

連載「誰も書けなかった GIS」第9回

地図データフォーマット変換の憂鬱（その1）

(株) エヌ・シー・エム 代表取締役社長 柳田聡（やなぎだ さとし）

1982年東京大学工学部土木工学科卒業。同大学院修士，博士課程を経て1985年より現職。工学博士。専門は画像処理及び地理情報システム。

1 はじめに：なぜこの文章を書きたくなったか

「地図データフォーマット変換」、筆者はこの語句を聞く度に憂鬱になります。と言うのは、巷ではこの用語は、よく吟味せずに、深くその意味を掘り下げることなく用いられているからです。そしてその結果、全く頭が痛いことなのですが、地図データのフォーマット変換は簡単、単純だと思われている、従って、金銭的な評価も余り高くないということに相成るのです。この事態を覆すべく、色々説明をと思うのですが、なかなか果たせません。と書くと、「それは、君の怠慢である。」と言われそうです。「わかりました。では出来るだけフォーマット変換のドロドロした部分を、弊社の数々の辛酸をなめた経験もまじえて、御説明致しましょう。」...というわけで、今回はフォーマットを取り上げます。

なお本文では、出来るだけ説明を明快にするために、現実に用いられているフォーマットを具体例として使用していますが、筆者は必ずしも、それらのフォーマットについての完全な情報を持っているわけではありません。単なる個人的な経験、または入手可能なドキュメントからの類推に基づいていることが大部分です。よって場合によってはピンと外れなことを記載しているかもしれません。その点は御容赦頂ければ幸いです。また今回の原稿においては、議論を具体的にするために、止む無く商用ソフトウェア及び実際のフォーマットの名称を明記しつつ話しを進めております。

2 フォーマット変換の難しさの感覚的な説明

「地図データフォーマット変換は難しい。」大抵の方は、こう言われてもピンと来ないのではないのでしょうか。ではこういう例はどうでしょうか。お饅頭が20個あるとします。この20個が横長の箱に入っているとします。それを縦横の長さが逆になっただけの縦長の箱に詰め替えることを要求されたとします。あなたは簡単に出来ますか？答えは多分 Yes でしょう。フォーマット変換もこのレベルであるならば簡単なわけです。つまり単なるフィールドの並べ替えですんでしまう場合です。ところが実際の世の中では、20個のお饅頭しか入らない箱に23個詰めることを要求されたりします。あるいは同じ20個でもお饅頭でなく、ショートケーキを詰めることを要求されることさえあります。これではたまったものではあ

りません。フォーマット変換もこれに似ています。以下ではこのことを出来るだけ論理的に御説明致します。

3 異なる地図データフォーマット間の高い壁

フォーマットとは何でしょうか。それは地図データをある手法でデジタル的に記述するための規則です。従って、「記述対象は何か(どんな地図データか、記述の範囲は何か)」、「それをどう記述するのか。」という2つの観点から評価出来るわけです。ここから先はいくつかの具体例をあげて、この評価の視点に沿って、説明します。

3-1 何を記述するか、記述の範囲は何か

本節では、「何を記述するのか、記述の範囲は何か。」という観点からみたフォーマット間の違いについて考えてみます。

- 汎用か特定か：例えば ArcView-Shape 形式は汎用フォーマットです。ある特定のデータのためのフォーマットというわけではありません。対して例えば、日本には「(財)日本デジタル道路地図協会」という財団法人があり、ナビゲーション・道路管理者用のデジタル道路データを作成、販売しております。この道路データフォーマットはまさにこの道路データ専用のフォーマットで、具体的なデータ像に直結しています。その結果、「(財)日本デジタル道路地図協会 - 道路データフォーマット」と言うと、必然的にデータの単なる入れ方の範疇を超えて、作り方、規則、精度、論理検査レベル、コードの範囲、そしてデータの内容など詳細な情報が明確になります。そういう意味では、このフォーマットは、フォーマットと言うよりデータ説明書・規定書と言った方が正確かもしれません。私にはこの2つのフォーマットが同じフォーマットという用語のもとに括られるのが気持ち悪くてなりません。そして当然のごとく、特定フォーマットから汎用フォーマットへの変換はやり方次第で可能ですが、その逆は不可能です。というのは道路の種別コードしか定義されていない特定フォーマットで、どうやって湖の種類を記述出来るのでしょうか。以上より、特定/汎用フォーマット間の壁が高いことがお分かり頂けたでしょうか。
- 位相構造の有無：位相構造の有無もよく問題になります。この件で楽しい難題をよく提供してくれるのが AutoCad-DXF です。DXF には位相構造がありません。にもかかわらず、位相構造を作ることを要請されることが多いのです。無から有を作る、これって(フォーマット)変換でしょうか?どっちかと言うと「データ編集」、「情報抽出」、「自動認識」と言った方が正確だと思うのですが?にも関わらず世間ではこの両者を区別していません。

