

Z座標ソート法による透視図と 洪水シミュレーション*

2Z-8

星 仰 ジャイメ・鶴田 友松 健一†

茨城大学‡

1 はじめに

地図を眺めているだけでは、実際どのような地形をしているか深く理解することは困難である。そこで、標高データを用いて透視図を表示し、更に、表示する地域を移動させることにより、あたかも3次元空間内を移動しているかのようになる。陰面消去の処理では高速なZ座標ソート法を用いて実現できる。標高データは、徳島県の吉野川周辺のものを使用し、ここでは、標高データより河川を判別する処理も行っている。また、視点の方向を変えることにより、4種類の透視図が表示可能なアルゴリズム開発をしている。吉野川が洪水氾濫した状況を推測する処理も行なっている。

2 透視図の表示

3次元空間の表示は、標高データから生成している。標高メッシュから、高さ方向の状況を把握して透視図を作成している。用いたデータは、全国の標高データである250mメッシュである。

2次元の画面である投影面に、遠近感を伴った映像を出す方法を説明する。これは1点透視投影法と呼ばれるもので、原理は単純で、遠いものは小さく見えるという方法である。投影の様子を、投影面の真横から眺めた図1で考えてみる。視線とA点を結ぶ直線を考え、この視線と投影面の交点A'を求められる。この点がA点の投影点であると考えられる。次に、視点とB点を結ぶ視線を考え、この視線と投影面の交点B'を求める。A'よりもB'の方が小さいY座標になることがわかる。これが、遠近感を出すための細工である。A'点とB'点のY座標は、三角形の比率配分で求められる。

$$A'_y = \frac{d}{d + A_z} \cdot A_y \quad (1)$$

$$B'_y = \frac{d}{d + B_z} \cdot B_y \quad (2)$$

図1より投影面の点の値がどの程度になるかは、三角形の底辺の長さで決まった。そして、三角形の底辺は、ポリゴンの空間座標と、視点の位置で決まる。ポリゴンの空間座標は固定であるから、視点をどこに置くかにより、遠近感の強弱が決まってくる。視点は、Z軸上の任

意の位置に配置できるので、視点を投影面から遠くすれば遠近感は弱まり、逆に視点と投影面を近づければ遠近感が強まることわかる。いままでは、Y軸上への投影であったが、X軸上でも同じである。すべての頂点のX座標とY座標を、この三角形の比率配分で投影する。この方法により、2次元である投影面に、3次元空間の表示が可能になる。

次に、投影面に物体を表示する処理のレンダリングを行う。1点透視投影法より求められた座標より点を結び、ワイヤフレームを描画できる。次の困難な作業として、近くのは遠くの物体を覆い隠すという、陰面消去処理がある。このためには、ポリゴンの頂点座標値だけを頼りに判断しなければならない。これには、いろいろな方法が考案されているが、どれも一長一短であり、決め手になるアルゴリズムは目当たらない。高精度な映像を要求すると投影時間が長くなり、高速な投影の要求と映像の写実性が合い反する面がある。Z座標ソート法は、高速に投影できる方法でアルゴリズムも非常に単純である。遠点ポリゴンを先に描画して、近点ポリゴンを後で描画するため、近点にあるものが遠点にあるものを着実に覆い隠すことができる。これは画家が風景を構成する考え方から、ペインターズアルゴリズムとも呼ばれている。この方法では、凸面体や交差した図形が正確に表示できないので、小領域に対象を区切ることにより、この欠点を補うことができる。本研究では、メッシュデータを描画時にはトライアングルサーフェースデータとしている。

透視図の視点移動による表示については、表示領域をずらす方法を採用する。この方法により、3次元空間内をフライトシミュレーションすることも可能となる。また、標高値によって表示色を変えているので、高低の判断が容易に行なわれる。図2と図3は、吉野川河口付近

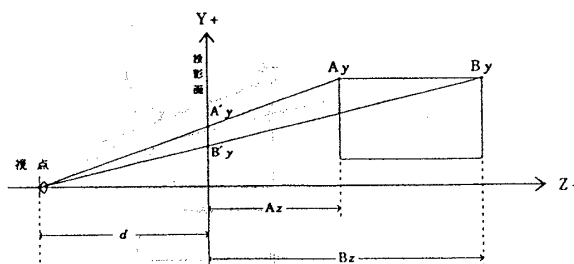


図1 1点透視投影法

*"Perspective Projection using Painter's Algorithm and Flood Simulation"

