

# パーソナルコンピュータを用いた地理的データ処理 および主題図作成

小 方 登\*

## Geographic Data Processing and Thematic Mapping Using Microcomputers

Noboru OGATA

### 1. 地理学教育におけるパーソナルコンピュータ利用の意義

近年におけるパーソナルコンピュータの普及はめざましく、業務・科学研究の分野だけでなく、最近では教育の現場においてもその活用法が模索されるようになった。いうまでもなく大学においては、今日に至るまで自然科学系の諸分野を中心とした研究・教育の様々な局面で、大型機を含めたコンピュータの利用法の開発が進められてきた。地理学においても専門的な研究におけるコンピュータの利用はすでに定着したと見てよい。ただ文系の学部レベルの地理学教育においてはコンピュータはまだ特殊なものとして受け取られることが多く、たとえば実習における標準的な教科内容として扱われることは一般的とはいえない。<sup>1)</sup>しかし今日では情報関連機器は社会のあらゆる分野に浸透しており、文系の学生といえども卒業後パソコンなどに接する機会も多い。大学の教育において何等かのかたちでコンピュータの利用に慣れさせることは必要なことと考える。本稿では大学の地理学教育の道具としてパーソナルコンピュータをどのように活用し得るかについて、主として地理的データ処理・主題図作成の方法を扱いつつ論じるものである。

今日では日本語ワードプロセッサ・表計算・データベースなど実務に耐える既製のソフトウェアが充実し、これらを活用することがパーソナルコンピュータの主たる利用形態となっている。コンピュータに慣れるという意味で表計算・データベースのような簡易言語を使用する既製ソフトウェアは大学における地理学の教育内容ともよく適合し、活用の余地は大きいといえる。たとえば、表計算ソフトウェアを用いて地理行列の形に統計データを集計したり、データベースのソフトウェアを用いてフィールドノートや文献の整理をするといった具合である。

ただ学生がそれぞれの研究課題に取り組む際には、分析方法におけるオリジナリティを求められるので、定型的な処理しかできない簡易言語の類では不十分な場合も生ずる。そのような場合たとえ初歩的であってもBASICのように習得容易なプログラミング言語についての知識があれば、与えられたプログラムを加工して、各自の目的に適するようなものとすることも可能となる。<sup>2)</sup>

ところで各種のデータ処理のうち地理学研究において特徴的なものとして空間的あるいは

\* 地理学研究室（昭和62年9月30日受理）

は地理的データの図化ということがあげられよう。このことは研究テーマに対応する主題図などの形で研究成果を読者へ伝達するための手段として活用されてきた。もしコンピュータを用いて主題図を描くことができるとしたら、地理学研究の道具としてだけでなく、学生に地理学研究や情報処理機器の活用を動機づけるデモンストレーションの効用を期待することができるであろう。

大型計算機を用いた地理的データ処理・地図製作は業務用を中心に既にかかなり定着した。<sup>3)</sup>しかし大型計算機で用いられる汎用プログラミング言語は通常グラフィック機能を持たず、図形処理を行うには専用の周辺機器とソフトウェアを必要とするため、誰にでも大型計算機を用いた地図製作ができるわけではない。その点パーソナルコンピュータは、標準的な機器構成においてもCRTディスプレイによるグラフィック機能を持ち、BASIC言語によるグラフィック命令も簡便である。図形処理が比較的簡便に行えるという利点を生かして近年ではパソコンを地理的データの処理とその地図表示のために活用する試みがいくつかなされている。<sup>4)</sup>

またパーソナルコンピュータ用として地理的データ処理・地図表示の手順を含む業務用ソフトウェアが各種供給されるようになってきている。これら既製のパソコン上のソフトウェアは業務用としては完成度の高いものである。しかし地理学研究ないしは地理学教育の道具としてみた場合、たとえばユーザー自身の手で地図データを作成できず、業者が供給する地図データを別途購入しなければならないものが多いといった不満な点も残っている。また完成品として供給されるがゆえに、研究目的に適合させるためユーザー自身がプログラムを加工する余地が乏しい。

一般に既製のソフトウェアを利用する場合には一連の処理手順が最初に与えられ、これにデータを投入するという形をとるが、現実の研究においては、まずデータを用意してその処理結果の検討を通じて必要とされる新たな処理手順（プログラム）を開発する、という手続きをふむ場合も多いと思われる。ベースとなるプログラムが習得容易な言語で書かれ、データ構造が公開されていれば、たとえば卒論研究の指導などにおいて教員が学生に各自の研究で必要な処理手順を実行させる場合などに有効となる。

次章以下ではパーソナルコンピュータを用いてユーザー自身がデータを入力し主題図を作成することができるシステムについて解説し、あわせて問題点や今後の課題について論じる。システムは2つあって、Ⅱ章で扱うのは格子型のデータから主題図を作成するもの、Ⅲ章で扱うのは市区町村域などポリゴン型のデータから主題図を作成するものである。本稿に掲載されるプログラムリストおよび出力図はすべて筆者が奈良大学地理学教室の機器を使用して作成したシステムのものである。<sup>5)</sup>また本稿は奈良大学地理学科の科目として1986年度以来開講している「応用実習」を筆者が担当した際の経験によるところが大きい。

## Ⅱ. 格子型データの処理・地図表示システム

### (1) 格子型データについて

ここでいう「格子型」の地理的データとは地形を記述するための標高の「格子点データ」や、メッシュ統計などの「メッシュデータ」をさす。これは画像処理やコンピュータ・グラフィックでいう「ラスタ型」のデータに相当する。各種統計数値の空間的分布を記述するための地域メッシュシステムは、集計単位の形が正方形や長方形をしており地表上の実質的な境界と整合しないなどの欠点が指摘されてきたにもかかわらず、特にコンピュータによる地理データ処理において広く採用されてきた。またコンピュータで扱うための地

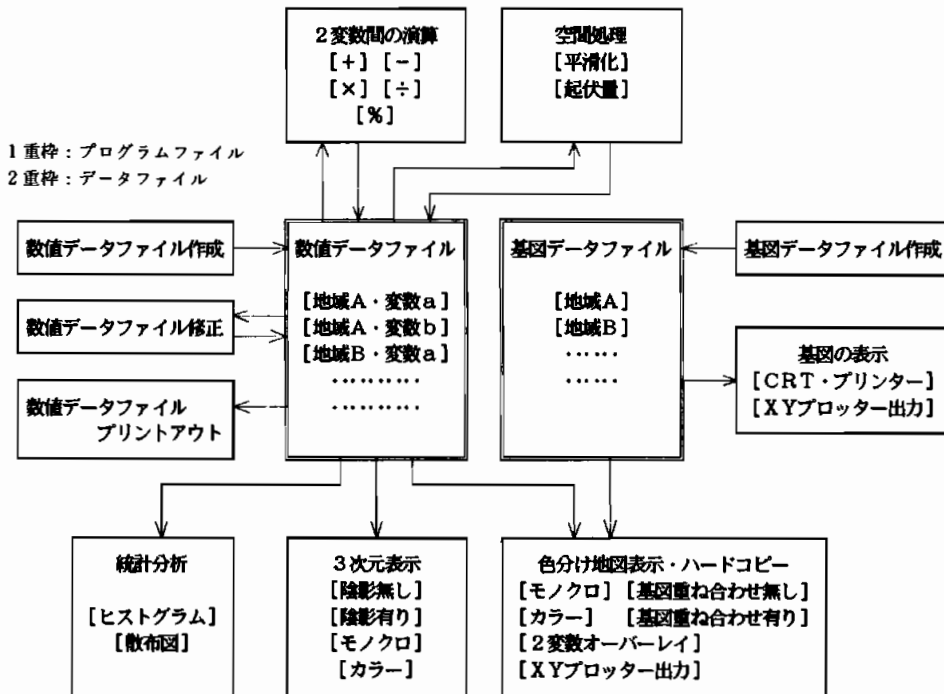
形のモデル（デジタル地形モデル）として、従来の等高線など代わりに規則的な格子の交点における標高等を行列形式でデータ化した「高度マトリックス Altitude Matrix」があり、国土数値情報の標高ファイルなどでもこの形式が採用されている。

このように格子型のデータはとりわけコンピュータを用いた地理データの処理において広く採用されてきた。その理由として、格子型のデータは、平滑化や微分処理といった空間的なデータ処理に有利であり、<sup>6)</sup> またオーバーレイ技法を用いて異なる変数を重ね合わせて分析・表示するのが容易であるといったことがあげられる。<sup>7)</sup> このようにメッシュ統計などの格子型データを扱う際に生じるメリットやデメリットについては、いろいろな機会に指摘されてきたが、地理学教育におけるパソコンの活用に関していえば、格子型のデータを扱うメリットとして、次の諸点を指摘し得る。

- i) 格子型のデータは2次元の行列 Matrix の形で与えられるため、キーボードからの数値の入力を原則とし、地図データの入力の際に通常必要とされるデジタイザやスキャナーなどの画像入力装置を特に必要としない。これらの装置はパソコンの標準的な機器構成からみて特殊かつ高価なものとみなされるのが普通で、このことが地理的なデータ処理へのパソコンの応用を普及させる上での障害となってきたと思われる。<sup>8)</sup>
- ii) ブロックダイアグラムなど3次元的なディスプレイが比較的単純なプログラミングで可能である。この他にも複数の変数のオーバーレイ表示なども含めて、単一の地理的データのセットから多様な視覚的表示が可能であることを学生に理解させるための素材とすることができる。

## (2) システムの概要

第1図に示したのが格子型データから主題図を作成するシステムの概要である。データ



第1図 格子型データのシステムの概要

入力・データファイル作成のためのプログラムをリスト1に示す。ここでの要点はデータファイルをプログラムファイルとは別個に保存することである。BASIC言語を用いたシステムでは“DATA”文を使用してプログラムの中にデータを書き込んだものを書き見受ける。確かにこの流儀はBASICに付属したスクリーン・エディタを活用できるため、データの編集には便利である。しかしパソコンを活用する様々な局面において「データ（のセット）」と「データを処理するための手順」とは区別されるのが通例であり、プログラムとデータが融合した状態は学習者にとって好ましくない。さらに、プログラムから独立した形でデータを蓄積してゆくことにより、データが蓄積した後新たな処理法を実行するためのプログラムを適用することができ、システムに発展性をもたせることができる。

