Title
CARTE: A THEMATIC MAPPING PROGRAM

Authors
Wood, P.M.
Austin, D.M.

Publication Date
1974-07-01

Presented at the Conference on Computer
Graphics and Interactive Techniques,
Boulder, CO, July 15 - 17, 1974

LBL-3073 c.2

CARTE: A THEMATIC MAPPING PROGRAM

P. M. Wood and D. M. Austin

July 1974

Prepared for the U. S. Energy Research and
Development Administration under Contract W-7405-ENG-48

CARTE: A Thematic Mapping Program

by

P. M. Wood and D. M. Austin
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

ABSTRACT

CARTE is the graphics display program of the LBL Computer Mapping System, producing thematic maps on microfilm at one hundredth the cost of producing negatives by hand. The program matches a geographical area or a symbol representing such an area with statistical data to produce graphic output on 35mm film in the form of cross-hatched maps for single-color printing (either computer-generated dot screens or total mask frames for photographic screening).

A versatile set of directives allows the user to design the map and a corresponding report, and to specify such features as automatic placement of area names, calculation of a smooth distribution for color coding, and boundary clipping to specified limits for sectioning a map.

TABLE OF CONTENTS

CARTE: A Thematic Mapping Program

by

P.M. Wood and D.M. Austin
Lawrence Berkeley Laboratory, University of California
Berkeley, California 94720

I. Introduction

The graphic display of statistical data by geographic area is known as thematic (or choropleth) mapping. Thematic maps have traditionally been produced using a combination of manual and photographic techniques. One technique requires cutting a sheet of textured material to fit a geographic boundary and pasting the material onto a paper outline of the map. A more recent manual technique involves scribing of the map outline of a photographically opaque material. A separate sheet is required for each data range (color tone or textured pattern). Each area in a data range is peeled from the material to form a photographic mask which is then reduced to print size.

Computer techniques for producing maps cheaply and quickly were pioneered by Harvard University's Laboratory for Computer Graphics and Spatial Analysis. The best known is the printer plot technique, which provides low-resolution maps with a minimum of special hardware [1].

With the advent of high speed, high precision devices for computer output on microfilm (COM devices), the possibility of producing high quality film images by computer became an attractive alternative to both the manual techniques and the printer plot compositions. Producing COM film images, as compared with manually prepared film images, has proven to be cheaper by almost two orders of magnitude, making thematic mapping a resource available to a much wider range of applications than ever before.

The primary problem with computerized mapping is in building a machine-readable geographic data base. Statistical data, such as the various census counts, have long been available in machine-readable form, but a data base of geographic areas of interest, such as state, county and census tract boundaries were not generally available and typically involved a laborious hand-digitizing and editing process.

In order to develop automated cartography into a useful tool for data display, the Mathematics and Computing Group of the Lawrence Berkeley Laboratory initiated a computer mapping project in collaboration with the Manpower Administration of the Department of Labor, the Geography Division of the Bureau of the Census and the Department of Housing and Urban Development. The purpose of the project is to create an accurate geographic data base containing the boundaries and identification codes for the United States by state and county and for all the Standard Metropolitan Statistical Areas (SMSAs) by census tract (some 35,000 tracts). The automated map digitizing and editing system is described in a separate report [2].

This data base is to be used in the production of statistical atlases for use by Federal agencies in areas such as manpower resource planning and evaluating population characteristics. This report describes the display portion of LBL's Computing Mapping System, a FORTRAN program for thematic mapping called CARTE.

Given a geographic data base and a set of data keyed to geographic area, CARTE produces film images of maps and tables which are photographically enlarged to produce printer plate negatives.

There are five major functional parts to the program, as shown in Figure 1. This report describes each part in succeeding chapters.

A. Components of a Thematic Map

The objective of a thematic map is to communicate geographical relationships. In order to accomplish this goal, there are several criteria which must be met, as the following list indicates.

a. Recognizability - The geographic area being mapped should be easily recognizable, usually by providing appropriate labels and by retaining as many geographic features (such as rivers, lakes and coastlines) as possible. Also, the scale should be chosen so that all the data is readable.

b. Titles - The map should contain enough titling information to completely specify the meaning of the map.

c. Clear Data Representation - The method for distinguishing data ranges on the map must provide a number of distinct patterns, either by color tones, textures or symbols. Legends should indicate unambiguously the correspondence between representation and data range.

Any program for thematic mapping must provide facilities for designing these features into a map. In addition, general design features which allow arbitrary placement of the map components is necessary because of the wide variation in shapes of geographic areas. CARTE provides a set of directives which allows the user to design both the map and a table listing the actual data that is being represented.

There are three forms of map graphics available in CARTE: cross-hatching, dot screening and total masking.

Cross-hatching is used for single-color maps or slides, and is therefore the least expensive to print. Variation of direction and spacing of the cross-hatching provides eight distinct textures on the COM device in use at LBL.

When multi-color maps are desired, usually three tones of each color are distinguishable by the human eye - 10%, 50% and 100% screens are usually used. CARTE provides a method of generating these dot screens on the COM device, so that all three tones for a single color can be drawn on a single frame. This is done by using three different dot sizes spaced on a regular array of raster points. The main advantage to dot screening is the savings in cost by combining three tones on one frame of film. Thus a 9-tone, three color map, plus black outline requires processing and registration of only four frames of film. The disadvantage in this technique is the loss of resolution caused by the dot sizes required to generate distinct tones. For example, on a device with 4096 adressable points, dot sizes of 2, 4 and 8 raster points in a hexagonal array reduce the resolution to 4096/8 = 512.

The total masking technique requires the complete masking of each area to be a certain tone of any color. That is, for a 9-tone, three color map, plus black outline, 10 frames of film are generated. The tone screens are produced photographically after enlargement of the 35mm film to print size, thus achieving a much finer screen. The main advantage to this technique is the use of 2-raster point vector to mask an area.

This results in a resolution of 4096/2 = 2048 raster points, a factor of 4 better than the dot screening. The disadvantage is, of course, the processing and registration of more than twice as many frames of film - not too serious a problem with pin-registered cameras and a good photographic shop.

B. Program Structure and Information Flow

The functional structure and control flow for CARTE are diagrammed in Figure 1. There are five functional modules: a control module where directives are processed, a setup module which generates the graphic data structure from the input map file, a report generation module, a distribution module which assigns representation codes (color or texture) from the data, and a graphic output module which generates the film images, including titles, legends, outlines and masks.

The program runs most efficiently when many maps of the same area are made in one job step. The initial processing of the map file done in the setup module creates a very efficient representation of the geographic area which is used in generating a series of maps from corresponding data sets - i.e., the data, title, legends, etc., change with each map, but the geographic area stays the same.

After the setup function, the map is ready to be combined with nominal, ordinal or interval data to produce a thematic map. The flow of control passes from the directive interpreter to the report module (if specified), where a film image of the input data is produced in tabular form. The distribution module then assigns a graphic code (color or texture) to each of the data items keyed to geographic areas. Control then passes to the graphic output module which creates the film images for the map.

The control loop from directives to table to graphic output is repeated for each data set provided, with the incremental cost of computer

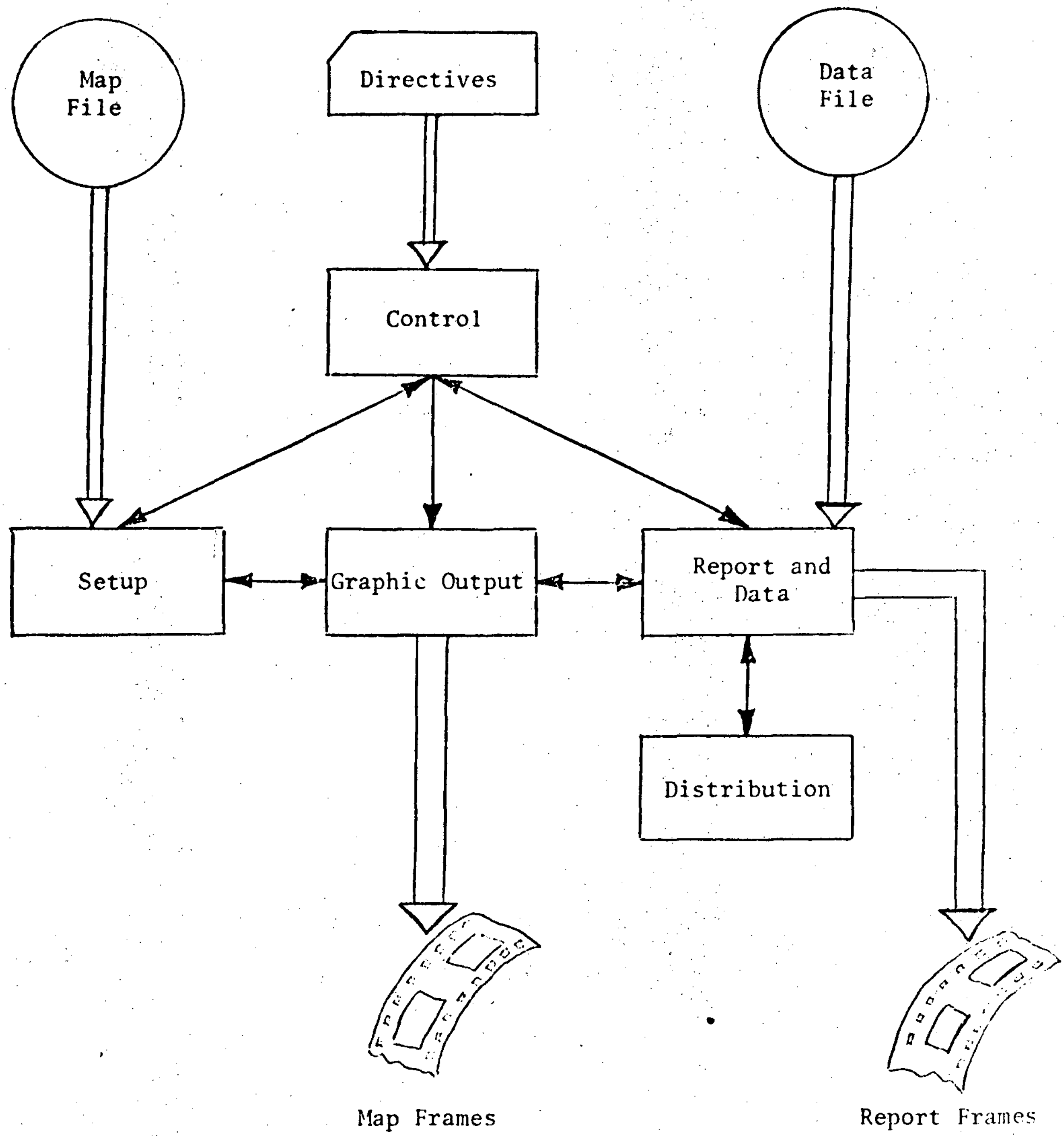


Figure 1. DIAGRAM OF PROGRAM STRUCTURE AND FLOW OF CONTROL

time per map decreasing with the number of data sets. In a typical example, the cost for producing maps of the Phoenix, Arizona SMSA (233 census tracts) was as follows:

setup	$5	
10 maps @ $0.50	$5	
	$10	or $1 per map

Comparable costs for producing these film images by hand has been estimated at over $100 per map.

The major expense in producing thematic maps, especially multi-color maps, is in the printing process, but the fact remains that preparation of film images for the printer has become economically feasible for large scale projects.

II. The Control Module

The control module interprets a set of directives which set the variables controlling program operation and allow the user to design the map layout in detail. The directives consist of a control character (an asterisk in column 1), and a keyword followed by parameters in free format. Keywords may be abbreviated, since only the first character is used.

Each directive has a number of parameters associated with it which, if left unspecified, take on default values. The parameters are numbers (either integer or floating point is acceptable) separated by commas. Following the parameters, user comments can be entered, as the scan is terminated on non-numerical data.

An example of a directive is:

*KEYS, 4= 8 KEY 4 on the data set matches KEY 8 on map.

Directives specifying titles and legends require a set of cards describing textual information to be placed on the map. Thus a packet of titles may be specified as follows:

```
*TITLES,2 There will be 2 title packets
1,1,4,    (Title 1, on 1 card, size 4)
TITLE ONE FOR MAP ONE
2,1,2,    (Title 2, 1 card, size 2)
LAWRENCE BERKELEY LABORATORY
```

The control character serves as an interrupt which causes the control module to terminate the processing of the current directive and examine the next directive. Thus, to title the second map, the directive would appear as follows:

```
*TITLES,2        Still two titles
1,1,4            (Title 1, 1 card, size 4)
TITLE ONE FOR MAP TWO
*GO
```

In this case, the second title remains the same as originally specified, and the new directive (*GO) causes the reading of titles to be terminated.

The need for multiple cards to specify long titles requires that the user specify the number of cards to be read as a parameter. This overrides the interrupt features for that number of cards, so that text may begin with an asterisk. Also, the comment feature does not apply to actual text cards - only the parameter cards. There are three directives (*MAPTYPE, *INTERVALS, *WATCH) which can have character parameters and to which the comment feature also does not apply.

The control language allows full flexibility in designing the map layout with a minimum of information. Default values are provided for all parameters except textual, and parameters need only be specified when the values change.

The directives fall into four classes:

a. Setup directives specifying information about the geographic area from the map file.

b. Graphic directives specifying information about titles, legends and other "cosmetics".

c. Report directives specifying information about the format of the table.

d. Distribution directives specifying how the data is to be treated

The setup directives are as follows:

*MAPTYPE
*ZOOM
*PICTURE SPACE UNITS
*XYMAP PICTURE SPACE
*CONSTANTS

The graphics directives are:

*ARROW
*TITLE
*SCALE
*OUTLINE
*LEGEND
*BOXES
*GO

The report directives are:

*FORMAT
*KEYS
*HEADINGS
*NOTES
*REPORT

The distribution directives are:

*DATA
*INTERVALS
*EXTRACT

III. The Setup Module

The setup phase of the program processes the map file according to the setup directives to produce a data structure which allows for efficient output of the map graphics. The functions carried out on this "first pass" on the mapfile include:

a. Clipping the map to prescribed boundaries

b. Establishing map limits and storage requirements for the mapfile and shading arrays

c. Addition of label boxes (rectangular areas for label placement which will not be shaded)

d. Generating a graphic data structure representation of the map in random access mass storage (RAMS)

e. Generating a test frame for COM setup purposes.

Some functions done in the setup module depend upon user-specified directives - e.g., clipping boundaries - while other functions are univeral - e.g., memory requirements. Figure 2 diagrams these functions of the setup module. The major algorithms used in this module are discussed below. The method for storing the graphic data structure and calculating the memory requirements are discussed in Chapter VI on the data structure.

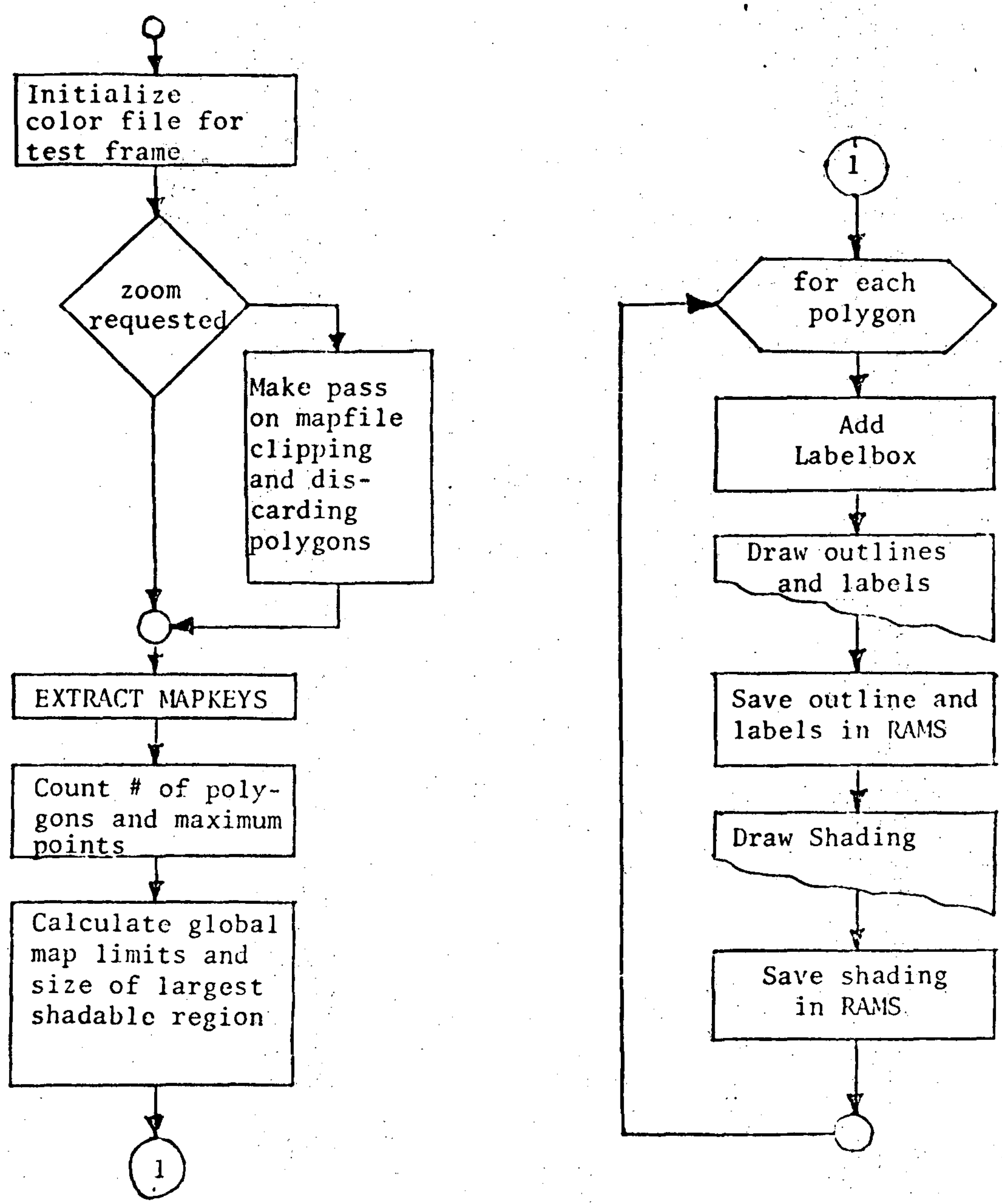


Figure 2. FUNCTIONS OF THE SETUP MODULE

A. Clipping Limits

In the absence of a *ZOOM directive, the map coordinate limits are taken from the map file - that is, the entire map is drawn and no clipping is necessary. However, it is often the case that only a portion of a map is wanted, either for clarity in viewing or for selection of subareas by coordinate limits. In this case, a directive of the form

*ZOOM,<long.min>,<long.max>,<lat.min>,<lat max>

specifies a rectangular area (in map coordinates) which is to be drawn (see Figure 3). There are three cases to consider:

a. A polygon lies entirely inside the limits

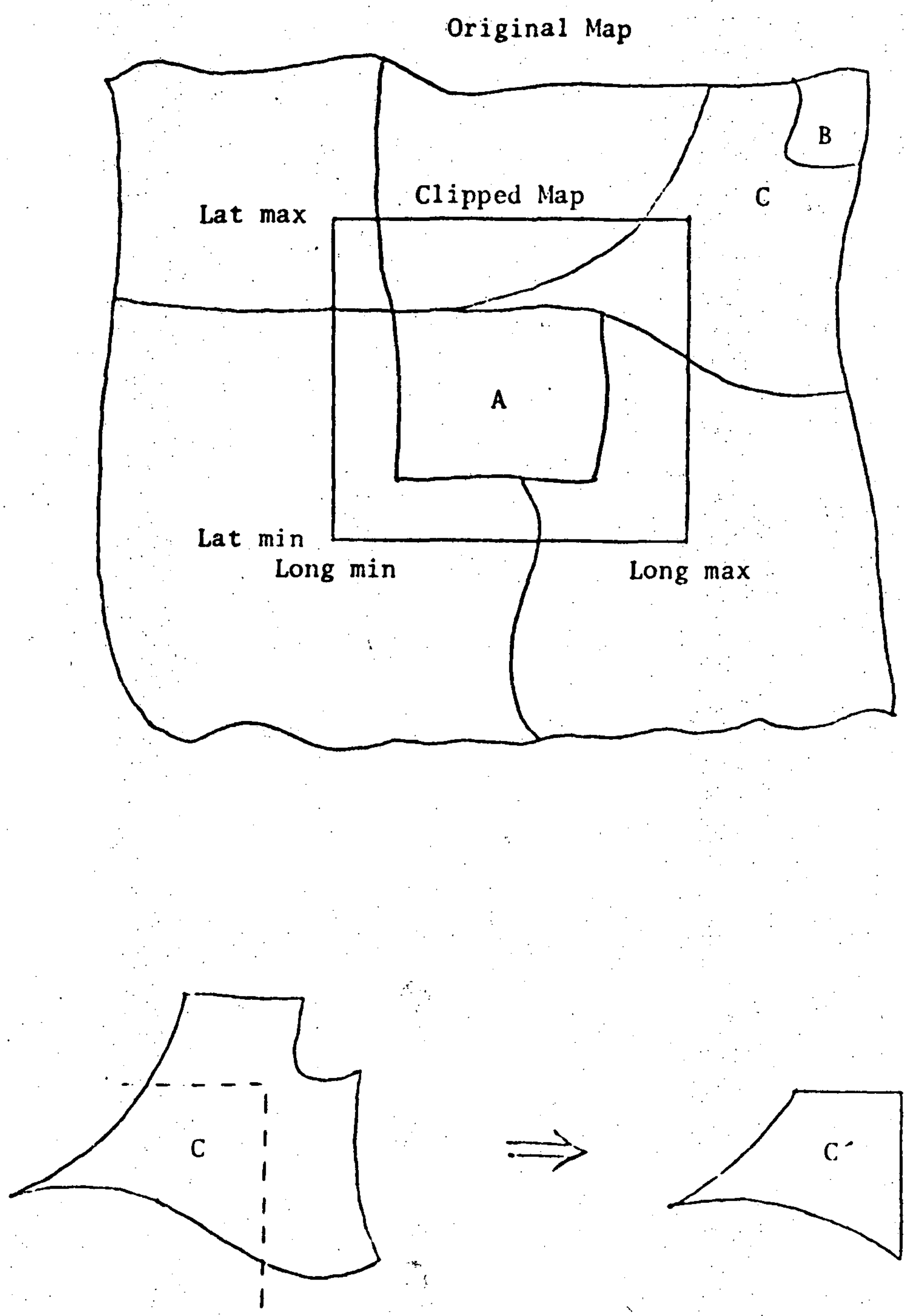
b. A polygon lies entirely outside the limits

c. A polygon intersects the limit rectangle.