†Takashi Hoshi, Jaime Tsuruta and Ken-ichi Tomomatsu

‡Ibaraki University

4-12-1 Naka-narusawa, Hitachi, Ibaraki 316, Japan

の様子を表示したものである。

3 文字の表示

機能を付加することにより、画面の内容をより理解することができる。文字の表示は、解像度により変化することもあるが、ここではプリンタのポイントを基準に表示することとする。フォントの選択も可能であり、その上、フォントの属性も指定することができる。属性とは、フォントの高さや幅、太さなどのことであり、独自の文字を表示できるのである。

4 河川の判別

標高データから、基準なく河川を判別することは、非常に難しい。しかし、河川の源流が分かれば、そこを起点にして分類が可能となる。まず、下流方向を確認してから、河川の右と左、下流の5方で判別を行う。本来なら、この5方向で標高が最も低いところが下流となる。しかし、標高の数値の単位は[m]のために、凹凸が多く必ずしも流水するシミュレーションができない面を含んでいる。この対策の一つとして、下流方向にウエイトを付加することとした。こうすることにより、河川が左右に蛇行することが少なくなり、より本来の河川床に近づくことができる。もちろん、ウエイトは河川の上流部と河口部では異なる値となっている。下流3方向についても適切な数値を設定している。とくに、河口部では標高がほぼ同値であるので、妥当な値を指定しなくてはならない。さもないと、意図しない方向を判別してしまう場合がある。

河川の判別の他に水域の判別も行うことができる。水域の場合、標高値で容易に処理できる。

5 洪水シミュレーション

大雨、雪解けなどにより河川の水が増加し、氾濫そして流水する事が洪水である。この流水した地域を求め、表示することを目的としている。簡単に述べると、河川の標高値より一定値高い地域に流出するということにな

る。もちろん、ほかにもいくつか基準はある。河口付近や上流では様子が変わってくるので、河川に源流からの距離により判別するようにしている。また、河川のどの箇所から流水したかも念頭においておかななくてはならない。

以上の条件より、洪水氾濫したときの危険区域を求めることが可能となる。

また、源流からの距離や標高の高低差、流水地域の幅などから到達時間も計算可能である。洪水氾濫したときの危険区域の変化を随時表示することにより、洪水の様子を視覚的に観測することができる。ただし、このシミュレーション計算では、現在のところ、支派川や高潮、道路等の微地形などの影響を考慮していない。

6 おわりに

標高データを3次的に表示することにより、誰でも地形の様子が一目瞭然に理解できる。これは、地図からでは得られない情報を含んでいると考える。さらに、河川や建築物などを表示することにより、その位置関係なども理解できる。洪水氾濫後の様子も視覚的に理解できる。洪水の氾濫地域は、上流部よりも河口部に近付くにつれて広がることがわかる。

表示する時間の短縮が今後の課題である。また、河川の判別でも述べたが、標高の数値の単位が[m]であるがために精度が若干粗いものとなっている。

参考文献

- [1] 新藤義昭：“Borland C++によるグラフィックスプログラミング入門”，技術評論社，pp.24-182，1994.9.
- [2] Christopher Lampton (河嶺友仁 訳)：“C++ フライトシミュレータプログラミング”，アスキー出版局，pp.185-479,1994.9.
- [3] Duke (NIFTY-Serve FGALST SYSOP)：“Windows プログラミング Borland C++ 4.0J”，BN N，pp.115-149,1994.12.

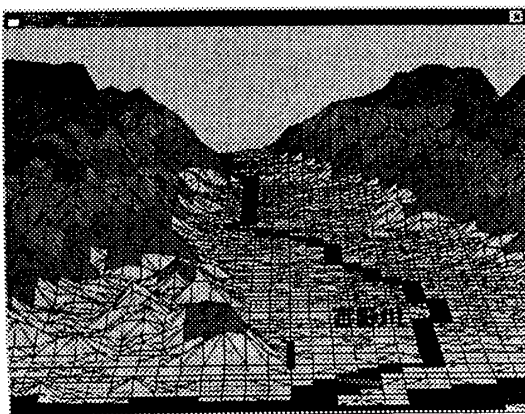


図2 吉野川河口付近(東方視点)

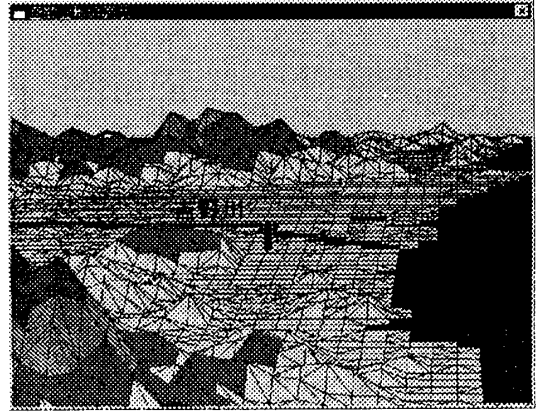


図3 吉野川河口付近(南方視点)