Reprinted from

Tenth International Symposium

Machine Processing of

Remotely Sensed Data

with special emphasis on

Thematic Mapper Data and

Geographic Information Systems

June 12 - 14, 1984

Proceedings

Purdue University
The Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907 USA

AN APL GEOGRAPHIC INFORMATION SYSTEM

A. REMY-PFLUCKER

IBM del Peru
Lima, Peru

I. ABSTRACT

The following study has been done using equipment and software available at the present time in many computer installations of Peru, with the objective of developing a geographic information system that will permit an understanding of the technique without having to use sophisticated hardware equipment or software packages.

The study was done using a map of Peru in the scale 1:5'000,000 published by the National Planning Institute of Peru in the "Atlas Histórico Geográfico de Mapas y Paisajes Peruanos" of 1969.

Existing little information in Peru on the subject matter of the research, the study covered not only the analysis and handling of maps by the computer, but also the data entry methodology for storing them.

II. APLGIS: AN APL GEOGRAPHIC INFORMATION SYSTEM

APLGIS is an APL(1) based system which provides a flexible and simple means of coding, storing, processing and displaying of geographic information and serves as a tool for studying the basic concepts of geographic information processing.

For demostration purposes, APLGIS has access to a geographic data base containing maps describing the political (departments and provinces) and geographical (coast, sierra and jungle) division of Peru. APLGIS has access also to social and economical information of Peru, stored in table form in a relational type of data base, which can be retrieved through APL using QBE(2).

The use of APLGIS is similar to the use of common techniques for the handling of geographic maps. The processing of the information is done by combining primitive operators, using comands in Spanish, in a form similar to natural language, avoiding in this way the need to know of a programming language.

APLGIS is written in APL/VS and runs under VM/370 facility using the IBM programs GRAPHPAK and GDDM(3). It uses also the APL auxiliary processor APL110 for the management of the CMS files. A virtual space of 2 megabytes and an IBM 3279 display terminal are needed to run APLGIS.

III. MAP ENCODING

In order for the computer to handle map information, this must be first computer coded. Several means exist by which geographic data may be encoded or digitized for computer processing. The simplest method is doing it manually, other more sophisticated methods employ the use of a hardware equipment known as digitizers.

Besides the input means of encoding map information, there also exists the encoding format for coding and storing the data in the computer. The most common are: the grid cell format and the polygon format.

The cell format asigns values to a quadrille superposed over the area of interest. The atributes for each area are asigned acording with the values contained in each cell. The smaller the cell, the greater will be the detail captured from the area under study (Figure 1a), but at the same time the data file space will increase considerably. If in turn a coarse cell is used to reduce file space, it may occur that several data types will fall in the same cell.

The polygon format is asociated to

the coding of feature boundary lines. This scheme describes areas of the earth surface as geometric figures called polygons, formed by nodes joined together by straight lines (Figure 1b). The polygons are encoded by determining the coordenates (x,y) of the nodes. This format adheres more to the geographic reality to be codified than the cell technique, requiring less data file space, although the former is more easily programmed.

The selection of the coding format depends then, not only on the data storage space available, but on the type of information to be encoded, on the precision desired for the information contained in the map and on the accuracy of its display.

APLGIS uses the polygon scheme. Two conditions contributed for selecting this technique:

1. The main objective of using available graphics software imposed the use of the GRAPHPAK program which uses point coordinates for "drawing".

2. The coding of the political division of Peru (departments and provinces) as well as the contour of the country itself, demanded the use of a more exact representation of the geographic reality.

The data entry of the maps that form the APLGIS geographic data base, consisting of 186 maps of Peru (156 provinces, 26 departments, 3 geographic regions, and the contour of the country) was done by hand, superposing a quadrille of 2mm by side over a map at scale 1:5'000,000. The map matrices have n rows and three columns, the first column is a color code and the other two are the (x,y) coordinates.

IV. MAP OPERATIONS

A. DATA BASE MANAGEMENT

ABRIR∆FILE
CERRAR∆FILE
LEER∆MAPA
LEE∆PROVINCIAS
LEE∆DEPARTAMENTOS
GUARDA∆MAPA

1. *ABRIR∆FILE* filename

Establishes comunication between APLGIS and CMS using auxiliary processor AP110. filename is the name of the CMS file containing the geographic information data base.

