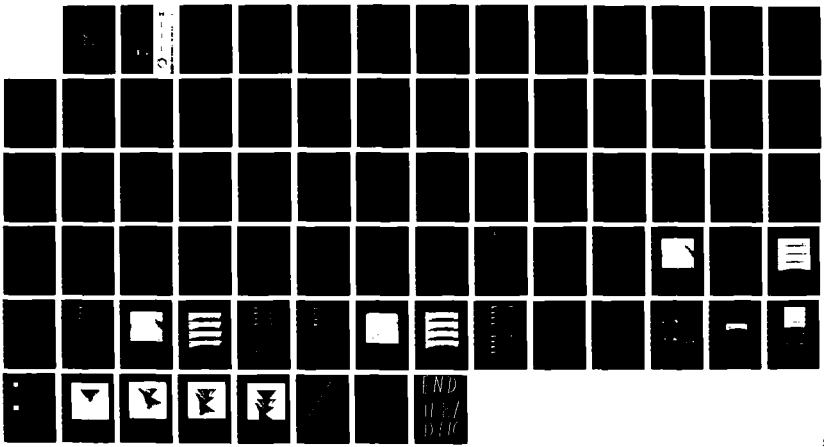
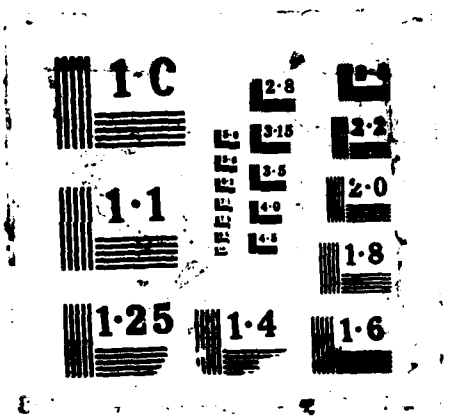
AD-A185 421 RANGE IMAGERY ALGORITHMS FOR THE DETECTION OF OBSTACLES BY AUUONOMOUS VYH..(U) MARYLAND UNIV COLLEGE PARK CENTER FOR AUTOMATION RESEARCH P A YEATCH ET AL. 1/1

UNCLASSIFIED JUL 87 CAR-TR-309 ETL-0461 DACA76-84-C-0004 F/G 17/7 NL




1·C
1·1
1·25
2·8
3·15
3·5
4·0
2·2
2·0
1·8
1·4
1·6

ETL-0461

(6)

AD-A185 421

Range imagery algorithms for the detection of obstacles by autonomous vehicles

Phillip A. Veatch
Larry S. Davis

Center for Automation Research
University of Maryland
College Park, MD 20742-3411

July 1987

DTIC
ELECTE
OCT 05 1987
S D
H

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Prepared for
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546

87

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

A185 421

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS N/A
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CAR-TR-309 CS-TR-1888	5. MONITORING ORGANIZATION REPORT NUMBER(S) ETL-0461
6a. NAME OF PERFORMING ORGANIZATION University of Maryland 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION U.S. Army Engineer Topographic Laboratories
6c. ADDRESS (City, State, and ZIP Code) Center for Automation Research College Park, MD 20742-3411	7b. ADDRESS (City, State, and ZIP Code) Fort Belvoir, VA 22060-5546
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Defense Advanced Research Projects Agency (DARPA) 8b. OFFICE SYMBOL (If applicable) ISTO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DACA76-84-C-0004
8c. ADDRESS (City, State, and ZIP Code) 1400 Wilson Boulevard Arlington, VA 22209-2308	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62301E PROJECT NO. TASK NO. WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
RANGE IMAGERY ALGORITHMS FOR THE DETECTION OF OBSTACLES BY AUTONOMOUS VEHICLES

12. PERSONAL AUTHOR(S)
Phillip A. Veatch and Larry S. Davis

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM _____ TO N/A	July 1987	75

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Image segmentation - range Autonomous navigation
09	02		
17	07		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Algorithms are presented which segment range images and classify regions as being navigable or unnavigable by a land vehicle. The algorithms are applied to data collected from an active laser range sensor mounted on an autonomous land vehicle and their comparative results are analyzed.

The sensitivity of various algorithms to uncertainty in the orientation of the range sensor is studied. Experiments on sensor calibration and image enhancement are also presented.

A computer model of an autonomous land vehicle and its environment is described which provides a valuable tool for investigating many issues of navigation with range sensors. Obstacle detection algorithms are used in conjunction with the model to demonstrate a vehicle navigating itself through an obstacle strewn world to a goal location.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Rosalene M. Holecheck	22b. TELEPHONE (Include Area Code) (202) 355-2767 22c. OFFICE SYMBOL ETL-RI-T

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Table of Contents

1 The Obstacle Detection Problem 1

2 Laser Range Scanners 6

2.1 General Characteristics 6

2.2 The ALV Range Scanner 7

3 Early Processing for Range Images 11

4 The Range Derivative Algorithm for Obstacle Detection 14

4.1 Central Ideas 14

4.2 Geometrical View of the Algorithm 17

4.3 Sensitivity to Scanner Perturbations 19

5 Implementation of the Range Derivative Algorithm 24

6 The ALV Simulator 29

7 Conclusions 35

Appendix A: Range Equation with Roll Error 36

Appendix B: Calculating Laser Beam Footprints on Planar Surfaces 38

Appendix C: Range Scanner Calibration 41

References 43

DTIC COPY INSPECTED 7

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

1. The Obstacle Detection Problem

A machine cannot understand what it cannot represent. In the domain of autonomous vehicles, a machine cannot avoid obstacles unless it can perceive an obstacle and correctly interpret its perception. This report presents an approach for imbuing an autonomous land vehicle with an awareness of obstacles.

While the algorithms presented in this work are applicable to any obstacle avoidance system, the implementation details are oriented toward the Autonomous Land Vehicle (ALV) project. The project's goal is to create a mobile robot capable of navigating along roadways and cross-country. The ALV is currently capable of navigating obstacle-free roads using only visual images. The next stage in the project is to traverse obstacle laden roads by integrating range information with the video data.

What is an obstacle? In its most general meaning, an obstacle is a region that a vehicle cannot or should not traverse. Avoiding regions where a vehicle is physically capable of traversing but for some reason should not go (such as not driving the wrong way down a one-way street) would require a level of artificial intelligence that is beyond the scope of this work.

Excluding places where a vehicle can go but shouldn't, one is left with regions that can be defined by their shape and material properties. Rocks, street signs, and steep slopes are all obstacles whose defining characteristic is their shape. Swamps and ice patches on the other hand may have sufficiently flat surfaces for navigation but their material properties make them obstacles for a land

vehicle that is not specially equipped. Although material properties are important for determining navigability, they are not readily measured by current remote sensing devices on autonomous vehicles, so for now the simplifying assumption will be made that regions can be adequately categorized by their geometry alone.

Given the presumption that obstacles will be defined by their shape, the next issue is how a region's geometry can best be determined by an autonomous vehicle. Perceiving geometry is essentially a depth perception problem. Approaches for creating depth (range) images generally fall into three categories: duplicating human visual ranging methods, contrived lighting methods such as structured light sensors, and direct, active ranging technologies. [jarvis83b] surveys many of these range finding techniques.

Human beings have a remarkable ability to form accurate depth perceptions using indirect methods. [haber80] details the wide variety of techniques that the human visual system employs to determine the distance of objects, including binocular perspective, texture and shadow clues, occlusion effects, and motion parallax. Unfortunately, persistent attempts to duplicate these methods using computer vision systems have resulted in limited successes that are not adequate for creating accurate range images while driving along a road.

Simple structured light schemes that use single spots of light or a plane of light to deduce range by triangulation can produce accurate results but are far too slow for real-time navigation. [le moigne84] and others have proposed using a grid pattern that covers the entire field of view of a camera to speed up the

image acquisition process. An alternative approach suggested by [schwartz83] and implemented by [carrihill85] requires taking only two intensity images: one with a constant illumination filter and one with a linear wedge filter in which the illumination is very bright on one side of the scene and decreases across the image. The ratio of the intensities in these two images provides range information throughout the entire scene and eliminates the slow process of obtaining and integrating many plane-of-light images. While grids and filters can form relatively high speed range images for short range indoor uses they have not yet been demonstrated in outdoor environments. The high variability of ambient lighting poses one handicap to outdoor structured light sensors. Another, perhaps more severe, problem is that the desired speed of autonomous vehicles requires range images that can 'see' at least 30–60 feet in front of the vehicle and preferably farther than this. It is unlikely that grids or filters could be formed over such large areas while maintaining adequate depth resolution.

Direct range sensing methods do not provide insight into human visual understanding but they are superior to indirect methods for creating fast, accurate range images. Radar, ultrasonics, and lasers are the most common active ranging systems. Unlike passive visual-based techniques they are not usually impaired by shadows, surface markings, or ambient light (or sound) sources. Furthermore, the transmitter and receiver in active systems are essentially coaxial which eliminates the "missing pieces" problem that structured light and stereo methods suffer from. There are no parts of a scene that one camera can see but the other camera or projector cannot. Certain exceptional conditions may

confuse active ranging systems (for example, atmospheric inversion layers can create false radar results [rogers79]) but these situations are infrequent and do not alter the general conclusion that active systems are more robust than current passive systems.

Software architecture designers for autonomous vehicles stress the neeed for rapid information processing [payton86, waxman85]. By using active ranging the entire computationally intensive step of converting a visual image by whatever means into a range image is eliminated, which allows processing of the range image to begin as soon as it is acquired.

Ultrasonics have been used for limited obstacle avoidance by mobile machines in both indoor [brooks85] and outdoor [parodi86] environments. While ultrasonic range sensors can reliably detect nearby obstacles they have poor directional resolution. In addition, they have a more severe problem with specular reflection causing the sensor's waves to be reflected outside of the receiver's view than do radar or laser sensors. [jarvis83a] points out that whenever an object's surface undulations are small in comparison to a sonic beam's wavelength the detector will not receive sufficient reflection to determine a range unless the angle of incidence is less than 40 degrees. A 50–60 KHz sonic range scanner will have wavelengths on the order of 0.5–0.67 cm as compared to a laser scanner's micrometer wavelengths. Clearly a much larger class of materials will specularly reflect an ultrasonic wave than will reflect a light wave.

Radar uses a shorter wavelength than sonic range scanners and thus has better resolution and specularly reflects off of fewer materials. However, even

very short wavelength radar is approximately three orders of magnitude longer than laser wavelengths and so has inferior resolution and worse specular properties. Short wavelength radar components are also generally more expensive than laser scanner parts.

The features, and deficiencies, of laser range scanners are discussed in more detail in Section Two. Section Three surveys low-level processing techniques for range images. The obstacle detection algorithm developed for this report is presented in Section Four along with an analysis of its sensitivity to perturbations of the range scanner. Implementation details and the results of applying the algorithm to outdoor range images are given in Section Five. A program for creating complex, synthetic range images and simulating ALV navigation is described in Section Six.

2. Laser Range Scanners

2.1. General Characteristics

There are two types of laser range scanners: time-of-flight (TOF) and phase-shift. TOF scanners measure how much time a laser beam takes to travel from the scanner to an object and back. They operate analogously to sonic scanners except that light travels one million times faster than sound. This creates a significant practical problem of measuring such small units of time. In order to resolve ranges to within 3 inches the detector must be able to accurately measure 170 picosecond units of time.

Phase-shift range scanners modulate the power amplitude of a laser beam and measure the phase difference between the reference wave form and the returning signal. Phase shifts are generally easier to measure than picosecond time differences. The main drawback of phase-shift devices is that one is not measuring an absolute range but rather the true range modulo the scanner's half-wavelength. The half-wavelength distance is called an "ambiguity interval" because range measurements are ambiguous by integer increments of the half-wavelength. For example, if the wavelength is 128 feet (i.e. an ambiguity interval of 64 feet) then a range measurement of 1 foot means that the true range to the object is 1 foot, or 65 feet, or 129 feet, etc. In practice, this is a serious problem only for hilly terrain. Note that the wavelength that determines the ambiguity interval is the power modulated wave, not the underlying electro-magnetic light wave. The underlying light has an extremely short wavelength, on the

order of microns, which gives the scanner its excellent directional resolution.

Older range scanners were seriously hindered by the length of time they needed to form a complete range image. In 1977, [nitzan77] built a phase-shift device that took two hours to acquire a 128x128 image. By 1983 [jarvis83a] reported a TOF scanner that could acquire a noisy 64x64 image in four seconds. A current state of the art range scanner is built by the Environmental Research Institute of Michigan (ERIM). It uses a phase-shift design to achieve 0.5 second acquisition rates for 8 bit 64×256 images.

2.2. The ALV Range Scanner

The processing of range images is affected by the details of the laser scanner that produces the image. This section describes the ERIM range scanner that is used by the Autonomous Land Vehicle project. More details are available in [larrowe86]. The parameters of the scanner are the result of optimizing many, often conflicting, features including power requirements, ability to penetrate atmospheric moisture, and safety hazards. The evolution of these features is described in [zuk83].

A 100 mW laser generating 0.82 um wavelength radiation is at the core of the range scanner. Modulation of the laser's power source creates a sinusoidal wave whose frequency (f) is 7.684 MHz. The scanner measures the phase shift (Δp) of the reflected laser beam. If ρ is the range to an object and Δt is the time the signal takes to travel the round-trip from the laser to the object and back then

$$\Delta p = 2\pi f \Delta t \tag{1}$$

and

$$\Delta t = \frac{2\rho}{c} \quad \text{where } c \text{ is the speed of light} \tag{2}$$

Combining equations (1) and (2) yields

$$\rho = \frac{\Delta p \; c}{4\pi f} \tag{3}$$

However, the measured Δp is the true Δp modulo 2π so the largest measurable value of Δp is less than 2π. Applying the frequency and the upper limit of Δp to equation (3) results in the largest measurable ρ having an upper limit of 64 feet. Therefore, 64 feet is the ambiguity interval of the ERIM scanner.

The intensity of the return signal is proportional to both the inverse square of the range and the reflectivity of the surface that reflects the signal. Most natural objects have sufficiently similar reflectivities that the intensity could be used to determine how many ambiguity intervals must be added to obtain the absolute range. Unfortunately, bare metal parts and specially reflective materials on street signs and road stripes render this method unusable for most situations.

Ambient sources of 0.82 μm light do not interfere with the return signal's measurement because the incoming light is filtered to remove radiation that is not modulated at the the laser's broadcast frequency of 7.684 MHz. It is extremely unlikely that natural sources would be modulated at this frequency (intentional interference in an adversarial situation is of course possible).

Figure 1 (from [larrowe86]) shows the major components of the scanner. The nodding mirror determines the plane that the beam will be in for any given

row of the range image. The angle within the plane is controlled by the rotating polygon mirror. Figure 2 illustrates the spherical coordinate system (θ, ϕ, ρ) that naturally describes a range image. The scanner, which is mounted on the ALV approximately nine feet above the ground, is at the origin (O) of the system. The positive Y axis points directly down toward the ground. The positive Z axis points out in the direction that the ALV is currently travelling. The length of the line segment OM is the range (ρ) to the point M.

The right triangle Ocb is in the YZ plane. $\angle cOb$ forms the vertical scan angle, ϕ. Each row in a range image is taken from a plane that contains the X axis and is ϕ degrees beneath the Z axis. The rectangle $OaMb$ is in this plane. $\angle aOM$ forms the horizontal scan angle, θ. Each column in a range image corresponds to a particular θ. This geometry results in the following relationships:

$$x = \rho\cos(\theta) \tag{4}$$

$$y = \rho\sin(\theta)\sin(\phi) \tag{5}$$

$$z = \rho\sin(\theta)\cos(\phi) \tag{6}$$

The 64 rows of the image are at equally spaced values of ϕ, and the 256 columns are at evenly spaced values of θ. An ERIM range image has a 30 degree vertical field of view in which ϕ goes from approximately 6 degrees to 36 degrees. The 80 degree horizontal field of view extends from a θ of 130 degrees to a θ of 50 degrees. Although the total magnitudes of the fields of view are fixed, the orientations can be altered either by internal controls in the scanner or by moving the external platform that the scanner is mounted on. Some practical considerations

in determining the scanner's field of view are discussed in Appendix C.

The ERIM scanner has a vertical sampling interval of 0.3125 degrees and a horizontal sampling of 0.46875 degrees. Since the laser beam has an angular divergence of 0.5 degrees, a scene is densely sampled. This removes the need for sophisticated interpolation techniques such as those proposed by [boult85] or [choi85] for sparse range data.

Multiple objects at various ranges may occur within the 0.5 degree solid cone that forms a single pixel's field of view. The signal that returns to the scanner will indicate a range that is a complex average of all the ranges encountered within the cone. For example, if half of a cone intercepts a tree and the other half travels on to the ground then the returning signal would yield a value that is somewhere between the distance to the tree and the most distant ground that is within the cone. This is called the mixed pixel problem. A strategy for avoiding range errors resulting from mixed pixels is presented in Section 5.

Due to the ambiguity effect, output range values are all between zero and 64 feet. They are quantized into three-inch units so that the final output of the ERIM scanner is a 64×256 array of 8 bit values ranging from zero to 255.

3. Early Processing for Range Images

There have been many proposed approaches for the initial segmentation of range images. Early work by [duda79] involved converting a pixel's spherical coordinates into Cartesian coordinates and then extracting horizontal planes by histogramming the vertical coordinate of each pixel. Vertical planes were found by projecting points onto the ground plane and detecting lines with Hough transforms. Pixels that were not found to be in vertical or horizontal planes were grouped by similar gray levels. [parvin86] also converted each pixel's spherical coordinates to Cartesian coordinates but then $\partial y/\partial x$ and $\partial y/\partial z$ were estimated at each pixel. Pixels were then grouped into common planes if their derivatives were sufficiently similar. Parvin reported that applying this procedure to a 128×128 range image took 4.5 minutes on a VAX 11-750, much too long for navigation purposes.

