NSG-7184

RENSSELAER POLYTECHNIC INSTITUTE
KNOWLEDGE AND THOROUGHNESS
1824

Rensselaer Polytechnic Institute

Troy, New York 12181

R.P.I. Technical Report MP-50

COMPUTER SIMULATION AND EVALUATION OF EDGE DETECTION ALGORITHMS AND THEIR APPLICATION TO AUTOMATIC PATH SELECTION

by

Betsy A. Longendorfer

NASA Grant NSG-7184

November 1976
Rensselaer Polytechnic Institute
Troy, New York

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author wishes to express her most profound gratitude to Professor Dean K. Frederick for the guidance and encouragement received under his supervision.

The author would also to to thank Professor Shen for providing access to the Rapid Estimation Scheme.

Special thanks are extended to the staff of the Computer Center, especially Lillian Jankowski, for assistance in the maintenance of this large computer simulation. The author is also grateful to Western Electric Engineering Research Center for providing the financial support required by the author to complete this project.

The project itself was supported by the National Aeronautics and Space Administration grant NSG 7184 as a joint faculty-student project at Rensselaer Polytechnic Institute.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The construction of an autonomous roving vehicle requires the development of complex data-acquisition and processing systems, which determine the path along which the vehicle travels. Thus, a vehicle must possess algorithms which can (1) reliably detect obstacles by processing sensor data, (2) maintain a constantly updated model of its surroundings, and (3) direct its immediate actions to further a long range plan.

The first function consisted of obstacle recognition. Obstacles may be identified by the use of edge detection techniques. Therefore, the Kalman Filter was implemented as part of a large scale computer simulation of the Mars Rover. Aditional edge detection algorithms were developed to deal with several problem situations, and the effects of parameter changes on the algorithms were studied. The algorithm proved to be rather reliable at ranges of 8 to 25 meters, even in the presence of noise or sloped surfaces.

The second function consisted of modeling the environment. The obstacle must be reconstructed from its edges and the vast amount of data must be organized in a readily retrievable form. Therefore, a Terrain Modeller was developed which assembled and maintained a rectangular grid map of the planet. It correctly identified all obstacles based on flat terrain but behaved unacceptably on slopes.

The third function consisted of directing the vehicle's actions. The grid map prepared by the Terrain Modeller was used in the classification of routes as acceptable or unacceptable, optimal or otherwise. A Path Selection Algorithm which navigated solely with the aid of the map was successfully demonstrated on flat obstacle-strewn terrain corrupted by noise.

Each of these algorithms require a large amount of computer time. Thus, this approach should be used primarily to determine a general steering direction, leaving an efficient short range sensor to map out a detailed route.

1. INTRODUCTION

The space program has recently generated interest in the construction of a roving vehicle for the exploration of other planets. The communications lag between Mars and Earth (on the order of 20 minutes), besides the low bit rate available, would compel the vehicle to be fairly self-sufficient. Therefore, the roving vehicle must be capable of:

(1) sensing its environment

(2) detecting obstacles and storing data in a readily retrievable form, and

(3) directing its immediate actions to achieve some long range goal.

The vehicle obtains information about its environment via a laser range-finder mounted on a mast attached to the vehicle's front axle. It provides range measurements given azimuth and elevation angles. Only medium sensor ranges, approximately 10 to 30 meters, are investigated.

1.1 Historical Review

The problems of path selection and obstacle detection by an autonomous roving vehicle have already been examined extensively. Krajewski[1]* developed a short range system, with an intended range of 0 to 5 meters, which employed laser triangulation techniques. Although results were promising. there were problems. Slopes were confused with obstacles and a less-than-optimal path to target often resulted, due to a path selection decision based solely on local features. Thus, the development of

* Superscripts refer to the reference number

alternate approaches was encouraged.

Short and medium range schemes based on range comparison techniques, such as those by Matthews[2] and Sharp[3], also suffered from noise and the confusion of slopes with obstacles.

Because of the failure of simple range comparison schemes, several edge detection algorithms were developed. These included Reed's[4] Four Dimensional Ratio, and the Kalman Filter and Rapid Estimation Scheme developed by Sonalkar and Shen.[5] The Four Dimensional Ratio was not successful in identifying sudden changes in terrain gradient, as is the case with the leading edge of a positive obstacle (boulder) on the trailing edge of a negative obstacle (crater). However, the Rapid Estimation Scheme (RES) proved to be successful in this regard. Implementation of RES (see for example, Leung[6]) offered two more major advantages: (1) the distinguishability of discrete obstacles and terrain slope and (2) an improved probability of selecting a globally optimal path because of the increased information available.

Path selection algorithms also were approached in a more mathematical way. A grid map of the type eventually employed was first developed by Lee[7] and modified by Lallman.[8]

1.2 Project Summary

The sensing function of the roving vehicle was implemented in the digital computer simulation with the addition of two programs: a laser range-finding sensor and a scan generator.

The obstacle recognition function was implemented with various edge detection schemes such as the Kalman Filter. Other edge detection algorithms were developed in response to operational problems experienced by the Kalman Filter. Noise sensitivity and parameter sensitivity were studied. For the first time, the obstacle detection block was physically separated from the terrain modelling block. The Terrain Modeller, which reconstructed an obstacle from its edges or parts of its edges, was developed. The Terrain Modeller also served as the system's memory by assembling and maintaining a rectangular grid map of the local planet environments.

The path selection function of the roving vehicle remained the eventual objective, however. A Path Selection Algorithm was developed for use with the grid map maintained by the Terrain Modeller.

1.3 Description of the Computer Simulation

The digital computer simulation of the vehicle is organized into five major modules, interfaced as shown in Figure 1.

1.3.1 The Vehicle Dynamics Block

The Vehicle Dynamics Block is described by Sharp[10]. It simulates the vehicle's movement across specified terrain, given the path selection decision concerning vehicle heading, velocity, and transit time or distance. The behicle is modelled as a point mass with a front wheel base of finite dimensions.

1.3.2 The Sensor Block

The Sensor Block consists of a laser range-finding sensor mounted on a 1.5 meter mast attached to the front axle of the vehicle. It supplies true or noise-corrupted range measurements for specified values of azimuth and elevation angles. A scan generator sweeps the sensor through a

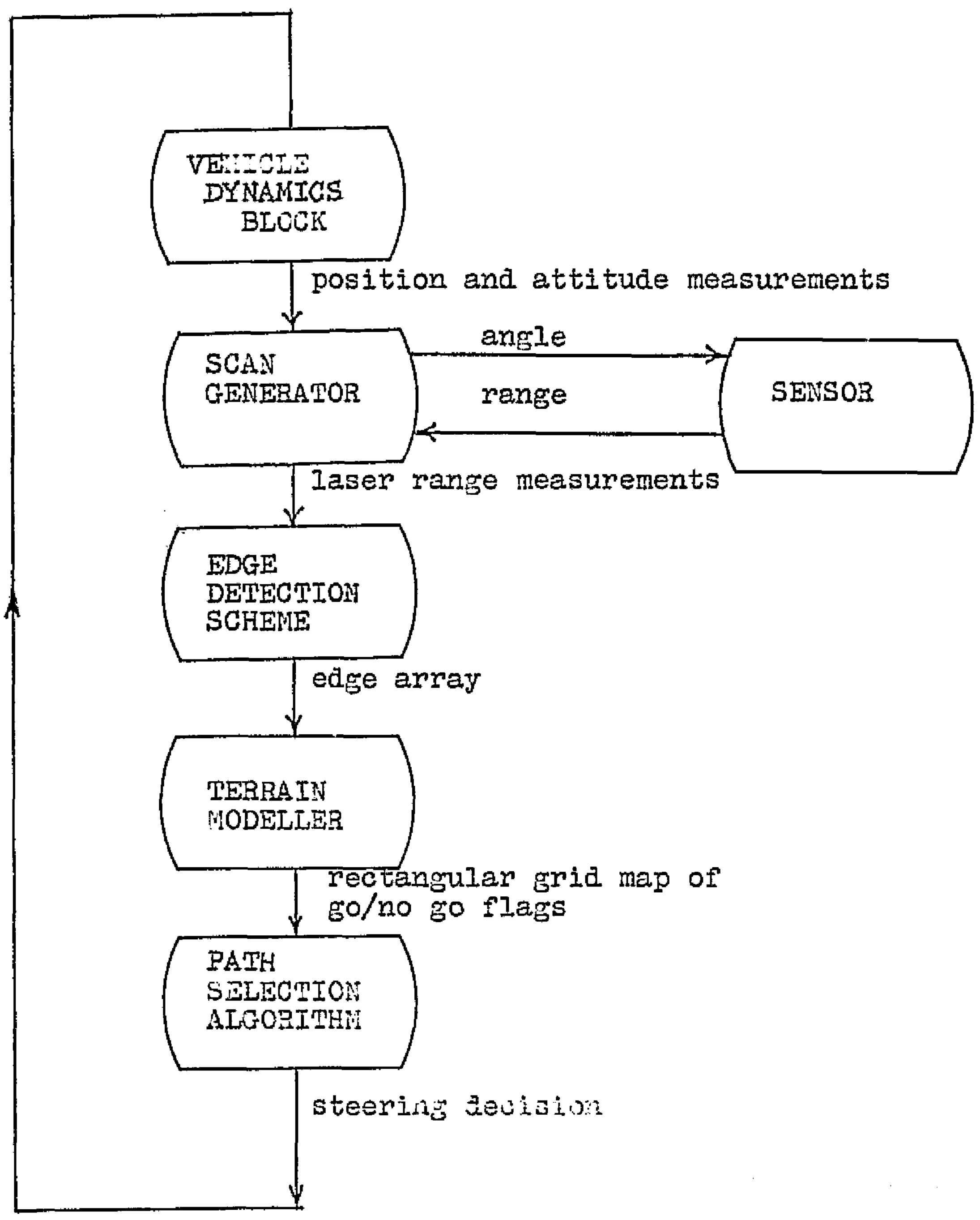


Figure 1. Main Program Modules of the Mars Simulation Package

particular field of vision and maintains the required data density.

1.3.3 The Edge Detection Scheme

The Edge Detection Scheme must simplify the matrix of range data to an array in which only the edges are indicated.

The Kalman Filter comprises the backbone of the edge detection module but additional processing, which was developed in response to problems arising during the simulation effort, is included in this general category. Examples of the additional processing are (1) a noise filter which removes edges without a certain number of adjacent edges and (2) a "moving average" or type of second difference processing.

1.3.4 The Terrain Modeller

The Terrain Modeller processes the matrix of edges in order to identify the obstacle location with respect to the edge. It also creates a rectangular grid map of the local planet surface, based on attitude and range data. The map serves as the system's memory. The map is continuously updated with information on target location, vehicle location, scanned and unscanned (unknown) regions, obstacle location and type, and the current average terrain height of each block of terrain.

1.3.5 The Path Selection Algorithm

A path selection algorithm based on the grid map approach of Lee[7] and Lallman[8] was developed. However, a more sophisticated blocking scheme was employed which identified dead-ends without previous knowledge of the location of all the obstacles. A steering decision, velocity, and travel distance are output to the vehicle dynamics block. Due to time constraints, the path selection algorithm described here has been only minimally employed in conjunction with the remainder of the simulation, but its feasibility as an approach has been demonstrated in test situations.

The following chapters are devoted to more detailed descriptions of each simulation block.

2. THE SENSOR

The Martian Roving Vehicle possesses a laser range-finding sensor, mounted on a 1.5 meter mast attached to the front axle. The laser is able to focus in several directions, which are defined by elevation angles and azimuth angles. The actual directions are determined partially by hardware constraints and partially by software data requirements.

The sensor was implemented within the large scale digital computer simulation of the Martian Roving Vehicle. This implementation operates in an extremely flexible manner, in contrast to previous sensor algorithms.

2.1 Hardware and Software Constraints

A laser range-finder is subject to many hardware and data processing constraints. The range-finder must focus at rather precise aximuth and elevation angles because of the large range error introduced by a comparatively small angular error. The laser must be able to change its focusing angles by motion of the laser, its mast, or a focusing mirror, or by simulating motion through replication of the sensing devices. Accurate range measurements must then be obtained. Thus, the data scan should be essentially instantaneous, or the vehicle must remain stationary, especially if a large field is being scanned. A convention for obtaining range values in cases where no laser signal is returned should be established. Finally, the data density must be maintained within certain parameters.

2.2 Implementation

The digital computer simulation divided the sensor functions described above into two classes: scan generation functions and data sensing functions. The scan generation functions are incorporated into MIDSCN, and the sensing functions are simulated in SENSRL. (Both MIDSCN and SENSRL are documented in Reference 9.)

2.2.1 The Scan Generator: MIDSCN

MIDSCN implements the mid-range scanning scheme which obtains the pattern of data required by the Kalman Filter and the Rapid Estimation Scheme (RES).

As input, RES requires range data obtained using constant angular spacing. The aximuth and elevation angle increments are not necessarily equal. Constant angular spacing means that the density of data points per unit of terrain will vary considerably at large elevation angles. MIDSCN accepts data density control input in three forms: (1) the actual values of the angles may be listed, (2) maximum, minimum, and incremental angles may be given, or (3) the length, width, maximum point separation and distance to center of the scanned field may be specified. Once one form of input has been received, all of the parameters listed above are calculated and made available to other program simulation blocks.

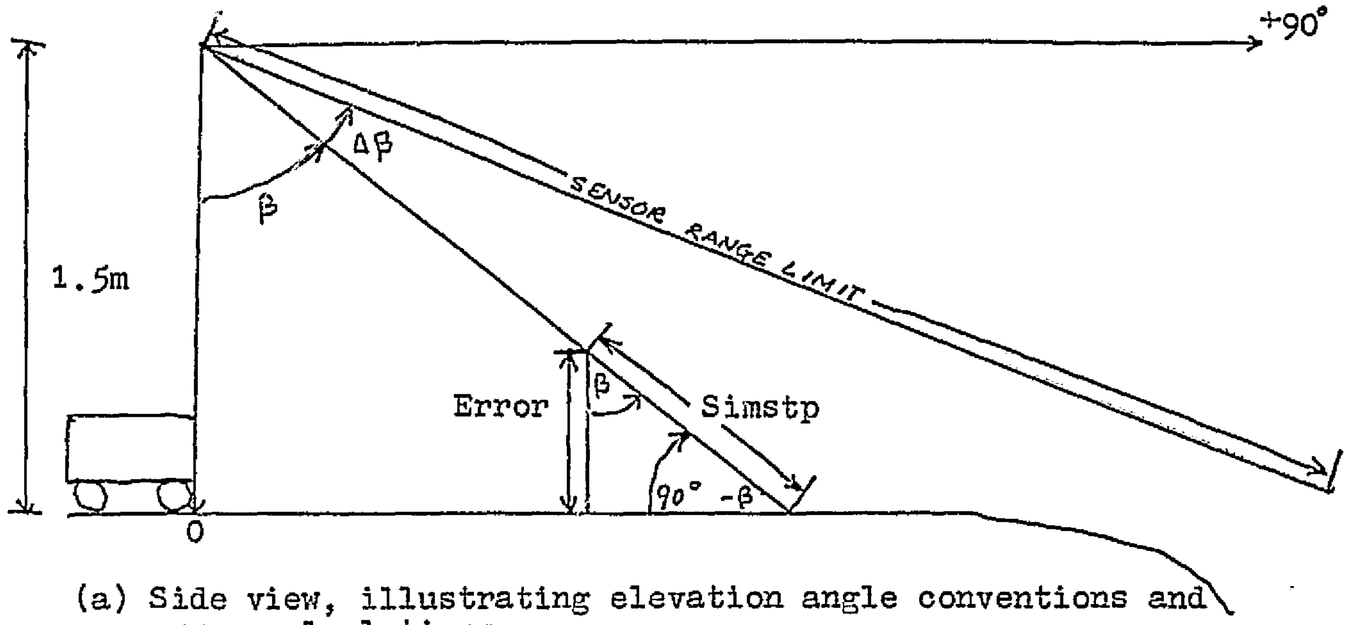
The sensor calculates the ranges for all azimuths of a particular elevation angle on any single call. The scan generator must call the sensor for each elevation angle and provide for time lapse before relinquishing control to the next program module. The scan generator also sets a flag indicating whether the current position is the same as the previous position. The flag allows other parts of the simulation to become more computationally efficient.

2.2.2 The Sensor: SENSRL

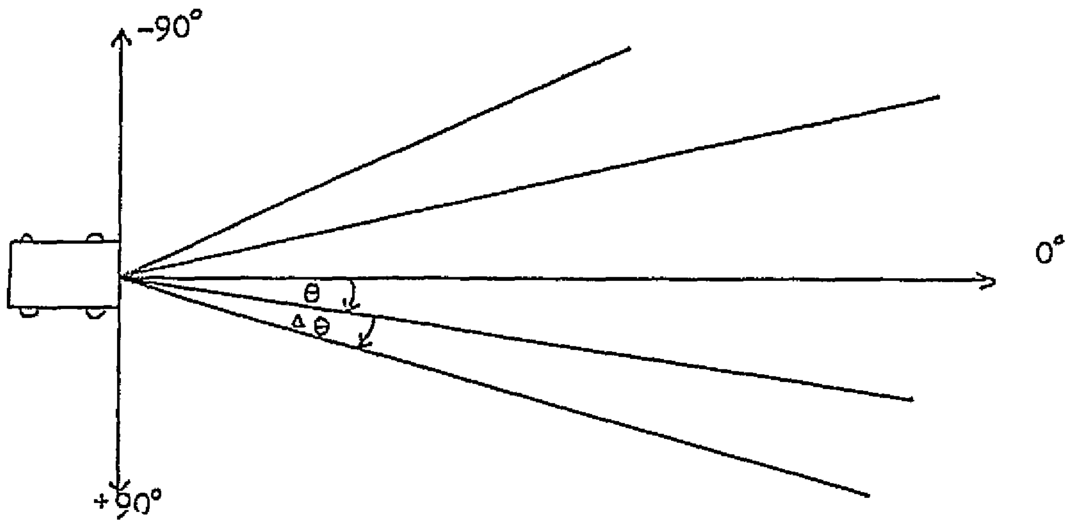
SENSRL is a laser range-finding sensor which may operate with up to fifty different azimuths and fifty different elevation angles simultaneously. Azimuths may take any value in the interval [-90°, +90°], while elevation angles are restricted to the interval [0°, +90°]. The angle conventions are illustrated in Figure 2.

The sensor algorithm is implemented as follows. The vehicle attitude is calculated and the transformation equations from the vehicle frame of reference to the planet frame are obtained. The transformation equations are used to find the true planet locations of the laser and the point at which its beam would strike ground, assuming perfectly level terrain. The line joining these two points in the planet frame is drawn. The sensor steps along the beam from the laser with a small user-specified increment until it detects a position below the local ground level, or reaches the limit of its sensing range. In the first case, iterations are performed using the bisection method until sufficient accuracy is attained. "Sufficient accuracy" is user-defined as the error in the range measurement SIMSTP, which must be internally converted to an error in terrain heights, ERROR, as shown in Figure 2. In the second case, the range is set equal to the sensor limiting range.

For computational efficiency, an initial guess of the range value is employed for all elevation angles except the smallest. The initial guess equals the range calculated for the greatest elevation angle smaller than



(a) Side view, illustrating elevation angle conventions and error calculations



(b) Top view, illustrating azimuth angle conventions

Figure 2. Sensor Angle Conventions

the current one, at identical azimuth angles. If the sensor finds itself below the terrain level, a constant is repeatedly subtracted from the initial guess until the terrain level is reached.

Noise may be added independently to each range measurement. The noise may be uniformly distributed, or it may be filtered with specified mean, maximum, deviation and filter constants.

Improvements in the sensor simulation are discussed in Section 6.1.1.

3. THE EDGE DETECTION SCHEME

The term "edge detection scheme" refers to a collection of edge detection algorithms which may be employed separately or simultaneously.

The Kalman Filter edge detection algorithm received primary attention because it had already demonstrated a high degree of success in dealing with isolated, well-defined obstacles. Noise sensitivity, parameter sensitivity, effective range, and minimum obstacle size were determined. These studies led to the development of "Average Processing", a second edge detection algorithm which may be employed alone, or in concert with the Kalman Filter. In addition, a noise filter was developed for use with either the Kalman Filter or Average Processing.

Because of the extensive mathematical theory leading to the Kalman Filter, a brief theoretical summary is first provided. Each edge detection algorithm is then explained in detail as implemented, and the results obtained are discussed.

3.1 Theoretical Discussion

The theoretical basis of the edge detection algorithms is discussed below, in preparation for a description of the implementation of each algorithm.

3.1.1 The Kalman Filter and Rapid Estimation Scheme

The theoretical results summarized below are discussed in depth by Sonalkar[11].

The Kalman Filter processes a large matrix of data, and attempts to detect changes in magnitude or in gradient between adjacent elements.

Range data obtained from the sensor is stored in a matrix whose top row corresponds to the largest elevation angle, and hence whose ranges represent the greatest horizontal distance from the vehicle. The bottom row corresponds to the smallest elevation angle and the least horizontal distance. The first column corresponds to the vehicle's leftmost (or most negative) azimuth angle, and the last column to the vehicle's rightmost (or largest positive) azimuth angle.

