SEGMENTATION OF THREE DIMENSIONAL IMAGES

by

Wesley E. Snyder

Senior member, IEEE

and

Griff Bilbro

Member, IEEE

Technical Report TR-84/7

Center for Communications and Signal Processing

North Carolina State University

The research reported in this paper was supported in part by the Corporate Research and Development Center, General Electric Company.

Abstract

In this paper, we describe the fundamental differences between three-dimensional (range) images and two-dimensional (luminance) images. A number of problems arise which are unique to range data, including in particular a strong sensitivity to quantization effects.

Although range images and luminance images are both arrays of scalars, the range image conceptually represents a surface in space and cannot be naively manipulated using the conventional image processing functions such as 3×3 convolution kernels.

If the range data are regarded as a sampling of a surface parametrized by the focal plane coordinates, it is possible to find a representation for the surface normal and for the surface curvature in terms of familiar-looking convolution kernels.

1. Introduction

Although not a new field,[4] analysis of three-dimensional imagery has become a popular field of research recently, as evidenced by the substantial numbers of papers presented at conferences[1].

This increase in popularity can be attributed to many causes, including the availability of range sensors[8, 9, 10] , and a recent emphasis on industrial machine vision[7, 6] , for which such sensors are uniquely appropriate.

The purpose of this paper is twofold: first to provide an analytic description of range imagery and show why the classical image processing operators are not well suited for dealing with such imagery; and second, to propose and discuss some strategies which are appropriate. In pursuit of the first objective, we will initially propose some segmentation strategies which seem reasonable, and then show why they do not work.

The first conference on Artificial Intelligence Applications, Dec. 84, has a session dedicated solely to 3-D, plus a session on vision in which several papers utilize 3-D concepts.

(1) Determine the *critical points* in the input range image. Intuitively, these are the points at which the surface curvature undergoes a rapid change. Typical such points include the edges of polyhedra.

(2) Considering only those points determined NOT to be critical points, perform a connected-component analysis to determine those regions in which the surface curvature changes smoothly. In the case of polyhedra, there should be no change in curvature.

(3) Assign critical points to smooth regions based on some distance metric.

(4) Extract an abstract description of the regions (faces) and derive a graph-based representation.

(5) Match the observed graph to models.

3. Determining critical points using surface normals

In this section we postulate that rate of change of normal direction can be used as an indication of the presence of critical points. Following this train of thought, we will attempt to analytically develop a 3-D edge finder.

3.1 Determining rate of change of normal

Given a surface described by z(x,y) a normal to that surface is

1.1 Notation

We define the input range image in terms of a 3-vector

$$[i,j,r]^T \equiv \mathbf{r}(i,j),$$

where i and j represent the integer-valued sampling points on a hypothetical focal plane (see Figure 1 for a model of the imaging system). For a given vector r, we can define a transformation λ, which yields Cartesian coordinates, by

$$\mathbf{x} = \lambda(\mathbf{r}); \text{ or} \tag{1}$$

$$\mathbf{x}(i,j) = [x(i,j), y(i,j), z(i,j)]^T, \tag{2}$$

where we have explicitly shown the dependence on the focal plane (sampling grid) coordinates.

$$z(i,j) = \lambda_z(\mathbf{r}) = \left[\frac{r^2 f^2}{f^2+i^2+j^2}\right]^{1/2} \tag{3}$$

$$x(i,j) = \lambda_x(\mathbf{r}) = \frac{iz(i,j)}{f} \tag{4}$$

$$y(i,j) = \lambda_y(\mathbf{r}) = \frac{jz(i,j)}{f} \tag{5}$$

The normal to a surface at a point x(i,j) is denoted by

$$\mathbf{N}(i,j) \equiv [n_x(i,j), n_y(i,j), n_z(i,j)]^T. \tag{6}$$

(We will continue to explicitly show the dependence on the sampling grid coordinates, since that dependence will turn out to be critical).

2. Philosophy

In this paper, we will make use of a bottom-up analysis philosophy. Whether this is the best philosophy remains to be seen. However, for the pedagogical purposes of this paper, it will suffice. This philosophy defines the steps in the analysis process as follows:

$$\mathbf{N} = \begin{bmatrix} \frac{\partial z}{\partial x} \\ \frac{\partial z}{\partial y} \\ -1 \end{bmatrix} \tag{7}$$

Points at which the normal undergoes a rapid change are likely to be critical points. To identify such points requires a scalar measure of change in a vector quantity, the normal. One such measure is the divergence,

$$D = \nabla \cdot \mathbf{N}, \tag{8}$$

where ∇ is the gradient operator:

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{bmatrix}^T. \tag{9}$$

Then

$$D = \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \tag{10}$$

and we have re-derived the Laplacian. That critical points should be determined by the Laplacian is intuitively pleasing, and no real surprise.[5] can be computed (presumably) by linear operators, we have some genuine hope for a real-time hardware implementation. However, when we apply this result to realistic range data, we will encounter some problems, as the next section illustrates.

3.2 Applying linear operators to range images

For a linear, shift-invariant operator Γ, operating on a continuously differentiable function f(x), we know that there exists a function h, such that

$$\Gamma(f(x)) = h(x) \times f(x). \tag{11}$$

Here, $\times$ represents convolution. This theorem has found application most often in image analysis in estimating derivatives; for it allows us to use a convolution operator to find the derivative of the image function.

Given that we have a library of image processing functions, including

convolution operators, we are tempted to use such library functions to estimate the derivative described in section 3.1. Doing so, however, will yield meaningless results, for the following reason:

Let S be a linear operator as defined in equation 11 which estimates a derivative, say, the partial derivative with respect to x. (The Sobel operator is typical). The application of S to z estimates

$$S(z(i,j)) \approx \frac{\partial z}{\partial i} \tag{12}$$

Thus, use of the convolution kernal estimates the derivative with respect to focal plane coordinates, not (as we might have hoped) with respect to spatial coordinates.