The most common approach for initial segmentation is probably the fitting of surface normals to each pixel. [hebert82] calculated surface normals and represented them on a Gaussian sphere so that planes appeared as points (or clusters), cylinders were circles, and other 3-dimensional shapes could be similarly extracted. [sethi84] projected surface normals onto an image plane and found equi-magnitude contours. [yang84] did the same but extended the analysis by also finding equi-orientation contours. [hoffman86a, hoffman86b, jain86] all used surface normals to classify surfaces as being planar, convex, or concave as the first step in object recognition algorithms. [lin86] extracted surface patches with common surface normals to match against possible models in another object

recognition scheme.

Higher order polynomials have also been fit to range data. [hall82] characterized surface patches in synthetic range images by their quadric coefficients. [jain86], however, reported that quadric fits did not work well for real range images. [sharma86] compared quadric and planar fits on outdoor range imagery for extracting road edges and found no advantages to using the slower quadric method. [olivier86] proposed a least-squares cubic fit and [vemuri84] used splines under tension.

[inokuchi82] applied a ring operator to range images in conjunction with a discrete Fourier transform to generate amplitude and phase information for characterizing pixels.

Outside of projects related to the Autonomous Land Vehicle, little work has been reported on outdoor range images. The investigations previously mentioned in this section, with the exception of [hebert82, jain86, sharma86], were all done using synthetic range images or indoor scenes with very fine gradations. The range images used by [hoffman86b], for example, had an ambiguity interval of approximately eight inches and a range resolution of 0.03125 inches. This level of fine resolution allows the use of more sensitive algorithms than are applicable to outdoor range data where the ambiguity interval is 64 feet and the resolution is theoretically three inches but, in practice, usually closer to six inches. The noise and the noncontinuous nature of outdoor scenes may make methods more complex than planar surface fitting unsuitable for outdoor navigation. In addition, complex techniques take longer to run which can severely limit the speed at

which they can control a moving vehicle.

[sharma86] and [hebert85] have shown that fitting planes can be used for low-level processing of outdoor range images, but even simpler algorithms may also be adequate. The fastest of the ALV obstacle detection algorithms, range differencing, simply subtracts the range image of an actual scene from the expected range image of a flat plane. While rapid, this technique is not very robust. Small errors in the orientation of the scanner or a mild slope in the land will result in false indications of obstacles. We propose using the first derivatives of the range with respect to the vertical and horizontal scan angles as an improved, fast obstacle detector. The details of this algorithm, an analysis of its robustness, and the results using actual outdoor range data are given in the following sections.

4. The Range Derivative Algorithm for Obstacle Detection

4.1. Central Ideas

When deciding if a surface is an obstacle or not the pertinent feature is the change in height across the surface. If the change is too rapid then the surface is unnavigable. A surface normal contains the necessary information on the change in height but calculating surface normals is computationally intensive. The surface normal at a point is a function of $\partial y / \partial x$ and $\partial y / \partial z$. Simply calculating the slope, $\partial y / \partial z$, would provide significant information concerning a surface's navigability. However, computing the slope directly from a range image is not much easier than calculating a surface normal. What can be done very quickly, though, is finding $\partial \rho / \partial \theta$ and $\partial \rho / \partial \phi$. The following section first shows how $\partial \rho / \partial \phi$ can be closely linked to $\partial y / \partial z$ and then how $\partial \rho / \partial \theta$ can be a measure of $\partial y / \partial x$. Using our knowledge of how range derivatives reflect changes in height across a surface we can then design a rapid obstacle detection algorithm.

The differential of a function $y(\phi,\rho,\theta)$ can be written as

$$dy = \frac{\partial y}{\partial \phi} d\phi + \frac{\partial y}{\partial \rho} d\rho + \frac{\partial y}{\partial \theta} d\theta \tag{7}$$

If θ is held constant so that the $d\theta$ term is zero then equation (7) applied to equation (5) gives

$$\Delta y = \rho \sin\theta \cos\phi \, \Delta\phi + \sin\theta \sin\phi \, \Delta\rho \tag{8}$$

where the infinitesimal terms dy, $d\phi$, and $d\rho$ have been replaced by their finite Δ equivalents. In a similar fashion, equation (6) can be differentiated to yield

$$\Delta z = -\rho \sin\theta \sin\phi \, \Delta\phi + \sin\theta \cos\phi \, \Delta\rho \tag{9}$$

Dividing Δy by Δz yields

$$\frac{\Delta y}{\Delta z} = \frac{\rho \cos\phi \, \Delta\phi + \sin\phi \, \Delta\rho}{-\rho \sin\phi \, \Delta\phi + \cos\phi \, \Delta\rho}$$

$$= \frac{\frac{\Delta\rho}{\rho} \frac{\tan\phi}{\Delta\phi} + 1}{\frac{\Delta\rho}{\rho} \frac{1}{\Delta\phi} - \tan\phi} \tag{10}$$

If ϕ is held constant then equation (7) becomes

$$\Delta y = \sin\theta \sin\phi \, \Delta\rho - \rho \sin\phi \cos\theta \, \Delta\theta \tag{11}$$

and equation (4) can be differentiated to obtain

$$\Delta x = \cos\theta \, \Delta\rho - \rho \sin\theta \, \Delta\theta \tag{12}$$

Dividing equation (11) by equation (12) and regrouping yields

$$\frac{\Delta y}{\Delta x} = \frac{\frac{\Delta\rho}{\rho} \frac{\tan\theta}{\Delta\theta} \sin\phi - \sin\phi}{\frac{\Delta\rho}{\rho} \frac{1}{\Delta\theta} - \tan\theta} \tag{13}$$

Excluding the terms in equations (10) and (13) that we know a priori, we see that the changes in height in the x and z directions are a function of $\Delta\rho/\rho$. If we used some approximation of ρ we would have a direct relationship between the easily calculated $\Delta\rho$ for a fixed θ or ϕ at a pixel and the slopes at that pixel. Our experiments with real range data suggest that the following is an adequate approximation:

$$\rho \approx \frac{H}{\sin\theta \sin\phi} \tag{14}$$

where H is the height of the range scanner above the ground. Equation (14) comes from substituting H for y in equation (5). In hilly terrain this

approximation is probably not adequate but it works well for many scenes and it will be shown in the next section that the derivative algorithm that uses this approximation is less sensitive to orientation errors than other algorithms of similar simplicity and speed.

Using equation (14) we can calculate what $\Delta\rho$ would be at each pixel if the slopes were zero. The difference between this predicted $\Delta\rho$ and the actual $\Delta\rho$ found in a range image is a measure of the actual slope. Large differences between predicted and actual $\Delta\rho$'s will be formed by edges of objects as well as surfaces with steep slopes. Thresholding the absolute values of these differences yields pixels that are likely to be on obstacles.

One could of course simply threshold the actual $\Delta\rho$'s without first subtracting the expected $\Delta\rho$'s and assume that large $\Delta\rho$'s indicate surfaces that have steep slopes and hence are not navigable. This approach, however, would severely reduce one's capability to detect obstacles. A perfectly flat surface will yield a $\Delta\rho$ of about 10 if it is 60 feet away but the same surface at a range of 10 feet only has a $\Delta\rho$ of about 0.3. This wide range in $\Delta\rho$'s leaves any thresholding algorithm in a bind. Small threshold levels would find nearby obstacles but more distant flat surfaces would be falsely labelled as obstacles. Conversely, larger threshold levels would hide significant obstacles that are near the range scanner. What is needed is a variable threshold setting. This approach points out another way of looking at the range derivative algorithm: we are, in essence, creating a variable threshold that changes across an image based on expected $\Delta\rho$'s. While this simplistic view is a useful description, the derivative

algorithm is founded on the mathematical relationships between $\Delta\rho$ and a surface's slopes and is not a randomly chosen heuristic for setting variable threshold levels.

4.2. Geometrical View of the Algorithm

Equation (10), which showed the relationship between $\Delta\rho$ and a surface's slope, was derived from the differential equations for dy and dz. Figure 3 illustrates the following description of how equation (10) can also be derived from the geometry of a range image in the neighborhood about a pixel. We wish to estimate $\Delta y / \Delta z$ at some Range[i,j] in an image. Consider projecting Range[i-1,j] and Range[i+1,j] into the YZ plane. Their projections would be the line segments, respectively, $\overline{ob}$ and $\overline{oa}$ as shown in Figure 3. If $\overline{da}$ is a line segment that is parallel to the Z axis then the tangent of $\angle dab$ is equal to $\Delta y / \Delta z$ between Range[i-1,j] and Range[i+1,j]. If $\Delta\phi$ is small then the tangent of $\angle dab$ is likely to be a reasonable estimate of the slope at Range[i,j]. This tangent can be found in the following manner. As Figure 3 s w

$$\angle dab = \phi_a + \gamma \qquad \text{where } \phi_a = \phi_{row=i+1}$$

and

$$\Delta y / \Delta z = \tan(\angle dab) = \tan(\gamma + \phi_a).$$

Applying the tangent summation formula to the last equation gives us

$$\Delta y / \Delta z = \frac{\tan\gamma + \tan\phi_a}{1 - \tan\gamma \tan\phi_a} \tag{15}$$

If $\angle bca$ were 90 degrees then $\tan\gamma = |\overline{bc}|/|\overline{ca}|$ (where $|\overline{bc}|$ is the length of segment $\overline{bc}$). $\angle bca$ is actually equal to $90 + \Delta\phi/2$ but $\Delta\phi/2 = 0.476$

degrees so we can reasonably approximate $\angle bca$ by 90 degrees which gives us

$$\tan\gamma \approx \frac{|\overline{bc}|}{|\overline{ca}|} \tag{16}$$

By definition, c is the point on $\overline{oa}$ where $|\overline{oc}| = |\overline{ob}|$. This means that $|\overline{ca}| = |\overline{oa}| - |\overline{ob}|$. Since $\overline{oa}$ and $\overline{ob}$ are projections of ranges into the YZ plane we know that

$$|\overline{oa}| = \text{Range}[i+1,j]\sin\theta = \rho_{i+1}\sin\theta \tag{17.1}$$

and

$$|\overline{ob}| = \text{Range}[i-1,j]\sin\theta = \rho_{i-1}\sin\theta \tag{17.2}$$

So,

$$|\overline{ca}| = \Delta\rho\sin\theta \qquad \text{where}\Delta\rho = \rho_{i+1} - \rho_{i-1} \tag{17.3}$$

If we draw a line from the origin to the midpoint of $\overline{bc}$ we will bisect $\Delta\phi$ and have a right triangle in which $\overline{ob}$ is the hypotenuse and $\overline{bc}\ /\ 2$ is the length of the side that is opposite of the $\Delta\phi\ /\ 2$ angle. Hence we know that

$$\frac{|\overline{bc}|/2}{|\overline{ob}|} = \sin(\Delta\phi/2) \tag{18}$$

But $\Delta\phi/2$ is quite small so $\sin(\Delta\phi/2) \approx \Delta\phi/2$. Applying this approximation and equation (17.2) to equation (18) yields

$$|\overline{bc}| = \rho_{i-1}\sin\theta\,\Delta\phi \tag{19}$$

Equations (17) and (19) show us that equation (16) can be restated as

$$\tan\gamma = \frac{\rho_{i-1}\sin\theta\,\Delta\phi}{\Delta\rho\sin\theta} = \frac{\rho_{i-1}\,\Delta\phi}{\Delta\rho} \tag{20}$$

Combining equations (15) and (20) and rearranging terms yields

$$\frac{\Delta y}{\Delta z} = \frac{\frac{\Delta \rho}{\rho_{i-1}} \frac{\tan\phi_a}{\Delta\phi} + 1}{\frac{\Delta \rho}{\rho_{i-1}} \frac{1}{\Delta\phi} - \tan\phi_a} \tag{21}$$

Comparing equation (21) to equation (10) we see that the two approaches for relating $\Delta y / \Delta z$ to $\Delta \rho$ yield the same result. A similar correlation could also be made for $\Delta y / \Delta x$ and the projections into the XY plane of Range[$i,j+1$] and Range[$i,j-1$].

4.3. Sensitivity to Scanner Perturbations

To know the shape of the world from a range image it is first necessary to know where the range scanner is and what its orientation is. When an ERIM range scanner is mounted on an ALV, the scanner's location (especially its height above the ground) can be found with sufficient accuracy. Its orientation, on the other hand, has been surprisingly difficult to measure. Martin Marietta, the integrating contractor for the ALV project, has reported that determining the roll, pitch, and yaw of the scanner has been a serious continuing problem. For this reason it is useful to study the sensitivity of the range derivative algorithm to errors in scanner orientation and to compare this sensitivity to that of the range difference and height difference algorithms. For height differencing, one first converts the range image to Cartesian coordinates and then differences the expected height (y coordinate) instead of the range.

There are many ways one could measure errors in obstacle detection algorithms. We chose to consider what happens when various algorithms are applied

to the range image of a flat plane in which the image was taken by a scanner rotated in some manner. The three algorithms studied were: our range derivative algorithm (broken down into the θ derivative and the ϕ derivative steps), the height difference algorithm, and the range difference algorithm. The output of these algorithms when applied to an image of a flat plane (i.e. a plane parallel to the XZ plane at a known $Y = H$) should be zero at each pixel. If the range image is taken by a scanner rotated about a particular axis then the results of applying an algorithm to the perturbed image is a good measure of the algorithm's sensitivity to that type of rotation. The following rotations were used to generate perturbed range images:

1) Rotating the scanner about the X axis (i.e. pitch error) by δ_x so that

$$\rho = \frac{H}{\sin\theta \sin(\phi+\delta_x)} \tag{22}$$

2) Rotating the scanner about the Y axis would not alter the range image because images of flat planes are invariant to this type of rotation (this of course is not generally true for planes that do not have a constant Y value). However, any error in the timing or orientation of the polygon mirror that determines the θ of the laser beam could lead to θ being off by some δ_y (which we will call the yaw error). If the data stream was not properly synchronized one could also believe that the range at some $\theta + \delta_y$ was the range for θ. Yaw error has the form

$$\rho = \frac{H}{\sin(\theta+\delta_y) \sin\phi} \tag{23}$$

3) If the scanner is rotated about the Z axis (roll error) by some δ_z then

$$\rho = \frac{H}{\cos\delta_z \sin\theta \sin\phi + \sin\delta_z \cos\theta} \tag{24}$$

A derivation for equation (24) is presented in Appendix A.

4) The most severe perturbation considered was the combination of roll, followed by pitch, followed lastly by yaw, so that

$$\rho = \frac{H}{\cos\delta_z \sin(\theta+\delta_y) \sin(\phi+\delta_x) + \sin\delta_z \cos(\theta+\delta_y)} \tag{25}$$

Equations (22–25) were used to produce four different range images in which H was assumed to be nine feet. The derivative, height, and range algorithms were applied to each image. When applying algorithms to real outdoor range images it is necessary to average ranges over a neighborhood to suppress noise. For these perturbation sensitivity experiments we used the following unnormalized summations at each pixel$[i,j]$:

For the θ derivative,

$$\Delta\rho = \sum_{k=i-1}^{i+1} \text{Range}[k,j+1] - \sum_{k=i-1}^{i+1} \text{Range}[k,j-1] \tag{26}$$

For the ϕ derivative,

$$\Delta\rho = \sum_{k=j-1}^{j+1} \text{Range}[i+1,k] - \sum_{k=j-1}^{j+1} \text{Range}[i-1,k] \tag{27}$$

For the height derivative,

$$y = \sum_{k=j-1}^{j+1} \text{Range}[i,k] \sin(\theta_{col=k}) \sin(\phi_{row=i})$$

For the range derivative,

$$\rho = \sum_{k=j-1}^{j+1} \text{Range}[i,k]$$

Table 1 summarizes the results of these experiments. The table contains two entries for each combination of algorithm and perturbation. The top entry is the largest absolute value in the entire image and represents a worst-case scenario. In many scenes, however, the road will be near the center of the image's horizontal field of view and large errors on the periphery are not critical. This scenario is represented by the bottom entry which is the largest absolute value within the central 30 degrees of the image (i.e. $105 \leq \theta \leq 85$).

Several important trends emerge from Table 1. The θ derivatives were very insensitive to all four rotational perturbations. The ϕ derivatives were somewhat more sensitive. When the entire image was considered, the maximum ϕ derivative errors for each rotation were always at least 25% less than the maximum height difference errors. Within the central 30 degrees of the horizontal field of view, the maximum ϕ derivative errors were 45%–75% less than the maximum height difference errors. The range difference algorithm was very sensitive to all forms of rotations. In several instances the range difference errors were a full order of magnitude larger than the derivative errors. These results clearly show

that the derivative algorithms are more robust under rotational uncertainties than either the height difference or the range difference algorithms.

For these experiments the true range values without ambiguity intervals were used. Ambiguity intervals were not simulated so as to avoid mixing the separate issues of ambiguity interval compensation and sensitivity to scanner perturbations.

5. Implementation of the Range Derivative Algorithm

There are several practical considerations that affect the implementation of a range derivative algorithm. Chief among these are: choosing an accurate $\Delta\rho$, reducing noise in an image, avoiding ambiguity interval errors, and inhibiting errors from mixed pixels. Our approach for these considerations are explained in this section and the results from applying them to actual range images are presented.

Equations (26) and (27) give the actual equations used to calculate $\Delta\rho$ for the θ and ϕ derivatives, respectively. (The term "ϕ derivative" is used loosely to mean $\Delta\rho$ calculated for a known $\Delta\phi$ while θ is held constant. Similarly, "θ derivative" means $\Delta\rho$ calculated for a known $\Delta\theta$ while ϕ is constant.) For the ϕ derivatives the choice of calculating $\Delta\rho$ across two rows was a compromise between two conflicting goals. On the one hand, one wants $\Delta\phi$ to be as small as possible so as to insure that the resultant $\Delta\rho$ accurately reflects the slope at the pixel $[i,j]$. The smaller $\Delta\phi$ is, the less likely it is that a surface will be sufficiently curved to cause an inaccurate $\Delta\rho$. On the other hand, decreasing the size of $\Delta\phi$ also decreases the magnitude of the $\Delta\rho$'s, which can impair the accuracy of the $\Delta\rho$ measurements. This is especially true for small $\Delta\rho$'s due to the quantization of range measurements into three inch units. The choice of using $\Delta\theta$ across two columns for the θ derivatives was made for the same reasons.

