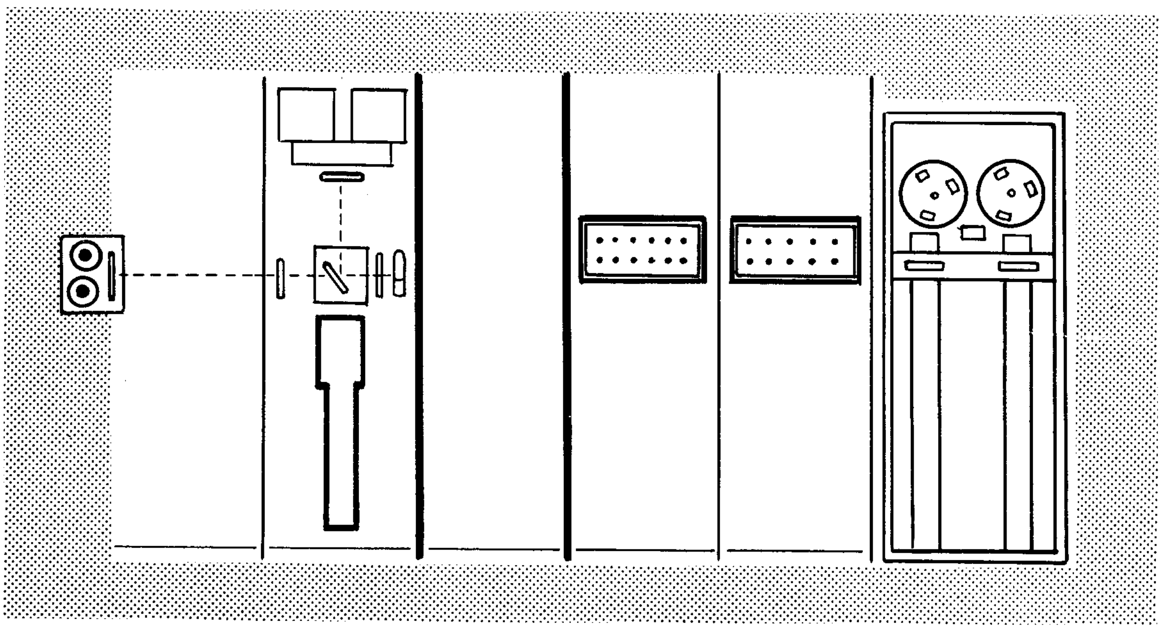
A Computer-Oriented System for Assembling and Displaying Land Management Information

Elliot L. Amidon



U S. FOREST SERVICE RESEARCH PAPER PSW- 17

Pacific Southwest
Forest and Range Experiment Station
Berkeley, California
Forest Service - U. S. Department of Agriculture

1964

Amidon, Elliot L.

1964. **A computer-oriented system for assembling and displaying land management information.** Berkeley, Calif., Pacific SW. Forest & Range Expt. Sta. 34 pp., illus. (U.S. Forest Serv. Res. Paper PSW-17)

Maps contain information basic to land management planning. By transforming conventional map symbols into numbers which are punched into cards, the land manager can have a computer assemble and display information required for a specific job. He can let a computer select information from several maps, combine it with such nonmap data as treatment cost or benefit per acre, and prepare overlays and tabulations. The report describes specific and potential uses for the system and includes cost estimates.

582:U518;2

Amidon, Elliot L.

1964. **A computer-oriented system for assembling and displaying land management information.** Berkeley, Calif., Pacific SW. Forest & Range Expt. Sta. 34 pp., illus. (U.S. Forest Serv. Res. Paper PSW-17)

Maps contain information basic to land management planning. By transforming conventional map symbols into numbers which are punched into cards, the land manager can have a computer assemble and display information required for a specific job. He can let a computer select information from several maps, combine it with such nonmap data as treatment cost or benefit per acre, and prepare overlays and tabulations. The report describes specific and potential uses for the system and includes cost estimates.

582:U518;2

Glossary

Block: a map or portion thereof; a unit for which tables will be produced; block contains one or more strips.
Card: an 80-column punch card (record).
Cell: the map area assigned a code number (analogous to one dot in a dot grid); cell is a rectangle one-fifth (measured horizontally) by one-sixth inch, or one-thirtieth square inch.
Code: unsigned, nonzero, two-digit integer (i.e., a number between 01 and 99 inclusive).
Code system: a list of up to. 98 code numbers and their definitions. Code 99, used for boundaries and other special purposes discussed elsewhere in this report, is ignored in all computations.
Line: a horizontal series of cells containing codes or blanks which are punched into one card.
MIADS: Map Information Assembly and Display System.
Source map: a map, photo mosaic, or similar data source.
Strip: rows of lines made by listing ("printing") cards. Strip width is fixed at 7.2 inches (owing to use of 36 x 2 = 72 card columns). Strips are fastened together on their vertical edges to make an overlay.

CONTENTS

The Author

ELLIOT L. AMIDON is conducting forest economics studies, with particular emphasis on multiple-use problems. A native of Washington, D. C., he earned a bachelor's degree (1954) in forest management at Colorado Agricultural and Mechanical College. Awarded a Fulbright Grant in 1955, he spent a year at the University of Helsinki and studied forestry in Finland. He received a master's degree (1960) in agricultural economics from the University of California. Assigned to the Pacific Southwest Station's Berkeley staff since 1959, he has also served at the U. S. Forest Service's Division of Forest Disease Research in Beltsville, Md.; Lake States Station in St. Paul, Minn., Division of Engineering in Alexandria, Va.; and Northeastern Station in Upper Darby, Pa.

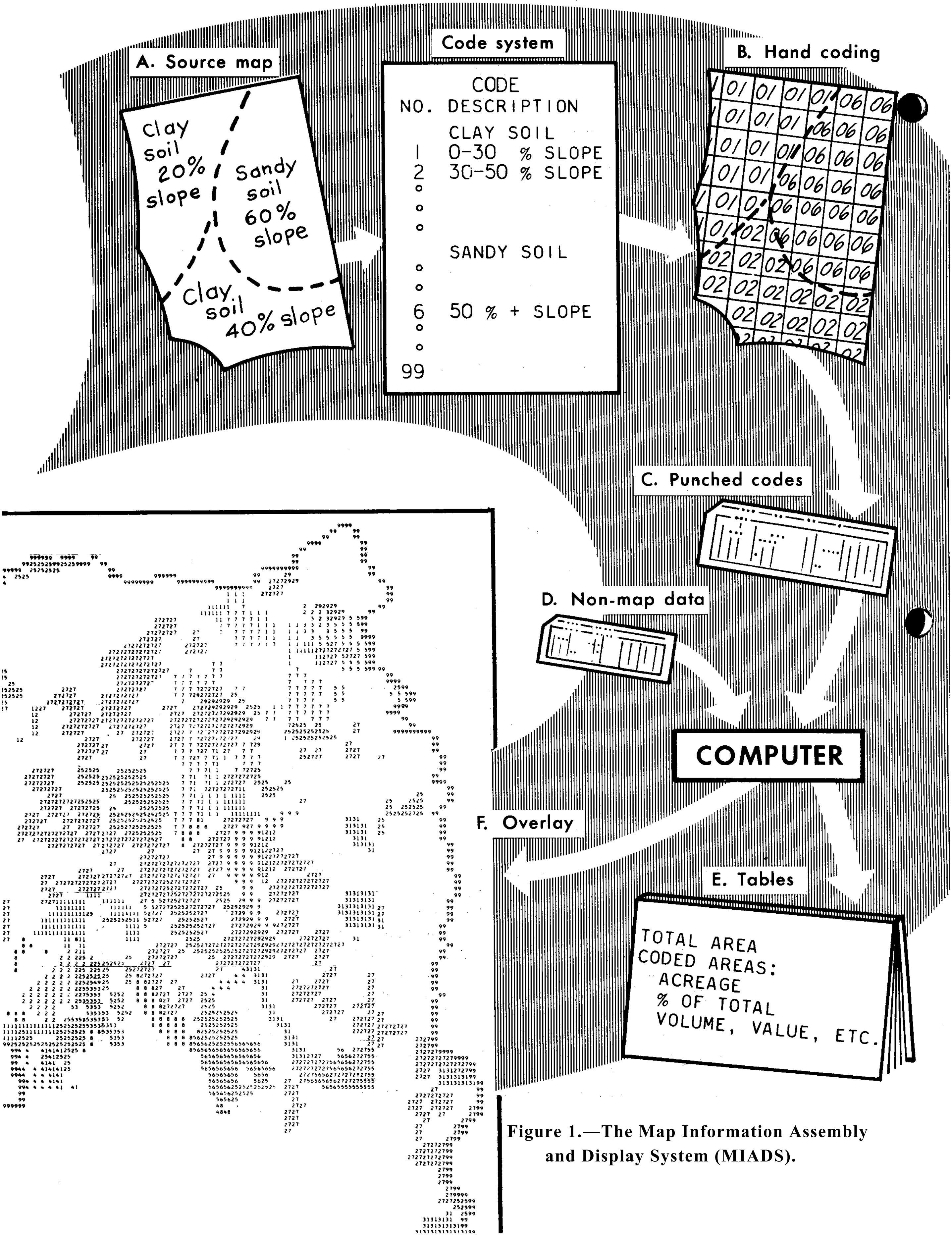


Figure 1.—The Map Information Assembly and Display System (MIADS).

Maps contain much of the information available to administrators responsible for managing large areas of land. To use information in making decisions, it must generally be assembled in a tabular form, which is more easily understood than maps themselves. To implement these decisions usually requires the preparation of overlays that show definite map locations. This can be an expensive process. Furthermore, vegetative cover, management objectives, or patterns of land can often change with the passage of time. Therefore, assembling and displaying map information is also a continuing process.

In a recent study we wanted to determine the possible impact of recreation development on timber production in a National Forest in California. Source data had to be assembled from timber and recreation management maps. It took us several man-months of work to assemble the initial data. We then had to do a great deal more work in order to simulate the impact of alternative levels of possible recreation development on timber yields.[1] This experience stimulated studies from which we have developed a computer-oriented system that can record, update, assemble, and display map information rapidly and efficiently.

How the System Works

The Map Information Assembly and Display System (MIADS)--as we call it--is a method for taking descriptive map and associated quantitative data and making them available for analysis and decision-making. It uses both handwork and a computer. The process starts with a *source map* (see also Glossary, p. i) that shows various land characteristics (fig. 1, A). Each different item of information wanted from the map is defined and given a 2-digit *code*. This limits each *code system* to 98 classifications (99 is the boundary code and is not tabulated). A rectangular grid is placed over the source map, and each grid *cell* is assigned a predetermined code number (B). The entire map is hand coded a *line* at a time; the wider the map, the more *strips*--columns of line--are required. Mapped codes are then transcribed to punch cards (C). The cards are then fed into a computer.

The assembled map data can be processed to calculate the acreage of an area represented by each code, and the proportion of an area in that code. The user, however, usually will want to combine with this map information certain supplementary *nonmap data* (D) which is also keypunched. For example, he may want to estimate the cost of converting from brush to grass, or to calculate the total volume of the sawtimber inventory. He supplies a brush-conversion cost or volume per acre for each relevant map code for computer multiplication by the corresponding code acreage. Given these "rates per acre," tabular output by the computer (E) for each *block*--group of lines--will contain information on acreages, proportions, and products (acreage times rates).

The computer at the same time produces a map card for each one that was entered, but with selected codes masked out. These output cards are put into an accounting machine. The resulting output consists of paper strips which, when fastened together, can become an overlay (F) of the same size and scale as the source map. Together the over-lay and the tables show the location and extent of certain mapped conditions.

MIADS uses numbers--chiefly because map characteristics can easily be designated by numbers. But any legal character acceptable to computer input-output can be used to represent these characteristics. Numbers have several advantages: their position in an overlay shows where they are; and counting them tells how much area there is of each map characteristic.

[1] Amidon, E. A., and Gould, E. M., Jr. The possible impact of recreation development on timber production in three California National Forests. U. S. Forest Serv. Pacific SW. Forest & Range Expt. Sta. Tech. Paper 68, 21 pp., illus. 1962.