2. *CERRAR∆FILE* filename

Closes the comunication between APLGIS and CMS.

3. mapa←*LEER∆MAPA* 'cod;nombre'

Reads map of class cod and name nombre from the file opened by *ABRIR∆FILE* and assigns the values to mapa.

4. *LEE∆DEPARTAMENTOS* ['deptname'|'deptname' *Y* 'deptname' *Y* 'deptname'...|*DEPTPERU*]

Reads the departments of name deptname or all the departments contained in the list *DEPTPERU*, from the file opened by *ABRIR∆FILE*.

5. *LEE∆PROVINCIAS* (Same as above)

Reads the provinces of the departments of name deptname, or all of the provinces of the department names contained in *DEPTPERU*, from the file opened by *ABRIR∆FILE*.

6. *GUARDA∆MAPA*

Saves in the file opened by *ABRIR∆FILE* the maps desired. The function is conversational and will ask the user information regarding the map to be saved.

The map is saved according to the following register format:

$[n_1|n_2|\text{Fac.}|\text{Rows}|x_1|y_1|x_2|y_2|...]$

where:

n_1 = Number of the department. If the map is the contour of Peru this number is zero.

n_2 = Number of the department within the country, or number of the province within the country, within the department n_1. If n_1 is zero n_2 is also zero.

Fac. = Convertion factor from coded area to real area.

Rows = Number of points (x,y) coded for the map.

x_1y_1 = Coordenates of points (x,y) coded for the map.

For any other map not being a province or department, n_1 and n_2 are zero.

B. DATA ENTRY

Since the data was coded by hand, the

following functions were programmed mainly to ease this operation. They are:

LEER∆DATOS
EXTRAE∆DATOS
AGREGA∆∆DATOS

1. mapa←*LEER∆DATOS*

Friendly function to create matrix mapa containing the (x,y) coordenates of the map.

2. mapa←*EXTRAE∆DATOS* mapa1

Friendly function which will extract data from mapa1 to create matrix mapa.

3. mapa←mapa1 *AGREGA∆∆DATOS* mapa2

This function catenates mapa2 to mapa1 creating a matrix of name mapa.

C. MAP HANDLING

In the development of the techniques for map handling with the computer the operations commonly done by hand were taken as examples.

At first we had only the political division of Peru in the computer, but since it didn't offer any challendge for operations such as intersection of maps, we then added the geographical division. This geographical division is a very broad one, but since the map coding was done by hand, divisions such as forrest types, agriculture crops, etc. which would have been more interesting to manage, would have taken longer time to code.

Of the functions to be presented, two demanded a more profound research: *UNE* (add) and *INTERSECTA* (intersect). These two functions are explained in greater detail in the Appendix.

The following are the operations which were studied:

- Statistics
- Addition
- Intersection
- Subtraction
- Surface Calculation

Statistics

CLASIFICA PROVPERU SEGUN criterio

This conversational function classifies the provinces of Peru contained in the list *PROVPERU* according to the criteria explained in "criterio".

SEGUN does the statistical classification and definition of the class limits. The maximun number of classes is eight, for the eight tones of gray available in GRAPHPAK.

CLASIFICA creates up to eight maps whose names are saved in a variable which is latter given to the function *MAPA∆TEMATICO* as argument for display (Figure 2).

Addition

UNE mapa1 *Y* mapa2

The function *UNE* adds mapa1 and mapa2 creating a new map. The name of this new map is given during the execution of the function. The criteria for adding these maps is their neighborhood. If this criteria is not met (no common points), *UNE* waits for the user's decision if the addition should be carried on. In this case a map will be created formed by two zones not united geographically, but that could be considered neighbors by some other condition known to the user.

Intersection

INTERSECTA mapa1 *Y* mapa2

This function is used to create a third map that is formed by the zone common to mapa1 and mapa2 (Figure 3).