The polygons are passed to a subroutine CLIPOLY for processing. For case a, CLIPOLY returns the original points. For case b, CLIPOLY returns no points. For case c, the subroutine discovers which vertices of the limit rectangle must be added to the polygon (eliminating external vertices) in order to have a closed polygon entirely within the limits (polygon c′ in Figure 3).

The CLIPOLY routine first determines the direction in which the polygon is drawn, so that the limit vertices can be inserted in the proper order. Then each side is sent thru the clipping routine TVCLIP, which flags lines intersecting the rectangular boundary and calculates the point of intersection (the cut point). If a side crosses the boundary,

Figure 3. CLIPPING TO A RECTANGULAR AREA



a flag is set and the cut point stored. Subsequent vertices are deleted until a side reenters the limit rectangle. A test is made on the exit and entry points to decide whether zero, one, two or three limit vertices must be inserted to preserve the closed polygonal structure.

Once all the polygons of a map file have been clipped to the limit rectangle, the polygons can be shaded with no further clipping. This represents a substantial saving of computer time if more than one map is to be made in a single run.

Another option is called INSET clipping, in which any polygon which intersects with the clipping rectangle is rejected. The routine will return either every polygon entirely inside the clipping rectangle or every polygon partially or entirely outside. This produces either a base map with irregularly shaped holes or the insets which exactly fit the holes in shape. More than one clipping rectangle can be specified for multiple insets.

B. Label Placement

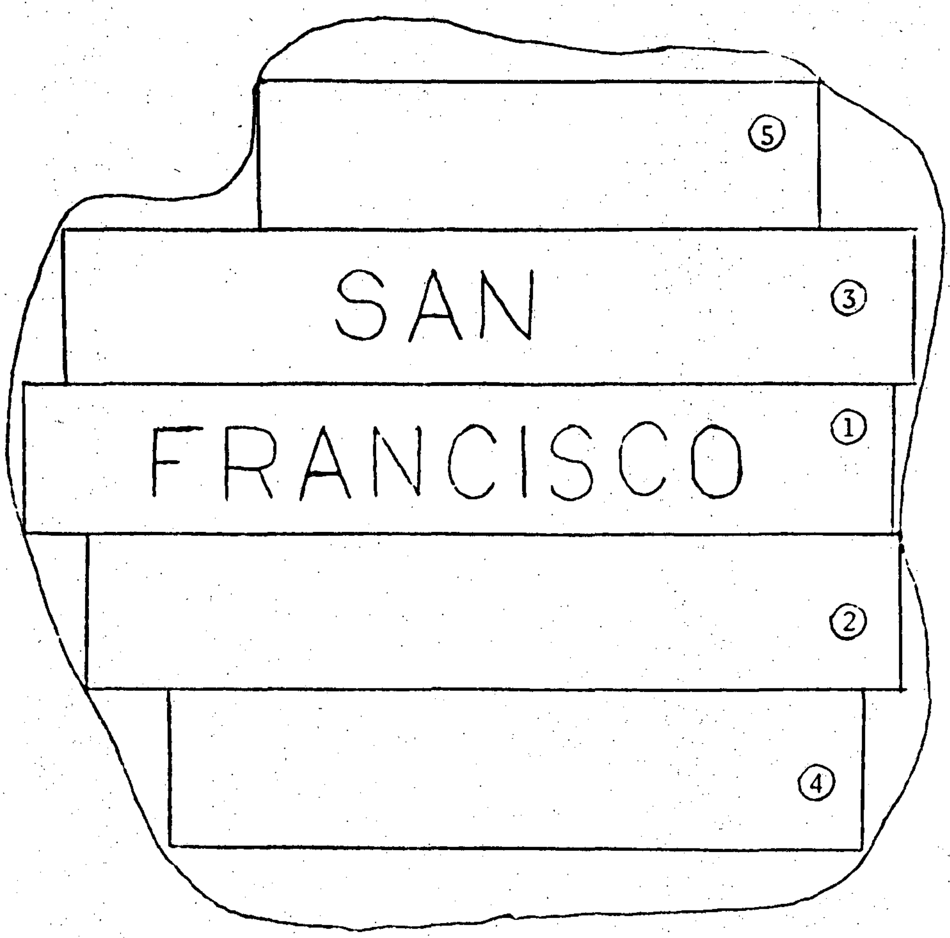
It is sometimes useful to display the name of a geographic area in a central part of that area, especially for single-color cross-hatch maps. Placing names manually is not only tedious, but must be redone when scale changes are made. Therefore, a label placing algorithm was developed which attempts to place a label of a user specified size within the polygonal boundaries of an area. The algorithm works as follows:

1. Calculate the limits of a rectangle into which the label of a specified character size will fit.
2. Starting at the center of the polygon and working outward symmetrically, search for a place where the rectangle will fit entirely within the polygon. If a place is found go to step 5.
3. Repeat step 2, only try for a verticle rectangle.
4. Check the label for an embedded blank to see if the label can be broken into more than one word. If so, repeat steps 2 and 3 for each word, retaining the proper order.
5. Place labels in largest area available, truncating the label to fit.
6. If the map is to be single-color, add the vertices of the rectangle to the polygon in a manner than retains a simple closed curve. This ensures that the labels will not be shaded over.

Feasible regions are found using the shading program to compute a set of parallel lines (horizontal or vertical) spaced a character height apart. These lines approximate a set of rectangles, and the number of characters which will fit into the rectangle is easily computed. In the first two attempts, the search for feasible regions begins in the center and works outward, building a set of rectangles sorted in descending order of length. These are then further sorted by spatial order, top to bottom. A list of suitable regions corresponding to each word of the label (step 4) is prepared, indicating the upper and lower spatial boundaries allowed for that word (i.e., the first word of a label must go above the second word for the horizontal case, and to the left for the vertical case). Finally, a loop is begun in which, at each iteration, the longest region in range for the longest word is chosen, the word is centered within the region (truncated if necessary) and the remaining range lists are updated.

In the example depicted in Figure 4, the label SAN FRANCISCO was broken into two words and placed in the optimum regions.

Figure 4. PLACEMENT OF LABELS IN A POLYGON



Regions

Order by Length	Vertical Order
1	5
3	3
2	1
4	2
5	4

Range	List
SAN	FRANCISCO
5	3
3	1
1	2
2	4

C. The Shading Algorithm

For multi-color maps, the shading algorithm is used during the setup phase to produce a masked region which is saved as part of the data structure. For single-color or cross-hatched maps, a different texture will be used for each characteristic being mapped, so the shading must be redone each time.

The shading algorithm in CARTE was adapted from a paper from the Polytechnic Institute of Brooklyn by William Dwyer, who credits Dr. Frank Sinden with its invention. The subroutine SHADE accepts a polygon of N vertices, a vector $\vec{u}$ normal to a family of shade lines, and a distance d between shade lines. It returns an array of cut points for each shade line sorted in order of albegraically increasing distance from the origin of the coordinate system.

The sides of the polygon are traversed in order from the vertex closest to the lowest shade line to the vertex farthest, and then back down. The cut points are stored in arrays X(I,J) , Y(I,J), for the Ith cut point along the Jth shade line. The cut points are then sorted as follows:

Let $\vec{W}$ be a unit vector normal to $\vec{u}$ (i.e., parallel to the shade lines). Then the points $\vec{P}_{i,j} = (X(I,J),Y(I,J))$ for shade line J are sorted in the order

$$(\vec{P}_{i,j} \cdot \vec{W}) < (\vec{P}_{i+1,j} \cdot \vec{W}).$$

In addition, if $|\vec{P}_{i,j} - \vec{P}_{i+1,j}| < \varepsilon$, where ε is some resolution parameter (typically, $\varepsilon \sim 0.01d$), this segment is eliminated. The cut points are thus paired, with the first point of a pair being the entrance to the polygon along a shade line and the second being the exit.

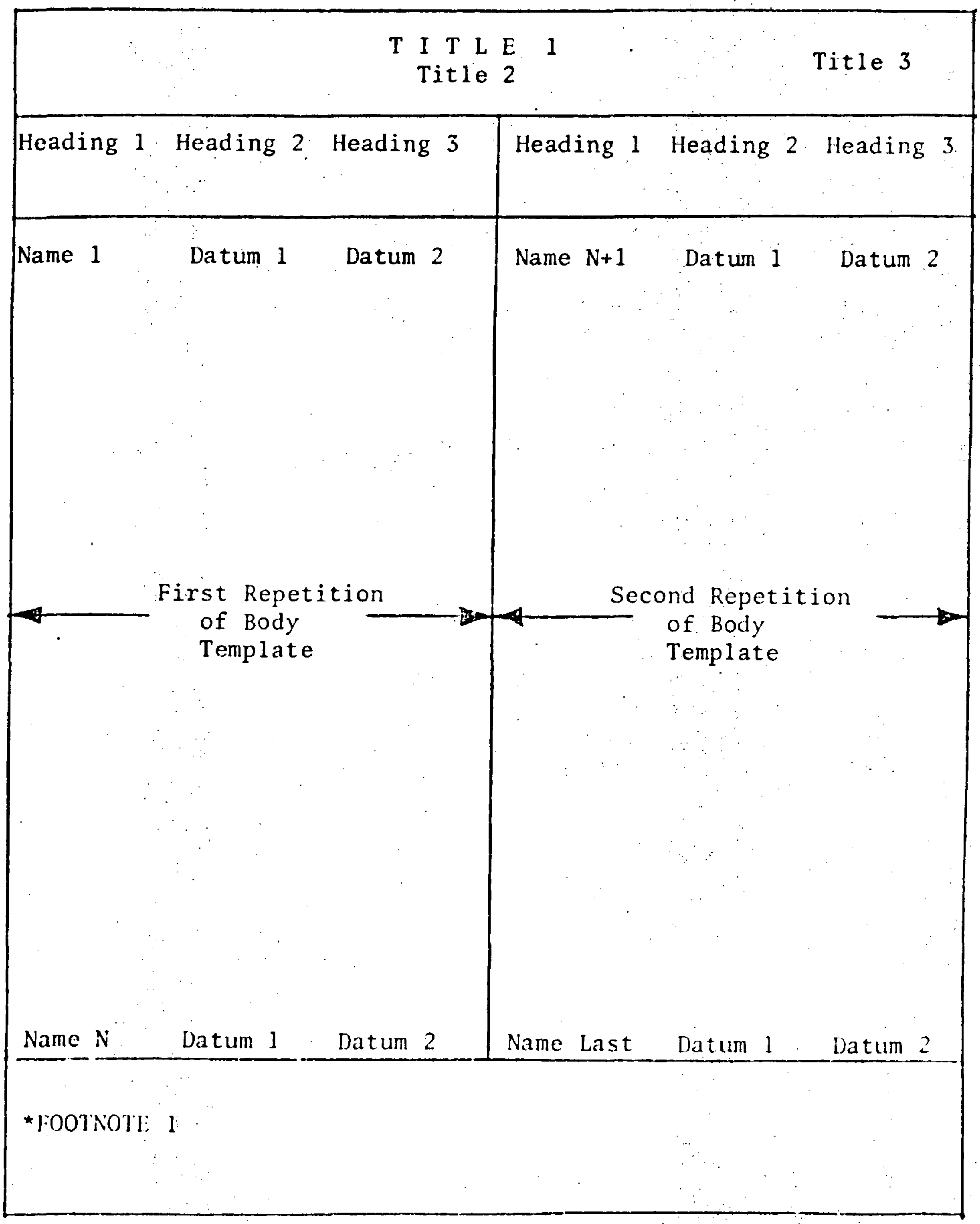
To produce a cross-hatched map, the shade routine is called twice - once with a vector $\vec{u}$ and once with a vector normal to $\vec{u}$. By varying the distance d between shade lines, and the direction of $\vec{u}$, at least eight distinct textures are available.

For dot screen or total masking (for multi-color maps) the shade routine is called with a vertical vector (horizontal shade lines) and a spacing appropriate to the dot size or line width. The best resolution is obtained by total masking using two raster point spacing (out of 4096 addressable points), providing a 2048 raster point resolution on the film. For dot screens, dot sizes range from 2 to 8 raster points in order to produce 3 distinct tones for each color, so the resolution is reduced to 512. This can give a stair-step effect along diagonal borders.

IV. The Report Generator and the Data Set Specifications

A table of the data being represented on a thematic map is often a useful addition to an atlas. The report generation module in CARTE allows user design of such tables thru directives which specify the format of the table (see Figure 5). The table is built from a template which contains key items specified by the directives, such as titles, column headings, and footnotes. Items in the table are taken from the data set, formatted according to the directives and output on a separate frame.

Figure 5. TEMPLATE FOR THE TABULAR REPORT



A. Form of the Data Sets

Data sets are composed of data items, each of which normally consists of three elements: one or more keys for matching, zero or more words (ten characters constitute a word) of area name, and one or more data values. The user may specify the format in which to read each data item by the *FORMAT directive. A data set is completed by a header card indicating the number of data items and the number of keys, words of area name, and the data values per data item.

The <u>KEYS</u> are the same geographic area codes that appear on the map file. The match between keys on the data and keys on the map file is specified by the KEY directive:

*KEY, key i on data = key j on map , (key k) = (key l,) ...

For example,

*KEY, 1 = 8

indicates that the first key on the data card must match the eighth key on the map file.

The NAMES are usually the names of the geographic areas on the map. Currently, the names are used only in the report, and can consist of up to 30 characters.

At least one datum is required on each data card - this is the statistic being represented on the thematic map. This item can be any of the following:

1. A number, to be used by the distribution module in assigning color or shade codes to the areas.
2. The color or shade code itself (i.e., a number between zero and the maximum number of colors or shades desired).

3. A nominal assignment (such as FORESTS, WETLANDS, etc.).

Any additional data items will appear in the table, but only one is used for generating the map. This is specified by the *EXTRACT directive.

B. The Report Template

Since the report generator must handle data for sets ranging from less than ten to several thousand areas, it was written to be flexible and algorithms were included to do most of the format specification automatically. Certain elements are assumed to be present for any table: the data to be displayed, one or more titles, column headings, and row headings (the area names from the data set). Figure 4 shows this basic report template. The report body template may be repeated several times across one frame, in order to fit more data on one page.

The text and placement of the titles may be specified thru the *REPORT directives. The text of column headings and footnotes is entered by the *HEADINGS and *NOTES directives. The placement of all other elements is done automatically within the general template above.

C. Some Features of the Report

In many applications the areas being mapped occur as elements of groups in the data (e.g. counties of states, states of regions). As it is important for these to be clearly delineated in the table, an array of keys, one for each list (the first specified on the *KEYS directive) is required, and, if an area's key value is zero, then it is set off from the rest of the body of the table by one line being skipped before it, its name being entered in capitals, and the text backspaced one character.

There are several other useful features which are included as options that the user can specify for table generation. Any column of the table, including the row names, can be input in either numeric or character form, column headings can be spread over more than one column, any footnote can be bound to a particular title, column or row heading, and as implied above, more than one column of data can be input in order to display with the characteristic mapped other values of interest. The program will also place commas in numbers as needed, indicate data suppressions by an asterisk, and make multi-page tables as necessary.

D. Generating a Table

With the possible exception of the titles, the user specifies only qualitative information to the table generator. The program handles the quantitative details.

Several operations must be completed before the format of the report body is decided.

First, the program counts the number of lines that will be skipped by examining the array of keys, obtaining an accurate count of the actual number of lines to be used by the table. Then the field width of each column is estimated by examining the row headings and data values, counting characters in the case of alphanumerics and computing a floating point format with space allowed for commas as needed, in the case of numerical values.

These estimates are then compared with the individual column headings and adjusted as necessary. The space required for titles and footnotes is computed. Finally, the values of the parameters defining the body of the report can be computed.

The key variable for the report is the number of times the report body template is to be repeated across the page. In CARTE, this is calculated by the following formula:

R = min((NL-1)/LP+1,w/(C*S(K)))

where

R = the number of times the body template is to be repeated

NL = the number of lines to be drawn in table body

LP = the number of lines possible per template repetition

W = frame width

C = total characters per data line

S(K) = horizontal width of characters at size K.

This formula is tested for varying values of K and LP until it is apparent that all the information will fit on one frame or not. Once this value is known, the tab stops for the body, centering information for headings, and actual number of data lines to be displayed per body template repetition are computed.

The table is then produced. While looping through the data items, at each new frame, the report sends out the titles and footnotes and a few major lines; at each body template repetition the module sends out the column headings and the tab stops for the body of the report. Each data item constitutes one line in a body template repetition which is encloded in a temporary buffer until it is complete.

V. The Distribution Module

The distribution module transforms a set of data into a set of color codes to shade the map.

There are three methods of generating colors from a data set in CARTE. Two of these merely divide the data into groups at given division points. This occurs when internal ranges are given or nominal data is used. The third, automatic distribution with rounding, should be selected when a user does not know where to divide the data. The algorithm is set to attempt an equal number of area items per division. This was chosen because it is a relatively simple yet informative method and in the case of nearly equal areas, results in a nearly equal division of the map into shaded regions. It was decided that the usefulness of the mean or equal area approach did not outweigh the cost, especially because the rounding process, needed for a pleasing display, disturbs most distributions anyway. Dividing data into equal groups is a straightforward task except in cases where the distribution is extremely skewed, in which case some arbitrary range must be imposed.

In CARTE's algorithm, the data is first sorted into 100 bins by a simple transformation. Thus the smallest unit of the data space is effectively 1/100th. Then these bins are divided into an arbitrary number of subgroups, usually eight or less, the number being set by the user, thus generating an expected number of items/group. While cycling through the bins, a count is kept of the number of items accumulated since beginning and since the last division was made. When the expected number of items is past a decision is made as to where to place the dividing line. After each such decision the expected number of items per subgroup is recalculated from the number of items actually allocated, and if needed, fewer subgroups are used.

After transforming the calculated division points back into the data space, the program procedes to round these into acceptable numbers for presentation on a map. The user controls the number of significant digits to remain in each division (ns) and the range these digits are allowed to take over the set of division points (nmag). This second parameter insures that the division points will all end in the same column. As an example, assume the division points calculated are:

.712, 6.711, 24.381, 78.172.

With ns=2, nmag=3 these will be rounded to:

.7, 6.7, 24.0, 78.0

A maximum of two significant digits are allowed, but the first point is allowed only one because the divisions must fit in a three column range. If nmag = 2, then the result of rounding would be:

1., 7., 24., 78.

The process the program follows is to normalize all division points to less than one, multiply by the calculated ns, add .5, fix the number and multiply again by the adjusted normalization factor. In some cases the division boundaries will now be equal. These are then adjusted by the minimal allowable amount (given ns, nmag) and divisions are eliminated if the adjustements make them extend too far beyond the range of the data.