We can correct this problem in two possible ways: use of the chain rule, and explicit calculation of derivatives. Let us examine these approaches briefly:

USE OF THE CHAIN RULE:

We know that

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial i} \cdot \frac{\partial i}{\partial x} \tag{13}$$

$\frac{\partial z}{\partial i}$ can be simply estimated by a convolution. However, $\frac{\partial i}{\partial x}$ must be evaluated along the surface; and at this point, the first major problem occurs. Consider a surface with equation x(i) = x_o, (figure 2). Along this surface, $\frac{\partial x(i)}{\partial i} = 0$, and $\frac{\partial i}{\partial x}$ is therefore infinite. That this should be the case is clear from the geometry, since an infinitesimal change in x will result in a finite change in i or z.

Here, we have the first fundamental difference between range and luminance imagery. In luminance imagery, derivatives can never be infinite. In range imagery, accidental alignment with the coordinate axes, and the resulting infinite derivatives, occurs routinely.

Even in the case of a surface which is not aligned with a coordinate axis, a simple linear operator applied to the focal plane image is not adequate for

estimating derivatives, for

$$\frac{dz}{dx} = \lim_{\Delta x \to 0} \frac{z(x+\Delta x) - z(x)}{\Delta x} \tag{14}$$

In luminance imagery, Δx is always taken to be one. When dealing with range imagery, however, we must note that

$$\Delta x = x(i+\Delta i) - x(i) \tag{15}$$

In section 3.1.2 and again in section 5.1, we will address the issue of computing both normals and derivatives.

3.2 Determining normals by cross products

Given a surface z(x,y), let $\mathbf{T}(x_o, y_o)$ represent the plane tangent to z at the point $[x_o, y_o]$. Then, given any two distinct vectors lying in **T**, say $\mathbf{V}_1$ and $\mathbf{V}_2$, the normal to z at $[x_o, y_o]$ can be found by

$$\mathbf{N}(x_o, y_o) = \frac{\mathbf{V}_1 \times \mathbf{V}_2}{|\mathbf{V}_1 \times \mathbf{V}_2|} \tag{16}$$

This formulation gives us an algorithm for estimating the normal which is independent of the coordinate axes and therefore does not suffer from the problem of infinite derivatives.

The algorithm is as follows: Let $\mathbf{X}(i,j) \equiv \begin{bmatrix} x(i,j) \\ y(i,j) \\ z(i,j) \end{bmatrix}$ as in equation 2.

(1) Compute $\mathbf{V}_1(i,j) = \mathbf{X}(i+1,j) - \mathbf{X}(i-1,j)$

(2) $\mathbf{V}_2(i,j) = \mathbf{X}(i,j+1) - \mathbf{X}(i,j-1)$

(3) Apply eq 16 to find **N**

At this point, we seem to have developed a robust technique for determining the surface normal vectors, and indeed we have. Whether such analysis will prove to be useful in the presence of noise is an issue which will be addressed in section 5. At this point, we will assume that the normals which we have determined are good local representations of the surface, and use them in a scene segmentation system.

4. Surface segmentation strategies

In this section, we will list three strategies for segmenting a range image into planar surfaces. The first two will make use of continuity in the surface normal as a criterion for region growing. The third will use perpendicular distance to a plane.

4.1 Use of variations in normal direction

A *region* in an image is defined as a set of connected pixels. The definition may be made recursive by saying that for a pixel, $[i,j]$, and a region R,

$$[i,j] \in R \text{ iff } [k,l] s.t. \text{ CONNECTED}([i,j],[k,l]) \bigcap [k,l] \in R \tag{17}$$

The predicate CONNECTED thus defines the essential criterion for region growing. Evaluation of the recursive definition given in eq. 17 may be performed by use of a push down stack [ ], or an equivalence table [Snyder and Savage]. The details need not concern us here. What is important, however, is the choice of the best CONNECTED predicate. One potentially attractive predicate would be

$$\text{CONNECTED}([i,j],[k,l]) <=>$$

$$\{(|i-k| = 1) \bigcup (|j-l| = 1)\} \bigcap \{\mathbf{N}(i,j) \cdot \mathbf{N}(k,l) < |Thr|)\} \tag{18}$$

for some threshold *Thr*. That is, two pixels are connected if they are adjacent on the focal plane and their normals point in roughly the same direction.

Figure 3 shows a range image of three polyhedra used to test these algorithms. As this is a range view, darker intensities indicate pixels which are closer to the sensor. In figure 4, we illustrate the results of applying the region growing algorithm. Each pixel is identified by a character indicating to which region (face) the pixel belongs. Pixels marked with an exclamation point represent points at which the normal direction changes rapidly.

In the center of faces 3 and 7, anomalous edge pixels occur, indicating one of the problems with this approach. Both faces are flat, and aligned with the focal (x - y) plane. However, due to noise, some pixels differ in depth by as much

as one unit. In Figure 5, the effect of such variations is illustrated in detail. A cross section through the surface is shown, in which noise at pixel 4 has resulted in a one-unit change in depth. Since the normal at a given point (e.g. pixel 3) is computed by considering pixels on either side of that point (pixels 2 and 4), the algorithm detects a significant direction change in the vicinity of pixel 4, as illustrated by the arrows.

This example would seem to indicate an extreme sensitivity to noise, although it might be argued that, since the surface really does drop at that point, an algorithm based solely on SURFACE properties *should* detect the drop. We will return to the discussion of noise sensitivity in section 5.

4.2 Subtraction of normals

Dean[1] has implemented an algorithm for detecting changes in the surface normal similar to that described in section 4.1, although in his algorithm, differences are detected by subtracting adjacent normal vectors, rather than taking dot products. The results in sensitivity were found to be similar to those described in section 4.1.

4.3 The three point seed method

Bhanu [3] describes a different strategy for finding planar faces which eliminates some of the previously described sensitivity problems. His algorithm may be described succinctly as follows: (See Figure 6)

Algorithm for three point seed method

1. From the list of surface points, select three points which are noncolinear and near relative to sampling distances.
2. Obtain the equation of the plane passing through the three points chosen in step 1.
3. Find the set of points **P** which are very close to this plane.
4. Apply a convexity condition to the set **P** to obtain a reduced convex set **P'** .