Application of the System

Although several techniques to assemble map data are available, most of them do not take advantage of computers. The trend has been toward greater reliance on machines to prepare or interpret maps. Recent advances in photogrammetry, coupled with the rapid development of analogue and digital computers, show that progress is being made toward automation of both map preparation and map interpretation.[2] Tobler has discussed the possibilities for automation in cartography.[3] Berry compared sampling methods for extracting flood data from maps.[4] Wallis and Bowden have developed a method and computer program for obtaining area-elevation information for hydrologists.[5] But much more research is needed before the task of transforming conventional map information into a form suitable for automatic data processing can be fully automated.[6]

At present the system requires handwork to transcribe data for machine processing. Thereafter, computer methods are used to take care of (a) retrieving, updating, and combining descriptive map data for display in the form of overlays; and (b) summarizing map and supplementary quantitative data for presentation in tabular form.

We have applied the system to wildland problems that range from estimating the water-holding capacity of a watershed to simulating the management of a forested area over a period of time. MIADS, however, can be applied to all land-oriented problems--including those dealing with agricultural and urban land uses.

In developing and testing MIADS, we coded and processed information on about a million acres of land. We found that:

- The system works. This paper is not a proposal for future development, but reports on a working, operational management tool.

- The system is not expensive to use. In one application, involving considerable data preparation and processing, the cost was only fifteen-hundredths of a cent per acre.

- The system does not require the use of either expensive equipment or highly trained personnel to prepare map information for transcribing to punch cards. This first and most costly phase of the work can be done by personnel in field offices.

After data preparation, however, the job must be handled by persons trained to use key punch machines and someone familiar with electronic computers. The necessary technical information on data processing and on the logic of the program for this phase of the system is given in the appendix.

Case Studies

Several different jobs were undertaken during development of this system to devise data compilating methods, test the computer programs, provide cost estimates of hand and machine work, and produce the desired information.

This paper describes three case studies to illustrate various types of conditions to which MIADS applies. The first example is a relatively simple, static case. The next two examples introduce the complexities of updating management information and retrieving selected data from a large amount of stored information.

Water-Holding Capacity of a Watershed

The resident watershed scientist for the 46,800-acre Big Creek Watershed Pilot Area, located within the Sierra National Forest in central California, needed an estimate of the water-holding capacity of the soil for his water accounting calculations. He could have obtained this information by dot-counting or planimetering the soil conditions

[2] Bertram, Sidney. The automatic map compilation system. Photogrammetric Eng. 29(4):674-679. July 1963.

[3] Tobler, Waldo R. Automation and cartography. Geographical Rev. 44: 526-534. 1959.

[4] Berry, Brian J. L. Sampling, coding and storing flood plain data. U. S. Dept. Agr., Agr. Handb. 237, 27 pp., illus. 1962.

[5] Wallis, J. R., and Bowden, K. L. A rapid method for getting area-elevation information, U. S. Forest Serv. Pacific SW. Forest & Range Expt. Sta. Res. Note 208, 10 pp., illus. 1962.

[6] Langley, P. G. Can forest photo interpretation be automated? (Paper presented at annual meeting of Amer. Soc. Photogrammetry, Columbia River Sec., Portland, Ore., Dec. 5, 1961.)

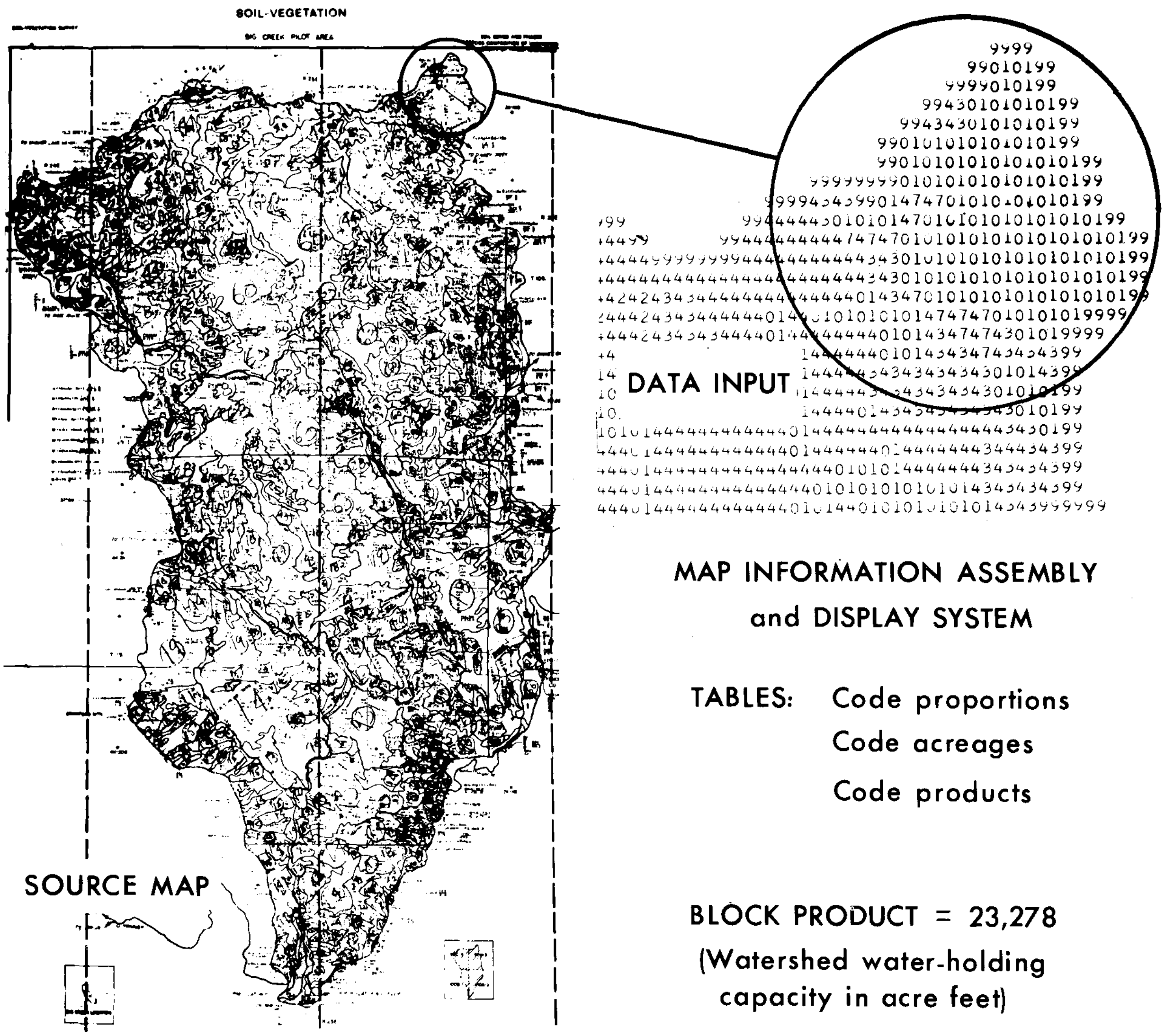


Figure 2.—Water-holding capacity of Big Creek watershed.

shown on a Soil-Vegetation Survey map of the area.[7] He did not need an overlay, and the key variables--soil types and depths--would change so slowly that updating mapped data was not a problem. In short, it was a straightforward data assembly task for the computer mapping program.

We established a code system before beginning hand coding. It allowed for 10 soil series groups, each containing 4 depth classes subdivided further into 2 slope classes. Three codes were added: for water, for unclassified soil (rock), and for the watershed boundary (code 99). Although the maximum number of codes in the system was 10 x 4 x 2 + 3 = 83, only 41 codes actually were used, primarily because not all depth classes occurred in each soil series classification. It took 3 man-days to hand code and verify the Big Creek Soil-Vegetation survey map, which had a scale of 2 inches per mile. Three strips, containing 390 lines were required to cover the watershed (fig. 2). Keypunching and keypunch-verifying took nearly another day.

The resident watershed scientist provided the supplementary data. Moisture-retention factors derived from laboratory analyses of soil profiles were already available for the watershed soils. A

[7]State Cooperative Soil-Vegetation Survey. Soil-Vegetation Surveys in California. 27 pp., illus., November 1958.

water-holding rate per acre was hand calculated for the average depth of each of the four possible depth classes in a soil series (table 1).

Further refinement of the rates to account for slope was not warranted.

Data processing took a half hour on an IBM 1620 computer. By contrast, the job of processing the same data on an IBM 7090 required only a half minute. The total water-holding capacity of the watershed soils was 23,278 acre feet.

Priorities for Type Conversion

This study taxed the capabilities of the mapping program much more than the first for four reasons. First, information was collected from several map

Table 1. Water-holding capacity and soil depths of Big Creek Pilot Area, Kings River North Working Circle, Sierra National Forest, by soil series

Item No.	Soil series: Number	Soil series: Name	Average soil depth	Water-holding capacity
			Inches	*Acre feet*
1	7319 & 851	Trabuco and Coarsegold	18	0.238
			30	.438
			42	.638
2	743 & 743V	Auberry and Auberry variant	30	.423
			42	.588
			54	.757
3	716 & 716V	Holland and Holland variant	30	.440
			42	.619
			54	.763
4	7121 & 7127	Corbett and Shaver	30	.321
			42	.477
			54	.633
5	7132	Stump Springs	42	.381
			54	.453
6	7117	Musick	42	.715
			54	.911
7	7129	Chawanakee	12	.118
			30	.296
8	854	Sacata	30	.414
			42	.552
			54	.720
9	749	Ahwahnee	30	.280
			42	.376
			54	.472
10	784	Tollhouse	12	.087

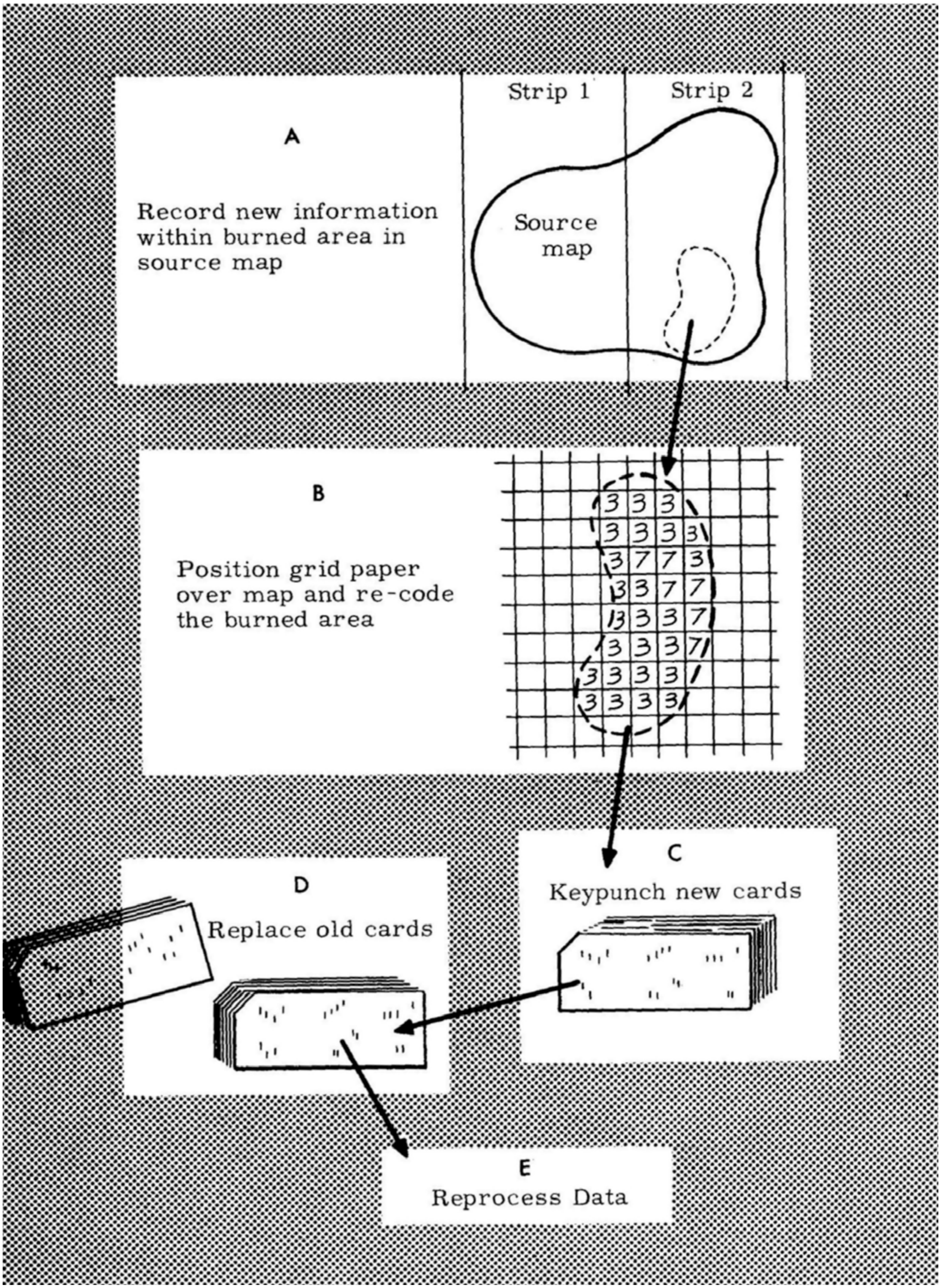


Figure 3.—Method for updating mapped information.