The colors are then assigned to the data items, followed by the encoding of the division ranges into the display form (the color shade code). The minimal field width for the set of divisions is calculated and used, including space for commas in large numbers as needed. Also, a histogram of the data as divided is automatically compiled and displayed.

The shades or textures, stored in data item order, are then assigned to the polygons on the map file. There are two steps in this process. First each polygon is assigned its appropriate color. Then the polygon list is reordered so that all areas with the same color are grouped together. This enables the use of one film file, thus allowing an essentially unlimited number of shades, and requires random access to the map in the graphic output module.

VI. The Graphic Output Module

The next function step in CARTE is to put out the map image as specified on the appropriate files. There are two parts to this process: putting out area outlines, labels and shading, and creating the titles, legends and other cosmetics.

A. Area Outlines, Labels, and Shading

There are two cases to be considered here. The first is that of cross-hatch mapping. In this case all of each map is on one frame, but more importantly, the areas are in NICKEL format [2]. Thus for each area, for each map, the display file is generated from the stored map. The process is straightforward conceptually, but the generation of shade lines relatively costly. It is the fact that the shading for an area will probably vary that requires us to store the map in its source form. The second case is for multi-color mask or dot maps. Here, in the setup phase, the display file for the entire map was generated and saved. Thus to generate an area's outline, labels, and shading, we have only to transfer its pre-generated display code to the color buffer with dot sizes being changed for dot maps. As can be expected, the cost of this method is much cheaper than that of cross-hatch.

The effect of the color, generated by the distribution module, depends upon the type of map being produced. In the case of cross-hatch, the orientation and spacing of the shade lines is affected. In the case of mask, the choice of output frame is affected. In the case of dot, the choice of output frame and dot size is affected.

B. Creating Titles, Legends, and Other Cosmetics

The titles and legends may vary from map to map. Thus the display file for them must be generated each time.

Every title is described internally by several characteristics: a centering point, character size and orientation, number of lines allowed below the centering point, width of a line in characters, number of characters, and the text of the title. The title drawing routine takes this information and centers it below the given point, creating as many centered lines of text (broken at blanks) as necessary.

Drawing the legend is a bit more complicated. The program is given an area in which to place the legend, the legend text, the text describing the color code, the allowable height and width range for the box for each color, and the allowable range of character sizes. Since the given legend area and/or the number of colors used, and/or text lengths may vary from map to map, the allocation of space within the legend box is recalculated for each map.

The process is as follows: first, for maximum character size, enough space is allocated to draw the legend. Then with the remaining area, the program decides whether to place the color code and boxes in a row horizontally or vertically. Once this is decided, an estimate is made of the unit width or height respectively (the space needed per interval). This estimate is then tested to see if the code and box for a color can fit in. If not, the estimate is revised by changing the number of text lines allowed, the code and legend character size, and the box size until a solution is reached or infeasibility is shown. The algorithm is set to maximize the size of the individual color boxes. The area required by the legends and color codes is derived by an inverse operation of the title drawing routine, i.e., given text, its display space is computed.

Once the space for each element within the legend box has been allocated, the elements are centered within the given area and drawn. If so directed, the program will give the type of fit tried, the text displayed, and the spacing information it computed.

VII. Data Structure and Memory Management

For large maps, memory becomes a scarce resource. Thus CARTE minimizes its memory requirements by storing the map on auxiliary storage in a compact form (for screen type maps); by having the size of many of its working arrays determined at execution time; by swapping these working arrays in and out of memory as needed; and by using only as much small core memory and auxiliary storage as needed.

A. Storing the Map

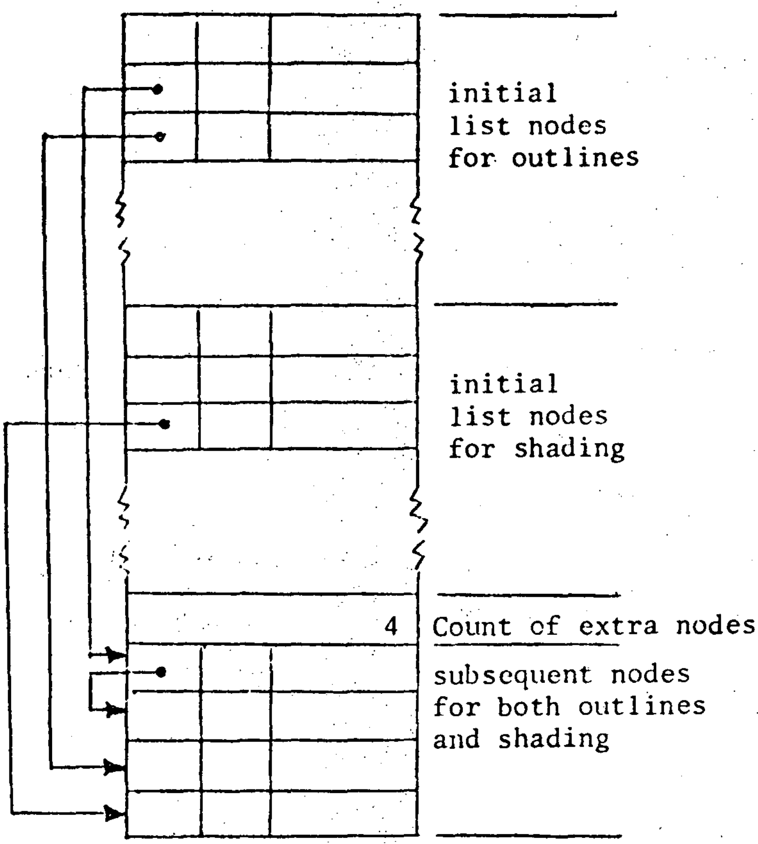
A map in CARTE consists of a series of closed polygons, each describing an area that can be shaded. Accompanying the points for each polygon is a set of descriptors primarily for matching with the data and a set of labels or area names associated with that polygon.

The set of polygons comprising a map is stored differently depending upon the type of map to be produced. In cross-hatch mapping, an image of the map input file is kept on disk thus allowing for later modification of the map (e.g., further windowing, changing the viewport). This approach requires the calculation of shade lines for each map. In the case of screen type maps, the shade lines need be generated only once, so a different, less expensive, approach may be used. The approach is to create a graphical data structure as described by Newman and Sproull [3]. In this case, the raw map is no longer the data base, the display file created in the setup phase of the program is used as the data base for the map. This has two major advantages: it requires less space to store and the cost of generating a new display file for each map is eliminated. Its only disadvantage is that the map cannot be easily redesigned later in a run.

In this approach, two blocks of memory are created for each polygon: one for labels and outline, one for shade lines. These display file blocks are stored in random access mass storage in sequence as generated. Since it is unpredictable whether or not the shading for a polygon will be used on any given map, pointers and block lengths are kept for each block.

The method of creating the graphical data structure is to save the contents of the color buffer as the test frame for the COM is generated. This has the advantage of minimizing memory requirements during the creation and later generation of maps. This implies that when the display file block for a polygon's outline or shading is longer than the buffer size it is broken up into buffer loads. Thus the array of map block pointers is an array of singly-linked lists of the buffer loads comprising a block. Figure 6 summarizes the structure of this array. It is organized so that the first buffer load (usually the only one) is accessed by the polygon number plus a constant.

Display code blocks for each polygon are
described by two singly linked lists -
one for outlines and labels
one for shading



Hypothetical Picture of
Map Pointer Array

List pointer for polygon i is
i for outlines and labels
i+npoly for shading

List node structure

Link	Word Count	RAMS Address

Link = 0 indicates terminal node of list
≠ 0 gives index of next node

Figure 6. Structure of Map Pointer Array

B. The Dynamic and Fixed Length Structures

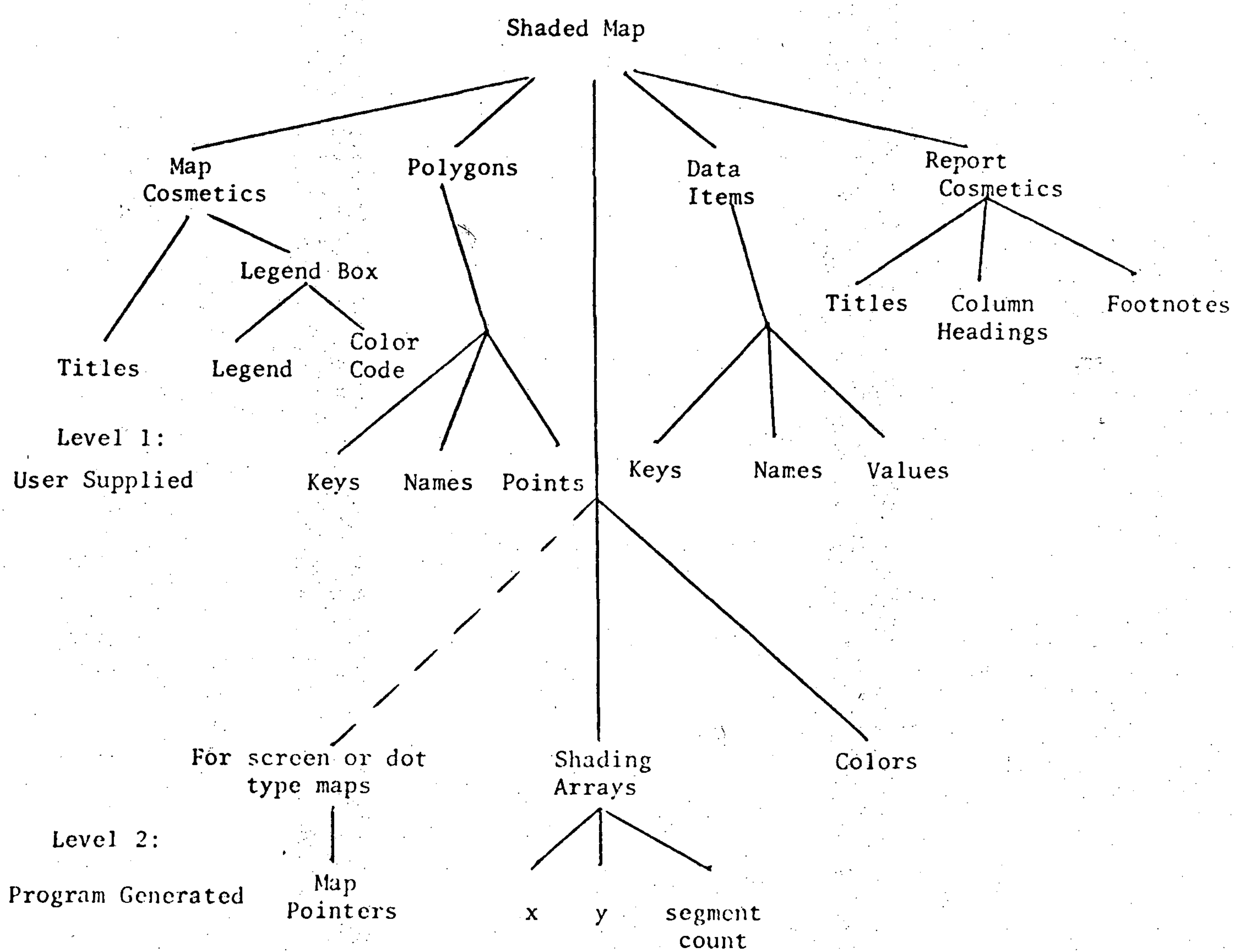
The maps drawn by CARTE so far have ranged from 9 to 3100 polygons. Thus the size of many arrays varies widely, and more importantly, would overflow memory if kept in core at one time. For these a variable sized dynamic allocation was adopted. Other arrays, such as those for map and table titles, have small variation in size. These are kept at a fixed size.

Figure 7 depicts the elements of CARTE's data structure. The terminal nodes of the tree comprise the major arrays of the program.

The fixed length arrays are those under map and table cosmetics since these vary within small ranges. Also of fixed length is the array of polygon names, because the names and points for only one polygon at a time need be in core. The points array is variable because the maximum number of points needed can range from ten to thousands. Thus including the points array there are ten arrays of variable sizes. The size of the variable data arrays is computed from the data set header card, while the size of the map arrays is computed from a preliminary pass through the mapfile in the setup module. The next section describes the management of the variable arrays.

Figure 7. DATA STRUCTURE OF CARTE

Terminal nodes are data structure elements



C. Memory Management

Because the operations of the program are known, the management of memory is also known. Small core memory, where the current arrays are kept, is treated like a modified stack. Thus the two basic stack operations are used: putting a block on top of the stack, and freeing the block on top of the stack.

Any block in the stack is accessible and the operation of freeing a block low on the stack frees all memory above it.

When a block of memory is created, a status flag indicates whether this is a place for new data or a place to put data being saved in auxiliary storage, in which case the block from auxiliary is moved into its new small core location. When a block of memory is freed, the status flag indicates one of three options: free the memory; move the block down to stack location x, freeing all above; or free the memory, saving the block at the next available location in auxiliary storage.

Thus, for each of the ten arrays, a block length, a small core address, and a large core address are maintained. The memory map in Figure 8 depicts the contents of SCM and auxiliary storage over time.

Figure 8. MEMORY MAP FOR CARTE* OVER TIME

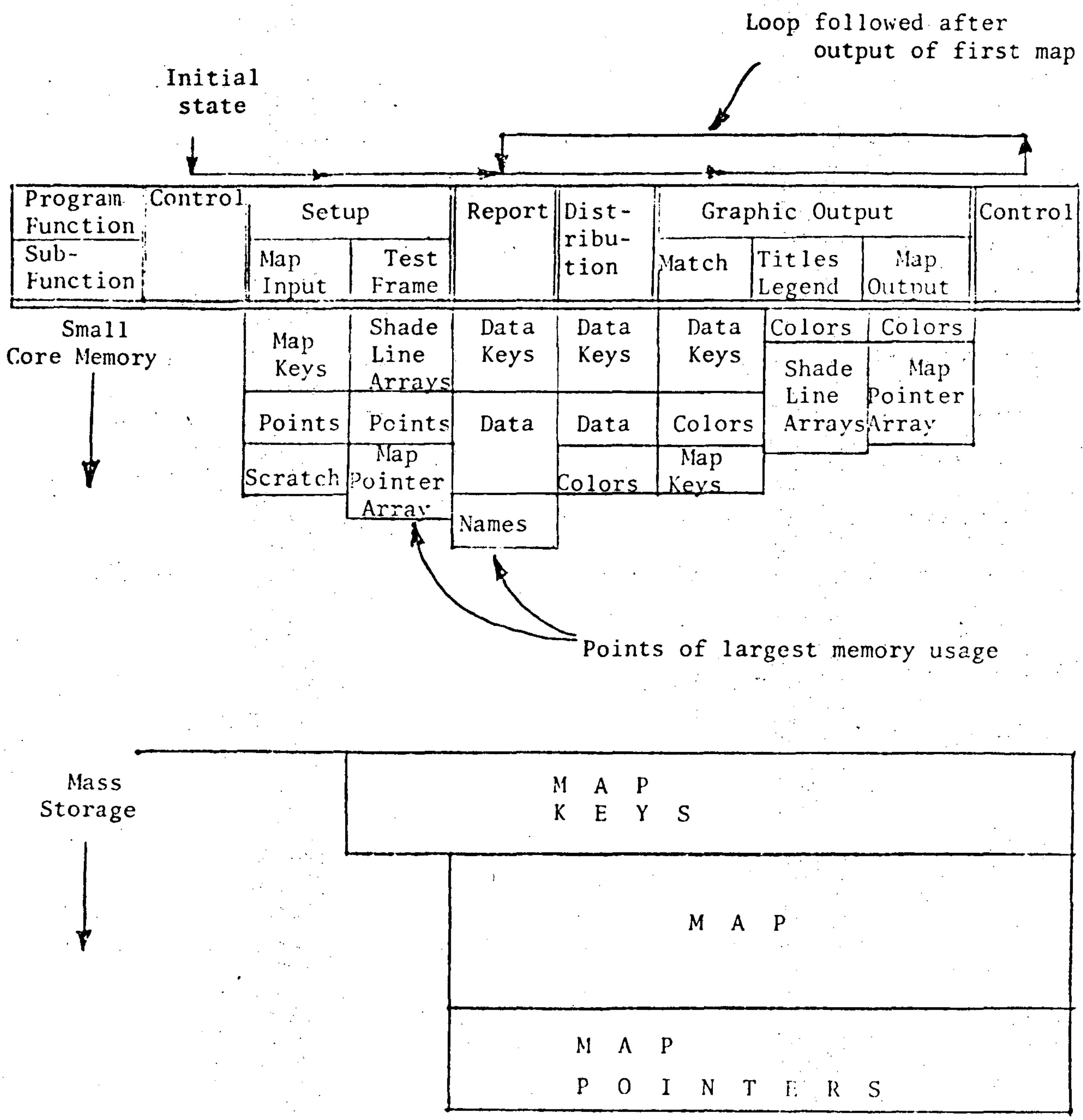


*Chart is for dot or mask type maps

D. Random Access Mass Storage

The CARTE program is designed to make efficient use of random access mass storage. On the CDC 7600 in use at LBL, this takes the form of 512,000 words of large core memory (LCM) and two 75 million word disk files. All processing takes place in 65,000 words of fast memory (called small core memory or SCM), so CARTE transfers blocks of data between fast memory and auxiliary storage as required. Large core memory is always treated as a random access file, with dynamic allocation of blocks of memory. In the case the data structure overflows LCM, the program automatically switches to random access disk files. This combination of fast memory, large core memory and disk files allows CARTE to handle a wide range of map files in an efficient and cost-effective manner. Experience with the program has shown that memory management is a necessary feature for a mapping program, since maps produced have ranged from nine areas with a few thousand points to over 3,000 areas with more than 100,000 points (over 200,000 words).

VIII. The Symbol Mapping Nodule

A. Introductory Considerations

CARTE produces shaded maps. It was first designed for choropleth mapping, where geographic entities are accurately described by polygons. Geographic entities may also be points and lines. To shade point data it is necessary to represent each point by a symbol. (Lines can be shaded by expanding them in width.) Shading point data is called symbol mapping. User-defined symbols, each defining a locality type, are placed at specified coordinates on a base map, and shaded according to the range of a common attribute. Examples are allocation of funding by type of sponsor and power output by type of generation facility.

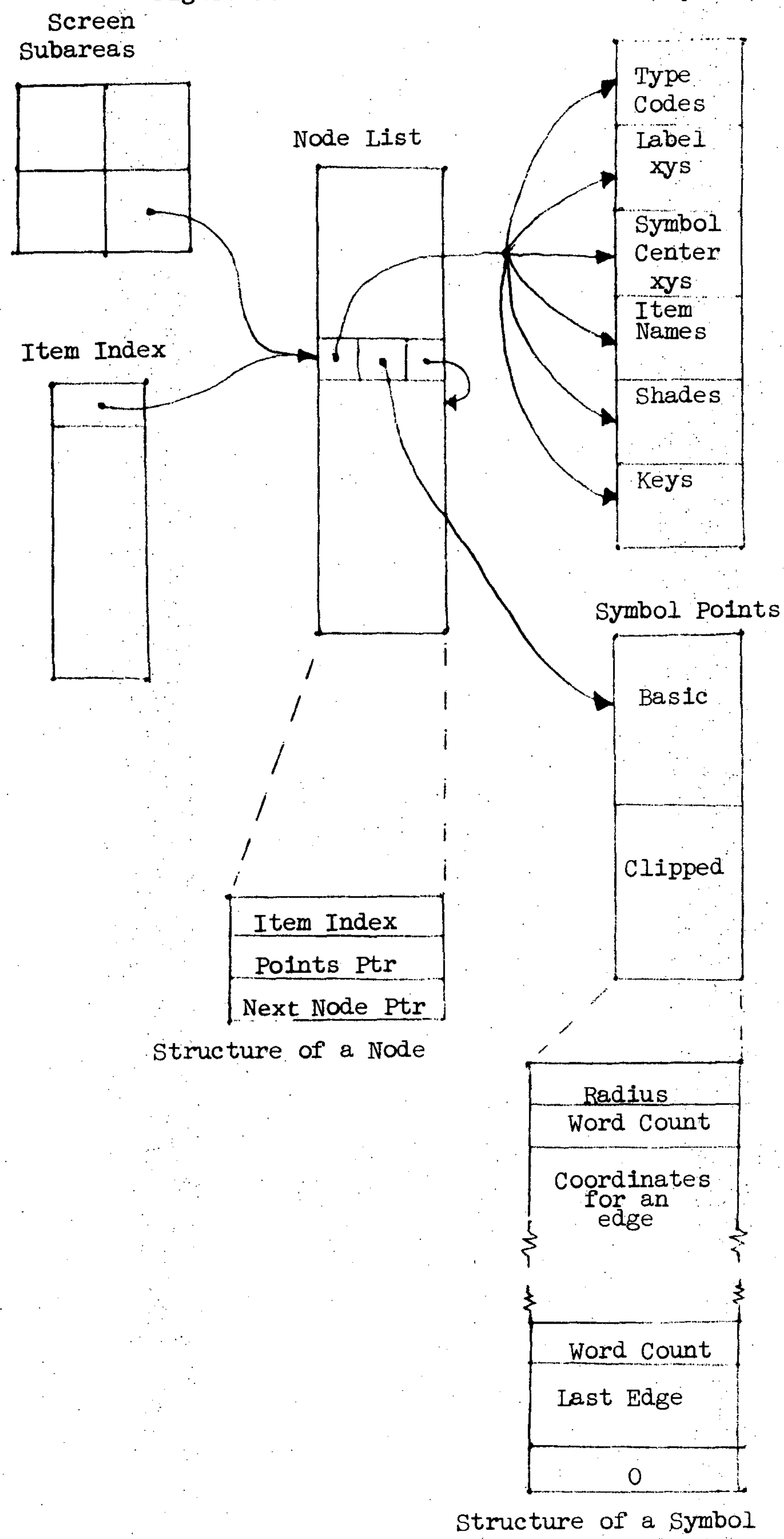
The placement and shading of a polygon presents no problem, but complications arise if symbols are allowed to overlap. (In choropleth mapping non-overlapping symbols are drawn.) If two symbols overlap, it must be decided which goes on top. Thus the conception of depth is added. Also, algorithms for general polygon-polygon clipping are needed, so that just the visible portion of a symbol will be drawn and shaded. And an efficient means of storing and searching the already compiled portion of a map is needed, in order to determine if symbols do indeed overlap.