このプログラムにより作成されるデータファイルの様式を第2図に示す。ファイルの冒頭にデータの属性を記述するためのヘッダー情報として「地域名」、「変数名」、「変数の単位」、「データのタイプ」、「X軸方向の要素数-1」、「Y軸方向の要素数-1」、「地域全体のX軸方向の長さ」、「地域全体のY軸方向の長さ」が記入される。ここで「データのタイプ」とは格子型のデータの数値が格子の交点ごとに与えられるか（格子点型）、格子のマス目ごとに与えられるか（メッシュ型）を識別するための数値コードである。このヘッダー情報の中のX軸方向・Y軸方向それぞれの要素数に従って $(XD+1) \times (YD+1)$ 個の数値を書き込み、また読み出すわけである。

さて、作成したデータを図化して表示する前に、変数の値の分布を検討するため、ヒストグラムなどを描かせたほうがよい。手作業に頼っている段階では、メッシュ統計などを扱う際などデータ量が大きいこともあってこのような手順は省略されがちであったが、計算機による自動化が可能になった段階では地図の色分けなども数値の分布をふまえて行うように習慣づける必要がある（リスト2・第7図）。

ディスクから読み込んだ格子型データを色分け地図として表示するためのプログラムをリスト3に示す。ここではアナログディスプレイを用い、16色の表示が可能である場合を想定している。これについても一度読み込んだデータを記憶して何度も試行し、数値の分布を最もよく表す境界値の設定を選択できるようになっている。カラー頁第8図に示した

地域名	変数名	数値の 単位	データの タイプ	X軸方向 要素数-1	Y軸方向 要素数-1	X軸方向 長さ	
RN\$	VN\$	UNIT\$	DT	XD	YD	XS	

Y軸方向 長さ	数値	数値	.....	数値	数値
YS	V(0,0)	V(1,0)		V(XD-1, YD)	V(XD,YD)

第2図 格子型データのシステムの数値データファイルの様式  
(アルファベットはリスト1中の変数名)

出力例は上記のものにさらにデジタイザを用いて作成した基図（線から構成される）をオーバーラップして表示したものである。また第9図では2変数の分布をオーバーレイ表示している。さらにXYプロッターを出力媒体として同じ構造のデータから作成した色分けメッシュ統計図を第10図に示す。

さて、たとえば若年人口数と総人口数との除算により若年人口率を算出してその分布を図示するといったことを行う機会が多い。複数の変数間の演算処理において、手作業に比べ精確化・省力化を図ることができるということは、コンピュータを用いて地理的データを入力・編集することの大きな利点である。このシステムでも複数の変数間の演算処理を行うためのプログラムを用意している。さらに、さきに指摘したように、格子型地理データのメリットとして各種の空間的処理が比較的単純なプログラミングによって可能となることがあげられる。ここでは空間的処理のひとつとして移動平均法による平滑化を行うためのプログラムをリスト4に示した。変数間の計算処理や空間的処理の結果には、新たに変数名などが与えられて直接入力して作成したデータファイルとまったく同じ様式でディスクに保存される。処理結果の地図表示や3次元表示も他の変数同様に行うことができる。カラー頁第11図・第12図はそれぞれ地形図の等高線から読み取った標高データとそのデータマトリックスに移動平均法による平滑化をほどこしたデータを3次元表示したものである。

### Ⅲ. ポリゴン型データの処理・地図表示システム

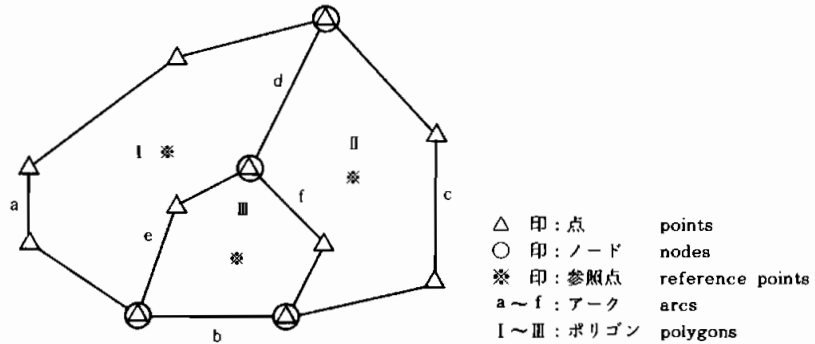
#### (1) ポリゴン型データについて

現在入手可能な地理的統計データは、多くの場合市区町村などの行政区画単位で与えられる。このようなデータを手作業により色分け統計地図として表示するには、まず対象区域の白地図を用意し、データが絶対量を表す変数ならばこれを比率尺度に変換した上で、白地図上の単位地区すなわちポリゴン（多角形）を彩色するという手順をふむ。主題図のこの様式はコロプレスマップ Choropleth Map と呼ばれる。コンピュータを用いてこの手順を実行するためには、数値データの他に図形データとしての白地図データが不可欠であり、システムの中に白地図データ生成のためのプログラムを用意する必要がある。さきに扱った格子型データのようにデータの行列 Matrix 中の位置がそのまま空間的位置を表すというわけにはいかないのである。

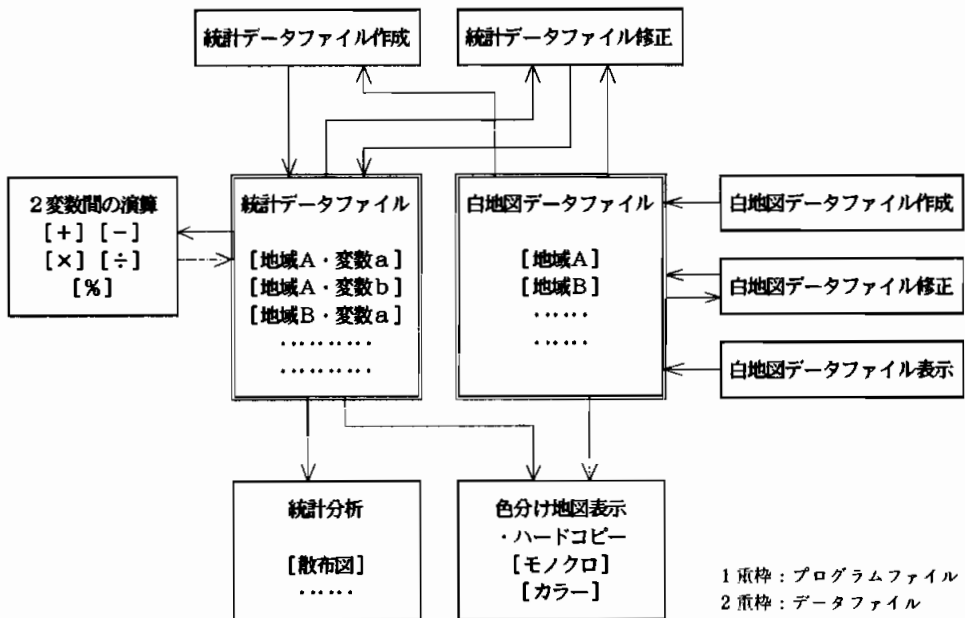
一般に白地図データは境界線のネットワークを構成する各点の座標値から構成されるが、それ以外にも各ノードと各アーク、また各アークと各ポリゴンとの位置関係などトポロジカルな情報が必要に応じて含まれる（ノード、アーク、ポリゴンなどネットワークのトポロジカルな構成要素については第3図参照）。ただし本システムでは、複雑な操作を避けるため、色分け統計地図を描くのに最小限必要なものとしてネットワーク各点の座標値のほか、色を塗り始めるための参照点（レファレンスポイント）を含めているに過ぎない。白地図データは図形データであるのでその入力にはデジタイザなどの図形入力機器が必要である。ここでは白地図データ入力のために小型のデジタイザ（タブレット）を使用し、各点のトポロジカルな属性を区別するため、カーソルは4ボタンのものとする。白地図を構成する各点の座標値を方眼紙上で読み取り、キーボードから入力する方法も考えられるが、操作の複雑さゆえに実用性に乏しいといわねばならない。

#### (2) システムの概要

第4図にポリゴン型データから主題図を作成するシステムの概要を示す。ここで重要と



第3図 境界線ネットワークのトポロジカルな構成要素



第4図 ポリゴン型データのシステムの概要

なるのは、ひとつの白地図をベースとし、その上に異なる種類の統計データを入力できるように、白地図データと統計データを別個のファイルとして保存することである。ここでは地名データと白地図データとは一体としてファイルとし、統計データは白地図データ・ファイルとは別の、また変数ごとに別個のファイルとするようなシステム構成となっている。

白地図データファイル作成のプログラムをリスト5に示す。このプログラムを用いて白地図を入力する作業の手順としては、最初にキーボードから地域名や単位地区数と一連の単位地区名とを入力する。次にデジタイザの板上に貼付けた白地図からカーソルで境界線を構成するネットワークを入力してゆく。そしてさらにデジタイザを用いて各单位地区の領域内に参照点（レファレンスポイント）を入力する。

ここでの要点は、ネットワークを入力する際、ノード（始点、終点、および分岐点）の

地域名 RN\$	単位 地区数 NU	ネットワーク 総点数 NP	左上端 X座標 XMIN	左上端 Y座標 YMIN	右下端 X座標 XMAX	右下端 Y座標 YMAX	<input checked="" type="checkbox"/>
-------------	-----------------	---------------------	--------------------	--------------------	--------------------	--------------------	-------------------------------------

単位 地区名 UNAMS(1)	単位地区 参照点 X座標 UX(1)	単位地区 参照点 Y座標 UY(1)	<input checked="" type="checkbox"/>	単位 地区名 UNAMS(2)	単位地区 参照点 X座標 UX(2)	単位地区 参照点 Y座標 UY(2)	<input checked="" type="checkbox"/>	.....
-----------------------	-----------------------------	-----------------------------	-------------------------------------	-----------------------	-----------------------------	-----------------------------	-------------------------------------	-------

.....	単位 地区名 UNAMS (NU)	単位地区 参照点 X座標 UX(NU)	単位地区 参照点 Y座標 UY(NU)	<input checked="" type="checkbox"/>	ネットワーク 構成点 X座標 X(1)	ネットワーク 構成点 Y座標 Y(1)	フラグ F(1)	<input checked="" type="checkbox"/>	.....
-------	----------------------------	------------------------------	------------------------------	-------------------------------------	------------------------------	------------------------------	-------------	-------------------------------------	-------

.....	ネットワーク 構成点 X座標 X(NP-1)	ネットワーク 構成点 Y座標 Y(NP-1)	フラグ F(NP-1)	<input checked="" type="checkbox"/>	ネットワーク 構成点 X座標 X(NP)	ネットワーク 構成点 Y座標 Y(NP)	フラグ F(NP)	<input checked="" type="checkbox"/>
-------	---------------------------------	---------------------------------	----------------	-------------------------------------	-------------------------------	-------------------------------	--------------	-------------------------------------