The matrix must be completely processed twice, once row-by-row, and once column-by-column, to detect magnitude or gradient changes in one of the directions.

Columns are processed first. Each column is processed separately since it is assumed to be independent of all of the other columns. Each element of a column corresponds to a stage of the calculation. A two component state vector is defined for the i^{th} stage of the calculation. It consists of x_1, the range estimate, and x_2, the estimated change in x_1, obtained from the difference between the present noisy range measurement z_i and the previous noisy range measurement z_{i-1}. The state equation is:

$$\underline{x}_{i+1} = F\,\underline{x}_i = \begin{bmatrix} 1 & 1 \\ 0 & f_i \end{bmatrix} \underline{x}_i \tag{3-1}$$

where

$$\underline{x}_i = \begin{bmatrix} d_i \\ g_i \end{bmatrix} \tag{3-2}$$

and where d_i is the laser range estimate at the i^{th} stage, g_i is the difference between the range measurement at the $(i+1)^{st}$ stage and the i^{th} stage, and f_i is the multiplicative factor relating g_{i+1} and g_i. Thus,

$$\varepsilon_{i+1} = f_i \varepsilon_i$$

The factor f_i has been analytically derived for a flat plane approximation of the planet surface. Figure 3 illustrates the relationship between these parameters.

At the i^{th} stage, both the state estimate $\bar{x}_i$ and its error covariance P_i are calculated. The Kalman Filter is performed at each stage. It consists of four steps, followed by a Bayesian state estimate. The four steps of the Kalman Filter are prediction, innovation, calculation of the Kalman Gain, and correction. Each step and the Bayesian state estimate are discussed below in more depth, but the rest of the algorithm will first be explained.

After the completion of the i^{th} stage calculations, the $(i+1)^{st}$ stage becomes the i^{th} stage. When a column has been completely processed, a new series of calculations is begun on the next column, with no memory of the previous column.

After all of the columns have been processed, the procedure is repeated, using the rows for input. Row filtering is a much simpler process than column filtering because only a change in range, not a change in gradient is expected. Therefore, only the first element of the state vector is retained. There are situations where a gradient change might occur, such as when the vehicle is tilted significantly with respect to the local vertical. However, the Kalman Filter implemented here has not been designed to deal with such a situation. Modifications might be deemed necessary at some future date.

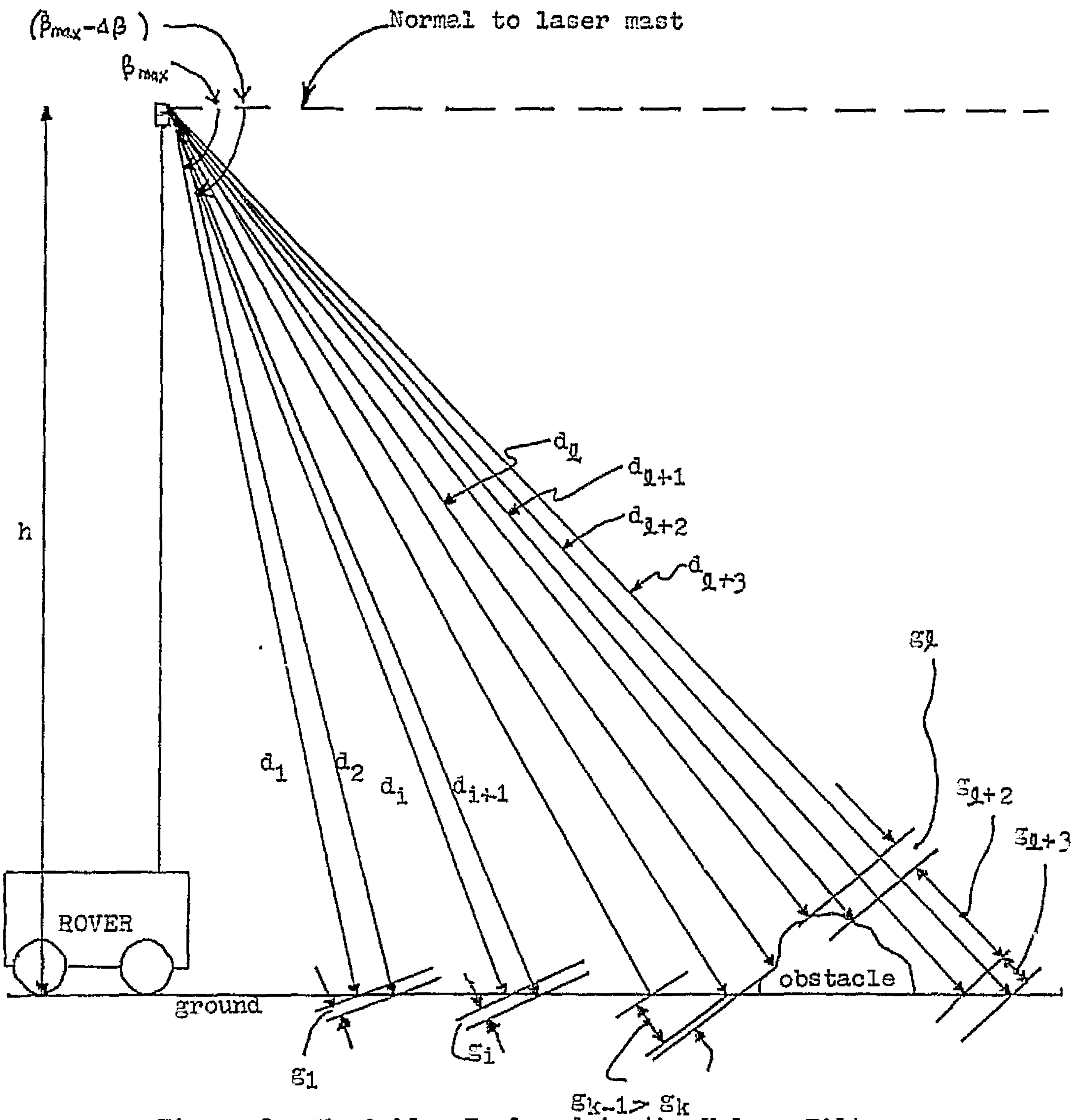


Figure 3. Variables Employed in the Kalman Filter

Prediction

Prediction is the process whereby a new state estimate is obtained from the previous state estimate. The defining equations are:

$$\bar{x}_{i+1} = E(x_{i+1}|Z_i) = F_i \hat{x}_i$$

$$M_{i+1} = E[(x_{i+1} - \bar{x}_{i+1})(x_{i+1} - \bar{x}_{i+1})^T]$$

$$= F_i P_i F_i^T + Q_i$$

where the variables are defined as in Table I.

The physical interpretation is as follows. Using equations (3-1) and (3-2), the state equations become

$$d_{i+1} = d_i + g_i$$

$$g_{i+1} = f_i g_i$$

Thus, the predicted range is the sum of the previous range and the range increment. The predicted range increment is proportional to the current range increment. The proportionality constant may be derived theoretically for a flat plane, based on the assumption of a constant angular increment during the measurement process.

The covariance matrix provides a measure of the allowed variation in the predicted state estimate.

Innovation

Innovation is the process whereby the unpredicted component of the state vector is obtained. This component will be compared in a later step with state estimates obtained from hypotheses about the edge location.

TABLE I

DEFINITION OF KALMAN FILTER VARIABLES

Variable	Name
B_ℓ	Bayes' risk for edge at ℓ^{th} stage
$c_{\ell m}$	Cost function for edge at ℓ^{th} stage, with the m^{th} hypothesis
d_i	Measured range
F_i	State transformation matrix, defined in Equation 3-1
g_i	Difference in range between successive measured ranges
H_i	Measurement matrix, defined as $[1\ 0]$
H_m	Bayes' risk hypothesis
K	Kalman Gain
I_n	Identity matrix of order n
M	Covariance of the predicted state vector, $\bar{x}_i$
P	Covariance of the state estimate $\hat{x}_i$
p_m	Prior probability of the presence of an edge at the m^{th} stage
Q	Plant noise
R	Observation noise covariance, usually 5 cm
W_i	Plant noise
$\bar{x}_i$	Predicted state estimate
$\hat{x}_i$	Corrected state estimate
y_i	Unpredicted component of state vector
z_i	Noisy range measurement

The defining equations are:

$$Y_{i+1} = Z_{i+1} - H_{i+1}\overline{X}_{i+1}$$

$$= Z_{i+1}\ [1\ 0] \begin{bmatrix} d_i \\ g_i \end{bmatrix}$$

$$= Z_{i+1} - d_i$$

$$W_{i+1} = E[y_{i+1}\ y_{i+1}^T]$$

$$= H_{i+1}\ M_{i+1}\ H_{i+1}^T + R_{i+1}$$

Thus, the estimated noise is simply the noisy measured range minus the predicted range. The covariance provides a measure of the noise contained in the state prediction.

Kalman Gain

The Kalman Gain is a factor which compensates for the state prediction error.

The defining equation is:

$$K_{i+1} = M_{i+1}\ H_{i+1}^T\ W_{i+1}^{-1}$$

Physically, the Kalman Gain is a measure of noise, or the presence of an edge.

Correction

Correction is the process whereby the state estimate is modified by the Kalman Gain factor.

The defining equations are:

$$\hat{x}_{i+1} = \bar{x}_{i+1} + K_{i+1}\, y_{i+1}$$

$$P_{i+1} = E[(x_{i+1} - x_{i+1})(x_{i+1} - x_{i+1})^T]$$

$$= (I_n - K_{i+1}\, H_{i+1})M_{i+1}(I_n - K_{i+1}\, H_{i+1})^T + K_{i+1}R_{i+1}K_{i+1}^T$$

Physically, the predicted state estimate is summed with the product of the Kalman Gain and the estimated noise component. The corrected state estimate is used in the hypothesis testing step to determine the presence or absence of an edge.

Bayesian State Estimate

After the Kalman Filter has been performed at the i^{th} stage, Bayes' Risk is used to indicate which of the three hypotheses is most probably true. A hypothesis is considered to be true if it has the minimum risk or cost function associated with it.

The three hypotheses are:

H_0: An edge occurs at the i^{th} stage

H_1: An edge occurs at the $(i+1)^{st}$ stage

H_2: An edge occurs beyond the $(i+1)^{st}$ stage.

Bayes' Risk for the ℓ^{th} stage is calculated as follows:

$$B_\ell = \sum_{m=0}^{2} p_m c_{\ell m}\, P(z|H_m); \quad \ell = 1,2,3$$

Bayes' Risk weights the probability that an edge occurred at a particular stage by the cost $c_{\ell m}$ of choosing or failing to choose each alternative, and by the prior probability of the occurrence of an edge. Each of the numbers p_m and $c_{\ell m}$ are arbitrarily chosen weights, which are however subject to a few rules stated by Sonalkar[11].

If H_0 is deemed correct (i.e., H_0 has the minimum risk B associated with it), the hypothesis is accepted and an edge is flagged. Otherwise, H_0 is rejected and the (i+1)st stage becomes the ith stage. The Kalman Filter calculations are repeated for the new ith stage.

3.1.2 Average Processing

The term "Average Processing" encompasses two separate procedures: (1) the calculation of maximum and minimum allowable range increments based on a maximum increment of in-path slope, and (2) a second-difference computational method.

The maximum and minimum tolerated range increments are computed for each pair of elevation angles, given the maximum permissible increment of in-path slope. The minimum range increment occurs with the maximum positive change in slope. The maximum range increment occurs with the maximum negative change in slope. If the maximum increment is larger than the limit of the sensor range, the maximum increment is set to the sensor limit minus the present range.

The equations for the maximum and minimum differences are derived below. Consider the largest triangle in Figure 4.

$$\begin{aligned} M &= 180° - (\theta + \Delta\,\theta) - (90° + s) \\ &= 90° - \theta - \Delta\,\theta + s \end{aligned}$$

where s is the maximum allowable slope increment, θ is the elevation angle, and $\Delta\theta$ is the elevation angle increment.

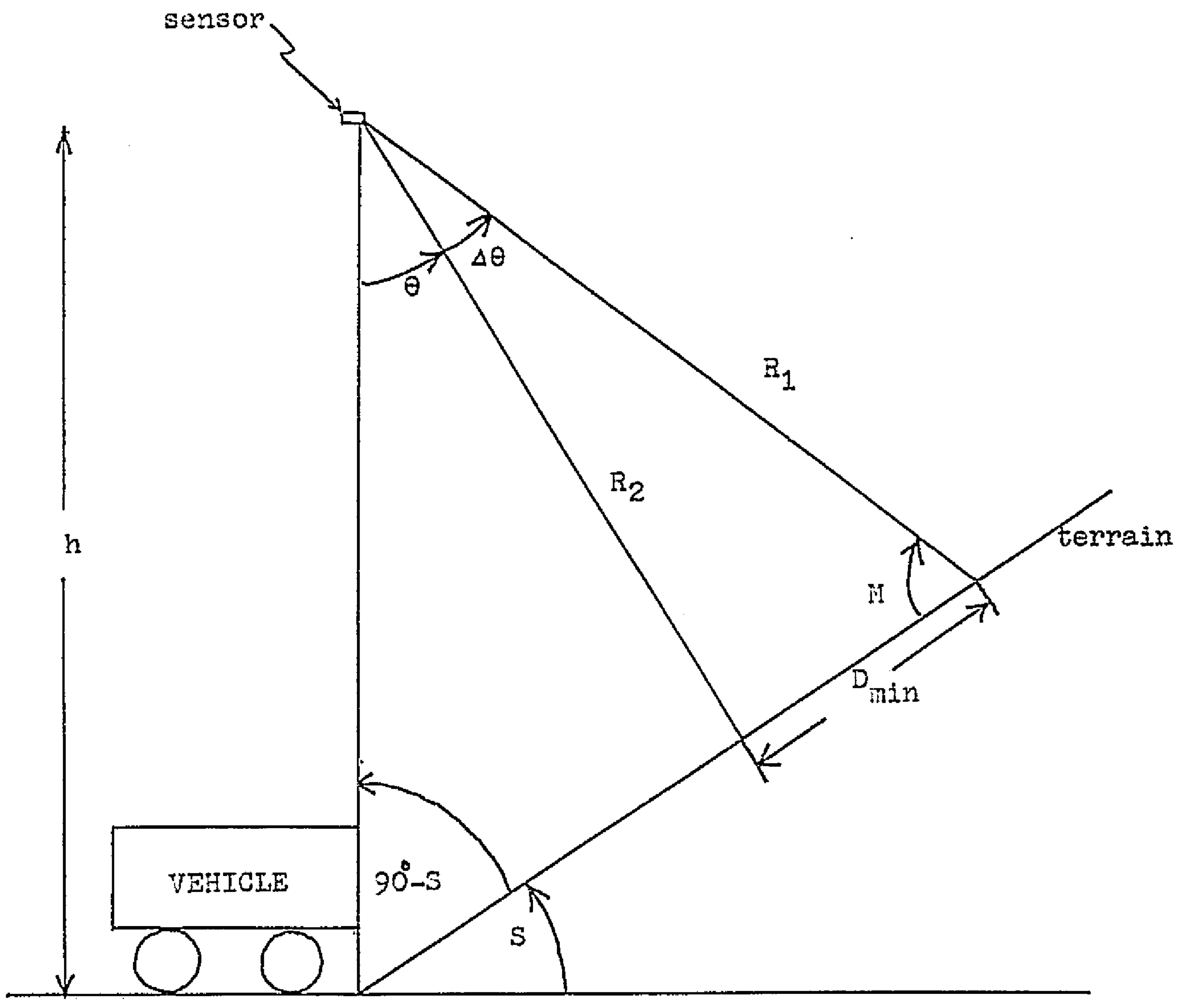


Figure 4. Calculation of Range Bounds During Average Processing

By the law of sines,

$$\frac{\sin\Delta\theta}{D_{min}} = \frac{\sin M}{R_2}$$

and

$$D_{min} = \frac{\sin\Delta\theta}{\sin M} R_2$$

where D_{min} is the minimum allowed range increment, and R_1 and R_2 are measured ranges.

To calculate the maximum range increment, note that the angle between the terrain level and the mast becomes $90° + s$. Therefore,

$$M = 90° - \theta - \Delta\theta - s$$

and

$$D_{max} = \begin{cases} \frac{\sin\Delta\theta}{\sin M} R_2 & ; \ M > 0 \\ R_{sensor\ limit} & ; \ M \leq 0 \end{cases}$$

where D_{max} is the maximum allowed range increment.

The other procedure denoted by "Average Processing" is a second difference calculation. The difference in range values for each successive pair of elevation angles is calculated at each stage. The difference is compared to an average difference calculated over the previous stages. If the absolute value of the difference of these values is greater than a specified fraction of the average difference, an edge is indicated. The average difference is updated in two ways. If an edge occurred, the average difference is set equal to the most recent difference. If no edge occurred, the average difference and the current difference are averaged with a positive integral weighting factor for the average difference and a weight of 1.0 for the current difference.

3.1.3 The Noise Filter

Noisy data sometimes resulted in the occurrence of spurious edges. It was assumed that the probability of adjacent spurious edges was small compared to the probability that an existing edge would not possess a neighbor. Therefore, an edge which did not possess an adjacent edge was declared spurious and was ignored.

3.2 Implementation

Implementation of the algorithms described above involved a diversity of tasks such as interfacing them with the existing computer simulation, testing their response to different types of terrain, designing additional processing to improve response, studying the effect of parameter changes, studying noise sensitivity, and discovering some optimal set of parameters useful for the path selection process.

3.2.1 The Kalman Filter Subroutine Package

The Kalman Filter subroutine package consists of eight distinct programs: KALMAN, KF, KF1, RES, PRDCT, ABAT, MATADD, and MATMUL. These are documented in Reference 9.

KALMAN contains the master logic for processing a matrix through the Kalman Filter, as described in Section 3.1. KF1 and RES are used for row processing only. They obtain new state and error covariances given the present state (this is the Prediction step). KF1 bases its calculations on the hypothesis that a jump occurred at the present stage, while RES uses the alternate hypothesis that a jump will occur at some later stage. Nine state estimates, denoted x_{jk}, are computed at the i^{th} stage. These are comprised of the state estimates for the i^{th}, $(i+1)^{st}$, and $(i+2)^{nd}$ stages (or j=1,2,3 respectively), based on each hypothesis H_0, H_1, and H_2 (or k=1,2,3 respectively). The predicted state estimates are processed through the remaining steps of the Kalman Filter and then compared using Bayes' Risk. The state estimate with minimum associated cost is chosen. If an edge occurred at the i^{th} stage, it is indicated within the matrix as an '0' or vertical edge, since it was identified by horizontal processing. If it is judged that an edge occurred at the $(i+1)^{st}$ or $(i+2)^{nd}$ stage, no edge is indicated. The $(i+1)^{st}$ stage becomes the i^{th} stage, and the entire process is repeated. The first state estimate in each row is the actual noisy measured range of the first column of that row.

Columns are processed similarly using PRDCT, which assumes a jump occurs at the present stage, and KF, which assumes that the jump occurred later. The first state estimate vector in each column consists of the measured ranges between the next-to-last row and the last row. State estimates are updated by Equation (3-1). The factor f_i has been analytically derived for a flat plane, assuming a constant increment in elevation angle between rows. If a horizontal edge is identified, it is indicated by a "*" (or an "X" in array positions where both horizontal and vertical edges were detected).

The remaining subroutines ABAT, MATADD, and MATMUL are utilities for matrix addition, multiplication, and the computation of terms of the form ABA^T, resepctively.

The Kalman Filter is controlled by the same input as the sensor: information about maximum, minimum, and incremental values of the azimuth and elevation angles. The only additional input needed is the parameter

UJUMP. Slope changes larger than UJUMP meters per meter will undergo additional processing to update the state and error covariance. Additionally, the programmer may change the costs and prior probabilities of the occurrence of and edge according to rules described by Sonalkar[11].

3.2.2 Average Processing

Test cases processed by the Kalman Filter indicated that detection of the near edge of a positive obstacle or the far edge of a negative obstacle is extremely sensitive to parameter values. Detection of these edges require a rather sparse placement of data points in order to obtain a large enough change in range or gradient. Hence, the size of an obstacle which could remain undetected might be unacceptably large. Average Processing attempted to overcome this difficulty by trying to detect those edges where only gradient changes occurred.

The theoretical description of Average Processing may be found in Section 3.1.2. Average Processing will not automatically be performed by the system, as the Kalman Filter is. If desired, it must be enabled by the user through a flag.

Average Processing is performed only on elements of the columns of the range matrix, simultaneously with the Kalman Filter. Each column is assumed to be independent of the others.

Average Processing requires a three-user specified parameters: S, THRESH, WEIGHT. S is a positive number indicating the maximum permissible increment in in-path slope. THRESH is a threshold value, usually set to 1.0. To set up the threshold test, the current difference is first subtracted from the average difference and the absolute value is computed. This absolute value is then compared to the average difference. If their ratio is larger than THRESH, an edge is indicated. WEIGHT is the weighting factor of the average difference relative to the current difference when updating the average difference.

During Average Processing, both the allowable slope calculation and the second-difference calculation are performed.