The summation of the ranges in a three column neighborhood for the ϕ derivative (and a three row neighborhood for the θ derivative) has been found to be an effective averaging method. A more sophisticated smoothing operator, the

Symmetric-Nearest-Neighbor operator ([harwood84]), was also tested on range images but it did not significantly improve obstacle detection.

The ambiguity interval problem has been approached in two different ways. For actual range images taken from the ALV we have found that if the vehicle is not travelling over very hilly territory, simply deleting the upper few rows of the image removes most of the pixels that are beyond the first ambiguity interval ($\geq$ 64 feet). This does not affect the obstacle detection algorithm significantly because the laser beam has spread out into a relatively wide cone by the time it has travelled beyond 50–60 feet, resulting in mixed pixels that are of little use for accurate obstacle detection. For example, at a range of 55 feet, a laser beam whose central axis strikes a planar surface at an incident angle of 5 degrees will form an elliptical footprint with major and minor axes, respectively, of 5.5 feet and 0.5 feet. (Appendix B derives a general equation for the size of laser beam footprints on planar surfaces.)

A more time consuming but somewhat more precise approach has been to examine each column from bottom to top in the image. Whenever adjacent pixels go from large values suddenly to very small values it is reasonable to assume that an ambiguity interval has been reached and that all pixels in the column beyond this point should have an additional 256 added to their ranges. This approach was used in the ALV Simulator that is described in the next section.

Figure 4 is a visual picture of Martin Marietta's ALV test track in Denver, Colorado. The picture was taken by the video camera that is mounted on the ALV for visual navigation. Figure 5 is a comparison of the fields of view of the

video camera and the range scanner. It is an outline of how much territory each device can 'see' on a flat road in a single frame. The camera's view extends further but is narrower than the range scanner's view. Since the camera's vertical field of view goes above the horizon, its view theoretically extends to infinity. The range scanner's view begins several feet closer to the vehicle than does the camera's. This can be clearly seen in Figure 6a: a montage of four range images in which the top image was taken by the ALV at the same time as Figure 4. Moving down from the top of the montage, each image was taken five feet further down the road. The carton in the lower right corner of Figure 4 can be seen in the top two images of Figure 6a. The cone that is on the right side of the road about half way up Figure 4 is present in all four range images. The cone is not very apparent in Figure 6a but it is clearly marked as an obstacle in Figure 6b, which is the thresholded output of the θ derivative algorithm. Figure 6c is the thresholded output of the ϕ derivative algorithm. The small, dark blob that is located in the center columns of the top few rows of each image in Figure 6a and that is marked as an obstacle in Figures 6b and 6c does not correspond to any actual object in the scenes but rather is spurious data generated by the range scanner's electronics.

All of the binary images in this section were thresholded manually. Automatic thresholding was not a topic of research so while the effect of differing threshold levels was observed, no attempt was made to determine how optimal levels should be set.

Figure 6d is the result of two processing steps. First, the binary images 6b and 6c were ORed together and then isolated obstacle pixels (i.e. pixels that had no 8-way neighbors marked as obstacles) were deleted.

Figure 7 is another visual image from the Denver test track. It was taken by the ALV at the same location as the top range image in the montage shown in Figure 8a. The cone in the lower right corner of Figure 7 can be seen in the top two range images. The cone is clearly marked in the corresponding thresholded θ derivative images (Figure 8b) and ϕ derivative images (Figure 8c) as well as in the logical OR of the two derivatives (Figure 8d). Isolated pixels were removed from Figure 8d.

The spurious dark blob from the scanner's electronics that was present in the previous set of images is also present in Figures 8a–d. In Figure 8a, the faint light patch to the left of the dark blob corresponds to the box in Figure 7 on the left side of the road about 30 feet beyond the cone. The box can be seen in Figures 8b–d in all four images. The boxes on the right side of the road in Figure 7 are just beyond the detection range of the range scanner.

Location errors due to mixed pixels can be minimized by using pixels that come from the interior of an obstacle and avoiding pixels at the edges. If one assumes that obstacle surfaces are usually not concave then the sign of the derivative at an edge pixel indicates the direction of the center of the obstacle. For ϕ derivatives (as defined in equation (27)), negative $\Delta\rho$'s occur at the top edges of obstacles. Positive $\Delta\rho$'s occur at bottom edges. For θ derivatives (as defined in equation (26)), left edges have negative $\Delta\rho$'s and right edges have

positive values.

Our implementation of this strategy has two parts. First, if an obstacle pixel is at location $[i,j]$ in the image, use the sign of the derivative to determine which direction is likely to be away from the edge and toward the interior. The direction determines which of the adjacent pixels will be used for associating a location with the obstacle found at $[i,j]$. If, for example, $[i,j+1]$ is determined to be toward the obstacle's center then the three-dimensional location of the pixel $[i,j+1]$ in the range image will be used to place the obstacle that was found at $[i,j]$ in the thresholded image.

Figures 9–11 show the improved positioning of obstacle pixels achieved by this mixed-pixel minimization algorithm. Figure 9 is the video image of a box located 25 feet in front of the ALV. A montage of range images in which the box is 25, 30, 35, and 40 feet from the ALV is given in Figure 10a. The obstacle pixels from the combined θ and ϕ derivatives are shown in Figures 10b and 10c. In Figure 10b, the pixels have been shifted toward the interior by the algorithm while in Figure 10c they have not been shifted. The projection onto the ground plane of the obstacle pixels is shown in Figure 11a (with shifting) and in Figure 11b (without shifting). The box's pixels are circled in both of these figures. The unshifted pixels are clearly more scattered than the shifted pixels. The trapezoid in Figures 11a and 11b is an approximate outline of the scanner's field of view, similar to the trapezoid in Figure 5. Figures 12a and 12b are the ground projections of the box at 40 feet with and without shifting, respectively.

6. The ALV Simulator

Maintaining an ALV is both expensive and time consuming. Furthermore, changes in weather and the movement of the sun make it very difficult to reproduce conditions exactly for testing purposes. A robot arm mounted with a range scanner and video camera that traverses scale model environments has been used to provide an efficient and relatively inexpensive testing ground for navigation programs [dementhon.)]. This section describes a computer simulation program that goes beyond mechanical modelling and provides a software testbed for autonomous navigation algorithms by simulating the movement of an ALV and constructing the video and range images that would be in the ALV's field of view as the vehicle moves. The program is based on an image flow simulator described in [sinha84].

The simulation process has four major components. First a synthetic world must be specified and a model created. After this initializing step a loop is begun consisting of: 1) creating visual and range images based on the ALV's current location, 2) applying navigation algorithms to determine where the ALV is to move to next, and 3) calculating and then applying a transformation matrix that 'moves' the ALV to its next location.

The simulator can model spheres, parallelepipeds, planar surfaces, cones, and cylinders. These objects can be of any dimensions. They can be translated and rotated in any fashion and may be positioned so that an object is partially or wholly inside of another object (an important property when constructing complex scenes from these basic building blocks). From the user's perspective, the

world that the ALV will drive through is specified by a list of objects. Each object consists of a shape (i.e. sphere, cone, etc.) and parameters describing its size, location, and orientation. Inside the simulator, each object consists of an array of surface control points. On a cone, for example, the control points are the tip of the cone and several equally spaced points on the rim of the cone's base. The centroid of an object is initially placed at the origin of the coordinate system and the locations of its surface points are set according to its shape and size. A transformation matrix is calculated that 'moves' the object from the origin to its location and orientation in the world. The object is then positioned by multiplying its control points by this matrix.

After each object has been positioned a visual image is calculated based on a perspective projection in which the focal point is at the origin of the coordinate system and the image plane is placed in front of it at $z =$ focal length. The focal length and the field of view are parameters that the user provides at the start of the program. These parameters, and all other input to the program, can either be read from a file or entered interactively in response to prompts.

The visual image is created by breaking an object's surface into triangles in which the object's control points are the vertices of the triangles. This triangulation obviously decreases the accuracy of the range image for curved surfaces but any desired level of accuracy can be achieved by increasing the number of control points.

An intensity value is calculated for the center of each triangle and all points within the triangle are assumed to have the same intensity. This assumption

leads to artifacts in the visual image. The simulator's main purpose is to aid range image research so no attempt has been made to eliminate the gray level artifacts.

The gray levels can be created with the light source at any position. Surface reflections are assumed to be Lambertian and all objects have an equal albedo (it would be a simple extension to add variable albedos). No compensation is made for lowering intensity due to increased distance from the image plane and the light source.

The vertices of each triangle are projected into the image plane and pixels within the projected triangle are all given the same gray level. At first, pixels were assigned z values based on simple interpolation of the z values of the three projected vertices. However, linear interpolation between rows of an image was found to be too inaccurate. Instead, pixels on the edge of the triangle in each row of the image are projected back out to the object and their actual z values are calculated. Within a row, linear z interpolation between the two edge pixels is usually sufficient. Hidden surfaces are removed by comparing z values at each pixel and choosing the surface that has the minimum z value.

From the visual image and the corresponding z distances we can create an equirectangular range image whose pixels are spaced at equal linear intervals on the image plane. However, the ERIM range scanner produces images that are at equal angular intervals on the image plane so the equirectangular image is resampled to accurately simulate the ALV's range scanning process. Interpolation of the equirectangular image is done using an intentionally crude algorithm to

introduce noise into the system (triangulating and digitizing the image of the objects has already introduced some noise). The final equiangular range image has all of the properties of an image produced by an ERIM scanner mounted on an ALV including the same field of view, eight bit range values, and 64 foot ambiguity intervals.

Once the range image is created, the program's modularity allows the use of any navigation algorithms to determine where the ALV should move to. In the program's current configuration the range derivative algorithm is applied to the equiangular range image and the resultant binary obstacle image is mapped from spherical coordinates into the Cartesian XZ ground plane. The ground plane map initially has four types of pixels: 1) traversable terrain, 2) obstacles or unnavigable terrain, 3) areas whose traversability is unknown because they are hidden by an obstacle (i.e. shadow regions), and 4) areas whose traversability is unknown because they are outside of the field of view of the simulated range sensor. The path planner will treat the ALV as if it were the size of a single pixel so a boundary the width of the ALV's radius is grown around all obstacle and shadow pixels.

Each pixel in a ground plane map corresponds to one square foot and the entire map covers approximately 65,000 square feet. The vehicle is always at the center of the current map. In addition to the regions seen in the most recent range image, the current map also contains information gathered from previous images and projected into the current coordinate system's ground plane.

At the start of the simulation the program requests the coordinates of the ultimate goal for the ALV. A straight line from the current location to this goal is plotted and a move along it is calculated. The endpoint of the move is passed to the path planner which tries to find a path through the ground map from the current location to the endpoint. The path planner was developed by Kambhampati and Davis and is described in [kambhampati86]. It uses a hierarchical algorithm based on a quadtree division of the ground map. The planner assumes that the vehicle can only travel through pixels that are marked as traversable. [puri87] describes an advanced version of this planner that determines when the vehicle should try to move to a different vantage point so as to see if shadow regions are actually traversable. This can significantly improve the vehicle's path when tall obstacles obscure large regions.

If the planner fails to find a path to the first endpoint a series of heuristics are used in sequence to select alternate subgoal locations. Each subgoal is passed, one at a time, to the path planner until one is found that can be reached. If all of the heuristics are exhausted without a reachable subgoal being found, the program notifies the user and gracefully terminates.

Once the endpoint of the next move is found a transformation matrix is calculated that will place the origin of the coordinate system at this new location. This matrix, when applied to each object's control points, will result in the next visual and range images being the scene that an ALV would see if it were driven to the endpoint. The matrix is fashioned so that the vehicle will be facing the ultimate goal location (other constraints on what direction the vehicle should be

facing or how long each move should be are adjustable parameters in the program). If the move's endpoint is the same as the goal location the program terminates. Otherwise the transformation matrix is applied, the new visual image is found and the program begins another pass at moving the simulated vehicle toward its goal.

A typical trip by the ALV through synthesized terrain is illustrated in Figures 13–21. The visual images at the start of each move are shown in Figure 13. The equirectangular range image at the start of the trip is given in Figure 14. It corresponds to the visual image labelled Time 0 in Figure 13. A montage of the equiangular range images is presented in Figure 15. The four scenes in the montage, in order from top to bottom, are from Times 0, 1, 2, and 3. Figure 16 shows the obstacle pixels found in each equiangular range image. These pixels are mapped into the ground plane in Figures 18–21. The solid black regions in the ground maps are obstacles. White areas are navigable terrain. Horizontal stripes are shadow regions while vertical stripes delimit the grown boundaries surrounding obstacles and shadows. Regions outside of the range scanner's field of view are gray. A key to these markings is provided in Figure 17.

7. Conclusions

We have shown how range derivatives can be used for fast, reliable obstacle detection by an autonomous vehicle. The range derivative algorithm works well for relatively flat, on-road scenes. It has significantly better performance, when the range scanner is perturbed, than other fast obstacle detection methods. For hilly terrain it will be necessary to use surface normals or the full $\Delta\rho/\rho$ term in place of the simpler $\Delta\rho$ approximation that was used in this report.

The ALV simulator is a useful testbed for developing range image and navigation algorithms. By freeing researchers from the costly and time-consuming requirements of an actual vehicle it can accelerate the algorithm development process. It would be useful if the simulator was extended to improve its ability to model rough, outdoor terrain realistically.

APPENDIX A

Range Equation with Roll Error

The derivation of equation (24) is based on three rotations of the scanner's coordinate system. The rotations begin with the Cartesian framework that is shown in Figure 2. We first wish to rotate about the Z axis by δ_z. The relationships between the unit vectors in the new coordinate system and the old unit vectors are

$$\hat{x}_1 = \hat{x}\cos\delta_z + \hat{y}\sin\delta_z$$
$$\hat{y}_1 = -\hat{x}\sin\delta_z + \hat{y}\cos\delta_z$$
$$\hat{z}_1 = \hat{z}$$

The range scanner scans downward by an angle ϕ which is equivalent to rotating the coordinate system about the X1 axis. Note that this is a left-handed rotation so

$$\hat{x}_2 = \hat{x}_1$$
$$\hat{y}_2 = \hat{y}_1\cos\phi - \hat{z}_1\sin\phi$$
$$\hat{z}_2 = \hat{y}_1\sin\phi + \hat{z}_1\cos\phi$$

The horizontal scan angle θ is now the equivalent of rotating about the $Y2$ axis by θ degrees. This is also a left-handed rotation which yields

$$\hat{x}_3 = \hat{z}_2\sin\theta + \hat{x}_2\cos\theta$$
$$\hat{y}_3 = \hat{y}_2$$
$$\hat{z}_3 = \hat{z}_2\cos\theta - \hat{x}_2\sin\theta$$

Notice that the direction of the range at a pixel with the spherical coordinates (ρ, θ, ϕ) is now $\hat{x}_3$. Therefore, the cosine of the angle between $\hat{x}_3$ and $\hat{y}$ is equal to H/ρ where H is the height of the scanner above the ground (recall that $\hat{y}$ is perpendicular to the plane defined in the unperturbed coordinate system by $Y = H$). Therefore,

$$\hat{y} \cdot \hat{x}_3 = \frac{H}{\rho}$$

Expanding $\hat{x}_3$ yields

$$\begin{aligned}
\hat{x}_3 &= \hat{z}_2 \sin\theta + \hat{x}_2 \cos\theta \\
&= [y_1 \sin\phi + z_1 \cos\phi] \sin\theta + x_1 \cos\theta \\
&= [(-\hat{x} \sin\delta_z + \hat{y} \cos\delta_z) \sin\phi + \hat{z} \cos\phi] \sin\theta \\
&\qquad + [\hat{x} \cos\delta_z + \hat{y} \sin\delta_z] \cos\theta
\end{aligned}$$

Taking the dot product of the last equation with $\hat{y}$ results in all of the $\hat{x}$ and $\hat{z}$ terms dropping out and leaving,

$$\frac{H}{\rho} = \cos\delta_z \sin\phi \sin\theta + \sin\delta_z \cos\theta$$

By a simple rearrangement of terms this equation becomes equation (24).

