

都市景観システムのための入力インタフェースの実現

- 建物の入力 -

加藤隆博 平本建志 *間瀬実郎 河合和久 大岩元 *瀬口哲夫
(豊橋技術科学大学 情報工学系・*建設工学系)

対話形式における入力の効率化に重点をおいた、都市景観システムのための建物情報入力インタフェースについて述べる。本稿では効率化の手段として、標準化モデルの導入と、計算機による局所的な図形認識とを用いた。標準化モデルの導入により、入力すべき情報を最低限必要なもののみとし、オペレータの入力作業の簡素化を図った。また、局所的な図形認識をおこなうことによって、地図から建物情報を取り出す部分を計算機にゆだね、オペレータの負担を軽減した。なお、システムの開発にはオブジェクト指向を用い、拡張性に優れたシステムを構築している。

IMPLEMENTATION OF AN INPUT INTERFACE FOR A TOWNSCAPE SIMULATION SYSTEM
- INPUT OF INFORMATION FOR BUILDINGS -

Takahiro Kato Takeshi Hiramoto *Jitsuro Mase
Kazuhisa Kawai Hajime Ohiwa *Tetsuo Seguchi

Dept. of Information and Computer Sciences,
*Dept. of Regional Planning
Toyohashi University of Technology

1-1, Hibirigaoka, Tempaku-cho, Toyohashi-shi, Aichi 441, Japan

We propose an interactive building data input method for the townscape simulation and design. This method employs two concepts named as the "standard input model" and "local recognition". Using the "standard input model" means only the least necessary amount of information is used, thus the inputting task becomes much simpler. And, using the "local recognition" means picking up 2-D information of a building from the map by computer, which reduces operator's workload. As a result, we can get an efficient input work.

1. はじめに

我々は、都市開発を支援する都市景観システムの開発を進めている。都市景観システムとは、都市の3次元情報である建物・地形などの情報を扱い、コンピュータ・グラフィックスを用いて景観シミュレーションを可能とするシステムで、都市の空間的な性質を評価、設計することを目的としている。本稿では、この3次元情報を、地図から得られる2次元情報に視覚的情報を付加する方法により、効率的に入力する方式について述べる。

地図からの2次元情報の入力手法として、地図情報の自動認識手法が各方面で研究されている。⁽¹⁾⁽²⁾など) それと同時に航空写真などから視覚的情報を自動認識する手段があれば、それらを用いて自動的に都市を構築することも可能である。

しかし、現時点での完全な自動認識手法は実用性に乏しく、実際の地図情報入力には人間の手を必要としているのが実状である。そのため、最近では人間と計算機との対話形式によって入力する方法が効果的であるという考え³⁾もでてきている。また、計算機によって航空写真から視覚的情報を認識させることも非常に困難である。そのため、視覚的情報も一つ一つ手入力しなければならない。これらの理由より、その入力方法として対話形式を用いた。

ところが視覚的情報というのは情報量が非常に多く、それゆえに入力方法も複雑なものが要求される。その結果、入力に費やす手間と時間は膨大なものとなる。そこで我々は、これら入力に重点を置き、快適で効率の良い入力インタフェースを構築することを目的とした。なお、本研究では、その一部である建物情報の入力インタフェースについて述べる。

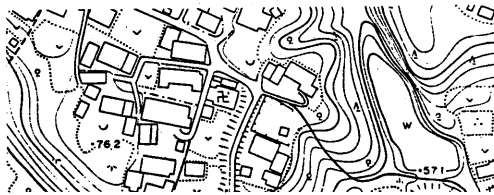


図-1 1/2500の都市計画基本図

2. システムの概要

本システムは、イメージスキャナから読み込んだ地図画像に対して、3次元情報である建物や地形(標高値)などの「物体」を入力し、その入力結果より都市景観画像を生成するものである。3次元情報入力のベースとなる地図には、家形が表記してあるような大縮尺の地図を用いる。(図-1) また、3次元視覚化は入力途中でも実行することができ、その入力状態を把握することが可能である。