This separates faces lying in the same plane.

5. Check the set $\mathbf{P}'$ obtained in step 4 for narrowness. The narrowness condition eliminates faces whose points all lie very near the same line.

Bhanu's method eliminates many of the problems inherent with the surface normal method, since it considers distance rather than direction. In a subsequent processing step (program ANALYZE) the authors have used a similar method for eliminating errors. This is in conformity with step 2 of the philosophy given earlier. That process is the subject of a separate paper.

5. Sensitivity to noise and quantization

Figure 7 illustrates the variations in surface normal due to the effects of quantization. That figure shows depth (z) on a single horizontal scan line of the image shown in Figure 3. The ripples appearing on the otherwise flat surfaces are due to quantization effects. Since the original data r(i,j) is quantized, (r takes on only integer values between 0 and 255) and sampled, (i and j take on integer values between 0 and 127), the application of equations 3-5 fails to reconstruct a flat surface exactly. Instead, we see rapid local fluctuations in surface tangent direction. Note that the frequency of the fluctuation is dependent on the angle between the point in question and the focal axis. Although small in absolute magnitude, these quantization effects have a radical impact on surface direction.

One's immediate reaction to this type of noise is to apply some sort of filter. However, such filters need to be applied very carefully. Consider the face maked "4" in Figure 7. That face subtends only six pixels in the image (See Figure 10, which illustrates the final segmentation results). Furthermore, face 8 is only one pixel wide! No linear noise filter will be able to smooth the quantization noise while at the same time preserving such obliquely-viewed faces.

At least two approaches are possible for dealing with quantization noise while preserving oblique edges. The first is to perform the segmentation in a coordinate system in which such artifacts do not occur, and the second is to develop an adaptive non-linear filter to remove the noise. The first option is explored in the next section.

5.1 An alternative coordinate system

Two major problems have been identified so far in this paper which are unique to 3-D imagery. The first is that derivatives go to infinity along the coordinate axes, and the second is that the transformation λ which takes $[i,j,r]^T$ into $[x,y,z]^T$ introduces quantization artifacts. Both of these problems may be eliminated, or their impact lessened, by a proper choice of a coordinate system.

One such coordinate system, referred to as Ω, is shown in Figure 8. This system uses modified spherical coordinates. Its use in this application is very attractive since it is "natural" to the range sensor. The three coordinates are:

r: range from sensor to surface

θ: angle between the x axis and the projection of $\mathbf{r}$ into the xz plane

ϕ: angle between the y axis and the projection of $\mathbf{r}$ into the yz plane

- Singularities can never occur, for any face that aligns with a coordinate axis is also invisible to the sensor
- Because no conversion to z is performed, the effects of quantization are lessened.

In Ω, the normal vector at a point is found by

$$\mathbf{N} = \begin{bmatrix} \frac{\partial r}{\partial i} \\ \frac{\partial \theta}{\partial i} \\ \frac{\partial \phi}{\partial i} \end{bmatrix} \times \begin{bmatrix} \frac{\partial r}{\partial j} \\ \frac{\partial \theta}{\partial j} \\ \frac{\partial \phi}{\partial j} \end{bmatrix} \tag{19}$$

$$= \frac{\partial \theta}{\partial i}\frac{\partial \phi}{\partial j}\,\mathbf{r} + \frac{\partial r}{\partial i}\frac{\partial \phi}{\partial j}\,\theta + \frac{\partial r}{\partial j}\frac{\partial \theta}{\partial i}\,\phi \tag{20}$$

Now, since

$$\theta = \tan^{-1}\frac{f}{i} \text{ and} \tag{21}$$

$$\phi = \tan^{-1}\frac{f}{j}, \tag{22}$$

$\frac{\partial\theta}{\partial i}$ and $\frac{\partial\phi}{\partial j}$ are straightforward to compute. Furthermore, since only a small number of values are possible for i and j (between 128 and 512 typically), these two terms may be found by lookup table.

Finally, $\frac{\partial r}{\partial i}$ and $\frac{\partial r}{\partial j}$ can be determined by application of a convolution kernal.

By performing the same segmentation algorithm defined in section 4, but in Ω, we therefore reduce sensitivity to quantization and axis alignment. It should be noted that Ω is not an orthogonal coordinate system. In using Ω rather that Cartesian or Spherical coordinates, we have avoided singularities at the price of orthogonality.

6. Determining curvature

The strategies described so far have implicitly assumed a scene composed of polyhedral objects. To analyze more realistic environments, we need to develop a technique for segmenting curved surfaces. One option would be to represent arbitrary surfaces as a set of convex planar surfaces [Bhanu], however, this representation makes higher-level model matching much more complex, since the segmentation becomes ambiguous and dependent on viewing angle. We desire, if possible, to follow the philosophy given in section 2, with appropriate extensions to cover the case of curved surfaces.

This problem is yet to be satisfactorily solved. In this section, we will define the problem carefully, and provide some suggested approaches.

Curvature is defined in terms of a curve in space, where a curve may be thought of as a warping of a line, just as a general surface may be considered to be a warping of a plane. To extend the definition of curvature from a property of lines to a property of a surface, we need to first observe that a surface in 3-space may be viewed as a set of vectors, and we must then examine the properties of such a set.