A symbol may have many instances; it may be placed at many different places on the map. Each instance of a symbol may remain whole, become partitioned into fragments, of have holes (islands) added to it. This requires a flexible data structure for storing symbols. Also, a user may not know the map coordinate for a symbol. He may wish it to be placed anywhere within a geoarea. Thus, the program must be able to place the symbol in a suitable location.

Experience has shown that it is useful to annotate each symbol. Thus some facility for generating (placing) symbol annotation must be provided.

In the output phase, as in choropleth mapping, the data items are output

Figure 9. Basic Data Structure for Symbol Maps



in shade, not data item, order. An index pointing to all necessary information to draw and shade each symbol must be built.

These considerations are built into the symbol mapping module of CARTE and will be discussed in more detail below. The basic data structure will be discussed first, as it is the foundation upon which the algorithms are built.

B. The Basic Data Structure

The following sections describe the basic elements of the data structure used in symbol mapping. They are basic in the sense that they are global to the compilation and output phase of producing a symbol map.

1. Symbol Point Storage. A symbol may be defined by a collection of closed polygons a code defining its type, and a text definition. The points of each user-defined symbol are stored in a block which includes the radius of the symbol about the origin, and the actual points as a series of closed polygons, also relative to the origin. This makes it easy to place a symbol anywhere on the screen and allows many data items to use the same symbol definition. The radius simplifies checking for overlap.

The outer edges of a symbol go in one direction while islands, or inner edges are indicated by a flag and go the other way, e.t., counter-clockwise on the outside, clockwise on the inside. This is essential for correct traversing of the symbols when clipping.

The collection of basic symbol definitions is saved in RAMS until it is time to compile the symbol map. At this time they are loaded into dynamic memory allowing them to be treated in the same way as generated symbols by other parts of the data structure. Variations on the basic symbols generated by overlap are stored in the same manner as the basic symbols. They are stored at the next available location in dynamic memory.

2. Data Item Elements. Associated with each data item to be represented on the map are, as in choropleth mapping, a set of keys, a name, and a shade. Also

stored in data item order are symbol type code, and space for a symbol and label xy coordinate (to be generated if not already supplied).

3. Structure for Searching the Map. The screen is partitioned into subareas, each larger than any permissible symbol radius. Each subarea of the screen is represented by a linked list of nodes indicating the data items placed in that subarea. As each data item is placed a new node is added to the appropriate subarea list.

A node consists of three parts -- the data item number, a pointer to the points defining that instance of the appropriate symbol type, and a pointer to the next node on the list. The data item number acts as a pointer to the elements described in section 2.

There may be data items which are not drawn. Thus the nodes are stored in dynamic memory interspersed with points for new symbols.

4. The Index. The index consists simply of pointers to the appropriate node locations in dynamic memory. These pointers are stored in data item order.

C. Algorithms

1. Overview. The process of producing a symbol map is divided into two phases — that of compilation and that of output. This enables the production of more than one map, i.e. different shading, from one compilation. Figure 10 shows a flow chart of the compilation phase, which is described in sections 2-5. This condensation allows the compilation phase to be relatively costly, since it is performed only once. The clipping algorithm also can be relatively costly, since it is executed for just a small subset of cases.

2. Depth. Selection by depth forms the outermost loop of the compilation module. Depth is defined by symbol type in order of definition. Thus if symbol type 1 to n are stacked on top of each other, symbol 1 will be completely visible, while symbol n will be least visible.

This means all data items with symbol type 1 are compiled first, all those with symbol type n last. If two symbols of the same depth (i.e. type) overlap, the one compiled first will be most visible.

3. Symbol Center and Label Coordinate Generation. In some cases, the symbol is to be placed anywhere within a polygon from the base map. Thus the symbol center must be generated. After finding the appropriate base map area by matching keys, the program shades the area finding feasible regions in the same manner as automatic label placement. These regions are tried until one with no overlap is found. If no such region exists, then the symbol center is the center of the rectangle circumscribing the area.

Label coordinates, if not supplied, are created so that the item name will appear one character width to the right or left of its symbol.

4. Searching the Map. After being transformed from data to screen space, the symbol center is combined with the points of its basic symbol type. These absolute points are used by the clipping algorithm. Using the radius of the symbol, part of the already compiled map is searched for overlap.

Figure 10. Overview of Symbol Compilation Phase

```
For depth d = 1 to n do

   Find data item with depth = d

      Load symbol center
      If center not supplied, then generate
      Transform center to screen space
      Load symbol points and radius

      For each of nine subareas do

         For each node on subarea list do

            If old symbol and new symbol overlap then

               Load points of old symbol
               Clip new symbol to old symbol

      Store new symbol node
      Store index entry
      Store points if clipping occurred
      Store center and label coordinates
```

The node lists for at most 9 subareas are traversed (the one the symbol center falls into and the eight surrounding). Since the screen is partitioned into 100 subareas, this search includes at most 1/10 of the screen. The surrounding subareas must be checked because, even though the new symbol center falls in only one subarea, its points may overlap into others, and symbols centered in other subareas may overlap into the subject's subarea.

For each node, the circle described by its center and radius is compared with the circle described by the new symbol's center and radius. If the circles overlap, it is probable that there is symbol overlap and the clipping routines are activated. Only then are the points of the old symbol loaded.

This method provides a quick and efficient means of searching the map. When the search is complete, a new node is added to the appropriate subarea list and the symbol points stored, if clipping occurred. An entry into the index for that data item is also made.

5. The Clipping Algorithm. When the probability of overlap has been established, a four step algorithm is followed to clip the new symbol so that it conforms to the boundaries of the already compiled symbol. A symbol is here defined as a set of one or more fragments (closed polygons representing outer edges), and zero or more islands (closed polygons representing inner edges). The four steps are -- build a relation table, compute intersection points, traverse the symbols, and compose the clipped symbol.

First a relation table is built, indicating precisely the type or relation that holds between each component of a new (subject) and old (object) symbol. There are four possibilities -- S inside O; S outside, enclosing O; S outside, disjoint from O; and S intersecting O. These relations are determined by testing each point of each component of S against the points of each component of O, using the Shrimrat Straddle algorithm described in the paper, Point in Polygon Algorithms by HRP Ferguson. If S is outside O, the reverse test of each point of the O component with respect to the S component is required to determine enclosure of disjointness.

Next, all intersections for the two symbols are computed. This step uses the point-slope method of computing intersection points between line segments. To facilitate traversing, each intersection point is represented as: x, y, its S line, its O line, and a status flag indicating whether or not it is active.

With this information and the different orientations of inner and outer edges, the symbols are easily traversed, generating fragments and/or composite symbols. The most difficult part of the traversing algorithm is finding the correct point to start the traverse. An intersection point must be found which can act as the origin of an S line segment. This is done by testing the midpoint of the line segment formed by a prospective intersection point and the next S vertex as to whether it is outside the O component the intersection point was generated from.

Once a correct starting point has been found, the algorithm is as follows:

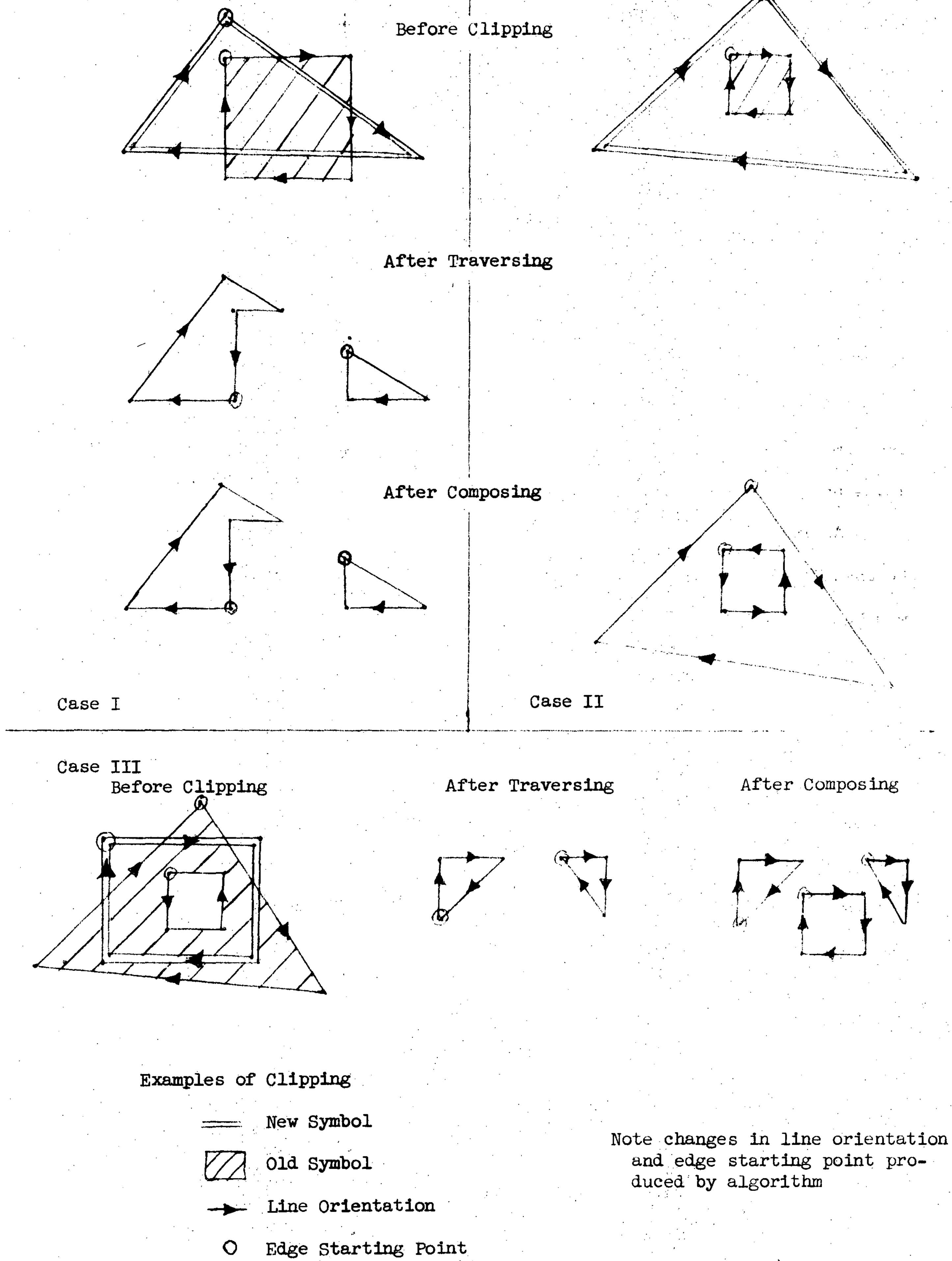
1. Follow S (in line direction) until next intersection point.
2. Follow O (opposite line direction) until next intersection point.
3. If not starting point, go to 1.
4. Compute next starting point, if any, and go to 1.

Finally comes the step of composing the clipped symbol from the collection of original S and O edges and the polygons generated by traversing. This task is made easy by the relation table. The algorithm is as follows:

1. ENTER all components of S not inside the outer edge(s) of O.
2. ENTER all components of S inside or intersecting an inner edge of O.
3. ENTER all components of O inside an outer edge of S and outside all inner edges of S.

For this procedure to work, two things are needed. First, the identity of each

Figure 11. EXAMPLES OF POLYGON CLIPPING



S and O component and polygon generated by traversing must be known. This can be done simply by keeping a list of pointers to the current use(s) of each component, updated during the traverse. In complex cases, several components can be used in one generated polygon, or one component can be split into several polygons.

Second, the ENTERing procedure must perform several additional functions besides simply entering points. It must make sure all uses of a component are entered, deactivate all components used, and preserve line orientation. Any outer edge of O will become an inner edge and any inner edge of O will become an outer edge of the composed symbol, if entered.

The process of composing a clipped symbol is illustrated in figure 11. The composed symbol replaces the original symbol being processed and the program moves on to the next node on the sector list.

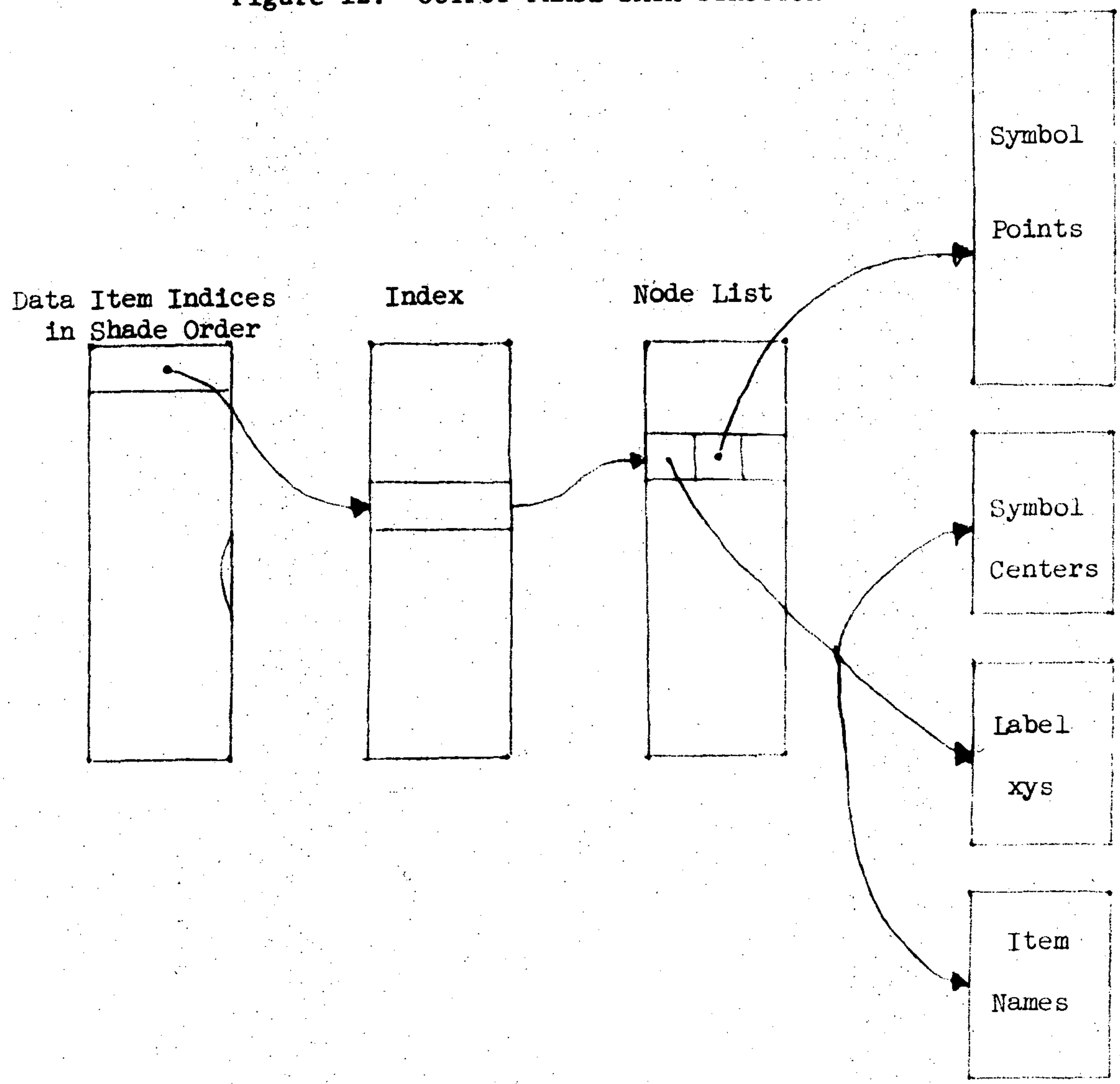
An interesting feature of the algorithm is that inner edges do not have to be associated with their enclosing outer edges. The program must simply know whether each component is an outer or inner edge. This is because symbols are considered as a whole both in the clipping algorithm and in the output phase of the program.

6. Producing the Map. Once the compilation phase is complete, the symbol map can be produced for many different sets of data values. Because output is on only one film file, two passes must be made through the data structure.

The first pass draws the symbol outlines. The program simply passes through the dictionary, drawing each component of each symbol separately. Names are also drawn during this pass.

The second pass produces the shading. First the shades are processed to produce a list indicating which data items are to be drawn for each color frame. Then, as each data item index is encountered, the appropriate dictionary entry is accessed, and, through the node, all points and the center for that symbol are combined and passed to the shading algorithm which generates the correct shading.

Figure 12. OUTPUT PHASE DATA STRUCTURE FOR SYMBOL MAPS



D. Added Directives

For the functions described above to be made available to the user, two (only) directives were required: One for defining symbol types, and one for plotting a symbol legend. They are:

```
*USER-DEFINED SYMBOL, code, text def, N1, points, N2, points....
                                      outer edge, 1st inner edge, etc.

*VERTICAL SYMBOL LEGEND, N, xmin, xmax, ymin, ymax
```

REFERENCES

1. SYMAP Manual, Version 5.15, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, Mass. 02138

2. Holmes, H.H., Austin, D.M., and Benson, W.A., "The MAPEDIT System for Automatic Map Digitization", Computers and Graphics, July, 1974.

