

5E-10

# 高速高精細な拡大縮小表示を可能とする イメージ地図格納方式

宮崎泰彦

NTT情報システム本部

## 1.はじめに

コンピュータマッピングシステムにおいて、データ入力のコストダウンを図るために、紙の地図をスキャナで読み込み、イメージのまま表示する方式が、近年使われるようになってきている。このとき、高解像度のスキャナで読み込まれた地図を、どの程度の画素密度でシステムに入力するかが、データ構築時の課題となる。実際、コンピュータマッピングシステムでは、全体を見渡すための縮小表示と狭い範囲の詳細を見るための拡大表示が、頻繁に繰り返される。拡大表示時の精細さを高めるために、画素密度を高いままにすると、縮小表示時の画素が大量になり、表示に時間がかかる。一方、縮小表示時の画素量を減らすために、画素密度を低くすると、拡大表示時の画質が低下する。本稿では、このようなジレンマを解決するため、NTTの設備管理マッピングシステムでとった手法について述べる。

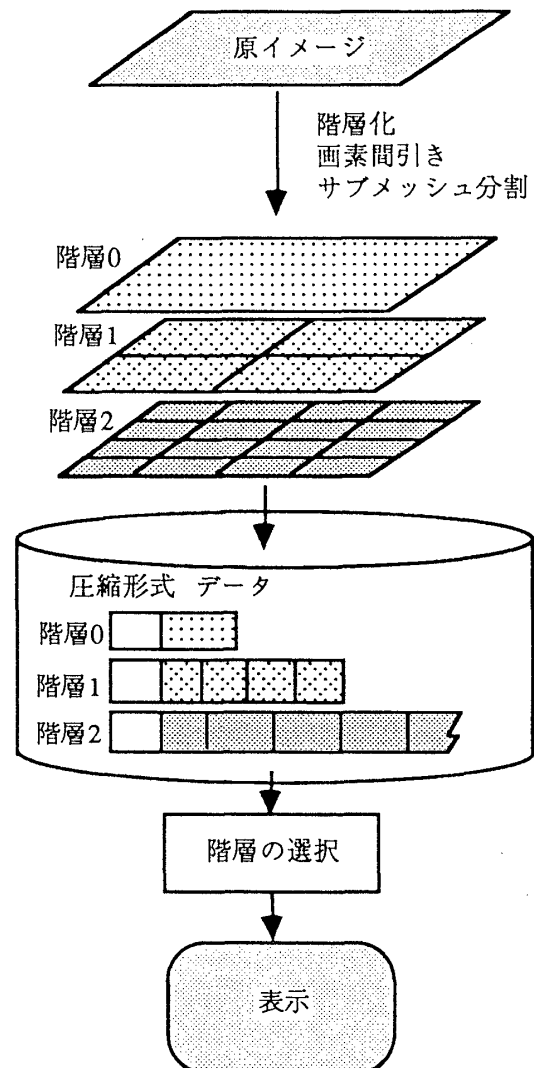
## 2.階層化と画素の間引き

表示品質と表示性能を両立するためには、データの画素密度と画面の画素密度がなるべく近くなるようにしておけばよい。そこで、スキャナで入力された地図の1メッシュ分の原イメージを表示倍率に応じた複数の階層に分ける。階層数は自由に設定できるが、ここでは階層0,1,2の3階層に分けることとして説明する。NTTの実際のシステムでも3階層である。

階層0は主にメッシュ全体の(1倍)表示用、階層1

は主に $N_1(>1)$ 倍程度の拡大表示用、階層2はさらに詳細な $N_2(>N_1)$ 倍程度の拡大表示用とする。また、スキャナから読み込んだ原イメージの画素数を $M_x \times M_y$ 、地図を表示するモニタの画素数を $R_x \times R_y$ とする。通常、スキャナは大型で高解像度のものを使用するため、 $M_x, M_y$ は $R_x, R_y$ に比較して大きな値となる。