すでに述べたように、地図からの情報のみでは都市景観、特に建物の形状は再現できない。よって、その入力用の原データとして、地図に加えて航空写真もしくは実地調査の情報を用いる。

なお本システムは Sun Microsystems 社製のエンジニアリングワークステーション (EWS) SUN 3/260 上で構築されている。この EWS は 32 M バイトの実装メモリを持ち、その入出力装置として、キーボード、光学式 3 ボタンマウス、1670 万色中 256 色同時発色のディスプレイを持っている。また、ウィンドウシステムとグラフィックス機能には X-Window⁴⁾ を用いた。

本研究では、開発言語としてオブジェクト指向型言語である C++⁵⁾ を用いた。C++ は、従来の C 言語の機能を失わずに、オブジェクト指向パラダイムの特徴である、モデルの階層表現、情報隠ぺい等の実現を可能としている。本システムはこのオブジェクト指向パラダイムを用いて、拡張性のあるシステムを実現するよう留意した。

3. 建物の入力

3.1 建物の入力とは

ここで、建物の入力というのは、ラスタ画像である地図に対して、視覚的な情報を加えながら、その家形をベクトル化する作業のことをいう。今回の研究の目的は、この入力作業をいかに効率的におこなうかである。

3.2 人間と計算機役割分担

入力を効率的におこなうには、その作業行程をできるだけ計算機にまかせてしまうことである。できることならば、すべての処理を計算機にまかせてしまいたい。しかし、それは現時点では非常に困難であるので、以下の方法を用いた。

地図からの情報の入力には各研究にもみられるように、ある程度の情報であれば計算機にとって認識しやすいものである。そこで、地図の局所的な図形認識を計算機におこなわせ、航空写真からの情報を人間が読み取ってそれに加える方法を用いた。要するに、オペレータは航空写真などの資料を片手に、画面上の地図に向かって入力処理をおこなうのである。このように、人間と計算機とが互いの図形認識能力を活用し、役割分担をおこなうことで入力の効率化を図った。

地図上から家形を探す方法として、地図情報から自動認識する研究が進められている²⁾。これを用いて、すべての家形を抽出した後、視覚的情報を入力する方法が考えられる。しかし、現時点では家形を完全に抽出することはできないため、後処理が必要となる。よって、視覚的情報を入力するために必要となる手間を考えれば、後処理をしてまで家形の自動認識をおこなうよりは、一つ一つ入力していく方が入力対象が明確になり、有効であると考えた。その結果、本システムでは、図形認識そのものは計算機がおこなうのだが、地図上から家形を探し出し、その位置を指定するのは人間の役割としている。

3.3 標準化モデル

建物を忠実に再現するためには、膨大な情報量が必要となる。よって、その入力には膨大な手間と時間と記憶容量が必要となる。しかし、本システムの目的とするのは、都市の空間的な性質の評価であるため、建物一棟一棟の詳細な情報は必要ないものとする。そこで、建物に必要な情報を簡略化した標準化モデルを設けることにより、入力の効率化を図った。

標準化モデルを導入するにあたり、次の二つの

点に着目した。

- ・本システムの目的は都市の空間的な性質の評価であるため、建物の空間的な役割が評価できなければならない。

- ・標準化モデルの目的は入力の効率化であるので、その情報量は必要最低限のものにとどめなければならない。

これらの条件を踏まえた上で以下のようなモデルを提案した。(図-2)

(1) 地図上の位置

地図上のどこにその建物が存在するかをその地図の左上を原点とした座標で表現する。

(2) 外形

地図上での家形の形状情報である。その頂点群の座標で表現される。

(3) 高さ

実際の高さでなく、階数で表現される。建物は、外形をそのまま上に引き上げたものの上に、屋根をつけた形で表現される。

(4) 屋根の形状

屋根の種類と、棟の方向とで表現される。屋根の種類には、陸屋根・切妻・寄棟・入母屋・片流・ポルト・ドームを用意した。棟方向は外形の形状に依存し、屋根は棟方向を向き、外形を包括する長方形で表現される。