3.2.3 The Noise Filter

Tests performed on the Kalman Filter using noisy data indicated that spurious edges sometimes occurred. Therefore, the entire edge matrix was filtered again. Filtering the matrix as a whole is advantageous because of the availability of a global view rather than the simple single stage outlook of the Kalman Filter.

Each edge is examined individually. If an adjacent edge exists (in an adjacent row or column position only, not in a diagonal position), the edge is assumed to be correct. Otherwise, it is spurious and is therefore erased. Diagonal edges (those indicated by "X") are additionally examined for edges that are adjacent diagonally before erasure.

The noise filter option must be activated by a user-input flag INFIL. It is not a system default.

3.3 Results

Prior demonstrations of the Kalman Filter had always employed a single large obstacle on flat terrain in the center of the field of view. The type of obstacle to be detected, and its size and position were always known. The test cases shown here attempt to focus on other problems. Eventually, a completely unknown terrain must be satisfactorily scanned, in order for a path selection algorithm to be successful.

An upper bound on the size of undetected obstacles first had to be established. At ranges of 10 to 15 meters, with elevation angle increments of 0.1°, boulders of 0.5 meters diameter were detected by the Kalman Filter whereas those of 0.25 meter diameter were not. However, Average Processing was able to detect boulders of 0.25 meters in diameter, but failed for diameters of 0.125 meters. As a rule of thumb, the Kalman Filter requires about eight data samples from an obstacle before detection occurs, whereas Average Processing requires three (assuming no large, easily detectable edge is available).

Further range studies indicate that the optimal operating range for the Kalman Filter in this simulation is approximately 8 to 25 meters. At short ranges, the difference in ranges is large compared to the actual range. At long ranges, the differences are a very small fraction of the total range, or the diameter of the smallest detectable obstacle is rather large. Both of these effects diminish the effectiveness of the Kalman Filter.

Detection of the near edge of a boulder or the far edge of a crater requires a particular relationship between parameters. The laser height, horizontal distance to obstacle to be detected, and elevation angle increment must combine to yield a horizontal distance of approximately 0.5 meters between data points on level terrain. These parameter relationships are illustrated in Figure 5, where ΔD should be approximately 0.5 meters. Typically, the values are set to a laser height Z_L of 1.5 meters, a nominal scanning range D of 10 meters, and an elevation angle increment

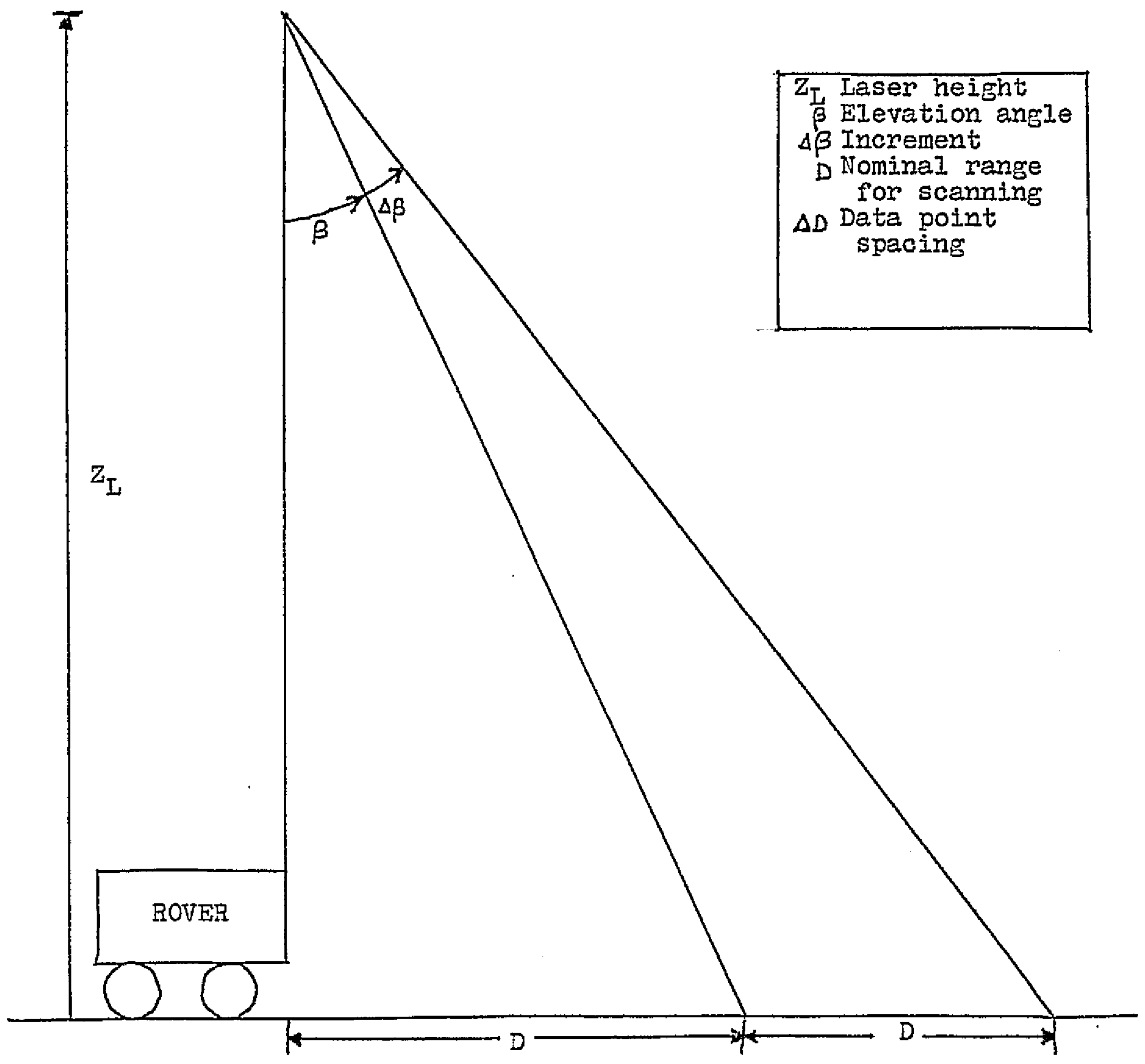


Figure 5. Parameter Relationships for Edge Detection With the Kalman Filter

$\Delta\beta$ of 0.4°. At 20 meters, the elevation angle increment $\Delta\beta$ is 0.2° with a laser height Z_L of three meters. Figure 6 shows the results of such a scan. Other results obtained with the Kalman Filter are also quite promising. Figures 7 and 8 show the results of scans over a cylindrical boulder and a crater, respectively.

Average Processing has been extremely effective in detecting the entire obstacle even when the ideal data point spacing described above is not used. Figures 9 and 10 show the results of Average Processing when applied to a boulder and a crater, respectively. Note the accurately rounded shape of the near and far edges. Values of 1.0 for THRESH, the edge threshold value and 3.0 for WEIGHT, the average weighting factor, have been shown to be the most effective. The process is extremely sensitive to the value of THRESH. In general, Average Processing is also sensitive to noise in the range measurements and should not be used where noise amplitude may exceed 0.2 meters.

The Kalman Filter was also tested with noise-corrupted range measurements, where the noise consisted of unfiltered white noise of a specified maximum amplitude. The approximate distance from the vehicle was 10 meters. Noise of 0.1 meter maximum amplitude (2%) did not affect edge detection results, as shown in Figure 11. Noise of 0.2 meter maximum amplitude (4%) could be successfully filtered by the noise filter. See Figures 12(a) and 12(b). Noise of 0.3 meter maximum amplitude (6%) began to cause spurious edges which could not be filtered out, as shown in Figures 13(a) and 13(b). Spurious edges consisted mainly of horizontal edges. This was not entirely unexpected. Horizontal edges result from a

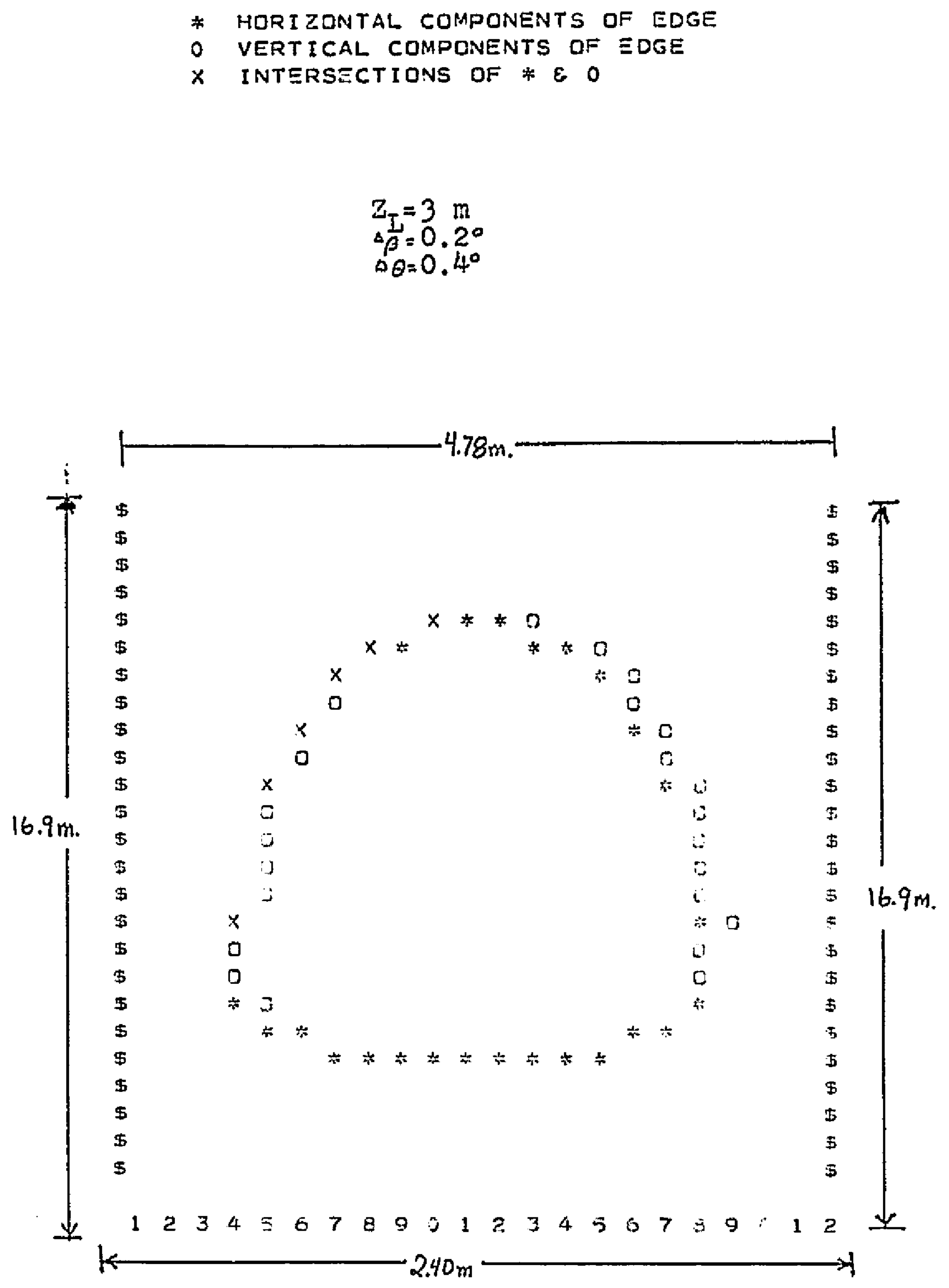
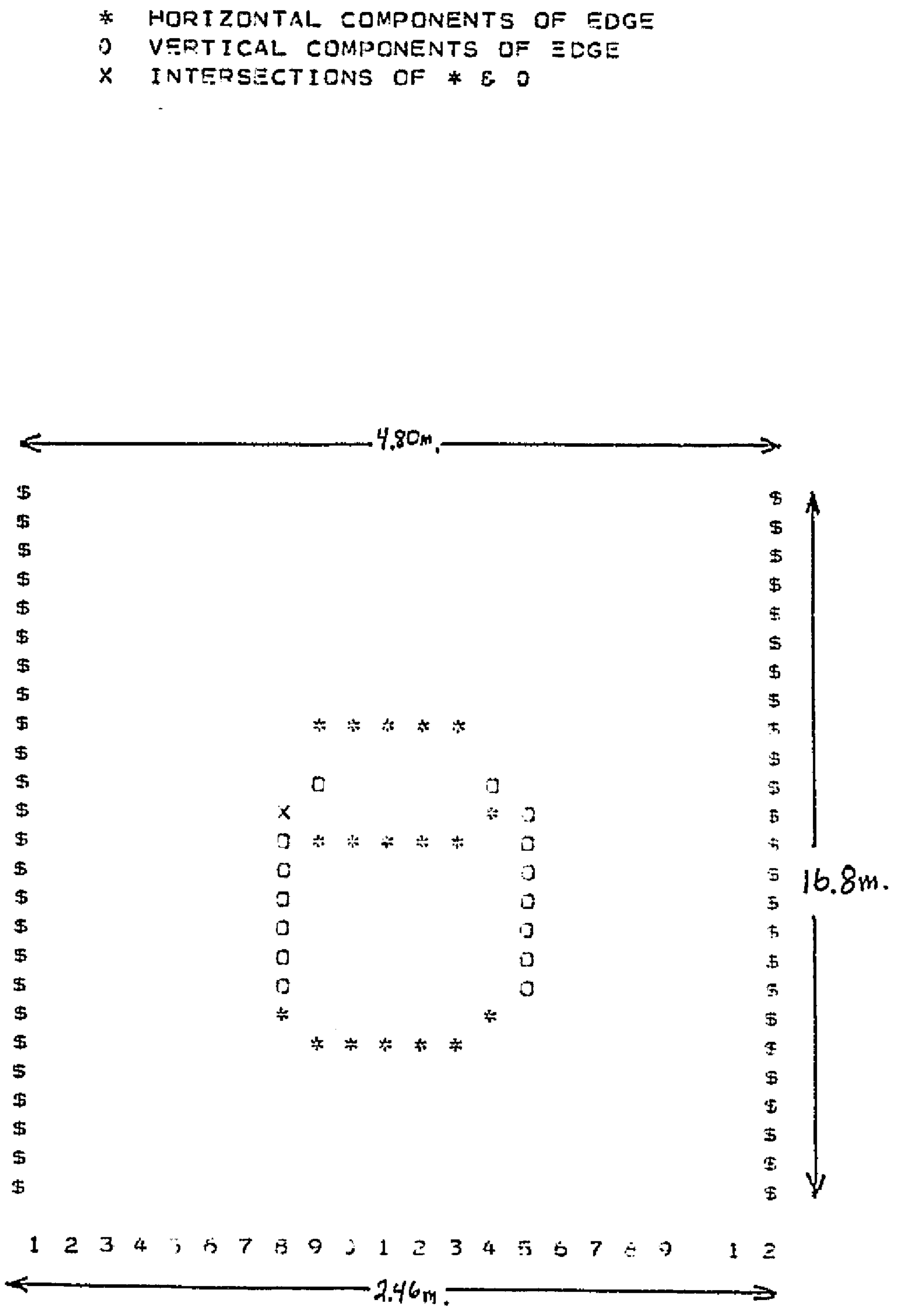


Figure 6. Two Meter Diameter Spherical Boulder at Twenty Meters



* HORIZONTAL COMPONENTS OF EDGE
0 VERTICAL COMPONENTS OF EDGE
X INTERSECTIONS OF * & 0
4.80m.
16.8m.
2.46m.
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 1 2

Figure 7. One Meter Diameter Cylindrical Boulder at Twenty Meters

```
*  HORIZONTAL COMPONENTS OF EDGE
O  VERTICAL COMPONENTS OF EDGE
X  INTERSECTIONS OF * & O
```

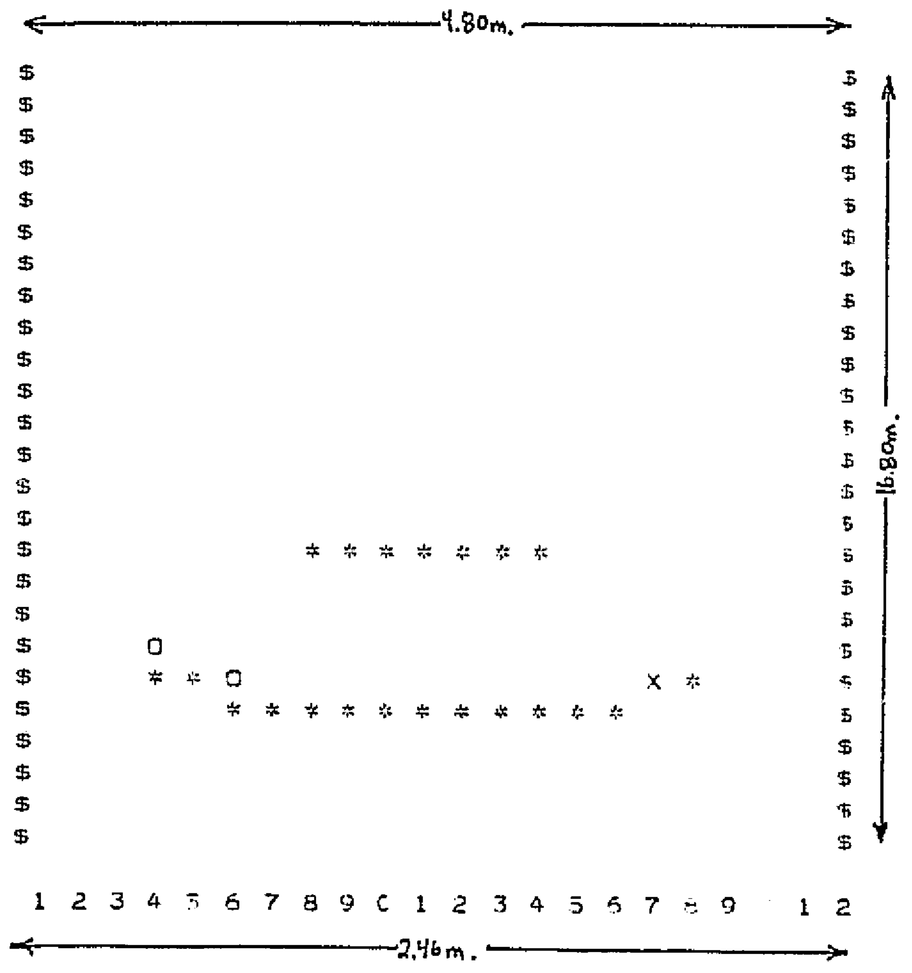


Figure 8. Two Meter Crater at Twenty Meters

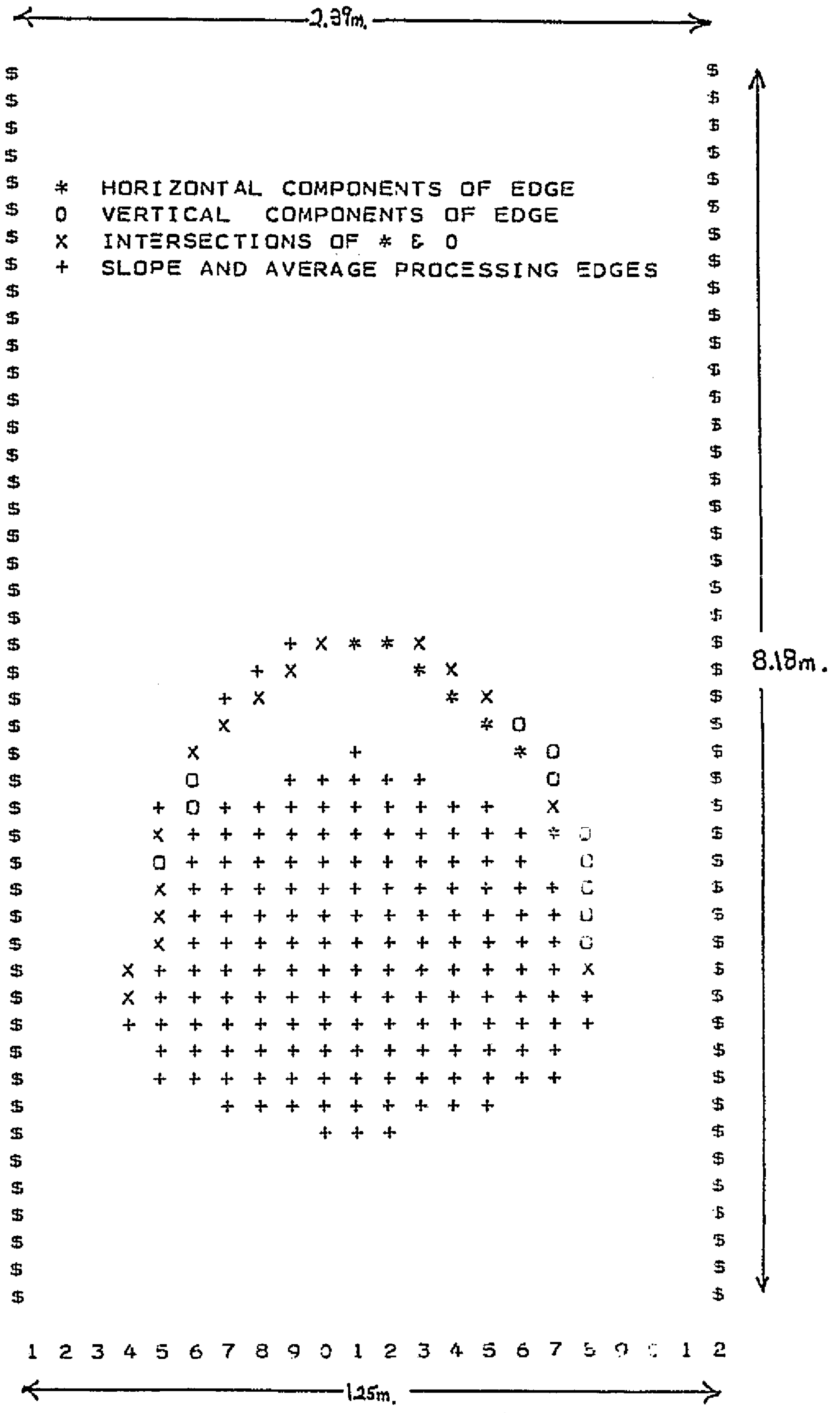

2.39m.
8.18m.
1.25m.
* HORIZONTAL COMPONENTS OF EDGE
O VERTICAL COMPONENTS OF EDGE
X INTERSECTIONS OF * & O
+ SLOPE AND AVERAGE PROCESSING EDGES
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