APPENDIX B

Calculating Laser Beam Footprints on Planar Surfaces

For any planar surface that is large enough to intersect all of a range scanner's laser beam, the resultant footprint will be an ellipse. The size of the footprint is of interest because it is a good measure of how mixed the resulting pixel will be. [larrowe86] gives the equations for calculating the axes of the ellipse when the surface is a flat plane (i.e. a plane parallel to the XZ axis). The parameters of Larrowe's equations are the height of the plane (i.e. its constant y value), θ, ϕ, and the angular width of the laser beam. The following derivation leads to a generalized set of equations for the axes of the ellipse formed by a surface at any orientation. The equations' parameters are:

$\rho =$ range along the laser beam's central axis (the beam is a cone)

$\gamma =$ angle between the central axis and the planar surface

$\delta =$ half the angular width of the cone

$M =$ major axis of the ellipse

$m =$ minor axis of the ellipse

Figure 22 illustrates the cross-section of the laser beam's cone and the planar surface that cuts through the cone. In the drawing, $M = |\overline{cf}| = |\overline{cd}| + |\overline{df}|$ and $\rho = |\overline{ad}| = |\overline{ab}| + |\overline{bd}|$. From the geometry of the figure we can find the value of $|\overline{cd}|$ in the following manner:

$$\frac{|\overline{bc}|}{|\overline{ab}|} = \tan\delta \quad so \quad |\overline{ab}| = \frac{|\overline{bc}|}{\tan\delta}$$

$$\frac{|\overline{bc}|}{|\overline{bd}|} = \tan\gamma \quad so \quad |\overline{bd}| = \frac{|\overline{bc}|}{\tan\gamma}$$

$$\rho = |\overline{ab}| + |\overline{bd}| = \frac{|\overline{bc}|}{\tan\delta} + \frac{|\overline{bc}|}{\tan\gamma}$$

Therefore,

$$|\overline{bc}| = \frac{\rho\tan\delta\tan\gamma}{\tan\delta + \tan\gamma}$$

The first section of M can now be written as

$$|\overline{cd}| = \frac{|\overline{bc}|}{\sin\gamma} = \frac{\rho\tan\delta\tan\gamma}{\sin\gamma(\tan\delta + \tan\gamma)}$$

$$= \frac{\rho\tan\delta}{\cos\gamma(\tan\delta + \tan\gamma)} = \frac{\rho\sin\delta}{\sin(\gamma+\delta)}$$

The other piece of M, $|\overline{df}|$, can be found in the same way:

$$\frac{|\overline{ef}|}{|\overline{de}|} = \tan\gamma \quad \text{so} \quad |\overline{ef}| = |\overline{de}|\tan\gamma$$

$$\frac{|\overline{ef}|}{|\overline{ae}|} = \tan\delta \quad \text{so} \quad |\overline{ef}| = |\overline{ae}|\tan\delta$$

Combining the two equations for $|\overline{ef}|$ and rearranging terms yields

$$|\overline{ae}| = |\overline{de}|\frac{\tan\gamma}{\tan\delta}$$

Substituting $\rho + |\overline{de}|$ for $|\overline{ae}|$ in the last equation gives

$$|\overline{de}| = \frac{\rho\tan\delta}{\tan\gamma - \tan\delta}$$

This allows us to solve for $|\overline{df}|$:

$$|\overline{df}| = \frac{|\overline{de}|}{\cos\gamma} = \frac{\rho\tan\delta}{\cos\gamma(\tan\gamma - \tan\delta)} = \frac{\rho\sin\delta}{\sin(\gamma-\delta)}$$

Adding together the equations for $|\overline{cd}|$ and $|\overline{df}|$ we get M:

$$M = \rho\sin\delta\left(\frac{1}{\sin(\gamma+\delta)} + \frac{1}{\sin(\gamma-\delta)}\right)$$

As in any ellipse, the major and minor axes are related to the eccentricity (e) by

$$m = M\,[1 - e^2]^{1/2}$$

In the case of a plane cutting through a cone we know that the eccentricity of the resulting ellipse will be $e = \dfrac{\cos\gamma}{\cos\delta}$ (see [thomas72] for details). So,

$$m = M\left[1 - \left(\frac{\cos\gamma}{\cos\delta}\right)^2\right]^{1/2}$$

APPENDIX C

Range Scanner Calibration

The vertical field of view of an ERIM range scanner is fixed at 30 degrees. The minimum ϕ, however, is dependent on how the scanner is mounted on the ALV. At the Martin Marietta test track, $\phi_{\min}$ is often not measured at the time images are produced. Calibrating $\phi_{\min}$ for the range images used in this report was done in the following manner.

The calibration problem has four basic constraints:

1) The scanner's hardware provides very accurate increments of ϕ between rows so if one assumes that ϕ at row i is x then ϕ at row j will be $x + (j-i)$ 30/63.

2) In each collection of range images there is at least one image taken of the test track in which the center of the image contains a flat plane (i.e. $y = H$) with no hills or obstacles.

3) $H = \rho \sin\theta \sin\phi$

4) θ is assumed to be 90 degrees in the center of the image.

If the scanner is properly adjusted then the last assumption is exactly correct. Since the calibration is based on $\sin\theta$, even a 5 degree error in θ will negligibly affect the results.

These constraints reduce the calibration problem to finding the best $(H, \phi_{\min})$ pair that fits the ρ's in the central column of the flat plane range

image. Unfortunately, Martin Marietta also did not measure the height of the scanner so H must be treated as an unknown.

Each combination of H ranging from 8 feet 6 inches to 9 feet 6 inches in one inch increments and $\phi_{\min}$ ranging from 0 degrees to 25 degrees in .25 degree increments is used to generate expected ρ's for pixels along the central column of the flat plane range image. At each pixel the expected ρ is subtracted from the actual ρ and then the difference is squared. The sum of the squares is used as a measure of how well the $(H, \phi_{\min})$ pair fits the scanner's actual height and $\phi_{\min}$.

The best fit for the images used in this report was (9 feet, 6 degrees).

REFERENCES

T. B. Boult and J. R. Kender, "On Surface Reconstruction Using Sparse Depth Data", *Proceedings: Image Understanding Workshop*, Miami Beach, Florida, December 1985, 197–208.

R. A. Brooks, "A Robust Layered Control System for a Mobile Robot", Artificial Intelligence Laboratory A. I. Memo 864, Massachusetts Institute of Technology, Cambridge, Massachusetts, September 1985.

B. Carrihill and R. Hummel, "Experiments with the Intensity Ratio Depth Sensor", *Computer Vision, Graphics, and Image Processing 32* (1985), 337–358.

D. J. Choi and J. R. Kender, "Solving the Depth Interpolation Problem with the Adaptive Chebyshev Acceleration Method on a Parallel Computer", *Proceedings: Image Understanding Workshop*, Miami Beach, Florida, December 1985, 219–223.

D. Dementhon, "Production of Smooth Range Images from a Plane-of-Light Scanner", Center for Automation Research Technical Report, University of Maryland, College Park, Maryland, 1987 (To be published.).

R. O. Duda, D. Nitzan and P. Barrett, "Use of Range and Reflectance Data to Find Planar Surface Regions", *IEEE Transactions on PAMI 1*, 3 (1979), 259–271.

R. N. Haber and M. Hershenson, *The Psychology of Visual Perception*, Holt, Rinehart and Winston, Inc., 1980.

E. L. Hall, J. B. K. Tio, C. A. McPherson and F. A. Sadjadi, "Measuring curved surfaces for robot vision", *Computer 15*, 12 (1982), 42–54.

D. Harwood, M. Subbarao, H. Hakalahti and L. S. Davis, "A New Class of Edge-Preserving Smoothing Filters", Center for Automation Research Technical Report 59, University of Maryland, College Park, Maryland, May 1984.

M. Hebert and J. Ponce, "A New Method for Segmenting 3-D Scenes Into Primitives", *Proceedings: 6th International Conference on Pattern Recognition*, Munich, Germany, October 1982, 836–838.

M. Hebert and T. Kanade, "First Results on Outdoor Scene Analysis Using Range Data", *Proceedings: Image Understanding Workshop*, Miami Beach, Florida, December 1985, 224–231.

R. Hoffman and A. K. Jain, "Segmentation and Classification of Range Images", *Proceedings: Computer Vision and Pattern Recognition*, Miami Beach, Florida, December 1986, 424–426.

R. Hoffman and A. K. Jain, "Segmentation and Classification of Range Images", Technical Report MSU-ENGR-86-002, Michigan State University, East Lansing, Michigan, 1986.

S. Inokuchi, T. Nita, F. Matsuda and Y. Sakurai, "A Three Dimensional Edge-Region Operator for Range Pictures", *Proceedings: 6th International Conference on Pattern Recognition*, Munich, Germany, October 1982, 918–920.

A. K. Jain and R. Hoffman, "Evidence Based Recognition of 3-D Objects", Technical Report MSU-ENGR-86-013, Michigan State University, East Lansing, Michigan, 1986.

R. A. Jarvis, "A Laser Time-of-Flight Range Scanner for Robotic Vision", *IEEE Transactions on PAMI 5*, 5 (1983), 505–512.

R. A. Jarvis, "A Perspective on Range Finding Techniques for Computer Vision", *IEEE Transactions on PAMI 5*, 2 (1983), 122–139.

S. Kambhampati and L. S. Davis, "Multiresolution Path Planning for Mobile Robots", *IEEE Journal of Robotics and Automation RA-2* (1986), 135–145.

V. Larrowe, "Operating Principles of Laser Ranging, Image Producing (3D) Sensors", *Environmental Research Institute of Michigan*, Ann Arbor, Michigan, March, 1986. (Work to be published).

J. Le Moigne and A. M. Waxman, "Projected Grids for Short Range Navigation of Autonomous Robots", *Proceedings: 7th International Conference on Pattern Recognition*, Montreal, Canada, July 1984, 203–206.

X. Y. Lin and W. G. Wee, "SDFS: A New Strategy for the Recognition of Objects Using Range Data", *Proceedings: IEEE International Conference on Robotics and Automation*, San Francisco, California, April 1986, 770–775.

D. Nitzan, A. E. Brain and R. O. Duda, "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis", *Proc. IEEE 65* (1977), 206–220.

J. L. Olivier and F. Ozguner, "A Navigation Algorithm for an Intelligent Vehicle with a Laser Range Finder", *Proceedings: IEEE International Conference on Robotics and Automation*, San Francisco, California, April 1986, 1145–1150.

A. M. Parodi, J. J. Nitao and L. S. McTamaney, "An Intelligent System for an Autonomous Vehicle", *Proceedings: IEEE International Conference on Robotics and Automation*, San Francisco, California, April 1986, 1657–1663.

B. Parvin and G. Medioni, "Segmentation of Range Images into Planar Surfaces by Split and Merge", *Proceedings: Computer Vision and Pattern Recognition*, Miami Beach, Florida, 1986, 415–417.

D. W. Payton, "An Architecture for Reflexive Autonomous Vehicle Control", *Proceedings: IEEE International Conference on Robotics and Automation*, San Francisco, California, April 1986, 1838–1845.

S. Puri and L. S. Davis, "Two Dimensional Path Planning with Obstacles and Shadows", Center for Automation Research Technical Report 255, University of Maryland, College Park, Maryland, January 1987.

R. R. Rogers, *A Short Course in Cloud Physics*, Pergamon Press, New York, 1979.

J. T. Schwartz, "Structured Light Sensors for 3D Robot Vision", Courant Institute Robotics Research Report No. 8, New York University, New York, New York, 1983.

I. K. Sethi and S. N. Jayaramamurthy, "Surface Classification Using Characteristic Contours", *Proceedings: 7th International Conference on Pattern Recognition*, Montreal, Canada, July 1984, 438–440.

U. K. Sharma and L. S. Davis, "Road Following by an Autonomous Vehicle Using Range Data", Center for Automation Research Technical Report 194, University of Maryland, College Park, Maryland, March 1986.

S. S. Sinha and A. M. Waxman, "An Image Flow Simulator", Center for Automation Research Technical Report 71, University of Maryland, College Park, Maryland, July 1984.

G. B. Thomas, Jr., *Calculus and Analytic Geometry*, Addison-Wesley, Reading, Massachusetts, 1972.

B. C. Vemuri and J. K. Aggarwal, "3-Dimensional Reconstruction of Objects from Range Data", *Proceedings: 7th International Conference on Pattern Recognition*, Montreal, Canada, July 1984, 752–754.

A. H. Waxman, J. LeMoigne, L. S. Davis, E. Liang and T. Siddalingaiah, "A Visual Navigation System for Autonomous Land Vehicles", Center for Automation Research Technical Report 139, University of Maryland, College Park, Maryland, July 1985.

H. S. Yang, K. L. Boyer and A. C. Kak, "Range Data Extraction and Interpretation by Structured Light", *Proceedings: Conference on Artificial Intelligence Applications*, Denver, Colorado, December 1984, 199–205.

D. M. Zuk and M. L. Dell'Eva, "Three-Dimensional Vison System for the Adaptive Suspension Vehicle", ERIM Report 170400-3-f, Ann Arbor, Michigan, January 1983.

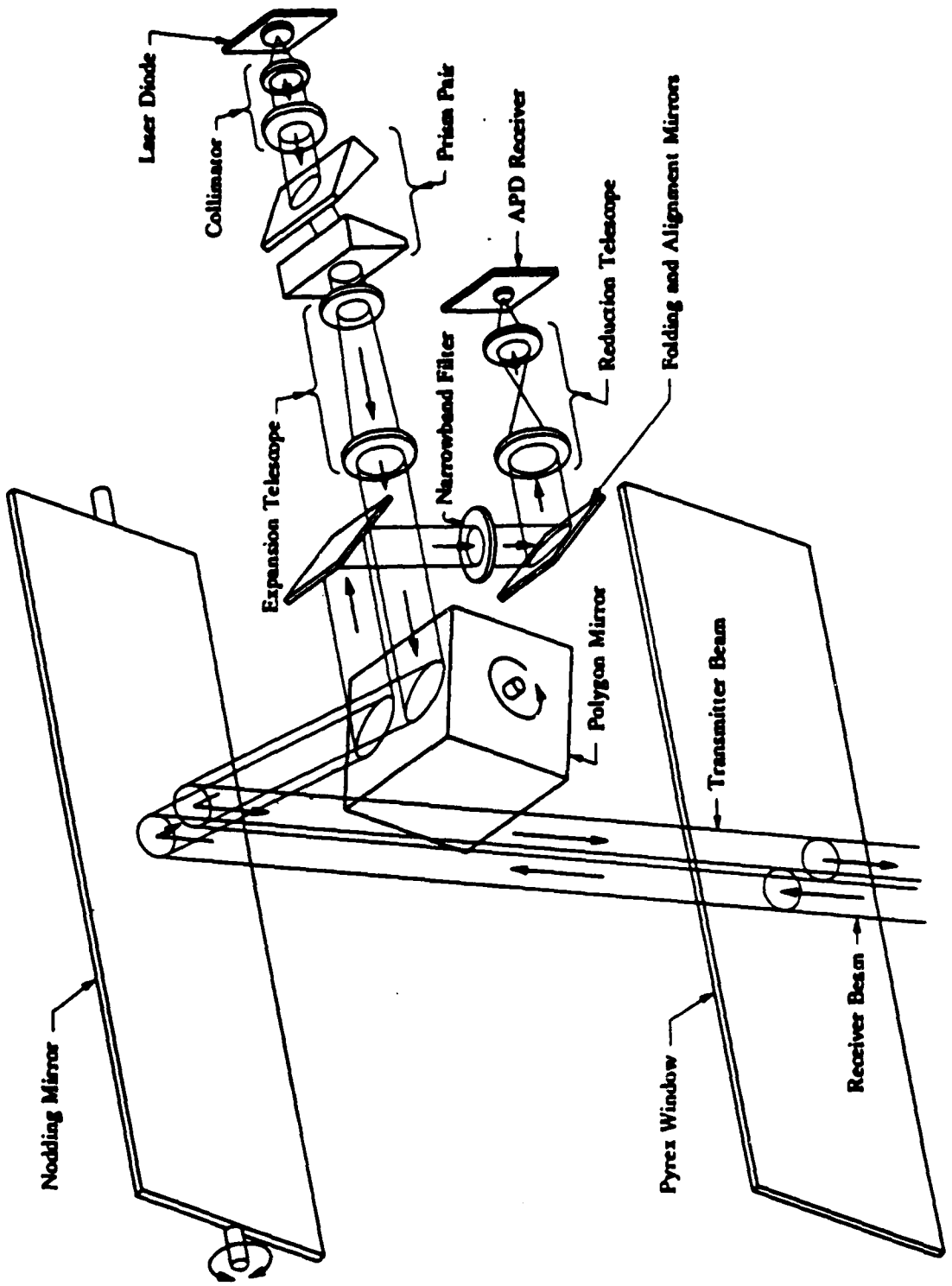
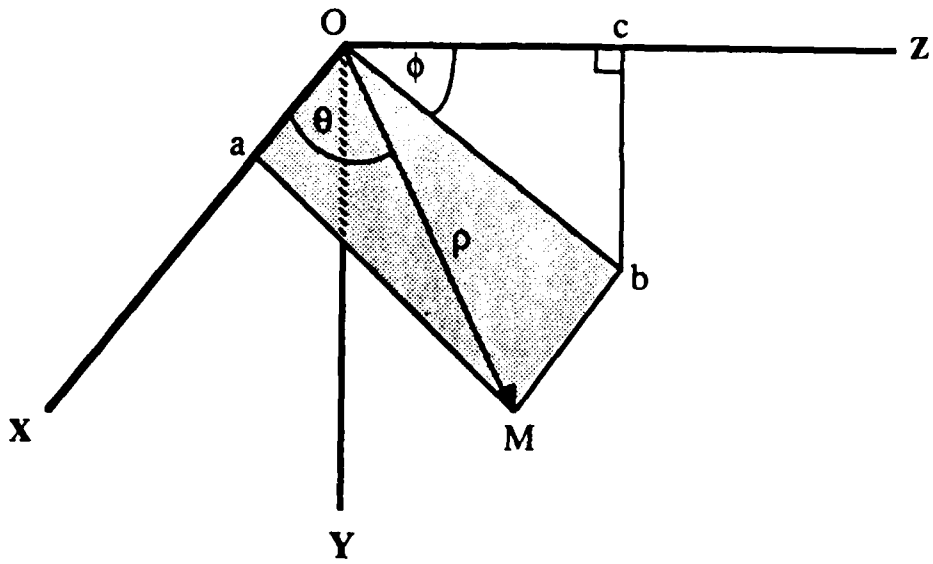


Figure 1: The ERIM Range Scanner (from [larrowe86])