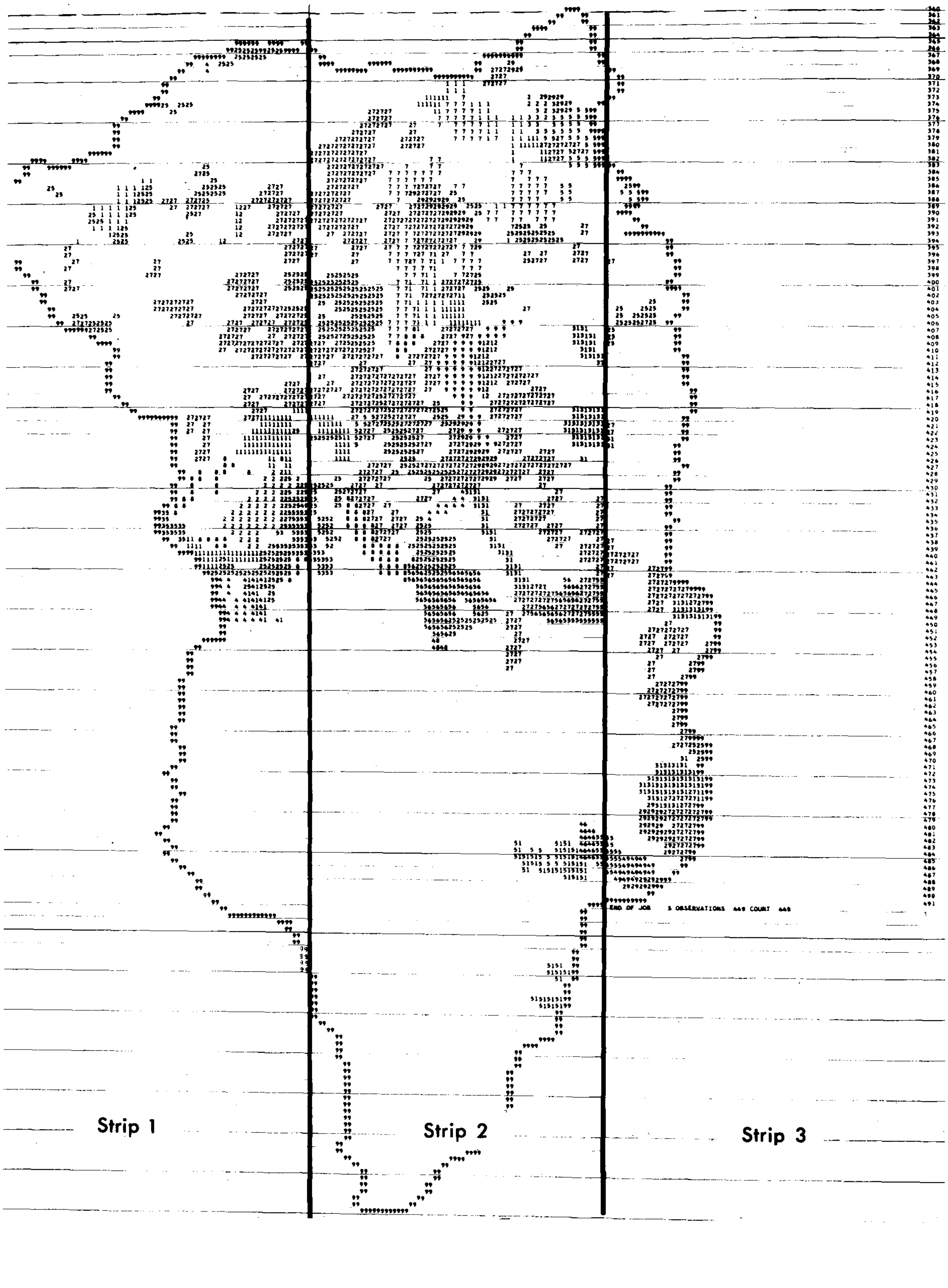


Figure 4.—Overlay showing timber priorities.

and non-map sources. Second, overlays were needed because the location and the amount of each condition were important. Third, fire had changed a portion of the ground cover and the data needed updating. Finally, several management functions were involved.

A schedule of priorities for converting existing vegetation types to more desirable types had been established for the Big Creek Pilot Area. There were 11 ranked treatments for timber, 6 for range, and 3 for wildlife. Ranking was determined by kind and density of existing vegetation, slope, and site quality, i.e., variables affecting treatment cost and productivity. Treatment categories, by public and private ownership, were combined during the hand coding stage into one map and into a 91-code system. Additional experience offset the increased coding complexity so that this study, like the first, required 3 days of hand coding and verifying.

After keypunching and processing, which took about one day or the same amount of time as in the first case, we found that a portion of the area had burned since the original map work was done. Part of the recent burn was already being converted into range and timberland. The burned area was delineated, several new codes added to the existing code system and the cards affected were pulled by hand and replaced by new data cards. If more extensive revision had been required, or if all the codes available had been used, the updated information would have been stored in a new code system until it was judged worthwhile to update the entire area by machine.

Treatment costs per acre for conversion to timber and range were obtained from foresters familiar with the area. A timber priorities overlay was a useful reference for this work (fig. 4).

Conversion at Big Creek Pilot Area has been confined to the highest priority classes on slopes less than 30 percent. Since site preparation generally requires complete removal of the existing vegetation, recent burns are scheduled for early treatment both to take advantage of the reduced cover and to prevent erosion. The cost estimates are only approximate because the lower priority classes have not actually been treated. Direct treatment costs--those for materials, equipment and labor, including supervision--at the site were $110 per acre for top priority brush conversion to range. The corresponding cost for conversion to timber ranged from $85 on recent burns to $125 on old brush fields.

The cost estimates and map data were processed simultaneously to give total treatment cost for each priority class. The results help determine the total budget required for a given amount of conversion work or, alternatively, help estimate the amount of work that can be accomplished for a given budget. For example, we found that about 1,100 acres, or 2.3 percent of the National Forest land in the watershed, was in the top priority class for conversion to timber. This was non- or poorly-stocked land with slopes less than 30 percent, timber site class 3 or better. The direct cost of tree planting this category would be about $112,000. In contrast, about 11,200 acres, or 23 percent of the National Forest land, were included in all priority classes, but this included land too steep to plant by present methods. All six priority classes for conversion to range, by brush removal and seeding, totaled 2,400 acres, or 5 percent of the National Forest land in the watershed. Only half this area, the top three priority classes, could be treated by known methods, at a direct cost of $143,000.

Timber and Soil Characteristics of a Large Area

This last study was the largest undertaking in two respects. First, the area covered was eight times larger than the cases just described and was coded twice. In effect, 760,000 acres were processed. Second, we needed a supplementary computer program that would take information from each of two code systems for subsequent processing and production of tables and overlays. We call this supplementary program the combinations program.

The Kings River North Working Circle on the Sierra National Forest contains about 380,000 acres of National Forest land. In 1961 it consisted of three timber management zones, now called primary, modified or deferred areas (fig. 5). Timber production is the main objective in the primary area. The modified area includes land surrounding lakes or beside streams and roads where watershed or recreation interests dominate. The deferred area consists largely of wilderness-type areas.

Two coding systems were established to collect detailed information only from the primary area. The first system of 97 codes contained data collected from a set of 22 Timber Stand-Vegetation

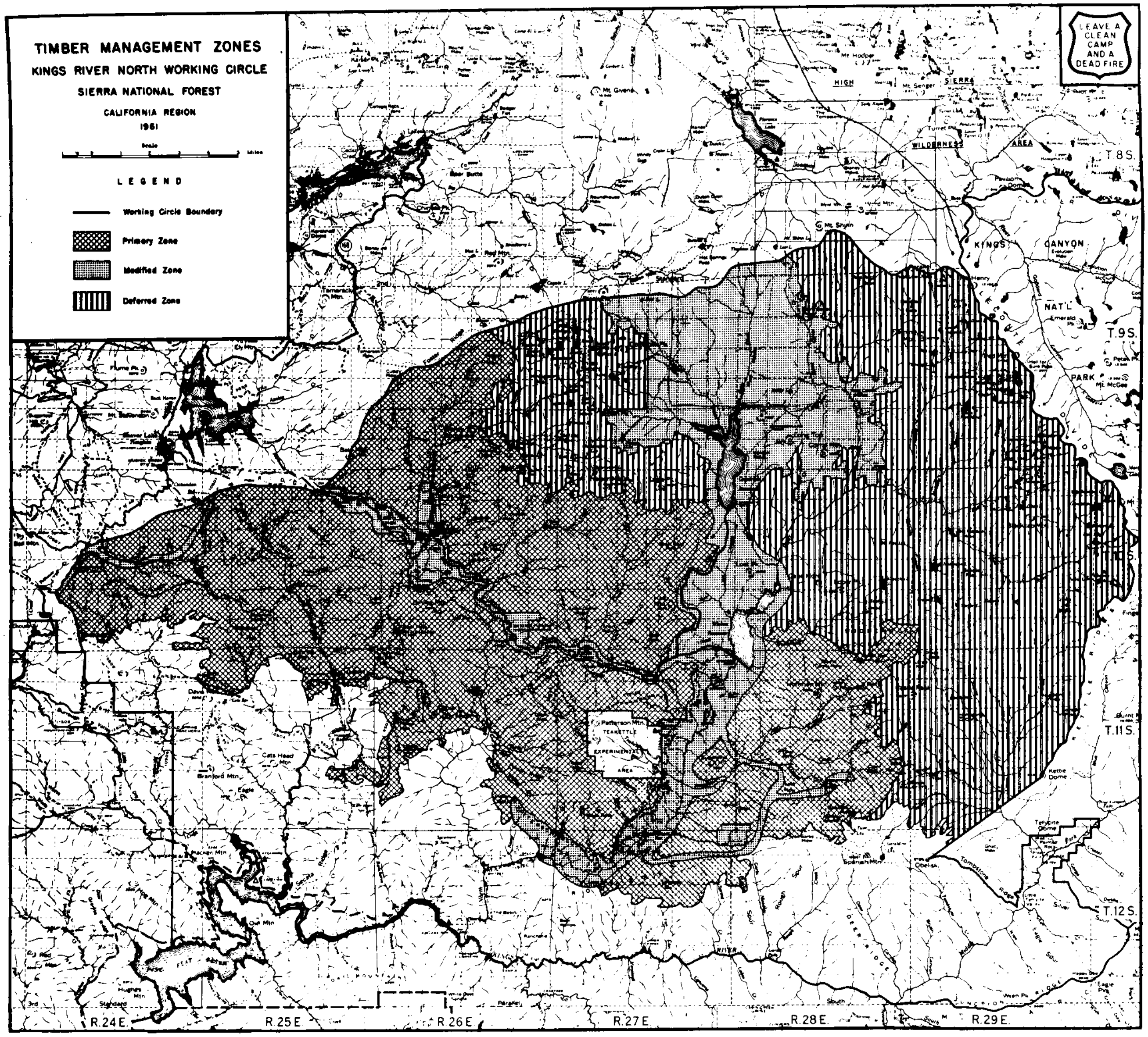


Figure 5.—Timber management areas of Kings River North Working Circle.

quadrangles at a scale of 2 inches per mile. Most of the codes represented timber stand age-density characteristics. The second, an 89-code system, drew on data derived from a corresponding set of Soil-Vegetation maps. These codes represented site quality, soil depth, and slope classifications. In each system, it took 5-6 man-weeks to hand code and verify and 7 man-days to key-punch the records. Both systems required 2,700 cards in 12 strips which, when joined together, formed a map about 4-½ by 6 feet long (fig. 6). For each sys-tem, producing block tables and one overlay under the control of the mapping program took 2-$^{1}/_{3}$ hours of IBM 1620 time.