When *INTERSECTA* is used the user should be aware that APLGIS asumes that mapa1 is the "intersecting map" and mapa2 is the "intersected map". If this way of intersecting the maps gives, from the user point of view, no satisfactory results, the operation may be repeated switching the maps.

INTERSECTA analyzes the information of both maps and generates a "switch" that tells the user what zone of the "intersecting map" is going to be traced to create the "resulting map". "Switch" has two positions: ON indicating that the "intersecting map" is going to be traced "looking" West, and OFF if it is traced "looking" East (The directions East and West are from the user's point of view, such concepts don't exist for APLGIS).

The final result could be one or several maps, or even null. It is the responsability of the user to select the result which best resolves the intersection or, in the latter case, reformulate the process.

Subtraction

RESTA mapa1 *A* mapa2

Is the opposite of addition. In this operation mapa1 is subtracted from mapa2.

The process is somewhat similar to *INTERSECTA*. The result is mapa2 without the zone corresponding to mapa1 (Figure 4).

Surface Calculation

AREA∆GRAFICO mapname

This function calculates the area of the map whose name is mapname. The area is calculated by integration and then it is multilied by a factor defined according to the size of the quadrille used in the coding of the map.

D. MAP DISPLAY

For displaying the maps, APLGIS uses the 3279 color display terminal or the IBM 3287 printer terminal.

The comands for displaying the maps are the following:

Displaying Peru based contour maps. The following comands draw (*DIBUJA*) or paint (*PINTA*) maps using the contour of the map of Peru as a basis. That is, if a depart-tment of Peru is displayed it will be shown within the map of Peru (Figure 5).

The sintax of the comands are:

[*DIBUJA* | *PINTA*] [Oper1] (Option A) [Oper2] (Option B)

where:

DIBUJA	draws the maps using the function *SKETCH* of GRAPHPAK.
PINTA	paints the maps using the function *FILL* of GRAPHPAK, interpreting the first column of the map matrix as the color code.
Oper1/ Oper2	are sintax functions only, therefore they are optional. These may be: *EL*, *LA*, *LOS*, *LAS*, *DEL*, *DE*
Option A	is a function that indicates, in general terms, what is to be drawn or painted.
Option B	indicates specifically what is to be drawn or painted, taking into consideration Option A.

These options may be:

Option A	Option B
MAPA	*PERU*
REGION∆NATURAL	['region' | 'region1' *Y* 'region2' ...]
REGIONES∆NATURALES	
DEPARTAMENTO	'deptname'
DEPARTAMENTOS	['deptname1' *Y* 'deptname2' *Y* ... | *PERU*]

For example:

DIBUJA EL MAPA DEL PERU (Figure 6)
PINTA LA REGION∆NATURAL DE COSTA
DIBUJA LOS DEPARTAMENTOS
DE 'LIMA' Y 'LA LIBERTAD'
DIBUJA LAS REGIONES∆NATURALES (Figure 7)

Displaying of maps. The following comand is used to paint (*FILL*) or draw (*SKETCH*) any type of map kept in the data base. The function is interactive and will ask the user for information regarding the different options needed for displaying the map (Figure 8). The comand is:

MUESTRA (Same as above)

MUESTRA locates the map to be displayed within the graphic area defined by the user when the APLGIS sesion is defined. To show the map *MUESTRA* will calculate the displacement needed to fit the map with respect to the upper-right-hand corner of the graphic area. The user can, if he so wishes, modify this displacement.

As before, Oper1/Oper2 are sintax functions, which in this case can be: *EL* or *LA*. The options A and B may be:

Option A	Option A
DEPARTAMENTO	'deptname'
PROVINCIA	'provname'
REGION	'mapname'

where, deptname and provname are names of departments or provinces of Peru; mapname is any other map generated by means of data entry or by a map handling operation.

Map Overlay.

RECUBRE CON (Same as above)

This function is similar in its use to *DIBUJA*, and is used to cover a previously displayed map with another (Figure 9).

Density Mapping.

MAPA∆TEMATICO lista

This function is used to paint, with gray tones, density statistics of the

Peruvian provinces which have been previously classified using the statistical scheme *CLASIFICA*. The names of the maps generated by *CLASIFICA* are given to *MAPAΔTEMATICO* in the variable lista (Figure 2 & 9).

