This thesis is submitted in fulfilment of the course requirements for the Degree of Master of Science in the University of St. Andrews, Scotland. April 1972.

(Supervisor)

COMPUTER DRAWN PERSPECTIVE LANDSCAPES FROM CONTOUR DATA

by Cecile S. Abramson

ABSTRACT

"Computer drawn perspective landscapes from contour data" describes a computer program which makes a plotter drawing of a landscape using map data. The user must supply 1) a matrix of heights in a certain area, 2) an observation point and 3), a point to indicate the boundary of the view and the direction the observer is facing. The user may also supply information about bodies of water, cities or towns. The program stores the input and calculates the lines of the landscape and draws them on the plotter. It also supplies a frame for the drawing.

The program calculates the landscape lines by forming a field of vision, the left radius being formed by the observation point and the view point (both supplied by the user). The arc of vision is divided into 240 radiating lines. The angle of elevation for 80 points along each radiating line is calculated and the points with the largest angles are connected to form the outlines which are drawn.

The first chapter is a general survey of computer graphics. The rest of the thesis is concerned with the program itself. First there is a general description of the project and the problems involved in going about it, and then a detailed description of the Fortran program. The last chapter describes further work that would be relevant to this project.

Also included are illustrations and the Fortran program itself.

Table of Contents

List of Illustrations

Introduction

My aim in writing this program was to have the plotter make a landscape drawing which is both artistic and accurate. I wanted to write a program which might be of some use to engineers, architects, surveyors, and anyone else who would benefit from a landscape drawing of an area he is dealing with. Through this program a user can, without actually travelling anywhere, choose a point in an area and have the computer draw the landscape he would be able to see from this point looking in any direction. The user must simply supply, 1) a matrix of heights in an area, 2) an observation point, and 3) a point to indicate the farthest limit of the view and the direction in which the observer is looking. Any additional information about the landscape is optional, such as bodies of water, or cities and towns. If the user would like these to be drawn he must supply their locations.

My studies previous to computer science had been in fine arts, so I was familiar with the landscape drawings of the masters and had some experience in making landscape drawings of my own. My intention in supplying a foreground and a frame was to make the landscape outlines part of a complete picture and to give the picture depth. Without this the outlines of the hills and valleys would look suspended in space.

Before actually getting into the project I had considered working in colour and shading the landscape in order to make it look three-dimensional. I soon discovered that just achieving an accurate outline drawing was difficult enough.

The procedure I adopted for carrying out the work was to obtain a picture derived from exact mathematical formulae and then to subsequently modify the results according to such aesthetic considerations as, the smoothness of line, resemblance to the actual landscape, and the placing of the frame. These were all decided visually rather than scientifically.

The work was done in four main steps: 1) the top outline of the landscape, 2) the lines of the landscape in front of the top outline, 3) water, towns and other objects in the landscape and 4) the frame and decorative objects in the drawing.

Chapter I

Computer Graphics

There are three display devices that can be used by the computer; the digital plotter, the cathode ray tube (CRT), and the printer. This chapter will concentrate on the first two since the printer is rather primitive as a display device compared to the plotter and the CRT.

The printer would generally be used when a CRT or plotter is not available. The result is a crude but quick drawing. It can be very useful as a method of showing areas of light and shade, something that is more difficult and time consuming on the plotter. However, it is impossible to get a line drawing on the printer.

The digital plotter is a computer peripheral for making ink drawings or graphs on paper. It is controlled by a computer program which specifies each coordinate for a series of points and indicates whether the pen is to be raised (not writing) or lowered (writing), before moving to the designated point.

Two types of plotting surfaces are used - the drum and the flatbed. The drum plotter is generally smaller, less accurate and less expensive than the flatbed. A revolving cylinder, or drum, forms the plotting surface on the drum-type plotter. The paper is rolled around the drum and the pen is held by a wire just over the paper. When the plotter is instructed to plot a set of coordinates, the drum and the pen will move simultaneously in order to bring the pen to that point. The pen moves to the left and right along the wire to obey each Y axis command, while the drum turns forwards and backwards to obey each X axis command.

The flatbed-type plotter uses a table as its plotting surface. The paper rests flat upon it and the pen is held just above the paper. In this type of plotter, the paper is stationary while the pen moves in both X and Y directions.

The two main qualities which would be of interest to a plotter user are speed and precision. These qualities are directly related to the construction of the plotter.

The speed to be considered is the throughput time- the time that is required for the plotter to accept data and turn it into plotter output. The rate at which data may be accepted is known as the input speed. This depends on the size of the plotter buffer and the speed at which this buffer may be emptied in order to accept more data. The speed at which the drawing is produced is the operating speed.

The operating speed is made up of the:

1) slewing speed - the rate at which the pen may be moved from point to point when the pen is raised.

2) plotting speed - the rate at which the pen may be moved from point to point when the pen is lowered.

3) the rate at which the pen may be raised or lowered.

The precision of a plotter can be judged by the following:

1) accuracy - "the tolerance to which a point may be plotted on the plotting surface and is specified as plus or minus a fraction of an inch".

2) repeatability - the ability to return to the same location.

3) resolution - the smallest distance that two points may be placed next to each other and still remain distinct, i.e. the plotter draws them separately and they would show as two separate points under sufficient magnification.

There are two features in the construction of the plotter which are very important in determining the precision and the quality of the line drawn. These are 1), the number of directions, or vectors that may be plotted. (Figure 1 shows an 8 vector structure), and 2), the increment size or "step" size of the plotter. If the increment size is 1/100th of an inch in an 8 vector structure, the plotter could move a minimum of 1/100th of an inch in any of the 8 directions. As most plotters cannot draw straight lines in every direction, subroutines are provided which approximate arbitrary straight lines by a sequence of steps in the directions in which the plotter pen can move (see Figure 2).

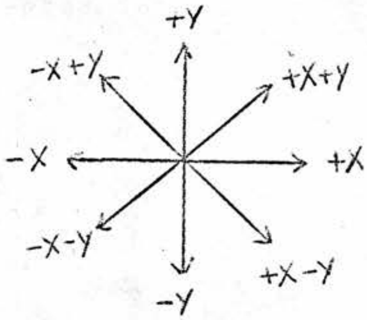


Figure 1

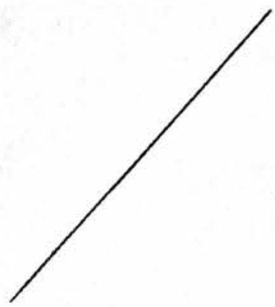


Figure 2a

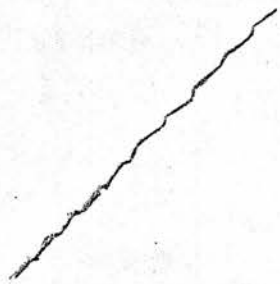


Figure 2b

In some plotters the increment size and number of directions is variable. In these plotters, the speed and precision are interdependent. A larger step size and fewer directions results in a faster speed. However, this also results in what is called a "staircase" plot, where the small steps that make up a line are visible. A slower speed resulting from a smaller step-size and greater number of directions would minimize this "staircase" effect.

Another factor which can influence the precision and quality of the line is the type of pen used. Among those available are ink, ball point (byro) or filter tip. The type of pen used affects the width of the line and its aesthetic appearance.

Plotters may be "on-line" or "off-line". With the on-line plotter computer data are sent directly to the plotter through an interface unit. With the off-line plotter magnetic tape, paper tape, punched cards or a manual keyboard are used to enter data into the plotter. Off-line operation provides the most efficient use of computer time, but is more expensive because of the complexity of the additional hardware necessary for controlling the plotting. The additional hardware usually consists of a tape drive and a control unit to operate the plotter.

The basic software that is necessary for plotting are subroutines which:

1) scale data - to assign certain physical values to the X and Y values which specify the coordinates that are actually plotted. This is done to make a scale model which will fit onto the plotter paper.
2) raise and lower the pen.
3) move the pen from its current location to a specified coordinate. This is done by calculating an approximation to the straight line between two points.[7]

The CRT is a picture tube similar to a television tube. The digital impulses control the movement and intensity of an electron beam so as to produce visual images upon a screen, much as a television image is produced.

Basically there are two types of CRT; Calligraphic displays and raster displays. "Calligraphic displays 'paint' the parts of a picture on the CRT in any sequence given by the computer. The electron beam in a calligraphic display is moved from place to place in a pattern that traces out the individual lines and characters that make up the picture. Raster displays make pictures in the same way that television sets do: the image is painted in a fixed sequence usually from left to right and from top to bottom. The calligraphic display has the advantage that information to displayed can be stored in computer memory in any order, whereas information for a raster display must first be sorted from top to bottom and from left to right so that it can be put on the screen in the correct sequence." [10] Because of this method of sorting, the raster display is largely used for presentation of text. It has the potential of producing pictures with a large range of light and dark tones. In addition, the raster display is less costly than the calligraphic display because suitable components for it are now mass produced.

Computer display equipment consists basically of a digital computer, which specifies the point-by-point coordinates of the desired display, together with circuits that convert the digital output into analogue form for deflecting the electron beam in a CRT.

The information to be projected on the face of the CRT is specified in terms of a coordinate system, or grid of points, that covers the tube face. Most display systems have a minimum of 1024 rows and 1024 columns. Analogue circuits automatically trace outlines on the screen in a continuous process. To post a character the character generator circuit produces a wave form for the X and Y deflection systems and an associated pattern for turning the beam on and off. To post a straight line on the screen, the analogue line generator will produce coordinated 'ramp' wave forms for the X and Y deflection systems so that the electron beam moves smoothly and continuously from one end of the straight line to the other.[10]

Plotter vs. Cathode Ray Tube

Both methods of display have been in use since the early 1950's. The type of work the user is doing would dictate which method of display would serve him best.

"Comparing these two forms of readout, the CRT display is used where fast writing speed and man-machine interaction are important. On the other hand, the plotter is used in those applications where writing speed and man-machine interaction are not as important as having a permanent, large area reproduction of computer-generated data."[7]

A CRT display can produce a straight line in 3 to 40 microseconds, depending on length. This fast writing speed makes it possible to generate a whole series of displays rather than one drawing at a time as on the plotter. Thus, it becomes possible to depict moving objects. For example, through a series of displays, a biologist could demonstrate to his students how blood flows through the arteries, or how a heart beats. In the same way, by photographing each display, an animated film can be made by a computer. In fact it is possible to make an animated film using the plotter (see page 13), but is very impractical because the production of thousands of individual drawings is generally involved.

The other very great advantage of the CRT, - man-machine interaction, makes it possible for the computer to help in such things as design work. The designer can give specifications for an object and by using a light pen and by pressing control function keys on an interactive console he can change, move, rotate or distort anything placed on the screen. Thus the CRT can be used as a terminal for input as well as output.

In the same way, a pilot could practice "take-offs and landings on a simulated airfield that can assume any orientation on the display screen as he operates 'controls' for engine power and aircraft altitude."[10]

If a user is primarily interested in a large and very accurate drawing he would most likely prefer a plotter to a CRT. In drafting, for example, where accuracy and size is of prime importance, the plotter would be more useful. (E.g., The Gerber Scientific Company produces large, precision, flatbed digital drafting machines. The model 32 has an accuracy of up to ± 0.0009 of an inch over a working area of 48 x 60 inches.) Other applications, such as contour mapping, weather mapping, the production of charts and graphs are better suited to a plotter.

Another advantage of the plotter is that it is possible to use colour on a drawing simply by changing the pen. The ability to use colour is not yet widely available on the CRT.

Finally, the cost of the CRT at present is much greater than the plotter.[7]

Examples

1) Photocomposition: the CRT in the Graphic Arts

Originally, the use of the computer in typesetting was limited to the justification and hyphenation of lines. To justify a line is to space out the words on a line so that they completely fill the specified column width. It "requires the adding up of the width of all characters which will print on the line, and then distributing the space left over between words (or letters). Distribution must be held within very narrow limits to get acceptable typographic appearance. The justification process may require hyphenation (the breaking of words at their proper syllables) to eliminate abnormal amounts of white space between words."

Today, with the aid of the CRT, a book can be completely typeset by computer. Each page is composed according to the instructions of the programmer-typesetter and displayed on the CRT screen. A photograph is taken of every page and sent to the printer who etches a negative on metal from the photographic positive. Printing is done from this negative.

The programmer-typesetter is able to choose the type style, arrange the spacing between lines, place running heads on each page, have the pages numbered, use justification and hyphenation, etc.

In one such system, the text of a book and its typesetting codes are fed into a computer through a terminal such as a typewriter. If the text is already on tape the typesetting coding can be inserted and the tape can be used as input. A high-speed print-out is proofread before the book is photocomposed.

A tape is created of the text of a book and its type-setting instructions. Basically these instructions consist of 1) size of type (measured in points), 2) type style or font (the three basic types are bold, italic and Roman, 3) line length, 4) body leading - the amount of vertical space between lines.

This tape becomes input to another program which reads the characters and format codings on the tape and displays a page of text at a time, as modified by the format codings, on a cathode ray tube.

A text character, for example, will be converted from the standard representation in which it appears on the tape into a pattern, as determined by the font being used and by the size of the type. The format codings do not appear in the display on the CRT but result in certain actions, e.g., they serve to note the end of lines, indents of paragraphs, etc.

A page of type is displayed on a CRT and photographed onto film or photographic paper. The photographs of the entire book is then sent to the printer.

For such publications as price lists, bibliographies, indexes, catalogues, membership lists and telephone books the computer has become a near economic necessity. In these books the format remains the same for each publication and only the text requires updating. It is a simple matter to change the text and rerun the program. If the book were done at a typesetter it would have to be completely re-typeset. The computer also has the advantage of speed. It is possible to typeset by computer in a very much shorter time than by metal.

The majority of publications however, are still typeset by hot metal machines turning out lead slugs which are used to make up galleys and locked into position for page makeup.[1,8,9]

2) the Plotter for Animation

A program called VISIT by University of Washington architectural students under David L. Bonsteel, uses the plotter to make an animated film. The film simulates what a driver would see along a proposed road. The film is three minutes long and required 202 individual computer plots of sequential views along the road.

The spacing between plots was determined by the speed limits set for the proposed road. At 50 mph, plots at intervals of slightly less than 50 feet were required, while at the terminal area, driving speed of 15 mph, plots were less than 15 feet apart. These plot distances were based on a film speed of 24 frames per second, with 16 frames per image.

To simulate the feeling of actually being inside an auto while viewing the film, a mask was made of the interior of a car windshield and was placed over the individual plots during animation.[5]

3) the Plotter in traffic

To aid in the analysis of traffic problems the plotter produces street maps where the width of the street is drawn proportional to the traffic density, divided to show traffic in both directions. Traffic flow in all directions at a given intersection can also be plotted when desired, the plotter also prints the exact traffic figures at the street exits. Maps of this sort were prepared in less than 15 minutes.[3]

4) the Plotter in space

Dr. D.D. Morrison of TRW space uses a plotter to draw a new type of map projection for satellite tracking, known as a Kepler projection. The map is designed to do for astronauts what the mercator projection does for sea captains. A ship travelling in a constant compass direction traces a straight line on a mercator projection; using a Kepler projection, the ground track of a satelite becomes a straight line on a map of the earth.[2]

5) The Plotter and golf

A golfing program created by Calcomp provides a view of each hole as it is played and plots the path of the ball for each drive or putt. The player communicates with the computer, selecting any one of 9 clubs and the angle of drive or putt. The computer calculates the distance based on the type of club, wind conditions, and the player's weight. It also keeps track of the score.[4]

Chapter II - Introduction to the Program

The Plotter and Art

The plotter is better suited to executing designs or abstract patterns than it is to executing realistic drawings because the unvarying line that the plotter produces is in keeping with the mechanical effect which is generally desired in a regular pattern. The computer can calculate curves and repeat complex patterns, and for this reason it is extremely useful in making this type of drawing. In fact it would be unthinkable to draw by hand many drawings that have been produced on the plotter. (see page 58).

The plotter is more limited in its uses in representational art. When an artist sets out to make a drawing, such as a landscape, he must make many decisions in order to get an interesting drawing. He views a scene and decides how much of it he would like to include in his drawing and what size paper to use. He then begins to compose his drawing according to what he would like to express about a certain scene. In doing so he can eliminate shatever he likes from the scene or he can add things from some other scene. He can emphasize something in the scene by its position in the drawing, or by its size, or by a heavier line or heavier shading, or by the colours he chooses.

Once an artist has drawn or imagined a particular scene he can get the plotter to repeat what he has done, but I don't believe the plotter could be programmed to do this on its own. It cannot be made to decide what to emphasize in a drawing or how to compose the drawing satisfactorily since those things must be done anew with each drawing and according to a completely subjective interpretation. The plotter can be made to distort a scene in order to achieve an interesting effect, but this would be more a matter of chance than anything else.

The pen line of the plotter, being uniform in its width and heaviness, does not have the interest that an artist can get in line quality. In addition, since the pen can only move in a limited number of directions most lines are made up of a combination of movements in these directions and since the minimum movement is a fraction of an inch, most lines tend to look wobbly when carefully examined. However, when the plotter user is trying to imitate an artists drawing the wobbly line may be more of an advantage than a detriment since it makes a softer and subtler line.

The project I have chosen uses the plotter to draw a realistic landscape. The intention was not to try to create a picture with the beauty of a Dutch landscape, but to try to draw a landscape with some practical value. In any kind of town or country planning it is useful to have different views of the area in order to see the effect of some new development in the area in relation to the landscape. In designing highways, roads, dams, electric power stations, etc., the builders must consider the landscape they are changing. In designing a house it is useful to be able to see the view from all parts of the house and to see the house as part of the surrounding scenery.

The program can easily be expanded to include any variations. It has been written to include, besides the landscape itself, any towns and bodies of water that are present on the map. In addition, the program will draw pylons on the landscape if their coordinates are given.

Objects not included in the map

It was necessary to frame the picture for several reasons. Without a frame the picture looks unfinished and suspended in air. The frame limits the picture. In this program there is a choice of two types of frame. It is a matter of taste as to which should be used. The first one is the simpler (illustration on page 59). One horizontal line is drawn $\frac{1}{2}$ inch below the lowest point in the landscape and another is drawn one inch above the horizon line. These figures were arrived at through trial and error. The decisions were made on a purely visual basis. The two vertical lines of the frame just touch the edges of the landscape. This type of frame accentuates the width of the scene (something like cinerama), and emphasizes the wide angle (120 degrees) chosen for the view.