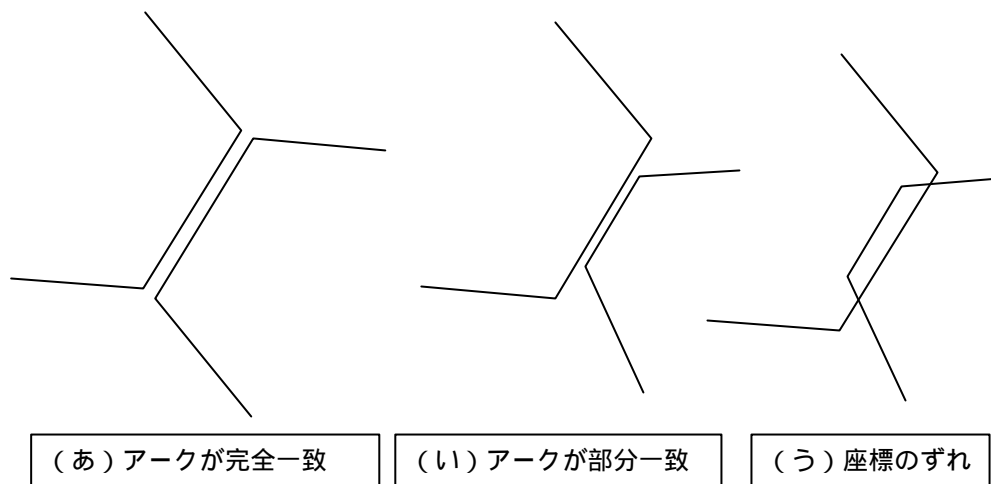


図 1 境界線共有の条件

まあブツブツ不平を言うのは止めて兎に角、位相構造がない場合は何とか作らなければいけない、その際に生じる問題として、位相構造の代表例である境界線共有の認識のお話をします。DXF では隣り合うポリゴン同士が境界線を共有しているか否かが明快にデータ上で記述されておりません。境界線の座標のみで判断しなければいけないわけです。その結果、一言で境界線共有と言っても何を判断基準にすれば良いのか分からないのです。例えば図 1 では 3 つの場合を考えています。左端の (あ) はアークが完全一致しているので境界線共有でしょう。でも (い) の場合はどうでしょうか。更に (う) の様に多少ダブっていたら？あるいは逆に多少離れていたら？これらは非常に頭が痛い問題なのです。更に腹立たしい、あるいは困ったことには、上記の問題に対する解は多くの場合、誰からも与えられません。全て自己責任のもとに決定しなければいけないわけです。自分自身でデータを作成したわけではないのに、です。勢い実際にデータの内容を見て、判断することになります。後述する様に、フォーマットでなくデータがフォーマット変換ロジックを大いに規定する一例です。DXF には後程また登場して貰いましょう。

- 描画スタイル情報の有無：DXF や MapInfo-MIF フォーマットでは、データ内に描画スタイルに関する情報（例えば赤実線で描画するなど）も保持しています。対して、ArcInfo-Export や ArcView-Shape では、持っていません。これより、前者より後者へ変換される場合は情報損失が起こり、後者より前者に行く場合は、描画スタイル情報は未定義にせざるを得ないことが分かります。