第5図 ポリゴン型データのシステムの白地図データファイルの様式  
(アルファベットはリスト5中の変数名)

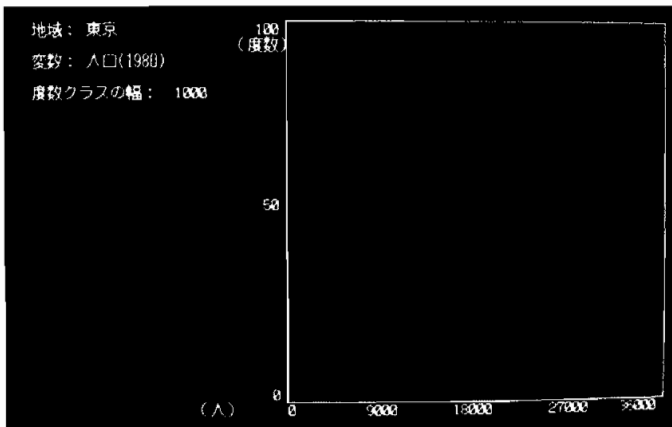
地域名 RN\$	変数名 VN\$	数値の 単位 UNIT\$	単位 地区数 NU	<input checked="" type="checkbox"/>	数値 V(1)	<input checked="" type="checkbox"/>
-------------	-------------	---------------------	-----------------	-------------------------------------	------------	-------------------------------------

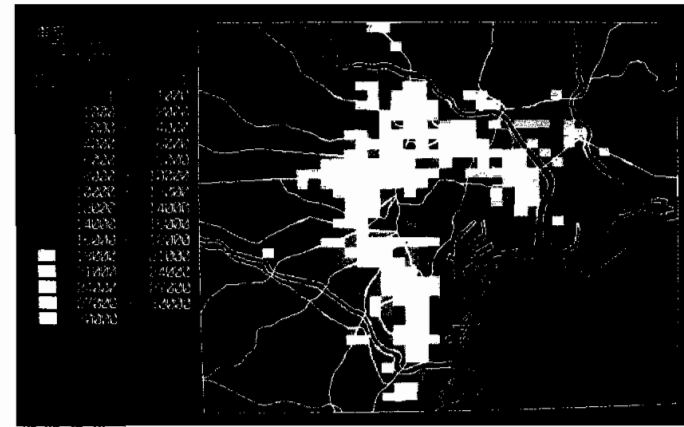
数値 V(2)	<input checked="" type="checkbox"/>	数値 V(3)	<input checked="" type="checkbox"/>	.....	<input checked="" type="checkbox"/>	数値 V(NU)	<input checked="" type="checkbox"/>
------------	-------------------------------------	------------	-------------------------------------	-------	-------------------------------------	-------------	-------------------------------------

第6図 ポリゴン型データのシステムの統計データファイルの様式  
(アルファベットはリスト6中の変数名)

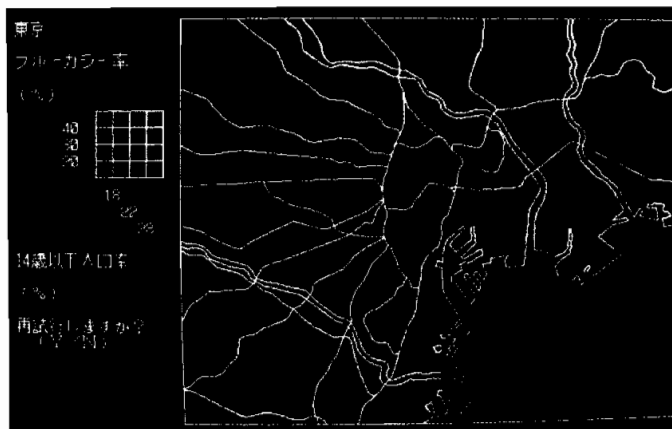
座標を統合することである。ノードを入力する場合、同一の点を複数回与えなければならぬが、手でカーソルを操作する際に生じる誤差の関係上、デジタイザで同一の座標値を



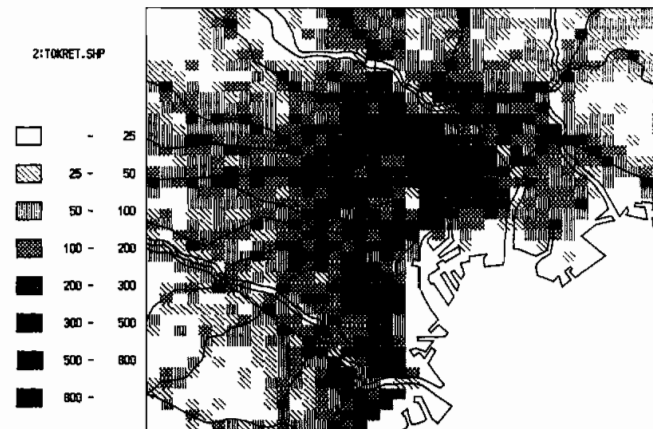
第7図 ヒストグラムの表示 (地域は東京大都市圏、変数は人口)



第8図 格子型データの地図表示 (地域は東京大都市圏、変数は人口)

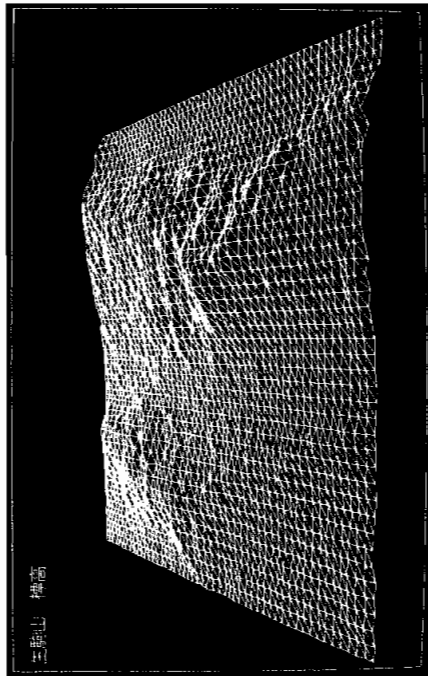


第9図 格子型データの2変数オーバーレイ表示 (地域は東京大都市圏、変数はブルーカラー率および14歳以下人口率)



第10図 X Yプロッターを用いた格子型データの地図出力 (地域は東京大都市圏、変数は小売商店数)

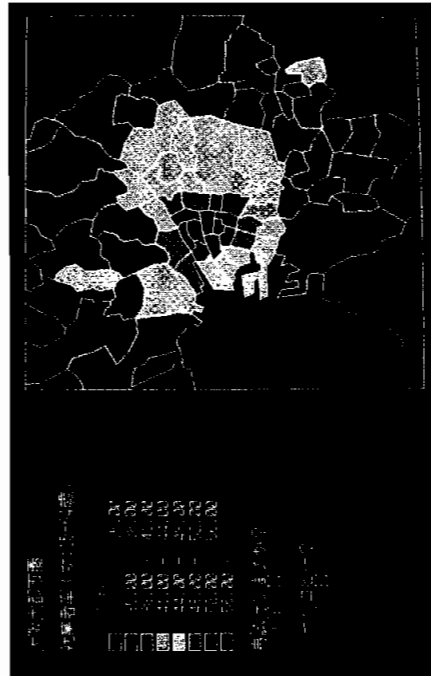




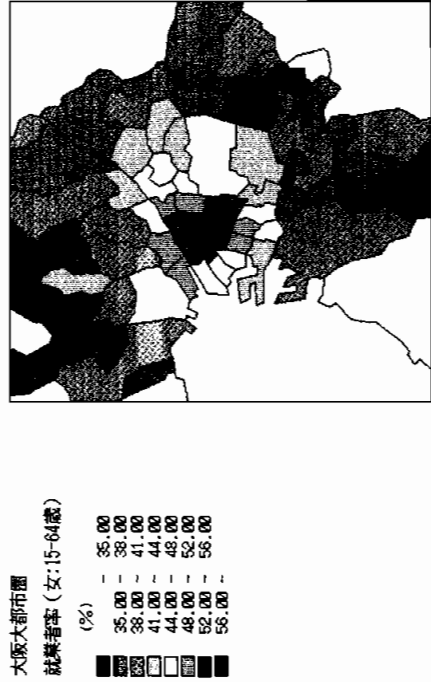
第11図 格子型データの3次元表示 (地域は生駒山、変数は標高)



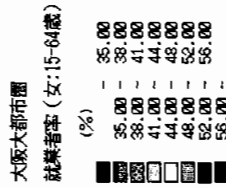
第12図 格子型データの3次元表示 (第11図のデータを平滑化したもの)



第13図 ポリゴン型データの色分け表示 (地域は大阪大都市圏、変数は女性就業率)



第14図 色分け統計地図のプリンタ出力 (地域、変数は第13図と同じ)



繰り返し入力することは、事実上不可能である。しかしノードつまりアークの端点を一致させなければ、領域をペイントする際に「境界線洩れ」のような不都合を生じる。この点を解決するために、このプログラムでは3mm以内で近接したノードの座標値を同一のものに置き換える手順を、自動的に行うようにしている。またこのシステムでは（厳密にはそうでない地形図なども含めて）入力する白地図は長方形または正方形であることを前提としている。しかしデジタイザの座標読み取り板の座標軸に正しく平行・垂直に地図を貼付することは時として困難であり、結果として入力された白地図データで地図の造作に対し枠線が傾いたり、描く範囲が意図するものとずれるといったことになりかねない。この点を解決するためにこのプログラムでは枠線の傾きをあらかじめ補正して地図上の各点の座標値を記録してゆく方式を採用している。

このプログラムによって作成される白地図データファイルの様式を第5図に示す。ヘッダー情報として「地域名」、「単位地区数」、「(ネットワークを構成する)点の総数」、「図の左下の点の座標値」、「図の右上の点の座標値」が記入される。白地図データをリードアウトする際にはこの単位地区数に従って単位地区名と単位地区の位置を示す参照点の座標値を読み出す。さらに点の総数にしたがってネットワークを構成する点の座標値を、その点が終点か否かを識別するためのフラグとともに読み出していくのである。

こうして作成した白地図データファイルをベースに、統計データを入力する。そのためのプログラムをリスト6に示す。手順としては白地図データファイルからヘッダーと地名データだけを読み出し、これに基づいて各単位地区についての数値を入力する。数値をディスクに保存する際、ヘッダー情報として「地域名」、「変数名」、「変数の単位」をファイル冒頭に記入する。これは白地図データファイルを介さずに統計データファイル进行操作する場合、必要となる。統計データファイルの様式を第6図に示す。