The second frame is more of an artistic landscape. (see illustration, page 60). The foreground has a tree, 2 nettles, 3 sheep and some grass. In the distance there are birds and clouds. A large tree was placed on the side of the drawing since this is a common device for leading into a landscape. The foreground gives the picture depth. The tree used was inspired by a Breughel drawing, the nettles by a Durer woodcut, and the sheep, birds, grass and clouds are original. These shapes were first drawn on graph paper, and then the coordinates for all the points that make up each shape were taken from this drawing. The points are taken quite close together to get as curved a line as possible. E.g., over 250 points make up the two nettles. The 'X' and 'Y' coordinates for all the points on these items (except the clouds) were stored in an array and drawn with a 'Call Plot (90,X,Y)' within a loop. The clouds are a series of ellipses.

When using this second type of frame , consideration should be given to the lay-out of the landscape, to decide which side has more interest. This is because the frame is reversable. If the more interesting part of the landscape falls on the left side of the paper, then it would be better to place the tree on the right, and vice versa. This helps to balance the picture. In the case of the East Lomond map the eye is immediately drawn to the side which has the peak of the East Lomond itself and so the tree was placed on the opposite side. (See next section for description of the maps).

The Map

Two Ordnance Survey maps were chosen for this project. One is of the East Lomond area in Fife, chosen mainly because it is a fairly hilly area and within close proximity of St. Andrews. The hills make a more interesting landscape. Without hills there would only be a horizon line. The proximity was very helpful in order to be able to visit the area and compare the plotter drawing with the actual landscape. The other map is of Ullswater in the Lake District of England, chosen because of its great variety of scenery within a small area.

The maps are drawn to a scale of $2\frac{1}{2}$ inches to the mile, with contours drawn 25 feet apart and grid lines drawn at one kilometre intervals. These maps have more detail and more contours than the average road map. (On the 1 inch to the mile maps the contours are at 50 foor intervals.) With the $2\frac{1}{2}$ inch map there will be greater accuracy in the landscape drawing.

The only necessary data for this program is a reasonable number of heights of points on the map, evenly spaced, to be read into a matrix. An interval of $\frac{1}{4}$ kilometre was chosen at which to take the readings. This distance between points is close enough to get most of the changes in height. A tracing paper with the appropriate grid lines drawn on it was placed over the map in order to take the heights.

After making several plotter drawings I went to visit the East Lomond to see how close the drawings were to the original. I discovered that the very round shape that characterises the top of the mountain was missing in my plotter drawing which was rather angular. The reason for this disparity is that the contour lines on the map leading to the peak are very close together and the gridpoints drawn at $\frac{1}{4}$ kilometre intervals were not sufficient to get the subtlety of change in height. Thus it was necessary to take an even finer reading of the grid points at the peak. For a square kilometre surrounding the peak points were taken at 1/8 kilometre intervals. This gave the desired effect.

The reading of the heights is started at the top left corner of the map and moves across in columns, always from top to bottom. The map heights are read into a 50 x 50 matrix. It is necessary to indicate to the program the number of grid lines in either direction and the distance between each grid line. Any grid point that falls on water is indicated with a minus sign in front of the height. The coordinates of any towns or cities are read in separately. Any additional features such as pylons are also read in separately.

After putting the heights of all the grid points on punch cards it is possible to check the data for any serious errors by using a subrouting written by Professor Cole called JCCONT.[6] JCCONT draws a contour map on the plotter using these map heights given as data. If there are any serious errors in the data, they will show up in this plot. The plot should look very much like the original map. A common error, such as a number punched in the wrong column will show up quite dramatically with several contour lines very close together where there is no sudden peak or valley on the map (see illustration page 61).

Any point on the map is referred to by its 'Y' value, counting from the top of the map, and then by its 'X' value counting from the left side of the map.

The map is still required for one last step. Two points must be selected as follows: The first point is the "Observation point" (point A on Figure 3). This is where the imaginary observer stands to view the landscape. The second point (point B on Figure 3) is called the "view point". These two points form the left radius of a 120 degree arc which is the "Field of Vision". It is from within this field of vision that the landscape will be found. Any points outside will not be considered.

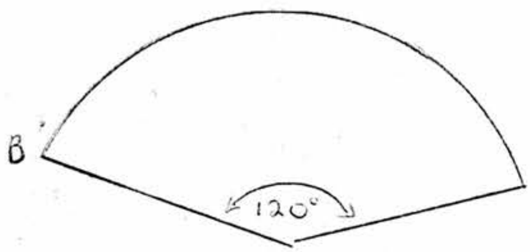


Figure 3

There should be a reasonably large distance between the observation point and the view point (eight kilometres should be sufficient in most cases). The reason for this is that the view point is the limit of the arc of vision. If too small a distance is chosen there is the danger that data which is behind the view point which would affect the landscape will be neglected. Possibly a high peak in the distance which is actually visible from the observation point will not be drawn on the landscape.

Chapter III

General description of the program

In order to be able to get the computer to draw a landscape, it was first necessary for me to analyse the components of a landscape; which points are seen and which are hidden, what makes one hill appear closer than another, what scale factor should be used.

How does an artist represent a landscape? He must first decide on the size of paper to be used. Next, the artist must, in any order he likes, draw the outline or horizon line, the foreground, and whatever occurs between the foreground and outline. The foreground could be some trees or shrubs within a few feet of where he is sitting. The outline would be the land or water line that meets the sky. It could be a line of hills in the distance or the horizon of the sea, or both.

Working from the top down, the next group of lines will be the continuation of any hills that are on the outline. It is these lines that will indicate which hill is in front of which. The next group of lines will be the middle ground, if it is present. This could consist of any lesser hills or any bodies of water that are in front of those hill that make up the outline.

The plotter program works in a similar manner to the artist. The size of the plot must first be decided. The width chosen for the plot was 12 inches, and the length varies with each landscape. The frame and foreground are plotted first. The foreground is necessarily hypothetical, since it would be impossible to take this information from a map. It next plots the outline and then whatever is between the outline and the foreground. Finally, it puts in the finishing touches by plotting any lakes or towns that are in the view.

The program attempts to look at as many points as possible within the field of vision to see which will form the lines that make up the landscape. It works in the following manner:

The 120 degree arc, which is the field of vision, is broken up into segments by 240 radiating lines, which are separated from one another by 0.5 degrees. Each radiating line is evenly divided into 80 parts. The program inspects all 80 points along the first radiating line, which is formed by the observation point and the view point, in order to find those points which will be part of the landscape. It then calculates the coordinates of a point 0.5 degrees to the right of the view point and this new point with the observation point will form the next radiating line. This line will be inspected for points which will belong to the landscape in the same manner as was the first line. This process goes on 240 times until 120 degrees have been covered.

Each radiating line is inspected, first of all, to see which point on it will appear highest to the observer. This is not necessarily the highest point. To determine, for example, if a mountain which is 1000 feet high and 2 miles away, would appear higher than a mountain which is 2000 feet high and 3 miles away, it would be necessary to calculate the angle of elevation each mountain top makes with the observer. The angle of elevation tells how many degrees the eye of the observer must rotate from the horizontal in order to see the top of an object. The mountain with the larger angle of elevation will appear the higher to the observer.

Every point within this 120 degree arc is the vertex of an angle (angle BAC in figure 4). AB is the line between the observation point and any point within the arc and AC is a tangent from the observation point to the earth (at sea level). Angle BAC is called the angle of elevation.

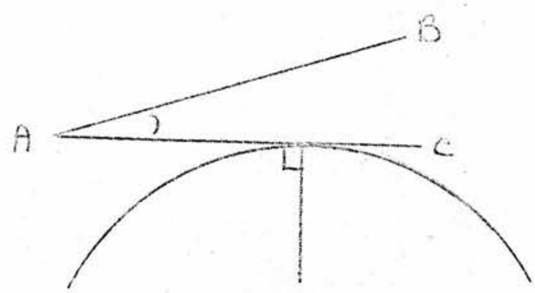


Figure 4

When an object in the distance appears to be the highest object it will have the largest angle of elevation within the field of vision. If the point with the largest angle of elevation is taken from each of the 240 radiating lines and these 240 points are projected onto a spherical screen and then drawn on the plotter, we will have the outline of a view. This outline will be referred to as "outline 1". (see illustration page 62). Nothing can be seen beyond this outline from the observation point.

After calculating outline 1 the program looks for any points which will belong to lines in front of outline 1. If the outline consists, for instance, of two hills (figure 5a), then there should be a line attached to the outline indicating which hill is in front of which (figure 5b). If this is not the case, the line just below the outline could be on a hill belonging to the middle ground (figure 5c).

Figure 5a

Figure 5b

Figure 5c

In order to construct any lines below outline 1, the second largest angle of elevation on each radiating line (considering only those points between outline 1 and the observation point) is found. Generally this second largest angle of elevation is of a point just below one of the peaks of outline 1 (see figure 6). If this is the case, this second largest angle of elevation does not appear as a line in the landscape drawing since it is part of the same slope indicated by the largest angle of elevation. In order for a point to be part of a line in a landscape, it must belong to a different slope from the point just above it which belongs to outline 1 (figures 5b,5c). If the second largest angle of elevation belongs to the same slope as the largest angle of elevation, a third largest angle of elevation is found and tested to see if it belongs to a different slope. In the case of figure 6, the fourth largest angle of elevation would be found to belong to a separate slope and this point would form part of the line below outline 1, called 'outline 2'.

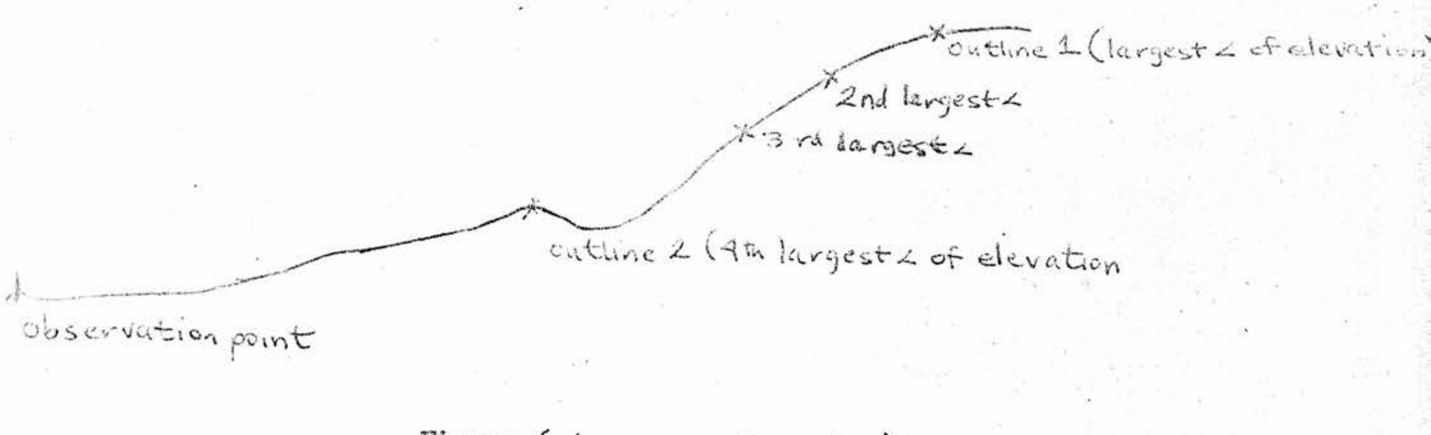


Figure 6 (cross section view)

If no angle of elevation is found that belongs to a separate slope from the outline, the observation point is finally reached, and in this case no point on this radiating line will belong to outline 2. Points belonging to outline 2 are looked for along each radiating line. It will probably be present on some of the radiating lines and not on others.

The program proceeds in this manner, looking at points between outline 2 and the observation point for the next largest angle of elevation which will be part of outline 3 if it belongs to a separate from the point in outline 2. If the observation point is reached before any point on a separate slope then that particular radiating line has all its outlines calculated and the next radiating line is inspected and so on until the field of vision is completed. The program allows for five possible outlines on any one radiating line. This is sufficient to take care of all slopes on these maps.

When all the outlines are selected from the field of vision they are ready to be plotted. Before any point can be plotted, its angle of elevation must be translated into a height in inches so that it may fit on the plotter paper. This is done by projecting the angles of elevation of those points in the outlines onto a sphere.

The plot width of 12 inches will represent the 120 degrees of the field of vision. Therefore each inch represents 10 degrees, and one degree is one-tenth of an inch. This same scale is used for the height of the plot. That is, for every degree of angle of elevation, one-tenth of an inch is added to the height. This method works as if a grid were placed over a sphere.

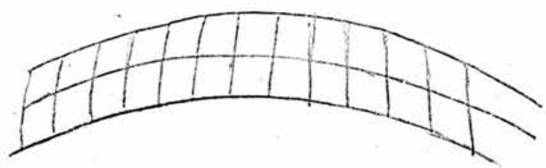


Figure 7

The actual results using this method were not acceptable. (see illustration page 63). The East Lomond, although not very high, is certainly more impressive than this. The scale would have to be exaggerated in the vertical in order to show the mountain as the eye sees it. Once again, the best solution to this problem was to visit the area and make some sketches of the East Lomond from various places on the map. Only when the vertical was drawn three times as large as the horizontal did it look like the sketches I made on my visit. Therefore, every degree in the horizontal would represent three-hundredths of an inch in the vertical. These vertical values are the 'Y' axis on the plotter. The 'X' axis has 240 steps, for the 240 radiating lines, in 12 inches, which means every increase of X by 1 represents one twentieth of an inch.

There is no difficulty in drawing outline 1 because all the points are connected. This is how it appears to the eye. However, in all the other outlines, the points are connected only when they belong to the same slope or when one slope passes directly in front of or behind another slope. This is described in more detail on page 46.

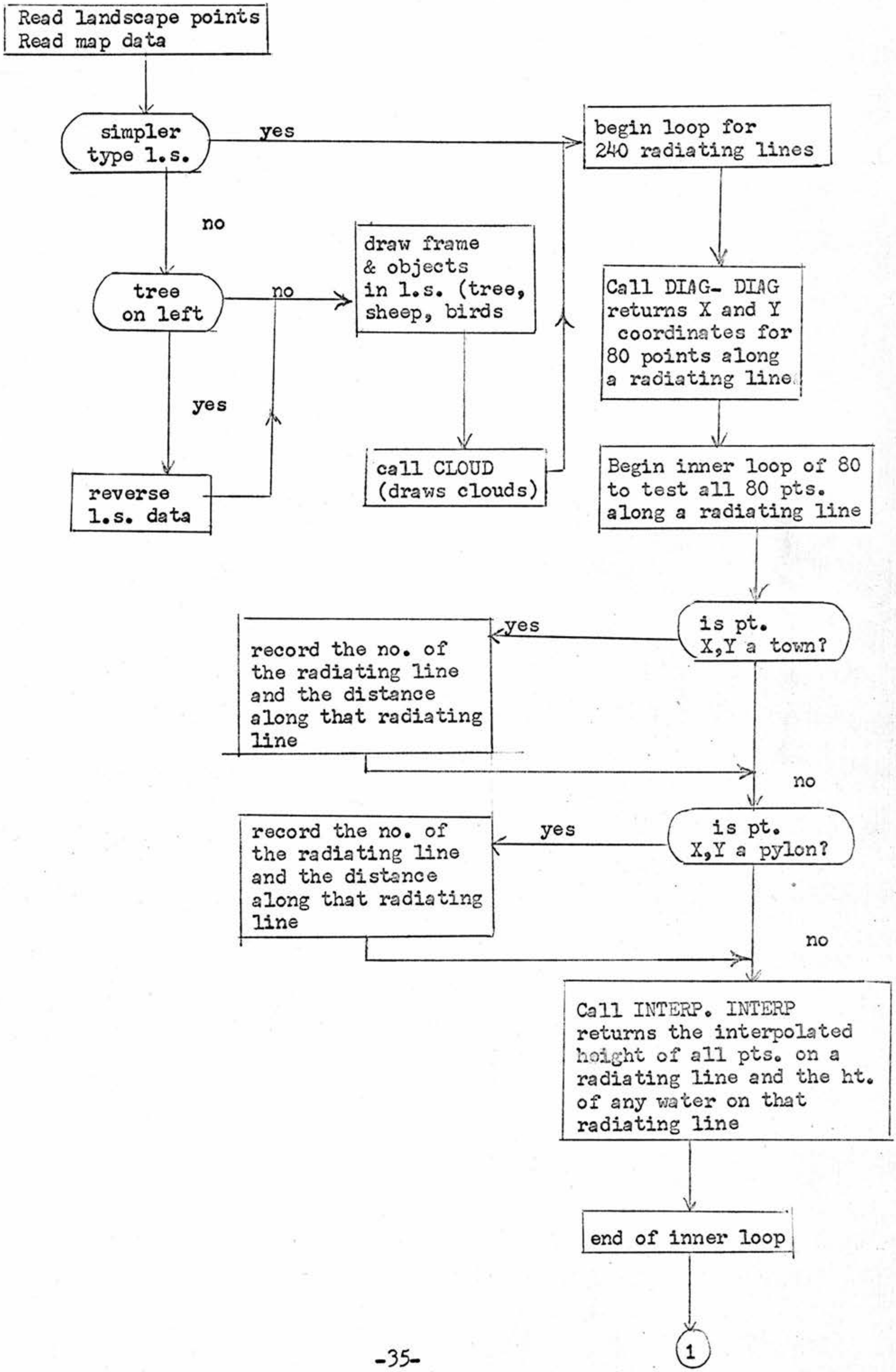
At the same time as the program is searching for points that will be in the landscape, it is also looking for any bodies of water in the field of vision. In order to be able to see a body of water the observer must be standing on higher ground than the water. This means that when the water is in the distance, it will only be considered if it has a negative angle of elevation.