The timber stand and soil vegetation tables by themselves presented useful inventory information. However, data were needed from both coding systems for such activities as tree planting or harvesting. For example, to find the most desirable timber cropland to plant within the primary zone, site quality, and soil depth data for slopes under 30 percent in one system needed to be combined with stand age-density classes from the second system (fig. 7). We also wanted information on the presence of bear clover (*Chamaebatia foliolosa*), also called mountain misery. This low plant competes strongly with tree seedlings for available moisture. To insure a high seedling survival rate, land must be terraced to scrape off bear clover roots before planting begins. The additional work due to bear clover can increase site preparation cost by about one-half.

From the two sets of working circle data, the computer combined high sites with low stand densities. High sites had no bear clover and were site class III or better. Low stand densities had less than 50 percent of sawlog stand, total-stand, and total woody vegetation density. To compare processing speed, the data were processed by two combination programs: first, a Fortran II, and then Fortran IV. The first run took 6-$^{1}/_{3}$ minutes, the second 3-$^{1}/_{3}$ minutes. The IBM 1620 was not used because each run would have required about 25 hours of machine time.

The combinations output was then processed by Fortran II and IV versions of the mapping program, requiring 3.8 and 2 minutes, respectively. Both printed tables and overlay cards were produced.

Mapping program output from earlier runs had shown that (a) 22,000 acres were free of bear clover and met the site and slope requirements for planting, and (b) 5,300 acres met the density criteria. The subsequent processing by the combinations and mapping programs showed that 1,850 acres met both sets of criteria. At a direct planting cost of $90 per acre, it would require $167,000 to plant these highly productive sites.

A great deal of planning and field work remains before specific planting activities can be scheduled. But the mapping and combinations programs have shown where to start and the size of the task.

Potential Uses of the System

Judging from these case studies, the system has greatest application where map characteristics, land treatment methods, and objectives change continually over time. The demand for wildland products is likely to increase in the future, and will require continual changes in land management inventories and plans.

Land managers are familiar with inventory maps, overlays, and tables and appreciate the effort required to keep their inventories current.

The uses to which the system is put will depend on the needs and imagination of the user. Many uses will be simply variations of inventory type work. Therefore, we consider two less obvious applications possible with the present system.

Assume that we must estimate the tax contribution to a county or other government unit. A map is available showing the distribution of assessment classes (i.e., use types) such as forest types, or other taxable lands. A second, administrative map shows taxation zones, and assessment rates are known. Once each map is hand coded independently, the contribution of each zone and the total estimated tax can be determined by the computer programs. The change in the total tax associated with incremental changes in the rate structure could also be determined easily.

Another use of the system is to simulate future forest conditions--20, 40, or 60 years away--which will result from decisions made both today and in the future. The present timber stands in the forest are classified by age and volume, and a timber rotation age is established. The combinations

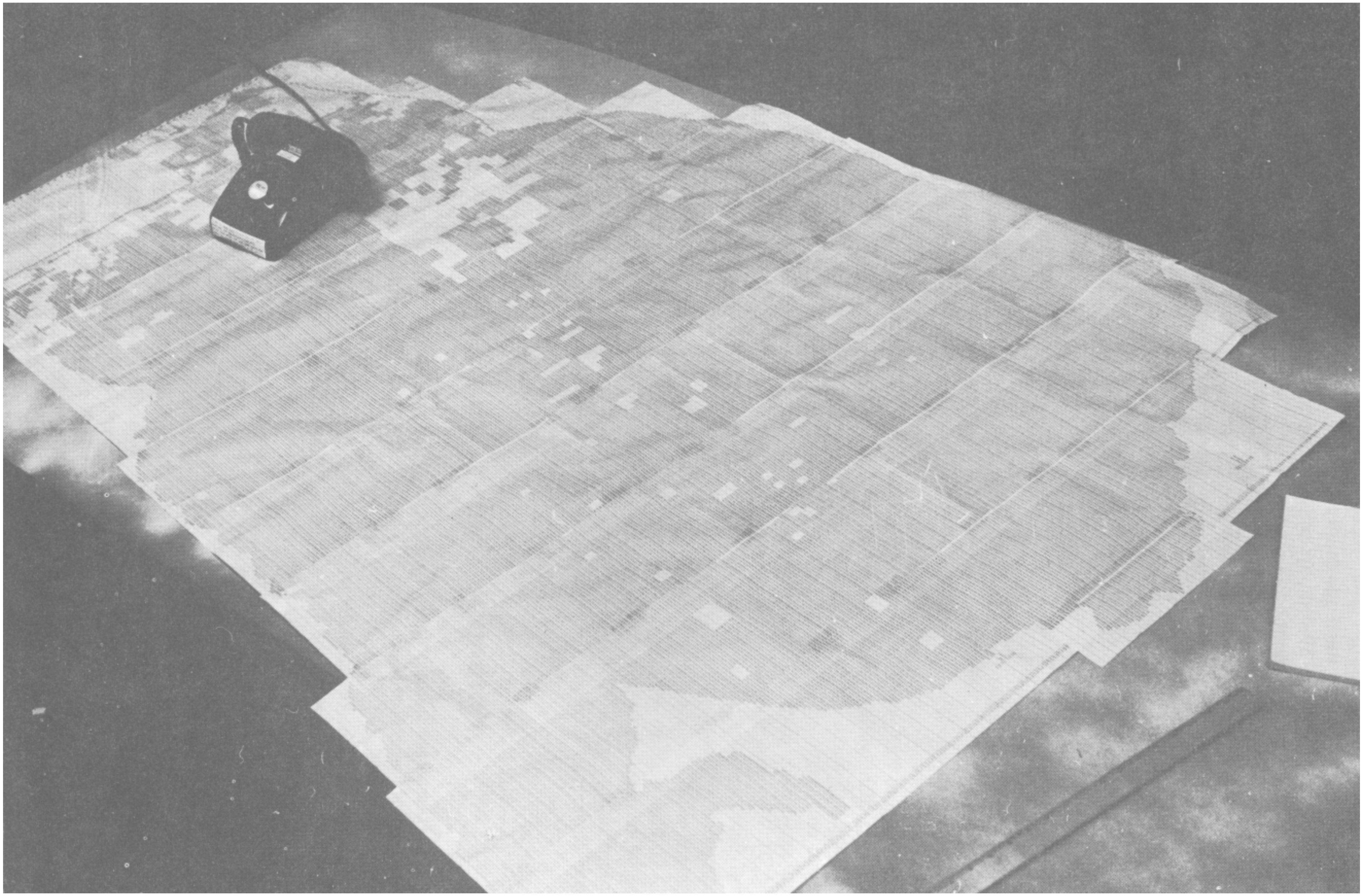


Figure 6.-The Kings River North Working Circle, coded into 2,700 lines, arranged in 12 strips.

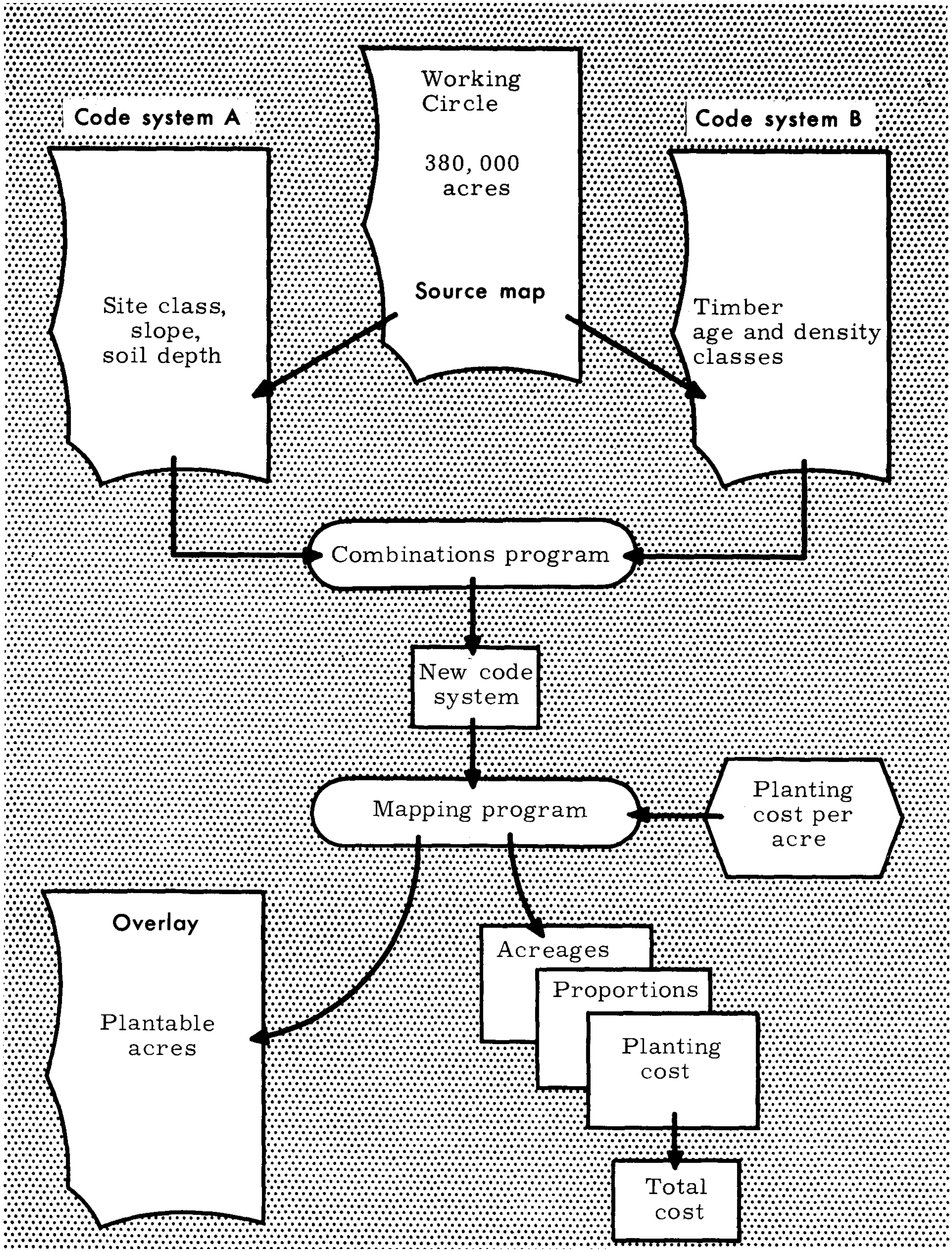


Figure 7.-Map data can be combined to determine plantable acres.