6.1 Parametrized Surfaces and Arrays of Vectors

In range imagery, the raw data may be converted into an array of vectors, representing points on surfaces in three space. This underlying surface may be discussed in terms of continuous versions of the focal plane indices. Arrays of vectors can be fit with arbitrarily complex models using a linear least squares procedure, if the focal plane array indices are regarded as a parametrization of the surfaces. Derivatives and other properties of the surface may be extracted from these fits or by other convenient means. Such a parametrization of a surface (two dimensions) embedded in three dimensions is a vector function of two arguments,

$$\mathbf{x}(\alpha,\beta) = \Big[x(\alpha,\beta),\ y(\alpha,\beta),\ z(\alpha,\beta)\Big]^T,$$

where α and β are real and continuous. It is usual (and sometimes useful) to think that α and β have some physical significance and that they are somehow attached to the surface, but neither interpretation is necessary. It is sufficient that the parameters are monotonically related to distance in the surface. Analogously, curves (one dimension) embedded in the plane (two dimensions) are often parametrized either by time for trajectories or by path length for boundaries. So that the curve is described in time by the 2-vector function, $\mathbf{x}(t) = [x(t),\ y(t)]^T$. The same trace is desribed by the related 2-vector function $\mathbf{x}'(s) = [x'(s),\ y'(s)]^T$, parametrized by path length. It is important to observe that such a curve can be equivalently parametrized by any monotonic function of any other parametrization (with a corresponding accomodation in the behavior of the functions x and y) and with no consequence to the actual shape of the curve. The penalty of course is that local measures of the curve like path length have less convenient (and less familiar) expressions. For example the differential of path length for the curve defined by x' can be extracted from the time-parametrization

$$ds = \left[\left(\frac{dx'}{dt}\right)^2 + \left(\frac{dy'}{dt}\right)^2\right]^{1/2} dt \qquad (23)$$

where the coefficient of dt is the magnitude of the instantaneous velocity vector.

On the other hand, for the parametrization by path length,

$$ds = \left[\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2\right]^{1/2} ds, \tag{24}$$

where the corresponding coefficient is the magnitude of the tangent vector which is unity. This is what makes the parametrization "by path length".

It is easily shown that distance in the focal plane is monotonically related to distance in the observed surface represented by the focal plane array (see figure 1). The array indices, i and j, are similarly related to distance in the focal plane. Therefore the array indices comprise a sampling of a perfectly acceptable (however peculiar) candidate for parametrizing the observed surface. Noise does not invalidate this result. We can think of the data $\mathbf{x}_{i,j} = [x_{i,j}, y_{i,j}, z_{i,j}]^T$ as a noisy sampling at descrete values of some function $\mathbf{x}(\alpha,\beta)$ where α and β are continuous versions of the discrete variables i and j. We will write this function $\mathbf{x}(i,j)$ with the convention that independent variables will be written as subscripts when they are discrete indices, and as parenthisized arguments when they are the corresponding continuous and real analogs. The problem is then to remove the noise from $\mathbf{x}_{i,j}$ and estimate $\mathbf{x}(i,j)$.

If the partials can be calculated, then $\frac{\partial \mathbf{x}}{\partial i}$ is a vector tangent to the surface, and $\frac{\partial \mathbf{x}}{\partial j}$ is a linearly independent tangent. The unit normal vector to the fitted surface at a point is

$$\mathbf{N} = \frac{\frac{\partial \mathbf{x}}{\partial i} \times \frac{\partial \mathbf{x}}{\partial j}}{\left|\frac{\partial \mathbf{x}}{\partial i} \times \frac{\partial \mathbf{x}}{\partial j}\right|}, \tag{24}$$

which is a restatement of equation 16. The change of this unit normal in a small neighborhood of the point is determined by the curvature of the surface. This change is linear map which can be represented in a given coordinate system by a tensor, which is a special kind of matrix. The two Eigenvalues of this tensor are the principal curvatures of the surface at this point.

In the study of surfaces, the Eigenvalues themselves are not used as much as are two particular quantities derived from the Eigenvalues. The first is the mean curvature, which is half the sum of the principal curvatures, or half the trace of the curvature map. The second is the gauss curvature, which is the product of the two principal curvatures, or the determinant of the curvature map. The mean and gauss curvatures are invariants: they depend only on the surface itself, not the coordinate system or the parametrization.

Much of the local behavior, including the mean and gauss curvatures of the surface, is determined by six well known scalars.[11] The three coefficients of the First Fundamental Form at some point on the surface describe the distance in the surface relative to distance in the plane tangent to the surface at that point,

$$E = \frac{\partial \mathbf{x}}{\partial i} \cdot \frac{\partial \mathbf{x}}{\partial i}, \tag{26a}$$

$$F = \frac{\partial \mathbf{x}}{\partial i} \cdot \frac{\partial \mathbf{x}}{\partial j}, \tag{26b}$$

and

$$G = \frac{\partial \mathbf{x}}{\partial j} \cdot \frac{\partial \mathbf{x}}{\partial j}. \tag{26c}$$

The three coefficients of the Second Fundamental Form describe the change of the surface normal near a point,

$$7a)e = \mathbf{N} \cdot \frac{\partial^2 \mathbf{x}}{\partial t^2}, \tag{2.uh}$$

$$f = \mathbf{N} \cdot \frac{\partial^2 \mathbf{x}}{\partial i \partial j}, \tag{27b}$$

and

$$g = \mathbf{N} \cdot \frac{\partial^2 \mathbf{x}}{\partial f^2}, \tag{27c}$$

where $\mathbf{N}$ is the unit normal vector defined above in terms of the first derivatives. Now, the mean curvature is given by

$$k_m = \frac{eG - 2fF - Eg}{EG - F^2}, \tag{28}$$

and the gauss curvature is given by

$$k_g = \frac{eg - f^2}{EG - F^2}. \tag{29}$$

These quantities address the shape of the surface, not its orientation or its absolute location in some coordinate system. Together, the mean curvature and the gauss curvature at a point on a surface completely characterize that portion of the second order behavior of the surface normal which does not depend on the coordinate system. A sphere of radius ρ, for example, has $k_g = \rho^{-2}$ and $k_m = \rho^{-1}$ everywhere. A cylinder of radius ρ has $k_g = 0$ and $k_m = 1/2\rho$. A cone has $k_g = 0$ everywhere since, like the cylinder, one of the principal curvatures is zero but k_m approaches infinity at the apex. A saddle point may have large negative k_g and have k_m=0.

It is possible to estimate these various mixed partial derivatives by convolutions:

$$\frac{\partial^\mu}{\partial i^\mu}\frac{\partial^\nu}{\partial j^\nu}\mathbf{x}(i,j) = \sum_{I,J=-H}^{I,J=H} \mathbf{x}_{i+I,j+J} K^{\mu,\nu}_{I,J}, \tag{30}$$

where the kernels look like Sobel operators, and are described in [Haralick] and are used in the same way that a Sobel operator is used. The only practical difference is that the data here is an array of vectors rather than an array of scalars. The various mixed partials must be computed separately for each component of the array. The resulting components can then be used to compute the cross product of Equation 24 and the dot products of Equations 26 and 27.

