of the exec structures. Implementation of the execs involved a substantial degree of knowledge of the IBM VM/370 system, CMS, several OS compilers, and UTPS. Although this process, in general, did not warrant the entering of the UTPS internal source code, in some instances it was unavoidable. Because the execs were developed with the intention of being used as an instructional tool for a one-week course, many of the accessory features of the programs, such as the plotting capabilities, were not incorporated into the execs. The end result was an exec that required little knowledge of CMS and VM/370 on the part of the user but some knowledge of the use of the UTPS programs in transportation planning.

It is recommended that this process be further crystalized and documented so that typical users may find it easier to implement UTPS in a CMS or any other timesharing environment. Interactive computing is now gaining widespread interest, and any interactive capabilities of software would thus make it more attractive.

The interactive software described in this paper has been turned over to the UMTA Office of Planning Methods and Support and is undergoing further development. It is expected that interactive versions of many UTPS programs will be used in further advanced UTPS training sessions.

ACKNOWLEDGMENT

This research was sponsored by UMTA's Office of University Research and Training. We wish to sincerely thank Philip Hughes, Nathaniel Jasper, and Judy Z. Meade of the Office of Policy Research for their support of this work. A large measure of credit for the short-course development goes to Larry Quillian of UMTA's Office of Planning Methods and Support. Ed DeLong, Chief of the Software Support Division, played a major role in helping debug the system. Most of the programming was done by William Collins, a 1981 graduate of Princeton University.

REFERENCES

1. M. Liotine and J.M. Lutin. Report on the Development of the Workshop on Interactive Applications of UMTA/FHWA Planning Tools. Urban Mass Transportation Administration, U.S. Department of Transportation, Sept. 1980.
2. Urban Transportation Planning System Introduction. Urban Mass Transit Administration, U.S. Department of Transportation, Feb. 1980.
3. IBM Virtual Machine Facility 1370: CMS User's Guide, 3rd ed. IBM Corp., White Plains, NY, March 1979.

Publication of this paper sponsored by Committee on Computer Graphics and Interactive Computing.

Interactive Model for Estimating Effects of Housing Policies on Transit Ridership

JEROME M. LUTIN AND BERNARD P. MARKOWICZ

This paper reports on computer graphics developed as part of an interactive computer model designed to assess the impact of housing policies on transit ridership in urban transit corridors. A set of programs was written in APL to implement the model in an interactive computer environment, with computer graphics used for both input and model output. A mode-split model that uses U.S. Bureau of the Census data predicts ridership for the transit line, based on discrete combinations of mode and access mode including walk-and-ride, park-and-ride, kiss-and-ride, and feeder bus. The program permits the analyst to input alternative residential patterns, with respect to location and density, in the transit corridor and to evaluate the effects on transit ridership by comparing various alternative housing policies. Computer graphics are used at two levels. First, as an input mode, graphics allow the planner to create new transit route alignments and station locations by using a screen cursor. The program then models station choice from the zones, on the basis of a number of variables, including driving or walking times to stations, transit fares, line-haul travel times, etc. As an output mode, graphics are used to display socioeconomic data, mode-split results, or any algebraic combinations of input or output data. Different types of graphic displays are used for data presentation at the zone level or station level. Throughout the development of the graphics, special attention was given to the readability of the output. The paper reflects the general effort to produce more visually attractive and commonly understandable outputs. Included in the paper are a description of the program design and organization, examples of graphic output, and a discussion of the ability of the model to provide useful output to policymakers.

Planners and urban policymakers have long recognized that a strong relationship exists between urban development forms and the existence of rapid transit systems in cities. In recent years, new transit systems have not led to significant positive changes in urban development. It is believed that the existing high level of automobile accessibility tends to obscure the increases in mobility achieved by transit. Many planners and policymakers believe that transit systems can be more effective in meeting the travel needs of the public, more energy efficient, and require less subsidy if land use planning in transit corridors can be coordinated with the planning of the transit system itself.

To achieve better coordination between transit planning and land use planning, the Urban Mass Transportation Administration (UMTA) has been making grants to cities to encourage urban development in transit corridors. However, there are major questions that need to be answered about the kinds of policies to be implemented. Planners need to know, for example, what kinds of housing should be encouraged in transit corridors. Should land close to transit stations be reserved for high-density apartments or be kept open to provide large lots for park-and-ride patrons? Given that land use regulations are difficult to enact and enforce, how does noncompliance with the plan affect the desired result? Because of the many unanswered questions, this research was directed toward the development of some quantitative tools that would provide planners with the ability to determine the likely effects of alternative land use plans on transit ridership.

