを動的に生成する必要がある。今回はビルに対象を絞った手法を述べる。

ビル壁面のデザインは単調なパターンで形成されている。それは現実の建物を見ても明らかである。そこで壁・窓を別々に用意し、壁に窓を重ねるように合成することで「壁の種類×窓の種類」個の壁面パターンが作成できる。図 9 はその用意されたテクスチャ画像の一部である。

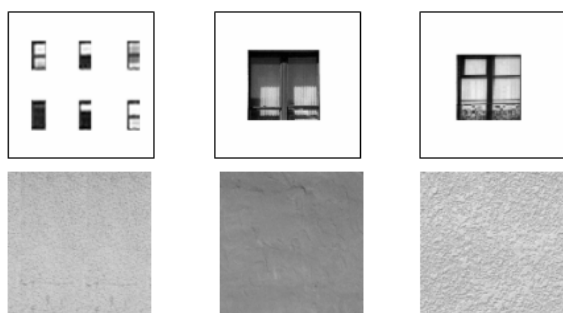


図 9 : 壁・窓のテクスチャ画像(128×128)

窓の白い背景部分を切り抜いて壁と合成することで、一箇所のパターンが作成できる。これを面の長さ・高さに合わせて繰り返す回数を変更することで、ビルの高さにあったテクスチャを張ることが可能となる。テクスチャの繰り返し回数の変更は OpenGL を用い

て行った。図 10 はその実行例である。

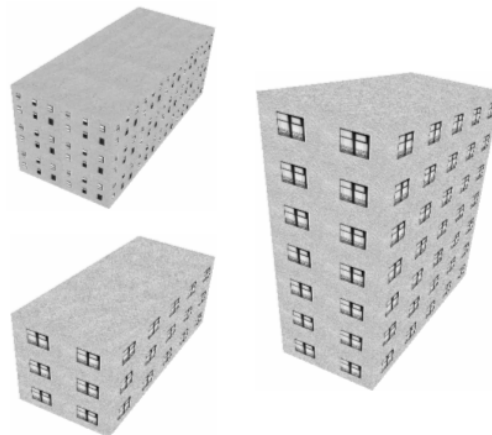


図 10 : テクスチャパターン変更の実行例

## 6. 実行結果

### 6.1 実験環境

本システム作成環境は DELL 社の Dimension4300 を用いた。CPU は Pentium 1.7GHz , Memory は 256MB , OS は Windows 2000 を使用し、 Visual C++ 6.0 でビルドした。3次元グラフィクス表示ライブラリに OpenGL を使用している。

### 6.2 実行結果

図 11,12 は同じ数値地図から作成したデータを描画したものである。テクスチャ自動生成の手法により、このような多くの建物オブジェクトが存在する空間を非常に手軽に作成することができた。また図 13 では正面判定を行った結果を形式的に描画した図である。