Locating by Geographic Coordenates.

UBICA 'ciudad'

This function shows in the map the location of the city ciudad using it's geographic coordenates (Figure 5).

V. CONCLUSION

The development of APLGIS has helped to get an understanding of computer map handling methods and, as was mentioned at the begining of this study, this was in essence the objective of the work being undertaken.

We have tried to include in APLGIS all the major operations described in the theory of geographic information systems. Yet, much is left to be done. One of the impediments for a more rapid advance in the study was the use of hand encoding techniques. Another was the lack of local expertise. Nevertheless, the latter forced a more thorough study on the subject which in the case of the polygon intersect routine, proved to be worthwhile.

We recognize that APLGIS is a very simple system, yet in its simple way it has served to develop local interest in this important area of computer usage. As an example, with some minor modifications, APLGIS is being used by a local university in the study of urban planning methodologys.

REFERENCES

1. "A Programming Language (APL)", IBM 5748-API, GH20-9064.

2. "Query-By-Example (QBE)", IBM 5796-PKT, G320-6062.

3. "Graphical Data Display Manager", IBM 5794-XXH, GC33-0100.

BIBLIOGRAPHY

1. Foley, J. D. and A. Van Dam, 1982, "Fundamentals of Interactive Computer Graphics", The Systems Programming Series, Addison-Wesley Pub. Co. Inc.

2. Hadley, G., 1961, "Linear Algebra", Addison-Wesley Pub. Co.

3. Lillesand, T.M. and R.W. Kiefer, 1979, "Remote Sensing and Image Interpretation", John Wiley & Sons, Inc.

4. Nagy, G. and S. Wagle, 1978, "Geographic Data Processing", Computing Surveys, Vol.11, No. 2, June 1978.

5. Polivka, R.P. and S. Pakin, 1975, "APL: The Language and its Usage", Prentice-Hall, Inc.

APPENDIX

A. ADDITION OF TWO MAPS (The *UNE* Function)

The process of adding two maps is done in APLGIS in the following way:

In the first part of the process, APLGIS looks if the maps to add are neighbors, that is to say, it tries to locate common nodes. For example, in Figure 10, map M will have nodes ABCDEFG coded, and map N the nodes EFGHIJK. The nodes EFG are common and therefore the maps are neighbors.

Once defined that they are neighbors, the nodes in common (EFG in the example), are removed from both maps. Of the extreme nodes (G and E in the example) only one is removed from each map. The new map is now registered as the catenation of the two said maps. In the example, the new map will be *MΔN* and its nodes will be ABCDEKJIHG. Another thing that is also considered in the process is the direction followed in the encoding of each map (clockwise/counterclockwise), since the resulting map must have only one direction. In the case of the example, the encoding direction of one of the maps will have to be changed before catenation.

B. INTERSECTION OF TWO MAPS (The *INTERSECTA* Function)

The intersection process is somewhat more complex than the addition, due mainly to the encoding scheme used (Polygonal) in which the maps appear registered as a list of coordenates (x,y). The node encoding for each map has been done taking arbitrarily whatever point seemed important considering: (i) the quadrille superposed over the map, and (ii) the presicion desired. The final result of the digitizing process is thus a matrix of n rows and two columns corresponding to the coordenates of the n nodes read. A third column is latter added to this matrix with the color code of the map. The number of rows n depends mainly on the precision of the encoding process.

This matrix can be "drawed" or "painted" using APLGIS and recreate the map. What we will be viewing on the display terminal is a polygon of n nodes joined together by a set of lines. If we now draw another map, coded using the same procedure, on top of the one drawed before, we will see them "intersecting". The problem arises when we want to obtain automatically the node matrix of the zone common to these two maps.

The solution method, used by APLGIS, to obtain this matrix is as follows.

Let us define first these maps as: "intersected map", the map with the largest number of nodes, "intersecting map" the map with less nodes, and "resulting map", the map of the zone common to both other maps.