それぞれの階層は、表示倍率に応じて、必要な



イメージデータの処理の流れ

Storage of Raster Data to Display Enlarged and  
Reduced Map Quickly and Clearly

Yasuhiko Miyazaki

NTT Information Systems Headquarters

1-6 Nakase Mihama-ku Chiba-shi, 261, Japan

だけ画素を間引く。例えば、階層0のイメージデータは、予め $\left[\frac{M_x}{R_x}\right] \times \left[\frac{M_y}{R_y}\right]$ 画素を1画素に間引いておくことにより、データの画素数が、約 $R_x \times R_y$ となる。つまり、メッシュ全体を表示する際の表示画素数と同じになる。同様に、階層1のイメージデータは $\left[\frac{M_x}{R_x N_1}\right] \times \left[\frac{M_y}{R_y N_1}\right]$ 画素を1画素に、階層2のイメージデータは $\left[\frac{M_x}{R_x N_2}\right] \times \left[\frac{M_y}{R_y N_2}\right]$ 画素を1画素に間引くことで、ほぼ、 $N_1$ 倍、 $N_2$ 倍で表示する際の画素密度になる。従って、階層2になるほど間引く率は減る。

こうして画素を間引くことによって、階層0,1,2のデータ画素数は、それぞれ約 $R_x \cdot R_y$ ,  $R_x N_1 \cdot R_y N_1$ ,  $R_x N_2 \cdot R_y N_2$ となる。階層0のデータ画素数は階層2のデータ画素数の $\frac{1}{N_2^2}$ にすぎない。

### 3. サブメッシュ分割

階層1や2は、拡大表示に使われるので、メッシュ内の一部しか表示されない。このため、メッシュ内のどの部分のイメージデータを表示すべきかを検索する処理が必要になる。予め小さなサブメッシュに分けておくと、この処理時間が短縮できる。

例として、 $N_2$ 倍で表示する場合、つまり、メッシュの $\frac{1}{N_2} \times \frac{1}{N_2}$ の範囲を表示する場合考える。階層2の画素データを、 $\frac{1}{N_2} \times \frac{1}{N_2}$ の大きさの $N_2^2$ 個のサブメッシュに分割すれば、サブメッシュの区切りと表示位置がずれている可能性を考慮しても、高々 $2 \times 2$ サブメッシュのデータだけを検索してくればよい。

同様に階層1も $N_1^2$ 個のサブメッシュに分割しておく。1つのサブメッシュの画素数は、全て、約 $R_x \cdot R_y$ となる。

### 4. データ格納方法

以上のように間引かれ、分割された画素データをデータベースとして格納する。こういった画素データの格納においては、MHやMMRをはじめとした様々な圧縮技術が確立している。しかしなが

ら、これらの圧縮形式には、各々圧縮効率や伸長処理速度などに特徴がある。また、圧縮する画素データによって、これらの性能は変化する。

一般的には、2値画像の場合、白と黒のパターンの反転が多く、連続したパターンが少ないほど、圧縮効率は落ち、伸長に時間がかかる。地図の画素データを一律に間引くと、パターンの反転は増える。実際にある地図についてMMRの圧縮効率を測定したところ、階層0,1,2の順で、58%, 27%, 13%となった。 $(N_1=2, N_2=4)$

このため、データの格納時に、階層ごとに格納形式を指定できるようにした。これによって、圧縮効率が悪い階層0は伸長時間の不要な無圧縮形式で、階層1,2はMMR形式で格納するといったことが可能になる。このとき、どの形式で格納したかという情報も一緒に格納しておく。

### 5. 表示方法

表示の際は、まず表示すべき倍率を求める。その倍率と $1, N_1, N_2$ を比較し、階層0,1,2のどれを使用するかを決定する。階層1や2のようにサブメッシュ分割してある階層を使う場合は、表示位置からどのサブメッシュを使うか計算し、必要なサブメッシュのみを検索する。同時に圧縮形式の情報も参照し、逆変換をかけて画素データに戻す。任意の倍率の表示に対応するため、最終的には画素の補間が必要であるが、適切な階層のデータを使用することで、極端な補間をする必要はない。

### 6. まとめ

コンピュータマッピングシステムにおいて、地図をイメージのまま表示する際に、全体表示時の処理の速さと、拡大表示時の精細さを両立させるため、データ構築時に画素密度を段階的に低くしたデータを生成し、階層的に格納する方式を考案した。更に、階層数や格納形式を利用者が指定できるシステムを作成することで、利用形態に応じた地図データが容易に構築できるようにした。