Figure 9. Average Processing Applied to a One Meter Diameter Boulder at Ten Meters

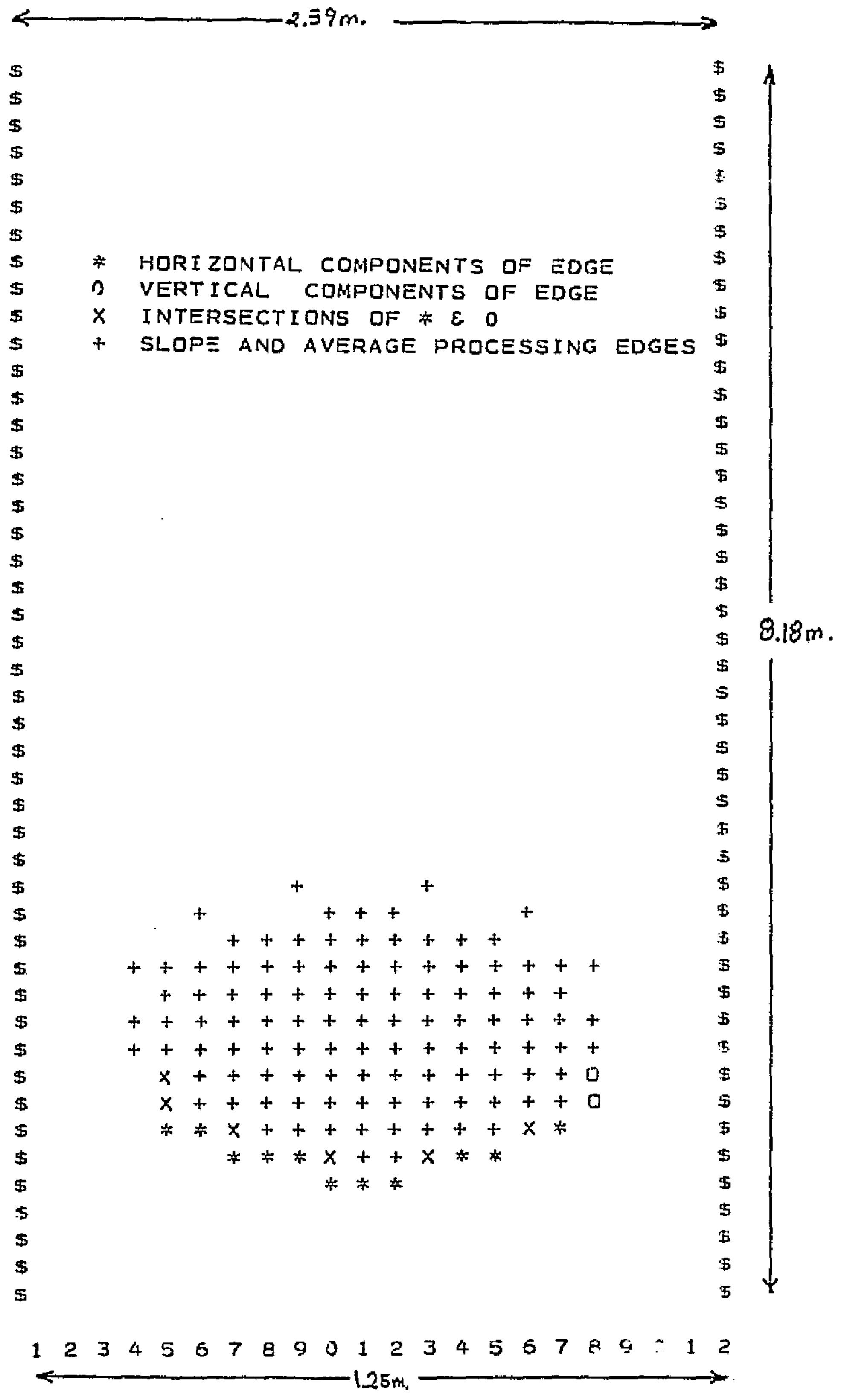

2.59m.
8.18m.
1.25m.
* HORIZONTAL COMPONENTS OF EDGE
0 VERTICAL COMPONENTS OF EDGE
X INTERSECTIONS OF * & 0
+ SLOPE AND AVERAGE PROCESSING EDGES
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

Figure 10. Average Processing Applied to a One Meter Diameter Crater at Ten Meters

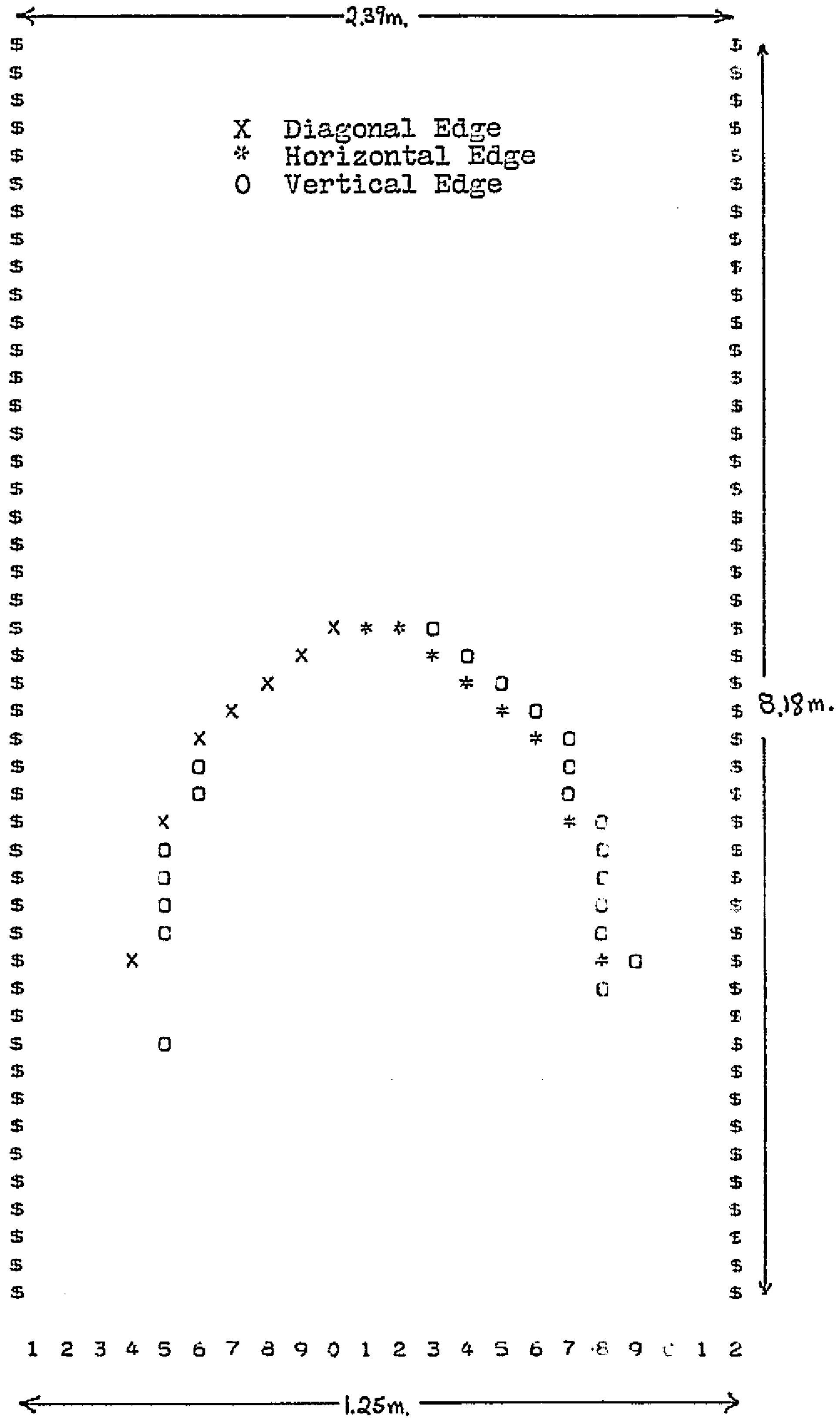


Figure 11. One Meter Diameter Boulder at Ten Meters Corrupted by Uniform 0.1 Meter Maximum Amplitude Noise

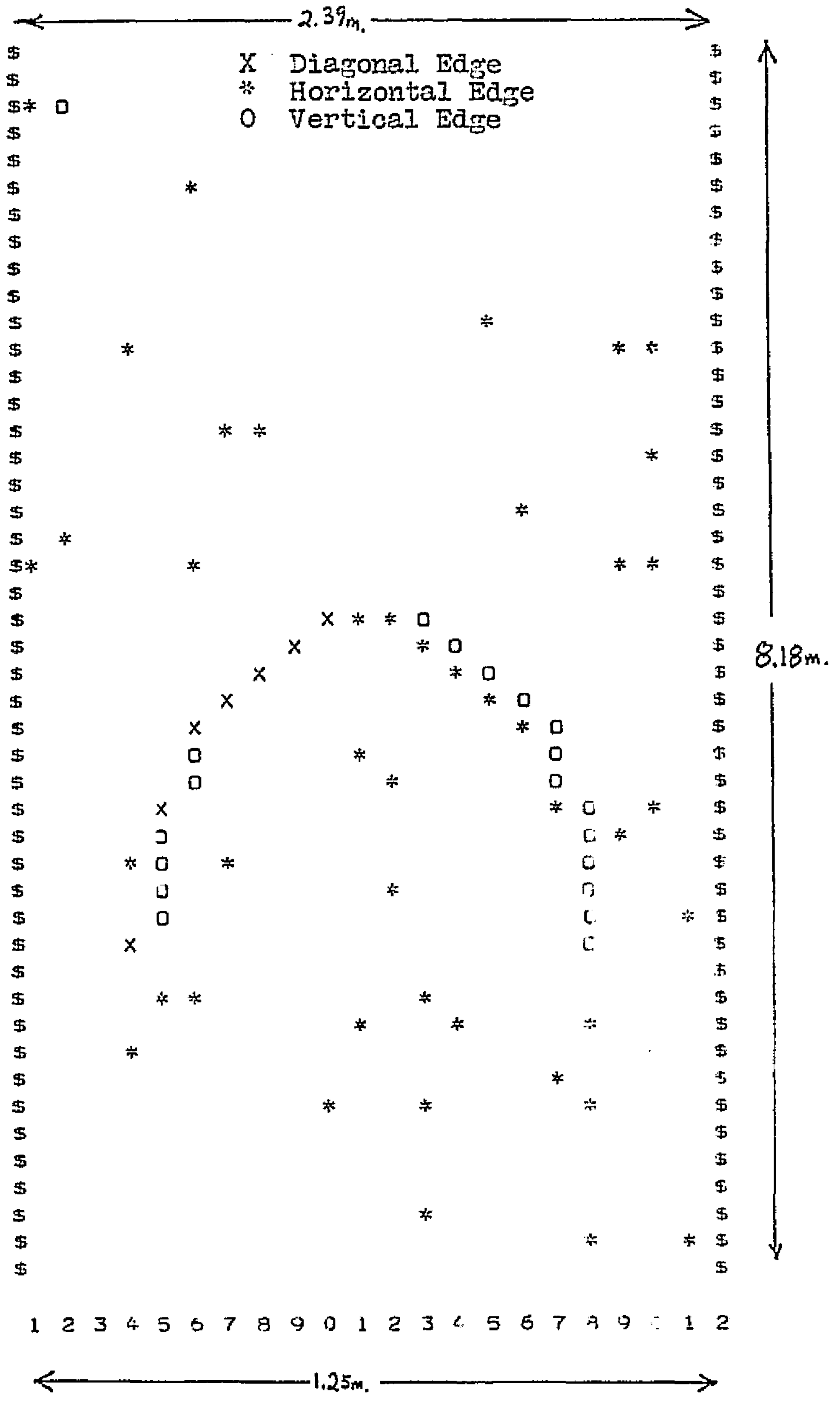


Figure 12. One Meter Diameter Boulder at Ten Meters with Uniform 0.2 Meter Maximum Amplitude Noise
(a) Kalman Filter

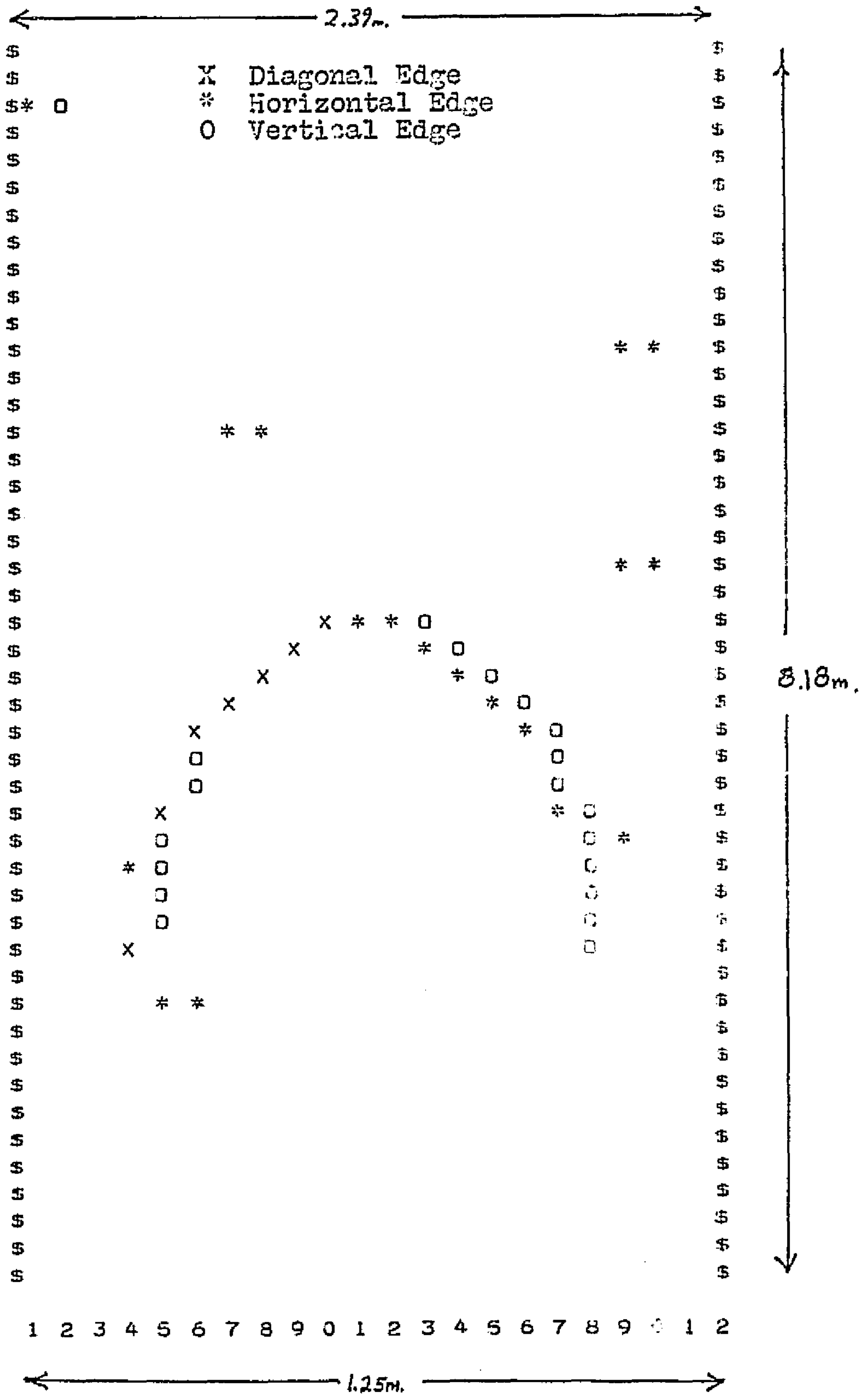


Figure 12. One Meter Diameter Boulder at Ten Meters with Uniform 0.2 Meter Maximum Amplitude Noise
(b) Kalman Filter and Noise Filter

2.39m.

```
$                           + +             $
$+       P  Positive Obstacle          +   $
$*       N  Negative Obstacle          * * $
$        X  Diagonal Edge         * +     $
$+       *  Horizontal Edge                $
$*       O  Vertical Edge         +       $
$          Average Processing Edge * +   + $
$      *           +     + U T T T   * * * $
$  + + + + +       * + +         +   + +   $
$    * + +   +         *         * +   + + $
$+ +   * * * * + + + + N + +   + +   * * * $
$            +   *       *     *           $
$  +       +         * + + +       + + +   $
$+ *     + +     + +   + +   +         *   $
$N X     X     +   *   X     *             $
$* O   + *     *       + + + _ +       + + $
$              +               +       + N $
$    + + + + +       + + +     +       * * $
$      + * * *       *   + +     + + +     $
$    * *       + + + + *           + +     $
$+ +     + + + *   +     + +   + + * N +   $
$* * +             X * + O         +   *   $
$      + + + *   X + P + * X         * + + $
$      * +     X + + + * P P X + +       + $
$*       * + X P P P + + + * * O   + + + * $
$+ + + +   X P + + P P P P P + * C * N N + $
$    *     X * P P + + + * + + P X     + + $
$N N N N + X + + * P * * P P P + O + *   + $
$        X + P P O P P P P P P P C + + * * $
$  + +   O P + + + + + + + + + * + X       $
$N N N N O P + * ' X * + + * + + X + + + + $
$  +   + X P P P + * + + + + P + + C * * * $
$  O +   X + * + P P P P P P P P P C + + + $
$+ O *   X + P + + P P P + + + + + O + N N $
$      * + + + P P X + + P *     +     + * $
$      + + P P P P + + + + P P +           $
$  + + + + + * + * P * * * + + + +     * + $
$+             + + + + +     + *     + +   $
$+   + + + + + + +       + + *   + + N N + $
$  + N *   + N N * X + +   * + + N + N N * $
$  + *       * *     * +           * *     $
$  +           +             + X + +       $
$  *           O + + + * + +     +         $
$+ + +       + +   + + +       + * +       $
$N N N + +           * N + +               $
$                          * +             $
$+ + +   +       + +   + +   +   *   * +   $
$* *   + + + * *   *         * + + * +     $
$                                          $

1  2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
```

8.18 m.

1.25m.