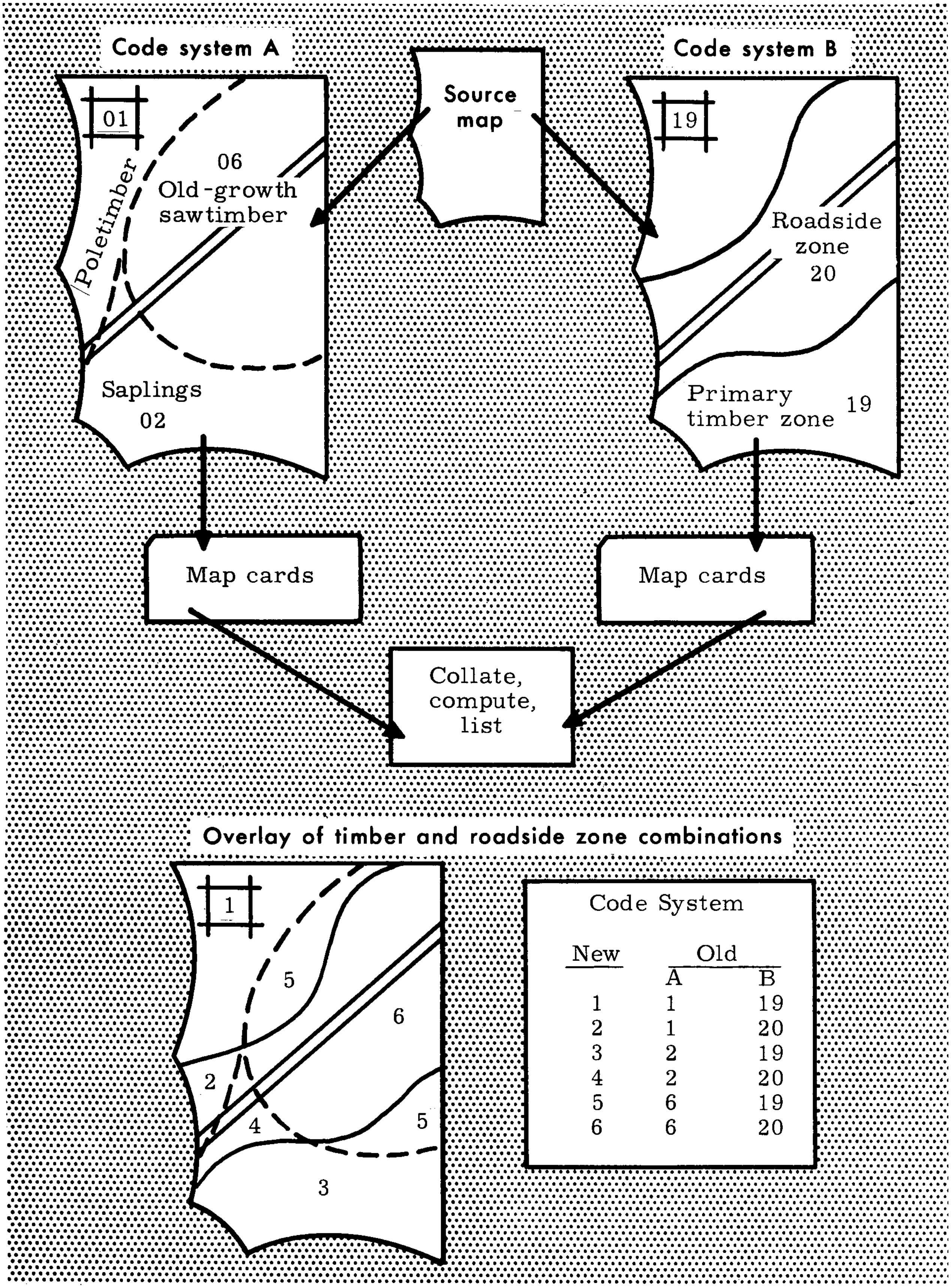


Figure 8.-Data from two source maps of a given area can be combined.

program is used to advance all stands by one age class and indicate those mature stands to be "cut" and "regenerated." The card output from this time increment becomes input to the mapping program. Then the age-volume data are used to compute growth and yield at the end of the end of the first time increment. The mapping program output in turn becomes input for the combinations program to start a second cycle. By repeated cycling, inventory tables and overlays can be produced for any number of future time periods. The impact of alternative rotation ages on future yields could be similarly predicted and displayed. This application should be useful in management planning and as a training aid for educational programs.

Special Characteristics of the System

Combinations Program

To make the mapping program of MIADS more versatile, we developed a supplementary computer program called the combinations program. The combinations program has two distinct functions: (a) changes (rearranges) codes within a code system, or (b) combines codes from 2, 3, or 4 systems simultaneously into one system of 99 codes or less. The first function allows a code system to be flexible. Codes inevitably must be rearranged because classification needs change over time, e.g., because of such agents as fire, or timber growth. The second function enables the user to combine selected information from several different maps of a given area into one code system. For example, one map might have timber classifications; the other land management zones.

The objective may be to produce a "new" map showing the distribution of timber stands within a roadside zone (fig. 8). The maps are hand coded independently, each with its own system of up to 99 codes. The grids are aligned so that coding begins from a common map reference point insuring that any given pair of cells will have the same coordinate position. Subsequent use of the combinations program should be anticipated in order to maintain this correspondence and avoid duplication of hand coding work. The two groups of map cards are merged and a computer searches the combinations. The desired combinations are assigned a new code in a single code system. As before, card output can be used to make a "new" map or, the map data can be processed by the mapping program to make an overlay showing particular conditions within the roadside zone and to produce simultaneously timber volume tabulations.

Computer Options and Output

A major objective throughout the development of the system was to produce computer programs adaptable to various machines. We wrote Fortran II programs specifically for the IBM 1620 computer and Fortran II and IV programs for the larger IBM 7090 computer. These programs can be adapted to other computers having Fortran compilers. For the programs developed, direct processing cost was always lower on the IBM 7090 than on the IBM 1620. The IBM 7090's greater expense per hour was more than offset by its higher computational speed. Land managers, particularly those located near smaller metropolitan areas, however, may have direct access to smaller installations so that the total cost for a particular job may be lower on the smaller machine. Since the programs were designed to be adaptable to quite different computers, additional features or increased speed may be gained by modifying the programs for specific installations.

Options

All options described below apply to both the IBM 1620 and the 7090 computer programs unless otherwise noted.

The land manager will often want data tabulated for each of a series of administrative or geographic units (block) as well as a grand total for all blocks. For example, in progressing from smaller to larger administrative units, he might want tables for each of five compartments in District "A" (with accumulated tables) followed by compartment tables in District "B" (with accumulated tables for A and B). He would be limited by computer memory storage to 20 individual blocks using the IBM 1620 or 99 for the IBM 7090.

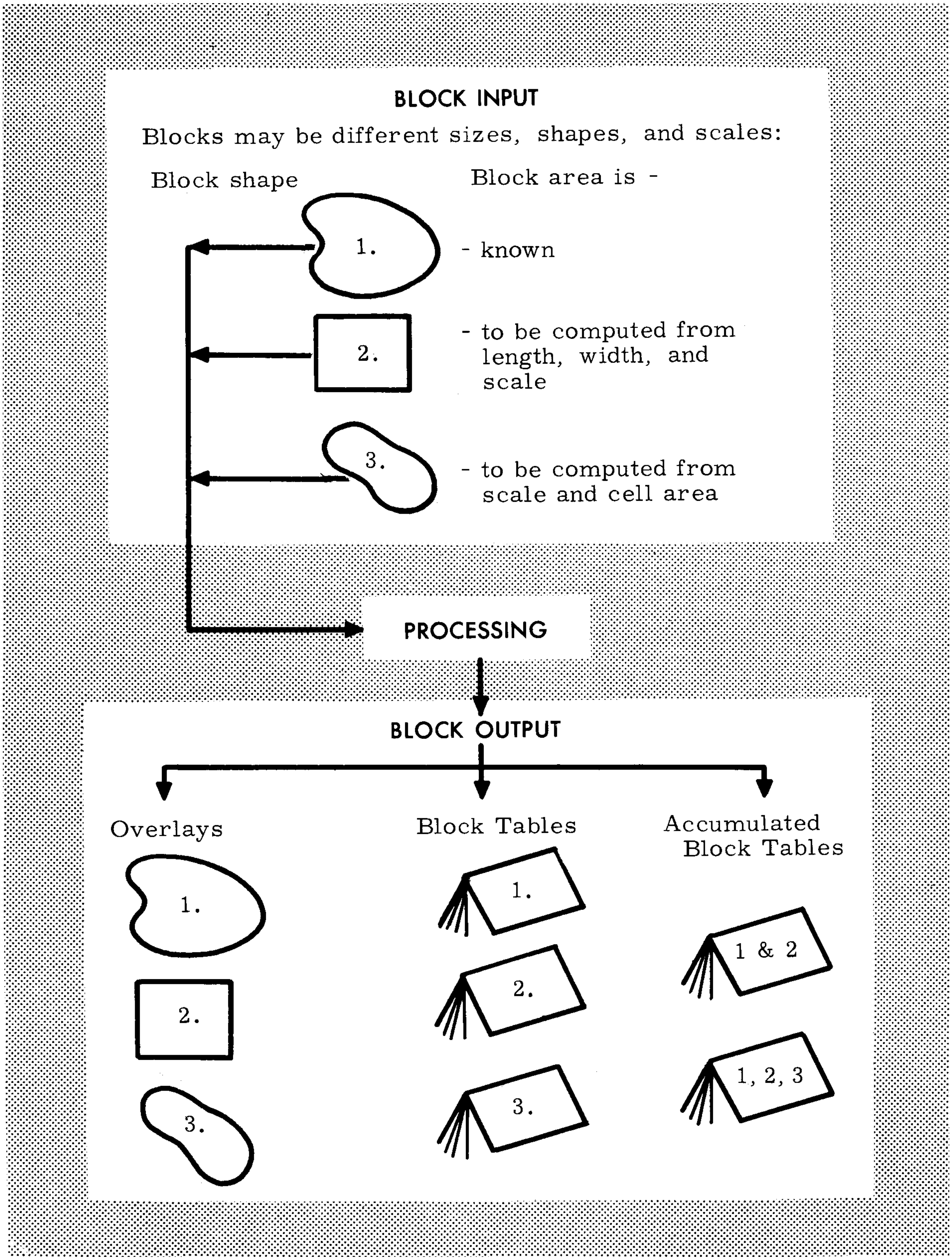


Figure 9.-Block input and output can be varied.

The area of each block can be supplied by the user or calculated from supplementary information. For a rectangular or trapezoidal area--such as an entire map quadrangle--block length, width and scale must be provided. Finally, the area of a block of any form can be calculated from its scale and the number of coded cells.

Figure 9 illustrates the preceding description. Three blocks, each with a different scale and whose areas are calculated by a different method, are processed in a single computer run. Three sets of individual block tables and two sets of accumulated block tables are produced. The user has the option of producing tables and/or map overlay cards.

Output

Overlays are of the same size and scale as the block input. Overlay scale cannot be varied within the computer. Blocks can be made the same scale externally by varying the spacing of the grid used while hand coding.

If rates per acre are provided, the user will get tables for each block or for accumulated block output. As mentioned before, the basic information contained in the tables are code acreages, proportions, and arithmetic products. The tables include cumulation of this information over codes which is similar to the accumulation of data over blocks discussed earlier. A complete list of the tables and miscellaneous items produced is shown below:

BLOCK TABLES

Block:

1. Code frequencies
2. Code proportions
3. Code acreages
4. Cumulative code proportions and acreages
5. Code products (acreages x rates)
6. Product proportions
7. Cumulative product proportions and acreages

BLOCK ITEMS

1. Block frequency total
2. Block acreage
3. Acreage represented by one grid cell
4. Total block product

ACCUMULATED BLOCK TABLES

Accumulated Block:

1. Code proportions
2. Code acreages
3. Cumulative code proportions and acreages
4. Code products
5. Products proportions
6. Cumulative products proportions and products

ACCUMULATED BLOCK ITEMS

1. Total block acreage
2. Total block product

Data Preparation

Hand Coding Map Data

Coding--translating map characteristics into numbers--by hand is the first and most expensive and time-consuming step of the entire process. At present there is no substitute for the human data interpreter.

Conceptually, there are many enumerative or stochastic methods of collecting the map data. An example of each method, respectively, is the planimeter and the dot-grid; both are commonly used in land management. The method used in this study seeks to reduce human work and errors as much as possible. Thus, the available machinery strongly influences the attributes of the system. For example, the systematic grid used and size and shape of grid cell (one-fifth by one-sixth of an inch) are dictated by the listing machine. Similarly, an accurate grid could be drawn on stable plastic, placed over a map, and codes recorded on a form --with some transposition errors. Instead, a less precise grid is printed as needed on a listing machine and the codes are recorded directly on the paper.

The user is not physically restricted to the techniques described in this section. For example, he could ignore the overlay option, devise any type of grid, collect data by any number of sampling schemes, and produce tables. However, only by following the recommended procedures will he be able to use all the system's options.