The objective of the research was to develop a model that would take as input various housing policies and translate these results into transit ridership figures. The model was designed to estimate the proportion of commuters traveling by transit, the mode split, given that population could be clustered at various distances from the transit stops. By changing the location of population clusters, one alters the relative travel times and costs encountered in traveling to both transit stations and the central business district.

To test the model, a case study area was chosen in southern New Jersey. A proposed branch line extension to an existing rail rapid transit system is currently under study, and the corridor it is projected to serve was chosen as a test area for the model. The triangular transit corridor, 30 miles long and 15 miles wide at the maximum, covers parts of Camden and Burlington Counties in New Jersey and includes a population of about 450 000. The initial data set used by the model comprises 60 variables recorded in the 1970 census, population projections for the year 2000, and developable land areas for each of the 116 census tracts that comprise the corridor. For the purposes of the research, some of these tracts were further subdivided into subzones, which increased the total to 212 subareas or zones for analysis. A basic map of the zone boundaries was digitized and stored on disk along with a map of the zone centroids (see Figure 1).

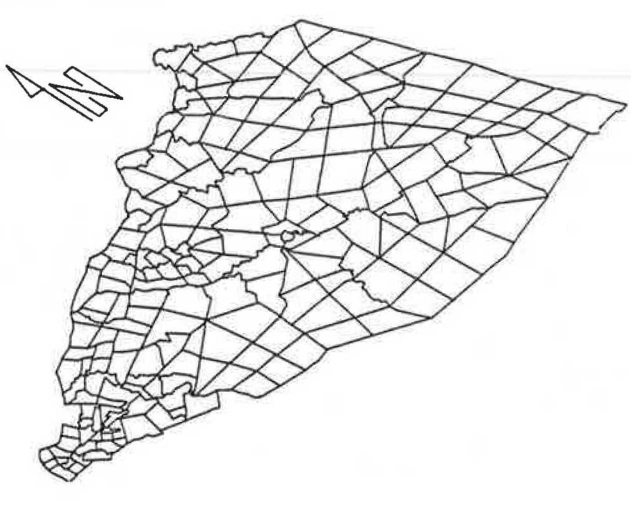
INTERACTIVE MODEL STRUCTURE

The nature of the research suggested that a number of alternative policies would be tested. This, coupled with the magnitude of the data base, led the investigators to use an interactive computer approach that would permit quick evaluation of many policy scenarios in a short period of time, with minimum data manipulation.

The interactive system is run under a virtual machine VM370/CMS operating system on an IBM 3033 computer at the Princeton University Computer Center. The model is programmed in VSAPL, enhanced by graphical processors, and uses auxiliary processors interfacing with CP, FORTRAN, and ASSEMBLER. The user accesses the timesharing system via a Tektronix graphics terminal.

The program is composed of a transit line input routine, a housing allocation model, a mode split model, and a routine to produce graphic output. These four routines are managed by a conversational program, called "POLIS", which controls the sequence of model execution and accesses the various routines and subroutines. The POLIS program contains user options to review or send output to a regular line printer, to modify interactively the numerous parameters used in the mode split model, to indicate which parameters bear non-standard values, and to control for valid parameters or command input names.

Figure 1. Base map.



The transit line input routine allows the user to input a new transit route alignment, to reset the program to a previous alignment, and to add or modify the number and location of stations. The functions of the input routine are to calculate the distance between each zone centroid and each station, to select the station nearest each zone based on the least "weighted" distance to all stations, to create around each station a new 0.5-mile² (0.8-km²) "special development district" zone to be superimposed on the original zones, and to assign to each zone a classification code based on the zone's location relative to both the destination--in this case the Philadelphia central business district (CBD)--and the nearest station. The 0.5-mile² zone is created in order to enable the user to apply special housing allocation policies to those areas within walking distance of the stations. The line input program is functionally divided into three subroutines. The first is designed to gather the input data, the second to perform the computations, and the third to create a new working data set. This data set includes characteristics for the new zones as well as adjusted data for the original zones.