(5) 開口部

ドアや窓の情報である。

(6) 色

壁や屋根の色である。3次元視覚化における基本色となる。これより、光線の加減によって表示の際の色を算出する。

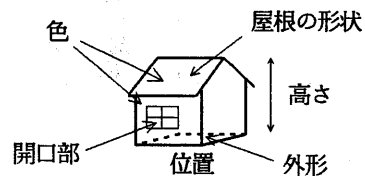


図-2 標準化モデル

これらを用いてシミュレーションをおこない、それについて建築の専門家を対象としたアンケート

- 1) 認識の対象となる家形領域を塗りつぶす
- 2) 塗りつぶした結果の輪郭線を抽出する
- 3) その輪郭線をベクトル化することにより、頂点を抽出し、多角形で近似する

この処理をおこなうために人間のすべきことは、地図から家形を探し出し、認識の対象となる家形領域の内部をマウスでクリックしてやることと、1)の塗りつぶし結果が正常であるかを確認することである。

家形領域を抽出するために塗りつぶしを用いているため、エラーとして塗り漏れが考えられる。(図-4)もちろん、エラーは入力妨げになるため、おこるべきものではない。しかし、元となるデータがイメージスキャナから読み込んだ画像である以上、明確な画像は期待できない。さらに、今回用いたような大縮尺の地図はもとから鮮明ではない。よってこのような画像では、1、2ドットの間隙は多分に生じうる。そして、通常の計算機における塗りつぶし方法ではこのような隙間からでも塗り漏れが生じるのである。

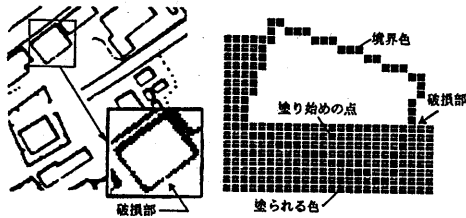


図-4 塗り漏れによるエラー

破損部はオペレータが発見して修繕する必要がある。しかし、このような小さな破損を人間が発見するには、画面を凝視する必要がある。これは人間にとって決して快い作業ではない。そこで、このような小さな破損が存在しても正常に塗りつぶしができるような方法として、凝結ペイント⁶⁾を導入した。これは2×2ないし3×3のマスクをかけて疑似的に境界線を太くし、破損を塞いでしまうものである。これを用いて入力実験をおこなった結果、307棟の家形に対して33棟(11%)あった塗り漏れが5棟(2%)にまで抑えられた。また、残りの5棟のような凝結ペイント

を用いても正常に塗りつぶしができないレベルの破損は、比較的発見しやすいものであるので、オペレータの負担はかなり軽減される。

しかし、この凝結ペイントにも欠点がある。それは、破損に対して圧倒的に大きなマスクを用いて塗りつぶしをおこなうと、その破損部から色のはみ出し、「おでき」状になってしまう(図-5)ことである。その結果として、頂点の抽出に支障がでてしまう。これはマスクの大きさを変えることによって対応できる。しかし、凝結ペイントを改良し、このような症状がでないようにするのも今後の課題のひとつである。

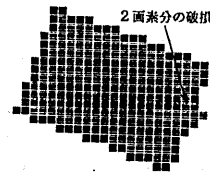


図-5 凝結ペイントにおけるエラー

3.7 マルチレイヤ構造を用いた建物情報の地図への付加

画面上に表示されている地図はマルチレイヤ構造を用いていて、必要に応じて他の情報を重ねて表示することができる。建物は入力すると、塗りつぶしに用いた色情報の他に、この機能を用いて地図にアクソメ図を重ねて表示することができる。これは建物の外形をそのまま上へ引っ張ったものに屋根の骨組みをつけた、極めて単純な立体図である。これによってオペレータはさらに屋根の形状を画面から入手することができる。また、ある程度の都市の状況を把握することも可能である。