ここで色分け統計地図と呼んでいるコロプレスマップは、単位地区の面積や形が一定しないため、人口や商業販売額といった絶対的な数量の分布を表現するには適さず、専ら人口密度や1店当り販売額といった相対的な指標の分布を表現するのに用いられる。入力する統計数値のほとんどが絶対的な数値であることを考えると、これを相対的な指標へと変換する必要がある。具体的には、複数の変数間で演算を行い、構成比などを算出してこれを新たな変数としてデータファイルを作成することになる。2つのデータファイルから変数値を読み出し、構成比をパーセントで算出して、ディスク上に新たなデータファイルを作成するためのプログラムをリスト7に示した。

ディスク上の白地図データファイルと統計データファイルとを読み出して色分け統計図を表示するプログラムをリスト8に示す。7色までしか表示できないデジタルディスプレイを使用する場合を考慮し、またカラーハードコピーを取ることができるよう、中間色モードは使用せず、カラータイルを組むことによって中間色を表示している。ここでの要点はドットインパクトプリンタを用いてカラーのハードコピーを取る際に、ディスプレイ上のものとは異なるカラータイルをカラーハードコピーのために用意していることである。ディスプレイ上ではR(赤)・G(緑)・B(青)の組合せによって各色を合成するが、カラープリンタはC(シアン)・M(マゼンタ)・Y(黄)の組合せで各色を表現する。このため画面ハードコピーを用いてもディスプレイ上で最も適した色の組合せがそのままプリンタ上で最適なものとはならないのである(カラー頁第13図・第14図)。

#### IV. 今後の課題

本稿で紹介したシステムは、比較的安価な機材を用い、プログラミングの知識のない学生にも手軽にコンピュータマッピングが行えることを主たる目的としてデザインされている。そこでの要点は最小限の手間で結果の地図を得ることである。ただ手軽さという点で、このシステムで問題なのはキーボードからのデータの入力・編集の手順である。カードによる入力や、ラインエディタによる編集が当然であった以前とは異なり、ワードプロセッサやパーソナルコンピュータから入門した初心者はスクリーンエディタによる編集を当り前のこととしている。このシステムでもスクリーンエディタに近い使い勝手でデータ入力・編集ができることが望ましい。

このシステムはスタンドアロン版のDISK BASICで書かれているが、そのため他のシステムとのデータの互換性に乏しく、また直接データファイルをリストアウトすることができないといった欠点がある。このシステムをMS-DOS版のDISK BASIC上に再構築するならば、いくつかのメリットが期待できよう。まず第一にMS-DOS上のデータ互換性を利用して市販のエディタ・表計算・データベースソフトなどをデータ入力・編集に利用することが考えられる。また同様に作図の段階で各種のグラフィックソフトを利用することが考えられる。全体としてシステムの中でも市販・既製のソフトウェアで代替できる部分はできるだけ置き換えるようにすることが、プログラミングの手間を省き使いやすさを高めることになろう。さらにまたMS-DOSを採用することで、TYPEコマンドやエディタを用いてデータファイルを直接リストアウトできるようになるが、地理的データ特に地図データの構造がどのようなものかを示す教材として有効であろうと思われる。

II章で紹介した格子型データのシステムについては、扱うことのできる区域が長方形または正方形に限られるという問題がある。いうまでもなく地理学の個々の研究においてはそれぞれの目的に適する対象区域を設定するのが原則であり、その形が長方形・正方形に限定されるべきではない。第7図の人口分布ヒストグラムを描く際、もし海上のメッシュまでが単位として含まれるとしたら不都合なことといわざるを得ない。ポリゴン型データのシステムにおいてはユーザー自身の手による地区設定が可能である。格子型データのシステムにおいても研究目的に適する(格子点・メッシュの集合としての)地区の設定を任意に行うことができ、これを単位として各種のデータ処理を実行することができるが望ましい。

III章で紹介したポリゴン型データのシステムに関しては、白地図データファイルの境界線ネットワークがデータ構造において相互に関連のないアークの羅列でしかないという問題がある。XYプロッターなどを用いてポリゴン内部をハッチングするためには、少なくともひとつのポリゴンがどのアーク群から構成されるかについての情報が必要である。しかしそのために入力時に複雑な操作を要求すれば、かえってシステム自体の利便性を損うことになりかねない。<sup>9)</sup>従って境界線ネットワークを構成する各点および線についての最小限の情報から、ネットワーク構成要素間のトポロジカルな関係をコンピュータが自動的に生成するための手順が必要となる。この手順を実現するためのアルゴリズムがBurrough (1986) に示されている。<sup>10)</sup>

最後に本稿で紹介した格子型・ポリゴン型両システムに共通することであるが、定量的なデータの分布を地図表示するためのこれらのシステムで、土地利用など定性的なデータ

を扱うことの問題がある。簡便な解決法として、たとえば土地利用の各カテゴリーに数値コードを与えて数値データと同じ様式のファイルとすることも考えられる。しかしその場合、凡例を表現をどうするか、数値コードの対象表をどこに保存するのかといった問題が生ずる。定性的な地理的データの処理・地図表示で必要とされるデータ構造は、定量的な地理的データの場合とは微妙に異なっている。両者がどの範囲までデータ構造や処理手順を共有することができるのかについて今後検討を進めてゆきたい。

〔付記〕 本稿での研究には、科学研究費補助金・奨励A課題番号62780276を使用した。

#### 註および参考文献

- 1) パーソナルコンピュータを利用した大学の地理学教育のためのテキストとして次のものがある。  
野上道男・杉浦芳夫 (1986)『パソコンによる数値地理演習』, 古今書院
- 2) 地理学研究・教育に役立つパーソナルコンピュータ用のプログラム開発を扱った論文として以下のものがあり、いずれもプログラミング言語としてBASICを用いている。  
東賢次・江口利香 (1983)「パーソナルコンピュータと地理情報処理」, 東北地理, 35-3, pp.110-115  
松井秀郎 (1984)「地理学研究のためのアプリケーションプログラム—単回帰分析法—」, 地域研究, 25-1, pp.28-37  
吉川博之 (1984)「パーソナルコンピュータの地理学的利用—修正ウィーバー法の改善—」, 地域研究, 25-1, pp.38-41  
また雑誌「地理」の特集号にも地理学のための応用プログラムが多数掲載されている。これらもBASIC言語で書かれている。  
金安岩男・久保幸夫・野上道男・寄藤 昂編 (1985)『地理とコンピュータ特集号』, 地理, 30-3増刊  
久保幸夫・野上道男・寄藤 昂・金安岩男編 (1986)『ジオ・グラフィックス特集号』, 地理, 31-9増刊
- 3) 大型計算機を用いた地理的データ (特に地図データ) の処理・図化のためのシステムが既に各種あって実用に供されており、「地理情報システム」「コンピュータマッピングシステム」などと呼ばれている。これらの多くは業務用として地図製作会社の他、大量の図面を扱う地方自治体や電力・ガス会社などで採用されている。  
国土庁計画・調整局編 (1986)『国土情報シリーズ6・地理情報システム』大蔵省印刷局  
地理学研究のツールとして開発されたものとしては久保幸夫らの“ALIS”が知られる。本稿ではパーソナルコンピュータを利用するという局面に限定して考察を進める。  
久保幸夫 (1980)「地理情報処理の動向」, 人文地理, 32-4, pp.328-350
- 4) パーソナルコンピュータのグラフィック機能を地図表示のために活用する試みを扱ったものとして以下のものがある。  
東賢次 (1984)「パーソナルコンピュータによる地図解析支援ツールの開発」, 地域研究, 25-1, pp.19-27  
白岩隆己・田中中央・藤森博美 (1986)『実例パソコン立体地図』, 講談社  
K. Westland 編 (1987)『パソコンマッピング入門—地図の図形処理—』, 日刊工業新聞社  
荒井良雄・茅野 基 (1987)「パーソナルコンピュータによる地域メッシュ統計の利用」, 昭和61年度文部省科学研究費成果報告書 (研究代表者・山口岳志)『国土情報の高度化に関する研究』

## 所収

杉谷 隆 (1987) 「パーソナルコンピュータ上における国土数値情報の利用」, 『国土情報の高度化に関する研究』(前掲) 所収

英国におけるマイクロコンピュータを利用した主題図作成システム開発の試みが次の特集号にくつか紹介されている。

V.Gardiner and D.J.Unwin (Guest Editors) (1985): "Thematic Mapping Using Microcomputers". Computers & Geosciences, 11-3, Special Issue

またパーソナルコンピュータを地理的データ処理・地図表示のために活用する上でのデータ処理能力やグラフィック機能などハードウェア環境およびソフトウェア環境について考察したものとして次の論文がある。

大竹義則 (1985) 「マイクロコンピュータによる小規模地理情報システム—その利用と問題点—」, 地理科学, 40-2, pp.100-110

## 5) ハードウェア構成は次の通り

パーソナルコンピュータ本体: PC-9801 VM21

アナログカラーディスプレイ: PC-KD854

カラードットインパクトプリンタ: PC-PR201H

(以上, 日本電気製)

デジタイザ(タブレット): MITABLET-II (グラフテック製)

XYプロッタ: DXY-880A (Roland DG製)

## 6) 平滑化・微分処理などは空間フィルタリング処理と総称され, それぞれコンピュータ画像処理においてノイズの除去・エッジの検出を行うための技法である。

田村秀行監修・日本工業技術センター編 (1985) 『コンピュータ画像処理入門』, 総研出版, pp.103, 118-125

これを地形モデルに当てはめた場合, 平滑化は接峰面など平滑面の生成に, 微分処理は境界線の抽出に対応する。

野上・杉浦 (前掲書), p.43

## 7) 主題図を表示作成するためのデータの構造についてラスタ型(本稿では格子型)・ベクター型(本稿ではポリゴン型)それぞれの長所・欠点を論じたものとして次の文献がある。

P.A.Burrough (1986): "Principles of Geographical Information Systems for Land Resources Assessment". Oxford University Press, New York, pp.11-38

## 8) もっとも図形・画像関係の周辺機器は入力装置のデジタイザやスキャナー, 出力装置のXYプロッターなどいずれも10万円代の製品が既にあり, 本格的で実務に耐えるものを要求しない限り必ずしも高価とはいえなくなっている。障害はむしろこれらの機器を地理学のために活用するソフトウェアの不足にあるといえよう。

## 9) Burrough も「すべてのポリゴンを正しく時計回りに入力し, 各アークの左右のポリゴンを明示するといったことをオペレーターに要求するシステムをしばしば見受けるが, このようなシステムを開発した者はたぶん長時間デジタイザに向かって作業したことがなく, このようなシステムがいかにエラーを生じやすいかについて考えたこともないのだろう」と指摘している。

P.A.Burrough (1986) (前掲書), p.29

## 10) P.A.Burrough (1986) (前掲書), p.29-32

### Summary

The present paper demonstrates the possibility of automated thematic mapping using microcomputers. The author suggests that in geographical teaching at universities and colleges it may be important to utilize microcomputers which became more efficient and less expensive in recent years. The paper presents two thematic mapping systems; one treats grid-formed data, while the other visualizes polygonal data, both of which are designed and programmed by the author. In chapter II, the way how geographical data in grid formats (i.e. census data provided in the areal grid format, altitude matrix derived from topographical information, etc.) can be manipulated and displayed in 2- or 3-dimensional styles are demonstrated. In chapter III, how to visualize polygonal data (ex. census data provided for administrative districts) is discussed, and it is emphasized that polygonal data systems should treat digitized base-map data and numerical data separately. In the final chapter, some points which are to be improved in the future are discussed.