ρ = range

ϕ = vertical scan angle

θ = horizontal scan angle

Figure 2: Range Image Coordinate System

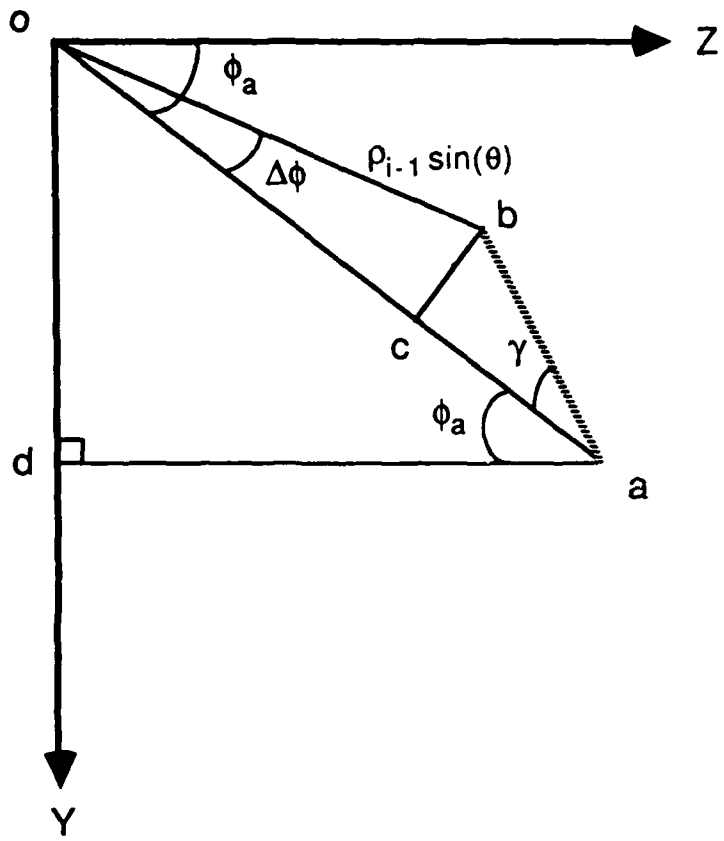


Figure 3: The Geometry of the Range Derivative Algorithm

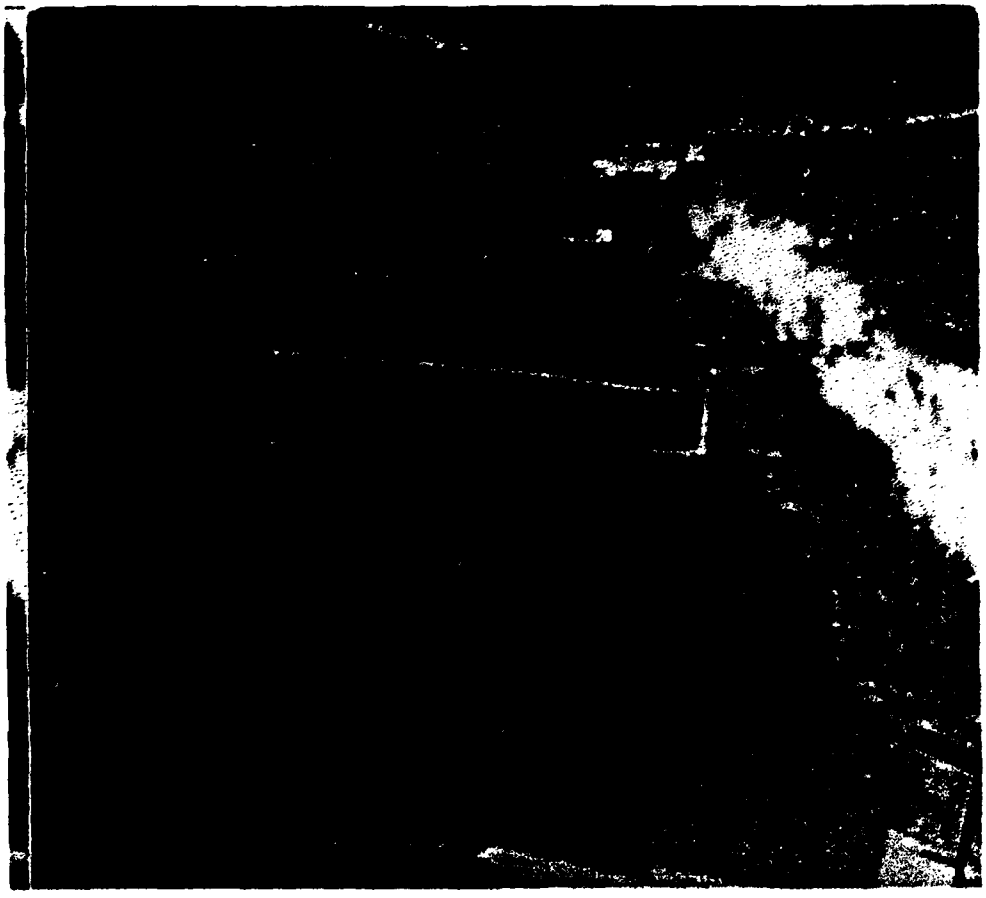


Figure 4: Video Image of Road Segment with Carton

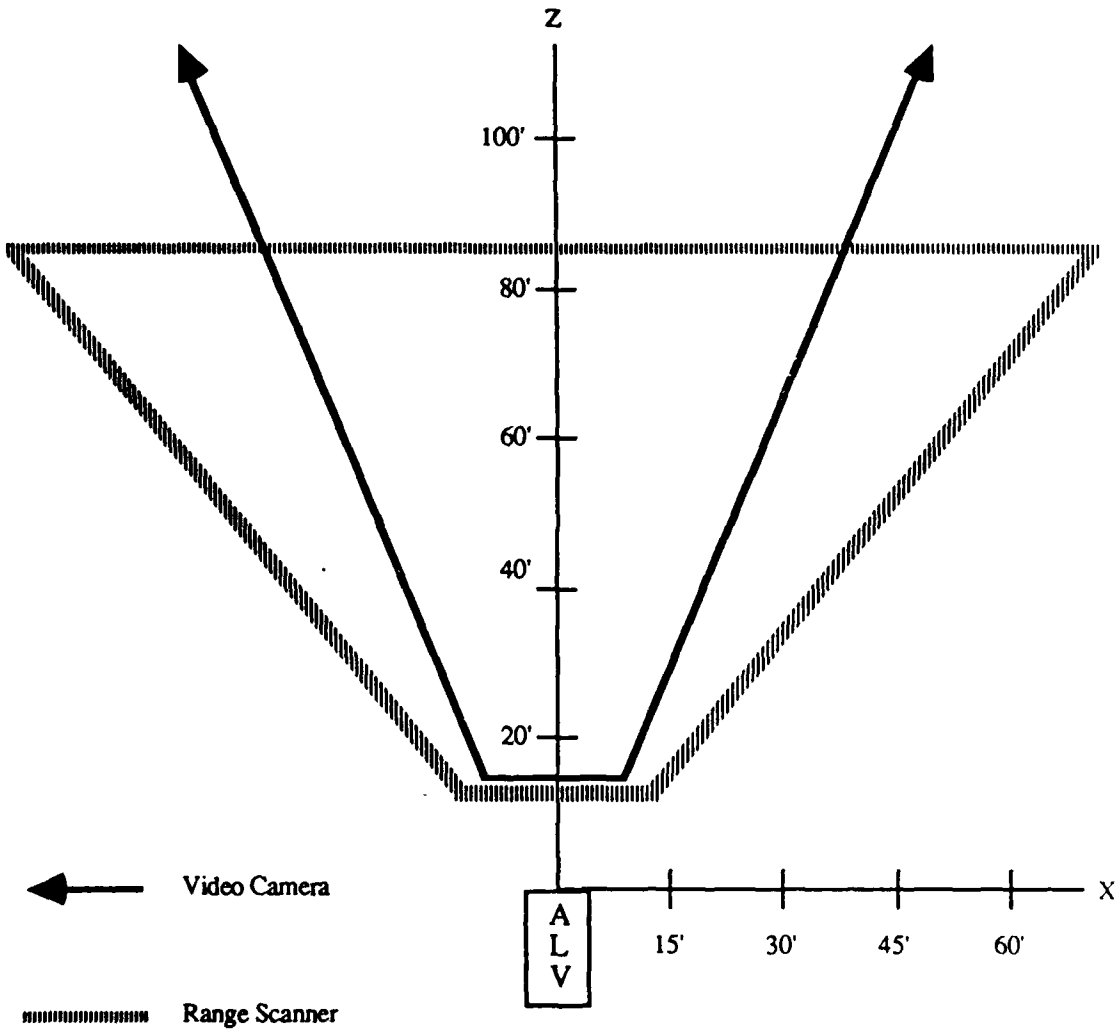


Figure 5: Field of View from the ALV of a Flat Road - Range Scanner vs. Video Camera