3-2 どう記述するか

本節では、「どう記述するか。」という観点からみたフォーマット間の違いについて考えてみます。

- ポリゴンの穴：ポリゴンの穴をどう記述するかは興味深い問題です。図 2 は実際に出会った DXF 形式データの例です。

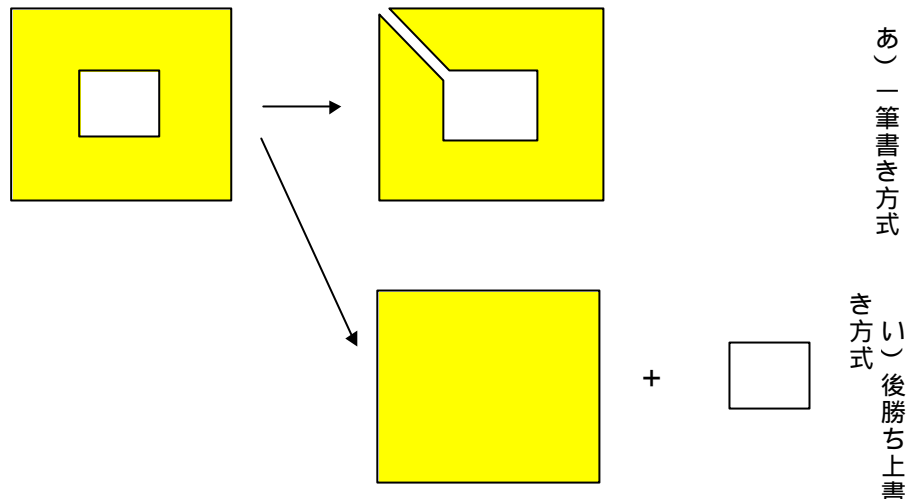


図 2 穴の記述

穴空きポリゴン DXF で表現するのに、当然、我々は (あ) の手法を予想していました。というのは単純に内部がないのだから、内部を抜いていると思ったのです。ところが結果は違いました。(い) なのです。これはどういう手法かと言うと、後勝ち上書きデバイスに描画する場合、最初に外枠内全体を塗りつぶし、その後内部を白で塗れば自ずと見た目で内部がくり抜かれて見えると言う発想です。なるほどこれも一つの手法だと思われるかも知れませんが、ちょっと待って下さい。例えば描画デバイスが後勝ち上書きデバイスでなく透過型だったらどうしますか。重なった内部が意図しない色で見えるでしょう。更にこのままで各ポリゴン毎の面積演算を行うと誤まった結果になることもお分かり頂けますね。このままでは穴の部分もカウントされ、過大な面積になってしまうのです。つまり一言で言うと、このデータは嘘なのです。そこで、いちいち自分の内部がどれに当たるかを見ることによる内部くり抜き作業が必要になりました。この演算をやる羽目になると別のことが心配になります。図 3 の様にポリゴンが重なっていたらどう解釈するかです。

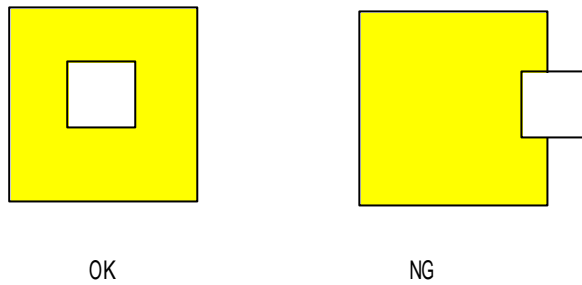


図3 重なり合いポリゴン

更に、もしかして図4の(い)のケースの様に、内部くり抜きでなく、まさにデータ通り2ポリゴンのオーバーレイ状態を記述したかったらどうでしょうか。これは最早フォーマットの問題ではなく、データの(作成)規則の問題ですね。