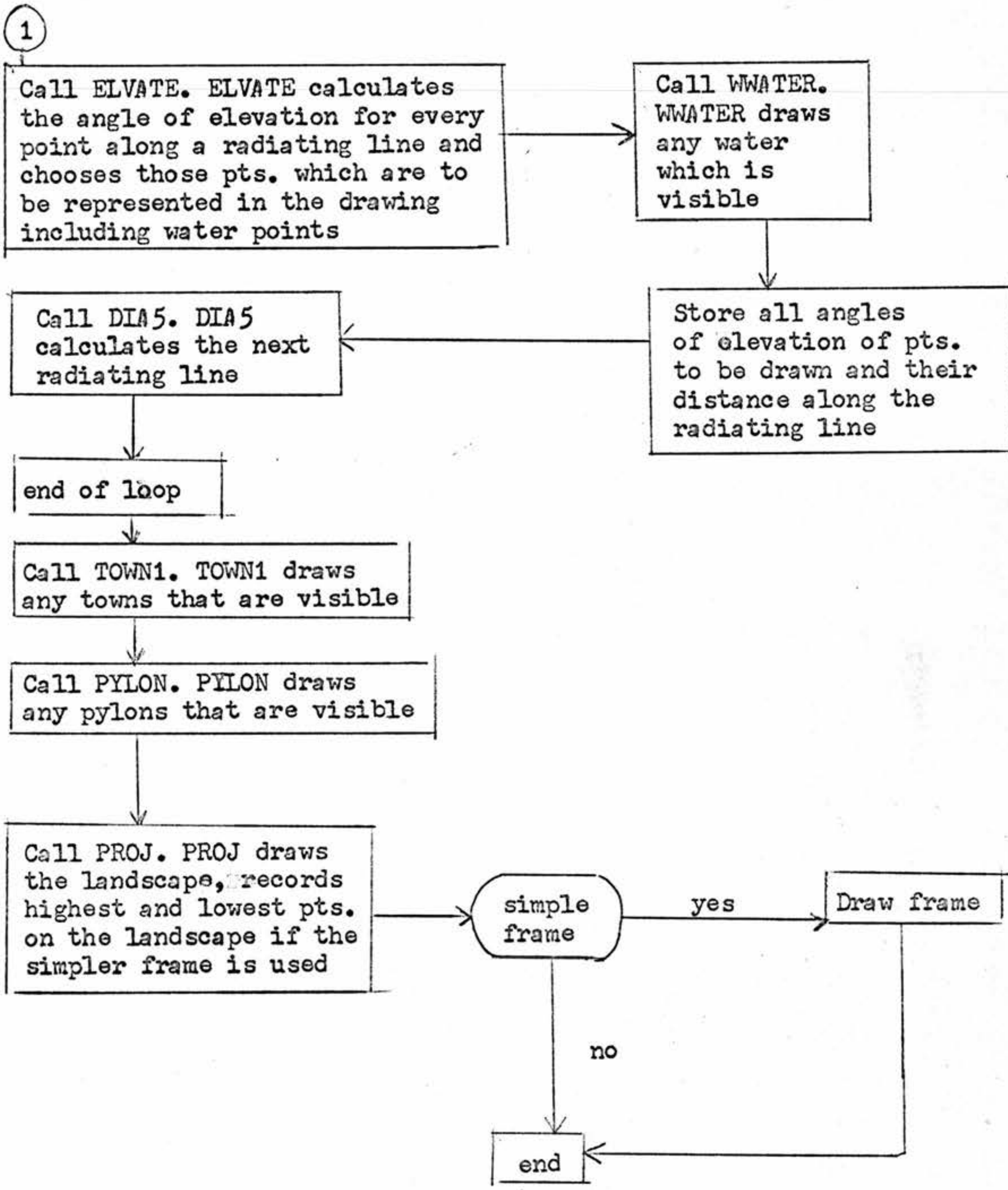
As the program moves through the 240 radiating lines a minus sign is looked for in front of any heights, which would indicate the presence of water. The angle of elevation of the water is calculated and tested to see if it is negative. if it is, the program then tests to see if the water is visible from the observation point. If there is a hill in front of the water with an angle of elevation higher than that of the water, then the water is not visible to the observer and is not drawn. When it is visible, it is represented with a diagonal line drawn from one side of the water to the other.

Towns are handled in a similar way to bodies of water. For any town whose coordinates are within the field of vision, the angle of elevation is calculated. If there is no point with a larger angle of elevation in front of this town, the town is visible and therefore drawn. A symbol for a town was devised consisting of over 350 points. The size of the drawing of the town depends on the distance of the town from the observation point. It is drawn about three-quarters of an inch wide when it is within a quarter kilometre from the observer and about one-tenth of an inch when about 20,000 feet away.

Pylons are handled exactly the same as towns. As with towns their size depends on their distance from the observer. The pylons are connected with a parabola if they are next to each other on the map. The wires between two pylons actually form a catenary, but, for practical purposes, a parabolic arc is sufficient.

If the frame being used is of the second type, with the foreground tree, every point must be tested to determine whether it is hidden behind the tree. If it is hidden the plotter pen is lifted and the point is not drawn. An imaginary rectangle is drawn around the tree which just touches the outer limits of the tree trunk. At first a test is made to see if the point to be plotted falls within this rectangle. If it does, then a more careful check is carried out to see if this point is actually hidden behind the tree. This means testing the point to be plotted against the outline of the tree. This method of testing saves computer time since to test every point drawn against every point on the tree would be very time consuming. Using this method the test is only made when the point is inside the rectangle.


Read landscape points
Read map data
simpler
type l.s.
yes
no
begin loop for
240 radiating lines
tree
on left
no
draw frame
& objects
in l.s. (tree,
sheep, birds
yes
Call DIAG- DIAG
returns X and Y
coordinates for
80 points along
a radiating line
reverse
l.s. data
call CLOUD
(draws clouds)
Begin inner loop of 80
to test all 80 pts.
along a radiating line
yes
is pt.
X,Y a town?
record the no. of
the radiating line
and the distance
along that radiating
line
no
yes
is pt.
X,Y a pylon?
record the no. of
the radiating line
and the distance
along that radiating
line
no
Call INTERP. INTERP
returns the interpolated
height of all pts. on a
radiating line and the ht.
of any water on that
radiating line
end of inner loop
1


1
Call ELVATE. ELVATE calculates the angle of elevation for every point along a radiating line and chooses those pts. which are to be represented in the drawing including water points
Call WWATER. WWATER draws any water which is visible
Store all angles of elevation of pts. to be drawn and their distance along the radiating line
Call DIA5. DIA5 calculates the next radiating line
end of loop
Call TOWN1. TOWN1 draws any towns that are visible
Call PYLON. PYLON draws any pylons that are visible
Call PROJ. PROJ draws the landscape, records highest and lowest pts. on the landscape if the simpler frame is used
simple frame
yes
Draw frame
no
end

Detailed description of the program

The program is broken up into 10 subroutines and one main program. The purpose of the main program is to call and coordinate these subroutines. Before any subroutines are called, the plot is scaled, the heights are read in together with any data for towns and pylons, and the necessary data for the foreground is read in. The frame and the foreground are drawn immediately only if the foreground chosen is of the second type - the more elaborate one. If the simpler foreground is chosen this will be drawn at the very end.

The first five subroutines (DIAG, INTERP, ELVATE, WWATER & DIA5) are called in a loop which is repeated 240 times, once for each radiating line.

DIAG divides the first radiating line (connecting the observation point and the view point) by 80, resulting in 80 points along this line. The 'X' and 'Y' value of each point is stored. Each point is given a 'distance number' according to its position along the line. The furthest point (in this case the view point) has a distance number of 1, and the closest point has a distance number of 79 (point 80 is always the observation point). The coordinates of the point with a distance number of 79 are found as follows: 78/80ths of the 'Y' coordinate of the view point is added to 2/80ths of the 'Y' coordinate of the observation point to get the 'Y' coordinate of point 79, and 78/80ths of the

'X' coordinate of the view point is added to 2/80ths of the 'X' coordinate of the observation point to get the 'X' coordinate of point 79. To get the coordinates of the point with distance number 78, 77/80ths of the view point and 3/80ths of the observation point are taken and the same procedure is followed.

The distance between the observation point and each of the 80 points along the radiating line is calculated in kilometres using the Pythagorean theorem.

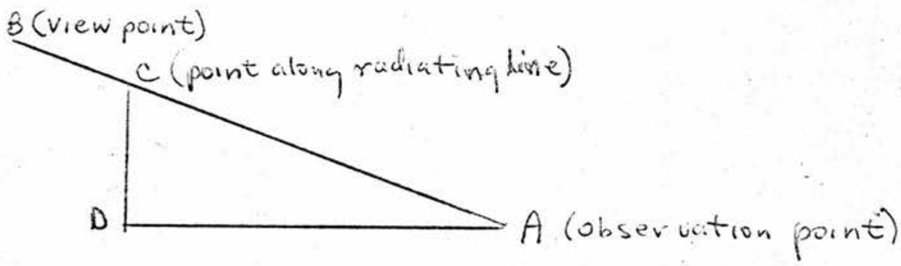


Figure 8

AD is equal to the 'X' grid point value of the observation point, minus the 'X' grid point value of C. CD is equal to the 'Y' grid point value of C, minus the 'Y' grid point value of A. The value calculated for AC will be in $\frac{1}{4}$ kilometres since the grid points are given in $\frac{1}{4}$ kilometre intervals, therefore AC is multiplied by .25 to get its value in kilometres. This distance is stored and will be used in calculating the angle of elevation.

It is possible that some of these points calculated will be off the map, since it is not necessary for the field of vision to be entirely on the map. Any points that are off the map will be ignored.

When DIAG divides the line by 80, most of the resulting points will not fall exactly on a grid point. (see figure 9a). Since the map heights are given only at grid points, the height at each of the 80 points along the line must be interpolated using the heights of the four closest grid points (points X1,X2, X3,X4 in figure 9b). This is done in subroutine INTERP. The formula for interpolation* used here is:

$$F = R3 \times S3 \times X1 + R \times S3 \times X2 + S \times R3 \times X3 + R \times S \times X4$$

where F is the height of a point found by interpolation

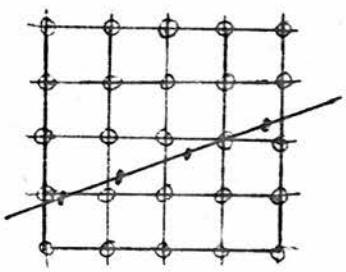


Figure 9a

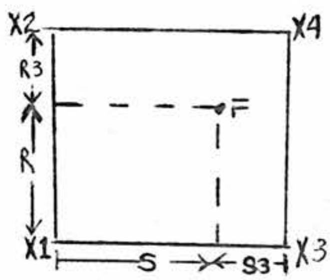


Figure 9b

o points where the heights are known

• points where the heights must be found by interpolation

If any of the four surrounding heights used in this formula are negative (indicating water) then a test is made to see if the point is in that quarter of the square closest to the negative height. If it is, then that point is stored as a water point and the height of the water is the negative grid point

* The distances R, R3, S, S3 should be normalised so that R+R3 = S+S3 = 1

closest to it. The height of the water is not interpolated since water is always the lowest point of the four.

As this subroutine makes its calculations along the line it also tests to see if any of the points belong to that square kilometre surrounding the peak of the East Lomond which was given a finer reading with intervals at 1/8th kilometre rather than at $\frac{1}{4}$ kilometre, as was the rest of the map. When a point is discovered which is in this square surrounding the peak, the interpolation is done substituting the matrix with the closer heights.

It is now possible, in subroutine ELVATE, using the distance of each point from the observation point and the height of each point, to calculate the angle of elevation to each point along this radiating line. The angle of elevation can be positive or negative depending on whether the observation point is higher or lower than the view. If it is lower it is an angle of depression. However, the calculation is done in exactly the same way, so I will only refer to the angle of elevation to describe it.

The earth's curve was taken into consideration in the calculation of the angle of elevation. This curve does not affect points close to the observation point, but can make a significant difference with points in the distance.

Double precision was used for this entire subroutine because more significant figures were needed in order to determine diferrences in values close to one another. All distances in kilometres are converted into feet before being used in the calculations.

The formulas used are:

$$1)\ X = \sqrt{(R + obs)^2 + (R + view)^2 - 2(R + obs)(R + view)(\cos \frac{D}{R})}$$

$$2)\ \theta = \text{arcsine } \left(\frac{R + view}{X}\right)\left(\text{sine } \frac{D}{R}\right) - \text{arcsine } \left(\frac{R}{R + obs}\right)$$

where X is the straight line distance from the observation point to the view point, and R is the radius of the earth. Obs is the height of the observer, view is the height of the view point and D is the map distance along the surface of the earth.

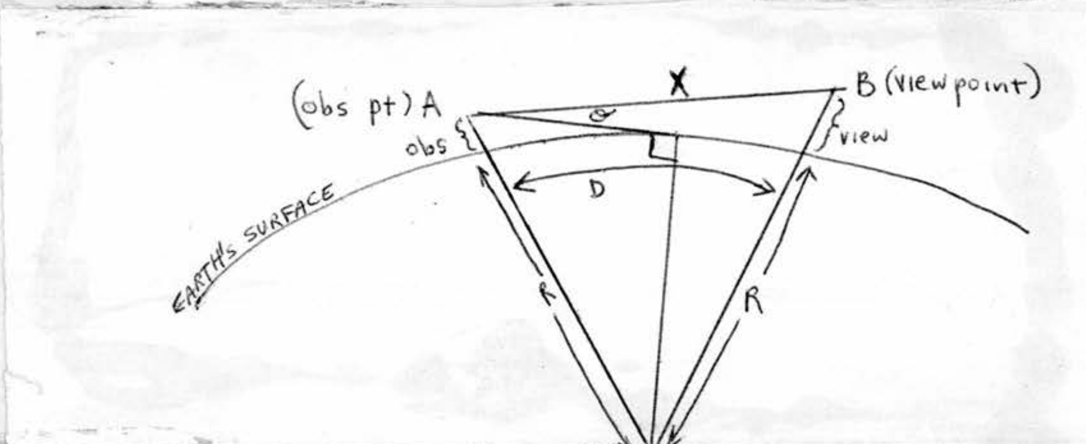


Figure 10

The resulting angle is in radians since the subroutines for the various trigonometric functions have their arguments in radians. This is later converted into degrees.

When all the angles have been calculated for one radiating line, those angles that represent the tops of slopes will be selected to be drawn. As already mentioned, the very largest angle of elevation on this radiating line will become a point on the highest outline, outline 1. For the next outline a test must be made on the next largest angle of elevation to see whether it is a separate slope. This is done in the following way: All 80 angles of elevation for each radiating line are stored in an array. The first angle in the array is the farthest from the observation point. The largest angle is found for outline 1. If this is, for example, the eighth angle in the array, the search for the next largest angle begins with the ninth angle in the array. If the next largest angle turns out to be the ninth angle this is not a separate slope since it is on the same slope as outline 1. If the next largest slope after the ninth is the tenth, this is also not a separate slope. In order for an angle to be on a separate slope, there must be at least one point with a smaller angle of elevation in between the two larger ones. This same process is repeated until the largest angle of elevation is the 79th. The number of the angle along the radiating line is called the distance number. The distance number for each angle which is on an outline is saved for later use in the subroutine 'PROJ'.

The angle of elevation of any water within the field of vision is calculated in this subroutine. If the angle is

Note: in actual practice, only 77 points along each radiating line are inspected for largest angle of elevation. It was found that points closest to the observation point gave an artificially high angle of elevation.

positive it is not considered further since the observer must be above the water to be able to see it. If the angle is negative it is tested to determine if there is some hill blocking it from the observer. The water is given a number from 1 to 79, a distance number, depending on its position along the radiating line. This number corresponds to the angles of elevation in the previous paragraph. If the water is at number 60 along the radiating line there must be no angle of elevation larger than that of the water in front of the water (with a number greater than 60).

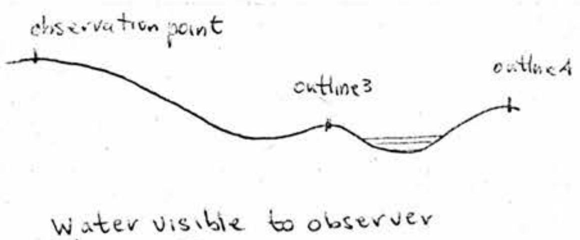


Figure 11a

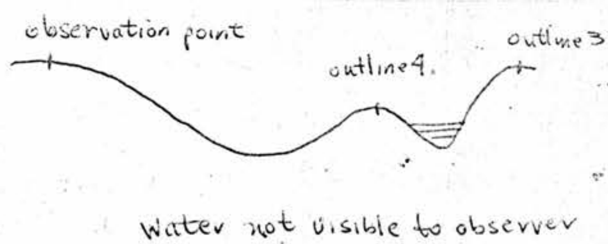


Figure 11b

If it is farther than outline 1, that is, its distance number is less than that of outline 1, it cannot be seen. If it is the same distance as any of the outlines, it is visible. If it is neither of these it must be somewhere in between two outlines. If, for example, it is behind outline 4 and in front of outline 3, then in order for the water to be visible its angle of elevation must be larger than the angle of elevation of outline 4.

If water is visible it is drawn right away instead of being saved up until the entire arc of vision is gone through, as are the outlines. This is because the water varies greatly from one map to another and it would call for too large an array to hold all the possibilities.

Subroutine WWATER actually draws any water that is visible on that one radiating line which is in the loop. This subroutine is not called very often. If the frame is of the first type, the 'X' value of the water is tested to see if it is hidden by the tree. If it is, the water is not drawn. (See page 52 for method of testing.) The only 'Y' values that are needed for a certain 'X' value are those for the farther shore and the closer shore. A line is drawn starting at the farther shore from the observer and connected to the closer shore. The pen is not lifted so this gives a zigzagged effect. The 'X' value is the number of the radiating line. The 'Y' value will be the angle of elevation of the water multiplied by .3 (the amount by which the vertical is exaggerated). Five inches is added to this result to place the drawing in the middle of the page. If the water occurs at a certain point along the 'X' axis the pen is first drawn from 'X' to 'X+1' using the 'Y' value of the farther shore (it moves 1/20th of an inch along the 'X' axis.) Then the pen is drawn to the closer shore with the same 'X' value as the far shore, and the 'Y' value for the closer shore. It is drawn in this manner to get a diagonal line. (See illustration page 62).

The first time the program goes through the loop, subroutine DIA5 calculates the point 0.5 degrees to the right of the view point. This point becomes the new view point and forms a new radiating line with the observation point. This new point is found using the rotation formulae:

1) $X = X'\cos\theta - Y'\sin\theta$

2) $Y = X'\sin\theta + Y'\cos\theta$

This procedure continues until 120 degrees have been covered and then the loop for the 240 radiating lines is ended. All the points in the landscape have been selected and are now ready to be plotted.

The subroutine PROJ contains the plotting instructions. As with water, to arrive at the 'Y' value, each angle of elevation is multiplied by .3 and then added to 5 inches to place it in the middle of the page. The 'X' value is the number of the radiating line. All of outline 1 is plotted in one continuous line from left to right, from X = 1 (the first radiating line), to X = 240 (the 240th radiating line), with the corresponding Y value.

The next four outlines are not continuous. The points will belong to different slopes and the pen must be lifted to show this. (See figure 12). Before any point is drawn on the next four outlines several tests must be made (see flow chart page 47). These tests serve to close most gaps in the outlines, to connect any points in outlines 2-5 to outline 1 when required and to separate the different slopes. These tests use only the distance number. Outline 2 is completed before outline 3 is begun. The following algorithm uses outline 2, radiating line number 100 for example (which means X = 100). This is the 100th place in the array outline 2.