The first step of the method is to reduce the intersected map to a set of nodes whose coordenates (x,y) are below the maximun coordenates (x,y) of the intersecting map. For example, in Figure 11, M2 is the intersected map and M1 is the intersecting map, M3 is the resulting map whose node matrix we will try to determine. The maximun coordenates of M1 are x_1 and y_1; then, the points of M2 that lay between A and B are the ones to be studied.

The nodes of M2 between A and B redefine the intersected map to a smaller set of nodes to consider. The intersecting map is not modified.

The next step is to reduce the section AB of M2 to a set A'B' that lay within M1. The algorithm developed for APLGIS is the following:

1. Take a node of the set AB and draw through it two coordenate axes.

2. Calculate for each node of M1 its new coordenates x',y' refered to the new axes.

3. Calculate the distance R from the center of the new axes to each node of M1.

4. Calculate Cos α = R/x'.

5. Find α for all the nodes of M1.

6. Add the α angles. If the value is 360 degrees, the node of the AB set is inside M1. If the value is less, the node lays outside M1.

7. If all nodes of AB have been considered, stop. If not return to 1.

Once obtained the new A'B' set of nodes the next step is to add it to a subset of M1 to create the resulting map. In the example of Figure 11, M1 is formed by M3+ M4, any of these two maps may be the resulting map. Which is the result we are looking for? APLGIS proceeds in the following way:

1. Take the extreme nodes of A'B'.

2. Calculate the distance of these to all the nodes of M1.

3. Take from M1 the nearest node to A', and the nearest to B'.

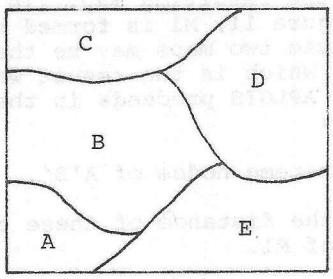
4. Ask the user which "side" of the intersection he wishes to add to A'B'. ("SWITCH").

5. According to 4., take from M1 starting from A', according to the node found in 3., those that depart from A' with increasing value of their coordenates x, or depart from A' with decreasing value of their coordenates x. Do this until the node defined in 3., as the one near B', is found.

6. Add this set of nodes to A'B'.

AUTHOR BIOGRAPHICAL DATA

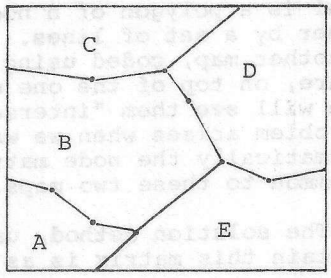
Alfredo Remy-Pflucker. Holds a B.S. in industrial engineering from the National University of Engineering of Peru and a M.S.I.E. from the University of Alabama. In 1980 he received the Professional Engineering Degree in Industrial Engineering awarded by the National University of Engineering of Peru. He is also a Visiting Professor to the National Agrarian University of Peru. Currently he is the manager of the Scientific Project Group of IBM Peru. Mr. Remy is a member of The Institute of Management Sciences, the Operations Research Society of America, the American Society of Photogrammetry and Alpha Pi Mu society.



Original Map

c	c	c	c	c	c	c	d	d	d	d
c	c	c	c	c	c	d	d	d	d	d
b	b	c	c	b	b	d	d	d	d	d
b	b	b	b	b	b	d	d	d	d	d
b	b	b	b	b	b	b	d	d	d	d
b	b	b	b	b	b	b	e	d	d	d
a	a	b	b	b	b	b	e	e	e	e
a	a	a	a	b	e	e	e	e	e	e
a	a	a	a	e	e	e	e	e	e	e