Figure 13. One Meter Diameter Boulder at Ten Meters with Uniform 0.3 Meter Maximum Amplitude Noise
(a) Kalman Filter

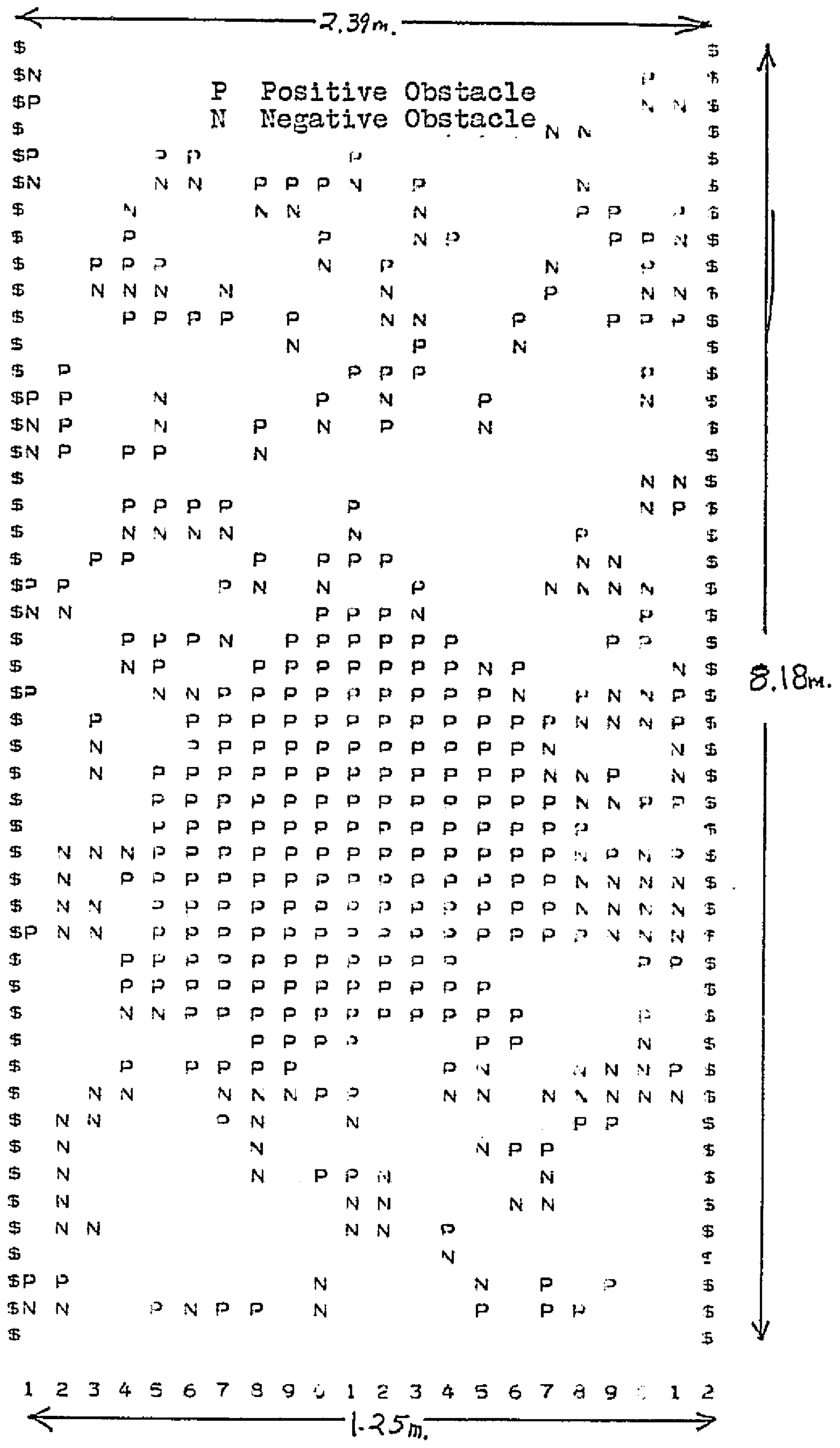
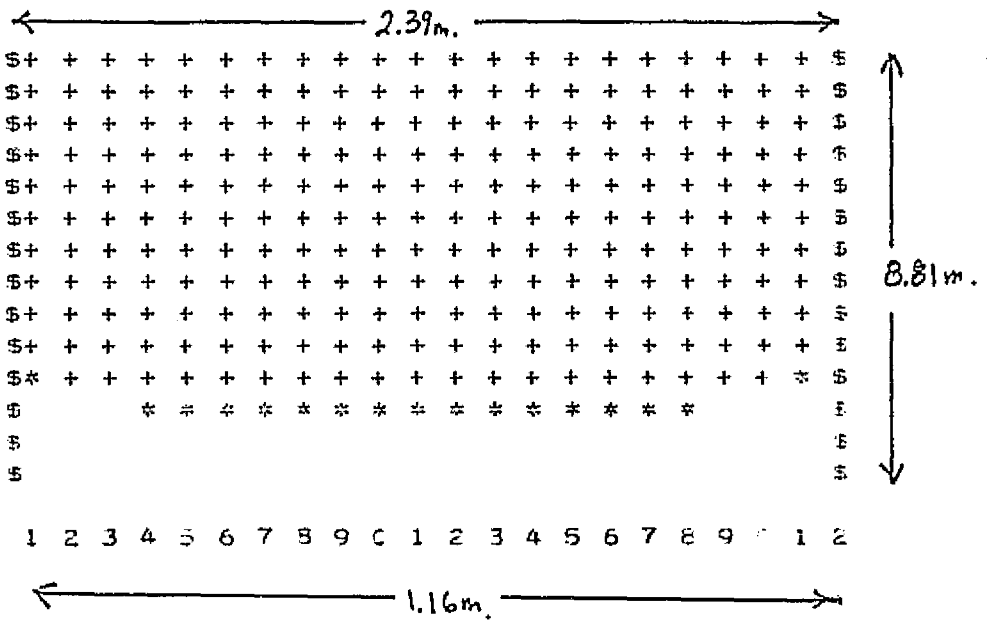


Figure 13. One Meter Diameter Boulder at Ten Meters with Uniform 0.3 Meter Maximum Amplitude Noise
(b) Kalman Filter and Noise Filter

two stage prediction process; the first predicts what the true next state should be, and the second predicts what the next state could be if edges occurred at different stages. Noise corrupts both stages of prediction. In contrast, vertical edges result from a single stage prediction process since the true value of the next state is already known, and only the edge predictions must be taken into account.

Other test cases showed that the edge detection scheme developed here performed quite well. It identified partial obstacles (Figure 14), double obstacles (Figure 15), obstacles on sine waves, and obstacles on sine waves with noise (of 0.2 meter maximum amplitude). It correctly flagged a small amplitude sine wave as no obstacle and a large amplitude sine wave as an obstacle. Higher masts or close ranges are needed in order to identify a large sine wave as a continuous function.

Problems inherent in the Kalman Filter become evident only in the Terrain Modeller, the next stage of computation. Because the Kalman Filter is performed left to right and bottom to top, the delineation of the obstacle is incorrect by one stage on the right side and the top. For example, the Kalman Filter will not detect the actual right edge of the boulder as an edge, because it is the next stage which actually represents the jump in range. Fortunately this type of error is only a slight inaccuracy rather than a serious miscalculation. Examples of this error will be seen in Terrain Model outputs.

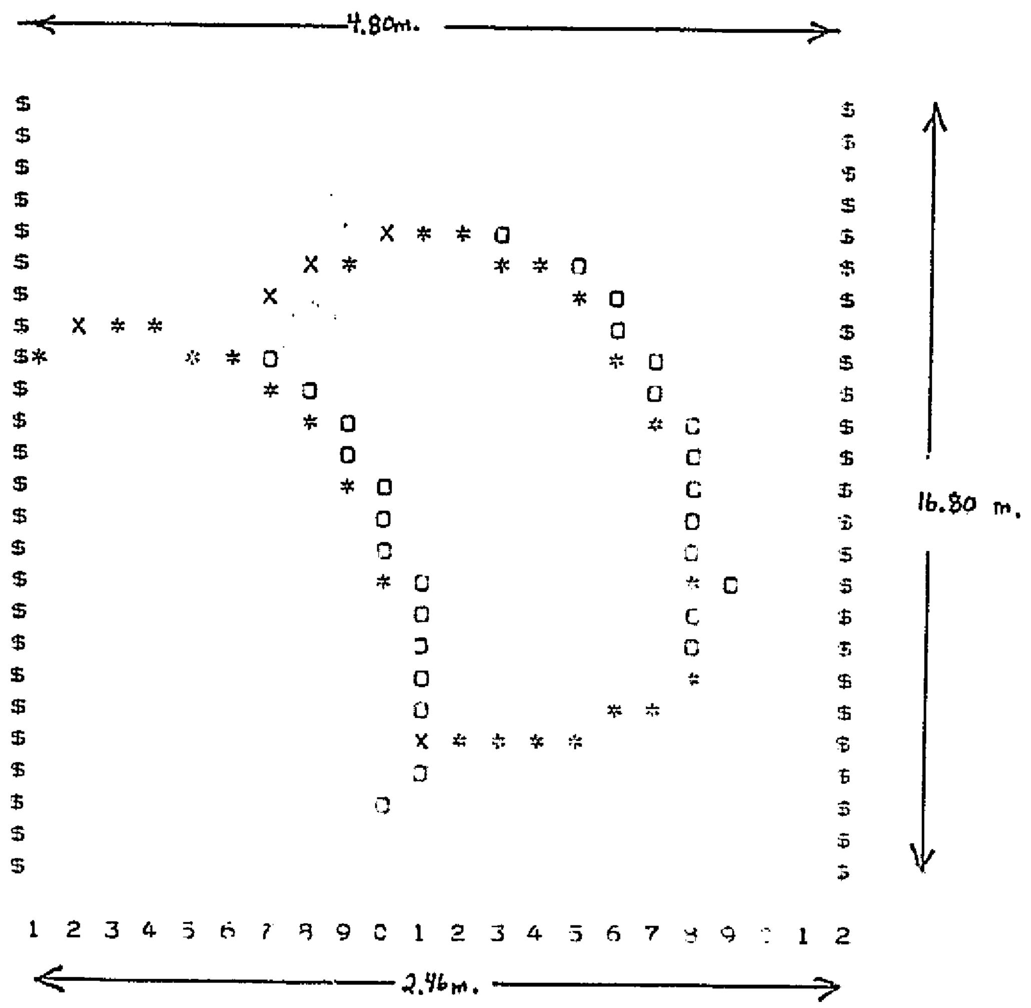


```
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$+ + + + + + + + + + + + + + + + + + + + + $
$* + + + + + + + + + + + + + + + + + + + * $
$      * * * * * * * * * * * * * * *       $
$                                          $
$                                          $

 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
```

* Horizontal Edge
+ Average Processing Edge

Figure 14. Partial Boulder at Ten Meters



*	HORIZONTAL COMPONENTS OF EDGE
O	VERTICAL COMPONENTS OF EDGE
X	INTERSECTIONS OF * & O

Figure 15. Boulder Partially Obscuring Another Boulder
(a) Kalman Filter

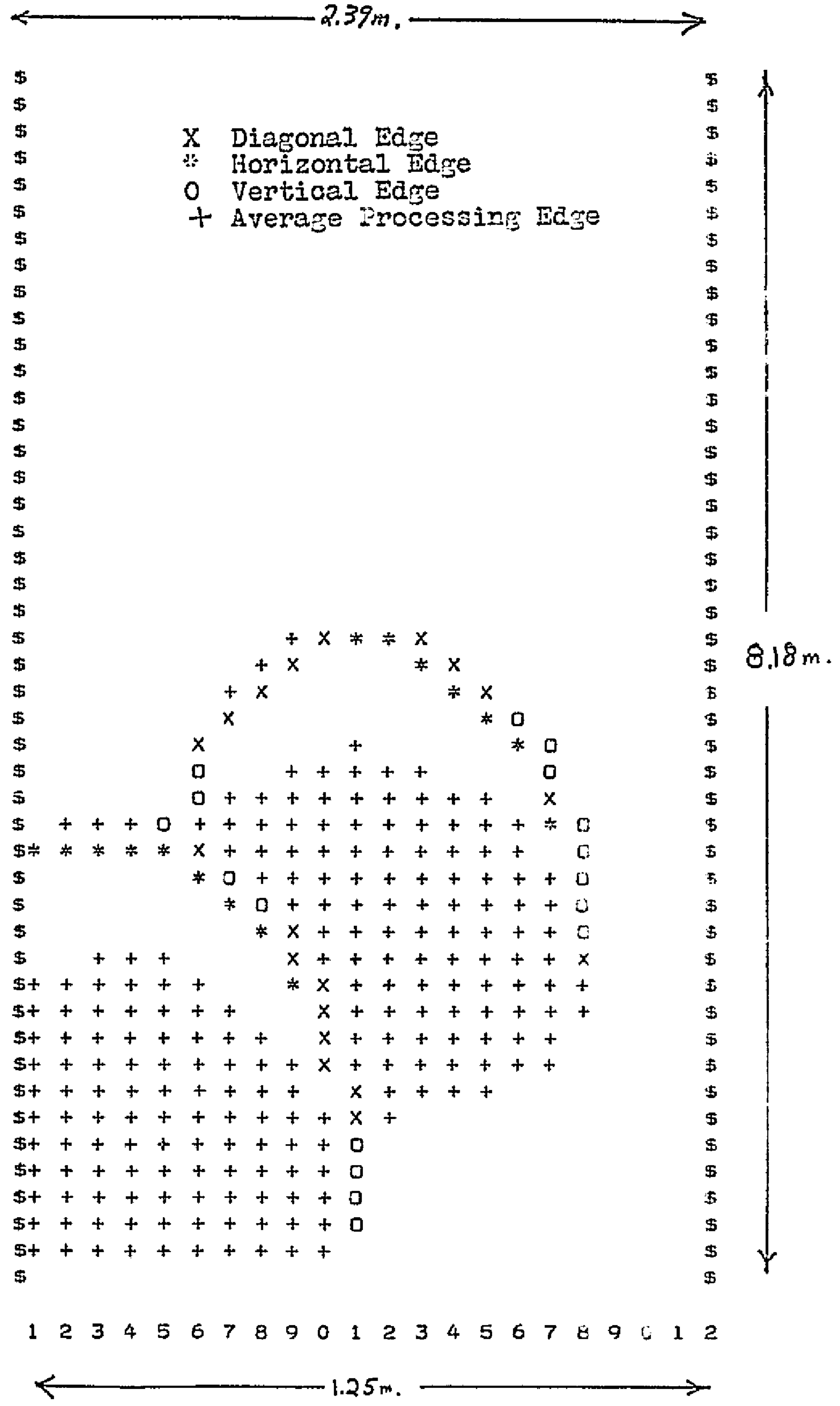


Figure 15. Boulder Partially Obscuring Another Boulder
(b) Kalman Filter and Average Processing

4. THE TERRAIN MODELLER

The Terrain Modeller must organize the mass of data obtained from the Sensor and Edge Detection Blocks into a form that may be readily interpreted by the Path Selection Algorithm. The approach taken here requires the Terrain Modeller to reconstruct the obstacles from their edges and to note the location of each obstacle on a rectangular grid map of the local planet surface.

4.1 Discussion of Modelling Algorithm

The Terrain Modeller must be able to reconstruct the shape and extent of an obstacle, given only its edges. Complicated problems in obstacle identification are commonplace, as for example, partial obstacles (where matching edges do not occur in the sensor field of view) or double obstacles (where distinct edges do not occur in the sensor field of view). The Terrain Modeller assumes that the Kalman Filter is absolutely correct, i.e., that the occurrence of an edge implies the presence of an obstacle at that point, although not necessarily on either side of it.

Another function of the Terrain Modeller is to maintain information about each obstacle in its memory, because of the limited sensor field of view and the high cost in time and computational effort in identifying each obstacle. The forms of memory chosen were a rectangular grid map and a terrain height memory. The rectangular grid map of the local planet terrain contains symbols indicating whether each square of the map is clear, unknown, or full of obstacles. The terrain height memory is an additional matrix whose elements are in one-to-one correspondence with the squares of the grid map. Each element of the terrain height memory contains the current average of all sampled terrain heights up to the present time which fell within the corresponding square on the grid map.

4.2 Implementation

The Terrain Modeller MODELL receives the edge matrix from the Edge Detection Block. It must decide for each edge whether an obstacle lies on either side of the edge or on both sides. Therefore, some sort of range prediction function must be obtained, in order to identify obstacles by comparison of the actual range with the predicted range. The range prediction function used by MODELL is the simple assumption that the terrain slope is constant, and it therefore is identical to the vehicle attitude.

The Terrain Modeller MODELL divides each row into segments. It possess each segment of each row separately. The segments consist of (1) the first element, and all succeeding elements up to but not including the first edge, (2) the edge itself, (3) all elements up to but not including the second edge, (4) the edge itself, etc. The process continues until the last column of that row has been processed.

Segment processing consists of (1) averaging the ranges corresponding to all elements comprising the segment, (2) subtracting the range estimate obtained from the range prediction function, and (3) comparing the remainder to a threshold value. The threshold value is a positive number (presently 0.25 meters). A difference in ranges larger in absolute value than the threshold signals an obstacle, where a positive (negative) difference implies a negative (positive) obstacle, and a difference smaller in absolute value than the threshold indicates no obstacle. The edge itself is processed similarly except that a threshold value is not employed. (By assumption, a Kalman Filter edge automatically implies the presence of an obstacle there.) Thus, the edge matrix is replaced by a matrix containing only "P" (indicating a positive obstacle, such as a boulder), "N" (indicating a negative obstacle, such as a crater), or blank (no obstacle).

During tests, it was noted that a segment would sometimes be artificially continued because one edge was identified by the Kalman Filter and the other edge was not. To correct this problem, a noise filter is applied. The noise filter will erase any "P" or "N" indicator which is not a Kalman edge itself or is not directly adjacent to one unless it has three or more adjacent edges.

The obstacle matrix is next transferred to a rectangular grid map of the local terrain. The Terrain Modeller calculates an appropriately placed coordinate system for the grid map. The origin coincides with a particular element. The positive x-axis of the planet is then assumed to be that part of the row containing the origin, which lies to the right of the origin. The positive y-axis is that part of the column containing the origin which lies above the origin. Because vehicle attitude, azimuth and elevation angles, and the range of each point in the edge matrix are known, the planet rectangular coordinate of the intersection of the beam with the surface may be calculated. The obstacle type indicator (positive, negative, or no obstacle) is transferred to the correct grid location and the actual height is recorded in a terrain height matrix, which maintains an average of all terrain heights measured up until that time for each square of the terrain map.

Each element of the terrain grid map represents a square of the planet surface with dimensions specified by the user. The program MODELL sets up vehicle and target locations, and changes the obstacle type indicator flag in each square to a "U", to indicate that it has never been scanned, and hence is totally unknown. MODELL assumes that some type of onboard short range sensor is operable and that initially the terrain is traversable from the vehicle to the point where the mid-range sensor becomes available. Therefore those squeares are indicated as clear (blank).

The terrain grid map contains an obstacle type indicator for each square of the map. It may be set to "P", "N", blank, or "U" (positive, negative, clear, or unknown). The indicators "P" or "N" will never be replaced with a clear or unknown signal, but the most recently calculated value of "P" or "N" will take precedence, if there happens to be a conflict concerning the type of obstacle.

The terrain height memory is updated by a rather complicated calculation. It must maintain an average of all sampled terrain heights within the corresponding square of the grid map. Memory space may be saved by incorporating two pieces of information in one height matrix, instead of introducing another matrix which remembers the number of samples previously averaged. Thus, a true average may be calculated instead of weighting the most recent sample most heavily. Multiple data may be stored in one matrix by the use of place value. Thus, the thousands digit and higher order digits record the number of samples represented by the average. The ones and tens digits represent the average height. However, negative heights cannot be represented without affecting the thousands digit unless the hundreds digit is non-zero. Thus, the hundreds digit is set to a zero level of 5 initially. Some examples will clarify this procedure. The entry 5502.6 represents an average height of (502.6-500.0)= + 2.6 meters obtained after five samples. The entry 7498.3 represents seven samples with an average height of (498.3-500.0)= - 1.7 meters.

The actual programmed calculation is as follows. The terrain height of each square is initialized to a value of 500.0 (corresponding to the zero level for each grid square). Every time that the square is scanned, 1000.0 is added to the value of the height. Thus the integral number of thousands indicates the number of times that the square has been sampled. Also the 1000-modulus (or remainder after all integral thousands have been subtracted) is averaged with the terrain height just calculated, and re-adjusted to a zero level of 500.0. The 1000-modulus is weighted by the integral number of thousands in the averaging process. Thus a true average of all of the sampled points is maintained. For example, an average of 4500.5 with a current sample height of 0.75 meters would yield an actual average height of

(4 samples x 0.5 meters + 1 sample x 0.75 meters)/5 samples

= 0.55 meters

which would be recoded as 5500.55.

4.3 Results

In general, the Terrain Modeller has been very successful in correctly identifying obstacles, including partial and multiple obstacles, and obstacles whose measurements have been corrupted by noise. Some examples are shown in Figures 16 through 18.

However, sine waves and other curved terrains cause a very poor performance. Obstacles placed on sine waves are correctly identified, but the positive parts of the sine wave surface are identified as positive obstacles, and the negative parts as negative obstacles, as in Figure 19. This is a direct result of the algorithm used to obtain a range estimate which is then compared with the actual measured range.

The range estimation algorithm assumes that the terrain has a constant slope, which can therefore be measured by the vehicle gyro. Hence, the vehicle itself sees only a flat plane. This is a rather poor approximation since the slope estimate is based on an extremely small portion of the total terrain, and because it assumes a planar type of terrain, excluding such terrains as sine waves or other gently rolling surfaces.

A method which approximated the planet surface between the vehicle and the sample point as a single variable polynomial curve was also tried. It failed because it did not incorporate sufficient general information and hence had to be recalculated many times. Thus it lacked continuity, and also required a lot of computational effort.