Once the line input routine has been executed, the POLIS program takes control and allows the user to review parameters, produce base data maps, and run the housing allocation model or the mode split model. The housing allocation model simulates a 1980 housing distribution by allocating specific increments of dwelling units to the 1970 base year. A distinction is made between free-market and policy-allocated dwelling units. The policy-allocated number of dwelling units strictly conforms to location and density patterns set by the user to simulate policies to increase development at locations within the corridor. The free-market dwelling units replicate population gains and losses projected by the regional planning commission if no transit-related development were to be induced. Developable land is calculated as a percentage of total vacant land. Housing units are allocated to zones until target densities have been reached, based either on existing density levels, or on the basis of user specified growth policies. Since there is ample vacant land to accommodate growth, the allocation process fills zones according to a priority index set in the transit line routine.

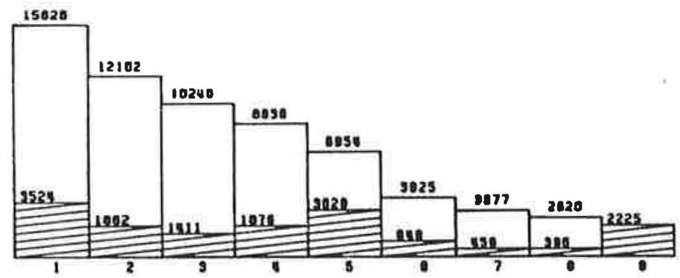
Each zone is assigned an index from 1 to 12. In turn each index number is associated with one of 12 classes of specific zones that share similar locational properties. The allocation model uses this index as a mean for specifying both the density and the priority class of each locational group of zones. The model takes the group of zones with the highest priority index and allocates to those zones a number of dwelling units compatible with the remaining vacant land and the specified target density, but not greater than the pool of dwelling units available for allocation. If the allocation of dwellings to this class of zones exhausts the vacant land, the program goes to the next priority class, and so forth, until the pool of dwelling units is allocated. If the pool to be allocated is greater than the capacity of the developable land, given the user-specified densities, the user is informed and allowed to get back to the top of the routine. Because of the large amount of vacant land in the corridor, this restriction applies only to the very low densities or large increments of dwelling units.

The housing allocation model first asks the user to input the total growth projected for the corridor, then to input the "percent effectiveness" that limits the percentage of vacant land available for policy allocation of housing. The allocation priority for each of the 12 classes of zones and the associated target density in dwelling units per acre are then input. Once the first subroutine has collected the information interactively from the user, a second subroutine computes the allocation and passes the output to a third subroutine that prints out a summary of the output to the screen and prepares a detailed report to be sent to the line printer. The POLIS program again takes control and, at user request, passes to the mode split model.

The mode split model is an eight mode and access mode stochastic choice model. The core of the program is a logit function that calculates the probability of choosing a given mode. The modes are automobile, carpool, express bus, rapid rail via park-and-ride access, rapid rail with kiss-and-ride access, rapid rail with feeder bus access, rapid rail with walk access, and rapid rail with bike access.

Figure 2. Screen summary of mode-split model.