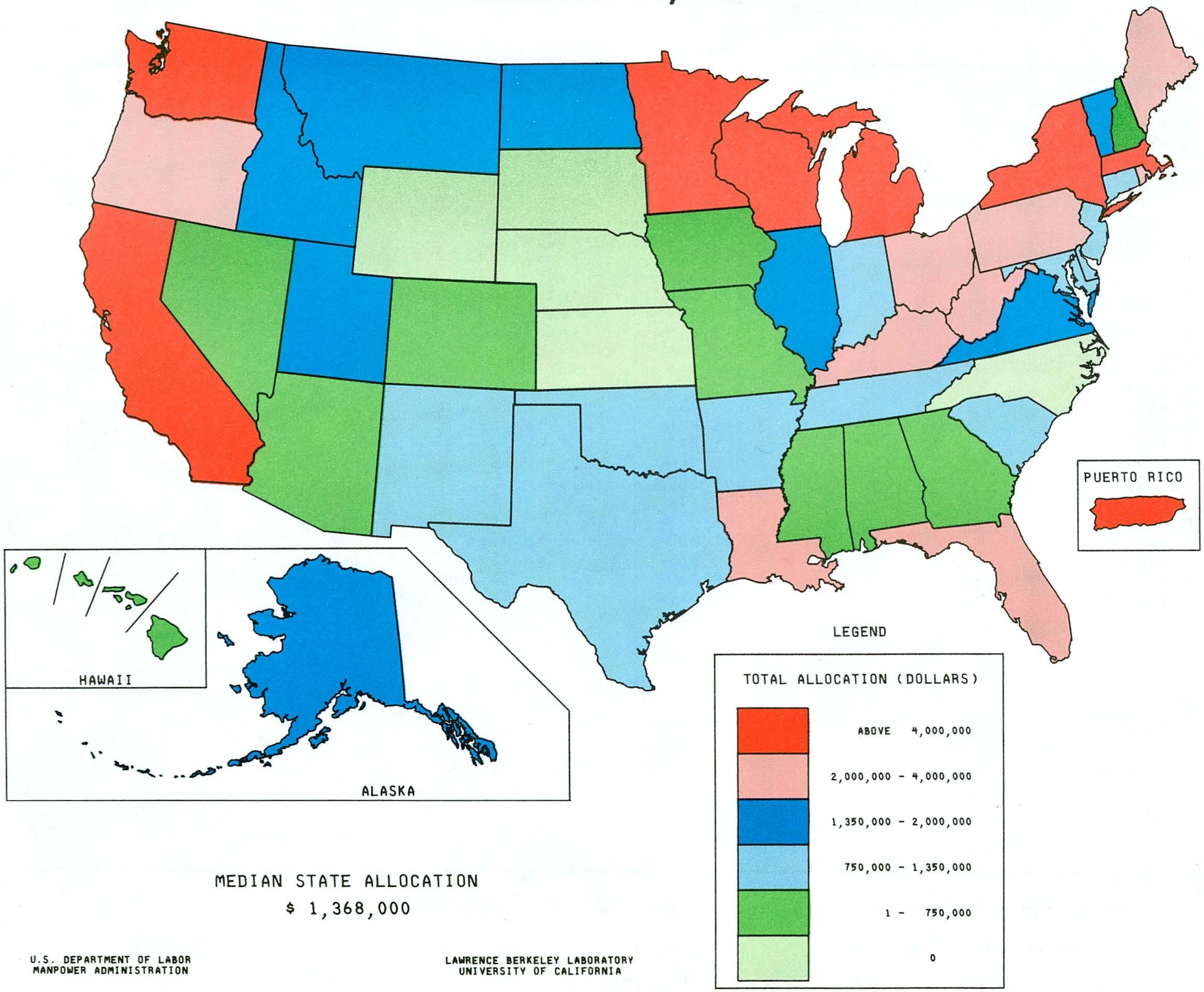
3. Newman, William M., and Sproull, Robert F., Principles of Interactive Computer Graphics, McGraw-Hill, 1973.

4. Robinson, A.H., and Sale, R.D., Elements of Cartography, John Wiley and Sons, 1969.

Map and Table Examples

1. U.S. by State (color)
2. U.S. by County (color)
3. U.S. by SMSA (color)
4. Denver base and inserts (color)
5. Upper Mississippi (cross hatch)
6. Prime Sponsor Symbol Map (color)

MAP 2 - Allocation of State Government Portion of CETA Title II Funds For Public Service Employment - FY 1974
United States by State



U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

TABLE 2 - ALLOCATION OF STATE GOVERNMENT PORTION[1] OF CETA TITLE II FUNDS FOR PUBLIC SERVICE EMPLOYMENT - FY 1974

UNITED STATES BY STATE

U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

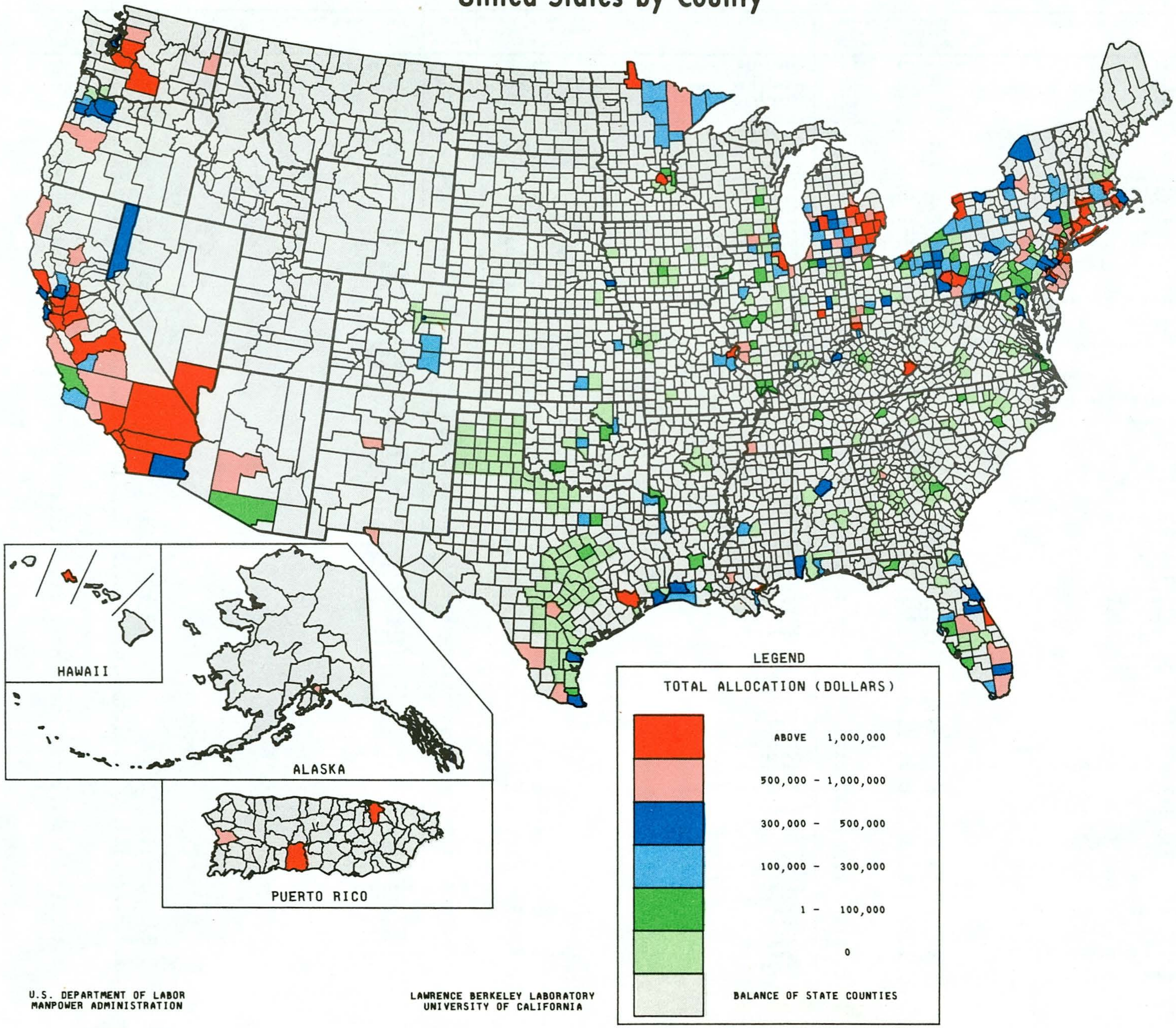
STATE	TOTAL ALLOCATION (DOLLARS)	STATE	TOTAL ALLOCATION (DOLLARS)
Alabama	479,465	New Hampshire	311,234
Alaska	1,842,989	New Jersey	1,019,144
Arizona	140,500	New Mexico	1,330,479
Arkansas	852,774	New York	5,576,351
California	5,212,978	North Carolina	0
Colorado	205,078	North Dakota	1,368,400
Connecticut	769,000	Ohio	2,741,000
Delaware	1,266,312	Oklahoma	947,054
District Of Columbia	2,258,500	Oregon	2,544,293
Florida	2,914,400	Pennsylvania	2,049,722
Georgia	273,353	Rhode Island	2,038,887
Hawaii	738,350	South Carolina	1,200,383
Idaho	1,996,800	South Dakota	0
Illinois	1,379,359	Tennessee	1,334,332
Indiana	1,221,564	Texas	914,039
Iowa	494,300	Utah	1,854,100
Kansas	0	Vermont	1,552,279
Kentucky	2,236,856	Virginia	1,779,814
Louisiana	3,915,830	Washington	5,321,565
Maine	2,771,077	West Virginia	3,261,649
Maryland	909,364	Wisconsin	4,496,520
Massachusetts	10,849,185	Wyoming	0
Michigan	4,845,776		
Minnesota	4,793,023		
Mississippi	702,000	Puerto Rico	10,677,698
Missouri	314,600	A.Samoa-Guam-Trust Territories	345,300
Montana	1,860,200	Virgin Islands	246,700
Nebraska	0	Indian Reservations	1,855,000
Nevada	473,759		

1. FUNDS TO BE ADMINISTERED BY STATE GOVERNMENT FOR BALANCE OF STATE AREA

MEDIAN ALLOCATION BY STATE $ 1,368,000

MEAN ALLOCATION BY STATE $ 1,963,000

MAP 3 - Allocation of Total CETA Title II Funds For Public Service Employment - FY 1974 United States by County



U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

TABLE 3 - ALLOCATION OF TOTAL CETA TITLE II FUNDS FOR PUBLIC SERVICE EMPLOYMENT - FY 1974

U S DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

UNITED STATES BY COUNTY

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

STATE AND COUNTY	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
UNITED STATES TOTAL	367,800,738	100.00
ALABAMA	1,339,610	.36
Autauga	0	.00
Baldwin	0	.00
Elmore	0	.00
Escambia	0	.00
Jefferson	421,471	.11
Madison	95,275	.03
Mobile	343,397	.09
Montgomery	0	.00
Tuscaloosa	0	.00
Balance Of Alabama	479,465	.13
ALASKA	2,615,641	.71
Anchorage Division	772,651	.21
Balance Of Alaska	1,842,989	.50
ARIZONA	1,002,288	.27
Maricopa	763,288	.21
Pima	98,500	.03
Balance Of Arizona	140,500	.04
ARKANSAS	971,574	.26
Faulkner	0	.00
Little River	37,100	.01
Lonoke	0	.00
Miller	81,700	.02
Pulaski	0	.00
Saline	0	.00
Balance Of Arkansas	852,774	.23
CALIFORNIA	64,769,428	17.61
Alameda	4,877,670	1.33
Butte	680,703	.19
Contra Costa	1,762,161	.48
Fresno	1,660,842	.45
Humboldt	730,005	.20
Imperial	388,857	.11
Kern	594,905	.16
Kings	176,483	.05
Los Angeles	20,086,364	5.46
Marin	485,600	.13
Merced	512,005	.14
Monterey	724,434	.20
Orange	1,769,800	.48
Riverside	1,353,925	.37
Sacramento	1,945,042	.53
San Bernardino	2,376,363	.65
San Diego	6,829,161	1.86
San Francisco	3,587,946	.98
San Joaquin	1,346,328	.37
San Luis Obispo	29,200	.01
San Mateo	485,200	.13
Santa Barbara	277,600	.08
Santa Clara	1,506,293	.41
Santa Cruz	741,589	.20
Solano	338,600	.09
Sonoma	1,044,513	.28
Stanislaus	1,704,636	.46
Tulare	616,200	.17
Ventura	644,600	.18
Yolo	279,409	.08
Balance Of California	5,212,978	1.42
COLORADO	858,978	.23
Adams	0	.00
Arapahoe	0	.00
Boulder	0	.00
Denver	406,500	.11
El Paso	107,300	.03
Jefferson	0	.00
Pueblo	140,100	.04
Balance Of Colorado	205,078	.06
CONNECTICUT	5,829,029	1.58
Fairfield	1,913,533	.52
Hartford	1,674,439	.46
New Haven	1,472,056	.40
Tolland	0	.00
Balance Of Connecticut	769,000	.21
DELAWARE	1,266,312	.34

STATE AND COUNTY	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
Statewide	1,266,312	.34
DISTRICT OF COLUMBIA	2,258,500	.61
District Of Columbia	2,258,500	.61
FLORIDA	8,982,431	2.44
Alachua	0	.00
Baker	0	.00
Brevard	1,286,721	.35
Broward	478,436	.13
Dade	772,173	.21
De Soto	0	.00
Duval	245,000	.07
Escambia	102,300	.03
Gadsden	0	.00
Hardee	0	.00
Highlands	0	.00
Hillsborough	90,500	.02
Lee	56,900	.02
Leon	35,300	.01
Monroe	147,100	.04
Nassau	0	.00
Okeechobee	0	.00
Orange	427,100	.12
Palm Beach	734,300	.20
Pasco	222,300	.06
Pinellas	118,300	.03
Polk	890,300	.24
Sarasota	52,600	.01
Volusia	408,700	.11
Balance Of Florida	2,914,400	.79
GEORGIA	1,284,951	.35
Bibb	0	.00
Burke	80,300	.02
Chatham	0	.00
Chattahoochee	0	.00
Clay	0	.00
Cobb	0	.00
Columbia	0	.00
Crawford	0	.00
De Kalb	931,298	.25
Emanuel	0	.00
Fulton	0	.00
Glascock	0	.00
Harris	0	.00
Houston	0	.00
Jefferson	0	.00
Jenkins	0	.00
Jones	0	.00
Lincoln	0	.00
Mcduffie	0	.00
Monroe	0	.00
Muscogee	0	.00
Peach	0	.00
Quitman	0	.00
Randolph	0	.00
Richmond	0	.00
Screven	0	.00
Stewart	0	.00
Talbot	0	.00
Taliaferro	0	.00
Twiggs	0	.00
Warren	0	.00
Wilkes	0	.00
Balance Of Georgia	273,353	.07
HAWAII	2,817,732	.77
Honolulu	2,079,382	.57
Balance Of Hawaii	738,350	.20
IDAHO	1,996,800	.54
Statewide [C]	1,996,800	.54
ILLINOIS	9,268,145	2.52
Alexander	52,210	.01
Bond	34,800	.01
Boone	0	.00
Cass	0	.00
Champaign	0	.00
Cook	4,794,000	1.30
Du Page	0	.00
Ford	0	.00

STATE AND COUNTY	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
Grundy	0	.00
Iroquois	0	.00
Johnson	52,210	.01
Kane	121,600	.03
Lake	175,800	.05
Lasalle	67,100	.02
Mchenry	0	.00
Macon	61,400	.02
Madison	787,203	.21
Massac	52,210	.01
Peoria	118,200	.03
Platt	0	.00
Pulaski	52,210	.01
Rock Island	76,600	.02
St Clair	964,734	.26
Sangamon	58,500	.02
Tazewell	0	.00
Union	52,210	.01
Washington	0	.00
Will	229,900	.06
Winnebago	137,900	.04
Balance Of Illinois	1,379,359	.38
INDIANA	4,308,851	1.17
Adams	0	.00
Allen	0	.00
Blackford	0	.00
De Kalb	0	.00
Delaware	0	.00
Dubois	0	.00
Elkhart	467,200	.13
Gibson	0	.00
Lagrange	0	.00
Lake	500,300	.14
La Porte	0	.00
Madison	414,415	.11
Marion	1,525,872	.41
Noble	0	.00
Perry	0	.00
Pike	0	.00
Posey	0	.00
St Joseph	49,200	.01
Spencer	0	.00
Tippecanoe	0	.00
Vanderburgh	130,300	.04
Vigo	0	.00
Warrick	0	.00
Whitley	0	.00
Balance Of Indiana	1,221,564	.33
IOWA	577,200	.16
Black Hawk	0	.00
Boone	0	.00
Dallas	0	.00
Jasper	0	.00
Linn	0	.00
Madison	0	.00
Marion	0	.00
Polk	82,900	.02
Scott	0	.00
Story	0	.00
Warren	0	.00
Woodbury	0	.00
Balance Of Iowa	494,300	.13
KANSAS	355,800	.10
Butler	0	.00
Johnson	0	.00
Leavenworth	0	.00
Sedgwick	131,800	.04
Shawnee	134,800	.04
Wyandotte	89,200	.02
Balance Of Kansas	0	.00
KENTUCKY	3,726,153	1.01
Bourbon	0	.00
Clark	0	.00
Fayette	41,200	.01
Franklin	0	.00
Jefferson	317,400	.09
Jessamine	0	.00
Kenton	130,400	.04
Madison	0	.00

TABLE 3 - ALLOCATION OF TOTAL CETA TITLE II FUNDS FOR PUBLIC SERVICE EMPLOYMENT - FY 1974

U S DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

UNITED STATES BY COUNTY

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

STATE AND COUNTY	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
Rural Cep	1,000,296	.27
Scott	0	.00
Woodford	0	.00
Balance Of Kentucky	2,236,856	.61
LOUISIANA	7,723,932	2.10
Caddo	173,849	.05
Calcasieu	338,469	.09
Cameron	126,400	.03
East Baton Rouge	613,880	.17
Jefferson	276,700	.08
Jefferson Davis	126,400	.03
Lafayette	76,300	.02
Orleans	2,008,803	.55
Ouachita	0	.00
Rapides	67,300	.02
Balance Of Louisiana	3,915,830	1.06
MAINE	2,771,077	.75
Balance Of Maine	2,771,077	.75
MARYLAND	2,642,539	.72
Allegany	301,297	.08
Anne Arundel	0	.00
Baltimore	491,300	.13
Carroll	0	.00
Frederick	67,700	.02
Garrett	203,400	.06
Harford	27,100	.01
Howard	0	.00
Montgomery	0	.00
Prince Georges	398,376	.11
Washington	244,000	.07
Balance Of Maryland	909,364	.25
MASSACHUSETTS	19,965,140	5.43
Bristol	1,038,068	.28
Hampden	1,489,377	.40
Middlesex	2,691,985	.73
Plymouth	407,983	.11
Suffolk	3,234,603	.88
Worcester	253,938	.07
Balance Of Massachusetts	10,849,185	2.95
MICHIGAN	42,391,110	11.53
Allegan	253,326	.07
Bay	624,883	.17
Berrien	816,042	.22
Calhoun	509,450	.14
Clinton	145,050	.04
Eaton	145,050	.04
Genesee	4,237,783	1.15
Hillsdale	319,700	.09
Ingham	1,161,832	.32
Ionia	391,776	.11
Jackson	119,900	.03
Kalamazoo	344,762	.09
Kent	1,797,777	.49
Lapeer	597,099	.16
Lenawee	603,186	.16
Macomb	1,943,238	.53
Monroe	273,600	.07
Montcalm	391,776	.11
Muskegon	664,813	.18
Oakland	3,637,669	.99
Oceana	100,650	.03
Ottawa	449,670	.12
Saginaw	1,153,416	.31
St Clair	1,007,012	.27
Shiawassee	964,425	.26
Washtenaw	1,072,560	.29
Wayne	13,818,883	3.76
Balance Of Michigan	4,845,776	1.32
MINNESOTA	10,128,462	2.75
Aitkin	127,023	.03
Anoka	47,600	.01
Carlton	127,023	.03
Carver	0	.00
Cook	127,023	.03
Dakota	0	.00
Hennepin	1,248,400	.34
Itasca	127,023	.03
Koochiching	127,023	.03
Lake	127,023	.03
Ramsey	718,100	.20
Rural Cep	1,558,570	.42
St Louis	970,324	.26
Scott	0	.00
Washington	30,300	.01
Balance Of Minnesota	4,793,023	1.30
MISSISSIPPI	850,000	.23
Hinds	148,000	.04
Rankin	0	.00
Balance Of Mississippi	702,000	.19
MISSOURI	2,201,656	.60
Cass	0	.00
Clay	0	.00
Franklin	143,200	.04
Greene	0	.00
Jackson	0	.00
Jefferson	260,900	.07
Platte	0	.00
Ray	0	.00
St Louis	1,482,956	.40
Balance Of Missouri	314,600	.09
MONTANA	1,860,200	.51
Balance Of Montana	1,860,200	.51
NEBRASKA	459,800	.13
Douglas	459,800	.13
Lancaster	0	.00
Sarpy	0	.00
Balance Of Nebraska	0	.00
NEVADA	2,049,406	.56
Clark	1,135,347	.31
Washoe	440,300	.12
Balance Of Nevada	473,759	.13
NEW HAMPSHIRE	311,234	.08
Hillsborough	0	.00
Rockingham	0	.00
Strafford	0	.00
Balance Of New Hampshire	311,234	.08
NEW JERSEY	18,457,128	5.02
Atlantic	888,419	.24
Bergen	602,478	.16
Burlington	779,100	.21
Camden	1,356,700	.37
Cumberland	702,005	.19
Essex	2,992,739	.81
Gloucester	477,600	.13
Hudson	3,033,866	.82
Mercer	465,709	.13
Middlesex	1,217,800	.33
Monmouth	1,157,100	.31
Morris	169,900	.05
Ocean	777,124	.21
Passaic	2,295,540	.62
Somerset	0	.00
Union	521,900	.14
Balance Of New Jersey	1,019,144	.28
NEW MEXICO	1,858,594	.51
Bernalillo	528,114	.14
Balance Of New Mexico	1,330,479	.36
NEW YORK	37,571,226	10.22
Albany	138,800	.04
Allegany	126,000	.03
Broome	152,400	.04
Cattaraugus	126,000	.03
Chautauqua	126,000	.03
Chemung	223,300	.06
Dutchess	73,400	.02
Erie	4,635,282	1.26
Monroe	395,573	.11
Nassau	262,000	.07
New York	19,540,226	5.31
Niagara	1,250,575	.34
Oneida	621,494	.17
Onondaga	273,900	.07
Orange	541,800	.15
Oswego	326,605	.09
Putnam	51,050	.01
Rensselaer	141,400	.04
Rockland	96,200	.03
St Lawrence	416,117	.11
Saratoga	118,700	.03
Schenectady	125,100	.03
Suffolk	1,229,700	.33
Ulster	355,200	.10
Westchester	648,050	.18
Balance Of New York	5,576,351	1.52
NORTH CAROLINA	113,200	.03
Buncombe	0	.00
Chatham	0	.00
Cumberland	28,200	.01
Durham	0	.00
Forsyth	85,000	.02
Gaston	0	.00
Guilford	0	.00
Johnston	0	.00
Lee	0	.00
Mecklenburg	0	.00
Onslow	0	.00
Orange	0	.00
Wake	0	.00
Balance Of North Carolina	0	.00
NORTH DAKOTA	1,368,400	.37
Balance Of North Dakota	1,368,400	.37
OHIO	12,027,601	3.27
Ashtabula	0	.00
Butler	514,000	.14
Clark	209,820	.06
Columbiana	0	.00
Cuyahoga	33,321,148	9.06
Delaware	0	.00
Franklin	428,100	.12
Geauga	0	.00
Greene	31,800	.01
Hamilton	1,334,974	.36
Lake	0	.00
Licking	0	.00
Lorain	173,400	.05
Lucas	908,200	.25
Mahoning	476,300	.13
Medina	32,600	.01
Montgomery	886,157	.24
Portage	192,200	.05
Preble	0	.00
Stark	207,500	.06
Summit	268,500	.07
Trumbull	243,800	.07
Wayne	0	.00
Wood	58,200	.02
Balance Of Ohio	2,741,000	.75
OKLAHOMA	1,510,787	.41
Canadian	0	.00
Cleveland	0	.00
Comanche	83,000	.02
Creek	27,700	.01
Logan	0	.00
Oklahoma	257,533	.07
Osage	0	.00
Tulsa	195,500	.05
Balance Of Oklahoma	947,054	.26
OREGON,	4,675,185	1.27
Clackamas	482,698	.13
Lane	686,349	.19
Marion	326,675	.09
Multnomah	0	.00
Polk	467,422	.13
Washington	0	.00
Yamhill	167,748	.05
Balance Of Oregon	2,544,293	.69
PENNSYLVANIA	20,414,396	5.55
Allegheny	2,639,440	.72