The rectangular grid map is an excellent way to store terrain data. The user may adjust the fineness or coarseness of the grid size to accommodate the desired level of detail. Caution must be used to adjust the

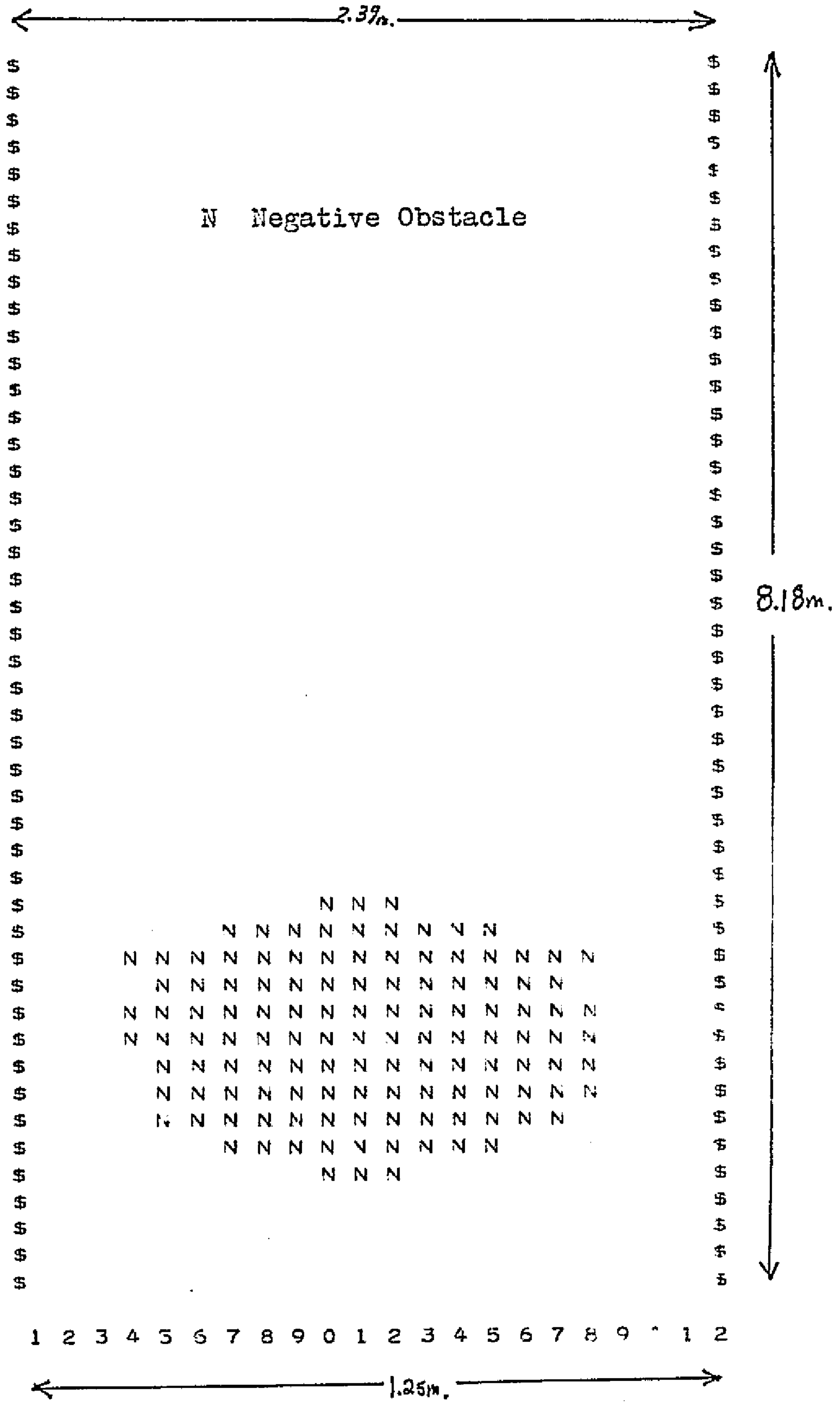


Figure 16. Terrain Modeller Processing of the One Meter Diameter Crater Shown in Figure 10. (a) Obstacle Identification

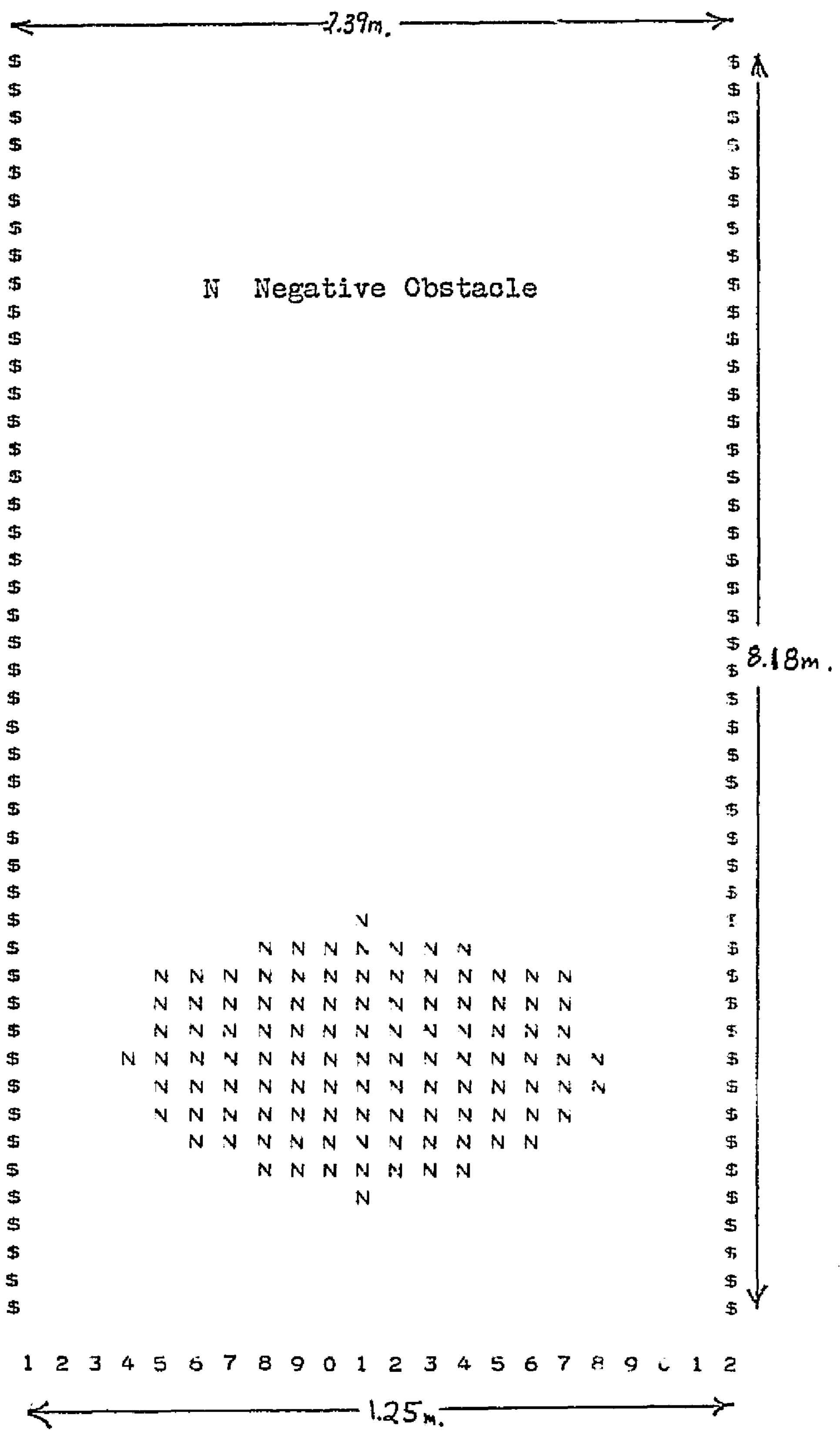
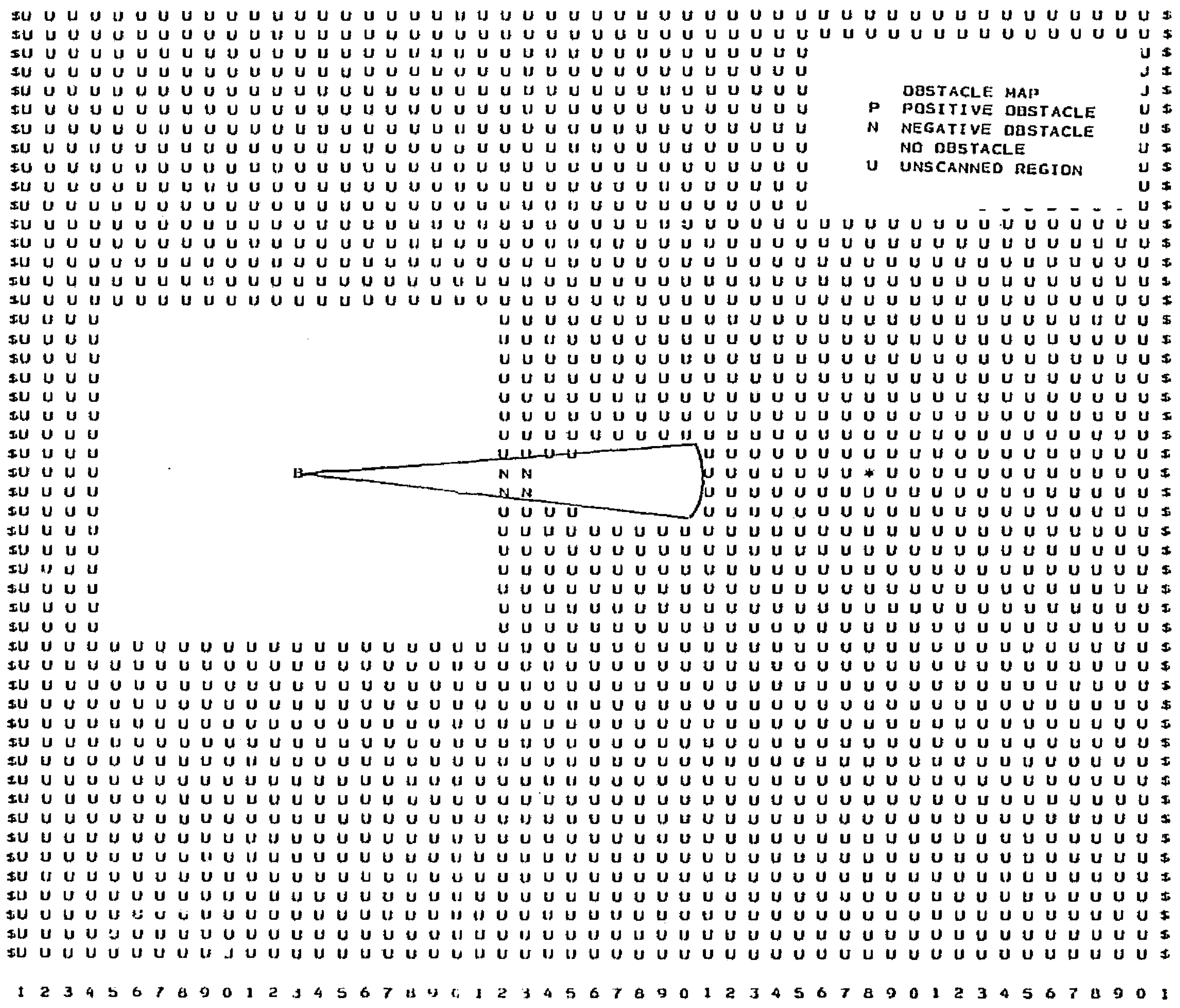


Figure 16. Terrain Modeller Processing of the One Meter Diameter Crater Shown in Figure 10.
(b) Noise Filter



B Vehicle

* Target

Figure 16. Terrain Modeller Processing of the Crater Shown in Figure 10
(c) Terrain Grid Map

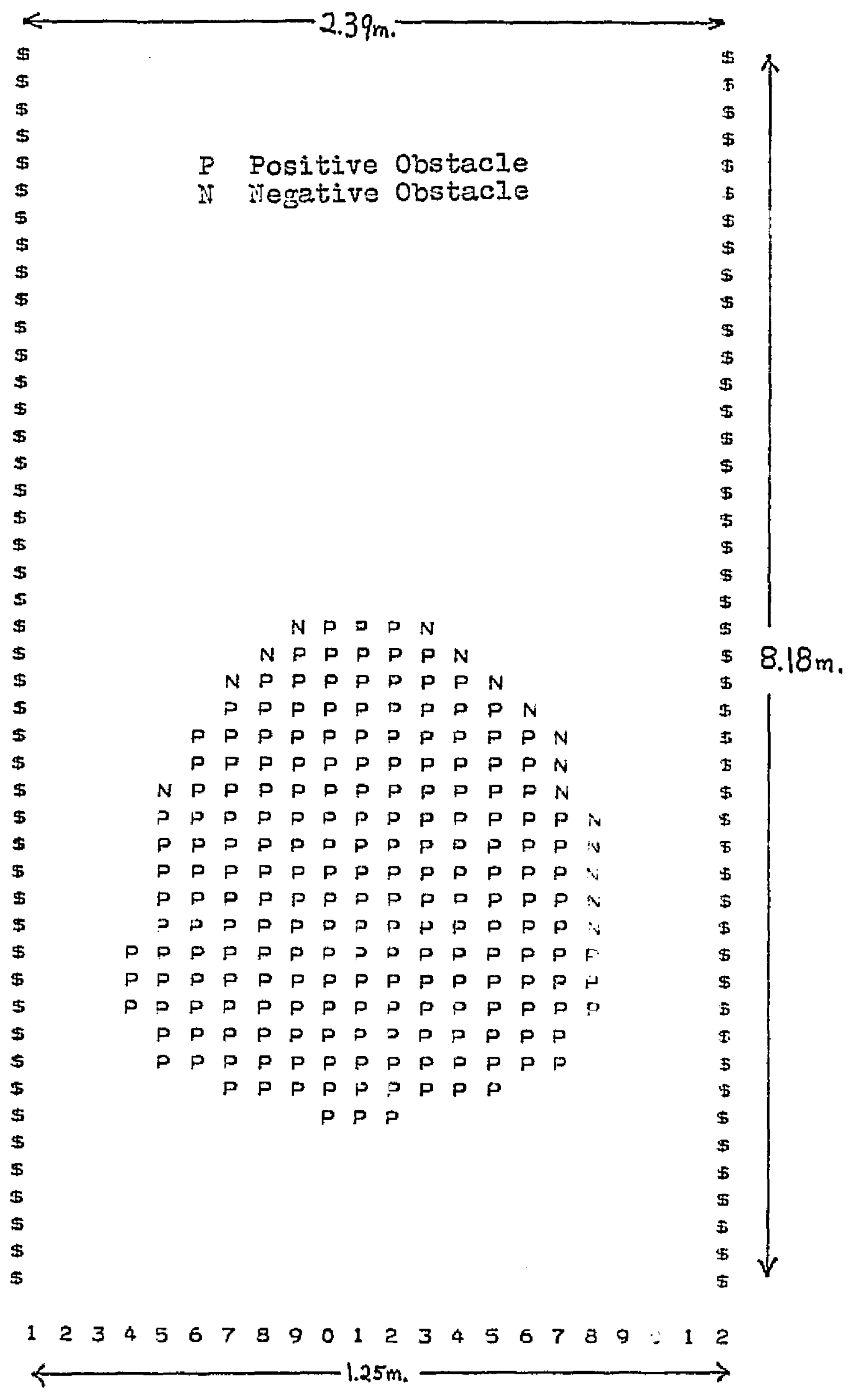


Figure 17. Terrain Modeller Processing of a One Meter Diameter Bculder at Ten Meters
(a) Obstacle Identification

2.39m.

```
$                                           $
$                                           $
$                                           $
$           P   Positive Obstacle           $
$           N   Negative Obstacle           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                   P P P                   $
$                 P P P P P                 $
$               P P P P P P P               $
$             P P P P P P P P P             $
$             P P P P P P P P P P           $
$           P P P P P P P P P P P N         $
$           P P P P P P P P P P P N         $
$         P P P P P P P P P P P P P         $
$         P P P P P P P P P P P P P N       $
$         P P P P P P P P P P P P P N       $
$         P P P P P P P P P P P P P N       $
$         P P P P P P P P P P P P P N       $
$         P P P P P P P P P P P P P P       $
$       P P P P P P P P P P P P P P P       $
$         P P P P P P P P P P P P P         $
$         P P P P P P P P P P P P P         $
$         P P P P P P P P P P P P           $
$               P P P P P P P               $
$                     P                     $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $
$                                           $

  1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
```

8.18m.

1.25m.

Figure 17. Terrain Modeller Processing of a One Meter Diameter Boulder at Ten Meters (b) Noise Filter

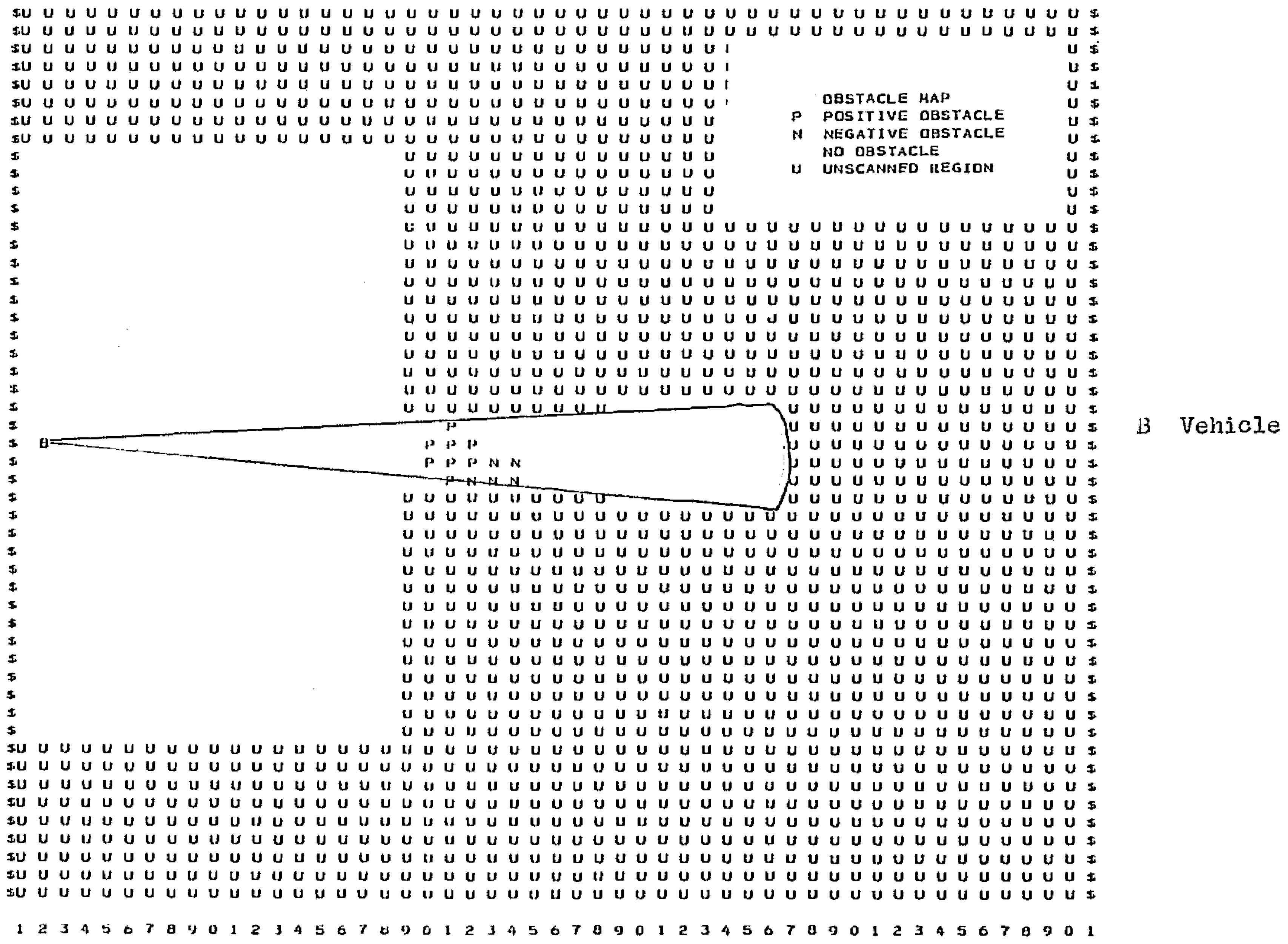


Figure 17. Terrain Modeller Processing of a One Meter Diameter Boulder at Ten Meters
(c) Terrain Grid Map

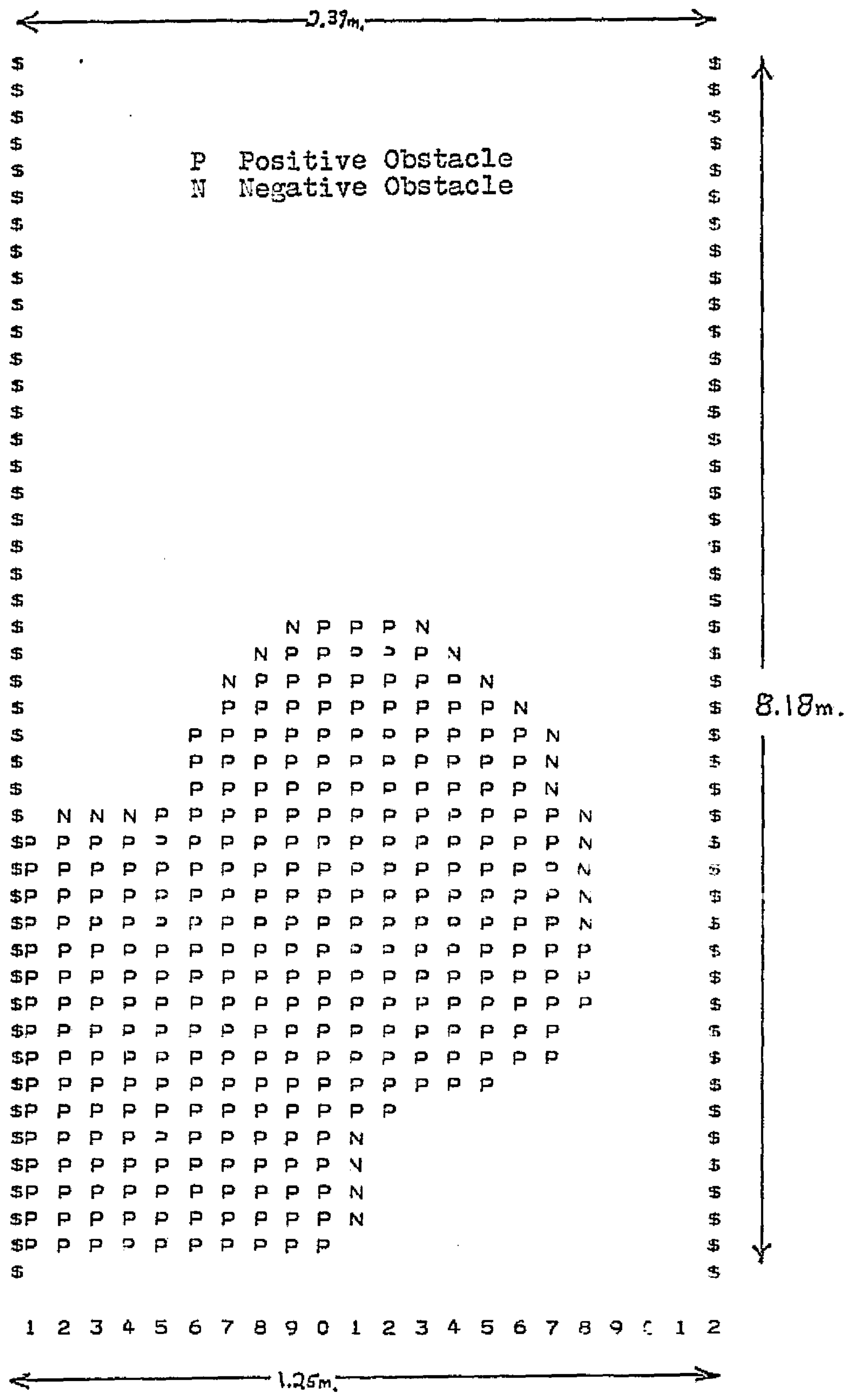


Figure 18. Terrain Modeller Processing of Double Obstacle of Figure 15
(a) Obstacle Identification

2.39m.