## リスト1

```

1000 'ラスタ-マップのための数値データ・ファイルの作成
1010 'PROGRAM NAME : "ALTAIR"
1020 '
1030 DIM V(100,100)
1040 CLS 3
1050 INPUT "データ・ボリウムのトライア番号 ";NM
1060 FILES NM
1070 PRINT
1080 INPUT "新しいデータ・ファイル名 ";DNM: PRINT
1090 INPUT "座標名 ";RNF: PRINT
1100 INPUT "座標名 ";VNB: PRINT
1110 INPUT "座標の単位 ";UNITS: PRINT
1120 INPUT "データのタイプ: (格子点数=0, メッシュ型=1) ";DT: PRINT
1130 INPUT "X軸方向の座標数 ";XD: XD=XD-1: PRINT
1140 INPUT "Y軸方向の座標数 ";YD: YD=YD-1: PRINT
1150 INPUT "座標全体のX軸方向の長さ (km) ";XS: PRINT
1160 INPUT "座標全体のY軸方向の長さ (km) ";YS
1170
1180 NXD=FIX(XD/20)
1190 FOR Y=0 TO YD
1200 FOR N=0 TO NXD
1210 CLS
1220 FOR X=N*20 TO N*20+19
1230 IF X=XD GOTO 1280
1240 PRINT "X=";X;"Y=";Y
1250 LOCATE 30,X=N*20
1260 INPUT "VALUE";V(X,Y)
1270 NEXT X
1280 '
1290 PRINT "OK? (Y/N/E)"
1300 B$=INKEY$
1310 IF B$="" THEN GOTO 1300
1320 IF B$="N" OR B$="E" THEN GOSUB +CORRECT
1330 IF B$="Y" OR B$="E" GOTO +FINISH
1340 NEXT N
1350 NEXT Y
1360
1370 OPEN DNM FOR OUTPUT AS #1
1380 WRITE #1,RNF,VNB,UNITS,DT,XD,YD,XS,YS
1390 FOR Y=0 TO YD
1400 FOR X=0 TO XD
1410 WRITE #1,V(X,Y)
1420 NEXT X
1430 NEXT Y
1440 CLOSE #1
1450
1460 +FINISH
1470 END
1480
1490 '
1500 +CORRECT
1510 PRINT "修正してください。"
1520 INPUT "X=";X
1530 IF X=N*20 AND X<N*20+19 AND X<XD GOTO 1650
1540 PRINT "Invalid X value " : GOTO 1520
1550 LOCATE 30,CSRLIN-1: INPUT "VALUE";V(X,Y)
1560
1570 CLS
1580 FOR X=N*20 TO N*20+19
1590 IF X=XD GOTO 1620
1600 PRINT "X=";X;"Y=";Y;"V=";V(X,Y)
1610 NEXT X
1620 RETURN 1290

```

## リスト2

```

1000 'ラスタ-型数値データのヒストグラムの表示
1010 'PROGRAM NAME : "ABCURBS"
1020 '
1030 DIM V(100,100),CLASS(200)
1040 CLS 3
1050 PRINT "座標のヒストグラムを表示します"
1060 PRINT : INPUT "データ・ボリウムのトライア番号 ";N
1070 PRINT : FILES N
1080 PRINT : INPUT "数値データ・ファイル名 ";DNM
1090
1100 OPEN DNM FOR INPUT AS #1
1110 INPUT #1,RNF,VNB,UNITS,DT,XD,YD,XS,YS
1120 FOR Y=0 TO YD
1130 FOR X=0 TO XD
1140 INPUT #1,V(X,Y)
1150 NEXT X
1160 NEXT Y
1170 CLOSE #1
1180
1190 CLS
1200 PRINT "座標名: ";RNF: PRINT
1210 PRINT "座標名: ";VNB: (" ";UNITS:)"
1220 PRINT : PRINT
1230 PRINT "よろしいですか? (Y/N) "
1240 K$=INKEY$
1250 IF K$="" GOTO 1240
1260 IF K$="N" OR K$="E" GOTO 1040
1270 CLS
1280 GOSUB +MAXMIN
1290
1300 CLMAX=0
1310 FOR N=0 TO 100
1320 CLASS(N)=0
1330 NEXT N
1340 CLS
1350 PRINT "座標を表示する範囲を設定してください"
1360 PRINT : LOCATE 20 : PRINT "最大値: ";MAX
1370 LOCATE 40,CSRLIN-1 : PRINT "最小値: ";MIN
1380 PRINT : LOCATE 20 : INPUT "上界 ";SMAX
1390 LOCATE 40,CSRLIN-1 : INPUT "下界 ";SMIN
1400 PRINT : PRINT
1410 PRINT "座標クラスの幅を設定してください"
1420 PRINT : LOCATE 20 : INPUT "座標クラスの幅: ";INTV
1430 PRINT : PRINT
1440 PRINT "各クラスは座標によくあいますか?"
1450 LOCATE 40,CSRLIN-1 : INPUT "(Y/N) ";ZEROS
1460 PRINT : PRINT
1470 PRINT "目盛り(2分割, 3分割または4分割)"
1480 LOCATE 40,CSRLIN-1 : INPUT "(2, 3 OR 4)";DV1
1490
1500 CL1=INT((SMAX-SMIN)/INTV)+1
1510 FOR Y=0 TO YD
1520 FOR X=0 TO XD
1530 IF (ZEROS="N" OR ZEROS="n") AND V(X,Y)<0 GOTO 1570
1540 IF V(X,Y)>INTV*(CL1+1) GOTO 1570
1550 V2=INT(V(X,Y)/INTV)
1560 CLASS(V2)=CLASS(V2)+1
1570 NEXT X
1580 NEXT Y
1590
1600 FOR N=0 TO CL1
1610 IF (N+1)*INTV<SMIN GOTO 1640
1620 IF N*INTV>SMAX GOTO 1640
1630 IF CLMAX<CLASS(N) THEN CLMAX=CLASS(N)
1640 NEXT N
1650
1660 CLS
1670 PRINT "座標の最大値は ";CLMAX;"です"
1680 PRINT : LOCATE 10 : INPUT "グラフ表示の上層を教えてください ";P$
1690 PRINT : LOCATE 10 : PRINT "目盛り(2分割または3分割)"
1700 LOCATE 40,CSRLIN-1 : INPUT "(2 OR 3)";DV2

```

```

1710 '
1720 CLS
1730 PRINT "座標: ";RNF
1740 PRINT : PRINT "座標: ";VNB
1750 PRINT : PRINT "座標クラスの幅: ";INTV
1760
1770 CONSOLE,,0 : SCREEN 3
1780 WINDOW [SMIN,-P$X]-(SMAX,0)
1790 VIEW (266,1)-(638,383),,7
1800
1810 GOSUB +PIGS
1820 IF DV1=4 THEN GOSUB +DIV1 ELSE IF DV1=3 THEN GOSUB +DIV3 ELSE GOSUB +DIV2
1830 IF DV2=3 THEN GOSUB +DIV23 ELSE GOSUB +DIV22
1840
1850 FOR N=0 TO CL1
1860 IF CLASS(N)=0 GOTO 1930
1870 IF (N+1)*INTV<SMIN GOTO 1920
1880 IF N*INTV>SMAX GOTO 1920
1890 X1=INTV*N : Y1=CLASS(N)
1900 X2=INTV*(N+1)
1910 LINE (X1,Y1)-(X2,0),2,BF
1920 NEXT N
1930
1940 LOCATE 0,14 : PRINT " 再試行しますか? " : PRINT " (Y/N) "
1950 K$=INKEY$
1960 IF K$="" GOTO 1950
1970 IF K$="Y" OR K$="y" THEN VIEW (0,0)-(638,399) : CLS 3 : GOTO 1290
1980 LOCATE 0,14 : PRINT " " : PRINT " "
1990
2000 WHILE [NKEY$="" ] WEND
2010 VIEW (0,0)-1639,599) : CONSOLE,,1 : CLS 3
2020 END
2030
2040
2050 +MAXMIN
2060 MAX=V(0,0) : MIN=V(0,0)
2070 FOR Y=0 TO YD
2080 FOR X=0 TO XD
2090 IF MAX<V(X,Y) THEN MAX=V(X,Y)
2100 IF MIN>V(X,Y) THEN MIN=V(X,Y)
2110 NEXT X
2120 NEXT Y
2130 RETURN
2140 '
2150 '
2160 +PIGS
2170 LOCATE 31,24 : PRINT SMIN;
2180 LOCATE 20,24 : PRINT "( ";UNITS:)"
2190 LOCATE 73,24 : PRINT USING "#####";SMAX;
2200 LOCATE 29,23 : PRINT 0
2210 LOCATE 25,1 : PRINT "座標";
2220 LOCATE 27,0 : PRINT USING "####";P$MAX;
2230 LOCATE 0,0
2240 RETURN
2250
2260 '
2270 +DIV1
2280 DVX=(SMAX-SMIN)/2+SMIN
2290 LINE (DVX,-P$IN)-(DVX,-P$AX),1,,#SHAATA
2300 LOCATE 52,24 : PRINT USING "#####";DVX;
2310 RETURN
2320 '
2330 '
2340 +DIV2
2350 DVY=S$MAX/2
2360 LINE (S$MIN,-DVY)-(S$MAX,-DVY),1,,#SHAATA
2370 LOCATE 25,11 : PRINT USING "#####";DVY;
2380 RETURN
2390 '
2400 '
2410 +DIV3
2420 DVX=(S$MAX-S$MIN)/3+S$MIN
2430 LINE (DVX,-P$IN)-(DVX,-P$AX),1,,#SHAATA
2440 LOCATE 44,24 : PRINT USING "#####";DVX;
2450 DVX=(S$MAX-S$MIN)*2/3+S$MIN
2460 LINE (DVX,-P$IN)-(DVX,-P$AX),1,,#SHAATA
2470 LOCATE 58,24 : PRINT USING "#####";DVX;
2480 RETURN
2490 '
2500 '
2510 +DIV23
2520 DVY=S$MAX/3
2530 LINE (S$MIN,-DVY)-(S$MAX,-DVY),1,,#SHAATA
2540 LOCATE 25,15 : PRINT USING "#####";DVY;
2550 DVY=S$MAX*2/3
2560 LINE (S$MIN,-DVY)-(S$MAX,-DVY),1,,#SHAATA
2570 LOCATE 25,7 : PRINT USING "#####";DVY;
2580 RETURN
2590
2600 '
2610 +DIV4
2620 DVX=(S$MAX-S$MIN)/2+S$MIN
2630 LINE (DVX,-P$IN)-(DVX,-P$AX),1,,#SHAATA
2640 LOCATE 52,24 : PRINT USING "#####";DVX;
2650 DVX=(S$MAX-S$MIN)/4+S$MIN
2660 LINE (DVX,-P$IN)-(DVX,-P$AX),1,,#SHAATA
2670 LOCATE 40,24 : PRINT USING "#####";DVX;
2680 DVX=(S$MAX-S$MIN)/4+3*S$MIN
2690 LINE (DVX,-P$IN)-(DVX,-P$AX),1,,#SHAATA
2700 LOCATE 64,24 : PRINT USING "#####";DVX;
2710 RETURN