TABLE 3 - ALLOCATION OF TOTAL CETA TITLE II FUNDS FOR PUBLIC SERVICE EMPLOYMENT - FY 1974

U S DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

UNITED STATES BY COUNTY

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

STATE AND COUNTY	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
Armstrong	125,024	.03
Beaver	194,400	.05
Bedford	292,334	.08
Berks	110,130	.03
Blair	159,840	.04
Bucks	420,636	.11
Butler	126,000	.03
Cambria	109,145	.03
Carbon	81,500	.02
Chester	0	.00
Clarion	95,900	.03
Clinton	384,432	.10
Crawford	210,300	.06
Cumberland	0	.00
Dauphin	41,200	.01
Delaware	505,666	.14
Erie	37,600	.01
Fayette	528,009	.14
Forest	0	.00
Franklin	0	.00
Fulton	292,334	.08
Huntingdon	292,334	.08
Indiana	71,300	.02
Lackawanna	772,497	.21
Lancaster	76,979	.02
Lawrence	347,046	.09
Lebanon	0	.00
Lehigh	0	.00
Luzerne	896,221	.24
Lycoming	111,277	.03
Mercer	365,847	.10
Montgomery	327,600	.09
Northampton	0	.00
Perry	0	.00
Philadelphia	6,327,164	1.72
Schuylkill	612,966	.17
Somerset	248,701	.07
Venango	0	.00
Warren	0	.00
Washington	337,365	.09
Westmoreland	1,183,483	.32
York	40,000	.01
Balance Of Pennsylvania	2,049,722	.56
RHODE ISLAND	2,688,691	.73
Providence	649,804	.18
Balance Of Rhode Island	2,038,887	.55
SOUTH CAROLINA	1,200,383	.33
Statewide [C]	1,200,383	.33
SOUTH DAKOTA	0	.00
Balance Of South Dakota	0	.00
TENNESSEE	2,060,732	.56
Davidson	67,200	.02
Hamilton	0	.00
Knox	33,700	.01
Shelby	625,500	.17
Sullivan	0	.00
Balance Of Tennessee	1,334,332	.36
TEXAS	6,424,732	1.75
Aransas	0	.00
Archer	0	.00
Armstrong	0	.00
Atascosa	0	.00
Bandera	0	.00
Bastrop	0	.00
Baylor	0	.00
Bee	0	.00
Bell	0	.00
Bexar	726,000	.20
Bosque	0	.00
Bowie	202,558	.06
Briscoe	0	.00
Brooks	0	.00
Burnet	0	.00
Caldwell	0	.00
Cameron	411,902	.11
Carson	0	.00
Castro	0	.00
Childress	0	.00
Clay	0	.00
Collingsworth	0	.00
Comal	0	.00
Coryell	0	.00
Cottle	0	.00
Dallas	47,600	.01
Deaf Smith	0	.00
Donley	0	.00
Duval	0	.00
El Paso	600,300	.16
Falls	0	.00
Fayette	0	.00
Foard	0	.00
Freestone	0	.00
Frio	0	.00
Galveston	0	.00
Gillespie	0	.00
Gray	0	.00
Guadalupe	0	.00
Hall	0	.00
Hamilton	0	.00
Hansford	0	.00
Hardeman	0	.00
Hardin	0	.00
Harris	1,146,500	.31
Hartley	0	.00
Hays	0	.00
Hemphill	0	.00
Hidalgo	813,172	.22
Hill	0	.00
Hutchinson	0	.00
Jack	0	.00
Jefferson	308,500	.08
Jim Wells	0	.00
Karnes	0	.00
Kendall	0	.00
Kenedy	0	.00
Kerr	0	.00
Kleberg	0	.00
Lampasas	0	.00
Lee	0	.00
Limestone	0	.00
Lipscomb	0	.00
Live Oak	0	.00
Llano	0	.00
Mclennan	63,500	.02
Mcmullen	0	.00
Medina	0	.00
Milam	0	.00
Mills	0	.00
Montague	0	.00
Nueces	384,000	.10
Oldham	0	.00
Orange	0	.00
Parmer	0	.00
Potter	0	.00
Randall	0	.00
Refugio	0	.00
Roberts	0	.00
San Patricio	0	.00
San Saba	0	.00
Swisher	0	.00
Tarrant	186,600	.05
Travis	0	.00
Webb	594,858	.16
Wheeler	0	.00
Wichita	0	.00
Wilbarger	0	.00
Willacy	25,200	.01
Williamson	0	.00
Wilson	0	.00
Young	0	.00
Balance Of Texas	914,039	.25
UTAH	1,854,100	.50
Statewide [C]	1,854,100	.50
VERMONT	1,552,279	.42
Balance Of Vermont	1,552,279	.42
VIRGINIA	3,329,672	.91
Alleghany	0	.00
Arlington	73,400	.02
Botetourt	0	.00
Chesterfield	0	.00
Craig	0	.00
Franklin	0	.00
Goochland	0	.00
Hanover	0	.00
Henrico	0	.00
Isle Of Wright	0	.00
James City	0	.00
New Kent	0	.00
Powhatan	0	.00
Prince William	0	.00
Roanoke	0	.00
Southampton	0	.00
York	0	.00
Alexandria [Ind City]	0	.00
Chesapeake [Ind City]	53,300	.01
Clifton Forge [Ind City]	0	.00
Covington [Ind City]	0	.00
Fairfax [Ind City]	0	.00
Hampton [Ind City]	91,500	.02
Newport News [Ind City]	228,004	.06
Norfolk [Ind City]	485,810	.13
Portsmouth [Ind City]	244,658	.07
Richmond [Ind City]	246,282	.07
Roanoke [Ind City]	0	.00
Salem [Ind City]	0	.00
Suffolk [Ind City]	126,900	.03
Virginia Beach [Ind City]	0	.00
Williamsburg [Ind City]	0	.00
Balance Of Virginia	1,779,814	.48
WASHINGTON	14,097,201	3.83
Clark	0	.00
King	3,791,618	1.03
Kitsap	421,756	.11
Pierce	1,562,304	.42
Snohomish	982,244	.27
Spokane	997,294	.27
Yakima	1,020,515	.28
Balance Of Washington	5,321,565	1.45
WEST VIRGINIA	3,261,649	.89
Statewide [C]	3,261,649	.89
WISCONSIN	6,926,912	1.88
Dane	0	.00
Fond Du Lac	0	.00
Kenosha	0	.00
Milwaukee	1,713,300	.47
Outagamie	0	.00
Ozaukee	0	.00
Racine	108,792	.03
Rock	555,300	.15
Walworth	0	.00
Washington	0	.00
Waukesha	0	.00
Winnebago	53,000	.01
Balance Of Wisconsin	4,496,520	1.22
WYOMING	0	.00
Balance Of Wyoming	0	.00
PUERTO RICO	16,376,891	4.45
Mayaguez Municipio	622,156	.17
Ponce Municipio	1,475,106	.40
San Juan Municipio	3,601,929	.98
Balance Of Puerto Rico	10,677,698	2.90
A.SAMOA-GUAM-TRUST TERRITORIES	345,300	.09
VIRGIN ISLANDS	246,700	.07
INDIAN RESERVATIONS	1,855,000	.50

TABLE 4 - ALLOCATION OF TOTAL CETA TITLE II FUNDS FOR PUBLIC SERVICE EMPLOYMENT - FY 1974

UNITED STATES BY STANDARD METROPOLITAN STATISTICAL AREA

U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

SMSA	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
UNITED STATES SMSA TOTAL	251,125,601	100.00
New York, Ny-Nj	20,938,004	8.34
Los Angeles-Long Beach,Ca.	20,086,364	8.00
Chicago,Ill.	5,321,300	2.12
Philadelphia,Pa-Nj.	10,194,466	4.06
Detroit, Mich.	20,976,401	8.35
San Francisco-Oakland, Ca.	11,198,577	4.46
Washington,Dc-Md-Va.	2,730,276	1.09
Boston,Mass	5,980,673	2.38
Nassau- Suffolk Ny.	1,491,700	.59
St.Louis Mo-Ill.	3,871,993	1.54
Pittsburg,Pa.	4,354,688	1.73
Dallas-Fort Worth,Tex.	234,200	.09
Baltimore, Md.	518,400	.21
Cleveland,Ohio	3,353,748	1.34
Newark, Nj.	3,684,539	1.47
Houston Tex.	1,146,500	.46
Minneapolis-St Paul,Minn-Wis.	2,044,400	.81
Atlanta Ga.	931,298	.37
Seattle Everett Wash.	4,773,862	1.90
Anaheim-St Ana-Garden Grove,Ca	1,769,800	.70
Milwaukee,Wis.	1,713,300	.68
Cincinnati, Ohio-Ky-Ind.	1,630,224	.65
San Diego, Ca.	6,829,161	2.72
Buffalo, Ny.	5,885,857	2.34
Kansas City,Mo-Kansas	89,200	.04
Miami,Fla.	772,173	.31
Denver-Boulder, Colo.	406,500	.16
Riverside-San Bern-Ontario,Ca.	3,730,288	1.49
Indianapolis,Ind.	1,525,872	.61
Tampa-St Petersburg,Fla.	431,100	.17
San Jose,Ca.	1,506,293	.60
New Orleans, La.	2,477,305	.99
Columbus, Ohio	475,900	.19
Postland, Oreg-Wash.	482,698	.19
Phoenix, Ariz.	763,288	.30
Rochester,Ny.	672,573	.27
Providence-Warwk-Pwtkt,Rl-Mass	649,804	.26
San Antonio,Tex.	726,000	.29
Louisville, Ky-Ind.	317,400	.13
Dayton,Ohio	934,307	.37
	0	.00
Memphis,Tenn-Ark-Miss.	625,500	.25
Sacramento,Ca.	2,470,451	.98
Albany-Schenectady-Troy,Ny.	716,154	.29
Birmingham,Ala.	451,671	.18
Toledo,Ohio-Mich.	1,240,000	.49
Greensboro-Win Sal-High Pt,Nc.	85,000	.03
Hartford,Conn.	1,044,600	.42
Salt Lake City-Ogden,Utah.	1,854,100	.74
Nashville-Davidson,Tenn.	67,200	.03
Oklahoma City, Okla.	257,533	.10
Norfolk-Va Beach-Prtsmth,Va-Nc	783,768	.31
Akron, Ohio	460,700	.18
Syracuse, Ny	713,705	.28
Gary-Hammond-East Chicago,Ind.	500,300	.20
Honolulu,Hawaii	2,079,382	.83
Northeast Pennsylvania	1,668,718	.66
Jacksonville, Fla.	245,000	.10
Fort Lauderdale-Hollywood,Fla.	478,436	.19
Jersey City, Nj.	3,033,866	1.21
Allentown-Bethlhm-Easton,Pa-Nj	132,000	.05
N Brunswick-Prth Am-Syrvlle,Nj	1,217,800	.48
Charlotte-Gastonia,Nc.	0	.00
Tulsa,Okla.	282,600	.11
Richmond,Va.	246,282	.10
Springfield-Chic-Hol,Mass-Conn	1,282,900	.51

SMSA	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
Omaha, Nebr-Iowa	989,400	.39
Grand Rapids,Mich.	2,247,447	.89
Youngstown-Warren,Ohio	720,100	.29
Flint,Mich.	5,202,208	2.07
Wilmington,Del- Nj Mo.	1,256,312	.50
Greenville-Spartanburg,Sc.	0	.00
Paterson-Clifton-Passaic,Nj.	2,295,540	.91
Long Branch-Asbury Park, Nj	1,157,100	.46
Orlando,Fla.	427,100	.17
Lansing-East Lansing,Mich.	1,843,708	.73
Raleigh-Durham,Nc.	0	.00
New Haven-West Haven,Conn.	685,317	.27
Fresno, Ca.	1,660,842	.66
Tacoma, Wash.	1,562,304	.62
Harrisburg Pa.	41,200	.02
Knoxville,Tenn.	66,600	.03
Bridgeport,Conn.	1,346,433	.54
Canton,Ohio	207,500	.08
Wichita,Kans.	131,800	.05
Mobile,Ala.	343,397	.14
Oxnard-Simi Valley-Ventura,Ca.	644,600	.26
Baton Rouge,La.	613,880	.24
Worcester,Mass.	253,938	.10
Chattanooga,Tenn-Ga.	29,100	.01
Davenport-Rock Isl-Mol,In-Ill.	76,600	.03
Fort Wayne,Ind.	0	.00
El Paso, Tex.	600,300	.24
Tucson, Ariz.	98,500	.04
West Palm Beach-Boca Raton,Fla	734,300	.29
Beaumont-Port Arthur-Orange,Tx	308,500	.12
Peoria,Ill.	118,200	.05
Utica-Rome,Ny.	740,394	.29
Charleston,Sc.	344,800	.14
Shreveport,La.	173,849	.07
Albuquerque,N.Mex.	528,114	.21
Newport News-Hampton,Va.	319,504	.13
York,Pa.	40,000	.02
Bakersfield, Ca.	594,905	.24
Little Rock-No Little Rock,Ark	0	.00
Austin,Tex.	0	.00
Columbia,Sc.	46,900	.02
Lancaster,Pa.	76,979	.03
Des Moines,Iowa	82,900	.03
Trenton,Nj.	465,709	.19
Binghamton Ny-Pa.	152,400	.06
Reading Pa.	110,130	.04
Madison,Wis.	0	.00
Stockton,Ca.	1,346,328	.54
Spokane,Wash.	997,294	.40
Huntington-Ashland,W.Va-Ky-Oh	424,581	.17
Evansville,Lnd- Ky.	170,500	.07
Corpus Christi,Tex.	384,000	.15
Huntsville,Ala	95,275	.04
South Bend- Ind.	49,200	.02
Appleton-Oshkosh,Wis.	53,000	.02
Augusta,Ga-Sc.	106,700	.04
Las Vegas Nev.	1,135,347	.45
Rockford,Ill.	137,900	.05
Lexington-Fayette,Ky	41,200	.02
Duluth-Superior,Minn-Wis.	970,324	.39
Sta Barbara-St Maria-Lompoc,Ca	277,600	.11
Erie,Pa.	37,600	.01
Johnstown Pa.	357,846	.14
Jackson Miss.	148,000	.06
Caurence-Haverhill,Mass-Nh.	275,106	.11
Kalamazoo-Portage,Mich.	344,762	.14
Charleston,W.Va.	291,819	.12
Lorain-Elyria,Ohio	173,400	.07
Vallejo-Fairfield-Napa,Ca.	379,400	.15
Shlinas-Seaside-Monterey,Ca.	724,434	.29
Pensacola, Fla.	102,300	.04
New London-Norwich Conn-Rl.	31,000	.01
Kingsport-Bristol,Tenn-Va.	0	.00
Colorado Springs,Colo.	107,300	.04
Columbus,Ga-Ala.	0	.00
Ann Arbor, Mich.	1,072,560	.43
Melbourne-Titusville-Cocoa,Fla	1,286,721	.51
Lakeland-Winter Haven,Fla.	890,300	.35

SMSA	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
Macon, Ga.	0	.00
Hamilton-Middletown,Ohio	514,000	.20
Montgomery,Ala.	0	.00
Poughkeepsie,Ny.	73,400	.03
Saginaw Mich.	1,153,416	.46
Lowell,Mass-Nh.	591,800	.24
Waterbury,Conn.	320,100	.13
Eugene-Springfield,Oreg.	686,349	.27
Fayetteville,Nc.	28,200	.01
Lima,Ohio	191,400	.08
Savannah,Ga.	0	.00
Stanford,Conn.	290,500	.12
Santa Rosa,Ca.	1,044,513	.42
Roanoke,Va.	0	.00
Modesto,Ca.	1,704,636	.68
Springfield,Ohio	209,820	.08
Salem,Oreg.	794,097	.32
Wheeling,W.Va-Ohio	37,533	.01
Mcallen Pharr-Edinburg,Tex.	813,172	.32
Topeka,Kans.	134,800	.05
Battle Creek,Mich.	509,450	.20
Lubbock,Tex.	27,200	.01
Muskegon-Muskegon Heights,Mich	765,463	.30
Terre Haute,Ind.	43,900	.02
Atlantic City,Nj.	888,419	.35
Springfield,Ill.	58,500	.02
Racine,Wis.	108,792	.04
Portland,Maine	57,500	.02
Galveston-Texas City,Tex.	0	.00
Fall River,Mass-Rl.	343,700	.14
Daytona Beach,Fla.	408,700	.16
Springfield,Mo.	0	.00
Lincoln,Nebr.	0	.00
Steubenville-Weirton,Ohio-W,Va	68,000	.03
Champaign-Urbana-Rantoul,Ill	0	.00
Cedar Rapids,Iowa	0	.00
New Bedford,Mass	549,400	.22
Asheville,Nc.	0	.00
Fort Smith,Ark-Okla.	0	.00
Biloxi-Gulfport,Miss	108,100	.04
Killeen-Temple,Tex.	0	.00
Green Bay,Wis.	143,705	.06
Brockton Mass.	407,983	.16
Parkersburg-Marietta,W.Va-Ohio	37,533	.01
Waco,Tex.	63,500	.03
Lake Charles,La.	338,469	.13
New Britain,Conn.	269,112	.11
Yakima,Wash.	1,020,515	.41
Amarillo Tex	0	.00
Jackson,Mich.	119,900	.05
Bronnsville-Harl-San Benito,Tx	411,902	.16
Anderson,Ind	414,415	.17
Provo-Orem,Utah.	0	.00
Altoona, Pa.	159,840	.06
St.Cloud,Minn.	0	.00
Lyncaburg Va.	0	.00
Waterloo-Cedar Falls,Iowa	0	.00
Manchester,Nh.	0	.00
Alexandria,La	67,300	.03
Mansfield,Ohio	279,350	.11
Wichita Falls,Tex.	0	.00
Muncie,Ind	0	.00
Petersburg-Col Hgts-Hpwell,Va	0	.00
Fayetteville-Springdale,Ark.	0	.00
Norwalk,Conn	127,000	.05
Norwalk,Conn	127,000	.05
Decatur,Ill.	61,400	.02
Anchorage,Alaska	772,651	.31
Santa Cruz,Calif.	741,589	.30
Abilene Tex	0	.00
Vineland-Millvile-Bridgeton,Nj	702,005	.28
Reno,Nev	440,300	.18
Sarasota,Fla.	52,600	.02
Fargo-Moorhead,N Dak-Minn.	0	.00
Clarksville-Hopkinsville,Tn-Ky	0	.00
Pueblo,Colo.	140,100	.06
Kenosha, Wis.	0	.00