P Positive Obstacle
N Negative Obstacle

```
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                                          $
$                  P P P                   $
$                P P P P P                 $ 8.18m.
$              P P P P P P P               $
$            P P P P P P P P P             $
$            P P P P P P P P P P           $
$          P P P P P P P P P P P N         $
$          P P P P P P P P P P P N         $
$    N N P P P P P P P P P P P P P         $
$  P P P P P P P P P P P P P P P P N       $
$P P P P P P P P P P P P P P P P P N       $
$P P P P P P P P P P P P P P P P P N       $
$P P P P P P P P P P P P P P P P P N       $
$P P P P P P P P P P P P P P P P P P       $
$P P P P P P P P P P P P P P P P P P       $
$P P P P P P P P P P P P P P P P P         $
$P P P P P P P P P P P P P P P P P         $
$P P P P P P P P P P P P P P P P           $
$P P P P P P P P P P P P P P               $
$P P P P P P P P P P P                     $
$P P P P P P P P P P N                     $
$P P P P P P P P P P N                     $
$P P P P P P P P P P N                     $
$P P P P P P P P P P                       $
$  P P P P P P P P                         $
$                                          $

 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
```

1.25m.

Figure 18. Terrain Modeller Processing of Double Obstacle of Figure 15
(b) Noise Filter

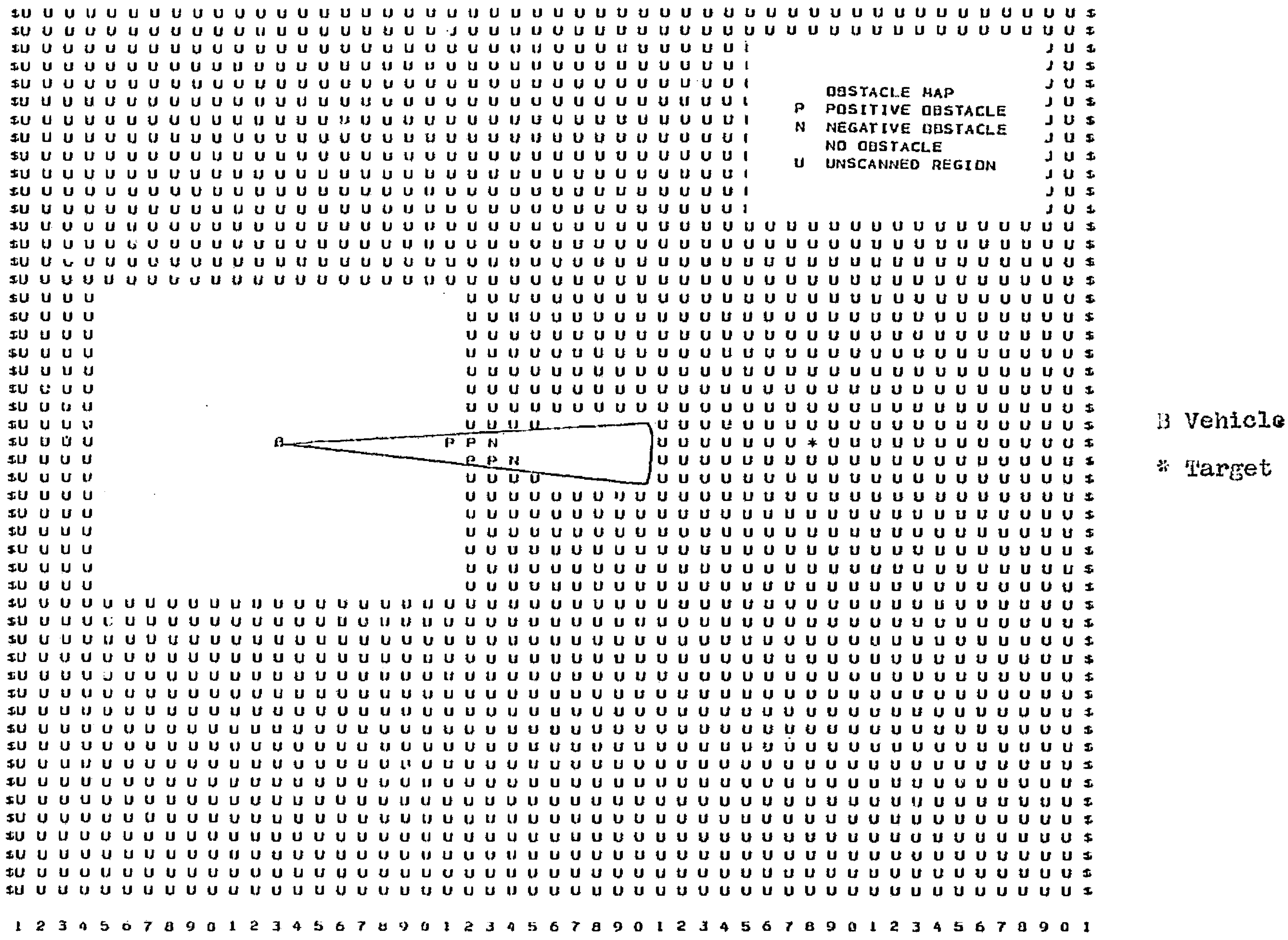


Figure 18. Terrain Modeller Processing of Double Obstacle of Figure 15(b)
(c) Terrain Grid Map

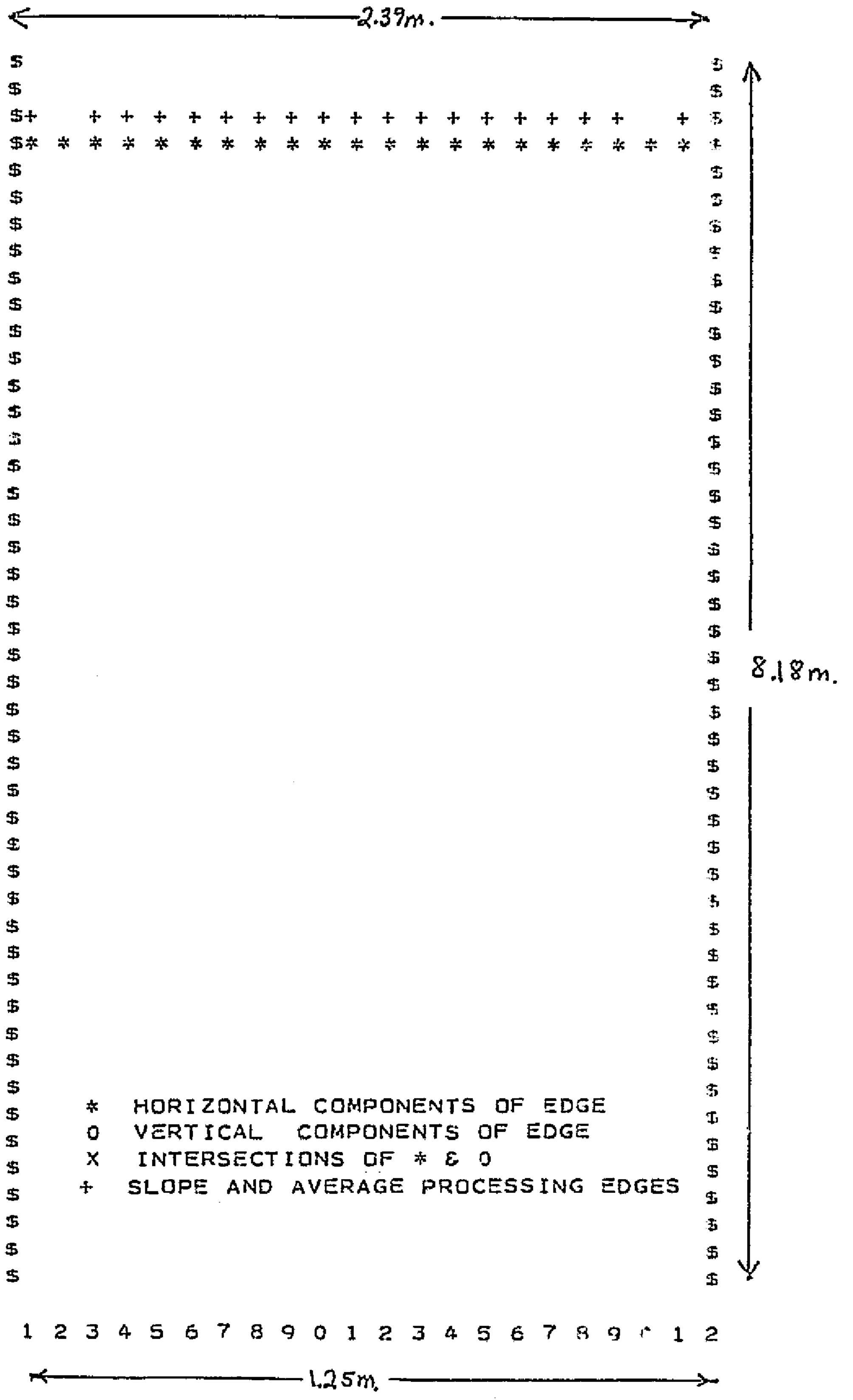


Figure 19. Terrain Modeller Processing of 1.0 Meter Amplitude Sine Wave
(a) Edge Detection Output

```
<————————————————————— 2.39m. —————————————————————>

$N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  $
$N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  $
$N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P                                      P  P  P  P  $
$P  P  P  P  P     P  Positive Obstacle             P  P  P  P  $
$P  P  P  P  P     N  Negative Obstacle             P  P  P  P  $
$P  P  P  P  P  .  .  .  .  .  .  .                 P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $   8.18m.
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $
$P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  P  $

 1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2

  <—————————————————————— 1.25m. ——————————————————————>
```

Figure 19. Terrain Modeller Processing of 1.0 Meter Amplitude Sine Wave
(b) Obstacle Identification

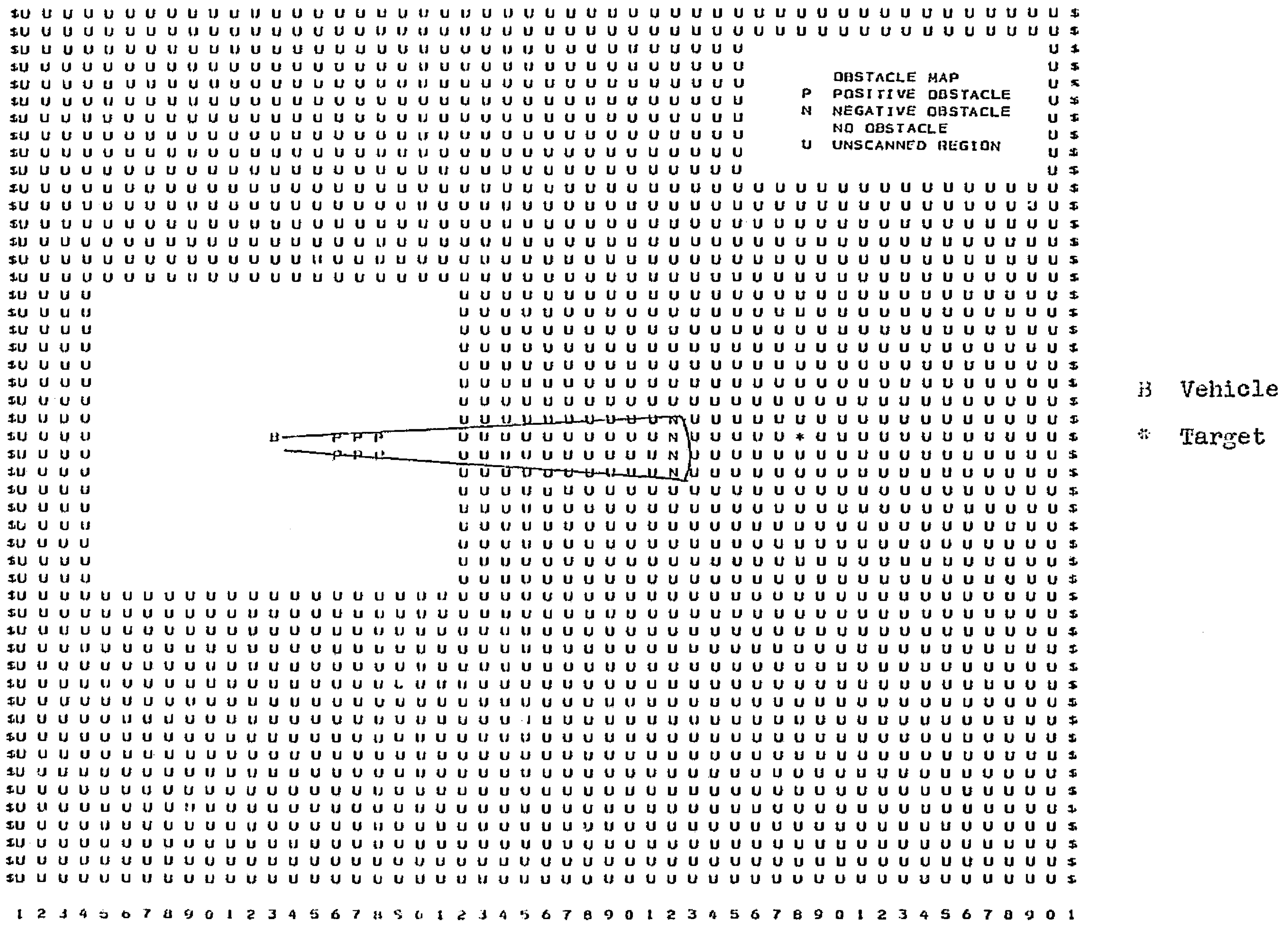


Figure 19. Terrain Modeller Processing of 1.0 Meter Amplitude Sine Wave

grid size along with the elevation angle increment in order to avoid creating gaps of unknown region in the midst of the scanned region.

There are two other minor problems. One problem has previously been discussed in Section 3.3. The right edge or top edge of an obstacle may consist of spurious edges. However, the spurious edge is classified as an obstacle and placed in the grid map's memory. Thus, the second problem arises. With the present program logic, a square containing an obstacle can never be declared clear. It has not yet proved necessary to challenge this assumption. Noise and spurious edges may eventually become a serious problem, though.

5. THE PATH SELECTION ALGORITHM

The Path Selection Algorithm is the program block which closes the loop in the computer simulation. The Path Selection Algorithm uses the rectangular grid map generated by the Terrain Modeller to evaluate acceptable paths to the target. A modified form of an existing algorithm was implemented, and promising results were obtained.

5.1 Discussion of Algorithm

The Path Selection Algorithm must generate a set of steering commands, given the rectangular grid of the terrain. Several simplifying assumptions have been made to facilitate a first pass at the problem solution:

(1) The vehicle dimensions are such that it can be contained in only one square of the obstacle grid map (usually one meter by one meter),

(2) the vehicle may choose only 0°, 90°, 180°, or 270° as heading angles, i.e., it may travel only from its present square to an adjacent square, and

(3) there is a short-range sensor aboard which would prevent the vehicle from hitting an obstacle which had slipped through its rather narrow mid-range sensor screen.

A path selection algorithm created by Lallman[8] dealt with much the same situation, although his assumptions differ from the above, particularly as to the amount of sensor information available. C. Y. Lee[7] developed a method for obtaining a minimal length path given such a grid containing obstacles, and this method has been modified and adapted for use here. Lee's algorithm assumed that all obstacles are known before any part of the path is chosen. He then numbered adjacent squares outward from the target to the present location with ascending positive integers. A minimal path consisted of traveling from one square to any adjacent squares containing a lower number. (Refer to Figure 20 for an illustration of this procedure.)

The path selection problem here differs drastically from Lee's problem in at least one respect. The presence or absence of obstacles is not known in advance. Therefore, two different matrices are used in making a path selection decision. One matrix is the terrain grid map maintained by the Terrain Modeller. It determines which squares are clear and which squares contain obstacles. The second matrix is called the Path Selection Map. Its squares are placed in one-to-one correspondence with those of the terrain grid map. The purpose of the Path Selection Map is to represent the priority that each square on the grid map has when a path selection decision is being made. The numbers provide a guideline for choosing a "near-optimal" path. The priority numbering system is similar to Lee's Algorithm, except that all of the squares are numbered as part of the initialization, assuming that no obstacles are present. The numbering of the squares is not altered during the entire simulation, except in the ways described below. This scheme prevents ambiguous or unnecessary renumbering whenever new information becomes available.

5.2 Implementation

A path is selected with the aid of the Path Selection Map described above. Usually, the vehicle will proceed from its present square to an

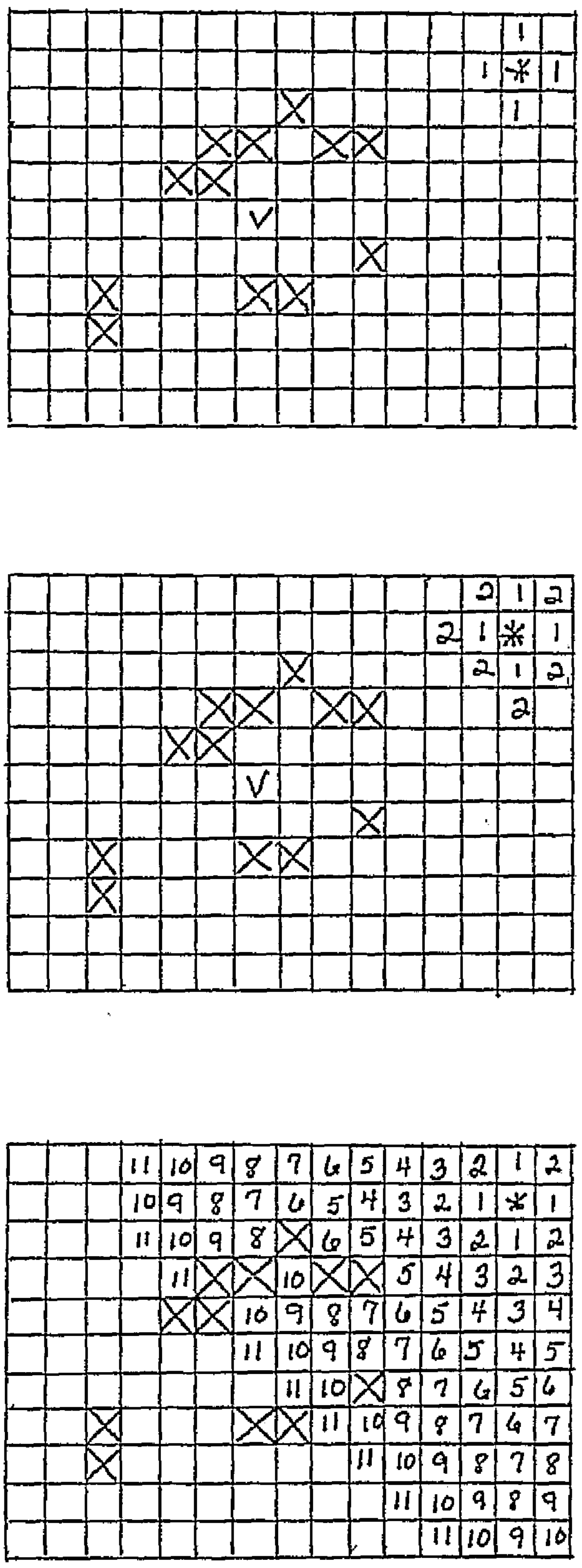


Figure 20. Lee's Algorithm

adjacent square whose priority number is one less than the priority number of the present square. However, there are two cases in which a square's priority number must be altered to prevent normal access to the square. The first case is when the corresponding square of the grid map contains an obstacle. The second case is when the normal priority scheme will lead the vehicle to a dead end.