```

リスト 3

```

1000 'マスター型デジタル・マップの色分け表示(18色)'
1010 'PROGRAM NAME : "SPICAS2"'
1020 '
1030 DIM V(100,100) : DIM COL(20) : DIM TH(20)
1040 COLOR ...
1050 COLOR=(0,BND) : COLOR=(1,BHT) : COLOR=(2,BHR) : COLOR=(3,BHR)
1060 COLOR=(4,BHTC) : COLOR=(5,BHQA) : COLOR=(6,BHBB) : COLOR=(7,BHD6)
1070 COLOR=(8,BHPO) : COLOR=(9,BHBP) : COLOR=(10,BHBP) : COLOR=(11,BHBP)
1080 COLOR=(12,BHBP) : COLOR=(13,BHBP) : COLOR=(14,BHBP) : COLOR=(15,BHBP)
1090 '
1100 '
1110 CLS 3
1120 INPUT "データ・ボリュームのドライブ番号 ";IN
1130 PRINT : FILES N
1140 PRINT : INPUT "階層データ・ファイル名 ";DN#
1150 '
1160 OPEN DN# FOR INPUT AS #1
1170 INPUT #1,RNS,VNS,UNITS,DT,XD,YD,XS,YS
1180 FOR Y=0 TO YD
1190 FOR X=0 TO XD
1200 INPUT #1,V(X,Y)
1210 NEXT X
1220 NEXT Y
1230 CLOSE #1
1240 '
1250 CLS 3
1260 '
1270 INPUT "クラス数(0 または 16)";CL1
1280 IF CL1<=8 AND CL1>16 GOTO 1270
1290 FOR CL=1 TO CL1
1300 IF CL=16 THEN COL(CL)=CL2-1 ELSE COL(CL)=CL-1
1310 NEXT CL
1320 '
1330 CLS
1340 PRINT "クラス間の境界線をえてください"
1350 PRINT
1360 FOR CL=1 TO CL1-1
1370 PRINT
1380 PRINT USING "クラス#";CL;PRINT "/";PRINT USING "クラス#";CL+1
1390 LOCATE 20,CSR(LN-1) : INPUT "境界線 ";TH(CL)
1400 NEXT CL
1410 CLS
1420 PRINT "小窓点以下を表示しますか?"
1430 PRINT : INPUT "(表示する=1, 表示しない=0) ";DC
1440 '
1450 '
1460 CONSOLE 0,25,0,0
1470 SCREEN 3,0,0,1
1480 CLS 3
1490 '
1500 '
1510 IF DC=1 THEN GOSUB +LEGENDE ELSE GOSUB +LEGEND1
1520 IF DT=0 THEN GOSUB +MESH2 ELSE GOSUB +MESH
1530 '
1540 PRINT : PRINT "再試行しますか?" : PRINT "(Y/N) "
1550 #M=INKEY#
1560 IF #M="" GOTO 1550
1570 IF #M="Y" OR #M="y" THEN VIEW (0,0)-(639,399),0 : GOTO 1250
1580 LOCATE 0,CSR(LN-2)
1590 PRINT " " : PRINT " "
1600 '
1610 NHILE INKEY#="" : WEND
1620 VIEW (0,0)-(639,399),0
1630 COLOR ...
1640 CONSOLE ..1 : CLS 3
1650 '
1660 END
1670 '
1680 '
1690 '
1700 +LEGEND1
1710 '
1720 WINDOW (0,0)-(639,399) : VIEW (0,0)-(639,399),0 : CLS
1730 CLS 3
1740 PRINT RNS
1750 PRINT VNS
1760 PRINT " (:UNITS:)"
1770 FOR CL=1 TO CL1
1780 IF COL(CL)=0 THEN LINE (0,50+(CL-1)*16)-STEP(18,12),15,B : GOTO 1800
1790 LINE (0,50+(CL-1)*16)-STEP(18,12),COL(CL),BF
1800 LOCATE 4
1810 IF CL=1 THEN PRINT USING " " - "####";TH(CL) : GOTO 1860
1820 IF CL=CL1 THEN PRINT USING "####" -";TH(CL-1) : GOTO 1860
1830 PRINT USING "####";TH(CL-1);
1840 PRINT " - ";
1850 PRINT USING "####";TH(CL)
1860 NEXT CL
1870 '
1880 RETURN
1890 '
1900 '
1910 +LEGEND2
1920 '
1930 WINDOW (0,0)-(639,399) : VIEW (0,0)-(639,399),0 : CLS
1940 CLS 3
1950 PRINT RNS
1960 PRINT VNS
1970 PRINT " (:UNITS:)"
1980 FOR CL=1 TO CL1
1990 IF COL(CL)=0 THEN LINE (0,50+(CL-1)*16)-STEP(18,12),15,B : GOTO 2010
2000 LINE (0,50+(CL-1)*16)-STEP(18,12),COL(CL),BF
2010 LOCATE 4
2020 IF CL=1 THEN PRINT USING " " - "###,##";TH(CL) : GOTO 2070
2030 IF CL=CL1 THEN PRINT USING "###,##" -";TH(CL-1) : GOTO 2070
2040 PRINT USING "###,##";TH(CL-1);
2050 PRINT " - ";
2060 PRINT USING "###,##";TH(CL)
2070 NEXT CL
2080 '
2090 RETURN
2100 '
2110 '
2120 +MESH
2130 '
2140 WINDOW (0,0)-(XS,YS)
2150 IF YS/XS<399/478 THEN LY=YS/XS+478 : VIEW (161,1)-(638,LY),15 : GOTO 2170
2160 LX=XS/YS+399 : VIEW (638-LX,1)-(638,398),15
2170 '
2180 X=XS/(XD+1) : YU=YS/(YD+1)
2190 FOR Y=0 TO YD
2200 Y1=Y*YU
2210 FOR X=0 TO XD
2220 X1=X*XU
2230 FOR CL=1 TO CL1-1
2240 IF V(X,Y)<TH(CL) THEN C=COL(CL) : GOTO 2270
2250 NEXT CL
2260 C=COL(CL1)
2270 LINE (X1,Y1)-STEP(XU,YU),C,BF
2280 NEXT X
2290 NEXT Y
2300 '
2310 RETURN
2320 '
2330 '
2340 +MESH2
2350 '
2360 WINDOW (0,0)-(XS,YS)
2370 IF YS/XS<399/478 THEN LY=YS/XS+478 : VIEW (161,1)-(638,LY),15 : GOTO 2390
2380 LX=XS/YS+399 : VIEW (638-LX,1)-(638,398),15

```

```

2390 '
2400 XU=XS/XD : YU=YS/YD
2410 FOR Y=0 TO YD
2420 Y1=(Y-.5)*YU : Y2=(Y+.5)*YU
2430 IF Y1<0 THEN Y1=0
2440 IF Y2>YS THEN Y2=YS
2450 FOR X=0 TO XD
2460 X1=(X-.5)*XU : X2=(X+.5)*XU
2470 IF X1<0 THEN X1=0
2480 IF X2>XS THEN X2=XS
2490 FOR CL=1 TO CL1-1
2500 IF V(X,Y)<TH(CL) THEN C=COL(CL) : GOTO 2530
2510 NEXT CL
2520 C=COL(CL1)
2530 LINE (X1,Y1)-(X2,Y2),C,BF
2540 NEXT X
2550 NEXT Y
2560 '
2570 RETURN

```

リスト 4

```

1000 'デジタル・マップ・データの平滑化'
1010 'PROGRAM NAME : "ANTARES1"'
1020 '
1030 DIM V(100,100),V2(100,100)
1040 CLS 3
1050 PRINT "マスター型データの平滑化を行います"
1060 PRINT : INPUT "データ・ボリュームのドライブ番号 ";IN
1070 PRINT : FILES N#1
1080 PRINT : INPUT "階層データ・ファイル名 ";DN#
1090 OPEN DN# FOR INPUT AS #1
1100 INPUT #1,RNS,VNS,UNITS,DT,XD,YD,XS,YS
1110 VNS=VNS*平滑率
1120 FOR Y=0 TO YD
1130 FOR X=0 TO XD
1140 INPUT #1,V(X,Y)
1150 NEXT X
1160 NEXT Y
1170 '
1180 CLOSE #1
1190 '
1200 FOR Y=0 TO YD
1210 IF Y=1 THEN Y0=Y-1 ELSE Y0=0
1220 IF Y=YD THEN Y2=Y+1 ELSE Y2=YD
1230 FOR X=0 TO XD
1240 IF X=1 THEN X0=X-1 ELSE X0=0
1250 IF X=XD THEN X2=X+1 ELSE X2=XD
1260 V2(X,Y)=(V(X0,Y0)+V(X,Y0)+V(X2,Y0)+V(X0,Y)+V(X,Y)+V(X2,Y)+V(X0,Y2)+V(X,Y2)+V(X2,Y2))/10
1270 NEXT X
1280 NEXT Y
1290 '
1300 CLS
1310 PRINT "階データ・ファイル名 ";DN#
1320 PRINT
1330 INPUT "階データ・ファイル名 ";DN#
1340 OPEN DN# FOR OUTPUT AS #2
1350 '
1360 WRITE #2,RNS,VNS,UNITS,DT,XD,YD,XS,YS
1370 FOR Y=0 TO YD
1380 FOR X=0 TO XD
1390 WRITE #2,V2(X,Y)
1400 NEXT X
1410 NEXT Y
1420 '
1430 CLOSE #2
1440 CLS
1450 END