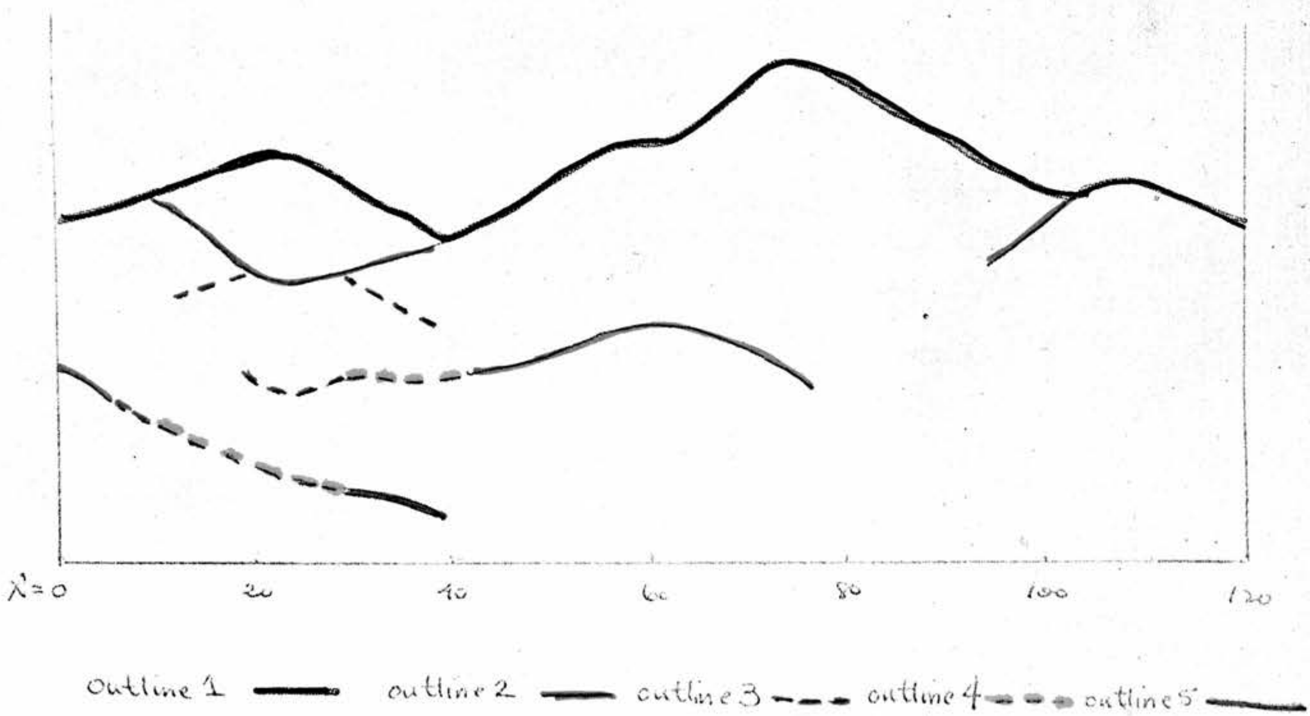


Figure 13a

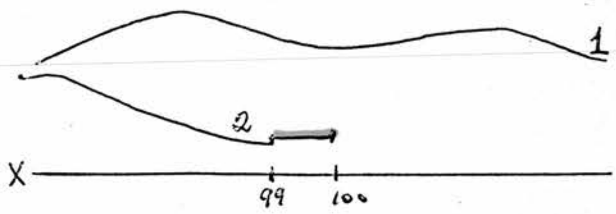


Figure 13b

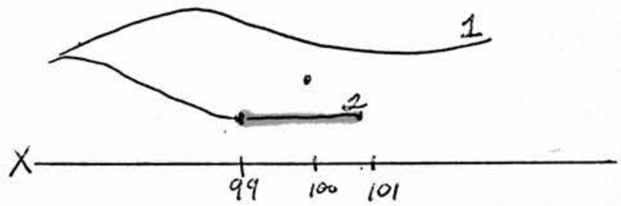


Figure 13c

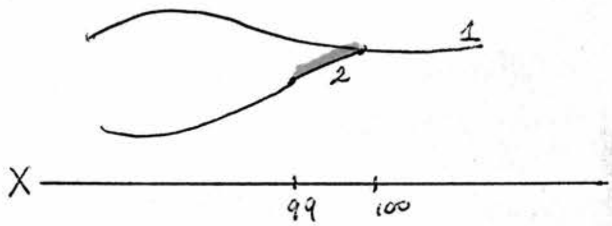


Figure 13d

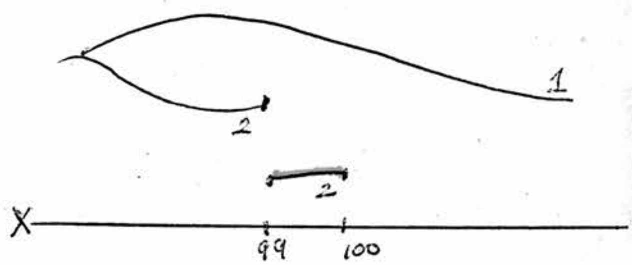


Figure 13e

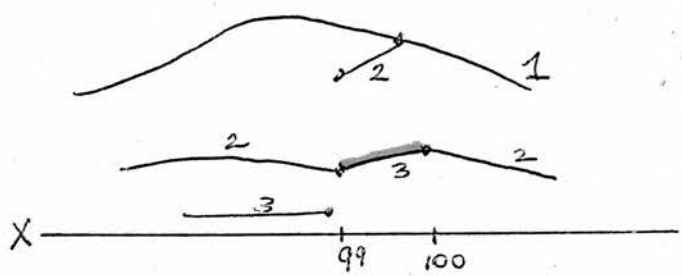
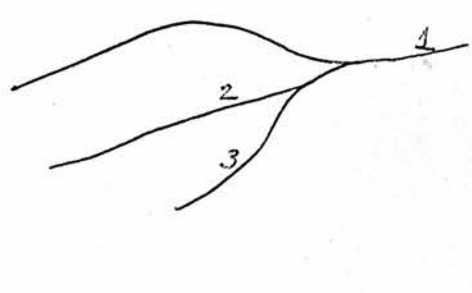
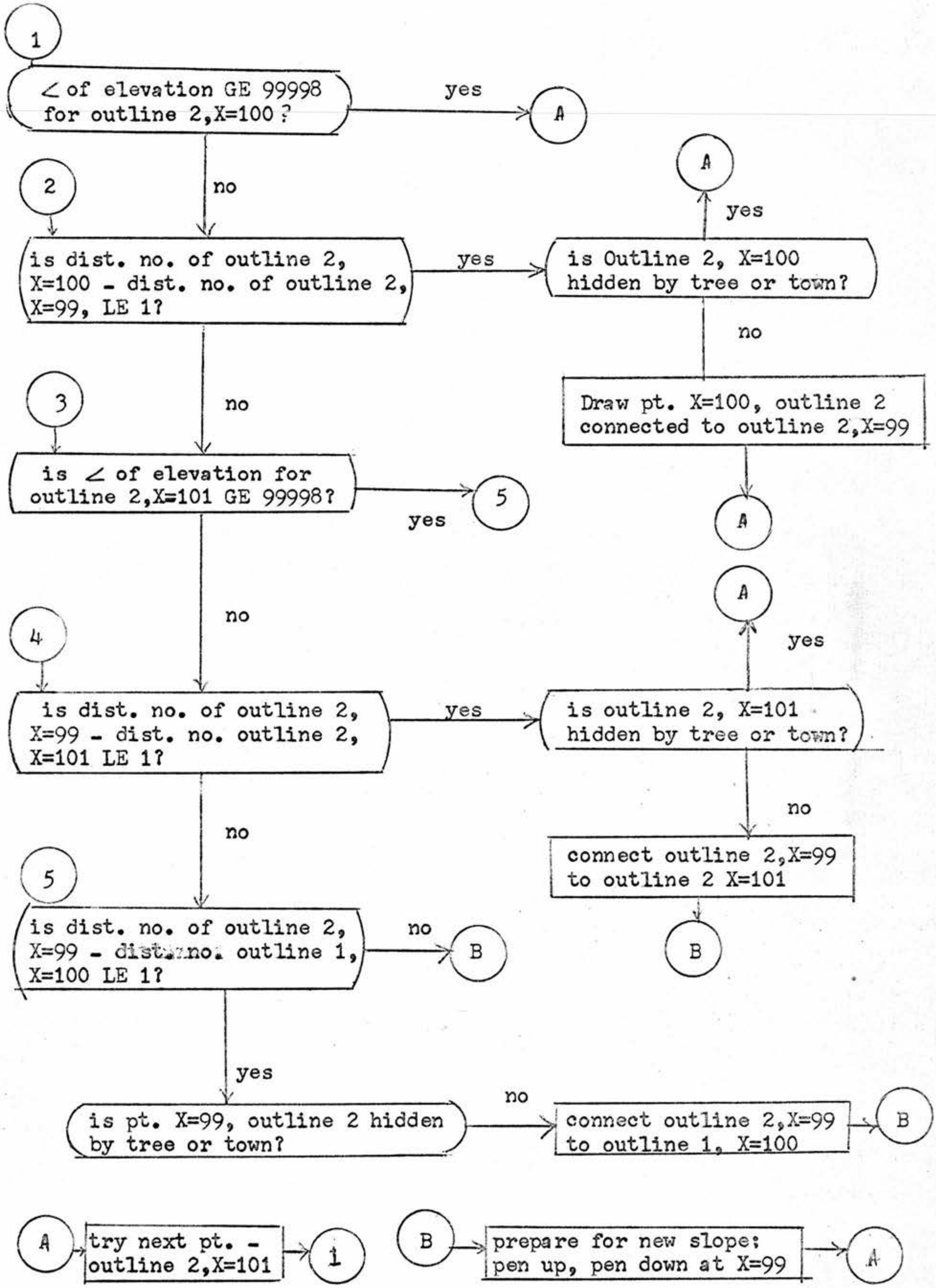


Figure 13f



Note: "1" indicates outline 1, "2" indicates outline 2, and "3" indicates outline 3.


1
∠ of elevation GE 99998 for outline 2,X=100 ?
yes
A
no
2
is dist. no. of outline 2, X=100 - dist. no. of outline 2, X=99, LE 1?
yes
is Outline 2, X=100 hidden by tree or town?
yes
A
no
Draw pt. X=100, outline 2 connected to outline 2,X=99
A
no
3
is ∠ of elevation for outline 2,X=101 GE 99998?
yes
5
no
4
is dist. no. of outline 2, X=99 - dist. no. outline 2, X=101 LE 1?
yes
is outline 2, X=101 hidden by tree or town?
yes
A
no
connect outline 2,X=99 to outline 2 X=101
B
no
5
is dist. no. of outline 2, X=99 - dist. no. outline 1, X=100 LE 1?
no
B
yes
is pt. X=99, outline 2 hidden by tree or town?
no
connect outline 2,X=99 to outline 1, X=100
B
A
try next pt. - outline 2,X=101
1
B
prepare for new slope; pen up, pen down at X=99
A

1) Is there an angle of elevation present for outline 2, radiating line 100?

Here, the test is made for an angle of elevation greater than 99998. The reason for this lies in subroutine ELVATE, which stores the angles of elevation for all the outlines. If there was no angle of elevation present on a particular outline for a particular radiating line, the number 99999 was written instead.

If there is no angle, move on to radiating line number 101 on outline 2.

2) Is the distance number of this point the same as the distance number of the point just before it in the same outline, or is the absolute value of the difference between them not greater than 1?

The present point is on outline 2, radiating line 100, and the point before would be on outline 2, radiating line 99. If the distance number of one is 17, and the distance number of the other is 16, these two points will be connected: X = 99 to X = 100 with the corresponding Y values. (see figure 13a). The next point on outline 2 is now taken (radiating line 101). Go to 1).

3) If 2) is not true, the point on outline 2, radiating line 100 belongs to a different slope from outline 2, radiating line 99. A test is now made to see if there is an angle of

elevation present on this outline for the point on outline 2, radiating line 101.

Is the angle of elevation for outline 2, radiating line 101 greater than 99998?

The present point is not connected to the point before it, so it is possible that the point before it is connected to the point after it. To avoid a gap, which would result if the pen were raised at X = 99 and lowered at X = 100, and then raised at X = 100 and lowered at X = 101 (see figure 13b), tests 3) and 4) are made.

If there is an angle of elevation on outline 2 radiating line 101, go to 4), otherwise go to 5).

4) Is the distance number for the point on outline 2 radiating line 99, the same as the distance number of outline 2 radiating line 101, or is the absolute value of the difference between them not greater than 1?

If the result is yes, the points on outline 2, X = 99 and outline 2, X = 101 are connected (figure 13b). The pen is now lifted and brought down at the point X = 99, Y = the Y value calculated for X = 100. The pen is brought down at X = 99 rather than X = 100 in case outline 2 is working with 2 slopes and alternating between the two of them (see figure 13e). In this example, if the pen was not brought down at X = 99, there would be a gap where there is a dotted line drawn. The pen is now ready to plot on a different slope. The next point on this outline is taken (101).

Go to 1).

5) Is the point on outline 2, radiating line 99 connected to outline 1?

Since the present point is not connected to the one before it, it is possible that the one before it is connected to outline 1 and the present point begins a new slope. This test is important since the drawing would be ambiguous if it were not determined which slopes outline 1 belongs to (figure 13f). The test is made on the distance number of outline 1 radiating line 100 with the distance number of outline 2 radiating line 99. If the two are the same or have an absolute difference not greater than 1, the two points are connected (figure 13c). The pen is lifted and brought down at X = 99, Y = the Y value calculated for X = 100 (the same reason as 4). The next point on outline 2 is looked at (101).

Go to 1).

If none of the above types are found the point is assumed to belong to a new slope and the pen is lifted and brought down at the point X = 99, Y = the Y value calculated for X = 100. (see figure 13d).

In addition to all the previous tests, there is one last test to be made before each point is drawn. This is done for every point on all five outlines, and for any water point. This is the test to see if a point is hidden behind a tree or town, when these are present. As mentioned on page 33 this is done by first testing to see if the point is inside an imaginary rectangle drawn around the tree or town. If it is, a more exact comparison is made.

1) If the frame is of the second type with the tree on one side of the landscape, the point must be tested to see if it is between X = 207 and X = 238, since these are the horizontal limits of the tree trunk. If it is, the subroutine TREE is called which makes a more careful check on the exact coordinates of the point.

2) In both frames, a test is made to see if a town is blocking the landscape. The limits of the town, if one is present in this landscape, are recorded in the subroutine TOWN1 and each point is tested against those limits.

3) In the first type of frame, the simpler one, the highest and lowest Y values of the drawing must be determined. A test is made when drawing each point in outlines 2-5 for the lowest point. 1 is added to the highest point to get the Y value of the top of the frame and 0.5 is subtracted from the Y value of the lowest point to get the Y value of the bottom of the frame.

Chapter IV

Further Work

There are many things that could be done with this program to make it more elaborate, more artistic and more varied. The landscape could be shaded, have coloured outlines, have more trees, people and buildings.

It would be interesting to experiment with different pens such as the felt tip, and to try different colour inks. By forcing the computer to pause at various places in the program, the pens could be changed so that the colours used could match the colours in nature. E.g., the water could be blue and the grass green, etc.

At some point during this project I tried to shade the drawing but after a few attempts I realised that this would take enough work to become a separate project. One method of shading would be structural shading, that is, shading according to the shape of the terrain, regardless of the position of the sun. The other method would be to decide on the time of day and shade according to where the sun would cast shadows.

The program is written to vary the size of the town according to its distance from the observer - the farther it is, the smaller it is drawn. In addition to this, it could vary in size according to its population or area. Possibly, one symbol could be used for a town and another for a city. This symbol could be repeated or added to in order to show greater size.

It should be possible for the user to add his own symbols if this is desired, in order to adapt the program for his particular needs. He might want to show a building, a park, a monument or a dam.

Probably the most useful thing that could be done at this point would be to generalize the program. As it stands, the program can be used with one type of input and results in, basically, one type of output. The user should have more freedom with this program. He should not have to arrange his input in such a specific format and should have more to say about what his final drawing will look like.

To begin with, this program was written for a very particular type of input. Heights are taken from a map scaled at $2\frac{1}{2}$ inches to the mile at intervals of $\frac{1}{4}$ kilometre. The data is held in a 50 x 50 matrix.

The program could be generalized in order to allow for various types of input, e.g., maps drawn to any scale and heights taken at intervals other than $\frac{1}{4}$ kilometre. It could be altered with the addition of another subroutine, to include heights chosen at random. This would save the trouble of manually finding the height of every point needed as data. It would also make it possible to do without a map altogether and use the heights of points from a surveyor's notebook. To do this it would be necessary to include a subroutine to calculate the heights of those points needed to fill the matrix that are not provided by the user. This could be done by interpolation, as in a program by Prof. Cole (6).

References

1. P.L. Anderson, "Phototypesetting - A quiet revolution", Datamation, Dec. 1970.

2. Calcomp Newsletter, August/September 1964.

3. Calcomp Newsletter, July/August 1966.

4. Calcomp Newsletter, May/June 1969.

5. Calcomp Newsletter, September/October 1970.

6. A.J. Cole, C. Jordan, & D.F. Merriam, State Geological Survey, the University of Kansas, Lawrence (Kansas).

7. Samuel Davis, "A Technical survey of digital Plotters", Proceedings of the Society for Information Display, Vol. 11, No. 1.