The kernals[2] through second order for the smallest neighborhood that gives non trivial results are reproduced here for convenience,

$$K^{0,0} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \tag{31a}$$

$$K^{1,0} = \frac{1}{6}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \tag{31b}$$

$$K^{0,1} = \frac{1}{6}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \tag{31c}$$

$$K^{1,1} = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \tag{31d}$$

$$K^{2,0} = \frac{1}{3}\begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}, \tag{31e}$$

and

$$K^{0,2} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix}. \tag{31f}$$

There exist two third order and one forth order kernal for a $3by3$ neighborhood, but they do not contribute to the second order partials at the center of the neighborhood. The zeroth order kernal is not used in this application, but it is instructive to notice that it is the neighborhood average. The first order kernals appear only in the computation of the local normal.

Conceptually, the range array is used to generate an x array for the surface, as well as arrays for y and z. The i and j derivatives of first and second order are obtained from each of these arrays by convolving each of the component arrays with the K arrays. Then the components of $\frac{\partial \mathbf{x}}{\partial i}$ as well as the other vector partials are known for each pixel in the focal plane. These vector partials can be used to compute curvatures or normals or projected areas for any pixel.

7. Conclusion

In this paper, we have described the fundamental differences between three-dimensional (range) images and two-dimensional (luminance) images. A number of problems have been described which are unique to range data, including in particular a strong sensitivity to quantization effects.

We have shown that although range images and luminance images are both arrays of scalars, the range image conceptually represents a surface in space and cannot be naively manipulated using the conventional image processing functions such as 3×3 convolution kernels.

We then show that by regarding the range data as parametrized by the focal plane coordinates, it is possible to find a representation for the surface normal and curvature in terms of familiar-looking convolution kernels.

References

1. Clayton Dean, *Three-Dimensional Object Identification*, PhD Thesis, University of Penn (1982).
2. R.M. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives," *IEEE PAMI*, (Jan. 1984).
3. T.C. Henderson and B. Bhanu, "Three point seed Method for the Extraction on Planar Faces from Range Data ," *IEEE Conference on Industrial Applications of Machine Vision*, (1983).
4. B. Horn, "Sequins & Quills - Representatinos for Surface Topography ," *Workshop . on Representations of 3-D Objects* , Univ of Penn., (May. 1979).
5. S. Hsu, J. Mundy, and R. Beaudet, "Web Representation of Image Data ," *4th International Joint Conference on Pattern Recognition*, (1978).
6. J.F. Jarvis, "Research Directions in Industrial Machine Vision: A Workshop Summary ," *IEEE Computer* , (Dec. 1982).

7. J.F. Jarvis and J.L. Mundy, eds., "Special Section on Industrial Applications of Machine Vision ," *IEEE PAMI*, (Nov. 1983).