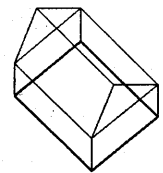


図-6 アクソメ図

4. 3次元視覚化

4.1 オブジェクト指向を用いた3次元視覚化本システムにおける3次元視覚化には、オブジ

ェクト指向による利点⁷⁾が多分に織り込まれている。ここで、それらの有効性を3次元視覚化のアルゴリズムにそって紹介する。

3次元視覚化のためにオペレータがおこなうべき作業は次のとおりである。

- 1) 視点の位置を決める
- 2) 光線の向きを決める
- 3) スクリーンの位置を決める
- 4) 3次元視覚化を計算機に指示する

ここで、光線には平行光線を使用し、光源は無制限に存在するものと仮定する。また、スクリーンの大きさは固定であり、スクリーンに写る範囲は、視点からスクリーンまでの距離によって調節するものとする。

オペレータから視覚化の指示がでると、視覚化を管理するオブジェクトが物体を管理するオブジェクトに視覚化の命令をだす。すると、物体を管理するオブジェクトはすべての物体に対して視覚化を命令する。この命令を受け取った物体は、自分が3次元上でどのようなポリゴン(多角形の板)の組み合わせで表現されるかを算出し、それらのポリゴンをポリゴンを管理するオブジェクトに渡す。

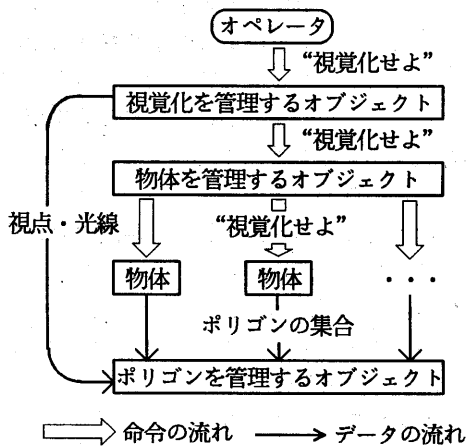


図-7 3次元視覚化(その1)

ここまでの処理を図-7に示す。

このようにポリゴン化を物体自身がおこなっているため、物体を管理するオブジェクトは物体がこの物体は2階建ての建物だからこのようにポリ

ゴン化して、というような配慮をする必要がまったくない。それどころか、物体が景観を形成するものであることさえ知っていれば、その物体が何であるかを知る必要さえないのである。これは、新しい種類の物体が増えても、物体を管理するオブジェクトはその視覚化のための処理を変更する必要がないことを意味している。

すべての物体のポリゴン化が終了すると、ポリゴンを管理するオブジェクトはポリゴン個々に、視点・光線・スクリーンの情報を与え、投映の命令をだす。するとポリゴンは、これらの情報より、自身がスクリーンにどの色でどんな形で投映されるかを算出し、自らを座標変換して、スクリーン

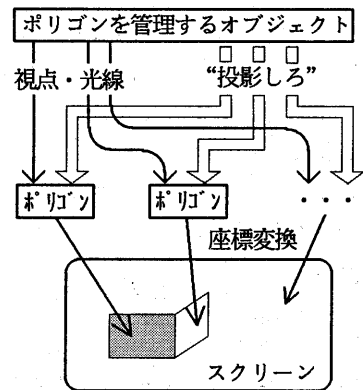


図-8 3次元視覚化(その2)

に投映する。このようすを図-8に示す。

物体のポリゴン化と同様に、投映の処理をポリゴン自身がおこなうので、ポリゴンを管理するオブジェクトはそのポリゴンの性質を知る必要がない。よって、もし特殊な投映法を必要とするポリゴンがあっても、ポリゴンを管理するオブジェクトは、それを配慮する必要はない。