8. Edwin R. Kolb "Forecast for '70s, Datamation, Dec. 1970.

9. Henry D. Sedgwick "Goodby Hot Metal, Hello Cool Tape", Datamation, Dec. 1970.

10. Ivan Sutherland "Computer Displays", Scientific American, June 1970, Vol. 222, No. 6.

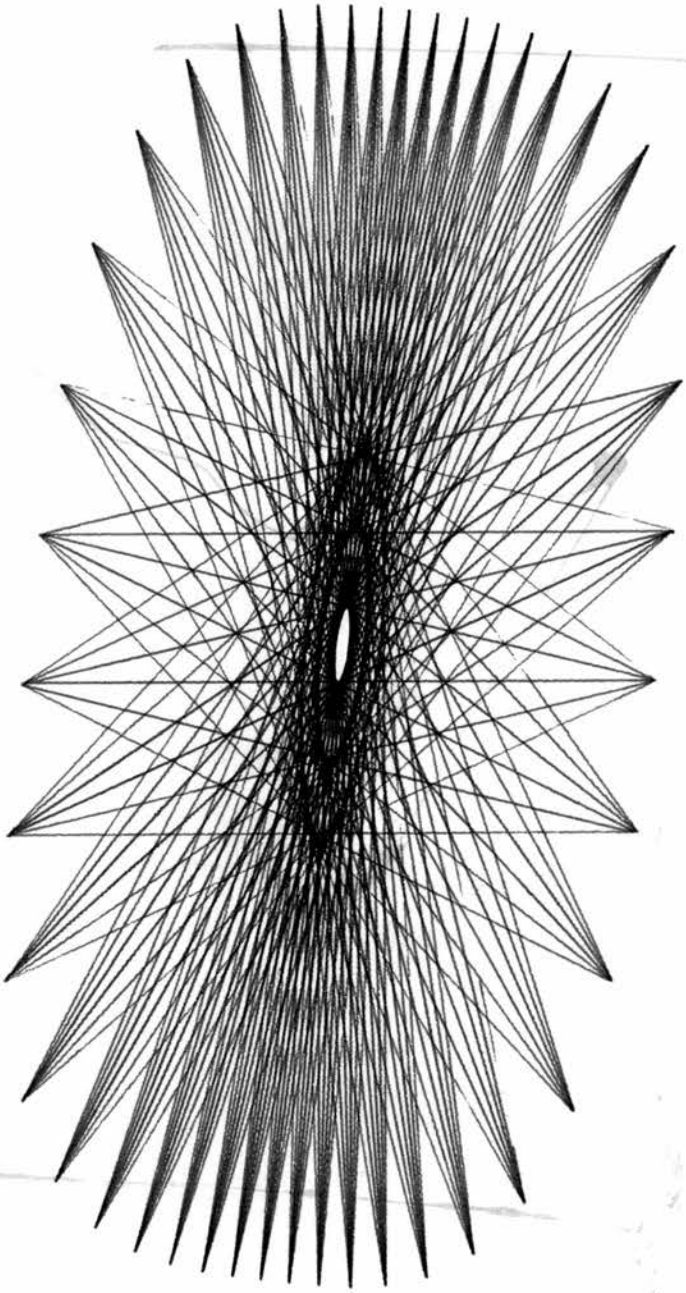


ILLUSTRATION 1

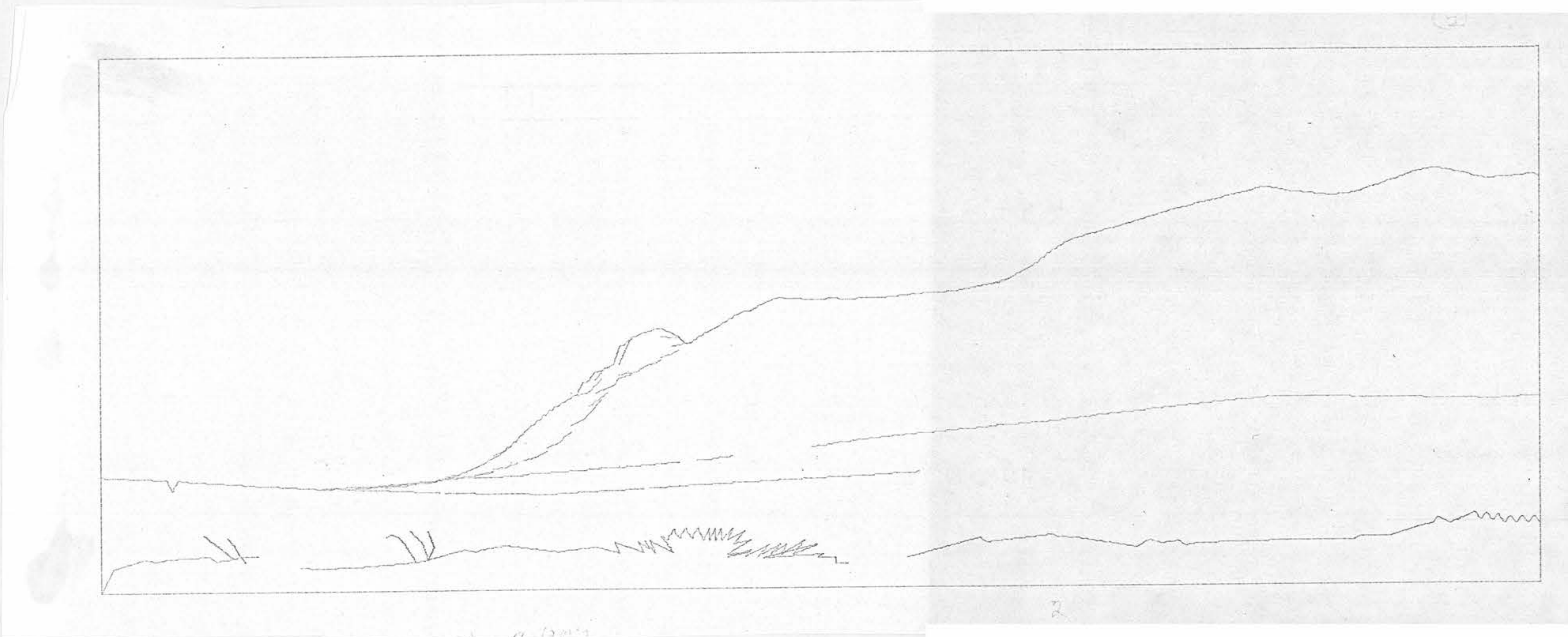


ILLUSTRATION 2

Simple Frame (East Lomond)

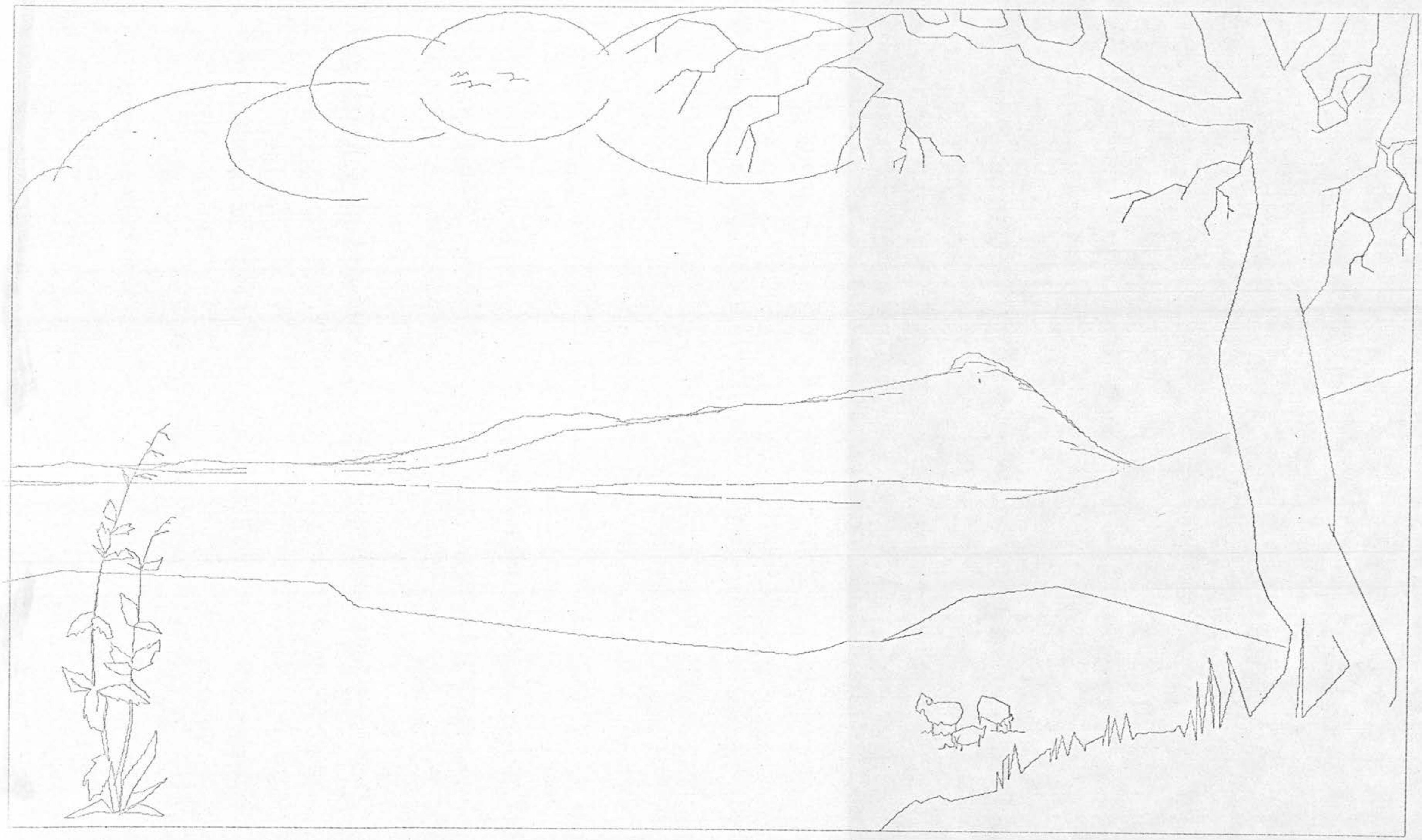


ILLUSTRATION 3

"ARTISTIC" FRAME (EAST LOMOND)

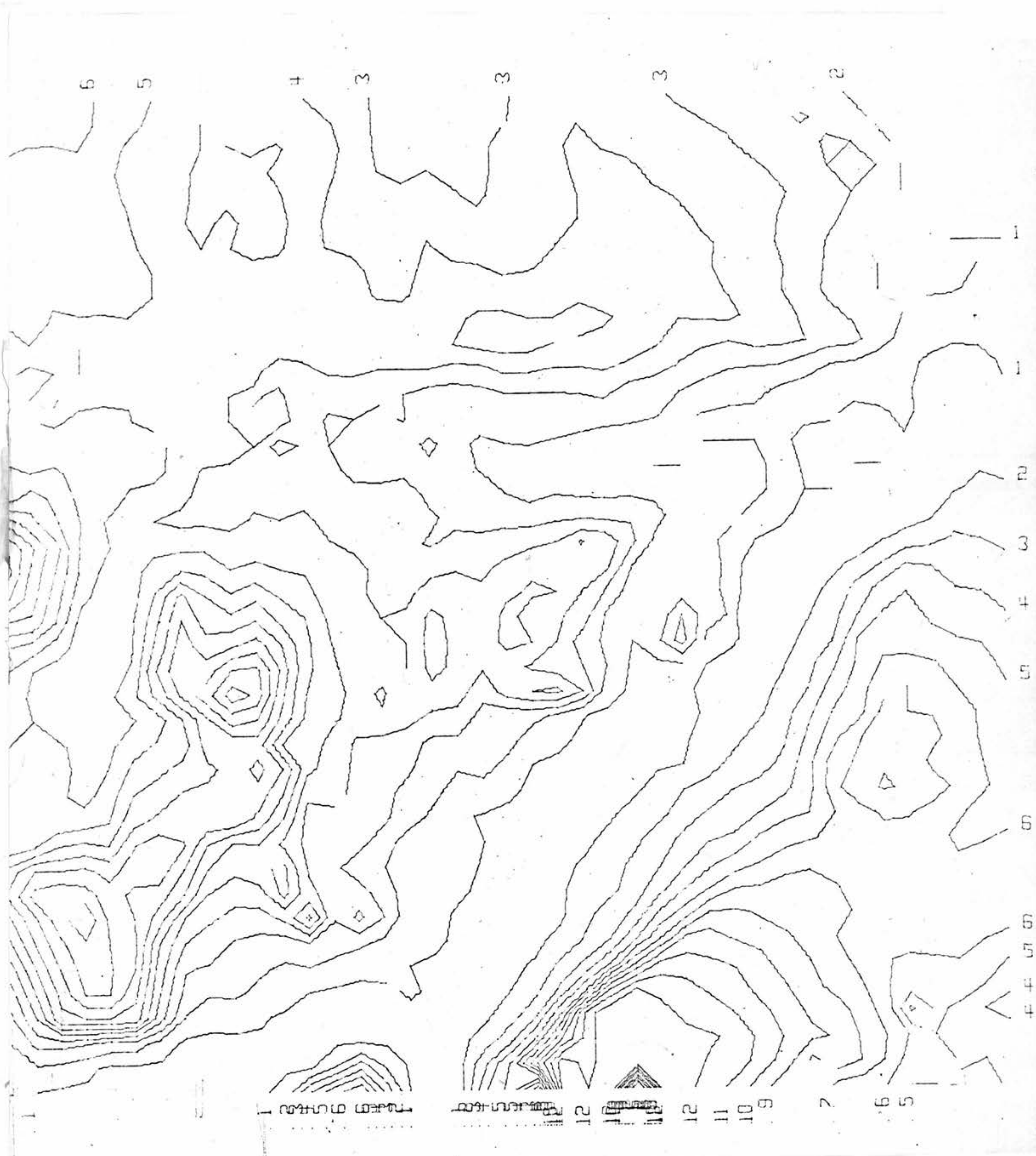


ILLUSTRATION 4

Contour map of ULLSWATER (showing error in punched card)

ILLUSTRATION 5

OUTLINE 1 (ULLSWATER)

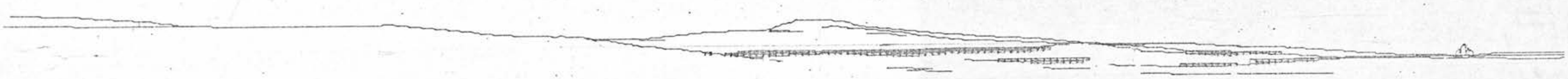


ILLUSTRATION 6

Actual size landscape (EAST Lomond)

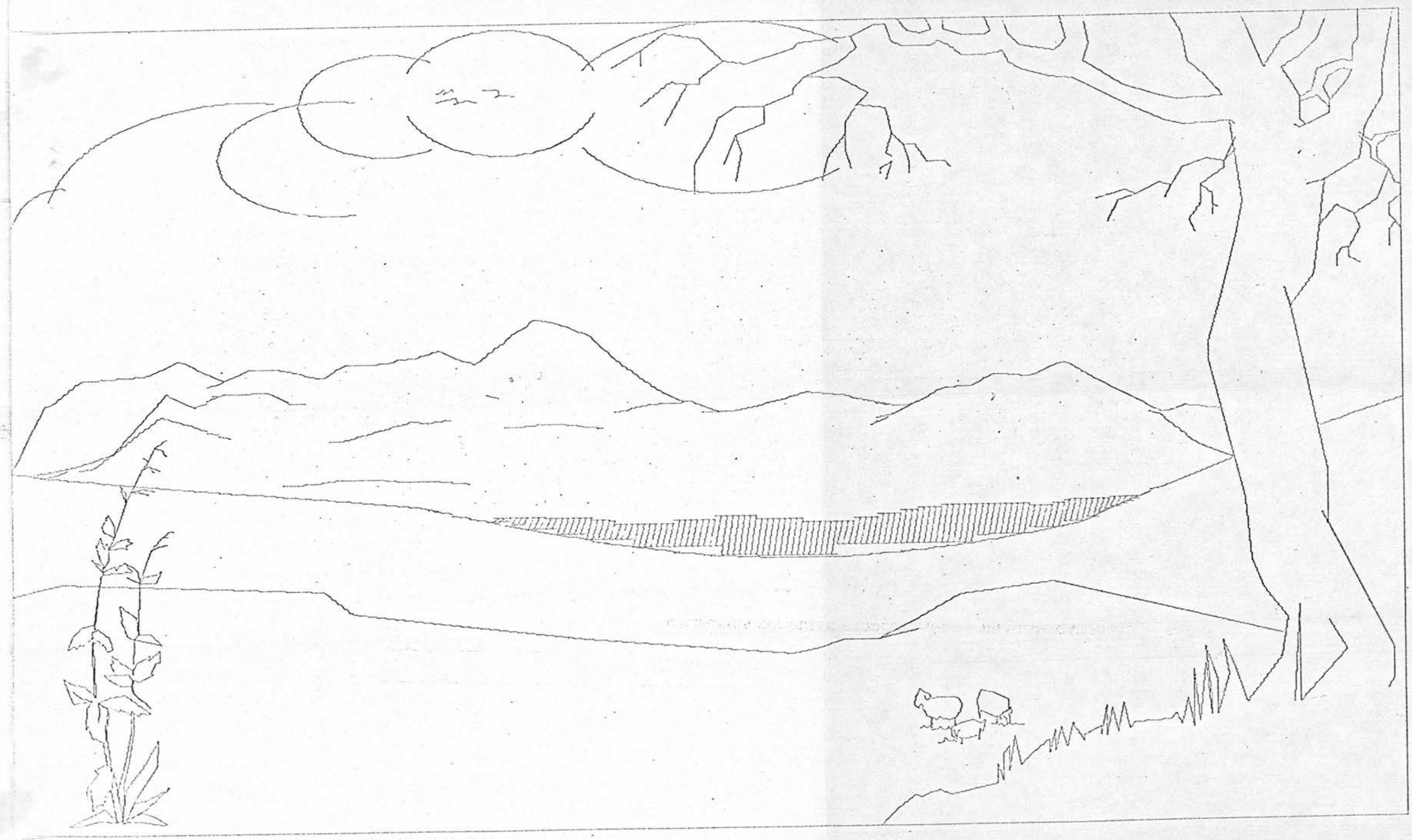


ILLUSTRATION 7

LANDSCAPE WITH WATER (ULLSWATER)