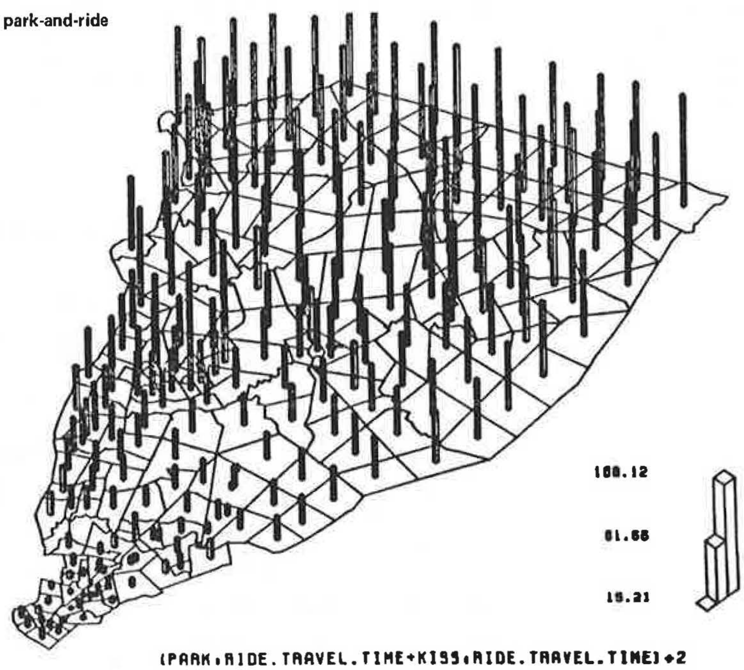
STATION AND CUMULATIVE RIDERSHIP



M O D E S P L I T					*T O T A L*	
AUTO :	26830	*PERCENT* :	.44			
CARPOOL :	18953	*PERCENT* :	.31		*COMMUTERS* :	61410
EXPR.BUS :	0	*PERCENT* :	.00			
PARK◦RIDE :	6208	*PERCENT* :	.10		*TRANSIT* :	15626
KISS◦RIDE :	5109	*PERCENT* :	.08			
FEEDER BUS :	2648	*PERCENT* :	.04		*PERCENT* :	.25
WALK◦RIDE :	1212	*PERCENT* :	.02			
BIKE◦RIDE :	450	*PERCENT* :	.01			

POLIS : INPUT COMMAND FOR :
TRANSIT LINE INPUT (L), THE HOUSING MODEL (H), THE MODE SPLIT MODEL (M)
TO REVIEW PR OUTPUT (R), SEND CASE PAGE (S), CHANGE PARAMETERS (C),
TO DISPLAY MAPS (D) OR TO QUIT ( ).

Figure 3. Screen output: average park-and-ride and kiss-and-ride travel times.



The mode split program calculates the impedance of each commuting trip at the zone level, including (a) travel time spent in vehicle, (b) travel cost (cost in dollars later transformed to income-earning minutes), and (c) excess time--time spent in waiting, transferring to, or accessing a mode. A subroutine makes the translation between real trip characteristics and perceived total impedance. This is done by multiplying travel time, cost, and excess time by weighting coefficients, and summing the terms. This sum, when exponentiated, represents the total trip impedance or disutility. The probability of choosing one mode is the ratio of its utility to the sum of all modal utilities. The zonal mode choice is expressed as the probability that an individual will commute to the CBD, multiplied by the population of the zone, and multiplied by the probability of selecting each mode. The model sends a summary of the corridor mode split to the user's screen, and prepares a detailed report to be sent to the line printer. The screen summary shown in Figure 2 also displays a graphic representation of the cumulative transit line ridership, along with the station loads (the shaded areas).

Following mode split calculation, the user again has the choice to go back to any main routine, to send the output to the printer or to produce computer graphics output. The computer graphics routine allows the user to display a variety of images on the screen, which can be copied via a hard copier or directed to a film plotter or a Calcomp plotter. The basic display includes a three-dimensional bar chart of a variable or combination of variables superimposed on a map of the corridor. The display routine offers a series of catalogued subcommands for plotting zonal mode split, total transit ridership, percentage of mode split, percentage by access modes, ridership versus density, or income. For documentation purposes, the routine also allows the user to plot any zonal census variable. Other subcommands enable the user to graph any variable by station.

BASIC COMPUTER GRAPHICS TOOLS

The program is implemented for use with a Tektronix graphics terminal attached to a Tektronix photo-processing hard copier, and through the main computer to a Calcomp 936 multicolor plotter. The Tektronix screen is composed of a 1023 by 780 matrix of addressable points, which are activated by an auxiliary graphic processor. The two dimensions of the matrix define the X and Y axes in screen coordinates. In addition to addressing screen coordinates, the processor accepts virtual coordinates for which a virtual window has been specified. The virtual window transforms the screen into a new cartesian system of coordinates with user-specified origin and scales along the X and Y axes. The graphics processor enables the programmer to define a straight line segment by two pairs of X and Y coordinates, absolute or virtual, or by two pairs of relative coordinates, each pair specifying the move along each axis from the previous point. Those two properties, the virtual system of coordinates and the relative coordinate system, are the basis for the computer graphics implementation.

The main objective of the graphics was to communicate quickly and efficiently the changes in mode split and transit ridership caused by changes in housing density. Figure 3 shows an example of the type of graphics developed in this case study. The zone boundary map is displayed in a simulated perspective mode. The perspective effect is obtained by rotating the map by an angle selected between 30° and 45°, and by compressing the figure along the Y axis. The first operation (rotation) is done only once, and the map coordinates are stored that way in the computer. The second operation (Y axis compression) is performed each time the map is drawn on the screen or on the plotter. It is achieved by setting a virtual window with dimensions not proportional to the 1023/780 ratio of the screen addressable matrix. If the ratio of the X range to the Y range is 1023 to 1560 for instance, the figure will be compressed along the Y axis and give an impression of perspective. The first advantage of this method over a spatial projection is that the computer is not required to solve complicated equations to locate coordinates. Another very important property is that, although virtually distorted, the computed distance between any two pairs of coordinates remains a scaled distance and not a projected distance.