8. R.A. Jarvis, "A Perspective on Range Finding Techniques ," *IEEE PAMI*, (March. 1983).

9. R.A. Jarvis, "A Laser time-of-flight Range Scanner for Robotic Vision ," *IEEE PAMI*, (Sept. 1983).

10. G.B. Porter and J.L. Mundy , "A Noncontact Profile Sensing System for Visual Inspection ," *6ICPR*, (Oct. 1982).

11. M.P. doCarmo, *Differential Geometry of Curves and Surfaces* , Prentice-Hall (1976).

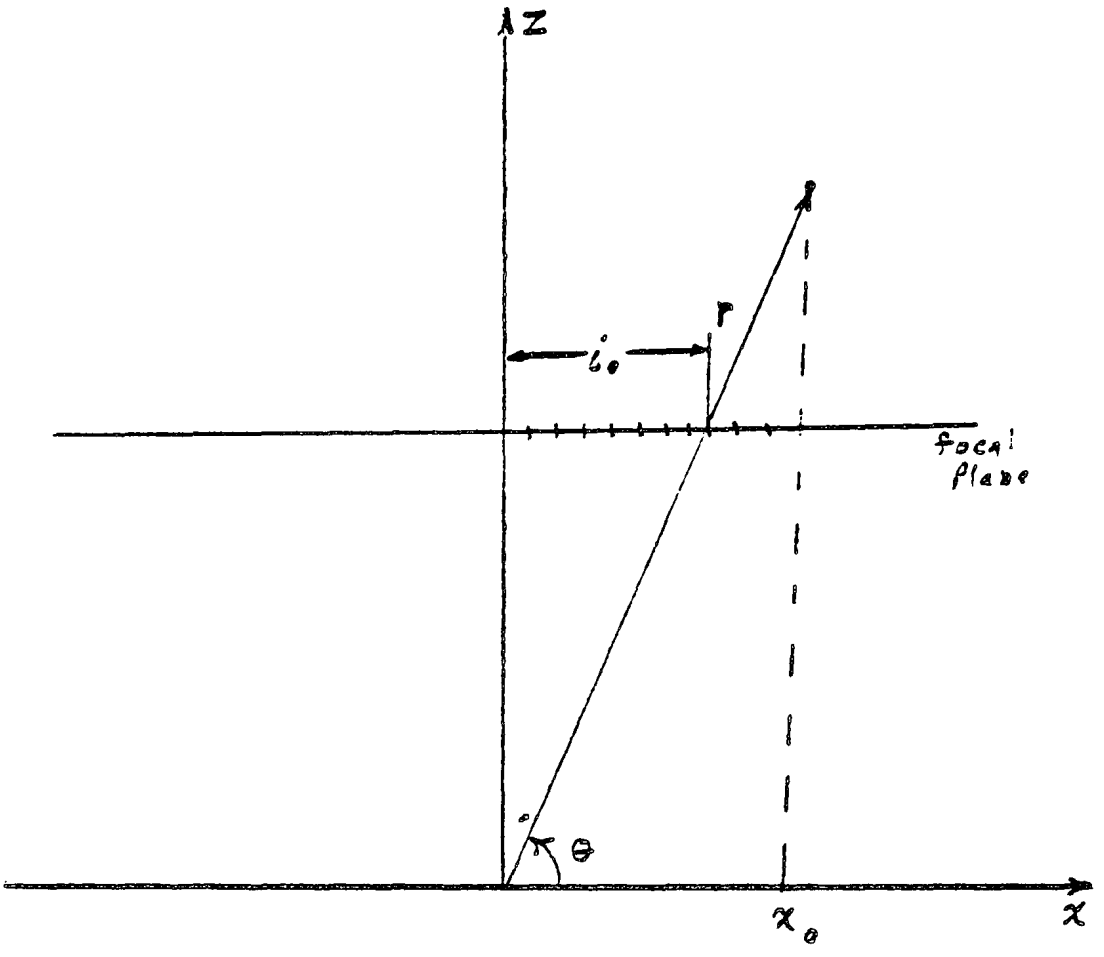


Figure 1, Imaging geometry
The y axis is perpendicular
to the x-z plane.

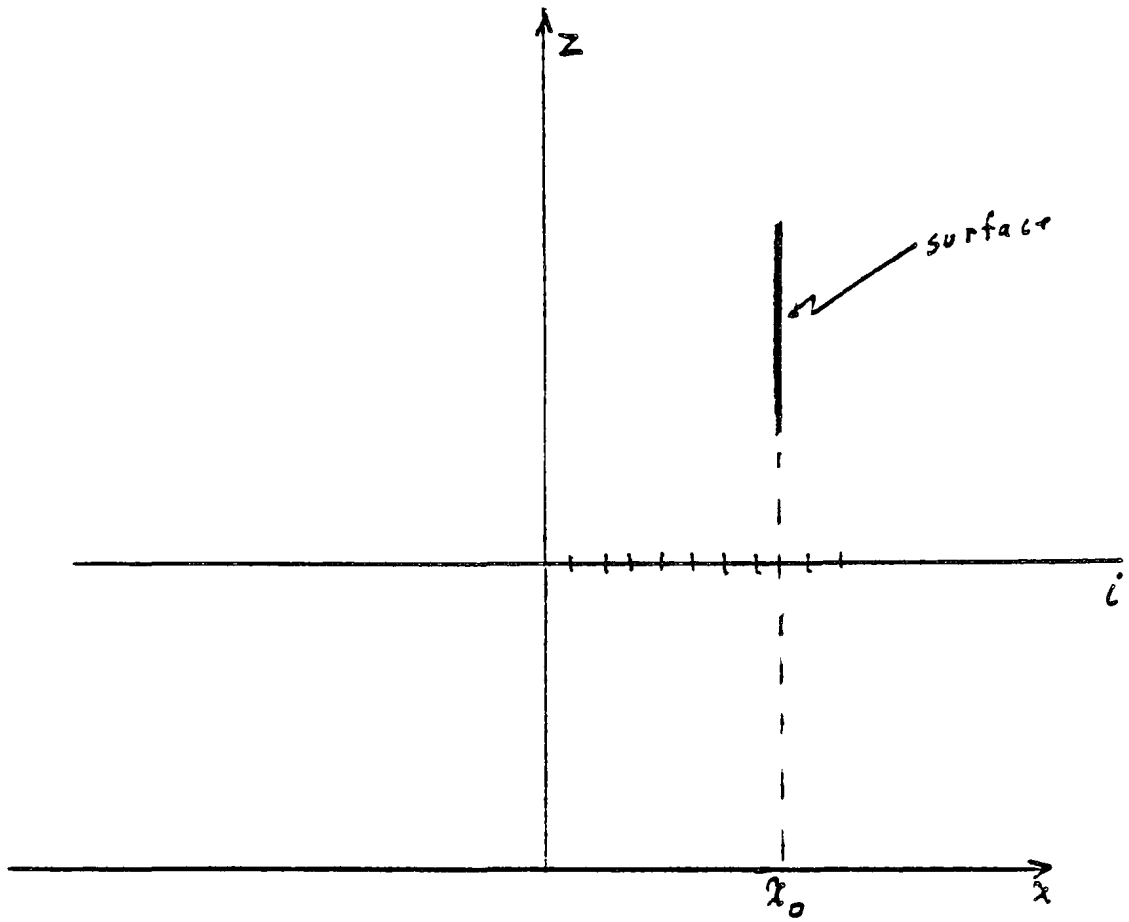


Figure 2
A singular surface alignment with the z axis causes $\frac{\partial z}{\partial x}$ to be infinite.

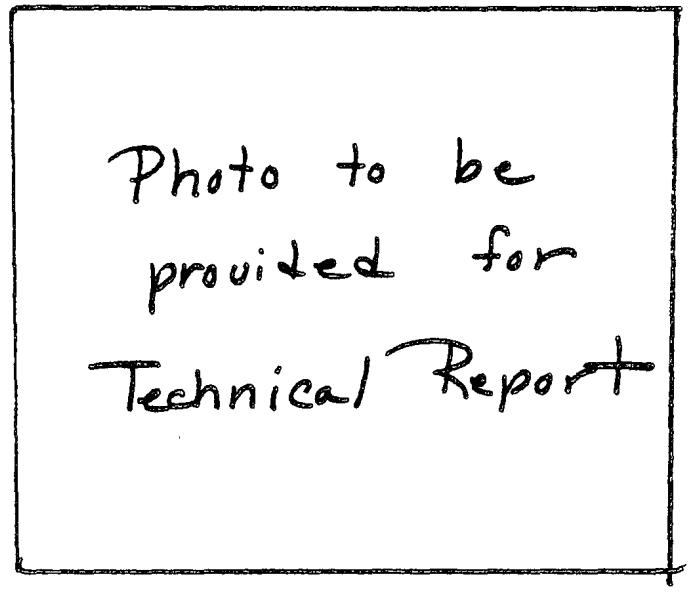


Figure 3

A Range Image

relabeled image:

Figure 4

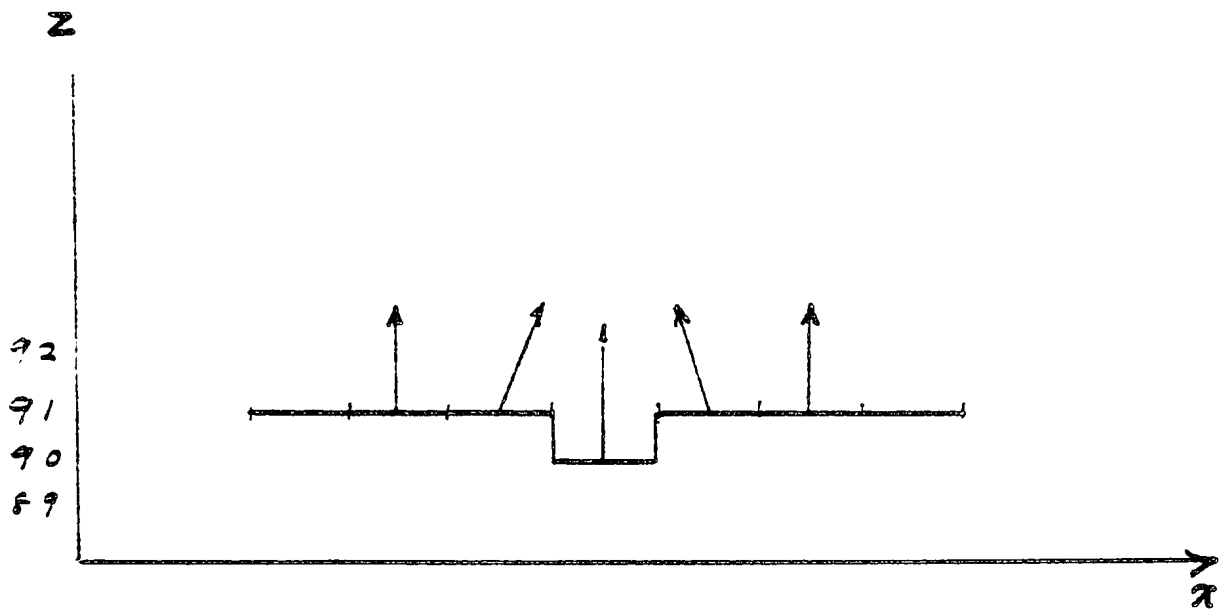
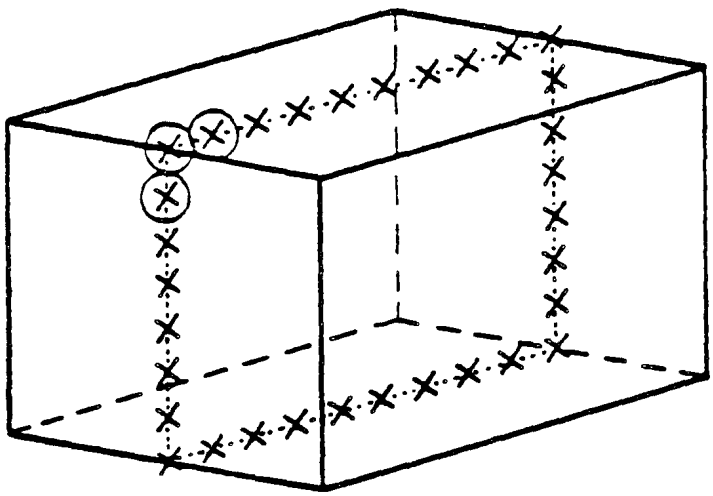
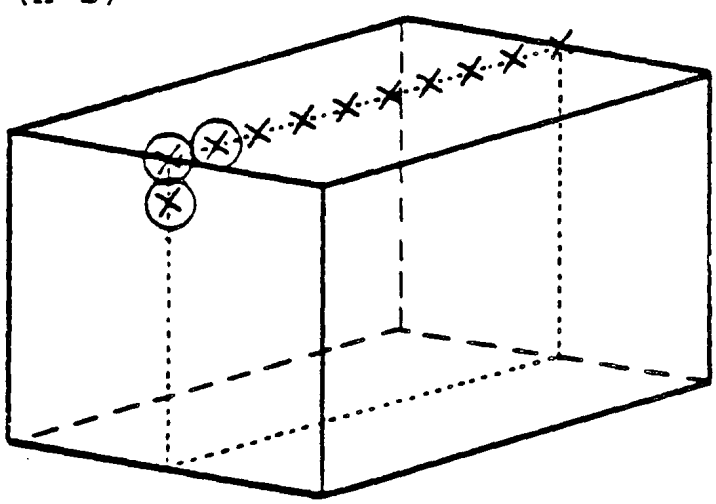


Figure 5

FIG. 6 CONVEXITY AND NARROWNESS CONDITIONS



(a) A three point seed (the circled x's) which straddles an edge produces a plane which cross-sections the object. The convexity condition reduces the set of points (x's)



(b) The narrowness condition excludes faces like this whose points all lie very near the same line