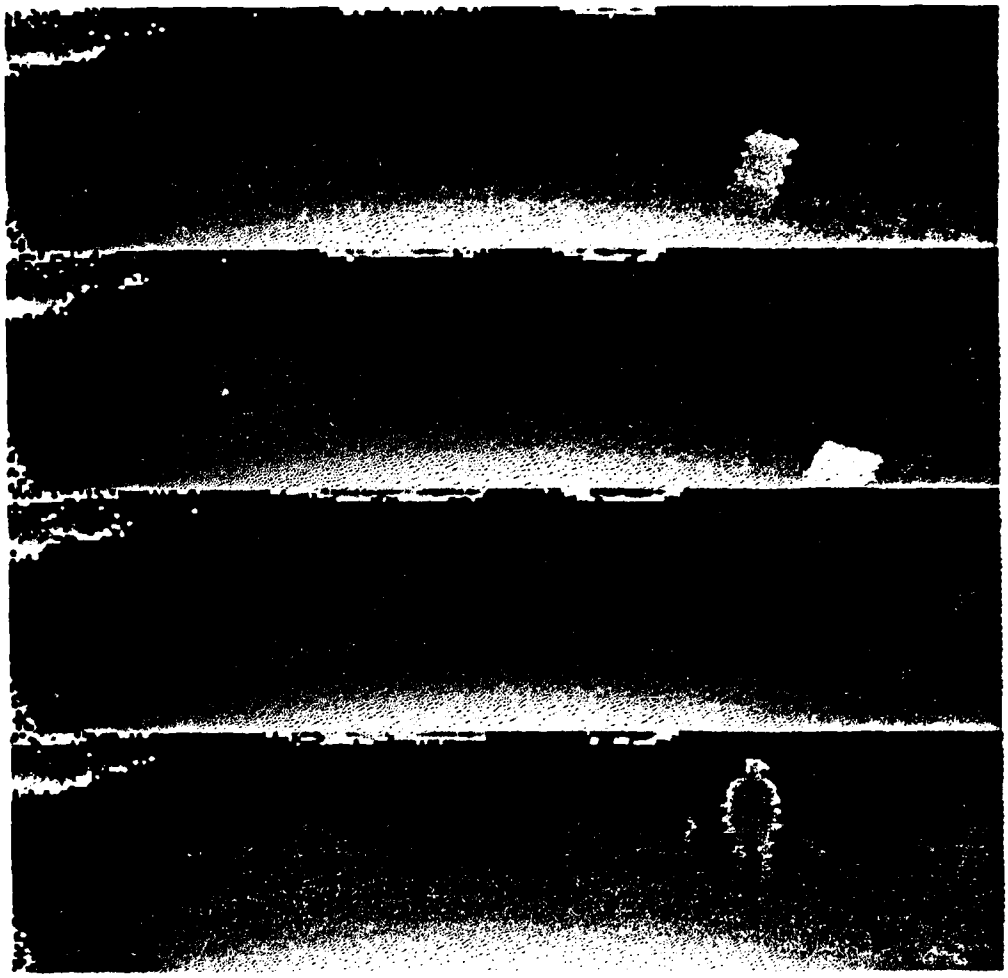


Figure 6a: First Montage of Range Images

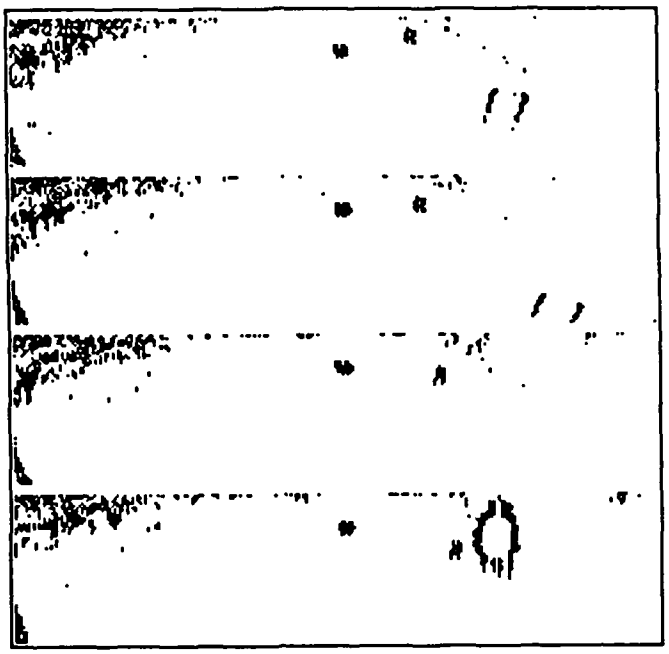


Figure 6b: Thresholded θ Derivatives of First Montage

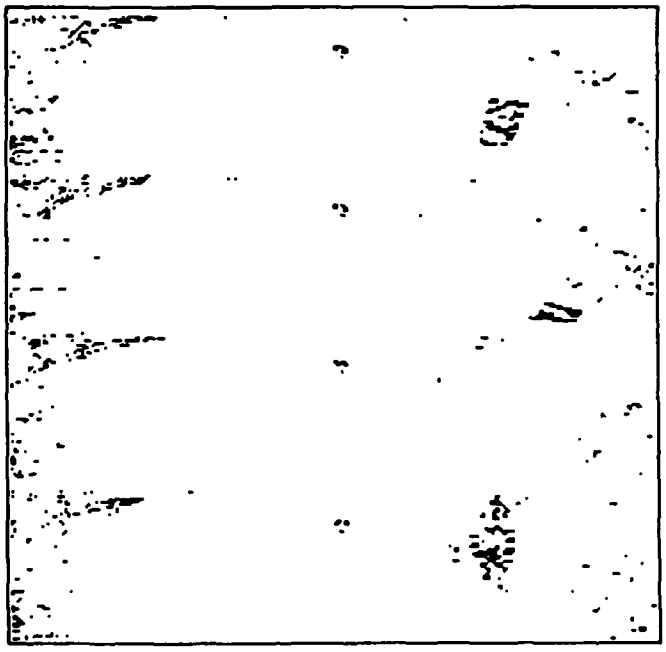


Figure 6c: Thresholded ϕ Derivatives of First Montage

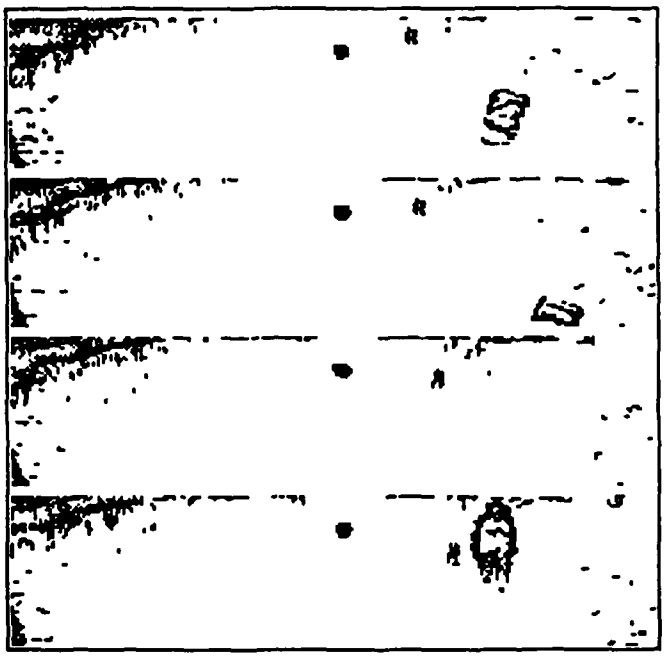


Figure 6d: Combined θ and ϕ Derivatives of First Montage

Figure 7: Video Image of Road Segment with Cone

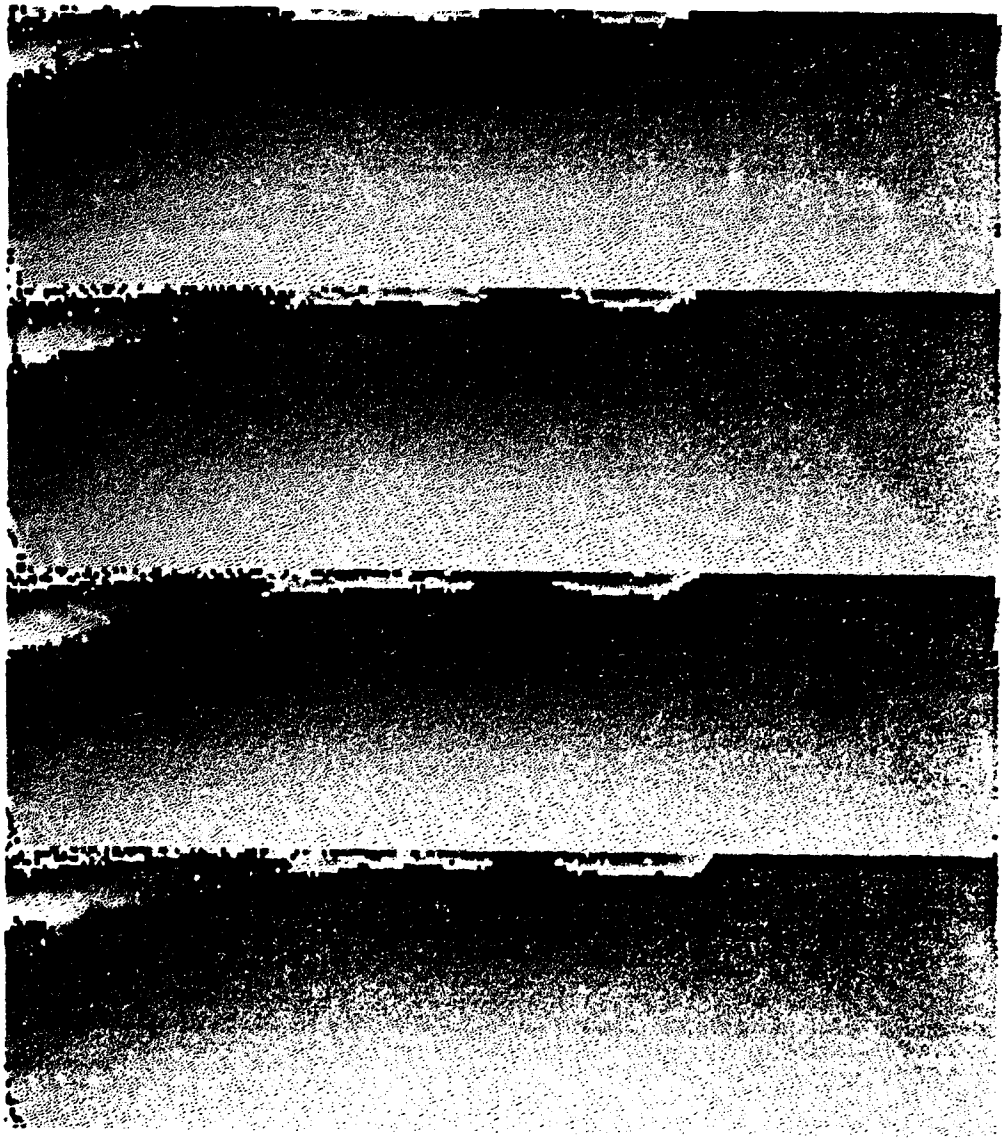


Figure 8a: Second Montage of Range Images

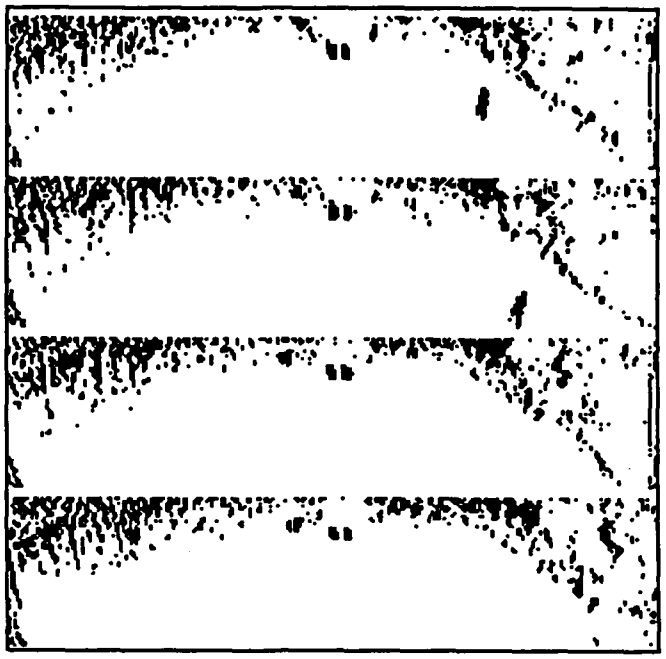


Figure 8b: Thresholded θ Derivatives of Second Montage

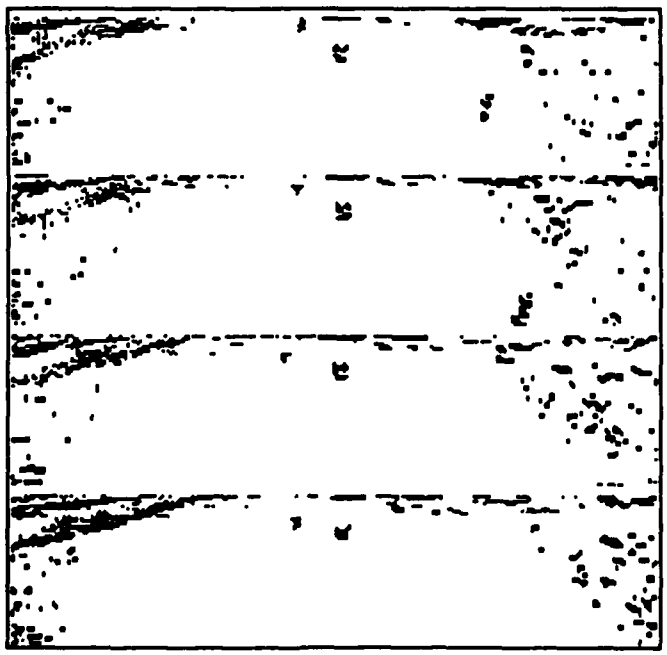


Figure 8c: Thresholded ϕ Derivatives of Second Montage

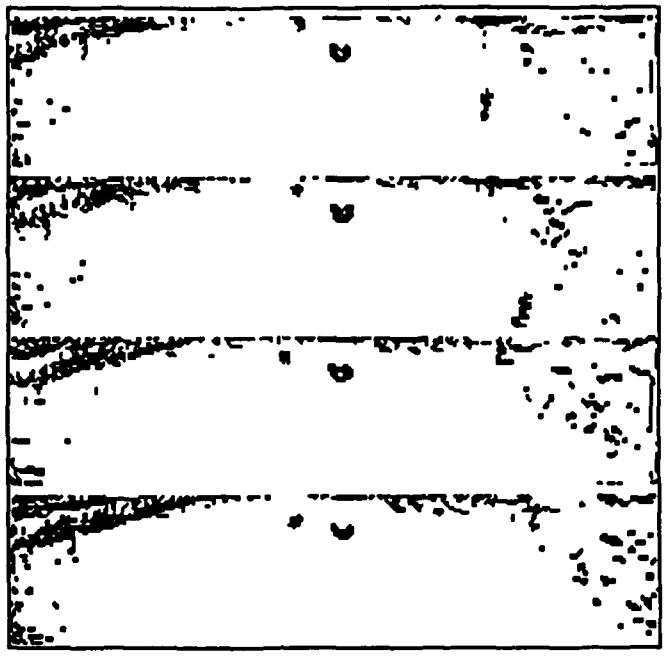


Figure 8d: Combined θ and ϕ Derivatives of Second Montage

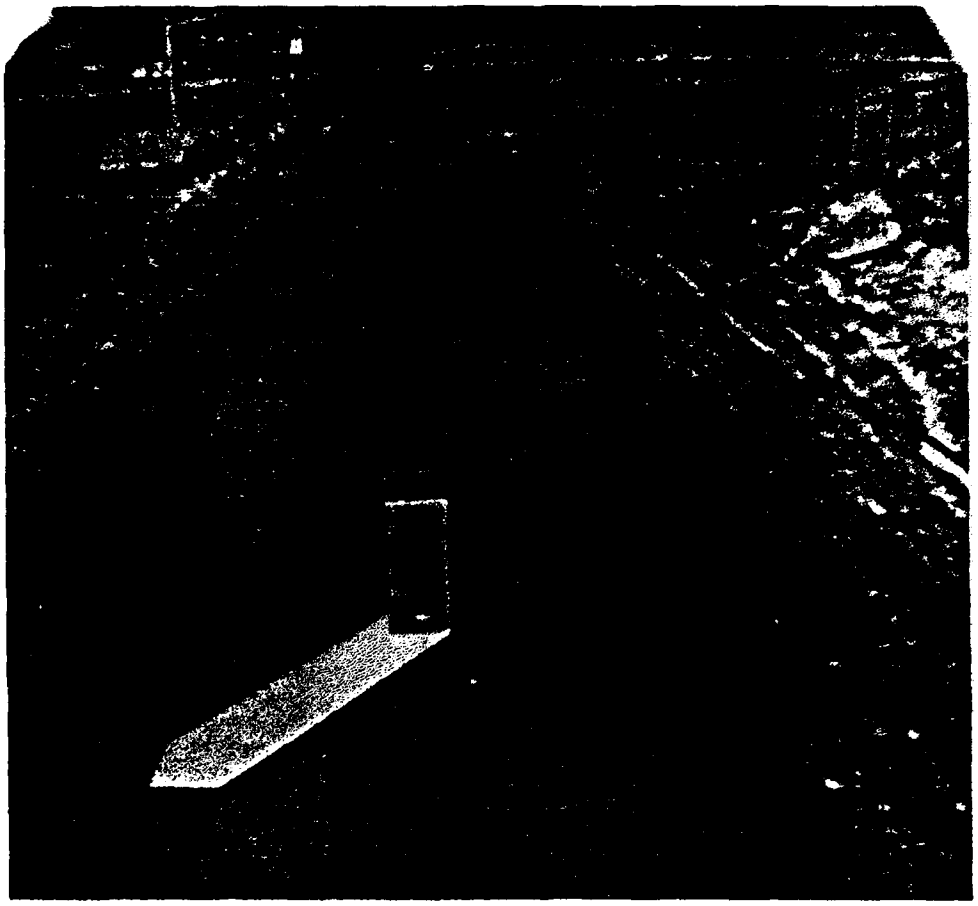


Figure 9: Video Image of Road Segment with Box

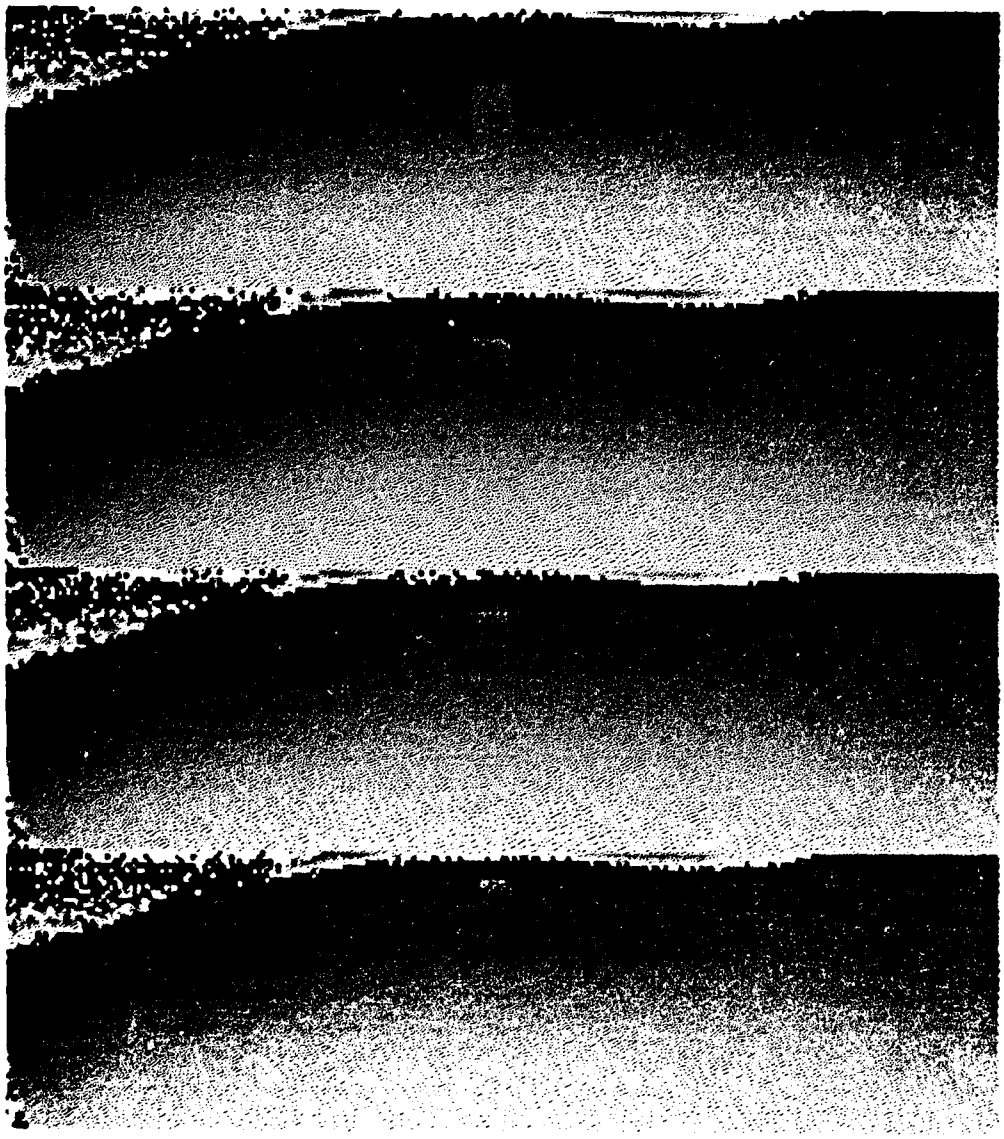


Figure 10a: Montage of Box Segment Range Images

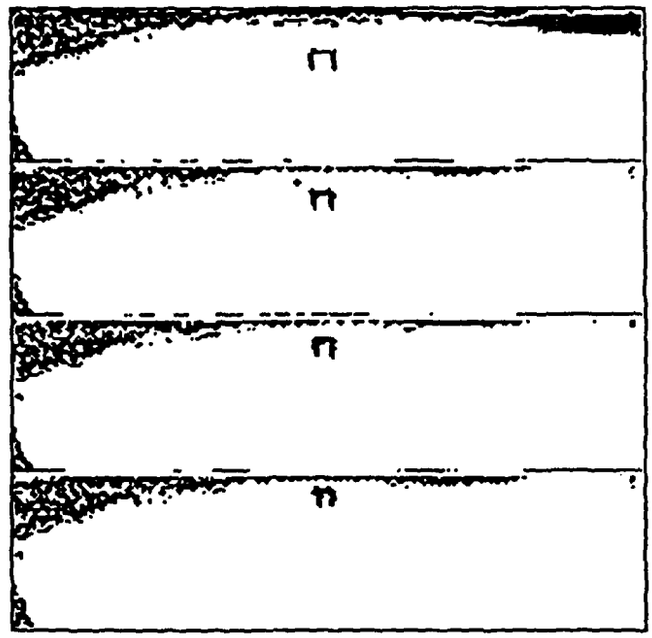


Figure 10b: Obstacle Pixels of Box Segment Montage
With Mixed Pixel Minimization

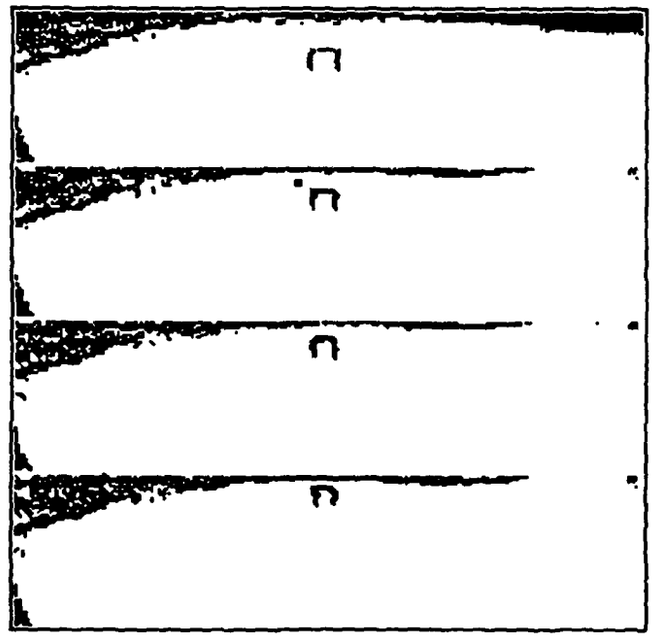


Figure 10c: Obstacle Pixels of Box Segment Montage
Without Mixed Pixel Minimization

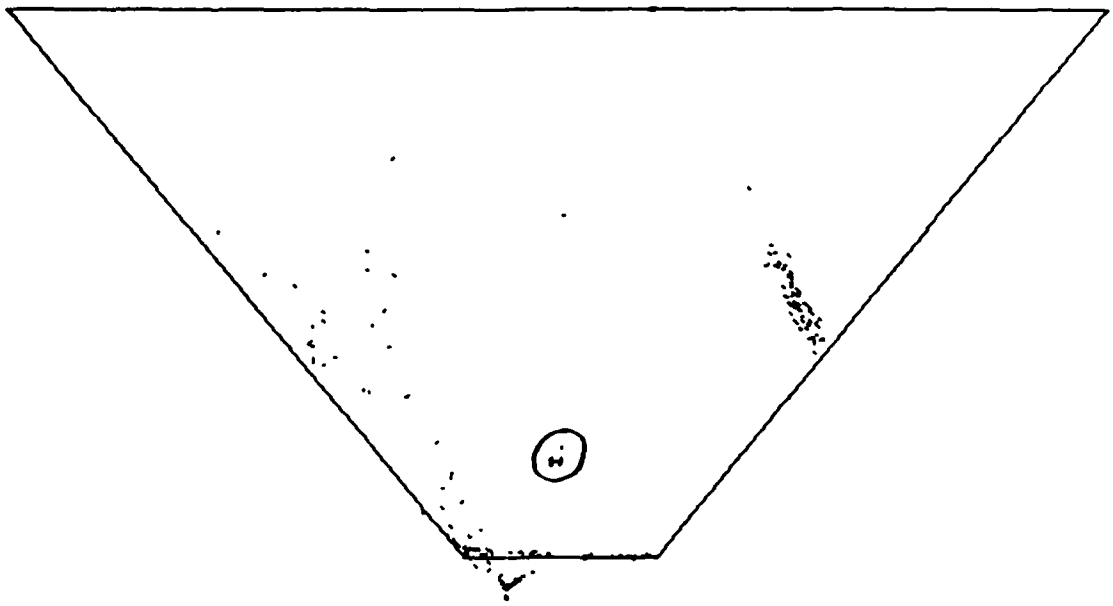


Figure 11a: Projection into Ground Plane of Obstacle Pixels
With Mixed Pixel Minimization: Box at 25 feet

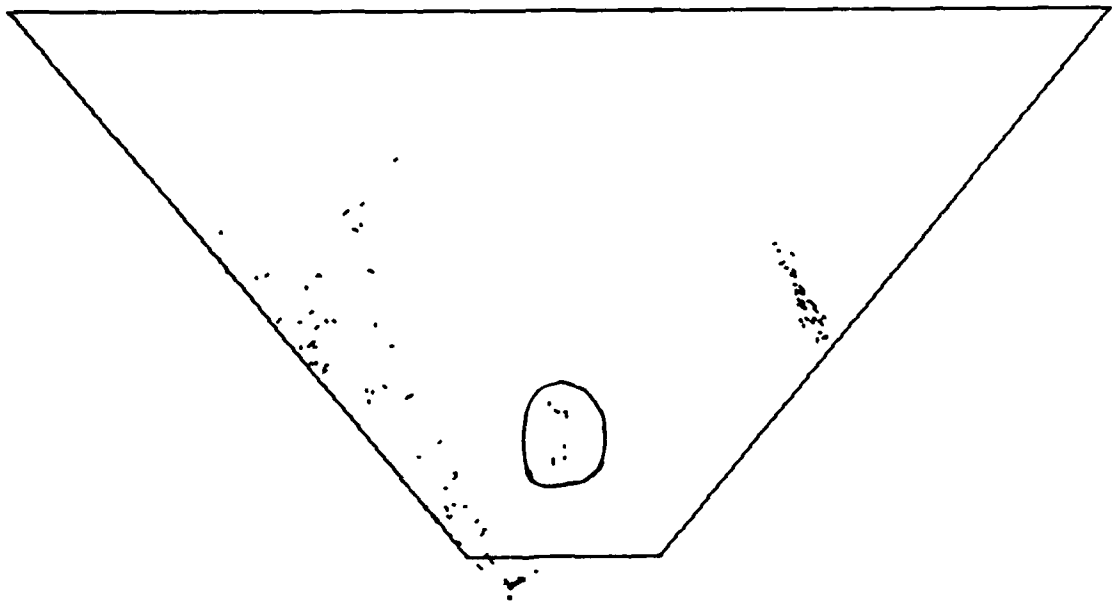


Figure 11b: Projection into Ground Plane of Obstacle Pixels
Without Mixed Pixel Minimization: Box at 25 feet