GRAPHIC INPUT MODE

The graphic input mode is an essential component of the transit line input routine. At the user's request a base map is drawn on the screen along with the planned transit route.

A graphical cross-hair cursor appears on the screen and allows the user to input station locations. After each station input the program responds by drawing the link to the previous station. Each station location is read by the computer as a pair of virtual coordinates. To quit the input

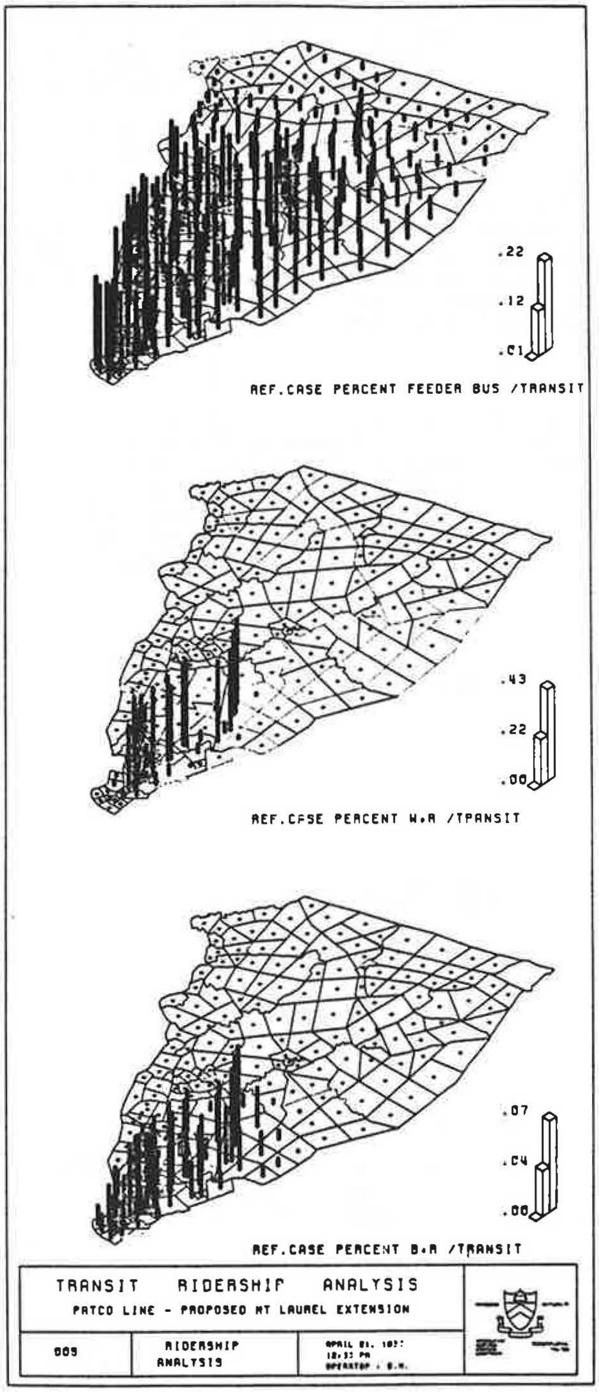
Figure 4. Graphic output mode: list of variables and input commands.

```
D
DISPLAY :INPUT COMMAND FOR :
         LIST AND BRIEF INSTRUCTIONS (L), MAP (M), OR SUB/QUIT ( )
L
CODE AND LIST OF VARIABLES
IN (M), INPUT ANY VALID APL EXPRESSION USING CODE VALUE
FOR NUMBERS OR NUMERICAL EXPRESSIONS, USE THE [   ]

 1  AUTO.DRIVERS          9  TRANSIT.RIDERSHIP  17  PROJECTED.POPULATIO
 2  CARPOOL              10  DWELLING.UNITS     18  AREA.OF.ZONE.(SQ.M.
 3  EXPRESS.BUS          11  POPULATION         19  VACANT.LAND.(ACRES)
 4  PARK∘RIDE            12  FREE.MARKET.ALLOC  20  WEIGHTED.AVG.TIME
 5  KISS∘RIDE            13  POLICY.ALLOCATION  30  WEIGHTED.AVG.COST
 6  FEEDER.BUS           14  TOTAL.COMMUTERS    40  WEIGHTED.AVG.EXC.TI
 7  WALK∘RIDE            15  DESTINATION.PROBAB 50  WEIGHTED.AVG.MILEAG
 8  BIKE∘RIDE            16  MEDIAN.INCOME
      21 - 28 : TRAVEL TIMES BY MODE / 31 - 38 : TRAVEL COST BY MODE
      41 - 48 : EXCESS TIME BY MODE  / 51 - 58 : MILEAGE BY MODE

DISPLAY :INPUT COMMAND FOR :
         LIST AND BRIEF INSTRUCTIONS (L), MAP (M), OR SUB/QUIT ( )
M
INPUT ALGEBRAIC EXPRESSION :
(24+25)÷[2]
PROPOSED TITLE :(PARK∘RIDE.TRAVEL.TIME+KISS∘RIDE.TRAVEL.TIME)÷2
ENTER NEW TITLE OR HIT RETURN IF O.K.
```