Squares that correspond to obstacles are renumbered so that they contain numbers larger than any present in the matrix through the normal adjacent squares numbering plan. For example, a 50 by 50 matrix causes them to be renumbered to 99.

The matrix is then processed to search out and block dead end squares. Any square that has no adjacent squares containing a lower number is indicated as a dead end. Its square number is changed to the corresponding negative integer.

A crude short range sensor, which detects boulders and craters, is also simulated.

The path selection decision logic ranks its options in the following order of importance.

(1) Only one step (or square) may be taken between scans.

(2) A path which will place the vehicle in a square containing an obstacle may never be selected.

(3) If the present square of the vehicle is indicated as a dead end (i.e., its path selection number is negative) then proceed in the direction of the nearest clear square, that is not a dead end. If the vehicle is totally blocked, go to the emergency algorithm. If the present square

is clear:

(4) Step to the adjacent square with the next smaller integer.

(5) If the square containing the next smallest integer is a dead end, step to an adjacent square with the next largest number.

(6) If a lower number cannot be found, check the special case where vehicle and obstacle are on a straight line with the target. Proceed, if possible, to the adjacent square with the next largest integer that is not a direct line with the target.

(7) If no decision has yet been reached, back up and block the square.

(8) If a backup is impossible, call the emergency algorithm.

The emergency algorithm is also called from the vehicle dynamics block and thus must contain some redundancy with the above algorithm. It will first attempt to back out of the problem situation, proceeding to the next larger number. If no way out is found or the number of calls to the emergency algorithm exceeds a user-specified maximum, the simulation will be terminated. The simulation is also terminated if the vehicle leaves the mapped area.

5.3 Results

The test cases shown in Figures 21 through 24 illustrate the performance of the Path Selection Algorithm. These were not run as part of the entire simulation package because of time and money considerations. A test program was designed, however, to fully demonstrate the capabilities of the Path Selection Algorithm itself, without introducing any system interface problems.

```
      B V V
          V
          V
          V
          V
          V
          V
          V
    P     V
  P P P   V
P P P P P V
  P P P   V
    P     V
          V
          V
          V
          V
          V
          V
          V
      * V V
```

B Vehicle Starting Location
P Obstacle
V Vehicle Location
* Target

Figure 21. Navigation Around A Boulder
(a) Path

```
                                       PATH SELECTION MAP, COLUMNS 1-25
   44   43   42   41   40   39   38  -37  -36  -35  -34  -35   36   37   38   39   40   41   42   43   44   45   46   47   48
   43   42   41   40   39   38   37  -36  -35  -34  -33  -34   35   36   37   38   39   40   41   42   43   44   45   46   47
   42   41   40   39   38   37   36  -35  -34  -33  -32  -33   34   35   36   37   38   39   40   41   42   43   44   45   46
   41   40   39   38   37   36   35  -34  -33  -32  -31  -32   33   34   35   36   37   38   39   40   41   42   43   44   45
   40   39   38   37   36   35   34  -33  -32  -31  -30  -31   32   33   34   35   36   37   38   39   40   41   42   43   44
   39   38   37   36   35   34   33  -32  -31  -30  -29  -30   31   32   33   34   35   36   37   38   39   40   41   42   43
   38   37   36   35   34   33   32  -31  -30  -29  -28  -29   30   31   32   33   34   35   36   37   38   39   40   41   42
   37   36   35   34   33   32   31  -30  -29  -28  -27  -28   29   30   31   32   33   34   35   36   37   38   39   40   41
   36   35   34   33   32   31   30  -29  -28  -27  -26  -27   28   29   30   31   32   33   34   35   36   37   38   39   40
   35   34   33   32   31   30   29  -28  -27  -26  -25  -26   27   28   29   30   31   32   33   34   35   36   37   38   39
   34   33   32   31   30   29   28  -27  -26  -25  -24  -25   26   27   28   29   30   31   32   33   34   35   36   37   38
   33   32   31   30   29   28   27  -26  -25  -24  -23  -24   25   26   27   28   29   30   31   32   33   34   35   36   37
   32   31   30   29   28   27   26  -25  -24  -23  ORIGINAL VEHICLE LOCATION   29   30   31   32   33   34   35   36
   31   30   29   28   27   26   25  -24  -23  -22                                  28   29   30   31   32   33   34   35
   30   29   28   27   26   25   24  -23  -22  -21  -20  -21   22   23   24   25   26   27   28   29   30   31   32   33   34
   29   28   27   26   25   24   23  -22  -21  -20  -19  -20   21   22   23   24   25   26   27   28   29   30   31   32   33
   28   27   26   25   24   23   22  -21  -20  -19  -18  -19   20   21   22   23   24   25   26   27   28   29   30   31   32
   27   26   25   24   23   22   21  -20  -19  -18  -17  -18   19   20   21   22   23   24   25   26   27   28   29   30   31
   26   25   24   23   22   21   20  -19  -18  -17  -16  -17   18   19   20   21   22   23   24   25   26   27   28   29   30
   25   24   23   22   21   20   19  -18  -17  -16  -15  -16   17   18   19   20   21   22   23   24   25   26   27   28   29
   24   23   22   21             18  -17  -16  -15  -14  -15   16   17   18   19   20   21   22   23   24   25   26   27   28
   23   22   21   20   BOULDER   17  -16  -15  -14  -13  -14   15   16   17   18   19   20   21   22   23   24   25   26   27
   22   21   20   19             16  -15  -14   99  -12  -13   14   15   16   17   18   19   20   21   22   23   24   25   26
   21   20   19   18   17   16   15  -14   99   99   99  -12   13   14   15   16   17   18   19   20   21   22   23   24   25
   20   19   18   17   16   15   14   99   99   99   99   99   12   13   14   15   16   17   18   19   20   21   22   23   24
   19   18   17   16   15   14   13   12   99   99   99   10   11   12   13   14   15   16   17   18   19   20   21   22   23
   18   17   16   15   14   13   12   11   10   99    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22
   17   16   15   14   13   12   11   10    9    8    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21
   16   15   14   13   12   11   10    9    8    7    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
   15   14   13   12   11   10    9    8    7    6    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
   14   13   12   11   10    9    8    7    6    5    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
   13   12   11   10    9    8    7    6    5    4    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
   12   11   10    9    8    7    6    5    4    3    2    3   TARGET     6    7    8    9   10   11   12   13   14   15   16
   11   10    9    8    7    6    5    4    3    2    1    2              5    6    7    8    9   10   11   12   13   14   15
   10    9    8    7    6    5    4    3    2    1    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
   11   10    9    8    7    6    5    4    3    2    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
   12   11   10    9    8    7    6    5    4    3    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
   13   12   11   10    9    8    7    6    5    4    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
   14   13   12   11   10    9    8    7    6    5    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
   15   14   13   12   11   10    9    8    7    6    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
   16   15   14   13   12   11   10    9    8    7    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
   17   16   15   14   13   12   11   10    9    8    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21
   18   17   16   15   14   13   12   11   10    9    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22
   19   18   17   16   15   14   13   12   11   10    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23
   20   19   18   17   16   15   14   13   12   11   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24
   21   20   19   18   17   16   15   14   13   12   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25
   22   21   20   19   18   17   16   15   14   13   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26
   23   22   21   20   19   18   17   16   15   14   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27
   24   23   22   21   20   19   18   17   16   15   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28
   25   24   23   22   21   20   19   18   17   16   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29
```

```
                            P
                            P
                            P
                            P
                            P
B                           P V V V V V V V V V V *
V                           P V
V                           P V
V                           P V
V                           P V
V                           P V
V V V V V V V V V V V V V V V V
```

B	Vehicle Starting Location
P	Obstacle
V	Vehicle Location
*	Target

Figure 22. Navigation Around A Wall

```
  P P P P P P P P P
                  P
                  P
                  P
                  P
B                 P V V V V V V V V V V *
V                 P V
V                 P V
V                 P V
V                 P V
V P P P P P P P P P V
V V V V V V V V V V V
```

B	Vehicle Starting Location
P	Obstacle
V	Vehicle Location
*	Target

Figure 23. Navigation Around A Keyhole

```
                                  P
                                  P
                                  P
                  P                         P
                  P                         P
                  P                         P
                  P                         P
                  P                         P
B P               P                         P V V *
V P               P                         P V
V P                                         P V
V P V V V V V V V V V V V V V V V V V V V V V V
V P V
V P V
V V V
```

B Vehicle Starting Location
P Obstacle
V Vehicle Location
* Target

Figure 24. Navigation Around Several Walls

The vehicle successfully chose a path around a boulder, a wall, a set of walls, and it avoided a keyhole that it had not yet entered, while choosing an optimal path. For example, refer to Figure 21. Part (b) shows the priority numbering of the Path Selection Map. According to rule (3) above, the vehicle proceeded to the right to the nearest clear square, and then chose a path to the target by following a descending sequence of numbers, by rule (4).

This powerful algorithm has been able to more fully utilize and interpret the large amount of information available than any other path selection algorithm to date. Another test case, not shown, consisted of a keyhole that the vehicle had already entered. This caused the vehicle to wander aimlessly without making any progress. Thus, a new procedure for defining dead ends may be needed.

A single test case involving the entire simulation package was run closed-loop and the results are shown in Figure 25. The vehicle initially faced towards the reader. It turned to the target and after five scans it had opened a path to the target. Unfortunately, monetary considerations forced a halt at this point. Therefore, the approach described here is definitely a feasible, although expensive solution.

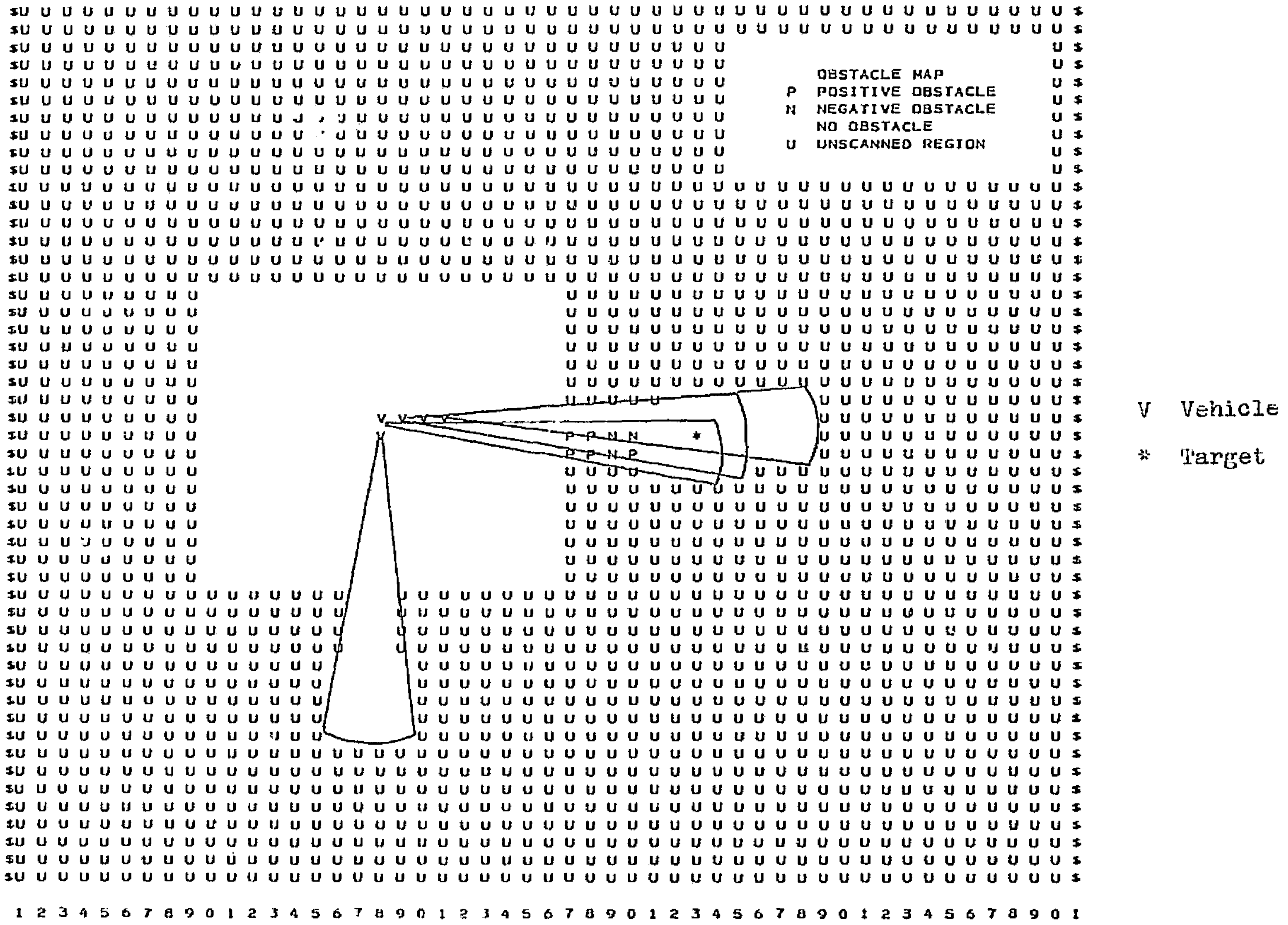


Figure 25. Closed Loop Path Selection

6. RECOMMENDATIONS AND CONCLUSIONS

The results obtained from the computer simulation have been evaluated. The recommendations are divided into two classes, namely those that pertain to the simulation itself, and those concerning the performance of the algorithms.

Each module is evaluated separately, and some general conclusions are then drawn.

6.1 Recommendations

Each of the four program modules previously described may be improved. The following sections list specific recommendations to improve both flexibility and simulation accuracy.

6.1.1 The Sensor

The sensor may be improved in several ways, which may be easily implemented within the existing simulation.

A more realistic sensor simulation could be achieved by the addition of noise to the specified values of azimuth and elevation angles. Noisy sensor angles facilitate simulation of errors in attitude measurement and the finite precision in the mechanical placement of azimuth and elevations.

Noise currently being added to the range measurements should be modified by a factor proportional to the range squared. This would provide a more accurate model of the physical process of loss of the return signal.

The field of view might also be expanded by allowing for additional azimuth and elevation angles. More efficient processing might be obtained because of the reduction in the number of scans required to obtain information about the environment. However, a trade off between computer storage and computational efficiency would then exist.

6.1.2 The Edge Detection Scheme

There are several approaches which could be taken to further improve the edge detection algorithm.

One way to solve the problem of the delayed edge discussed in Sections 3.3 and 4.3 would be to perform the Kalman Filter again, reversing the direction of processing (right to left, top to bottom). The resulting edge matrices would be compared and the correct edge locations would be synthesized from their union.

System performance in the presence of noise should be studied with respect to the parameters B, C, and R, which have remained fixed throughout this study. The B matrix is a matrix of prior probabilities of the occurrence of an edge. C is the cost matrix where C_{ij} is the cost of the i^{th} decision (about edge location) given the j^{th} hypothesis (about the edge location). R is the assumed plant noise variance, presently equal to 0.0025 square meters.

Maximum and minimum slope processing might also be handled in a somewhat different manner. Special indicators could be used to distinguish slope problems from discrete obstacles. Becuase the vehicle attitude is known, the maximum and minimum values of incremental slope might also be used to test for maximum and minimum values of absolute slope.

6.1.3 The Terrain Modeller

It is absolutely imperative that an improved method of obtaining range estimates be developed. This would assure that curved surfaces would be correctly identified, instead of being flagged as obstacles. From experience, the range estimation function should probably be a two-variable curve such as a plane. However, it should preferably be of degree two in order to accommodate some nonlinearity.

Other improvements which might conceivably be justified are: (1) development of an algorithm to remove a spurious obstacle from the grid map, and (2) expansion of the memory storage of the grid map, allowing for more detail and/or coverage of a larger area. The latter could be done simply by using pointers to indicate bounds on current memory space, replacing the unused parts with newer terrain information.

6.1.4 The Path Selection Algorithm

Much more testing of the Path Selection Algorithm capabilities should be done.

Certain basic assumptions should also be modified. For example, the vehicle size could be extended to its actual physical dimensions by a slightly more complicated square blocking scheme. The choice of paths could be modified to include the other four squares at 45° increments from the adjacent squares. This would ease constraints on the turning radius imposed by the vehicle dynamics.

The dead end zone created by an obstacle presently extends infinitely far back from the obstacle on the side of the target. Its extent could be limited, and the vehicle might therefore never enter it or never be caught in a blocked zone.

The short range sensor simulation contained in the Terrain Modeller might also be improved, or an existing short range sensor might be interfaced with this simulation.

A better emergency algorithm might be developed which backs the vehicle exactly along its previous path rather than choosing any available back-up path.

Obstacles resulting from slopes should be treated in a different manner than discrete obstacles. Cases of combinations of slopes and small obstacles which are traversable in themselves, but barriers when combined, should also be examined.

6.2 Conclusions

The path selection system developed requires a large amount of computational time and storage. Hence, it should be used primarily "in the large" rather than at each step along the route. An efficient short range system should be used for detailed path selection, once the global trend of the terrain has been determined.

The interdependence of program modules must be cut to an absolute minimu. An example of this would be the introduction of additional noise processing to remove spurious obstacles from the terrain grid map. Often, however, problems are difficult to isolate and are identified only at a later stage of the calculations. Thus, system design is necessarily an interactive process.

A Path Selection Algorithm and Terrain Modeller which also address slope problems as well as discrete obstacles should be developed.

Despite the problems encountered, the edge detection approach has several major advantages, such as:

(1) a comparative lack of noise sensitivity, compared to previous sensor systems,

(2) the ability to distinguish obstacles from slopes and,

(3) a long-range permanent system memory.

Thus, the edge detection approach is a promising basis for a path selection system.

7. LITERATURE CITED

1. Krajewski, Marjan, Jr., "Development and Evaluation of a Short Range Path Selection System for an Autonomous Planetary Rover", RPI Master's Project Report, Rensselaer Polytechnic Institute, Troy, New York, April 1976.

2. Matthews, David W., "Development and Evaluation of a Short Range Path Selection System with Forward and Back-Up Mode Memory", RPI Master's Project Report, Rensselaer Polytechnic Institute, Troy, New York, May 1975.

3. Sharp, David R., "Development and Evaluation of a Three Beam Mid-Range Path Selection System", RPI Master's Project Report, Rensselaer Polytechnic Institute, Troy, New York, May 1975.

4. Reed, Martin A., "Recognition of Three Dimensional Obstacles by an Edge Detection Scheme", RPI Technical Report MP-45, Rensselaer Polytechnic Institute, Troy, New York, May 1974.

5. Sonalkar, Ranjan V. and C. N. Shen, "Simultaneous Bayesian Estimate of States and Inputs", Joint Automatic Control Conference, Purdue University, July 1976.

6. Leung, Kin-Ling, "Terrain and Obstacle Characterization for a Martian Roving Vehicle", RPI Master's Project Report, Rensselaer Polytechnic Institute, Troy, New York, May 1976.

7. Lee, C. Y., "An Algorithm for Path Connections and Its Application", <u>IEEE Transactions on Electronic Computers</u>, Vol. EC-10, pp. 346-365, September 1961.

8. Lallman, Fred, "Automatic Path Selection", RPI Master's Project Report, Rensselaer Polytechnic Institute, Troy, New York, May 1968.

9. Longendorfer, Betsy A., "User's Guide to the Edge Detection Approach", unpublished manuscript, Rensselaer Polytechnic Institute, Troy, New York, October 1976.

10. Sharp, David R., "Vehicle Dynamics Block: Simulation, Description and Flowcharts", unpublished manuscript, Rensselaer Polytechnic Institute, Troy, New York, September 1974.

11. Sonalkar, Ranjan V., "A Decision-Directed Rapid Estimation of States for Systems Subject to Unknown Input Sequences", RPI Doctoral Thesis, Rensselaer Polytechnic Institute, Troy, New York, November 1975.