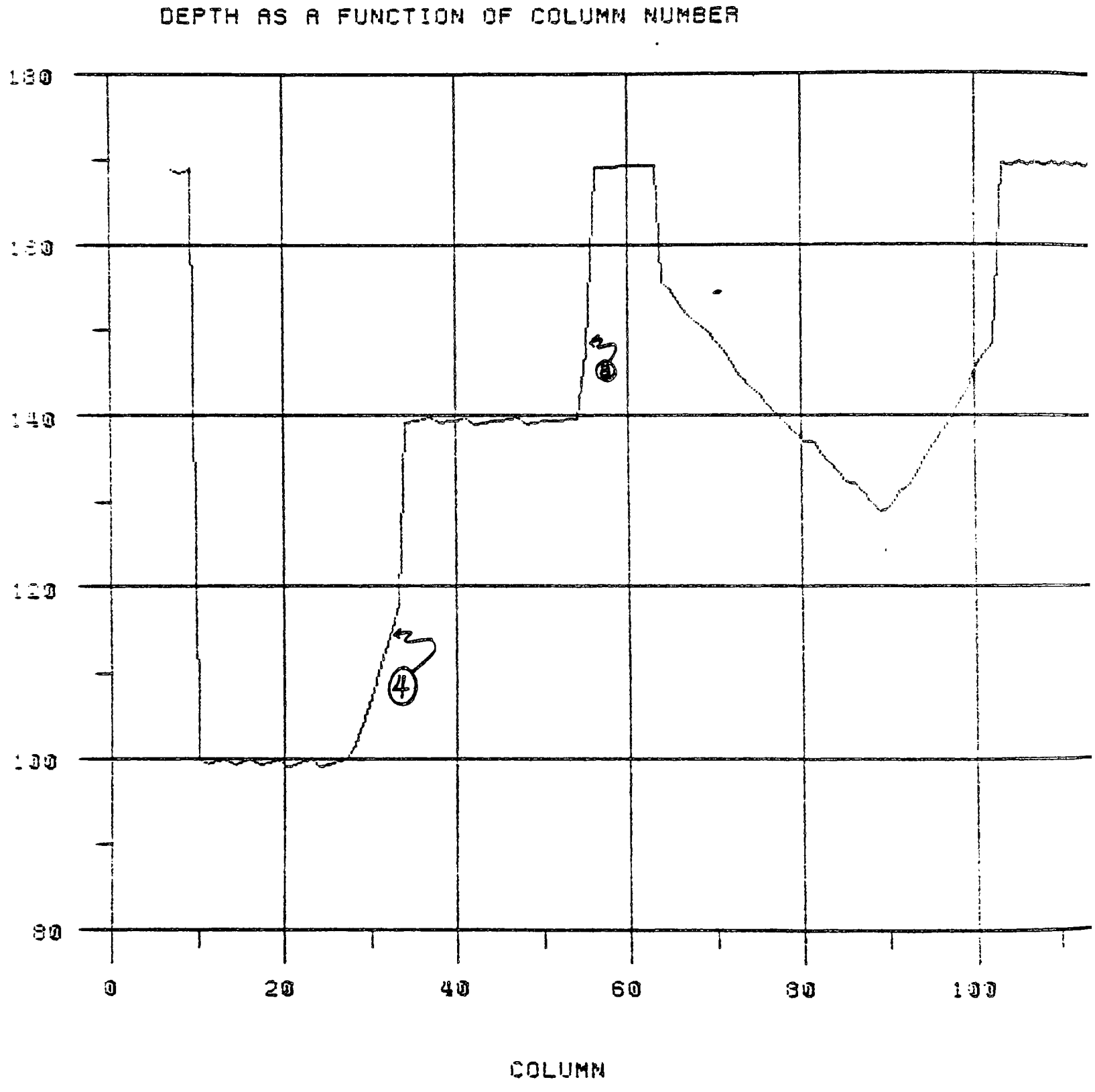

DEPTH AS A FUNCTION OF COLUMN NUMBER
180
160
140
120
100
80
0
20
40
60
80
100
COLUMN
14:59:10

Figure 7

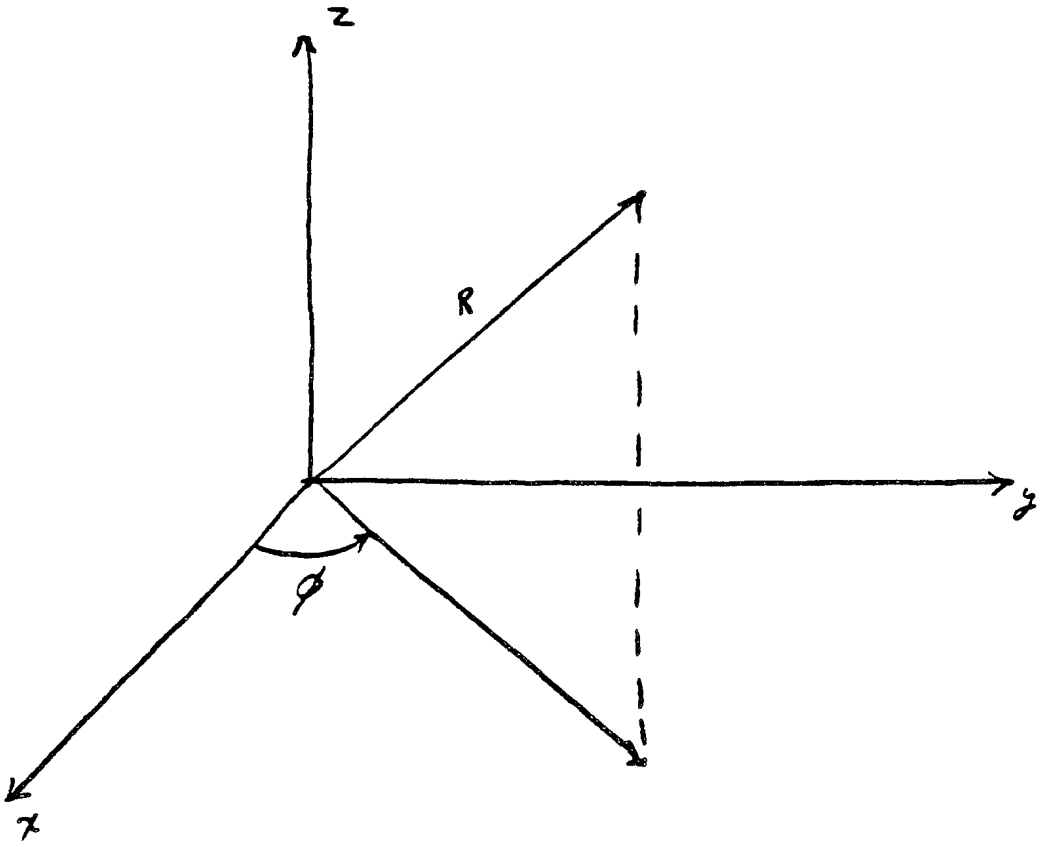


Figure 8

An alternative coordinate system. Only ϕ is shown. θ results from a similar projection onto the y-z plane.