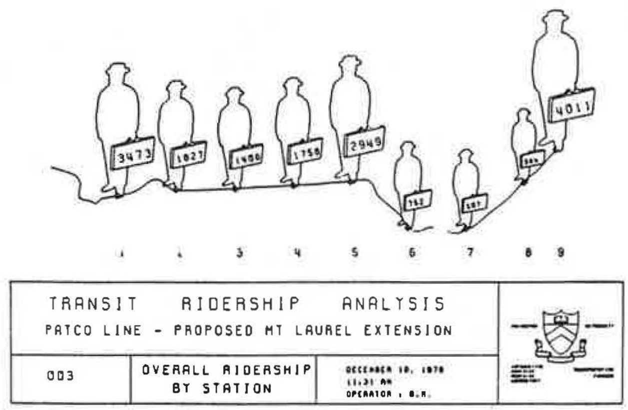
Figure 5. Sample of printed output.



mode, the user presses the Q key to signal that all stations have been input.

In order to create the 0.5-mile² (0.8-km²) special development districts mentioned earlier, the computer has to relate those districts to the original zones they intersect. The problem of boundary recognition is computationally difficult, and through interactive graphics, is left to the user's visual capabilities. A non-distorted portion of the map surrounding the first station is drawn on the screen along with the special development district. The user indicates which zones are included in the development district by pointing at their centroids with the cursor. Pressing a specific key directs

Figure 6. Height of character (commuter) is a function of projected loadings at each station.



the program to go on to the next station. Once all stations have been reviewed, the routine passes to the computing phase.

GRAPHIC OUTPUT MODE

The basic format used in the graphic output mode is a perspective-like map on which an output variable is represented by a vertical rectangular prism for each zone. The prism is located at the zone centroid and its height indicates the value of the variable. A subroutine automatically scales the output according to the absolute maximum and minimum values of the variable. To allow visual comparison between several outputs, the user has the ability to specify virtual maximum and minimum values. As shown in Figure 3, a scale automatically appears on the right of the display.

The plotting of variable size prisms is easily achieved in the relative coordinate system. Each pair of point coordinates (x and y) is expressed relative to the previous point in the array by the distance in units on the x and y axes. Therefore, by modifying the relative y coordinates of only three points in this case one can specify any height for the prism.

The display routine has been designed to allow the user a maximum of freedom. Entering the display program the user can review the list of variables that can be currently accessed as seen in Figure 4. By using the numerical code, the user then inputs any algebraic expression of the variables. Numbers of numerical expressions are expressed in brackets to differentiate them from numerical codes referring to variables. By using its own algebra, the program checks for a number of mistakes in the input expression and, if valid, proposes a title based on the algebraic expression to be shown on the map. The user is allowed to input his or her own title or to keep the proposed one.

Taking advantage of the high resolution of the Calcomp 936 plotter, as well as its increased useful surface, the programmer has the option to group plots by three. Figure 5 shows an example of this format. The size of the plot has been set to be reducible to an 8.50x11-in sheet on a Kodak 150 copier. A scale is automatically calculated and displayed with each map.