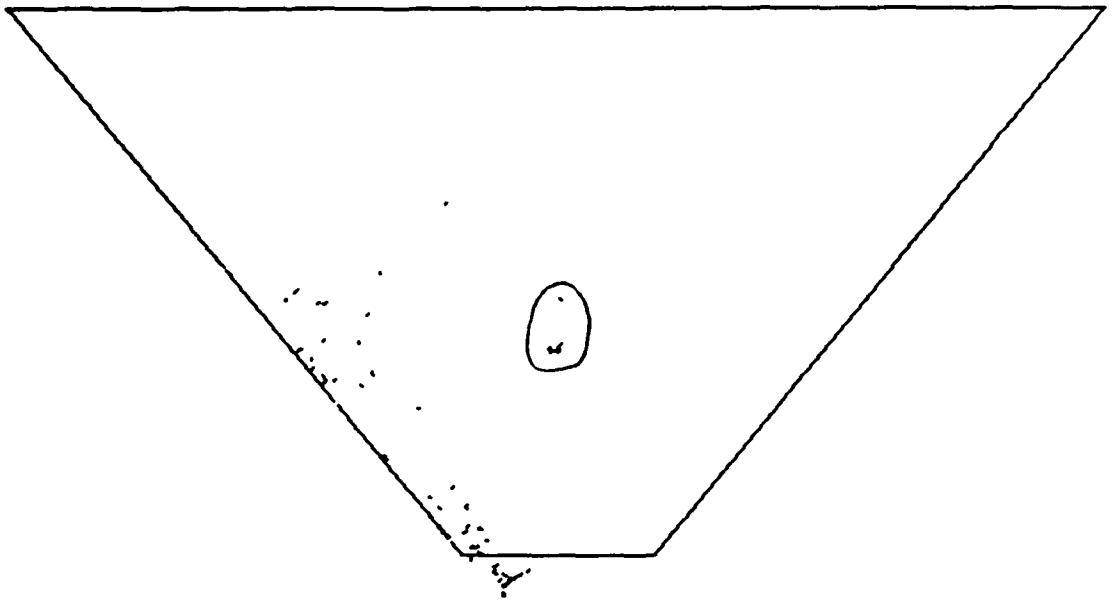


Figure 12a: Projection into Ground Plane of Obstacle Pixels
With Mixed Pixel Minimization: Box at 40 feet

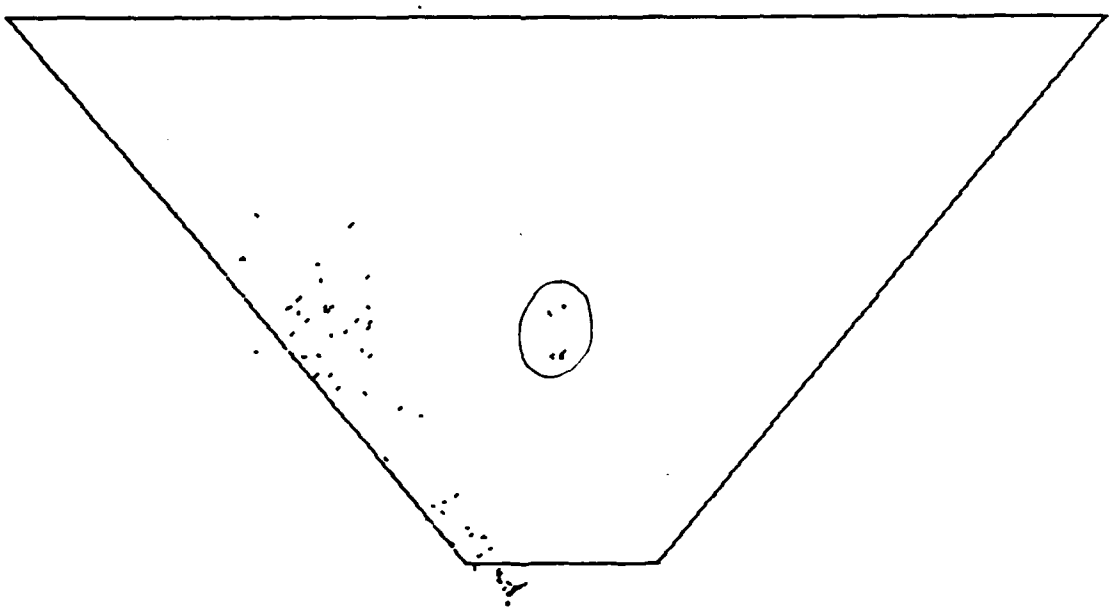


Figure 12b: Projection into Ground Plane of Obstacle Pixels
Without Mixed Pixel Minimization: Box at 40 feet