MEDIAN ALLOCATION BY SMSA $ 246,300

MEAN ALLOCATION BY SMSA $ 930,100

TABLE 4 - ALLOCATION OF TOTAL CETA TITLE II FUNDS FOR PUBLIC SERVICE EMPLOYMENT - FY 1974

U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

UNITED STATES BY STANDARD METROPOLITAN STATISTICAL AREA

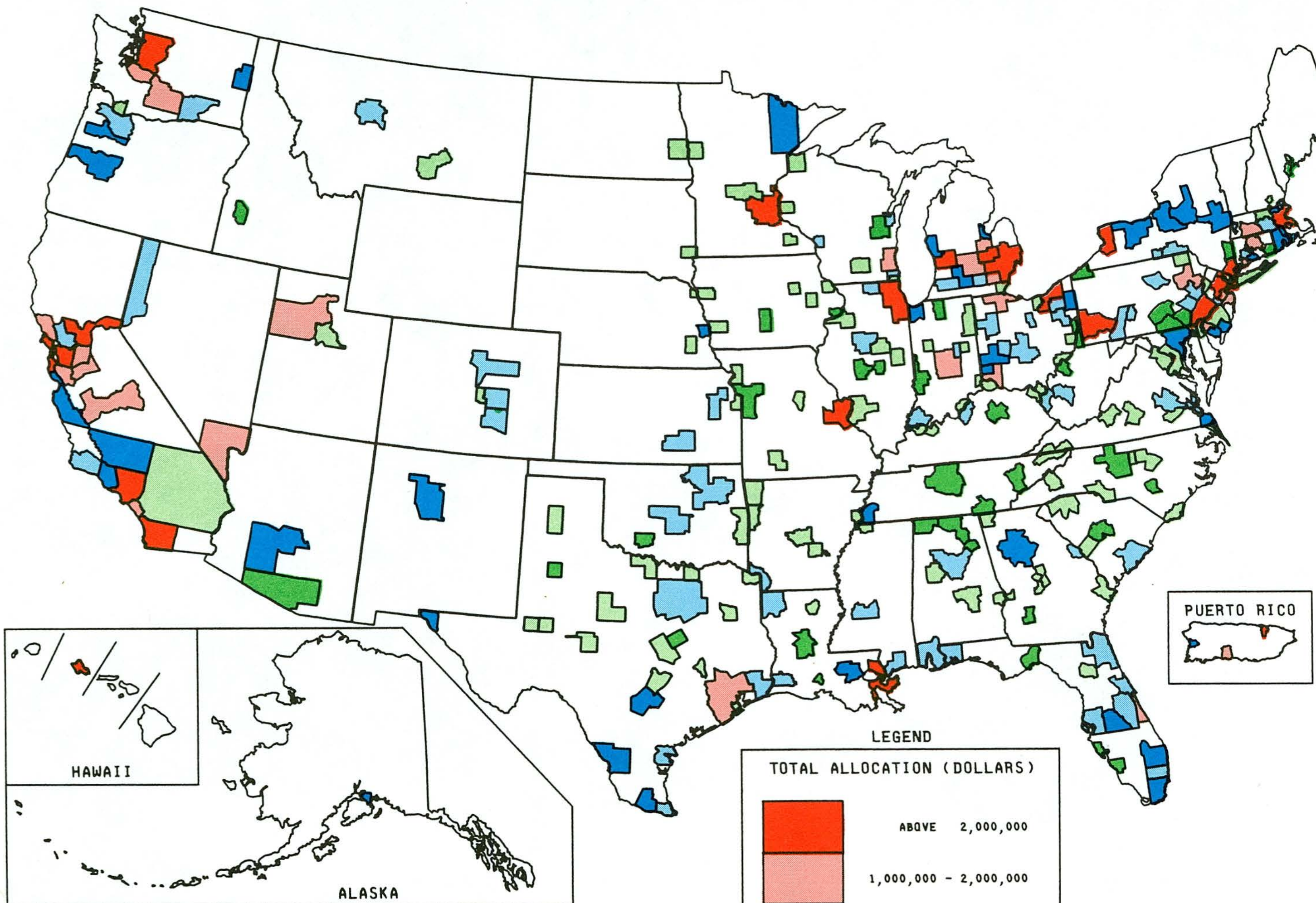
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

SMSA	TOTAL ALLOCATION (DOLLARS)	PERCENT OF NATION
Florence,Ala	40,400	.02
Bay City,Mich.	624,883	.25
Sioux City,Iowa-Nebr.	0	.00
Tuscaloosa,Ala	0	.00
Danbury,Conn	149,600	.06
Monroe,La	0	.00
Williamsport,Pa	111,277	.04
Texarkana,Tex-Texarkana,Ark	321,358	.13
Boise City,Idaho	48,100	.02
Lafayette,La	76,300	.03
Lafayette-West Lafayette,In	0	.00
Tallahassee,Fla.	35,300	.01
Lawton,Okla.	83,000	.03
Wilmington,Nc	0	.00
Fort Myers,Fla.	56,900	.02
Gainesville Fla.	0	.00
Bloomington-Normal,Ill.	0	.00
Anniston,Ala	0	.00
Elmira Ny.	223,300	.09
St.Joseph.Mo.	0	.00
Fitchburg-Leominster,Mass.	0	.00
Tyler,Tex.	0	.00
Pittsfield,Mass.	188,700	.08
Albany,Ga	90,000	.04
Burlington Nc.	0	.00
Sioux Falls,S.Dak.	0	.00
Gadsden,Ala.	75,216	.03
Richland-Kennewick,Wash.	275,900	.11
Odessa,Tex.	0	.00
Dubuque,Iowa	0	.00
Billings,Mont.	0	.00
Nashua,Nh.	0	.00
Pine Bluff,Ark	0	.00
Rochester,Minn.	0	.00
Sherman-Denison Tex	0	.00
Great Falls, Mont.	181,200	.07
Columbia Mo.	0	.00
La Crosse Wis.	150,100	.06
Owensboro Ky.	0	.00
Laredo,Tex	594,858	.24
Lewiston-Auburn,Maine	0	.00
San Angelo Tex.	0	.00
Bristol Conn.	198,500	.08
Midland,Tex	0	.00
Bryan-College Station,Tex.	0	.00
Meriden, Conn.	313,400	.12
PUERTO RICO		
San Juan	3,601,929	1.43
Ponce	1,475,106	.59
Caguas	856,384	.34
Mayaguez	622,156	.25

MEDIAN ALLOCATION BY SMSA $ 246,300

MEAN ALLOCATION BY SMSA $ 930,100

MAP 4 - Allocation of Total CETA Title II Funds For Public Service Employment - FY 1974 United States by Standard Metropolitan Statistical Area



LEGEND

TOTAL ALLOCATION (DOLLARS)

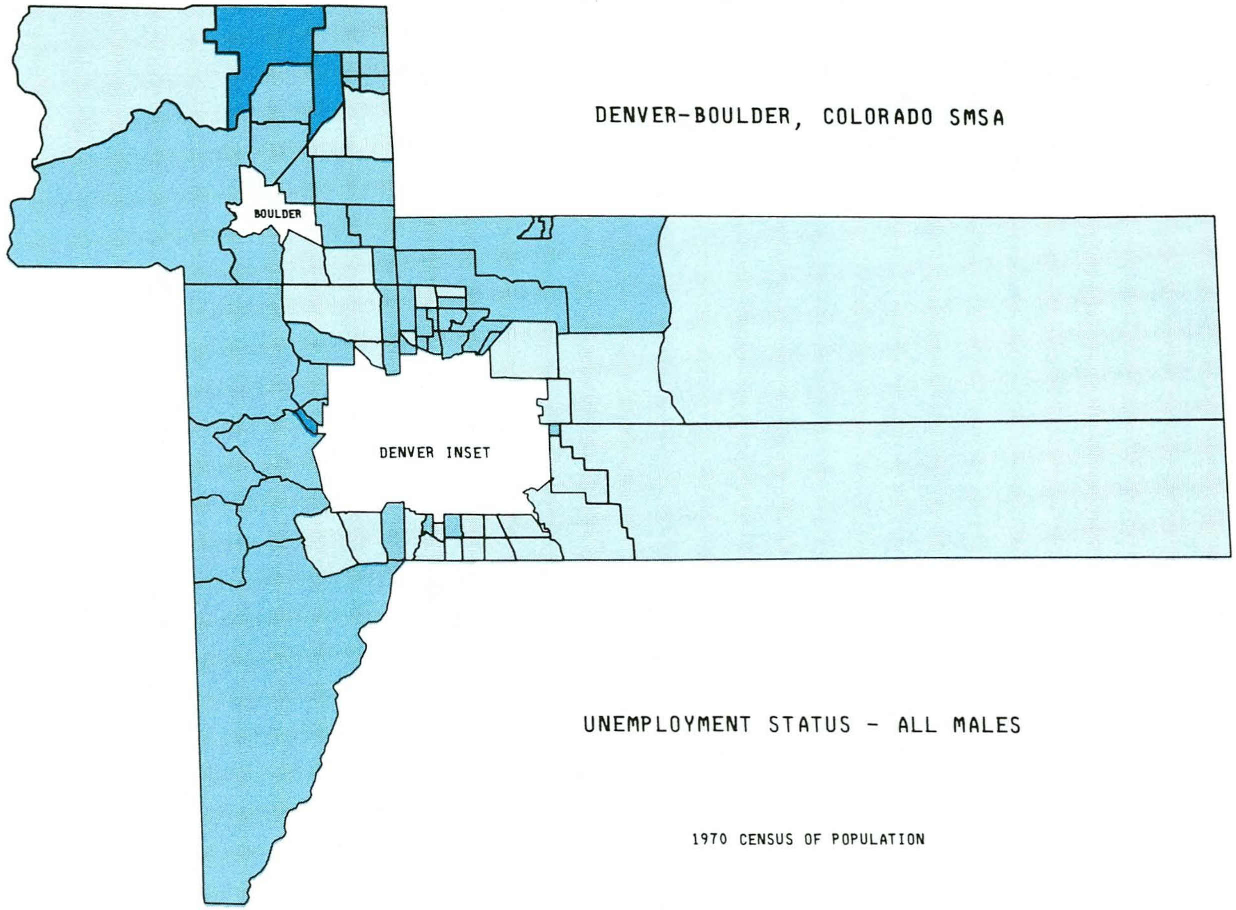
(red)	ABOVE 2,000,000
(pink)	1,000,000 - 2,000,000
(dark blue)	500,000 - 1,000,000
(light blue)	100,000 - 500,000
(dark green)	1 - 100,000
(light green)	0

MEDIAN SMSA ALLOCATION
$ 246,300

U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

DENVER-BOULDER, COLORADO SMSA



UNEMPLOYMENT STATUS - ALL MALES

1970 CENSUS OF POPULATION

LEGEND

PERCENT OF TRACT

DATA NOT AVAILABLE | BELOW 2 | 2 - 5 | 6 - 12 | 13 - 20 | 21 - 30 | ABOVE 30

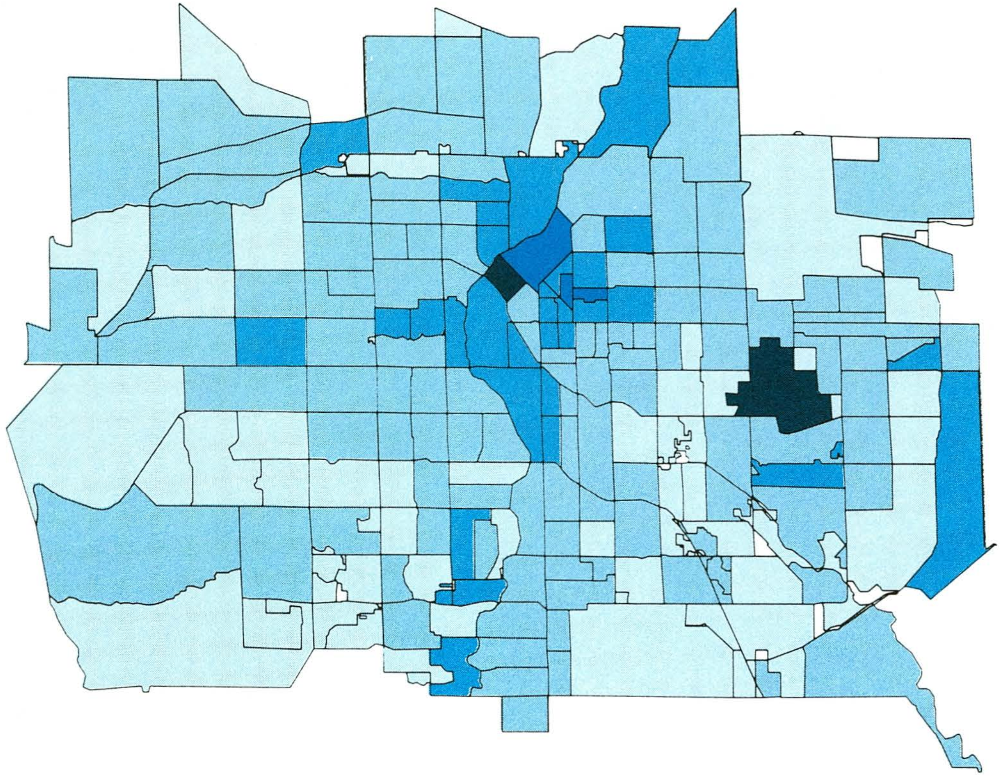
U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

UNEMPLOYMENT STATUS - ALL MALES

COUNTY AND CENSUS TRACT	CIVILIAN LABOR FORCE	NOT EMPLOYED	PERCENT OF TRACT	PERCENT OF SMSA
DENVER	310,147	11,153	3.6	100.0
ADAMS	45,342	1,452	3.2	13.0
7800	1,098	50	4.6	.4
7900	1,150	57	5.0	.5
8000	1,409	46	3.3	.4
8100	552	5	.9	.0
8200	1,359	34	2.5	.3
8302	408	5	1.2	.0
8303	148	0	.0	.0
8400	518	10	1.9	.1
8501	2,570	88	3.4	.8
8502	2,214	41	1.9	.4
8503	653	25	3.8	.2
8504	733	18	2.5	.2
8601	773	31	4.0	.3
8602	1,359	64	4.7	.6
8701	57	0	.0	.0
8702	2,672	162	6.1	1.5
8703	1,268	48	3.8	.4
8801	1,079	60	5.6	.5
8802	1,175	57	4.9	.5
8901	729	59	8.1	.5
8952	428	4	.9	.0
9000	1,981	42	2.1	.4
9100	1,614	43	2.7	.4
9200	1,343	34	2.5	.3
9301	2,531	89	3.5	.8
9302	1,963	55	2.8	.5
9303	729	16	2.2	.1
9304	987	46	4.7	.4
9305	1,450	13	.9	.1
9401	1,423	19	1.3	.2
9402	1,057	22	2.1	.2
9501	1,101	10	.9	.1
9502	1,201	37	3.1	.3
9553	911	20	2.2	.2
9601	2,169	43	2.0	.4
9602	1,334	62	4.6	.6
9750	1,196	37	3.1	.3
ARAPAHOE	40,714	1,078	2.6	9.7
4052	55	0	.0	.0
4452	0	0	.0	.0
4852	52	4	7.7	.0
4950	322	3	.9	.0
5250	41	0	.0	.0
5350	130	0	.0	.0
5403	197	6	3.0	.1
5551	410	29	7.1	.3
5552	631	8	1.3	.1
5553	870	56	6.4	.5
5602	1,097	21	1.9	.2
5603	1,315	5	.4	.0
5604	2,671	88	3.3	.8
5605	1,633	0	.0	.0
5606	988	12	1.2	.1
5651	1,862	36	1.9	.3
5700	911	49	5.4	.4
5800	989	29	2.9	.3
5900	1,527	63	4.1	.6
6000	984	33	3.4	.3
6100	843	44	5.2	.4
6200	1,146	26	2.3	.2
6300	1,040	41	3.9	.4
6400	1,010	34	3.4	.3
6500	1,566	64	4.1	.6
6601	1,119	30	2.7	.3
6602	2,070	37	1.8	.3
6702	392	0	.0	.0
6703	463	6	1.3	.1
6751	1,145	12	1.0	.1
6805	126	0	.0	.0
6806	5	0	.0	.0
6851	*	*	*	*
6852	31	0	.0	.0
6854	306	10	3.3	.1
6951	441	10	2.3	.1
6952	347	13	3.7	.1
7003	551	15	2.7	.1
7004	504	12	2.4	.1
7005	1,396	10	.7	.1
7007	742	19	2.6	.2
7008	200	5	2.5	.0
7009	16	0	.0	.0
7010	50	3	6.0	.0
7011	171	0	.0	.0
7051	41	5	12.2	.0
7052	280	6	2.1	.1
7056	375	5	1.3	.0
7100	527	10	1.9	.1
7200	1,629	45	2.8	.4
7300	1,493	51	3.4	.5
7400	1,349	29	2.1	.3
7500	612	13	2.1	.1
7600	613	46	7.5	.4
7701	626	22	3.5	.2
7702	804	13	1.6	.1
BOULDER	34,456	1,363	4.0	12.2
12101	1,300	23	1.8	.2
12102	1,670	68	4.1	.6
12103	815	13	1.6	.1
12104	121	0	.0	.0
12201	1,357	95	7.0	.9
12202	1,082	90	8.3	.8
12203	616	26	4.2	.2
12300	1,012	68	6.7	.6
12401	1,377	58	4.2	.5
12402	536	34	6.3	.3
12501	331	17	5.1	.2
12502	794	27	3.4	.2
12503	341	31	9.1	.3
12504	1,935	66	3.4	.6
12505	1,217	58	4.8	.5
12506	1,695	39	2.3	.3
12601	2,173	91	4.2	.8
12602	155	4	2.6	.0
12701	262	13	5.0	.1
12702	773	19	2.5	.2
12703	570	0	.0	.0
12704	103	0	.0	.0
12800	577	18	3.1	.2
12900	1,121	44	3.9	.4
13000	957	26	2.7	.2
13101	2,252	46	2.0	.4
13102	35	0	.0	.0
13201	254	6	2.4	.1
13202	202	10	5.0	.1
13203	548	49	8.9	.4
13204	773	10	1.3	.1
13205	327	0	.0	.0
13301	1,700	64	3.8	.6
13302	1,590	63	4.0	.6
13400	794	46	5.8	.4
13500	1,336	47	3.5	.4
13601	557	44	7.9	.4
13602	76	0	.0	.0
13700	1,122	50	4.5	.4
DENVER	127,270	5,583	4.4	50.1
101	921	17	1.8	.2
102	1,000	21	2.1	.2
201	1,145	25	2.2	.2
202	1,115	94	8.4	.8
301	1,574	43	2.7	.4
302	1,134	47	4.1	.4
303	1,428	74	5.2	.7
401	979	41	4.2	.4
402	1,904	74	3.9	.7
500	2,029	52	2.6	.5
600	840	48	5.7	.4
701	970	85	8.8	.8
702	1,420	112	7.9	1.0
800	343	32	9.3	.3
901	2,166	68	3.1	.6
902	1,569	47	3.0	.4
903	1,321	53	4.0	.5
1000	1,011	38	3.8	.3
1101	722	75	10.4	.7
1102	844	59	7.0	.5
1301	1,206	61	5.1	.5
1302	1,062	28	2.6	.3
1401	1,497	23	1.5	.2
1402	1,107	38	3.4	.3
1403	871	45	5.2	.4
1500	800	95	11.9	.9
1600	834	123	14.7	1.1
1701	298	89	29.9	.8
1702	1,006	59	5.9	.5
1800	557	44	7.9	.4
1900	864	74	8.6	.7
2000	377	20	5.3	.2
2100	1,625	117	7.2	1.0
2300	1,394	115	8.2	1.0
2401	905	111	12.3	1.0
2402	465	90	19.4	.8
2500	361	28	7.8	.3
2601	1,071	75	7.0	.7
2602	757	71	9.4	.6
2701	1,742	124	7.1	1.1
2702	1,723	100	5.8	.9
2703	1,975	123	6.2	1.1
2801	1,027	52	5.1	.5
2802	1,564	122	7.8	1.1
2803	1,335	59	4.4	.5
2901	1,041	65	6.2	.6
2902	1,256	38	3.0	.3
3001	1,814	76	4.2	.7
3002	893	49	5.5	.4
3003	1,007	44	4.4	.4
3004	1,381	14	1.0	.1
3005	478	10	2.1	.1
3101	609	86	14.1	.8
3102	1,245	106	8.5	1.0
3201	1,662	82	4.9	.7
3202	874	47	5.4	.4
3203	865	36	4.2	.3
3300	1,005	40	4.0	.4
3400	1,977	57	2.9	.5
3500	1,275	72	5.6	.6
3601	1,080	64	5.9	.6
3602	1,112	105	9.4	.9
3603	1,073	42	3.9	.4
3701	684	55	8.0	.5
3702	1,459	62	4.2	.6
3703	960	37	3.9	.3
3800	1,110	50	4.5	.4
3901	1,125	13	1.2	.1
3902	1,261	61	4.8	.5
4001	2,161	61	2.8	.5
4002	1,084	15	1.4	.1
4003	1,723	52	3.0	.5
4004	732	9	1.2	.1
4101	817	22	2.7	.2
4102	1,187	25	2.1	.2
4103	1,698	62	3.7	.6
4104	1,380	41	3.0	.4
4105	19	0	.0	.0
4201	1,388	57	4.1	.5
4202	1,126	64	5.7	.6
4301	1,386	38	2.7	.3
4302	738	10	1.4	.1
4303	1,253	9	.7	.1
4304	1,576	34	2.2	.3
4305	1,150	10	.9	.1
4401	1,828	78	4.3	.7
4402	57	12	21.1	.1
4501	1,671	80	4.8	.7
4502	1,474	77	5.2	.7
4601	1,808	34	1.9	.3
4602	1,416	43	3.0	.4
4603	1,825	73	4.0	.7
4700	1,493	15	1.0	.1
4801	1,116	31	2.8	.3
4802	1,019	16	1.6	.1
4900	489	5	1.0	.0
5000	1,640	50	3.0	.4
5101	1,275	24	1.9	.2
5102	1,462	21	1.4	.2
5200	1,170	29	2.5	.3
5300	421	8	1.9	.1
5401	18	0	.0	.0
5402	670	42	6.3	.4
5501	*	*	*	*
5502	149	5	3.4	.0
5503	505	9	1.8	.1
5601	281	0	.0	.0
6701	33	0	.0	.0
6801	1,010	3	.3	.0
6802	682	21	3.1	.2
6803	1,461	28	1.9	.3
6804	36	0	.0	.0
6901	638	5	.8	.0
6902	125	4	3.2	.0
7001	624	14	2.2	.1
7002	22	0	.0	.0
8301	1,232	36	2.9	.3
9700	0	0	.0	.0
10601	348	0	.0	.0
11500	24	0	.0	.0
11901	204	0	.0	.0
11902	930	4	.4	.0
11903	431	5	1.2	.0
12001	156	0	.0	.0
JEFFERSON	62,365	1,677	2.7	15.0
9801	731	20	2.7	.2
9802	355	6	1.7	.1
9803	657	13	2.0	.1
9804	881	21	2.4	.2
9805	572	15	2.6	.1
9806	1,139	19	1.7	.2
9807	513	22	4.3	.2
9808	572	22	3.8	.2
9809	776	40	5.2	.4
9810	1,050	30	2.9	.3
9900	456	19	4.2	.2
10000	960	65	6.8	.6
10100	998	51	5.1	.5
10201	3,192	66	2.1	.6
10202	1,674	25	1.5	.2
10301	2,836	51	1.8	.5
10302	2,476	61	2.5	.5
10402	1,186	30	2.5	.3
10403	1,148	36	3.1	.3
10451	935	61	6.5	.5
10501	1,971	42	2.1	.4
10502	1,656	13	.8	.1
10602	2,116	69	3.3	.6
10651	15	0	.0	.0
10700	2,119	65	3.1	.6
10800	1,781	33	1.9	.3
10900	1,885	48	2.5	.4
11000	1,742	36	2.1	.3
11100	1,612	98	6.1	.9
11200	1,691	60	3.5	.5
11300	1,073	56	5.2	.5
11400	1,370	36	2.6	.3
11550	1,743	70	4.0	.6
11600	2,037	50	2.5	.4
11701	860	10	1.2	.1
11702	1,665	10	.6	.1
11703	1,564	19	1.2	.2
11704	1,605	26	1.6	.2
11705	1,060	15	1.4	.1
11706	239	5	2.1	.0
11707	1,132	34	3.0	.3
11801	1,055	37	3.5	.3
11802	2,344	39	1.7	.3
11951	301	12	4.0	.1
11952	*	*	*	*
11953	66	0	.0	.0
12002	1,590	33	2.1	.3
12003	16	0	.0	.0
12004	290	0	.0	.0
12005	174	2	1.1	.0
12006	1,024	46	4.5	.4
12007	131	0	.0	.0
12008	613	13	2.1	.1
12009	695	27	3.9	.2
12051	19	0	.0	.0