The computer programs will accept only map characteristics translated into 98 (or fewer) integer codes; code 99, reserved for boundaries is not counted. Letters have mnemonic value and with a redesigned program could be processed about as

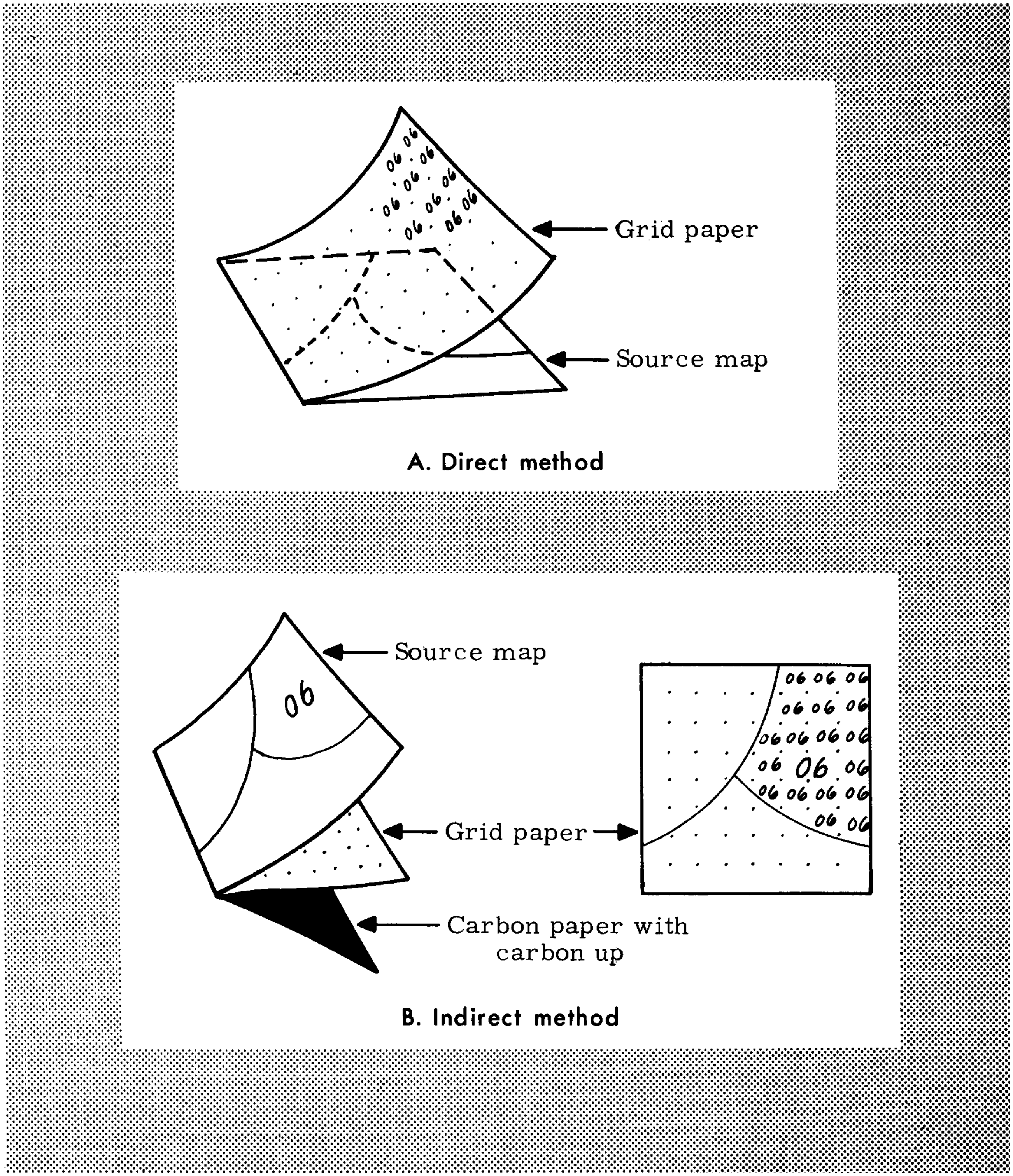


Figure 10. Two hand coding methods can be used in the system.

fast as numbers. The addition of pairs of letters or numbers and letters (e.g., AZ, B9) would allow more than a thousand categories in a code system. Although a map may have thousands of distinct categories, generally 98 or fewer groups will yield sufficient information for making particular decisions. Development of a larger code system is planned, however, because more intensive information requirements are expected in the future.

Two Coding Methods

Basically, we used two hand coding techniques. Only a light table was required as special equipment for the first one. Neither technique had a clear-cut advantage in all cases. Before the data are ready for the computer, they must be coded, verified, key-punched, and keypunch verified. Keypunch work is the same for both methods.

The first, or *direct*, method is used if the original source map cannot be reproduced cheaply because it is color-coded or for some other reason. The opaque grid and map are fastened to a light table. If machine-produced grid paper or its equivalent is used, the code numbers can be written within each cell (fig. 10). A second person should verify the original hand coding to minimize errors, particularly those arising from a systematic misinterpretation of a map characteristic. After verification the grid paper is sent for keypunching and keypunch-verifying.

Since a person looks through the grid both for coding and for verifying, the coding rate will depend on the clarity of the mapped characteristics. The most difficult situation arises when colors have been added to the original map, e.g., blue shading on a blue-lined map. The coder's symbols may obscure map features and make the verifier's job more difficult. Finally, the more maps being coded simultaneously the more light table area needed. Our experience with problems of this type led to the development of an alternative method.

In the second, or *indirect* method, codes are recorded directly on a copy of the original map. Nearly all coding work during program development was done by this technique. Coding, especially code verifying, was easier because the operators looked directly at the map; the grid paper lay underneath (fig. 4). The indirect method consists of four steps:

1. Fasten carbon paper with the carbon side up to any smooth, hard surface. Affix the grid paper to the carbon paper. Tape the map to be coded over the grid.
2. Delineate the boundary of each mapped condition with a hard pencil or ball point pen. Write the characteristic's code number once within the outlined area.
3. Verify the code number assigned to each condition and remove the grid paper. It will have a carbon impression on the reverse side.
4. Place the grid paper on a white surface so that the carbon impression is visible through the paper. If the impression is faint, a light table may be needed. Use a fine red ball point pen to write the same code number in each cell within a delineated area. This copy work could actually be done by a keypunch operator.

A variant of this method is to place the carbon paper on top of the grid. However, the impressions made by delineating and numbering the characteristics--and by accidental smudges--slow subsequent keypunching.

Those persons collecting data from sources other than paper maps should plan to develop their own techniques and equipment. Based on considerable experience with the indirect method, we found that about 13½ square inches of grid paper can be coded, verified, and numbered per hour. Given this rate, expense per acre will vary with map scale and labor cost. For a wage rate of $2 per hour and a map scale of 2 inches per mile, hand coding and verifying would cost one-tenth of a cent per acre. A hand coder must be familiar with maps and have methodical work habits. His coding speed will depend directly on his ability to remember each map symbol and its corresponding code number.

Appendix

This appendix describes the logic behind major program segments, gives more precise information for estimating cost, and includes detailed instructions for preparing and processing data. From this point on, an elementary knowledge of data processing would be quite helpful. Moreover, it is assumed that anyone using the system can get assistance from a computer programmer. Very likely the computer programs in this report will need minor modifications, even for other IBM 1620 and 7090 machines. Even with identical computers and peripheral equipment, job processing procedures may vary between installations. For this reason, and given the rapid changes in machines and in machine languages, a programmer will be needed for minor revisions of the programs.

A. Map Input Data Preparation Map

Card Format

Item	Field	Columns	
(1)	36 I2	1-72	Map codes (1-99)
(2)	I1	73	Block terminus indicator (1, 2, or blank)
(3)	I2	74, 75	Block number (7090:1-99), (1620:1-20) if any
(4)	I5	76 -80	Identification number

Details of map card preparation:

(1) Map codes unsigned, nonzero, two digit integers (i. e., 1-99) are processed. Blanks or zeroes are ignored.

(2) Block terminus indicator: an integer showing that the end of a block of mapping program input has been reached. The number 1 calls for tables based on the last block and for all tables up to and including that block. The number 2 calls for tabulations over the last block only. Blank (or zero) means that the end of the block has not been reached.

(3) Block number: the number of the block for which tabulations are to be made. Block integers must start with 1 and ascend sequentially (i. e. , 1, 2, 3, ...). A block number must follow (on the same record) a terminus indicator.

(4) Identification number: may be blank, or an integer. Card sequence numbers are not checked, any sequence can be used.

(5) Map card output has the same format as input. A block terminus indicator or number on an input map card will be punched into the corresponding output card.

Hand Coding

Hand coding represents such a large portion of total cost that the two methods used were described in detail earlier in this report. Hand coding cost data were based on the smallest possible staff and fixed investment: two men and a light table. Even remote field stations with sporadic hand-coding needs can expect to attain the rate achieved in this study: 13 1/2 square inches per hour for coding, verifying, and numbering. Unfortunately, this rate is not likely to decrease drastically given a basic dependence on human map data interpretation. Within the procedure described, costs may be reduced by training a keypunch operator to interpret map characteristics, thereby reducing or eliminating hand numbering.

One additional detail--strip alignment--must be determined at the hand coding stage. Usually two or more strips are needed to cover the map area to be coded. Coding must start from a definite, if arbitrary, base point and line. For example, the corner and edge of a map quadrangle makes a convenient starting position. This reference can be recorded separately or incorporated into the coded data deck. Map corners can be indicated by intersecting rows and columns of code 99's. Also, as for the overlay shown earlier (fig. 6), code 99's can be used to delineate an outside boundary, because they are not counted. Finally, card identification numbers insure vertical alignment between strips. A note is made on a form that record No. 1, first strip, is opposite record 133, second strip, which in turn is opposite record 354, third strip. By using code 99's to indicate reference points or boundaries and recording matching strip record numbers, every grid cell will have an implicit coordinate position and overlay placement over the source map will be uniquely determined.

Record Keeping

The type of forms required will depend on the magnitude of the job undertaken and can be devised by the user as the need arises. We found two types of forms useful in this study for keeping track of about 30 maps during coding and verifying. The first form had definitions of the map characteristics coded along with the assigned code numbers. The second form provided a continual record of each stage of processing for each map. It also had a place to sketch the physical relationship of the strips with one another, along with reference points and matching record identification numbers.

Listing

Map overlay strips are printed by listing computer card output. Ordinary machine paper may be used but with some risk of misinterpreting the codes. Codes are printed continuously, e.g., 9373.., therefore it is not immediately obvious that a particular cell is 93, 37, or 73. Special paper with 5 vertical lines per inch would avoid ambiguity in distinguishing codes, i.e., show 93/37/73. However, experience has shown that interpretation without the lines is not difficult if only a few codes are printed on the overlay and boundary or reference code 99's are printed.

Listing can be done on an IBM 407, 408, or 409 accounting machine or its equivalent. A special control panel can be wired to prevent zeroes or negative signs from being printed so that the overlay will not be cluttered. Listing is continuous unless a skip card

(X-punch in Col. 80), placed between strips, is encountered. All 80 columns of the map output cards are listed with a type bar setting of 10 characters per line inch and 6 lines per inch. This is a common spacing. No other spacing can be used without changing the computer program.

Grid paper can be made by listing a special deck of cards. Each card has a period (12-8-3 punch) in every other column beginning with 1 and ending with 73. The remaining columns (76-80) are used for sequence numbers, from 1 up to whatever number is convenient.

Map input records should contain only code numbers; no other characters are permissible. Therefore, any single card column should have only one punch. Letters and other characters are recorded in card columns by multiple punches. Erroneous multiple punches can be detected by passing all input records through a reproducing punch (e.g., IBM 514) using a double punch detector board. Two single punch characters, a plus (+) sign (12 punch) and a minus (-) sign (11 punch), cannot be detected in this manner. List and scan the data input to spot and change these signs before computer processing.

Those persons familiar with data processing may wish to use computer programs for the data checking and card handling methods described.

All 7090 table output is printed. IBM 1620 output will contain map and table cards intermixed. The two types of cards are separated on a card sorter by sort-suppressing on Column 2. A general (80-80) control panel is used to list the table output.

B. Supplementary Data Preparation

Only the mapping program uses supplementary information. The user must decide how block acreage will be computed and provide the necessary data. The only other type of supplementary data--rates per acre--is optional. These rates are multiplied by the computer times the acreage of each code.

Calculating Block Acreage

Block acreage may be computed in one of four ways, depending on the form of the map data or the user's preferences. The criteria for selecting a method and the computational procedure used are:

Method of determining block area:	Select this method when block--	Computer calculations [1]
1. Given by user	Area is already known	None
2. Block length, width, reciprocal of representative fraction	Is a square, rectangle, or trapezoid	A=(R2/K)(L)(W)

3. User provides reciprocal of representative fraction and program computes cell area with:	Has an irregular (or any) form	
a. Every cell coded, or		$A=(R^2/(K)(D))(N)$
b. Every other cell coded		$A=(R^2/(K)(D))(N)(2)$

[1] A = Block acreage
R = Reciprocal of representative fraction (e. g., 31680 for an R. F. of 1:31680)
K = (43560) (144)
L = Block length, inches
W = Block width, inches
D = Number of cells per square inch (30)
N = Block cell frequency

Rates per Acre

The rates, or weights, per acre may represent board-foot volumes, treatment cost, animal-unit months, or other units. Such diversity means that the magnitude of the rates may vary widely. In order to allow for this variation, rates are entered on control cards in E-format. Tables of arithmetic products, however, are in F-format (F10. 0) because this form is easier to read. Those preparing input data have the responsibility of scaling the rates accordingly. If scaling is overlooked, or done incorrectly, overflow may occur. Fortunately, the IBM 1620 will punch the correct value of an entry in E-format when-ever the F-format is exceeded. In this case, nothing is lost except some computer time. The larger computer does not have this feature so that as a partial substitute, block totals are printed in both formats. If these values are unequal, a gross scaling error has been made. Eight significant digits for each table entry probably will provide enough leeway, but if scaling proves bothersome a programmer can change table output to E-format. This change removes a chance for error at the price of additional clutter and misinterpretation of the exponential-type format by the ultimate user.