(a) Cell Format



(b) Polygon Format

Figure 1. Map Encoding Formats

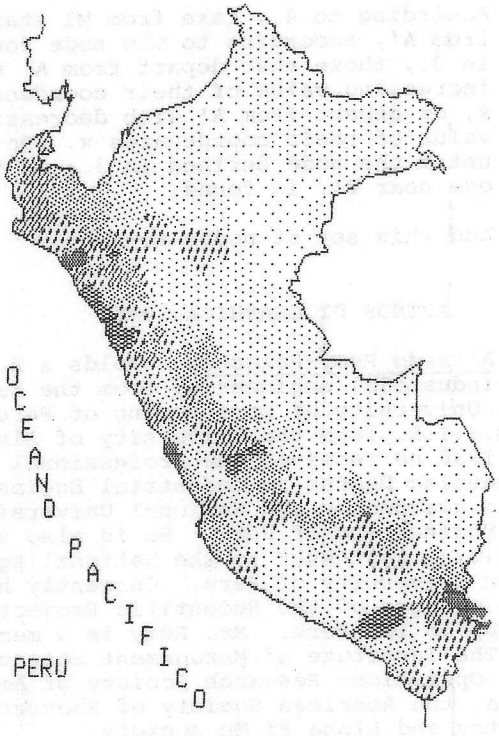


Figure 2. *CLASIFICA PROVPERU SEGUN (HABITANTES VS SUPERFICIE)*
MAPA∆TEMATICO REGIONES

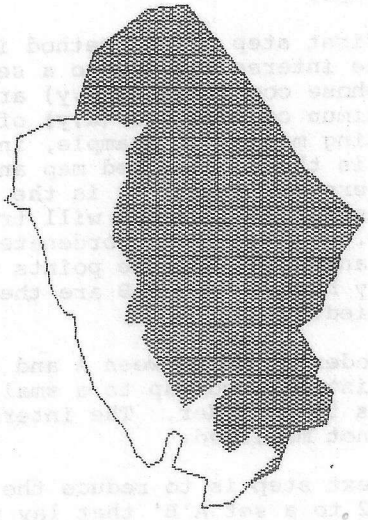


Figure 3. *INTERSECTA 'DEPT∆ ANCASH' Y 'MAPA∆SIERRA'*
MUESTRA LA REGION 'SIERRA∆44∆ ANCASH'
MUESTRA EL DEPARTAMENTO DE 'ANCASH'

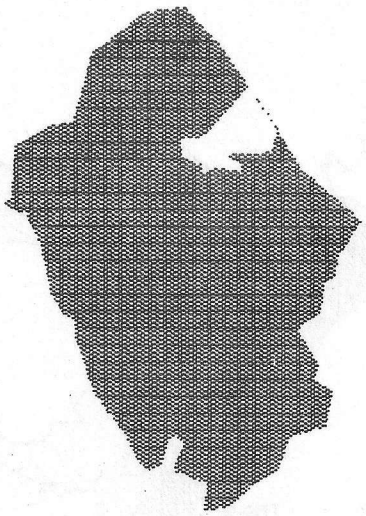


Figure 4. *RESTA 'PROV∆P6' A 'DEPT∆ ANCASH'*
MUESTRA LA REGION 'P6∆18∆ANCASH'

Figure 5. *DIBUJA LOS DEPARTAMENTOS DE 'SAN MARTIN' Y 'UCAYALI' Y 'APURIMAC'*
UBICA 'MENDOZA'
UBICA 'ABANCAY'
UBICA 'PUCALLPA'