DENVER AREA INSET



UNEMPLOYMENT STATUS - ALL MALES

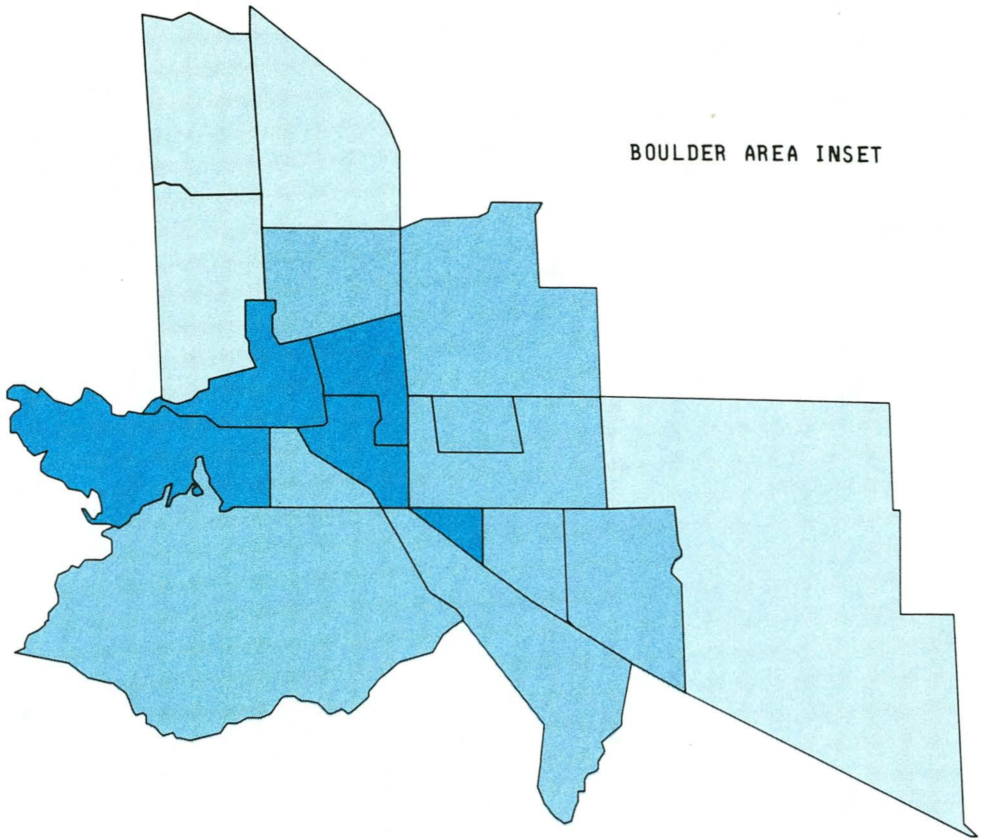
1970 CENSUS OF POPULATION

LEGEND	PERCENT OF TRACT					
DATA NOT AVAILABLE	BELOW 2	2 - 5	6 - 12	13 - 20	21 - 30	ABOVE 30

U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

BOULDER AREA INSET



UNEMPLOYMENT STATUS - ALL MALES

1970 CENSUS OF POPULATION

LEGEND

PERCENT OF TRACT

DATA NOT AVAILABLE	BELOW 2	2 - 5	6 - 12	13 - 20	21 - 30	ABOVE 30

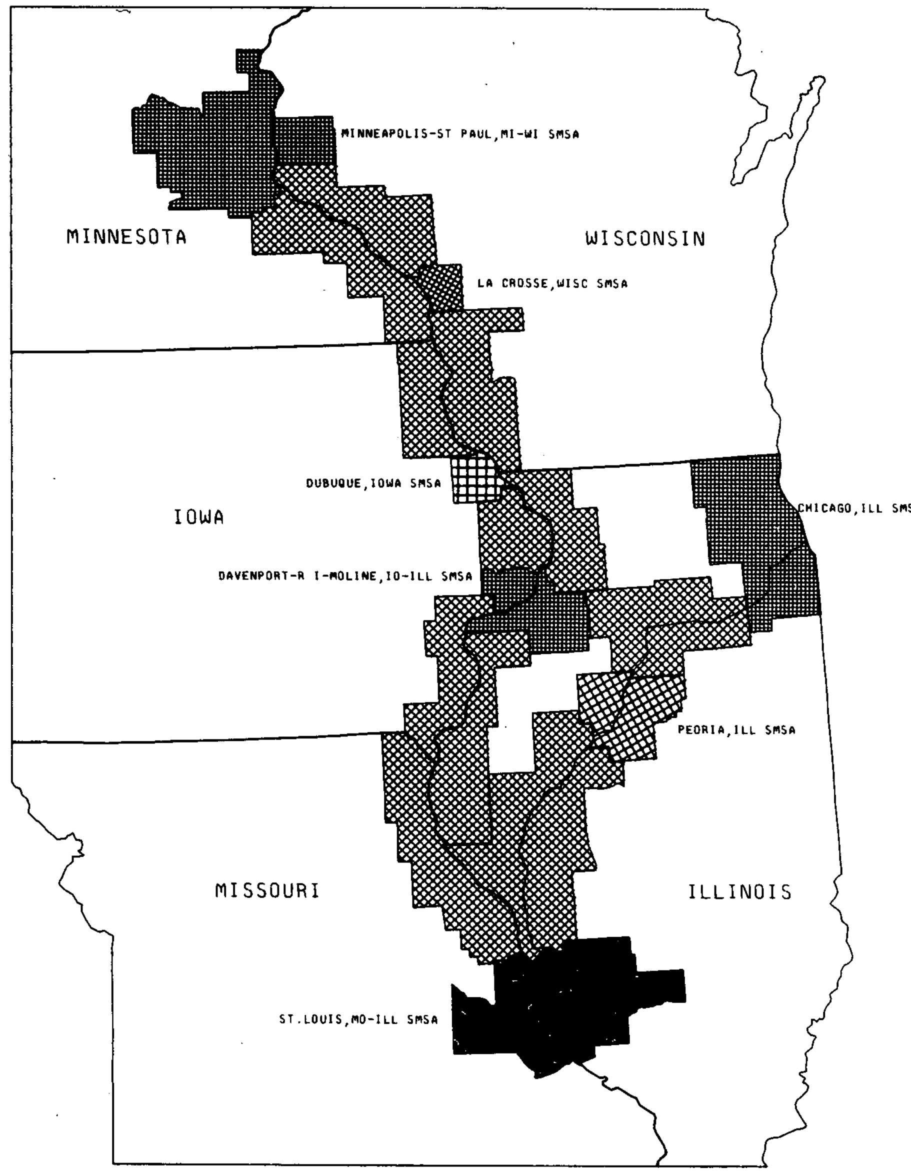
MAP 83 -- EMPLOYMENT IN THE PUBLIC ADMINISTRATION INDUSTRY

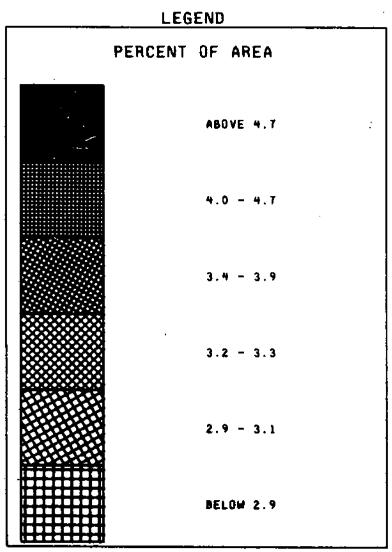
SOCIO-ECONOMIC STUDY

LOCK AND DAM 26

UPPER MISSISSIPPI RIVER
AND
ILLINOIS RIVER

1970 CENSUS OF POPULATION





ST.LOUIS, MISSOURI DISTRICT
U.S. ARMY CORPS OF ENGINEERS

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

TABLE 83 -- EMPLOYMENT IN THE PUBLIC ADMINISTRATION INDUSTRY

RUN DATE 01/10/75

SOCIO-ECONOMIC STUDY, LOCK AND DAM 26

UPPER MISSISSIPPI - ILLINOIS RIVERS

ST.LOUIS, MISSOURI DISTRICT
U.S. ARMY CORPS OF ENGINEERS

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

SMSA AND NON-SMSA COUNTY AGGREGATION AREAS	TOTAL EMPLOYED ALL INDUS.	PUBLIC ADMIN. EMPLOYMENT	PERCENT OF AREA	PERCENT OF REGION
REGION TOTAL	5,382,425	238,396	4.4	100.0
Illinois River	138,278	4,628	3.3	1.9
Chicago,Ill Smsa	2,852,017	126,867	4.4	53.2
Peoria,Ill Smsa	134,501	3,954	2.9	1.7
St.Louis,Mo-Il Smsa	914,474	51,355	5.6	21.5
Davenport-R.Island,Io-Il Smsa	139,926	6,570	4.7	2.8
Dubuque,Io Smsa	33,408	805	2.4	.3
La Crosse,Wi Smsa	30,005	1,028	3.4	.4
Minneapolis-St.Paul,Mn-Wi Smsa	815,273	32,685	4.0	13.7
Upper Mississippi River	324,543	10,504	3.2	4.4

MAP 13 - Allocation of Total CETA Title II Funds For Public Service Employment - FY 1974 Region IX by Prime Sponsor

ARIZONA - CALIFORNIA - NEVADA -
HAWAII - AMERICAN SAMOA - GUAM -
TRUST TERRITORIES OF THE PACIFIC ISLANDS
FEDERAL REGION IX

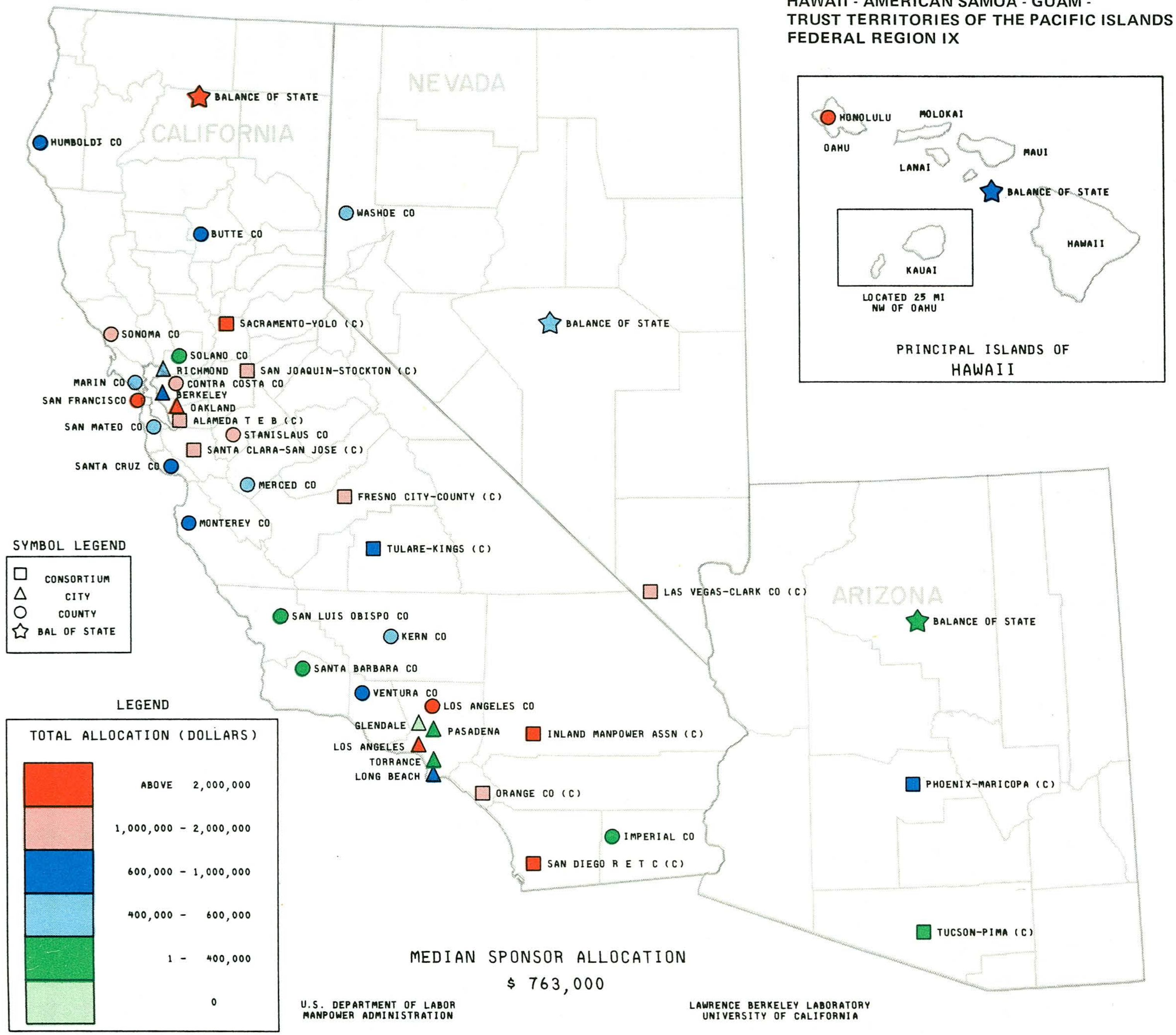


TABLE 13 - ALLOCATION OF TOTAL CETA TITLE II FUNDS FOR PUBLIC SERVICE EMPLOYMENT - FY 1974

RUN DATE 12/13/74

U.S. DEPARTMENT OF LABOR
MANPOWER ADMINISTRATION

REGION IX BY PRIME SPONSOR

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

PRIME SPONSOR	TOTAL ALLOCATION (DOLLARS)	PERCENT OF REGION
REGION IX	70,984,153	100.00
ARIZONA	1,002,288	1.41
Phoenix-Maricopa [C]	763,288	1.08
Tucson-Pima [C]	98,500	.14
Balance Of State	140,500	.20
CALIFORNIA	64,769,428	91.24
Alameda T E B [C]	1,752,888	2.47
Fresno City-County [C]	1,660,842	2.34
Inland Manpower Assn [C]	3,730,289	5.26
Orange Co [C]	1,769,800	2.49
Sacramento-Yolo [C]	2,224,452	3.13
San Diego R E T C [C]	6,829,161	9.62
San Joaquin-Stockton [C]	1,346,328	1.90
Santa Clara-San Jose [C]	1,506,293	2.12
Tulare-Kings [C]	792,683	1.12
Berkeley	891,643	1.26
Butte Co	680,703	.96
Contra Costa Co	1,332,931	1.88
Glendale	0	.00
Humboldt Co	730,005	1.03
Imperial Co	388,857	.55
Kern Co	594,905	.84
Long Beach	974,187	1.37
Los Angeles	10,324,021	14.54
Los Angeles Co	8,549,332	12.04
Marin Co	485,600	.68
Merced Co	512,005	.72
Monterey Co	724,434	1.02
Oakland	2,233,139	3.15
Pasadena	180,824	.25
Richmond	429,230	.60
San Francisco	3,587,946	5.05
San Luis Obispo Co	29,200	.04
San Mateo Co	485,200	.68
Santa Barbara Co	277,600	.39
Santa Cruz Co	741,589	1.04
Solano Co	338,600	.48
Sonoma Co	1,044,513	1.47
Stanislaus Co	1,704,636	2.40
Torrance	58,000	.08
Ventura Co	644,600	.91
Balance Of State	5,212,978	7.34
HAWAII	2,817,732	3.97
Honolulu	2,079,382	2.93
Balance Of State	738,350	1.04
NEVADA	2,049,406	2.89
Las Vegas-Clark Co [C]	1,135,347	1.60
Washoe Co	440,300	.62
Balance Of State	473,759	.67
A.SAMOA-GUAM-TRUST TERRITORIES	345,300	.49

MEDIAN ALLOCATION BY PRIME SPONSOR $ 763,000

MEAN ALLOCATION BY PRIME SPONSOR $ 1,571,000