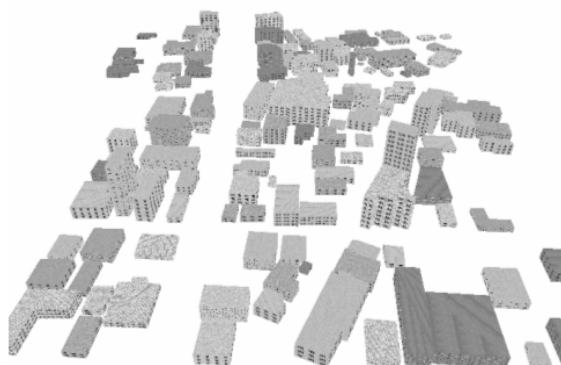


図 11 : 全体図

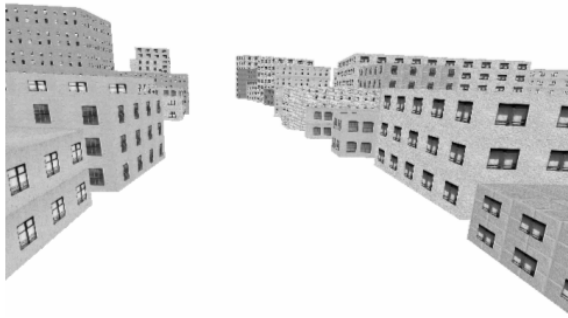


図 12：ウォークスルーに近い構図

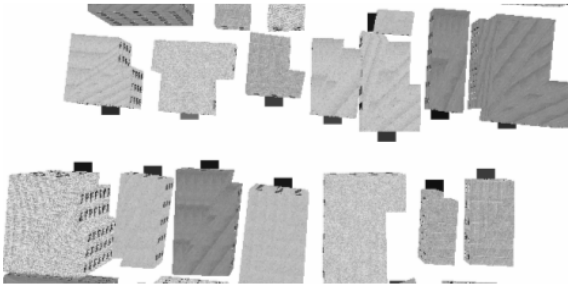


図 13：正面判定結果

図 13 において、四角いポイントがされている面が正面と判定されている。ある一つの建物オブジェクトの正面の判定が他のオブジェクトとは異なっているが、その原因は図欄外に他の建物オブジェクトが存在し、それとの交点の方が距離が遠い、つまり空間的スペースが大きかったため、正面の判定が横に立ち並ぶ建物オブジェクトとは異なっている。しかし、その他はほぼ理想通りの結果となっている。このことから今回の正面判定の手法は実用性があるといえる。さらにアルゴリズムを追加することで、今回正面方向が一つ異なっていることも解消できるだろう。

## 6. まとめ

本研究では数値地図から、素早く・手軽に 3 次元空間を構築する手法についての検討を行った。街並みを構成するだけのオブジェクトを手作業によって構築するには大きなコストが必要である。しかしながら近年は 3 次元チャット、ゲーム等仮想空間を作成する機会が高まっており、また冒頭に述べたような個人が CG を作成する流れも来るだろう。それに応えるべく本研究では、テクスチャの動的な生成による多数の建物オブジェクトの自動生成、入り口や

道路作成に応用できる正面の判定などを行った。しかし、現時点では建物オブジェクトとして表現できるのはビルのみであり、家屋、商店といったリアルな街並みを構成する上では必要なものが生成できていない。また計算により算出された正面のデータを活用することができなかった。正面判定のデータは数多くの用途に利用できる可能性があり、今後に残る課題の中でも最も重要なものとなると考える。また今回はあえて簡単な情報のみの数値地図を用いた。これは将来的には数値地図及び GeoBase を用いずとも街並みを作成可能なシステムを考えていることに起因する。そうすることで手軽に 3 次元空間が構築できると考える。

## 謝辞

本研究は、通信・放送機構の地域提案型研究開発制度の支援を受けて実施された。ここに記して謝意を表す。

## 参考文献

- [1] 除 剛、写真から作る 3 次元 CG - イメージ・ベースド・モデリング&レンダリング - 2001
- [2] 3 次元 GIS と消防支援システムへの利用  
[http://www.isad.or.jp/magzin/61\\_3zigen.htm](http://www.isad.or.jp/magzin/61_3zigen.htm)
- [3] 杉原 健一、GIS と CG の統合化による 3 次元街並み自動生成システム  
[http://www.gis.pasco.co.jp/community/doc\\_lib/uc2000/user\\_pdf/p2000\\_4..pdf](http://www.gis.pasco.co.jp/community/doc_lib/uc2000/user_pdf/p2000_4..pdf)
- [4] ボロノイ多角形とドロネー三角形について、  
<http://www.alles.or.jp/~mameo/kaeru/FrogCells.html>
- [5] デローニ三角形分割  
<http://www.roy.hi-ho.ne.jp/pastel/home/VC/delauney.html>
- [6] Mark J.Kigard, OpenGL Programming for the X Window System(日本語版) 1997