4.2 ポリゴンにおけるインヘリタンス

ここでは、特殊なポリゴンにはどのようなものがあるか、それにどう対応しているかを述べる。

実際のポリゴンは、図-9のようなクラス構成になっている。この図で、上にあるのは基本クラスで、その下につながっているのはそのクラスの導

出クラスである。導出クラスは、基本クラスの情報・機能をそのまま受け継ぎ、それに独自の情報・機能を追加したものである。これらをクラスの継承と呼び、導出クラスが基本クラスの性質を受け継ぐことをインヘリタンスと呼ぶ。

ポリゴンの共通の情報である、頂点数と各頂点の座標はすべてのポリゴンの基本クラスである「ポリゴン」が持つ。それ以下の導出クラスは、それぞれ特有の処理をおこないたいがために存在する。

「建物ポリゴン」は、建物をポリゴン化する時に発生するすべてのポリゴンの総称である。実際には、建物は屋根を「建物ポリゴン」で、壁を「壁ポリゴン」で構成する。壁ポリゴンが建物ポリゴンの導出クラスとなっている理由は、壁には開口部が存在するからである。壁ポリゴンは自らをスクリーンに投射するとき、開口部をあたかも壁の模様のように描画する。実際には開口部は小さなポリゴンで表現されており、その管理は壁ポリゴンがおこなっている。

開発初期段階では開口部は扱わず、このクラスも存在しなかった。しかし、開口部がないと町並みのイメージが湧かないことがわかり⁹⁾、開口部を追加することにし、その対策として壁ポリゴンを作成した。この開口部導入のために必要となった3次元化処理の変更は、クラス「壁ポリゴン」の作成と建物のポリゴン化に壁ポリゴンを発生させるようにしたことのみである。よって、物体やポリゴンの管理者における処理の変更はいっさい必要としなかった。

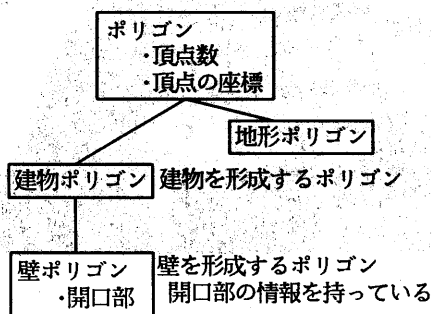


図-9 ポリゴンにおけるクラスの継承

このように、特殊な場合にもクラスの継承を用いることによって、容易に対応することが可能である。これはポリゴンのみにとどまらず、物体やシステムそのものにおいても同じことがいえる。

5. 評価

標準化モデルを用いたことにより、入力すべき情報を簡略化し、局所認識を用いたことによって人間の負担を軽減した。これらの結果として、本研究が目的とした効率的でかつ高速な入力インタフェースが実現できた。

実際に入力実験をおこなった結果、オペレーター一人が40棟を入力するのに、開口部のない建物では約10分、開口部のある建物では約30分で入力できた。これは、それぞれ240棟・人/時間、80棟・人/時間となる。

比較対象として、ディジタイザを用いて、本システムと同じく1/2500の都市地図に建物の外形を入力した例⁹⁾をあげる。これは、外形情報に加えて、屋根形状と階数を選択するとワイヤフレームモデルを作成するものであり、本システムにおける開口部のない場合と同等である。この例では、400棟の入力に12時間かかっている。これは、約33棟/時間となる。実際に入力している数が違うので単純には比較できないが、この結果より本システムが高速で入力することが可能であることがわかる。

また景観シミュレーションについては、図-10の画像を生成するのに、約1分かかる。これは、入力した情報の確認するために、たびたびシミュレートした画像を出力するという作業をするうえで、十分な速度といえる。