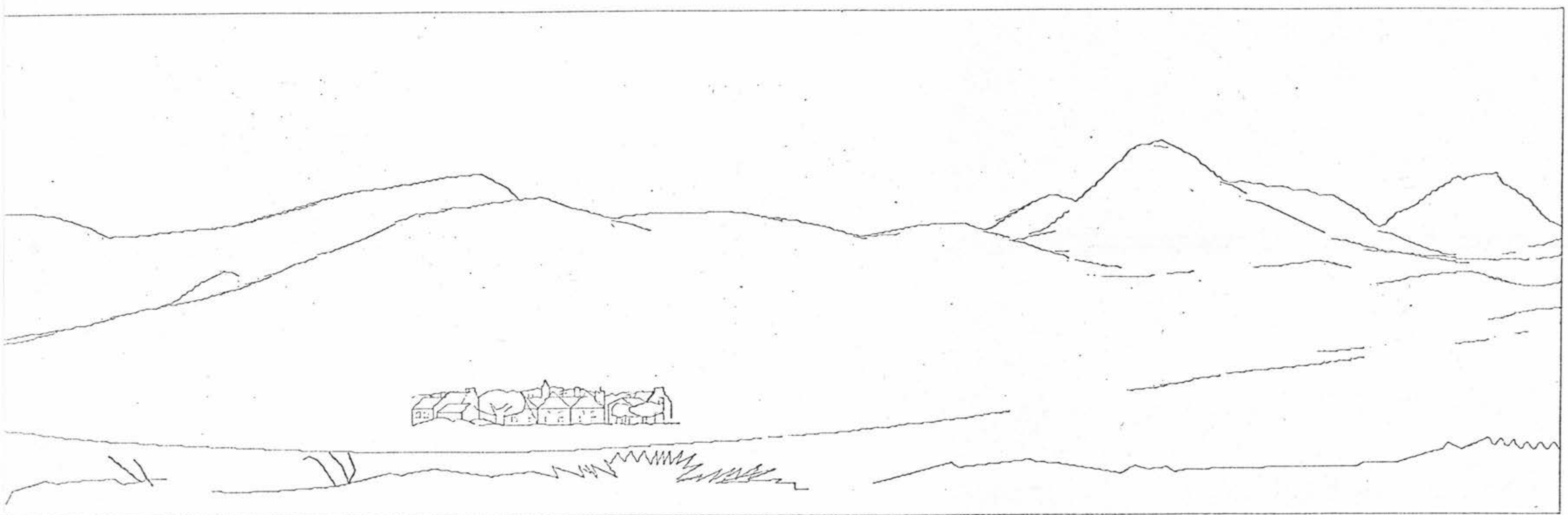


ILLUSTRATION 8

LANDSCAPE WITH TOWN (ULLSWATER)

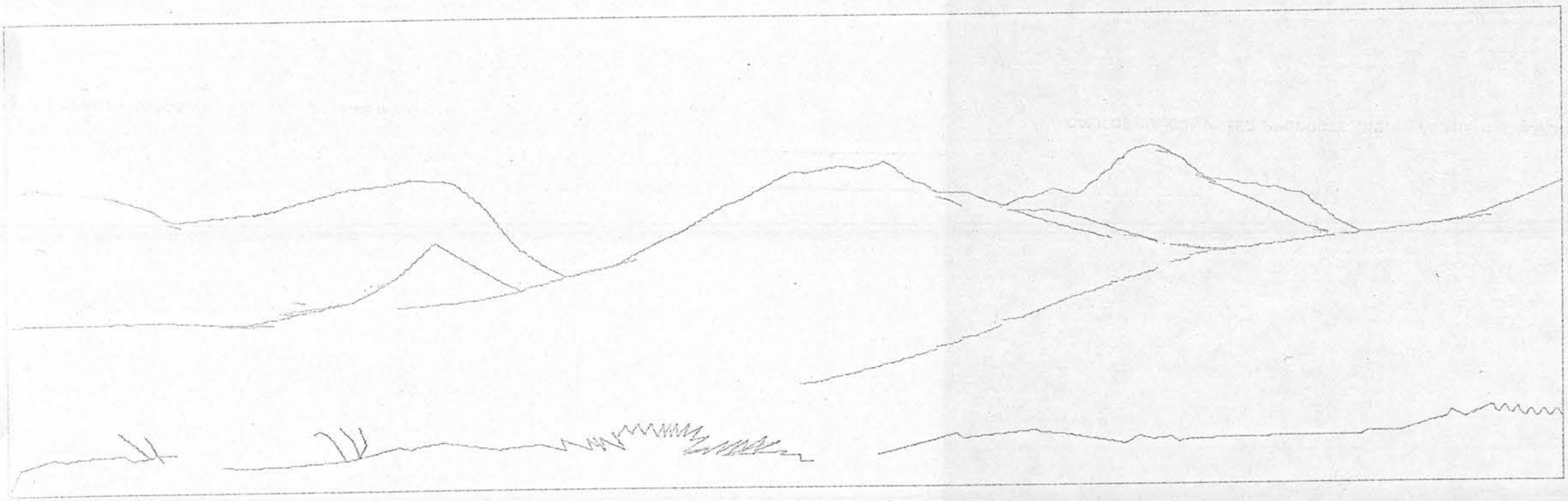


ILLUSTRATION 9

LANDSCAPE (ULLSWATER)

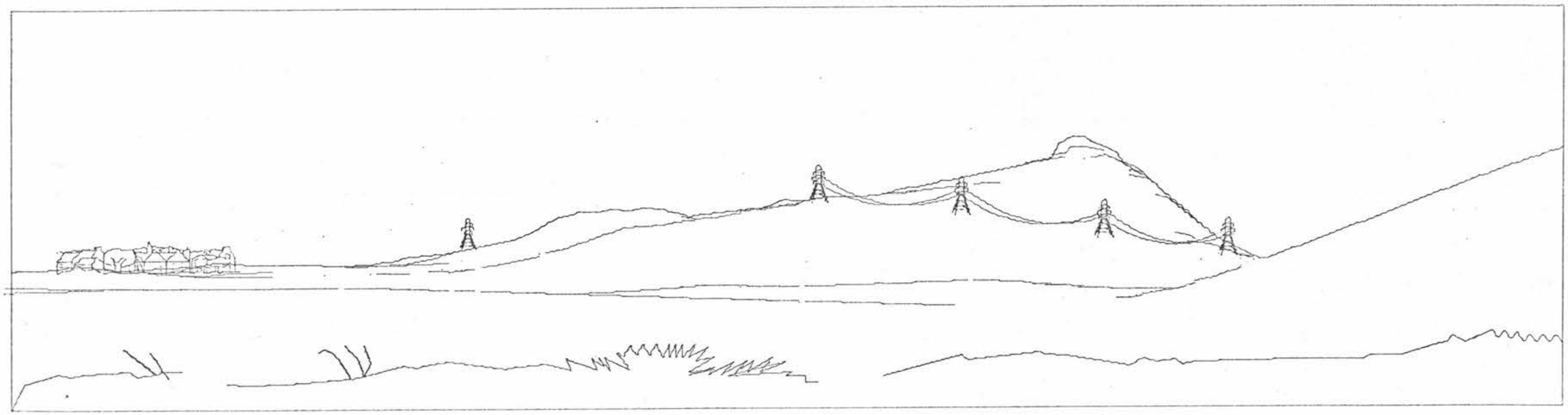


ILLUSTRATION 10

LANDSCAPE WITH PYLONS (EAST LOMONDS)