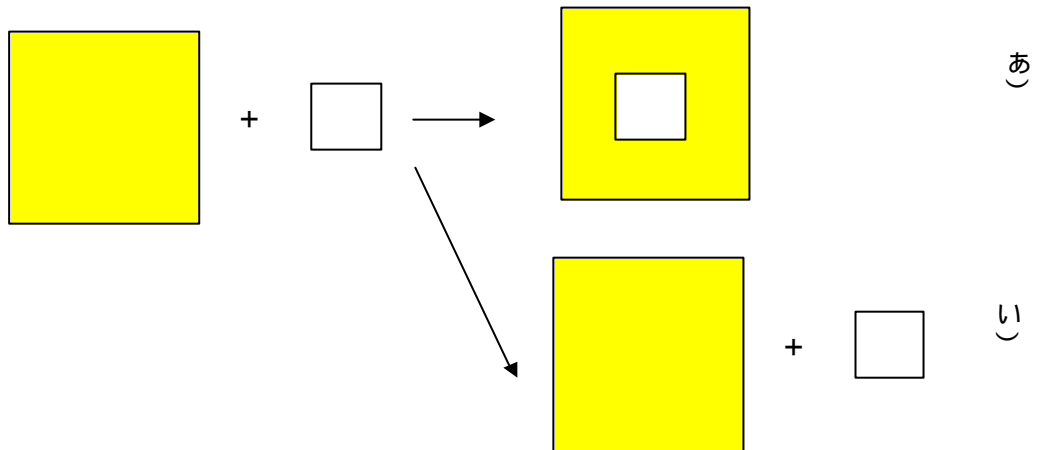


図4 穴に関するデータの解釈

いずれにせよ、DXFの件はこうまとめられると思います。DXFでは位相構造がなく、従ってポリゴンの記述は自由です。ポリゴンの重なり合いOK、隙間があいてもOK、言わば無法地帯です。対してArcInfo-Exportや弊社のGeoStar-Vecは、厳密なポリゴン規則(例えば1レイヤー内では重なり合っては駄目、互いに隣接して、隙間の無い平面を作らねばならないなど。)を要求します。よって白黒曖昧な状態から白黒はっきりした状態に変換する、この処理が高い壁にぶち当たるのは自明の理です。

これ以外にもポリゴンには頂点で接しているポリゴンを許すか否か(図5参照)など面白い問題が一杯あるのですが、きりがないので先に進みます。

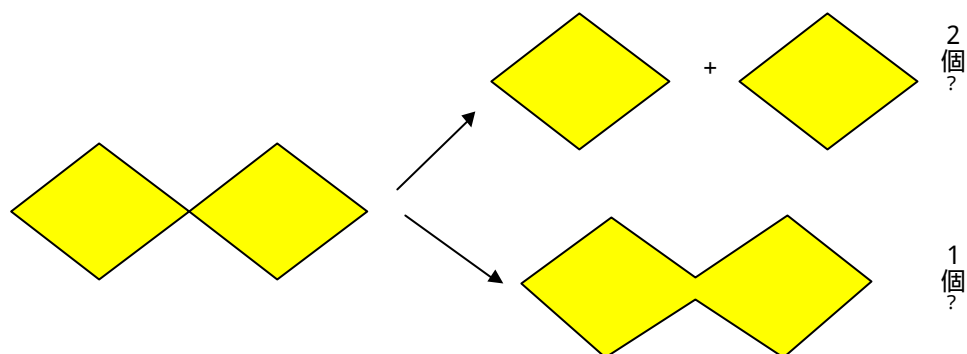


図5 頂点で接するポリゴン：1個か2個か

- 座標の精度：かつて某市の都市計画データを ArcInfo-Export フォーマットから取り込もうとした時に、発生した問題について御説明致します。ArcInfo-Export フォーマットでは座標は実数で記述されています。対して、例えば「地図センター - 数値地図2.5万」、「(財)日本デジタル道路地図協会 - 道路データフォーマット」などでは座標は整数で記述されています。弊社の GeoStar-Vector フォーマットでも整数です。整数にした理由は、座標の最小単位、精度を明確に示せるからです(例えば、1:25,000 地形図を(10,000,10,000)で正規化した場合、座標の最小単位は1:25,000 地形図の1/1万であると明確に分かります。)。さて実数座標の ArcInfo-Export データを整数座標の GeoStar-Vector に変換して取り込んだところ、ポリゴン自動生成においてアーク追跡エラーが発生しました。ここでポリゴンは、図6の様なロジックで自動生成されます。キーポイントはアークの自動追跡ですが、例えば図6のノードn1ではアークa1から次にはアークa2ではなくて、アークa3に行かなければいけません。この際の判断基準はアークの角度です。ところが線分の角度を求めるためには、当然、線分には長さが必要です。しかし前述の様に、ArcInfo-Export データを取り込んだ時に実数 => 整数変換を行なったため、極めて値が近い実数座標が同一の整数座標に変換され、その結果、長さ0の線分が発生してしまいました(アークの起終点が一致してしまったのです。)