C. Computer Programs

Mapping Program

Computational procedure. --The mathematical bases for the tables and miscellaneous items calculated by the mapping program are shown below in algebraic notation. Note that "cumulation" is applied to within-block and "accumulation" to over-block summing of code items.

1. <u>Code defined</u>

c_{ijk}= ith number, jth code; kth block

$1 \leq c_{ijk} \leq 99$ $\quad$ $i = 1,n$

$j,k = 1,m$

$1 \leq m \leq 99$

2. <u>Block code frequencies</u>

$\sum_{i=1}^{n} c_{ijk}$ = the number of times the jth code occurs in the kth block

3. <u>Block frequency total</u>

$\sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}$ = the frequency of all codes in the kth block

4. <u>Block proportions</u>

$\sum_{i=1}^{n} c_{ij} / \sum_{j=1}^{n} \sum_{j=1}^{m} c_{ij} = p_{jk}$ = proportion jth code is of all codes in the kth block

5. <u>Block code acreages</u>

b_k = total acreage of kth block

$(P_{jk})(b_k) = a_{jk}$ = acres of jth code in kth block

6. <u>Acreage represented by one grid cell</u>

$$g = b_k / \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}$$

7. <u>Block cumulative code proportions and</u> <u>acreages</u>

a. Proportions

$S_k = P_{1k} + (P_{1k}+p_{2k}) + \quad . + (p_{1k}+p_{2k}+ . . . +p_{mk})$

b. Acreages

$s_k = a_{1k} + (a_{1k}+a_{2k}) + . . . + (a_{1k} + a_{2k} + . . . + a_{mk})$

8. Block code products

r_j = rate per acre, jth code

$(a_{jk})(r_{jk}) = v_{jk}$ = product of jth code, kth block

$$\sum_{j=1}^{m} v_{jk} = y_{jk} = \text{total block product}$$

9. Block product proportions

$$\frac{v_{jk}}{y_k} = t_{jk}$$

10. Block cumulative product proportions and products

a. Proportions

$$W_k = t_{1k} + (t_{1k}+t_{2k}) + \ldots + (t_{1k}+t_{2k}+ \ldots t_{mk})$$

b. Products

$$w = v_{1k} + (v_{1k} + v_{2k}) + \ldots + (v_{1k} + v_{2k} + \ldots + V_{mk})$$

11. Accumulated block code proportions

$$\sum_{k=1}^{m} b_k = B = \text{total accumulated acreage}$$

$$\frac{\sum_{k=1}^{m} a_{jk}}{B} = P_j$$

12. Accumulated block code acreages

$(P_j)(B) = A_j$ = accumulated acreage, jth code

13. Accumulated block cumulative code proportions and acreages

a. Proportions

$$S' = P_1 + (P_1+P_2) + \ldots + (P_1+P_2+ \ldots + P_m)$$

b. Acreages

$$s' = A_1 + (A_1+A_2) + \ldots + (A_1+A_2+ \ldots + A_m)$$

14. Accumulated block code products

$A_j r_j = V_j$ accumulated product of jth code

$$\sum_{j=1}^{m} A_j r_j = Y = \text{total accumulated product}$$

15. Accumulated block product proportions

$$\frac{A_j r_j}{Y} = T_j .$$

16. Accumulated block cumulative product proportions and products

a. Proportions

$$W' = T_1 + (T_1 + T_2) + . . . + (T_1 + T_2 + . . . + T_m)$$

b. Products

$$w' = V_1 + (V_1 + V_2) + . . . + (V_1 + V_2 +. . . + V_m)$$

Numerical example. --The arithmetic behind the mapping program tables is shown below in a simplified example. The objective of the hypothetical illustration is to show how all tables are calculated with minimum numerical detail. The raw data and hand calculations are followed by IBM 1620 output based on exactly the same information.

(a) Hypothetical data on which the tables are based

1. Coded map data

	Cell code numbers	
Block 1	1, 1, 1, 2, 2, 2, 2, 2, 3, 3	(First map input card)
Block 2	1, 1, 1, 1, 2, 2, 2, 3, 3, 3	(Second map input card)

2. Total block areas. This is the first option--the user supplies the acreage of each block. Since each block contains 10 coded cells and their total area differs, then so do their scales.

	Acreage
Block 1	100
Block 2	1000

3. Rates per acre for each code

Code	Rate/acre
1	.4
2	.5
3	.6

(b) The 14 tables. The method of calculating table entries is shown in footnotes.

1. Block code frequencies

	Frequency	
Code	Block 1	Block 2
1	1/ 3	4
2	5	3
3	2	3

[1] 1+1+1=3

2. Block code proportions

	Proportions	
Code	Block 1	Block 2
1	1/ .30	.40
2	.50	.30
3	.20	.30

[1] 3/10

3. Block code acreages

	Acreage	
Code	Block 1	Block 2
1	1/ 30	400
2	50	300
3	20	300

[1] (.30) (100)

4. Block cumulative code proportions and acreages

	Proportions		Acreages	
Code	Block 1	Block 2	Block 1	Block 2
1	.30	.40	30	400
2	1/ .80	.70	2/ 80	700
3	1.00	1.00	100	1000

[1] .30+.50 [2] 30+50

5. Block code products

	Products	
Code	Block 1	Block 2
1	1/ 12.0	160
2	25.0	150
3	12.0	180
	49.0	490

[1] (30) (.4)

6. Block product proportions

	Product proportions	
Code	Block 1	Block 2
1	1/ .245	.326
2	.510	.306
3	.21.5	.367

[1] 12.0/49.0

7. Block cumulative produce proportions

	Proportions	
Code	Block 1	Block 2
1	.245	.326
2	1/ .755	.632
3	1.000	.999

[1] .245+.510

8. Block cumulative products

Code	Products Block 1	Products Block 2
1	12.0	160
2	[1] 37.0	310
3	49.0	490

[1] 12.0+25.0

9. Accumulated block code proportions

Code	Proportions
1	[1] .391
2	.318
3	.291

[1] (30+400)/(100+1000)

10. Accumulated block code acreages

Code	Acreage
1	[1] 430
2	350
3	320

[1] 30+400

11. Accumulated cumulative code proportions and acreages

Code	Proportion	Acreage
1	391	430
2	[1] .709	[2] 780
3	1.000	1100

[1] .391+.318

[2] 430+350

12. Accumulated block code products

Code	Product
1	[1] 172
2	175
3	192
	539

[1] (430) (.4)

13. Accumulated block product proportions

Code	Proportion
1	[1] .319
2	.325
3	.356

[1] 172/539

14. Accumulated cumulative product proportions and products

Code	Proportions	Products
1	.319	172
2	[1] .644	[2] 347
3	1.000	539

[1] .319+.325

[2] 172+175

Input

```
     1
DERIVATION OF MAPPING PROGRAM TABLES. NUMERICAL EXAMPLES.  11/22/63
     1     2     3     1     2     3 2
3
1         100
1         1000
     4E-1        5E-1         6E-1
10101020202020202030303                                                      2 1     1
1010101020202020303030 3                                                     1 2     2
```

Output

```
JOB NUMBER      1 OF      1 JOBS


 JOB NUMBER      1 OF      1 JOBS
DERIVATION OF MAPPING PROGRAM TABLES. NUMERICAL EXAMPLES.  11/22/63
1 1 1 2 2 2 2 2 0 0 0 0 0 0 0                                                2 1     1
                          BLOCK   1 CODE FREQUENCIES
 NO T BL IDENT
                                1          2          3          4          5          6
  1 2  1    1                  3.         5.         2.         0.         0.         0.
 BLOCK   1 FREQUENCY TOTAL =             10.
                          BLOCK   1 CODE PROPORTIONS
 NO T BL IDENT
                                1          2          3          4          5          6
  1 2  1    1            .3000000   .5000000   .2000000  0.0000000  0.0000000  0.0000000
                          BLOCK   1 CODE ACREAGES
 NO T BL IDENT
                                1          2          3          4          5          6
  1 2  1    1                 30.        50.        20.         0.         0.         0.
 BLOCK   1 ACREAGE =                 100.
 ONE CELL =      10.00000000 ACRES
                 BLOCK   1 CUMULATIVE CODE PROPORTIONS AND ACREAGES
 PROPORTION    AREA PROPORTION     AREA   PROPORTION     AREA   PROPORTION     AREA
           1                     2                     3                     4
 .3000000          30.   .8000000          80.  1.0000000         100.  0.0000000          0.
                          BLOCK  1 CODE PRODUCTS
 NO T BL IDENT
                                1          2          3          4          5          6
  1 2  1    1                 12.        25.        12.         0.         0.         0.
 BLOCK   1 PRODUCT =                  49.
                          BLOCK   1 PRODUCT PROPORTIONS
 NO T BL IDENT
                                1          2          3          4          5          6
  1 2  1    1            .2448979   .5102040   .2448979  0.0000000  0.0000000  0.0000000
                 BLOCK   1 CUMULATIVE PRODUCT PROPORTIONS AND PRODUCTS
 PROPORTION PRODUCT PROPORTION PRODUCT  PROPORTION PRODUCT  PROPORTION PRODUCT
           1                     2                     3                     4
 .2448979          12.   .7551020          37.   .9999999          49.  0.0000000          0.
1 1 1 1 2 2 2 0 0 0 0 0 0 0                                                  1 2     2
                          BLOCK   2 CODE FREQUENCIES
 NO T BL IDENT
                                1          2          3          4          5          6
  1 1  2    2                  4.         3.         3.         0.         0.         0.
 BLOCK   2 FREQUENCY TOTAL =             10.
                          BLOCK   2 CODE PROPORTIONS
 NO T BL IDENT
                                1          2          3          4          5          6
```

Output (continued)

```
 1 1   2     2         .4000000  .3000000  .3000000 0.0000000 0.0000000 0.0000000
                       BLOCK  2 CODE ACREAGES
NO T BL IDENT
                               1         2         3         4         5         6
 1 1   2     2             400.      300.      300.        0.        0.        0.
BLOCK  2 ACREAGE =               1000.
ONE CELL =     100.00000000 ACRES
         BLOCK  2 CUMULATIVE CODE PROPORTIONS AND ACREAGES
PROPORTION   AREA PROPORTION    AREA  PROPORTION    AREA  PROPORTION    AREA
         1                   2                   3                   4
.4000000       400.  .7000000      700. 1.0000000     1000. 0.0000000        0.
                       BLOCK  2 CODE PRODUCTS
NO T BL IDENT
                               1         2         3         4         5         6
 1 1   2     2             160.      150.      180.        0.        0.        0.
BLOCK  2 PRODUCT =                490.
                       BLOCK  2 PRODUCT PROPORTIONS
NO T BL IDENT
                               1         2         3         4         5         6
 1 1   2     2         .3265306  .3061224  .3673469 0.0000000 0.0000000 0.0000000
         BLOCK  2 CUMULATIVE PRODUCT PROPORTIONS AND PRODUCTS
PROPORTION PRODUCT PROPORTION PRODUCT  PROPORTION PRODUCT  PROPORTION PRODUCT
         1                   2                   3                   4
.3265306       160.  .6326530      310.  .9999999      490. 0.0000000        0.
                       ACCUMULATED CODE PROPORTIONS
NO T BL IDENT
                               1         2         3         4         5         6
 1 1   2     2         .3909090  .3181818  .2909090 0.0000000 0.0000000 0.0000000
                       ACCUMULATED CODE ACREAGES
NO T BL IDENT
                               1         2         3         4         5         6
 1 1   2     2             430.      350.      320.        0.        0.        0.
ACCUMULATED ACREAGE =            1100.
         ACCUMULATED CUMULATIVE CODE PROPORTIONS AND ACREAGES
PROPORTION   AREA PROPORTION    AREA  PROPORTION    AREA  PROPORTION    AREA
         1                   2                   3                   4
.3909090       430.  .7090909      780.  .9999999     1100. 0.0000000        0.
                       ACCUMULATED CODE PRODUCTS
NO T BL IDENT
                               1         2         3         4         5         6
 1 1   2     2             172.      175.      192.        0.        0.        0.
ACCUMULATED PRODUCT =             539.
         ACCUMULATED BLOCK PRODUCT PROPORTIONS
NO T BL IDENT
                               1         2         3         4         5         6
 1 1   2     2         .3191094  .3246753  .3562152 0.0000000 0.0000000 0.0000000
         ACCUMULATED CUMULATIVE PRODUCT PROPORTIONS AND PRODUCTS
PROPORTION PRODUCT PROPORTION PRODUCT  PROPORTION PRODUCT  PROPORTION PRODUCT
         1                   2                   3                   4
.3191094       172.  .6437847      347.  .9999999      539. 0.0000000        0.
END OF JOB NUMBER     1 OF     1 JOBS
CONTROL CARD COUNT =      2, COMPUTER CARD COUNT =      2
IDENTIFICATION NUMBER OF LAST(NO.     2) CARD PROCESSED IS      2

 END OF JOB NUMBER     1 OF     1 JOBS
 CONTROL CARD COUNT =      2, COMPUTER CARD COUNT =      2
 IDENTIFICATION NUMBER OF LAST(NO.     2) CARD PROCESSED IS      2
```

Control card preparation. --The instructions below for preparing mapping program control cards are for the IBM 1620 and IBM 7090 computers except as noted.