Finally, the display routine allows the user to access special graphic devices for station-by-station or housing class representation. Figure 6 shows a map of the proposed rail transit route with stations indicated by diamonds. At each station, a figure of "Mr. Commuter" is drawn. The size of the figure is proportional to the number of workers using the line to commute to work and selecting each station. The number of commuters is shown on the figure's briefcase. This display, while rather whimsical, may prove useful in communicating results to the public. This graphic is processed by using another property of the relative coordinate system. By multiplying all elements of a two-dimensional array of relative coordinates by a single factor, one can enlarge or reduce the object to be represented.

CONCLUSIONS

The software developed for this research takes advantage of the capabilities of interactive computer graphics in several ways. First, the quick turnaround of the system permits the analyst to explore a larger number of alternatives in a given time than is possible for a batch-mode computer model. The ease with which a model run can be performed encourages the user to explore a wide range of solutions and gives the user the opportunity to follow analytical paths that might not otherwise have been pursued. Subsequent runs can be made quickly, so that an idea can be tested while it is still fresh.

Second, the graphic form of the output shows at a glance the results of the model run. By using hard copies made from images on the terminal screen, one can immediately acquire an understanding of how results change by comparing visual outputs from one run to the next. One can pick out major shifts much more quickly from examining visual images than from examining numerical output. In addition, the pictorial quality of the output allows policymakers and non-technical individuals to grasp the implications of the analysis with greater clarity than can be achieved through examining reams of computer printout.

In an academic environment, the use of conversational computer graphics programs allowed undergraduates with little computer training to participate in the research. Undergraduates received 2 h of classroom instruction and then were assigned various policies to test. The easy manipulation of model parameters and sophisticated output gave students the sense of using a powerful lever and increased their motivation for working on the project.

ACKNOWLEDGMENT

This research was partially funded under a contract from the Program of University Research, U.S. Department of Transportation. We wish to thank Robert J. Ravera, Director, Office of University Research, and Richard I. Cohen, Office of Policy and Program Development, Urban Mass Transportation Administration, for their support and encouragement of this project.

Publication of this paper sponsored by Committee on Computer Graphics and Interactive Computing.

Network Simulation Interactive Computer Graphics Program

SHIH-MIAO CHIN AND AMIR EIGER

An overview of the deficiencies of the network simulation (NETSIM) program with regard to data input, data debugging, and analysis of the output is presented. Interactive computer graphic (ICG) enhancements are suggested as measures to eliminate many of the difficulties. The NETSIM/ICG program, which provides ICG capability in input-data preparation, input-data display, and both real-time and passive displays of link-specific measures of effectiveness, is described. The interactive data input, both graphical and keyboard in free format, follows a systematic procedure for obtaining the necessary information needed by NETSIM without reference to the user's manual. By using input-data display and input-data modification capabilities provided by the pre-NETSIM (PRENET) enhancement program, the user can easily comprehend and debug the NETSIM input data. As a consequence, significant reductions in the costs associated with data preprocessing are anticipated. The capabilities of providing both real-time and passive displays enable the user to more easily assimilate the information generated by the NETSIM simulation model and to comprehend the overall operation of the network. Consequently, these programs provide a heuristic approach to determining high-performance solutions at a minimal cost of both personnel and computer time.

As traffic flows through street networks, it experiences periods of congestion that may result from inadequate geometric design, signalization, or simply excessive demand. Traffic-simulation techniques are important tools for the traffic engineer in investigating the impacts of various traffic-control strategies. These simulation experiments can yield an enormous amount of data that could not be obtained in real life for economic or other reasons.

Among the network traffic simulation models, the network simulation (NETSIM) model produced for the Federal Highway Administration has been the most popular. NETSIM is an extension of the UTCS-1/SCOT simulation model, developed originally as an analytical tool for studying computer control of urban traffic networks. It has been extensively validated and is generally considered to yield reasonable results. The program is a microscopic simulation that deals with the movement of individual vehicles in an urban street network according to car-following, queue-discharge, and lane-changing theories. NETSIM defines the traffic network in terms of streets (links) and intersections (nodes). Each vehicle that travels through the network has associated with it data for, among other things, current speed, acceleration, and position. Detailed information concerning the operational characteristics of NETSIM may be found elsewhere (1,2).

THE PROBLEM

Although the NETSIM model is useful for accurately simulating traffic flows within an urban street network, certain deficiencies quickly become evident. These can be classified into three groups--data in-