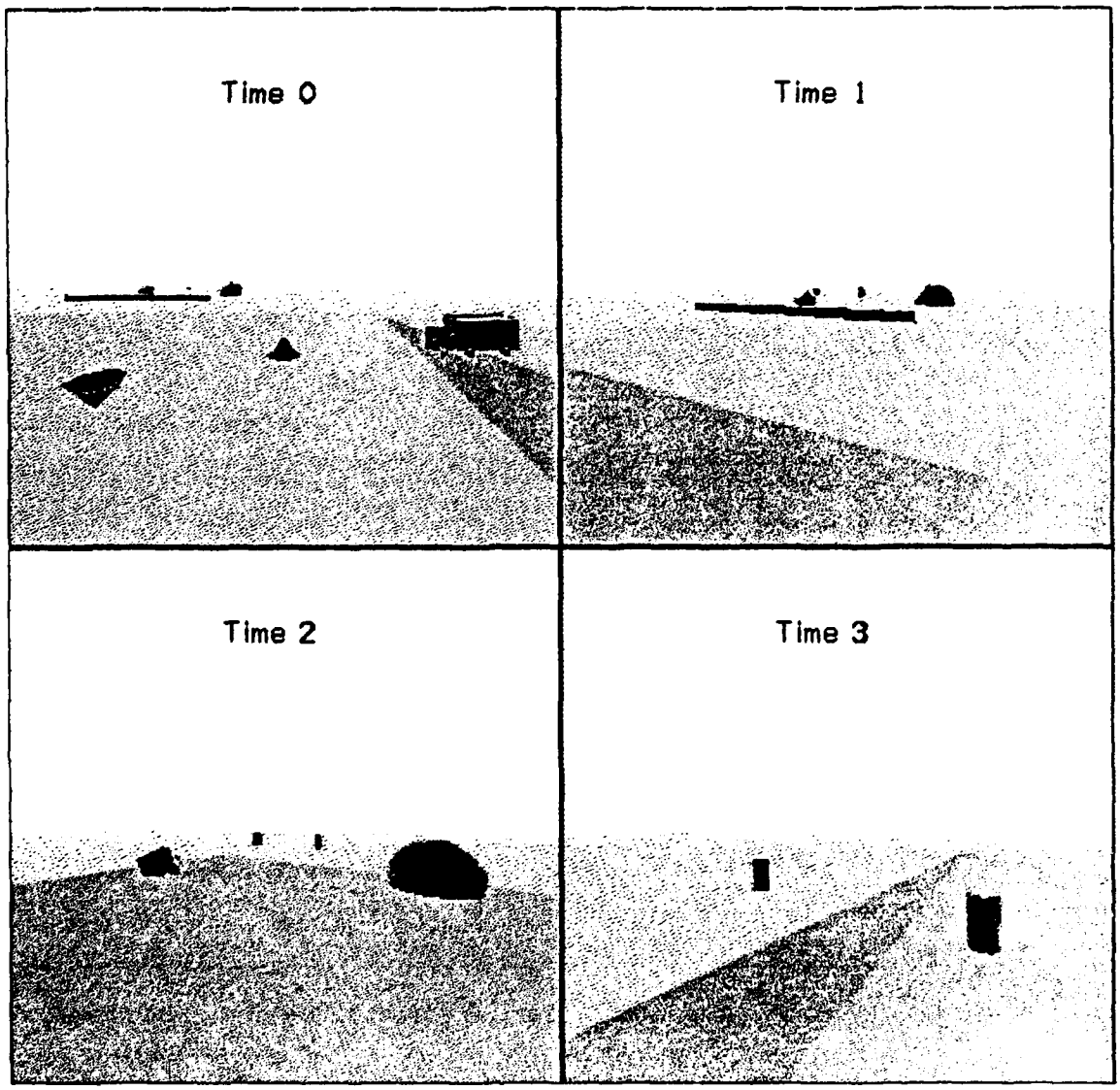


Figure 13: Visual Images from ALV Simulator

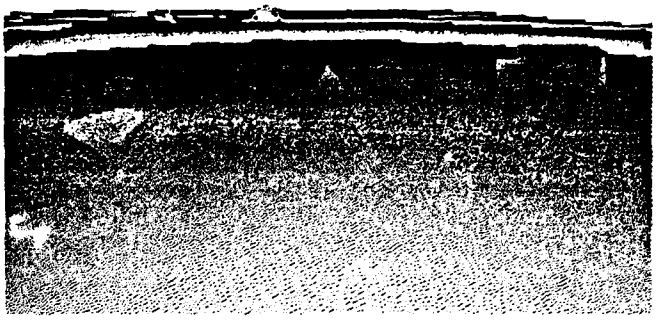


Figure 14: Equirectangular Range Image from Time 0 of Simulator

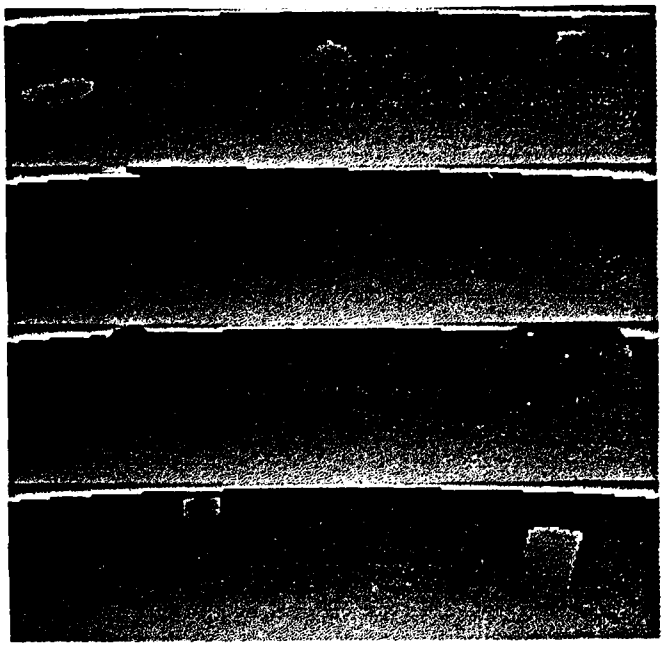


Figure 15: Montage of Equiangular Range Images from ALV Simulator

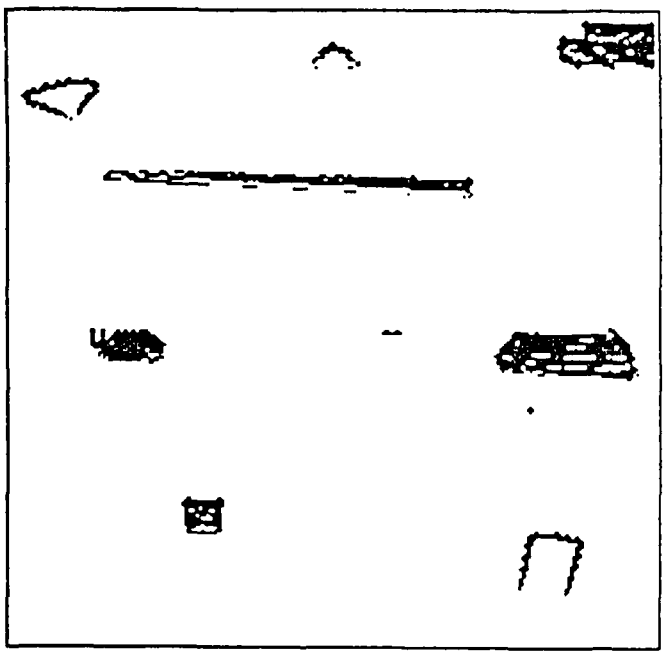


Figure 16: Montage of Thresholded Obstacles

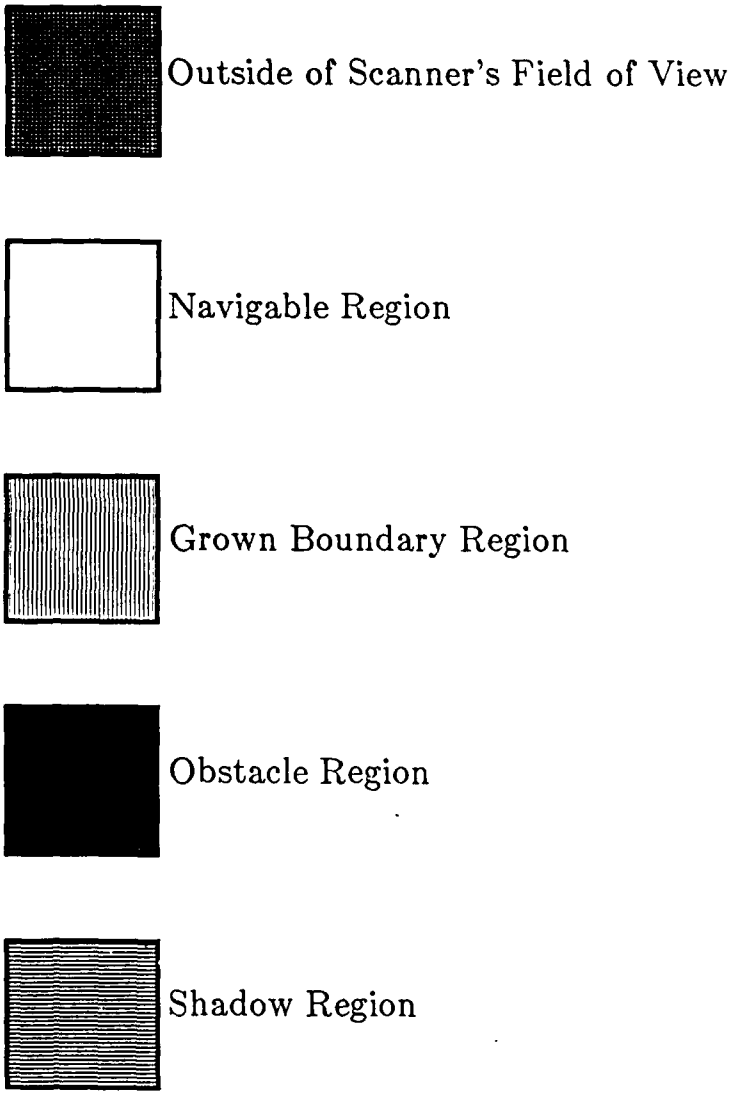


Figure 17: Key for Ground Plane Maps

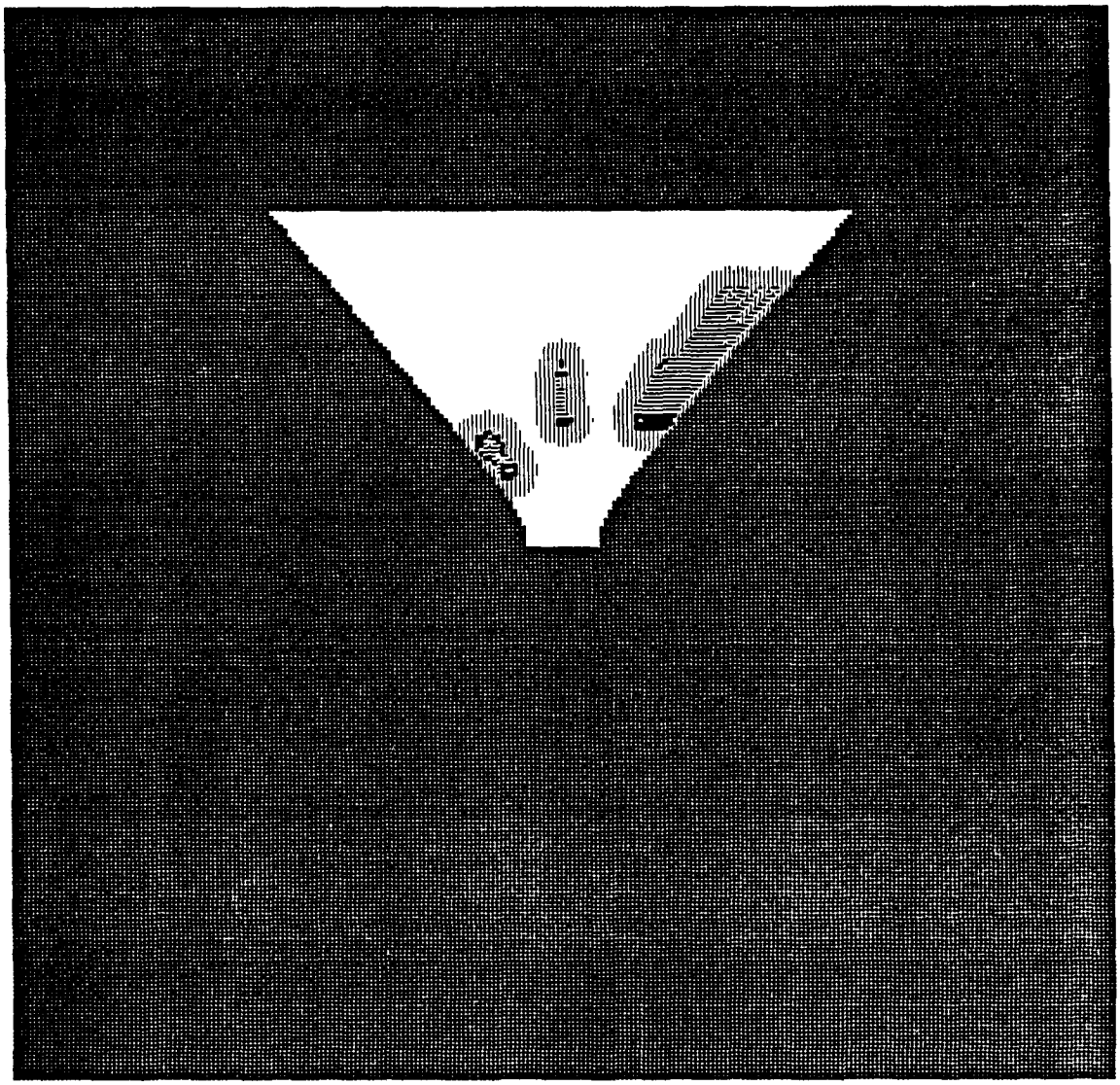


Figure 18: Ground Plane Map from Time 0

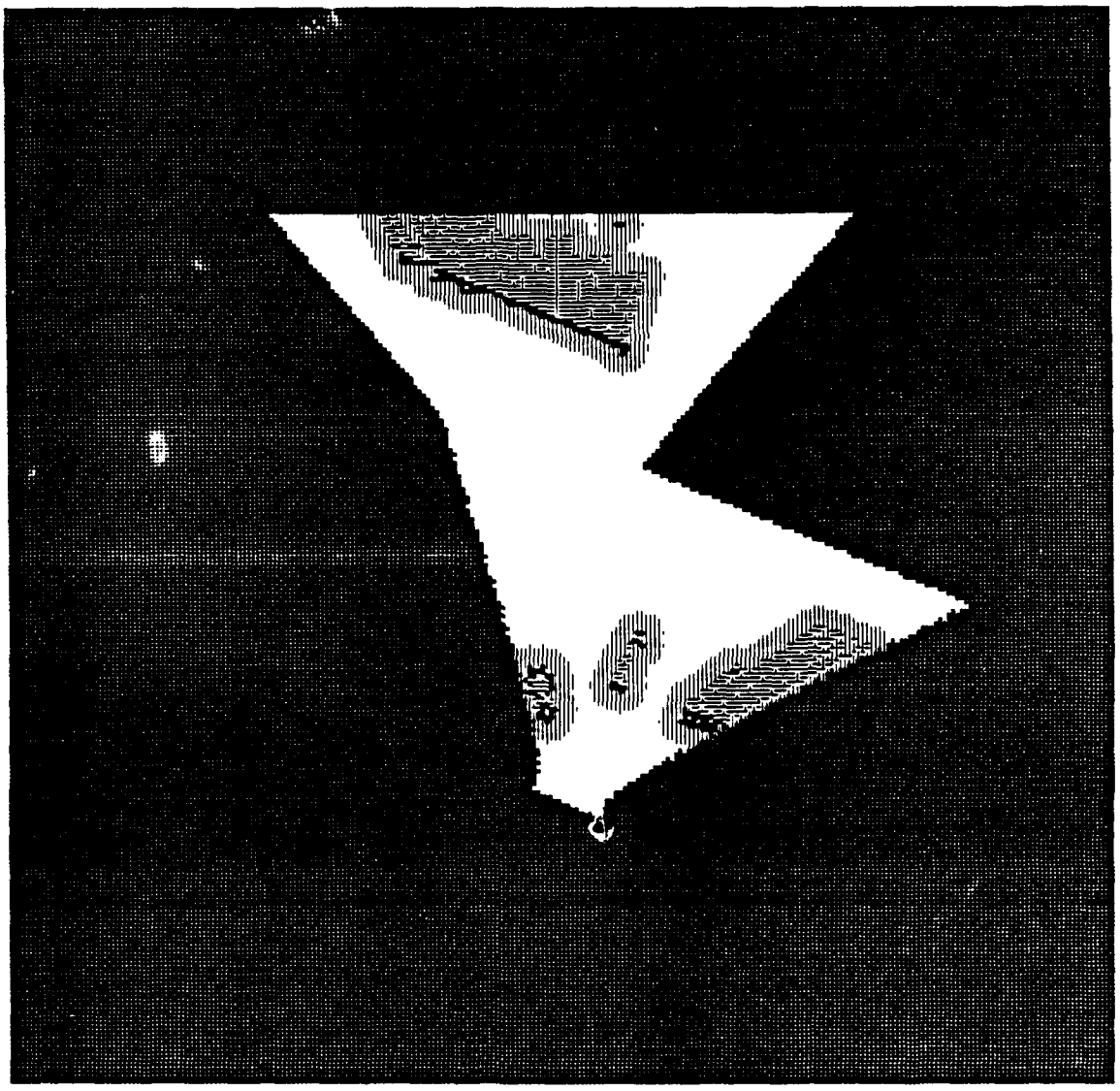


Figure 19: Ground Plane Map from Time 1

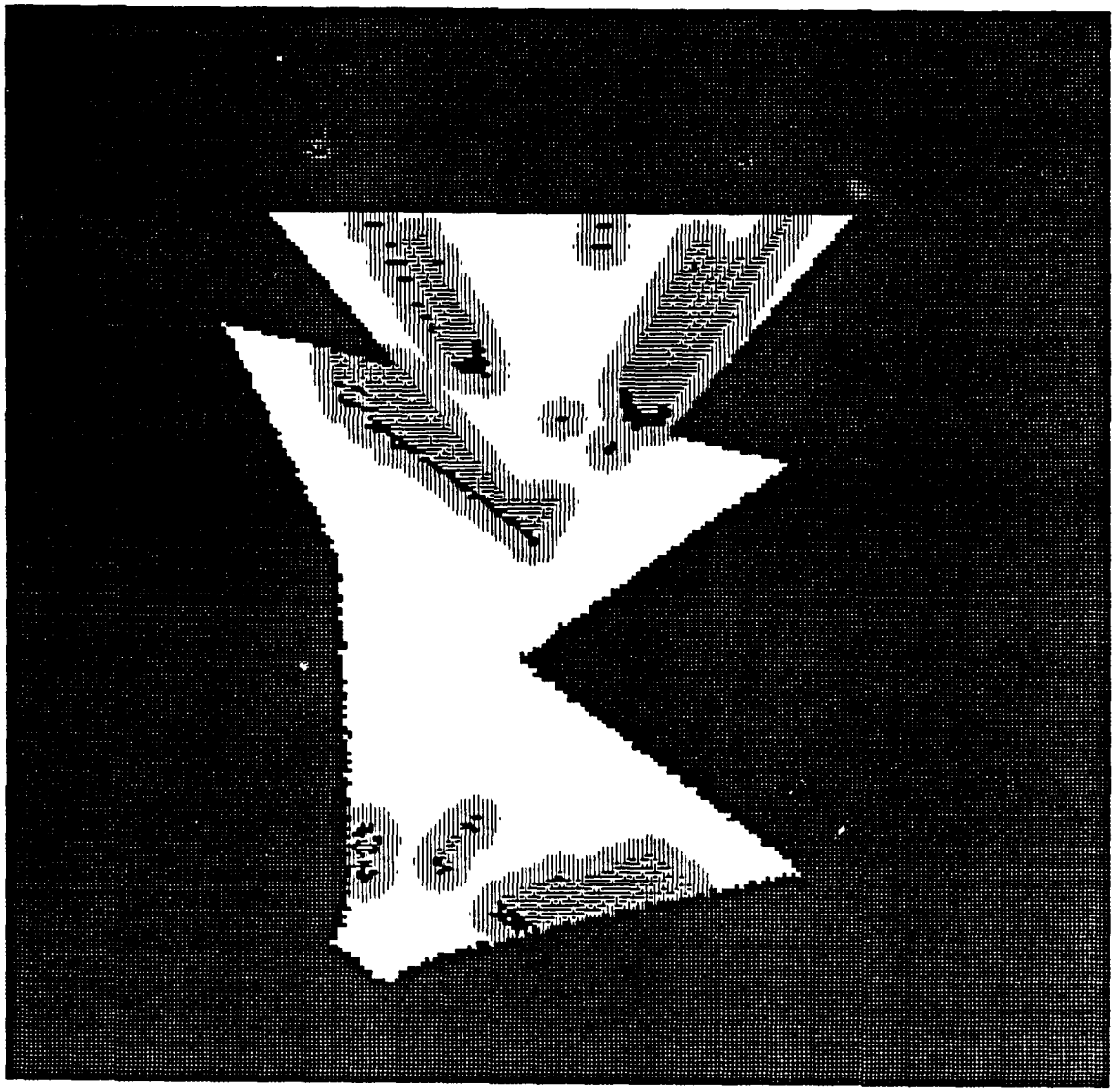


Figure 20: Ground Plane Map from Time 2

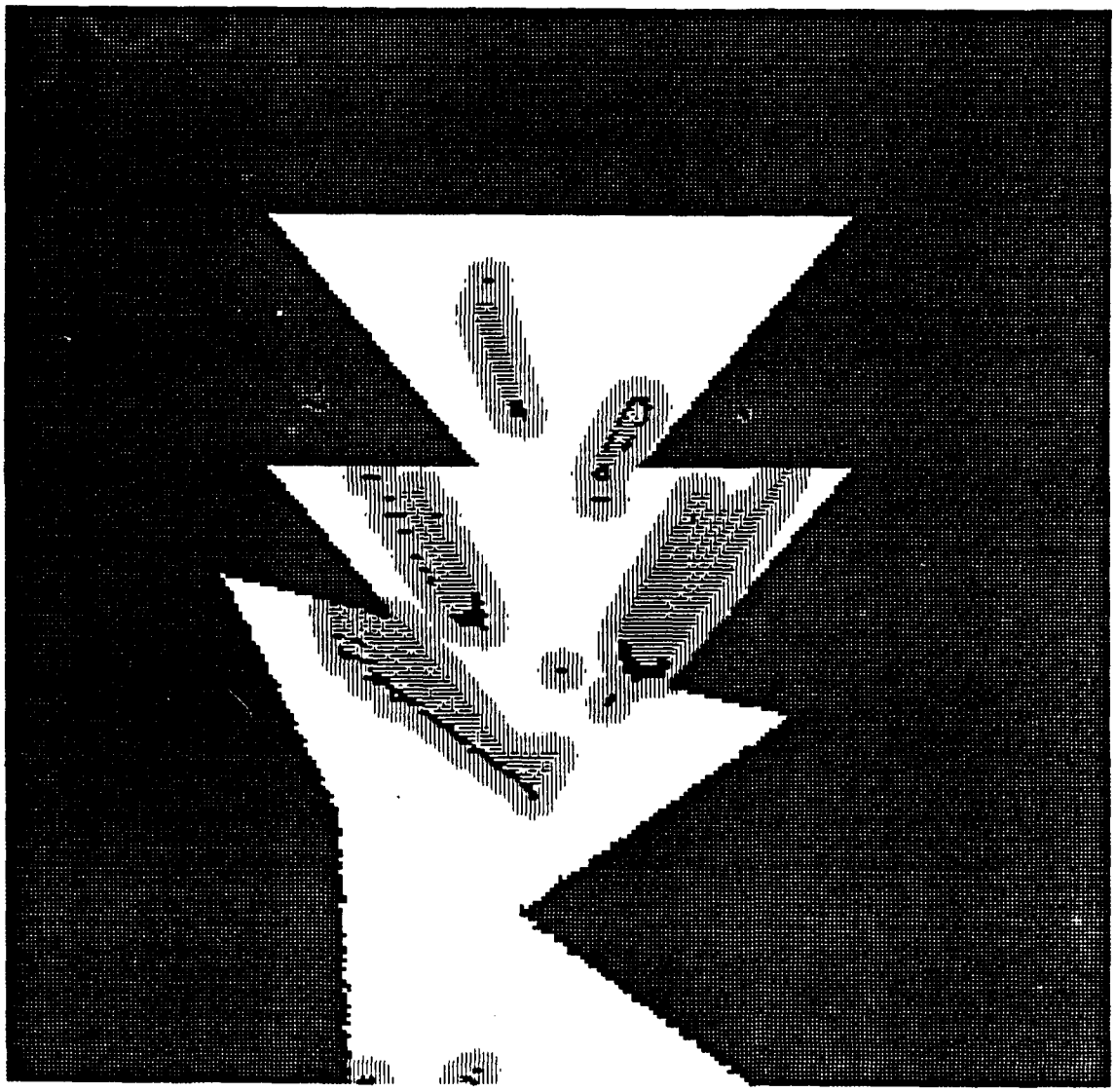


Figure 21: Ground Plane Map from Time 3

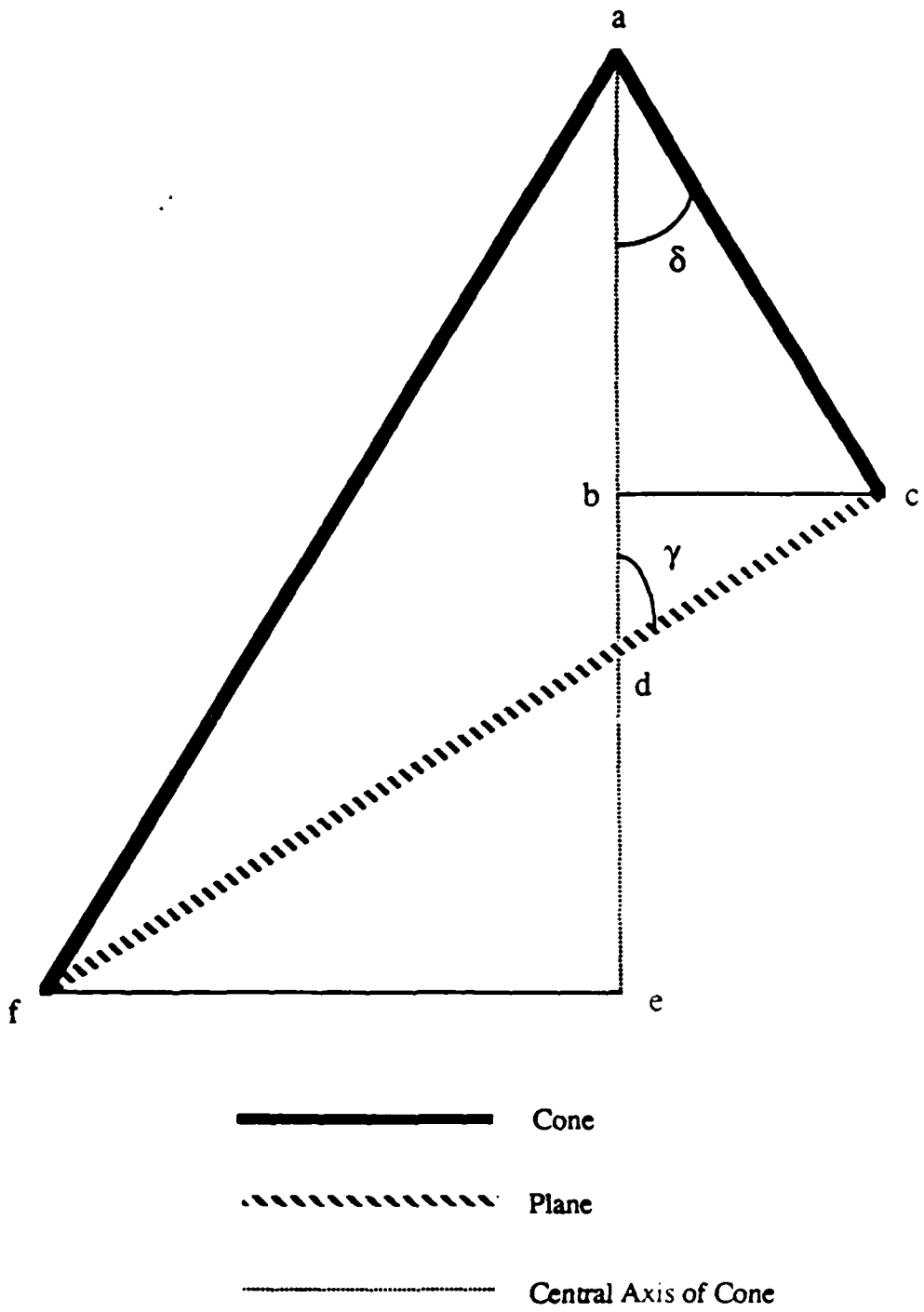


Figure 22: Cross Section of a Laser Beam Intersected by a Plane

Table 1: Comparison of Obstacle Detection Algorithms for Sensitivity to Scanner Perturbations

Perturbation	Magnitude of Errors (1)			
	Theta Derivative Algorithm	Phi Derivative Algorithm	Height Difference Algorithm	Range Difference Algorithm
3 degrees in horizontal angle	0.8 (2) 0.3 (3)	1.5 0.4	5.1 1.6	24.7 6.4
3 degrees in roll angle	3.7 1.1	13.6 2.8	21.2 6.0	103.3 23.1
3 degrees in vertical angle	1.1 0.3	17.3 13.8	25.4 25.4	123.7 98.5
3 degrees in each angle	9.7 2.6	53.5 21.3	72.2 38.9	351.4 150.7

(1) The absolute values of the expected $\Delta\rho$ (or ρ or y) from an unperturbed scanner minus the $\Delta\rho$ (or ρ or y) from a scanner that has been rotated in the manner listed in the 'Perturbation' column.

(2) Maximum error in image.

(3) Maximum error in central 30 degrees of image.

END

11-87

DTIC