(c) Control card number and format

Control Card Format

Type	Number of cards	Field	Columns	Description
A	1	I5	1-5	Header card - number of separate jobs
B	1		2-80	Title
C 1.	1	I5	1-5	Job number
2.		I5	6-10	Number of data cards
3.		I5	11-15	Largest code number used
4.		I5	16-20	Number of code numbers removed
5.		I5	21-25	Number of blocks
6.		I5	26-30	Number of rates per acre
7.		I2	31, 32	Select type of output
D	1-3	36 I2	1-72	Codes removed from card output
E 1.	1-99	I2	1, 2	Basis for computing block area
2.		F10.0	3-12	Block area as given by user
3.		F10.0	13-22	Map reciprocal of representative fraction
4.		F6.2	23-28	Map length in inches
5.		F6.2	29-34	Map width in inches
F	1-13	8E10.0	1-80	Rate per acre for each code

(a) Details for preparing and using each type of control card
(b)

1. Type A. A single card, used only once, showing the number of separate jobs.

2. Types B to F. These precede the data cards of each separate job.

3. Type B. Title. One card containing any legal characters. Leave Column 1 blank. If a title is not wanted use a blank card.

4. Type C. Note: Each of the six items below must at least equal one <u>regardless</u> of options selected.

(1) Job number: The number of the last job must equal the number of jobs shown by the Type A control card.

(2) Number of data cards: The number of map cards.

(3) Largest code number used: The largest code number occurring in the map cards. If unknown, put 99.

(4) Number of code numbers removed: The number of codes (1-99) to be removed (changed to zero) from map card output.

(5) Number of blocks: A number between 1 and 20 (IBM 1620) or between 1 and 99 (IBM 7090).

(6) Number of rates per acre: A number equaling the largest code number to which a rate per acre is applied. For example, if one rate per acre is to be multiplied by code 88 acreage, put 88, not 1.

(7) Select type of output.

a. Zero (blank) - Card output, i. e. , map overlay cards only.

b. One - printed table output only

c. Two - both printed tables and card output

Note: There is no option as to <u>which</u> printed tables are produced.

5. Type D. Codes removed from card output. Codes may be in any order, i. e. , 01, 99, 14, etc. If no codes are to be removed, insert one blank card.

6. Type E. Basis for computing block area. There are four ways to compute block area. For each block there is one card indicating which method is used, and the necessary information for that particular way. If no area is desired, insert a blank card for each block.

(1) Basis for computing block area

a. One - Block area given by user.

b. Two - Block area computed from map "scale" (reciprocal of representative fraction), length and width.

c. Three - Block area is computed from map "scale, " given that every grid cell is coded.

d. Four - As above, given that every other (horizontally) grid cell is coded.

(2) Block area given by user. The area in acres, being already known, is supplied by the user.

(3) Map reciprocal of representative fraction. A number necessary under options 2, 3, or 4. If, for example, the map scale" is 1/62, 500, put 62500.

(4) Map length in inches. Needed for the second option only. Measurement is to nearest hundredth inch, e. g. , exactly 10 inches would be 10. 00 inches.

(5) Map width in inches. As above.

7. Type F. Rate per acre for each code. Rates must be sequentially ordered in 1 - 99 successive fields corresponding to the code numbers used. Some rates may be zero. For example, assume that only codes 8 and 33 are to be multiplied by rates "A" and "B". The first card would be blank in the first 7 fields (70 columns) with rate "A" punched in the last, eighth field (Cols. 71-80). The following three cards, representing codes 9 - 16, 17 - 24, and 25 - 32, respectively, would be blank. The first field (Cols. 1 - 10) of the fourth (and last) would be punched with rate "B. "

If the rate per acre option is not used, insert one blank card.

<u>Processing cost</u>. --IBM 1620 and 7090 computer processing times are not precisely comparable. For the IBM 1620 used, which had a 60K memory and automatic floating point hardware, data processing includes card input and output. For the larger computer used, the IBM 7090 at the Computer Center, University of California, Berkeley, off-line input and output time is not included in total processing time.

Time estimates are presented for specific operations on the IBM 1620. For the faster IBM 7090, much grosser time estimates suffice.

Mapping program processing time estimates:

IBM 1620:		
	1. Load squeezed object deck	2 min.
	2. Type control cards	2-7 min.
	3. Process map cards	
	a. With overlay cards punched	18-25 min.
	b. Without overlay cards (tables only)	30-40 min.
	4. Compute each table	1/2 min.
IBM 7090:		
	Fortran II - Process map cards with overlay and table output cards	1-1 / 2 min/ M
	Fortran IV - as above	3/4 min/M cards

Combinations Program

Code changing and combining method. --Code changing applies to one code system. Code combining refers to 2, 3, or 4 code systems processed in a single computer run.

Assume a code system contains 3 codes: 1, 7, and 9. The user wants, say, to sequence his code system and simultaneously remove a code from an overlay. He writes 1, 7, and 9 in the first three fields of a control card. In the corresponding three fields of another control card he writes 1, b (nothing), and 2. The computer will leave code 1 alone, change 7 to zero and 9 to 2.

Combining codes is somewhat more complex. Suppose the user wants one combination, 0308 (03 in the first and 08 in the second code system) to become 04 in the new system. He writes 03 in the fourth field of one control card and 08 in the fourth field of a second control card. Now, however, he must decide what code number all excluded (unwanted) combinations will have. If he assigns zero, excluded combinations will be set equal to zero and subsequent processing of the new code system by the mapping program will ignore zeroes. If he chooses a nonzero code for excluded combinations, they will be counted subsequently. Therefore, his choice of zero or a positive integer affects the acreage base used on subsequent mapping program calculations.

Detailed instructions follow for handling both included and excluded combinations from two to four code systems simultaneously. The instructions apply to both computers.

Control card preparation. --

(a) Control card number and format

Control Card Format

Type	Number of cards	Field	Columns	Description
A	1	I5	1-5	Header card - number of separate jobs
B	1	--	2-80	Title
C 1.	1	I5	1-5	Job number
2.		I5	6-10	Number of data cards
3.		I2	11, 12	Number of code systems
4.		I2	13, 14	Number of codes to be changed or combined
D 1.	1-3	40 I2	1-80	Map codes to be changed
2.	1-3	40 I2	1-80	Map codes after they are changed
E 1.	1	I2	1, 2	Code number of an excluded combination
2.	1-5	20F4.0	1-80	Combinations selected from 2 code systems
3.	1-8	13F6.0, 2X	1-80	Combinations selected from 3 code systems
4.	1-10	10F8.0	1-80	Combinations selected from 4 code systems

(a) Details for preparing and using, each type of control card.

1. Type A: a single card, used only once, showing the number of separate jobs.

2. Type B to E: these precede the data cards of each separate job.

3. Type B: one card containing any legal characters. Leave Column 1 blank. If a title is not wanted, insert a blank card.

4. Type C:
   (1) Job number: the number of the last job must equal the number of jobs shown by the Type A control card.
   (2) Number of data cards: for one code system, this is the number of. map data cards. When combining codes, state only the number of data cards in one system, not the total number for all systems.
   (3) Number of code systems: a number between 1 and 4, inclusive.
   (4) Number of codes to be changed or combined: a number between 1 and 99, inclusive.

5. Type D: used only when changing codes within one system.
   (1) Map codes to be changed, and
   (2) Map codes after they are changed.

Special instructions:

a. A code whose numeric value is not changed, but which is wanted in the output, must be changed to itself (e. g. , 98 to 98) or it will be changed to zero.

b. As a result of "a., " above, a nonzero code can be changed to zero by omission. The converse is not true; a zero value cannot be changed to a nonzero code.

c. A code to be changed, say in the fifth position of (1) above, will be changed to the value in the fifth position of (2) above; that is, code values before and after changing must have corresponding array positions.

6. Type E: used only when combining codes from 2 to 4 code systems.

Number of systems combined	Control card type used
2	E.1.
	E.2.
3	E.1.
	E. 3.
4	E.1.
	E.4.

(1) Code number of an excluded combination. As indicated above, this card is required when combining 2, 3, or 4 code systems. Previous control cards have shown which code combinations are wanted; this card tells what to do with an unwanted (excluded) combination. An excluded combination may be changed to zero or a number between 1 and 99, inclusive. A zero value will not be counted later by the mapping program, a positive value will be tabulated. If a nonzero value is selected, say 7, then the corresponding position (seventh) in types E. 2., E. 3., or E. 4. must be blank. The blank prevents ambiguity; it avoids two combinations (1 included and all excluded) being assigned the same code number--7 in this case.

(2) Combinations selected from 2 code systems. Selected (included) combinations of 2 code numbers. As explained above, one position may be blank. The position (index) of an included combination is its output code number. For example, the first two fields might be the desired combinations: 99998972. The combination 9999 would be punched 01 and 8972 punched 02.

(3) As for (2), but for 3 code systems.

(4) As for (2), but for 4 code systems.

(c) Data input arrangement

1. Map card format

The format is exactly the same as for the mapping program and was described earlier.

2. Map card input

(1) One code system. The data cards are simply placed behind the control cards, in any order.

(2) Two to four code systems. Each card in a system must have an identification number identical to, and in the same position as, the corresponding card in the other system(s). The map cards can be match-merged and

placed behind the control cards if the identification numbers within each system are in ascending order. If the sequence of identification numbers vary within a system, compare the numbers on a reproducer and then merge the cards.

In merging two or more code systems, one must decide the system sequence. The sequence is determined once the selected combinations control cards (E2, 3, or 4) are made. For example, in combining two systems, the first desired combination might be 02 from system 1 and 33 from system 2. The first field of control card E. 2. would be punched 0233. When merging the two decks of cards, the first card must be from system 1, the second from system 2, and not vice-versa.

Processing cost. --IBM 1620 and 7090 computer processing times are not precisely comparable for the reasons mentioned earlier.

IBM 1620:

1. Load squeezed object deck	1 1/2 min.
2. Type control cards	2-7 min.
3. Process map cards	
a. Changing map codes	
(1) Blank cards	26 /min.
(2) Full cards	23/min.
b. Combining two code systems	
(1) Blank cards	9/min.
(2) Full cards	1 /min.

IBM 1620:

Fortran II - Combinations from 2 code systems	2 1/3 min. /M cards
Fortran IV - As above	1 1/4 min. /M cards

Additional Information Available

Those interested in using the system may obtain the following forms and additional information by writing to the Director, Pacific Southwest Forest and Range Experiment Station, P. O. Box 245, Berkeley, California 94701:

1. Data collection forms used
2. Reproducing punch control panel wiring diagram
3. Sample grid paper
4. Computer program listings (specify which of the below lists are needed):

   a. Mapping program
   (1) Fortran II, IBM 1620
   (2) Fortran II or IV, IBM 7090[8]

   b. Code changing and combining program
   (1) Fortran II, IBM 1620
   (2) Fortran II or IV, IBM 7090[8]

[8] Source deck can be compiled for either Fortran II or IV program.

GPO-969-749