```



## リスト5

```

1000 *** 統計地図作成のための白線画人力プログラム ***
1010 *** PROGRAM NAME : "MAP" ***
1020 DIM X(2000),Y(2000),F(2000),JN(400),JX(400),JY(400)
1030 DIM UNAME(100),UX(100),UY(100)
1040
1050 CLS 3
1060 INPUT "地域名 :";NM
1070 PRINT : INPUT "単位地区数 (100以下) :";NU
1080 IF NU<100 GOTO 1070
1090
1100 CLS
1110 FOR N=1 TO NU
1120 PRINT USING "地区番号 ##":N
1130 INPUT " 単位地区名 ";UNAME(N)
1140 PRINT
1150 NEXT N
1160
1170 PRINT "よろしいですか? (Y/N)"
1180 K$=INKEY$
1190 IF K$="" GOTO 1180
1200 IF K$="Y" OR K$="y" GOTO 1040
1210
1220
1230 OPEN "COM:N3" AS #1
1240 PRINT #1,"P"
1250
1260 CLS
1270 PRINT "デジタルで図の点を与えてください (Z)"
1280 INPUT #1,X1,Y1,F1 : PRINT
1290 PRINT "デジタルで図の点の下の点を与えてください (Z)"
1300 INPUT #1,XBL,YBL : PRINT
1310 PRINT "デジタルで図の点の右の点を与えてください (Z)"
1320 INPUT #1,XBR,YBR,F1
1330 PRINT : PRINT
1340 PRINT "どれかのキーを押してください"
1350 WHILE INKEY$="" : WEND
1360
1370 TH=ATN(YBR-YBL)/(XBR-XBL)
1380 A=COS(TH) : B=SIN(TH)
1390 DEF FNX(X1,Y1)=X1+Y1*B
1400 DEF FNY(X1,Y1)=2000-X1+B*Y1
1410
1420 XMIN=FNX(X1,Y1) : YMIN=FNY(X1,Y1)
1430 XMAX=FNX(XBR,YBR) : YMAX=FNY(XBR,YBR)
1440
1450
1460 CONSOLF ,0 : SCREEN 3 : CLS 3
1470 WINDOW (XMIN,YMIN)-(XMAX,YMAX)
1480 LX=XMAX-XMIN : LY=YMAX-YMIN
1490 IF LY/LX<398/478 THEN LY/LX=478 : VIEW (161,1)-(638,LY),T : GOTO 1610
1500 LX=LX/LT+398 : VIEW(638-LX,1)-(638,398),T
1510
1520 CLS 3
1530 N=1 : F(0)=8
1540 PRINT "デジタルで白線画の上の" : PRINT "各点を与えてください"
1550 PRINT : PRINT " 端点・分岐点 (1)"
1560 PRINT : PRINT " 中点 (2)"
1570 PRINT : PRINT " 終点 (3)"
1580 PRINT : PRINT " やり直し (Z)"
1590 PRINT : PRINT " 人力終わり (3)"
1600
1610
1620 INPUT #1,X1,Y1,F(N)
1630 X(N)=FNX(X1,Y1) : Y(N)=FNY(X1,Y1)
1640
1650 IF F(N-1)=8 AND F(N)=8 GOTO *EXIT
1660 IF F(N)=4 THEN GOSUB *CORRPT
1670
1680 IF F(N-1)<3 THEN LINE (X(N-1),Y(N-1))-(X(N),Y(N)),1
1690 IF F(N)=2 OR F(N)=8 THEN CIRCLE (X(N),Y(N)),10,2
1700 N=N+1 : GOTO 1610
1710 EXIT
1720 N=N-1
1730
1740 PRINT : PRINT "よろしいですか? (Y/N)"
1750 K$=INKEY$
1760 IF K$="" GOTO 1750
1770 IF K$="Y" OR K$="y" THEN CLS 3 : GOTO 1610
1780
1790 PRINT #1,"S"
1800 CLOSE #1
1810
1820
1830 CLS
1840 J=1
1850 FOR N=1 TO NP
1860 IF F(N)=1 GOTO 1860
1870 JN(1)=N : JX(1)=X(N) : JY(1)=Y(N)
1880 J=J+1
1890 NEXT N
1900 NJN=C(J)
1910
1920 FOR J=1 TO NJN-1
1930 FOR J=1 TO NJN
1940 IF (JX(1)-JX(J))^2+(JY(1)-JY(J))^2>1000 GOTO 1940
1950 X(JN(J))=X(JN(1))
1960 Y(JN(J))=Y(JN(1))
1970 CIRCLE (X(JN(1)),Y(JN(1))),30,4
1980 NEXT J
1990 NEXT J
2000
2010 PRINT : PRINT " どれかのキーを押してください"
2020 K$=INKEY$
2030 IF K$="" GOTO 2020
2040
2050 FOR N=1 TO NP
2060 IF F(N)=8 THEN F(N)+1 ELSE F(N)=0
2070 NEXT N
2080
2090 GOSUB *SKELETON
2100
2110 PRINT : PRINT " どれかのキーを押してください"
2120 K$=INKEY$
2130 IF K$="" GOTO 2120
2140
2150
2160 OPEN "COM:N3" AS #1
2170 PRINT #1,"P"
2180
2190 FOR N=1 TO NU
2200 CLS : PRINT
2210 PRINT " デジタルで単位地区の位置を与えてください (Z)"
2220 PRINT : PRINT USING " ## ":N;
2230 PRINT UNAME(N)
2240 INPUT #1,X1,Y1,F1
2250 UX(N)=FNX(X1,Y1)
2260 UY(N)=FNY(X1,Y1)
2270 CIRCLE (UX(N),UY(N)),30,3
2280 NEXT N
2290
2300
2310 CLS : PRINT : PRINT " よろしいですか? (Y/N)"
2320 K$=INKEY$
2330 IF K$="" GOTO 2330
2340 IF K$="Y" OR K$="y" THEN GOSUB *SKELETON : GOTO 2180
2350
2360 PRINT #1,"S"
2370 CLOSE #1
2380

```

```

2390
2400 CLS : PRINT : PRINT " データを保存しますか? (Y/N)"
2410 K$=INKEY$
2420 IF K$="" GOTO 2410
2430 IF K$="Y" OR K$="y" GOTO *DATASAVE
2440 GOTO *FINISH
2450
2460
2470 *DATASAVE
2480 CLS : PRINT : PRINT " 白線画データ・ファイルを作成します"
2490 PRINT : INPUT " データ・ボリュームのドライブ番号 ";NK
2500 FILES NK
2510 PRINT : INPUT " データ・ファイル名 ";DMS
2520 OPEN DMS FOR OUTPUT AS #2
2530
2540 WRITE #2,RN$,NU,NP,XMIN,YMIN,XMAX,YMAX
2550
2560 FOR N=1 TO NU
2570 WRITE #2,UNAME(N),UX(N),UY(N)
2580 NEXT N
2590
2600 FOR N=1 TO NP
2610 WRITE #2,X(N),Y(N),F(N)
2620 NEXT N
2630
2640 CLOSE #2
2650
2660
2670 *FINISH
2680 VIEW (0,0)-(679,399)
2690 CONSOLE ,,1
2700 CLS 3
2710 END
2720
2730
2740
2750 *SKELETON
2760 CLS 3
2770 FOR N=1 TO NP
2780 IF F(N)=0 THEN LINE (X(N-1),Y(N-1))-(X(N),Y(N)),5
2790 NEXT N
2800 RETURN
2810
2820
2830 *CORRPT
2840 N=N-1
2850 LINE (X(N-1),Y(N-1))-(X(N),Y(N)),0
2860 CIRCLE (X(N),Y(N)),10,0
2870 RETURN 1610

```

## リスト6

```

1000 *** 統計地図作成のための白線画人力 ***
1010 *** PROGRAM NAME : "JUPITER" ***
1020 DIM V(100),UNAME(100)
1030 CLS 3
1040
1050 PRINT : INPUT "データ・ボリュームのドライブ番号 ";NK
1060 PRINT : FILES NK
1070 PRINT : INPUT "白線画データ・ファイル名 ";DMS
1080 OPEN DMS FOR INPUT AS #1
1090
1100 INPUT #1,RN$,NU,NP,XMIN,TMIN,XMAX,TMAX
1110
1120 FOR N=1 TO NU
1130 INPUT #1,UNAME(N),UX,UY
1140 NEXT N
1150
1160 CLOSE #1
1170
1180
1190 CLS 3
1200 PRINT "地域名 :";NM
1210 PRINT : PRINT "単位地区数 :";NU
1220 PRINT : PRINT " 地域番号 :";NM
1230 PRINT : INPUT " 単位地区名 ";UNAME(N)
1240
1250 CLS
1260 FOR N=1 TO NU
1270 PRINT USING " ## ":N;
1280 PRINT UNAME(N)
1290 LOCATE 30,CSRLIN-1 : INPUT " 隣接 ";V(N)
1300 PRINT
1310 NEXT N
1320
1330 CLS
1340 PRINT "人力が終わりました (どれかのキーを押してください)"
1350 K$=INKEY$
1360 IF K$="" GOTO 1360
1370
1380 CLS
1390 TT=PI*(NU/5)
1400 FOR T=0 TO TT
1410 CLS
1420 FOR N=1 TO T+5
1430 IF N>NU GOTO 1480
1440 PRINT USING " ## a" & " ";N,UNAME(N);
1450 PRINT " 隣接=";V(N)
1460 PRINT
1470 NEXT N
1480 PRINT "よろしいですか? (Y/N)"
1490 K$=INKEY$
1500 IF K$="" GOTO 1480
1510 IF K$="Y" OR K$="y" THEN GOSUB *CORRDATA
1520 NEXT T
1530
1540 CLS
1550 PRINT "データを保存しますか? (Y/N)"
1560 K$=INKEY$
1570 IF K$="" GOTO 1560
1580 IF K$="Y" OR K$="y" THEN GOSUB *DATASAVE
1590
1600 CLS
1610 END
1620
1630
1640 *CORRDATA
1650 PRINT : PRINT "修正してください"
1660 PRINT : INPUT " 地域番号 ";NN
1670 IF NN=NU GOTO 1860
1680 PRINT : INPUT " 隣接 ";V(NN)
1690 RETURN 1410
1700
1710
1720 *DATASAVE
2390 CLS
1740 PRINT "統計データ・ファイルを作成します"
1750 PRINT : INPUT "データ・ボリュームのドライブ番号 ";NK
1760 PRINT : FILES NK
1770 PRINT : INPUT "統計データ・ファイル名 ";DMS
1780 OPEN DMS FOR OUTPUT AS #2
1790
1800 WRITE #2,RN$,NM$,UNITS,NU
1810
1820 FOR N=1 TO NU
1830 WRITE #2,V(N)
1840 NEXT N
1850
1860 CLOSE #2
1870 RETURN