6. まとめ

オブジェクト指向を用いたことにより、新しい物体の追加はクラスの追加という形で実現され、建物の入力や3次元視覚化処理などすでに完成している環境を、そのままその物体の環境として用

いることができる。また、導出クラスを用いることによって複雑な形状に対応することも可能である。このようにオブジェクト指向を用いて拡張性の高いシステムを構築することができた。

今後の課題として、樹木などの新しい物体の追加などがあげられる。

注釈・参考文献

- 1) 鈴木・星野・山田 (1986年) : 「地図情報入力システムMARIS - 自動認識処理 -」情報処理学会第33回全国大会4N-4, pp. 1485, 日本情報処理学会
- 2) 堀・古賀・恒川 (1987年) : 「総合化地理情報システム (3) 図形認識機能をもった図形入力エディタ」情報処理学会第34回全国大会E-3, pp. 1791, 日本情報処理学会
- 3) 山川・小田 (1987年) : 「会話型図面入力システム」日経CG No. 7, pp. 120, 日経マグローヒル社

4) X Window system X11R4, MIT, U. S. A.

5) C++ version 1.2, AT&T

6) 間瀬・瀬口・加藤・平本・河合・大岩 (1990年) : 「都市景観システムのための地図情報入力インタフェースについて」第6回ヒューマン・インタフェース・シンポジウム論文集, p. 313, 計測自動制御学会: ヒューマン・インタフェース部会

7) 藤阪・河合・大岩 (1988年) : 「CGとオブジェクト指向」PIXEL No. 65, pp. 63, 図形処理情報センター

8) 間瀬・瀬口 (1990年) : 「都市景観シミュレーションのための建物形状データ入力支援システムの研究 - 外壁開口の表現 -」第25回日本都市計画学会学術研究論文集, pp. 265

9) 笹田 (1988年) : 「建築設計・都市計画におけるコンピュータグラフィックス」情報処理Vol. 29 No. 10, pp. 1177, 情報処理学会

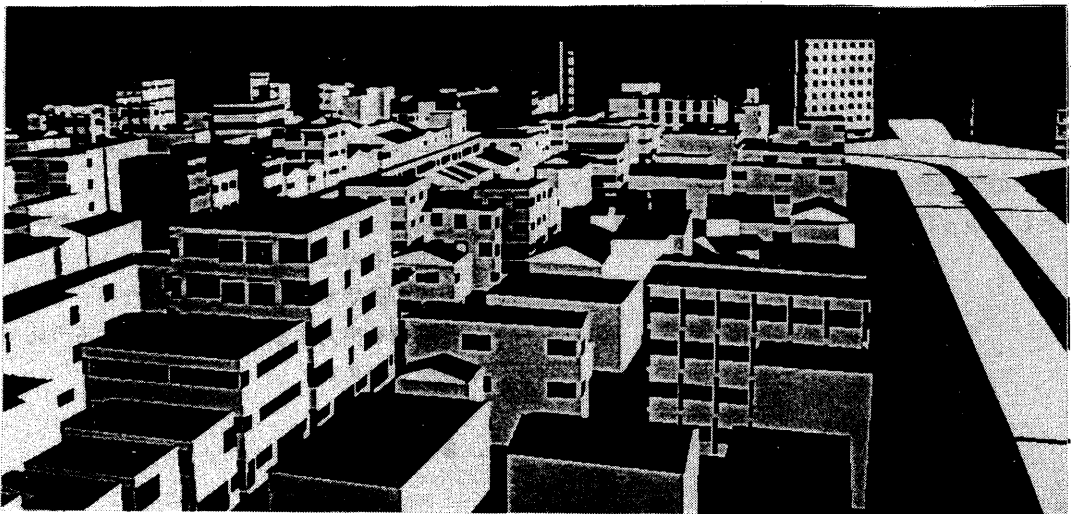


図-10 3次元視覚化の出力例