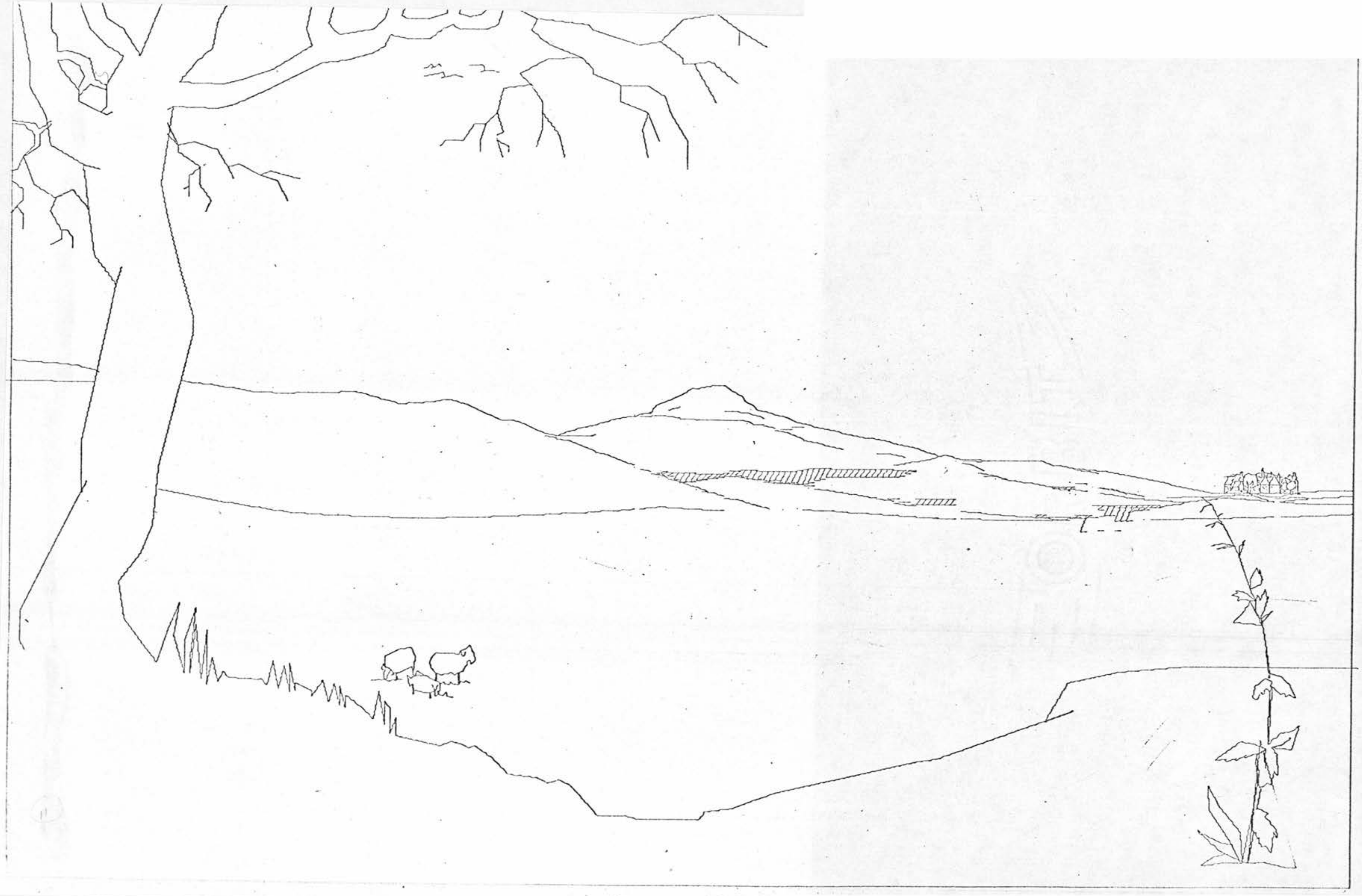


ILLUSTRATION 11

LANDSCAPE (EAST LOMOND)

```
 MODEL 44  PS        VERSION 3,  LEVEL 3   DATE  70237

      REAL*8 DIST(80),FHT(80),F,BIG15(5)
      REAL*8 WATER(80),WAHT(80)
      REAL*8 WAH,AHT
      DIMENSION B(50,50),DIAGD(80),DIAGA(80)
      DIMENSION OUTLIN(5,240),JKLBIG(5,240),IBIG15(5)
      DIMENSION WAT(80)
      DIMENSION TOWND(15),TOWNA(15),TD(15),TA(15),JTOWN(15),LTOWN(15)
      DIMENSION X(40,9),Y(40,9)
      DIMENSION XT(41,9),YT(41,9)
      DIMENSION XB(4,8),YB(4,8),XSH(19,7),YSH(19,7)
      DIMENSION FINE(9,9)
      DIMENSION XN(32,8),YN(32,8)
      DIMENSION  XP(5,12),YP(5,12),PYD(50),PYA(50),JPY(50),LPY(50)
      DIMENSION XSIMP(14,10),YSIMP(14,10)
C    OUTLIN HOLDS THE FIRST, SECOND, THIRD FOURTH, AND FIFTH HIGHEST ANGLES OF
C ELEVATION IF ANY OF THESE ANGLES ARE NOT PRESENT THEY ARE INDICATED WITH
C    99999 IN PLACE OF AN ANGLE.
C  JKLBIG CORRESPONDS TO OUTLIN AND HOLDS THE DISTANCE OF EACH POINT TO THE
C  OBSERVER.
C     DIST CONTAINS THE DISTANCE TO EACH POINT ALONG ONE RADIATING LINE
C   FHT CONTAINS THE HEIGHTS OF 80 POINTS ALONG ONE RADIATING LINE
C     WATER CONTAINS THE ANGLE OF ELEVATION OF ANY WATER ALONG A
C     RADIATING LINE WHICH IS VISIBLE TOTHE OBSERVER. WAT IS THE SAME
C     AS WATER IN SINGLE PRECISION
C     WAHT CONTAINS THE HEIGHT OF ANY WATER ALONG ONE RADIATING LINE
C     B IS A MATRIX WHICH HAS THE HEIGHTS OF ALL GRID POINTS ON THE MAP.
C     ANY POINTS WHICH ARE WATER ARE INDICATED WITH A MINUS SIGN IN
C     FRONT OF THE HEIGHT
C     DIAGD AND DIAGA HOLD THE Y AND X VALUES OF EACH POINT ALONG ONE
C     RADIATING LINE
C     TOWND, TOWNA HOLD THE Y AND X GRID VALUES OF UP TO 15 TOWNS ON THE
C     MAP. JTOWN WILL INDICATE WHICH PARTICULAR RADIATING LINE THE TOWN IS
C     ON. LTOWN WILL HOLD THE DISTANCE TO THE TOWN. THE GRID POINTS OF THE
C     TOWN ARE LATER TRANSFERRED TO TD, AND TA AS THEY ARE ENCOUNTERED ON THE
C     RADIATING LINES.
C     N HOLDS THE NUMBER OF POINTS ON ONE RADIATING LINE THAT ARE OFF
C  THE MAP. IF N IS 79 THE ENTIRE LINE IS OFF THE MAP.
      IJK=KLOCK(I,J)
C    READ IN DATA FOR FOREGROUND AND TREE
C    X AND Y HOLD TREE AND FOREGROUND DATA.
      DO 105 J=1,40
 105  READ(5,13)(X(J,I),Y(J,I),I=1,9)
 13   FORMAT(9(F3.0,F4.2,1X))
C    READ IN CHOICE FOR TREE POSITIONED ON RIGHT OR LEFT SIDE OF PAPER
C    FOR RIGHT SIDE USE A 1, FOR LEFT SIDE USE A 2.
      READ(5,14)ITREE
  14  FORMAT(I5)
C   READ IN DATA FOR NETTLES
      DO 250 J=1,32
```

```
 250   READ(5,251)(XN(J,I),YN(J,I),I=1,8)
 251   FORMAT(8(F5.1,F4.2))
C     READ IN DATA FOR BIRDS
       DO 221 J=1,4
 221   READ(5,251)(XB(J,I),YB(J,I),I=1,8)
C     READ IN DATA FOR SHEEP
       DO 222 J=1,19
 222   READ(5,208)(XSH(J,I),YSH(J,I),I=1,7)
 208   FORMAT(7(F5.1,F4.2,1X))
C     READ IN DATA FOR SIMPLE FOREGROUND
       DO 300 J=1,14
 300   READ(5,270)(XSIMP(J,I),YSIMP(J,I),I=1,10)
 270   FORMAT(10(F3.0,F3.2,1X))
C     READ IN TYPE OF FOREGROUND. 2 IS FOR A SIMPLE FOREGROUND, 1 IS FOR
C    TREE, NETTLES, ETC.
       READ(5,14)IFORE
C    READ IN NUMBER OF GRID POINTS
       READ(5,14)IGRID
C      READ IN CONTOUR HGTS
       DO 20 I=1,IGRID
 20    READ(5,10)(B(J,I),J=1,IGRID)
 10    FORMAT(10F7.0)
C     WHEN IFYES IS L, THIS INDICATES THAT A FINE SCAN IS PRESENT.
       READ(5,14)IFYES
       IF(IFYES.EQ.2)GO TO 246
       DO 31 I=1,9
 31    READ(5,29)(FINE(J,I),J=1,9)
 29    FORMAT(9(F5.0))
C    THESE VARIABLES GIVE THE 4 CORNERS OF THE SQUARE KILOMETRE THAT HAS BEEN
C     GIVEN A FINE SCAN.
       READ(5,32)FD1,FD2,FA1,FA2
 32    FORMAT(4F3.0)
       GO TO 245
C      READ IN THE NUMBER OF TOWNS INTO NTOWNS
 246   FD1=100.
       FD2=100.
       FA1=100.
       FA2=100.
 245    READ(5,30)NTOWNS
 30    FORMAT(I5)
C      READ IN Y AND X VALUES FOR EACH TOWN FOR UP TO 15 TOWNS
       DO 40 I=1,NTOWNS
 40    READ(5,45)TOWND(I),TOWNA(I)
 45    FORMAT(2F5.0)
       NT=NTOWNS
C     XT AND YT HOLD TOWN DATA.
       DO 113 J=1,41
 113   READ(5,13)(XT(J,I),YT(J,I),I=1,9)
       NB=1
```

```
C     READ PYLON DATA INTO XP AND YP
      DO 71 J=1,5
 71   READ(5,72)(XP(J,I),YP(J,I),I=1,12)
 72   FORMAT(12(F2.0,F3.2,1X))
C     READ IN NUMBER OF PYLONS
      READ(5,73)NPYS
 73   FORMAT(I5)
      DO 75 I=1,NPYS
C     READ IN THE X AND Y VALUES FOR UP TO 50 PYLONS
 75   READ(5,76)PYD(I),PYA(I)
 76   FORMAT(2F5.2)
C    READ IN OBSERVATION POINT
      READ(5,76)OBSERD,OBSERA
C    READ IN A VIEW POINT
      READ(5,76)PTD,PTA
      CALL PLOT(201,0.,240.,12.,240.,0.,10.,10.,10.)
      CALL PLOT(99)
      IF(IFORE.EQ.2)GO TO 22
      IF(ITREE.EQ.1)GO TO 16
C     REVERSE TREE, NETTLES, BIRDS AND SHEEP FOR LEFT SIDE TREE
      DO 18 J=1,40
      DO 18 I=1,9
      IF(X(J,I).GE.998.)GO TO 18
      X(J,I)-241-X(J,I)
 18   CONTINUE
      DO 253 J=1,32
      DO 253 I=1,8
      IF(XN(J,I).GE.998.)GO TO 253
      XN(J,I)=241-XN(J,I)
 253  CONTINUE
      DO 260 J=1,4
      DO 260 I=1,8
      IF(XB(J,I).GE.998.)GO TO 260
      XB(J,I)=241-XB(J,I)
 260   CONTINUE
      DO 106 J=1,19
      DO 106 I=1,7
      IF(XSH(J,I).GE.998.)GO TO 106
      XSH(J,I)=241-XSH(J,I)
 106  CONTINUE
 16   CALL PLOT(98,1.,1.)
C  DRAW THE TREE, NETTLES, BIRDS AND SHEEP.
      DO 101 J=1,40
      DO 101 I=1,9
      IF(X(J,I).GE.998.)GO TO 102
      CALL PLOT(90,X(J,I),Y(J,I))
      GO TO 101
 102  CALL PLOT(99)
 101  CONTINUE
```

```
      DO 255 J=1,32
      DO 255 I=1,8
      IF(XN(J,I).GE.998.)GO TO 254
      CALL PLOT(90,XN(J,I),YN(J,I))
      GO TO 255
  254 CALL PLOT(99)
  255  CONTINUE
      DO 235 J=1,4
      DO 235 I=1,8
      IF(XB(J,I).GE.998.)GO TO 204
      CALL PLOT(90,XB(J,I),YB(J,I))
      GO TO 235
  204 CALL PLOT(99)
  235 CONTINUE
      DO 244 J=1,19
      DO 244 I=1,7
      IF(XSH(J,I).GE.998.)GO TO 212
      CALL PLOT(90,XSH(J,I),YSH(J,I))
      GO TO 244
  212 CALL PLOT(99)
  244 CONTINUE
C     FRAME THE PICTURE.
   21 CALL PLOT(90,1.,2.)
      CALL PLOT(90,240.,2.)
      CALL PLOT(90,240.,9.)
      CALL PLOT(90,1.,9.)
      CALL PLOT(90,1.,2.)
      CALL PLOT(99)
      IF(ITREE.EQ.1)GO TO 22
C     REVERSE LEFT SIDE TREE AGAIN IN ORDER TO TEST FOR HIDDEN LANDSCAPE
      DO 19 J=1,40
      DO 19 I=1,9
       IF(X(J,I).GE.998.)GO TO 19
      X(J,I)=241-X(J,I)
   19 CONTINUE
   22 DO 51 I=1,NPYS
   51 JPY(I)=99999
      DO 50 I=1,NTOWNS
   50 TD(I)=99999.
 2006 IOD=OBSERD
      IOA=OBSERA
      AHT=B(IOD,IOA)
      DO 1000 J=1,240
      DO 46 I=1,80
   46 WAHT(I)=99999.
C     TO FIND 80 POINTS ALONG EACH OF 240 RADIATING LINES
      CALL DIAG(OBSERD,OBSERA,PTD,PTA,DIAGD,DIAGA,DIST,N)
      IF(N.GE.79)GO TO 810
      DO 800 L=N,79
```

```
      IF(NT.LT.1)GO TO 795
C     TEST FOR TOWN ALONG RADIATING LINE.
      DO 790 K=1,NTOWNS
      IF(((ABS(DIAGD(L)-TOWND(K))).GT.0.2).OR.((ABS(DIAGA(L)-TOWNA(K)
     2).GT.0.2)))GO TO 790
      NT=NT-1
      TD(NB)=TOWND(K)
      TA(NB)=TOWNA(K)
      TOWND(K)=99999.
      TOWNA(K)=99999.
      JTOWN(NB)=J
      LTOWN(NB)=L
      WRITE(6,99)TD(NB),TA(NB),TOWND(K),TOWNA(K),K,JTOWN(NB),LTOWN(NB),
     2NT,NB
 99   FORMAT(' ',4F10.5,5I5)
      NB=NB+1
 790   CONTINUE
C     TEST FOR PYLON ALONG RADIATING LINE
      DO 200 I=1,NPYS
      IF(((ABS(DIAGD(L)-PYD(I))).GT.0.2).OR.((ABS(DIAGA(L)-PYA(I)).GT.
     20.2)))GO TO 200
      JPY(I)=J
      LPY(I)=L
 200  CONTINUE
C  TO FIND INTERPOLATED HEIGHT OF EACH POINT ALONG EACH DIAGONAL
 795  CALL INTERP(DIAGD(L),DIAGA(L),F,B,WAH,IW,FINE,FD1,FD2,FA1,FA2)
      IF(IW.EQ.1)WAHT(L)=WAH
      FHT(L)=F
 800  CONTINUE
C     FIND ANGLE OF ELEVATION ALONG EACH POINT
      CALL ELVATE(AHT,FHT,DIST,BIG15,IBIG15,N,IYES,WATER,WAHT)
C     IYES INDICATES IF THIS RADIATING LINE HAS ANY VISIBLE WATER
      IF(IYES.EQ.0) GO TO 820
C     CONVERT WATER TO SINGLE PRECISION TO PLOT IT
      DO 805 I=1,80
 805  WAT(I)=WATER(I)
      CALL WWATER(WAT,J,N,ITREE,IFORE)
      GO TO 820
C THE DIAGONAL LINE CALCULATED IS OFF THE MAP
 810  DO 815 K=1,5
 815  BIG15(K)=99999.
C     HIGHEST POINT PUT INTO OUTLIN(J)
 820  DEGREE=180/3.14159
      DO 830 K=1,5
      OUTLIN(K,J)=BIG15(K)*DEGREE
 830  JKLBIG(K,J)=IBIG15(K)
C FIND COORDINATES OF NEXT DIAGONAL LINE 0.5 DEGREES TO THE RIGHT OF LAST ONE
 960  CALL DIA5(OBSERD,OBSERA,PTD,PTA,GNUX,GNUY)
      PTD=GNUX
```

```
MODEL 44  PS        VERSION 3,  LEVEL 3   DATE  70237

 1000 PTA=GNUY
C PROJECT THE HIGHEST ANGLES OF ELEVATION ONTO A SCREEN AND DRAW PROJECTION
      DO 1020 I=1,NTOWNS
      IF(TD(I).GE.99998.)GO TO 1030
      JT=JTOWN(I)
      ITD=TD(I)
      ITA=TA(I)
      TSA1=0.
      TSA2=0.
      TSD1=0.
      TSD2=0.
      CALL TOWN1(OBSERD,OBSERA,TD(I),TA(I),JT,LTOWN(I),AHT,
     2B(ITD,ITA),JKLBIG(1,JT),JKLBIG(2,JT),JKLBIG(3,JT),JKLBIG(4,JT),
     3JKLBIG(5,JT),OUTLIN(1,JT),OUTLIN(2,JT),OUTLIN(3,JT),OUTLIN(4,JT),
     4OUTLIN(5,JT),XT,YT,TSA1,TSA2,TSD1,TSD2,ITREE,IFORE)
 1020  CONTINUE
 1030 CALL PYLON(XP,YP,PYD,PYA,JPY,OBSERD,OBSERA,NPYS,B,LPY,JKLBIG,
     2OUTLIN,IFORE,ITREE,IOD,IDA)
      CALL PLOT(98,1.,1.)
      CALL PROJ(OUTLIN,JKLBIG,TSA1,TSA2,TSD1,TSD2,ITREE,IFORE,YTOP,YBOT)
      CALL PLOT(99)
      IF(IFORE.EQ.1)GO TO 948
      YTOP=YTOP+1.0
      YBOT=YBOT-0.8
      CALL PLOT(90,1.,YBOT)
      CALL PLOT(90,240.,YBOT)
      CALL PLOT(90,240.,YTOP)
      CALL PLOT(90,1.,YTOP)
      CALL PLOT(90,1.,YBOT)
      DO 951 J=1,14
      DO 951 I=1,10
      IF(XSIMP(J,I).GE.998.)GO TO 952
      CALL PLOT(90,XSIMP(J,I),YBOT+YSIMP(J,I))
      GO TO 951
 952  CALL PLOT(99)
 951  CONTINUE
      GO TO 949
 948   CALL CLOUD(ITREE,YTOP)
 949  CALL PLOT(7)
      TIME=(KLOCK(I,J)-IJK)/3000.
      WRITE(6,950)TIME
 950  FORMAT(' ',F6.2,' MINUTES')
      STOP
      END
```

```
MODEL 44  PS        VERSION 3,  LEVEL 3   DATE  70237

      SUBROUTINE DIAG(A,B,C,D,X,Y,DISTB,N)
      REAL*8 DISTB(80)
      DIMENSION X(80),Y(80)
C  N KEEPS TRACK OF THE NUMBER OF POINTS THAT GO OFF THE MAP ALONG DIAGONAL
      N=1
      AA=A/80.
      BB=B/80.
      CC=C/80.
      DD=D/80.
      DO 50 I=1,80
      X(I)=(I-1.0)*AA+(81.0-I)*CC
      Y(I)=(I-1.0)*BB+(81.0-I)*DD
      IF(X(I).GE.1.0.AND.X(I).LE.41.AND.Y(I).GE.1.0.AND.Y(I).LE.41)GO
     2TO 45
      N=N+1
      GO TO 50
 45   Q=X(I)-A
      R=Y(I)-B
      DISTA=SQRT(Q*Q+R*R)
C  DISTB IS THE DISTANCE IN KILOMETERS TO EACH POINT ALONG DIAGONAL
      DISTB(I)=DISTA*.25
 50   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INTERP(QD,QA,F,B,WAHT,IW,FINE,FD1,FD2,FA1,FA2)
      REAL*8 F,WAHT
      DIMENSION FINE(9,9)
      DIMENSION B(50,50)
      II=QD
      JJ=QA
C TAKE THE INTERPOLATED VALUE OF THE 4 CLOSEST GRID POINTS TO THE POINT QD,
      KK=1
      IF(QD.GE.FD1.AND.QD.LE.FD2.AND.QA.GE.FA1.AND.QA.LE.FA2)GO TO 110
      X1=ABS(B(II,JJ))
      X2=ABS(B(II+1,JJ))
      X3=ABS(B(II,JJ+1))
      X4=ABS(B(II+1,JJ+1))
      GO TO 120
110   LL=(QD-FD1)*2.+1
      MM=(QA-FA1)*2.+1
      X1=FINE(LL,MM)
      X2=FINE(LL+1,MM)
      X3=FINE(LL,MM+1)
      X4=FINE(LL+1,MM+1)
120   R=QD-II
      S=QA-JJ
      R3=1-R
      S3=1-S
      F=R3*S3*X1+R*S3*X2+S*R3*X3+R*S*X4
      CALL OVERFL(I)
      IF(I.EQ.3) WRITE(6,1)II,JJ,X1,X2,X3,X4,R,S,R3,S3,F
1     FORMAT(' ',2I8,4F7.0,4F10.5,F14.6)
      IF(B(II,JJ).LT.0) GO TO 10
70    IF(B(II+1,JJ).LT.0)GO TO 30
80    IF(B(II,JJ+1).LT.0)GO TO 40
90    IF(B(II+1,JJ+1).LT.0)GO TO 50
      IW=0
      GO TO 20
10    IF(R.LT.0.5.AND.S.LT.0.5)GO TO 200
      GO TO 70
30    IF(R3.LT.0.5.AND.S.LT.0.5)GO TO 210
      GO TO 80
40    IF(R.LT.0.5.AND.S3.LT.0.5)GO TO 220
      GO TO 90
50    IF(R3.LT.0.5.AND.S3.LT.0.5)GO TO 230
      IW=0
      GO TO 20
200   WAHT=B(II,JJ)
      GO TO 60
210   WAHT=B(II+1,JJ)
      GO TO 60
220   WAHT=B(II,JJ+1)
      GO TO 60
```

```
220   WAHT=B(II,JJ+1)
      GO TO 60
230    WAHT=B(II+1,JJ+1)
60    F=-F
      IW=1
20    RETURN
      END
```

```
MODEL 44  PS        VERSION 3,  LEVEL 3   DATE  70237

      SUBROUTINE ELVATE(AHT,FHT,D,BIG15,JNBIG,N,IYES,WATER,WAHT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BIG15(5),JNBIG(5)
      DIMENSION FHT(80),PHI(80),D(80)
      DIMENSION WATER(80)
      DIMENSION WAHT(80)
C   R IS THE RADIUS OF THE EARTH IN FEET
C   D(DISTANCE) IS FIRST CONVERTED TO MILES(FROM KILOMETERS) AND THEN FEET
C   CALCULATIONS ARE IN DOUBLE PRECISION BECAUSE OF TINY NUMBERS
      R=3965.65*5280.0
      D9=0.622*5280.0
      DO 100 I=N,79
      D1=D(I)*D9
      DOR=D1/R
      RA=R+DABS(AHT)
      RC=R+DABS(FHT(I))
      X=DSQRT(RA*RA+RC*RC-2.*RA*RC*DCOS(DOR))
      IF(X.LE.1)GO TO 65
 51   P3=(RC/X)*DSIN(DOR)
C   IF P3 IS GE 1, THE 2 HEIGHTS ARE VERY SIMILAR,AND ANGLE PHI SET TO ZERO
      IF(P3.GE.1.0)GO TO 65
      P1=DARSIN(P3)
C   TEST IF COS ALPHA LT 0 . IF YES ADD TO PHI
      DCA= (X*X)+(RA*RA)-(RC*RC)
      IF(DCA.LT.0.0)P1=P1+2*(3.14159/2-P1)
      P4=R/RA
      IF(P4.GE.1.0)GO TO 65
      P2=DARSIN(P4)
      PHI(I)=P1-P2
      CALL OVERFL(J)
      IF(J.EQ.3) WRITE(6,1)R,D(I),DOR,RA,RC,X,P3,P1,P4,P2,PHI(I),I
 1    FORMAT(' ',6D16.8,/,5D16.8,I4)
      GO TO 100
 65   PHI(I)=-0.001
 100  CONTINUE
C   FIND LARGEST ANGLE OF ELEVATION CALLED BIG AND ELIMINATE ALL PTS BEHIND IT
      M=N+1
      JNBIG(1)=N
      BIG15(1)=PHI(N)
      DO 200 J=M,77
      IF(BIG15(1).GE.PHI(J))GO TO 200
      BIG15(1)=PHI(J)
C   JBIG,KBIG,ETC. HOLD LARGEST ANGLE,2ND LARGEST,ETC. AND ARE PUT INTO JJBIG,
C     KKBIG,ETC.
C   VERY RARELY ARE THERE ALL 5 ANGLES
      JNBIG(1)=J
 200  CONTINUE
C     FIND SECOND LARGEST ) ON SEPARATE SLOPE
C     ONLY CONSIDER )'S IN FRONT OF 'BIG'
```

```
      DO 400 I=2,5
      JB=JNBIG(I-1)+1
      IF(JB.GE.77)GO TO 350
      JNBIG(I)=JB
      JB1=JB+1
      MNO=1
C  FIND 2ND,3RD,4TH,AND 5TH LARGEST ANGLES OF ELEVATION(OR DEPRESSION) ALL
C     ON SEPARATE SLOPES FROM THE PREVIOUS ONES,ALWAYS ELIMINATING ALL PTS
C     BEHIND THE CURRENT POINT
 250  BIG15(I)=PHI(JB)
      DO 300 K=JB1,77
      IF(BIG15(I).GT.PHI(K))GO TO 300
      BIG15(I)=PHI(K)
      JNBIG(I)=K
 300  CONTINUE
      IF(JB.GT.76)GO TO 350
      IF((JNBIG(I)-JNBIG(I-1)).NE.MNO)GO TO 400
      JNBIG(I)=JB1
      JB=JB+1
      JB1=JB1+1
      MNO =MNO+1
      GO TO 250
 350  BIG15(I)=99999.
      JNBIG(I)=0
 400  CONTINUE
 700  IYES=0
      DO 750 I=1,80
 750  WATER(I)=99999.
      DO 800 I=N,79
      IF(FHT(I).GT.0)GO TO 800
      CALL TOWN2(AHT,WAHT(I),D(I),PHIW)
      IF(I.GT.JNBIG(1).AND.I.GT.JNBIG(2).AND.I.GT.JNBIG(3).AND.I.GT.
     2JNBIG(4).AND.I.GE.JNBIG(5))GO TO 790
      IF(PHIW.GT.0.)GO TO 800
      IF(I.EQ.JNBIG(1).OR.I.EQ.JNBIG(2).OR.I.EQ.JNBIG(3).OR.I.EQ.
     2JNBIG(4).OR.I.EQ.JNBIG(5))GO TO 790
      IF(I.LT.JNBIG(1))GO TO 800
      IF(I.LT.JNBIG(2))GO TO 900
      IF(I.LT.JNBIG(3))GO TO 910
      IF(I.LT.JNBIG(4))GO TO 920
      GO TO 930
 900  IF(PHIW.GE.BIG15(2))GO TO 790
      GO TO 800
 910  IF(PHIW.GE.BIG15(3))GO TO 790
      GO TO 800
 920  IF(PHIW.GE.BIG15(4))GO TO 790
      GO TO 800
 930  IF(BIG15(5).GE.99998.)GO TO 790
      IF(PHIW.GE.BIG15(5))GO TO 790
```

```
      GO TO 800
  790 IF(WAHT(I).GE.99998.)GO TO 800
      WATER(I)=PHIW*180/3.14159
      IYES=1
  800 CONTINUE
      RETURN
      END
```