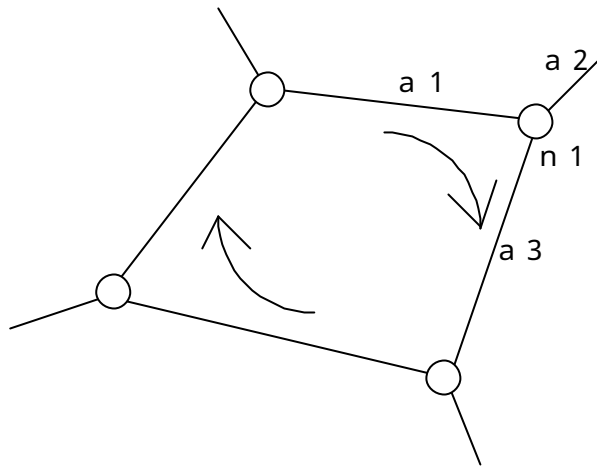


図6 アークの自動追跡

この結果、ポリゴン自動生成がうまく行かなかったのです。ちなみに短いアークとは地上で長さ8.5cmでした。1:2,500データにこのような短アークがあるとは予想していませんでした。この時は整数化の際の量子化単位を上げてしのぎました。こう書くと、「下手に実数⇒整数変換の際に精度を落とすから悪いのだ。」と言われそうですが、では逆にこのような例はいかがでしょうか。

同じArcInfo-Export都市計画データの話ですが、図郭線の抽出を行なう必要に迫られました。ところが図郭線であるという明確な定義がデータ内に含まれていなかったため、やむなく座標値で判定することにしたのです。しかし実際のデータでは図郭の座標値が宣言、予想されていた値から微妙にずれているケースが見られました(図7参照)。よってこの場合は、ある許容範囲内ならば、図郭線とみなすと言う、逆に座標精度を低目に評価する処理を要求されたのです。

いずれも、座標(の精度)って何だろうと考えさせられる事例です。

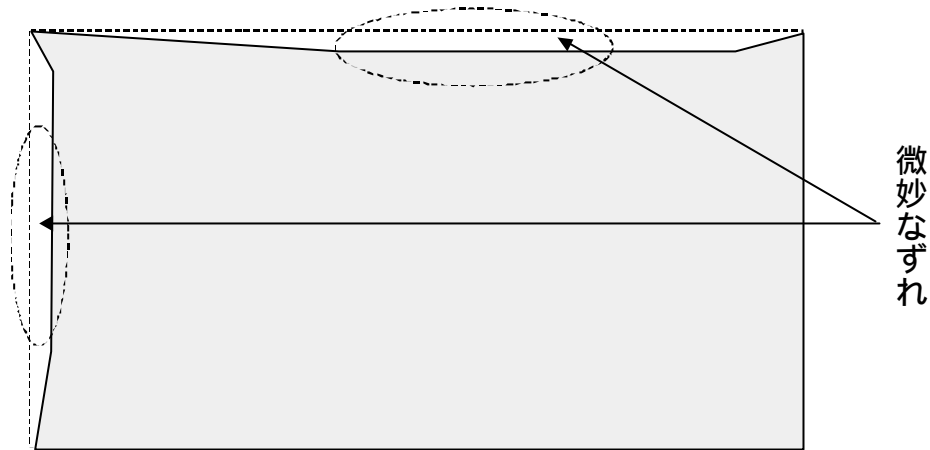


図7 図郭線の認識

お話しが佳境に入って来た所で、ページ数が尽きて来たので、ここから先は次回にします。またまた、「一回一回の読み切りにします。」と言う連載開始時のお約束を破ることになってしまいました。申し訳御座いません。お詫びに、次回の連載内容の概要を以下に紹介します。

4 フォーマット変換の曖昧さ

以上の事象を材料にフォーマット変換(の目的)の曖昧さを論じます。

5 データの強い主張、影響

5 - 1 データの強い個性

フォーマットだけではなく、そのフォーマットに準拠して作られた実際の地図データの形態もフォーマット変換アルゴリズムに強い影響を与えることを述べます。

5 - 2 新規の問題：データの質の重要性

データの質の善し悪しがフォーマット変換に影響を与えている現状の不合理を指摘します。

6 まとめ

以上の議論をまとめます。

7 今後どうなるか

「OpenGIS」(データ共有を目指す企業連合)の動きを踏まえて今後は占います。