```

リスト7

```

1000 *** 2変数の比率 (%) の計算 ***
1010 PROGRAM NAME "GOL"
1020 DIM V1(100),V2(100),V3(100)
1030
1040 CLS
1050 PRINT "2変数の比率をパーセントで計算します"
1060 PRINT : INPUT "データ・ボリュウムのドライブ番号 ";NX
1070 PRINT : FILES NX
1080 PRINT : INPUT "統計データ・ファイル名:実数1 (分子) ";DFN1#
1090 OPEN DFN1# FOR INPUT AS #1
1100 INPUT #1,DN1#,VN1#,UNIT1#,NU1
1110 FOR N#1 TO NU1
1120 INPUT #1,V1(N)
1130 NEXT N
1140 CLOSE #1
1150
1160 PRINT : INPUT "統計データ・ファイル名:実数2 (分母) ";DFN2#
1170 OPEN DFN2# FOR INPUT AS #1
1180 INPUT #1,BN2#,VN2#,UNIT2#,NU2
1190 IF NU1<NU2 THEN CLS : PRINT "単位地区数不一致!!" : GOTO *OWARI
1200 FOR N#1 TO NU2
1210 INPUT #1,V2(N)
1220 NEXT N
1230 CLOSE #1
1240
1250 FOR N#1 TO NU1
1260 IF V2(N)=0 THEN V3(N)=0 ELSE V3(N)=V1(N)/V2(N)*100
1270 NEXT N
1280
1290 PRINT : INPUT "算出データ・ファイル名 (%) ";DFN3#
1300 PRINT : INPUT "算出データ名 ";VN3#
1310 UNIT1#="%"
1320 OPEN DFN3# FOR OUTPUT AS #1
1330 WRITE #1,BN1#,VN1#,UNIT1#,NU1
1340 FOR N#1 TO NU1
1350 WRITE #1,V3(N)
1360 NEXT N
1370 CLOSE #1
1380
1390 *OWARI
1400 END
    
```

リスト8

```

1000 *** 統計地図作成プログラム (カラー) ***
1010 *** PROGRAM NAME "VENUS3" ***
1020 DIM V(100),UNAN#(100),UX(100),UY(100),X(2000),Y(2000),P(2000)
1030 DIM TH(9),TILE(8)
1040 CONSOLE ,0 : SCREEN 3 : CLS 3
1050
1060 PRINT : INPUT "データ・ボリュウムのドライブ番号 ";NX
1070 PRINT : FILES NX
1080 PRINT : INPUT "白地図データ・ファイル名 ";DN1#
109# PRINT : INPUT "統計データ・ファイル名 ";DN2#
1100
1110 OPEN DN1# FOR INPUT AS #1
1120
1130 INPUT #1,RN#,NU,NP,IMIN,YMIN,IMAX,YMAX
1140
1150 FOR N#1 TO NU
1160 INPUT #1,UNAN(N),UX(N),UY(N)
1170 NEXT N
1180
1190 FOR N#1 TO NP
1200 INPUT #1,X(N),Y(N),P(N)
1210 NEXT N
1220
1230 CLOSE #1
1240
1250
1260 OPEN DN2# FOR INPUT AS #1
1270
1280 INPUT #1,BN2#,VN#,UNIT#,NU2
1290 IF NU<NU2 THEN CLOSE : GOTO *QEND
1300
1310 FOR N#1 TO NU
1320 INPUT #1,V(N)
1330 NEXT N
1340
1350 CLOSE #1
1360
1370
1380 CLS : PRINT "地域名: ";RN#
1390 PRINT : PRINT "実数名: ";VN#
1400 PRINT : PRINT "実数の単位: ";UNITS
1410 PRINT : PRINT "単位地区数: ";NU
1420 PRINT : PRINT "どれかのキーを押してください"
1430 K#="INBEY"
1440 IF K#="" GOTO 1430
1450
1460 CLI#B
1470
1480 GOSUB *TILEPATTERN
1490 GOSUB *THRESHOLD
1500 IF DC=1 THEN GOSUB *LEGEND2 ELSE GOSUB *LEGEND
1510 GOSUB *PAINTMAP
1620
1530 PRINT : PRINT "再実行しますか?": PRINT " (Y/N) "
1540 K#="INKEY#
1550 IF K#="" GOTO 1540
1560 IF K#="Y" OR K#="y" GOTO 1490
1570
1580 PRINT : PRINT "ハードコピー?": PRINT " (Y/N) "
1590 IF K#="" GOTO 1590
1610 IF K#="N" OR K#="n" GOTO *FINISH
1620
1630 GOSUB *TILEPATTERN
1640 IF DC=1 THEN GOSUB *LEGEND2 ELSE GOSUB *LEGEND
1650 GOSUB *PAINTMAP
1660
1670
1680 *FINISH
1690 WHILE INBEY#="" : WEND
1700 VIEW (0,0)-(639,399) : CLS 3
1710 CONSOLE ,,1
1720 CLS 3
1730 END
1740
1750
1760
1770 *THRESHOLD
1780
1790 VIEW (0,0)-(639,399) : CLS 3
1800 PRINT "クラスタ間の境界線を手立てください"
1810 PRINT : PRINT " (クラスタは8です) " : PRINT
1820 FOR CL#1 TO CL#1
1830 PRINT
1840 PRINT USING "クラス #";CL : PRINT USING "/クラス #";CL+1
1850 LOCATE 20,CSR(LIN-1) : INPUT "境界線: ";TH(CL)
1860 NEXT CL
1870
1880 PRINT : PRINT "小地区以下を縮小しますか?"
1890 PRINT : INPUT " (表示する=1, 表示しない=0) ";DC
1900
1910
    
```

```

1920 RETURN
1930
1940
1950 *LEGEND
1960
1970 VIEW (0,0)-(639,399) : WINDOW (0,0)-(639,399) : CLS 3
1980 PRINT RN# : PRINT
1990 PRINT VN# : PRINT
2000 PRINT " (";UNITS;")"
2010 FOR CL#1 TO CL#1
2020 LINE (0,82+(CL-1)*16)-(STEP(16,12),7,B
2030 PAINT (6,91+(CL-1)*16),TILE#(CL),7
2040 LOCATE 4
2050 IF CL#1 THEN PRINT USING " - #####" ;TH(CL) : GOTO 2100
2060 IF CL#CL THEN PRINT USING "##### -";TH(CL-1) : GOTO 2100
2070 PRINT USING "#####";TH(CL-1);
2080 PRINT " ";
2090 PRINT USING "#####";TH(CL)
2100 NEXT CL
2110
2120 RETURN
2130
2140
2150 *LEGEND2
2160
2170 VIEW (0,0)-(639,399) : WINDOW (0,0)-(639,399) : CLS 3
2180 PRINT RN# : PRINT
2190 PRINT VN# : PRINT
2200 PRINT " (";UNITS;")"
2210 FOR CL#1 TO CL#1
2220 LINE (0,82+(CL-1)*16)-(STEP(16,12),7,B
2230 PAINT (6,91+(CL-1)*16),TILE#(CL),7
2240 LOCATE 4
2250 IF CL#1 THEN PRINT USING " - ##" ;TH(CL) : GOTO 2300
2260 IF CL#CL THEN PRINT USING "###" ;TH(CL-1) : GOTO 2300
2270 PRINT USING "###";TH(CL-1);
2280 PRINT " ";
2290 PRINT USING "###";TH(CL)
2300 NEXT CL
2310
2320 RETURN
2330
2340
2350 *TILEPATTERN
2360
2370 FOR CL#1 TO CL#1
2380 TILE#(CL)=""
2390 FOR N#1 TO 6
2400 READ A# : TP=VAL("AH"+A#)
2410 TILE#(CL)=TILE#(CL)+CHR#(TP)
2420 NEXT N
2430 NEXT CL
2440
2450 RETURN
2460
2470
2480 *QEND
2490 PRINT : PRINT "単位地区数不一致!!"
2500 CONSOLE ,,1
2510 GOTO 1730
2520
2530
2540 *PAINTMAP
2550
2560 WINDOW (XMIN,YMIN)-(XMAX,YMAX)
2570 LX=IMAX-IMIN : LY=YMAX-YMIN
2580 IF LY/LX<396/478 THEN LY=LY/LX*478 : VIEW (161,1)-(638,LT),7 : GOTO 2600
2590 LX=LX/LY*396 : VIEW (638-LX,1)-(638,399),,7
2600
2610
2620 FOR N#2 TO NP
2630 IF P(N)=0 THEN LINE (X(N)-1),Y(N)-1)-(X(N),Y(N)),7
2640 NEXT N
2650
2660 FOR N#1 TO NU
2670 FOR CL#1 TO CL#1
2680 IF V(N)<TH(CL) GOTO 2710
2690 NEXT CL
2700 CL#CL
2710 PAINT (UX(N),UY(N)),TILE#(CL),T
2720 NEXT N
2730
2740 RETURN
2750
2760
2770 DATA AA,00,00,85,00,00
2780 DATA BB,00,72,00,88
2790 DATA CC,00,AA,00,88
2800 DATA DD,AA,55,00,85,AA
2810 DATA DD,FF,AA,00,FF,55
2820 DATA DD,FF,22,00,FF,88
2830 DATA DD,FF,00,00,FF,00
2840 DATA AA,FF,00,55,FF,00
2850 DATA FF,00,77,FF,00,DD
2860 DATA FF,00,FF,FF,00,FF
2870 DATA DD,AA,FF,00,56,FF
2880 DATA DD,77,FF,00,DD,FF
2890 DATA DD,FF,FF,00,FF,DD
2900 DATA DD,FF,77,00,FF,DD
2910 DATA DD,FF,AA,00,FF,55
2920 DATA DD,FF,00,00,FF,00
    
```