MODEL 44 PS VERSION 3, LEVEL 3 DATE 70238 P

```
      SUBROUTINE WWATER(WATER,J,N,ITREE,IFORE)
      DIMENSION WATER(80)
      JJ=0
      DO 100 I=N,79
      IF(WATER(I).GE.99998.) GO TO 100
      X=J
      H=WATER(I)*.3
      Y=H+5.
      IF(JJ.NE.0)GO TO 10
      IF(IFORE.EQ.2)GO TO 20
      IF(ITREE.EQ.1)GO TO 23
      XA=241-X
      GO TO 25
23    XA=X
25    IF(XA.GT.207.AND.XA.LT.238)CALL TREE(XA,YD,HID)
      IF(HID.EQ.1)GO TO 100
20    CALL PLOT(90,X,Y)
      CALL PLOT(90,X+1,Y)
      JJ=1
10    IF(WATER(I+1).LT.99998.)GO TO 100
      CALL PLOT(90,X,Y)
      CALL PLOT(90,X+1,Y)
      CALL PLOT(99)
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PROJ(PHI,JNBIG,TSA1,TSA2,TSD1,TSD2,ITREE,IFORE,YTOP,
     2YBOT)
      DIMENSION PHI(5,240),JNBIG(5,240)
C  THIS SUBROUTINE ARBITARILY TAKES A DISTANCE OF 1000 FEET TO PROJECT
C  THE ANGLE OF ELEVATION (PHI) TO A HEIGHT H WHICH IS DIVIDED BY 50 IN
C  ORDER TO FIT THE PAPER AND PLACED 5 INCHES ABOVE X AXIS
      YTOP=0.
      DO 20 I=1,240
      HID=0
      X=I
C  IF ANGLE IS NOT PRESENT (GE 99998) THE PEN IS LIFTED
      IF(PHI(1,I).GE.99998.)GO TO 19
      GO TO 18
 19   Y=5.
      CALL PLOT(0,X,Y)
      GO TO 20
 18   H=PHI(1,I)*.3
      Y=H+5.
C    TEST FOR HIDDEN LINES IN LANDSCAPE
      IF(IFORE.EQ.2)GO TO 1
      IF(ITREE.EQ.1)GO TO 21
      XX=241-X
      GO TO 22
 21   XX=X
 22   IF(XX.GT.207.AND.XX.LT.238)CALL TREE(XX,Y,HID)
      IF(HID.EQ.1)GO TO 20
 1    IF(X.GT.TSA1.AND.X.LT.TSA2.AND.Y.GT.TSD1.AND.Y.LT.TSD2)GO TO 20
      CALL PLOT(90,X,Y)
      IF(Y.GT.YTOP)YTOP=Y
 20    CONTINUE
      YBOT=5.
      DO 40 K=2,5
      CALL PLOT(98,1.,1.)
      DO 40 J=1,240
      HID=0
      X=J
      IF(PHI(K,J).GE.99998.)GO TO 30
      GO TO 25
 30   CALL PLOT(99)
      GO TO 40
 25   IF(ABS(JNBIG(K,J)-JNBIG(K,J-1)*1.).LE.1)GO TO 35
C  IF LINE IS NOT CONTINUOUS PEN IS LIFTED(DETERMINE BY DISTANCE APART)
      IF(PHI(K,J+1).GE.99998.)GO TO 26
C  IF LINE IS CONTINUOUS BUT SEPARATED BY ONE POINT
      IF(ABS(JNBIG(K,J-1)-JNBIG(K,J+1)*1.).LE.1)GO TO 45
C  IF 2ND-5TH LINE IS ATTACHED TO MAIN OUTLINE
 26   IF(ABS(JNBIG(K,J-1)-JNBIG(1,J)*1.).LE.1)GO TO 37
      X=X-1
 33   CALL PLOT(99)
```

```
35    H=PHI(K,J)*.3
      Y=H+5.
      IF(IFORE.EQ.2)GO TO 2
      IF(ITREE.EQ.1)GO TO 27
      XX=241-X
      GO TO 28
27    XX=X
28    IF(XX.GT.207.AND.XX.LT.238)CALL TREE(XX,Y,HID)
      IF(HID.EQ.1)GO TO 40
2     IF(X.GT.TSA1.AND.X.LT.TSA2.AND.Y.GT.TSD1.AND.Y.LT.TSD2)GO TO 40
      CALL PLOT(90,X,Y)
      IF(Y.LT.YBOT)YBOT=Y
      GO TO 40
45    X=X+1
      H=PHI(K,J+1)*.3
      Y-H+5.
      IF(IFORE.EQ.2)GO TO 3
      IF(ITREE.EQ.1)GO TO 38
      XX=241-X
      GO TO 39
38    XX=X
39    IF(XX.GT.207.AND.XX.LT.238)CALL TREE(XX,Y,HID)
      IF(HID.EQ.1)GO TO 40
3     IF(X.GT.TSA1.AND.X.LT.TSA2.AND.Y.GT.TSD1.AND.Y.LT.TSD2)GO TO 40
      CALL PLOT(90,X,Y)
      X=X-2
      IF(Y.LT.YBOT)YBOT=Y
      GO TO 33
37    H=PHI(1,J)*.3
      Y=H+5.
      IF(IFORE.EQ.2)GO TO 4
      IF(ITREE.EQ.1)GO TO 41
      XX=241-X
      GO TO 42
41    XX=X
42    IF(XX.GT.207.AND.XX.LT.238)CALL TREE(XX,Y,HID)
      IF(HID.EQ.1)GO TO 40
4     IF(X.GT.TSA1.AND.X.LT.TSA2.AND.Y.GT.TSD1.AND.Y.LT.TSD2)GO TO 40
      CALL PLOT(90,X,Y)
      X=X-2
      IF(Y.LT.YBOT)YBOT=Y
      GO TO 33
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DIA5(A,B,C,D,X,Y)
C  ROTATION FORMULAE
      THETA =.5*3.14159/180.
      X=(C-A)*COS(THETA)+(D-B)*SIN(THETA)+A
      Y=(A-C)*SIN(THETA)+(D-B)*COS(THETA)+B
      RETURN
      END
```

```
      SUBROUTINE TOWN1(OD,OA,TD,TA,JT,LTO,AHT,THT,JB,KB,LB,MB,NB,O1,O2,
     2O3,O4,O5,XT,YT,TSA1,TSA2,TSD1,TSD2,ITREE,IFORE)
      DIMENSION XT(41,9),YT(41,9)
      REAL*8 AHT,D3,PHIT,TTHT
      D1=OD-TD
      D2=OA-TA
      D=SQRT(D1*D1+D2*D2)*.25
      D3=D
      TTHT=THT+25.
      CALL TOWN2(AHT,TTHT,D3,PHIT)
      PHIT=PHIT*180/3.14159
      WRITE(6,10)THT,PHIT,TD,TA,JT,LTO,JB,KB,LB,MB,NB,O1,O2,O3,O4,O5
   10 FORMAT(' ',2D16.8,2F6.3,7I5,5F9.3)
      IF(LTO.GT.JB.AND.LTO.GT.KB.AND.LTO.GT.LB.AND.LTO.GT.MB.AND.LTO.GE.
     2NB)GO TO 80
      IF(LTO.EQ.JB.OR.LTO.EQ.KB.OR.LTO.EQ.LB.OR.LTO.EQ.MB.OR.LTO.EQ.NB)
     2GO TO 80
      IF(LTO.LT.JB)GO TO 15
      IF(LTO.LT.KB)GO TO 25
      IF(LTO.LT.LB)GO TO 35
      IF(LTO.LT.MB)GO TO 45
      GO TO 60
   15 IF(PHIT.GE.O1)GO TO 80
      GO TO 90
   25 IF(PHIT.GE.O2)GO TO 80
      GO TO 90
   35 IF(PHIT.GE.O3)GO TO 80
      GO TO 90
    45 IF(PHIT.GE.O4)GO TO 80
      GO TO 90
   60 IF(O5.GE.99998.)GO TO 80
      IF(PHIT.GE.O5)GO TO 80
   90 RETURN
   80 D-D3
      D=D*.001
C    IF THE TOWN IS 1000 FEET OR CLOSER THE PROJECTION IS 1.1 INCHES
C    IF THE TOWN IS 20000 FEET AWAY OR MORE THE PROJECTION IS 0.2 INCHES
C ICORRECT HEIGHT FOR THAT DISTANCE
      TDIS=.70
      IF(D.LE.1.)GO TO 150
      DO 140 I=1,20
      TDIS=TDIS-0.020
      IF(D.LE.I)GO TO 150
  140 CONTINUE
  150 PHI=PHIT
      XX=JT
      H=PHI*.3
      YY=H+5.
      CALL PLOT(98,XX,YY)
```

```
      DO 111 J=1,41
      DO 111 I=1,9
C   TO TEST IF TOWN EXTENDS OUT OF PICTURE
      HID =0
C   TO TEST IF TREE IS IN FROMT OF TOWN
      XA=XX+XT(J,I)*TDIS
      IF(XA.LT.1.OR.XA.GT.240)GO TO 112
      YD=YY+YT(J,I)*(.5*TDIS)
      IF(IFORE.EQ.2)GO TO 113
      IF(ITREE.EQ.1)GO TO 115
      XAA=241-XA
      GO TO 116
  115 XAA=XA
  116 IF(XAA.GT.207.AND.XAA.LT.238)CALL TREE(XAA,YD,HID)
      IF(HID.EQ.1)GO TO 112
  113 CALL PLOT(90,XA,YD)
      GO TO 111
  112 CALL PLOT(99)
  111 CONTINUE
      TSA1=XX-32*TDIS
      TSA2=XX+30*TDIS
      TSD1=YY
      TSD2=YY+.8*TDIS
      WRITE(6,199)TSA1,TSA2,TSD1,TSD2
  199 FORMAT(' ',F10.5)
      RETURN
      END
```

```
      SUBROUTINE TOWN2(AHT,THT,D,PHIT)
      IMPLICIT REAL*8(A-H,O-Z)
      R=3965.65*5280.0
      D=D*0.622*5280.0
      DOR=D/R
      RA=R+DABS(AHT)
      RC=R+DABS(THT)
      X=DSQRT(RA*RA+RC*RC-2.*RA*RC*DCOS(DOR))
      P3=(RC/X)*DSIN(DOR)
      IF(P3.GE.1.0)GO TO 65
      P1=DARSIN(P3)
      DCA=(X*X)+(RA*RA)-(RC*RC)
      IF(DCA.LT.0.0)P1=P1+2*(3.14159/2-P1)
      P4=R/RA
      IF(P4.GE.1.0)GO TO 65
      P2=DARSIN(P4)
      PHIT=P1-P2
      GO TO 100
 65   PHIT=-.001
 100  RETURN
      END
```

```
      SUBROUTINE TREE(X,Y,HID)
      IF(X.GE.232.AND.Y.GE.4.)RETURN
      IF(X.GE.227.AND.Y.GE.4.55)RETURN
      IF(X.LE.213.AND.Y.LE.4.80)RETURN
      IF(X.LE.212.AND.Y.GE.7.10)RETURN
      IF(Y.LE.6.)GO TO 10
      IF(Y.LT.6.3)GO TO 50
      IF(Y.LT.6.5)GO TO 52
      IF(Y.LT.6.7)GO TO 54
      IF(Y.LT.6.9)GO TO 56
      IF(Y.LT.7.1)GO TO 58
      IF(Y.LT.7.3)GO TO 60
      IF(X.LT.214.OR.X.GT.226)RETURN
      GO TO 100
50    IF(X.LT.207.OR.X.GT.222)RETURN
      GO TO 100
52    IF(X.LT.209.OR.X.GT.220)RETURN
      GO TO 100
54    IF(X.LT.210.OR.X.GT.222)RETURN
      GO TO 100
56    IF(X.LT.211.OR.X.GT.224)RETURN
      GO TO 100
58     IF(X.LT.212.OR.X.GT.225)RETURN
      GO TO 100
60    IF(X.LT.213.OR.X.GT.225)RETURN
      GO TO 100
10    IF(Y.GT.5.8)GO TO 20
      IF(Y.GT.5.6)GO TO 22
      IF(Y.GT.5.4)GO TO 24
      IF(Y.GT.5.2)GO TO 26
      IF(Y.GT.5.0)GO TO 28
      IF(Y.GT.4.75)GO TO 30
      IF(Y.GT.4.55)GO TO 32
      IF(Y.GT.4.25)GO TO 36
      IF(Y.GT.4.0)GO TO 38
      IF(Y.GT.3.75)GO TO 40
      IF(X.LT.218.OR.X.GT.235)RETURN
      GO TO 100
20    IF(X.LT.208.OR.X.GT.223)RETURN
      GO TO 100
22    IF(X.LT.209.OR.X.GT.223)RETURN
      GO TO 100
24    IF(X.LT.210.OR.X.GT.224)RETURN
      GO TO 100
26    IF(X.LT.211.OR.X.GT.225)RETURN
      GO TO 100
28    IF(X.LT.212.OR.X.GT.226)RETURN
      GO TO 100
30    IF(X.LT.213.OR.X.GT.226)RETURN
```

```
      GO TO 100
32    IF(X.GT.226)RETURN
      GO TO 100
36    IF(X.GT.228)RETURN
      GO TO 100
38    IF(X.LT.216.OR.X.GT.230)RETURN
      GO TO 100
40    IF(X.LT.218.OR.X.GT.233)RETURN
100    CALL PLOT(99)
      HID=1
      RETURN
      END
```

```
      SUBROUTINE PYLON(XP,YP,PYD,PYA,JPY,OD,OA,NPYS,B,LPY,JB,OUT,IFORE,
     2ITREE,IOD,IOA)
      REAL*8 AHT,PHT,D,PHIP
      DIMENSION PDIS(50),LPY(50),X(50),Y(50),PYD(50),PYA(50),JPY(50),
     2JB(5,240),OUT(5,240),XP(5,12),YP(5,12),B(50,50)
      IYES=0
      DO 10 I=1,NPYS
      IF(JPY(I).GE.99998)GO TO 8
      D1=OD-PYD(I)
      D2=OA-PYA(I)
      D=SQRT(D1*D1+D2*D2)*.25
      IPYD=PYD(I)
      IPYA=PYA(I)
      PHT=B(IPYD,IPYA)
      AHT=B(IOD,IOA)
      CALL TOWN2(AHT,PHT,D,PHIP)
      D3=D
      PHI=PHIP*180/3.14159
      J=JPY(I)
      LP=LPY(I)
      IF(LP.GT.JB(1,J).AND.LP.GT.JB(I,J).AND.LP.GT.JB(3,J).AND.LP.GT.
     2JB(4,J).AND.LP.GE.JB(5,J))GO TO 20
      IF(LP.EQ.JB(1,J).OR.LP.EQ.JB(2,J).OR.LP.EQ.JB(3,J).OR.LP.EQ.
     2JB(4,J).OR.LP.EQ.JB(5,J))GO TO 20
      IF(LP.LT.JB(1,J))GO TO 50
      IF(LP.LT.JB(2,J))GO TO 55
      IF(LP.LT.JB(3,J))GO TO 60
      IF(LP.LT.JB(4,J))GO TO 65
      GO TO 75
50    IF(PHI.GE.OUT(1,J))GO TO 20
      GO TO 8
55    IF(PHI.GE.OUT(2,J))GO TO 20
      GO TO 8
60    IF(PHI.GE.OUT(3,J))GO TO 20
      GO TO 8
65    IF(PHI.GE.OUT(4,J))GO TO 20
      GO TO 8
75    IF(PHI.GE.OUT(5,J))GO TO 20
8     X(I)=99999.
      GO TO 10
20    X(I)=JPY(I)
      IF(IFORE.EQ.2)GO TO 21
      IF(ITREE.EQ.1)GO TO 22
      IF(X(I).LT.34.AND.X(I).GT.11)GO TO 10
      GO TO 21
22    IF(X(I).GT.207.AND.X(I).LT.230)GO TO 10
21    IYES=IYES+1
      H=PHI*.3
      Y(I)=H+5.
```

```
      D3=D3*.001
      PDIS(I)=.60
      IF(D3.LE.1)GO TO 15
      DO 17 J=2,20
      PDIS(I)=PDIS(I)-0.03
      IF(D3.LE.J)GO TO 15
17    CONTINUE
15    CALL PLOT(98,X(I),Y(I))
      DO 30 J=1,5
      DO 30 K=1,12
      IF(XP(J,K).GE.98.)GO TO 29
      CALL PLOT(90,X(I)+XP(J,K)*PDIS(I),Y(I)+YP(J,K)*PDIS(I))
      GO TO 30
29    CALL PLOT(99)
30    CONTINUE
10    CONTINUE
      DO 45 I=1,NPYS
      MLINE=0
      IF(X(I+1).GT.99998.)GO TO 101
      IF(X(I).GE.99998.)GO TO 101
      X1=X(I)-3.*PDIS(I)
      X3=X(I+1)-3.*PDIS(I+1)
      X2=(X3+X1)/2.
      Y1=Y(I)+.5*PDIS(I)
      Y3=Y(I+1)+.5*PDIS(I+1)
      Y2=((Y1+Y3)/2.)-.15
      Y23=Y2-Y3
102   XP23=X2*X2-X3*X3
      X23=X2-X3
      Y12=Y1-Y2
      XP12=X1*X1-X2*X2
      X12=X1-X2
      BB=(XP12*Y23-XP23*Y12)/(XP12*X23-XP23*X12)
      A=(Y12-(BB*X12))/XP12
      C=Y1-A*X1*X1-BB*X1
      CALL PLOT(90,X1,Y1)
      XH=X1
      WRITE(6,104)A,BB,C,X1,X2,X3,Y1,Y2,Y3,Y23,XP23,X23,Y12,XP12,X12
104   FORMAT(' ',8F15.5/7F15.5)
      IF(X1.GT.X3)GO TO 110
      DO 100 J=1,200
      XH=XH+1
      YH=A*XH*XH+BB*XH+C
      CALL PLOT(90,XH,YH)
      IF(XH.GE.X3)GO TO 103
100    CONTINUE
110   DO 150 J=1,200
      XH=XH-1
      YH=A*XH*XH+BB*XH+C
```

```
      CALL PLOT(90,XH,YH)
      IF(XH.LE.X3)GO TO 103
150   CONTINUE
103   IF(MLINE.EQ.3)GO TO 45
      X1=X(I)+3.*PDIS(I)
      X3=X(I+1)+3.*PDIS(I+1)
      X2=(X3+X1)/2.
      MLINE=3
      CALL PLOT(99)
      GO TO 102
101   CALL PLOT(99)
45    CONTINUE
70    RETURN
      END
```

```
      SUBROUTINE CLOUD(ITREE,YTOP)
      K=1
      L=11
      C=10.
      D=7.
      A=10.
      B=.625
      J=1
    5 DO 10 I=K,250
      X=I
      XX=I
      IF(ITREE.EQ.2)XX=241-I
      IF(I.GT.L)GO TO 15
      YYY=(1.-(X-C)*(X-C)/(A*A))*B*B
      IF(YYY.LT.0.0)GO TO 10
      Y=D+SQRT(YYY)
      CALL PLOT(90,XX,Y)
   10 CONTINUE
   15 CALL PLOT(99)
      IF(J.LT.3)GO TO 17
      DO 11 I=K,250
      X=I
      IF(I.GT.L)GO TO 16
      YYY=(1.-(X-C)*(X-C)/(A*A))*B*B
      IF(YYY.LT.0)GO TO 11
      Y=D-SQRT(YYY)
      CALL PLOT(90,X,Y)
   11 CONTINUE
   16 CALL PLOT(99)
   17 J=J+1
      IF(J.EQ.2)GO TO 20
      IF(J.EQ.3)GO TO 30
      IF(J.EQ.4)GO TO 40
      IF(J.EQ.5)GO TO 50
      IF(J.EQ.6)GO TO 60
      GO TO 80
   20 K=8
      L=47
      A=35
      B=1.1
      C=42.
      D=7.25
      GO TO 5
   30 K=36
      L=61
      A=20
      B=.5
      C=57
      D=7.85
```

```
      GO TO 5
39    J=4
40    K=46
      L=74
      A=15.
      B=.45
      C=66.
      D=8.3
      GO TO 5
50    K=70
      L=102
      A=17.
      B=.55
      C=86
      D=8.4
      GO TO 5
60    K=100
      L=144
      A=28.
      B=.75
      C=125
      D=8.25
      GO TO 5
80    RETURN
      END
```