Figure 6. *DIBUJA EL MAPA DEL PERU*

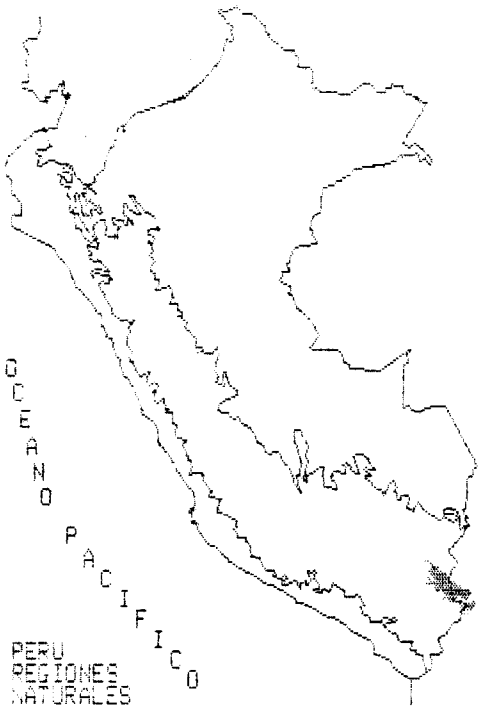


Figure 7. *DIBUJA LAS REGIONES NATURALES*

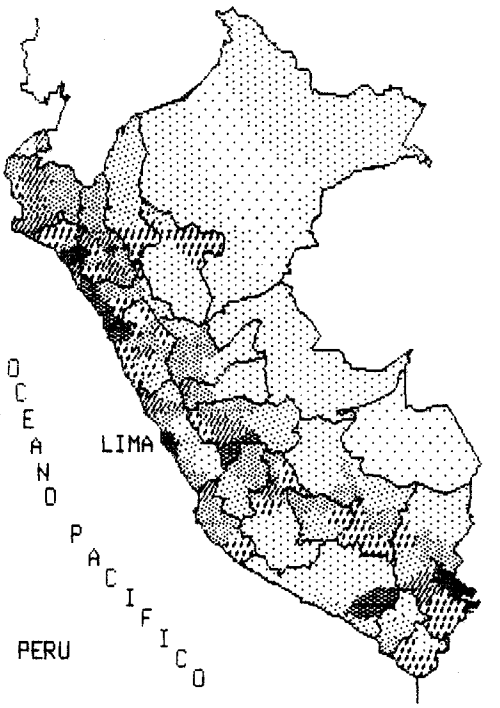


Figure 9. *CLASIFICA PROVPERU SEGUN (HABITANTES VS SUPERFICIE)*
MAPA∆TEMATICO REGIONES
RECUBRE CON LOS DEPARTAMENTOS DEL PERU

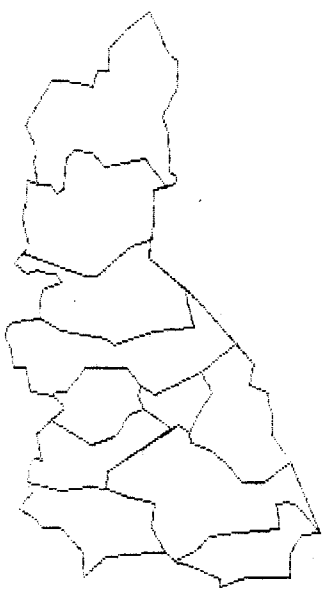


Figure 8. *MUESTRA EL DEPARTAMENTO DE 'CAJAMARCA'*

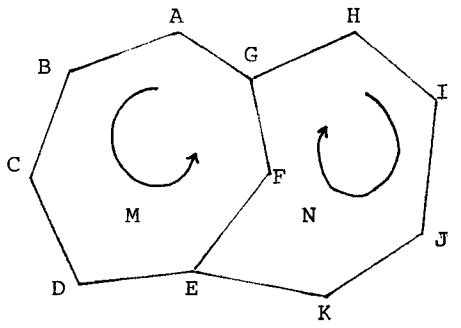


Figure 10. Addition of two Maps. The *UNE* Function.

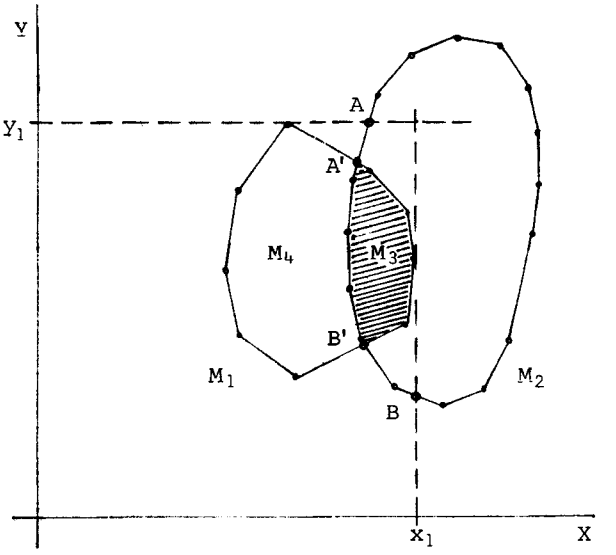


Figure 11. Intersection of two Maps The *INTERSECTA* Function.