SEGMENTATION OF EDGES IN 3-D OBJECT SPACE

Amnon Krupnik

Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

In our feature-based matching approach, zero-crossings are matched and represented in 3-D object space by a sequence of densely spaced points. These spatial curves form the basis for reconstructing the surface. Since edges are likely to correspond to object boundaries, the 3-D curves also serve as an important input for object recognition. In this paper we address the problem of segmenting the contours in straight lines and regular curves. We compare different methods, such as split-and-merge and a 3-D version of the $\psi - S$ method.

KEY WORDS: Edge Segmentation, Curve Decomposition, 3-D, Image Matching, Feature Extraction

1 INTRODUCTION

One of the goals of digital photogrammetry is to automatically recognize and extract man-made objects from aerial images. An essential step toward this goal is to extract features and to match them. We have adopted this approach and described it in several papers, e.g., (Schenk, 1989, Schenk et. al, 1991a, Zong et. al, 1991). A similar approach is also accepted by the computer vision community (e.g., Grimson, 1985).

The features detected in the images are discontinuities of gray values, or edges. In our current implementation, the images are convolved with the Laplacian of a Gaussian (LoG) operator (Marr and Hildreth, 1980). The resulting zero-crossings are the edges. In the automatic orientation module, which is the first stage of our system, the zero-crossings are matched for determining conjugate points (Schenk et. al, 1991b, Stefanidis et. al, 1991, Zong et. al, 1991). Once the orientation parameters are established, the images are resampled to epipolar geometry (Cho et. al, 1992). Now we begin to reconstruct surfaces where many edges are matched (Zong and Schenk, 1992), resulting lists of densely spaced points in object space (3-D edges)

The feature-based matching approach offers two major advantages:

- Surface discontinuities are most likely to show up as edges in the image. By detecting these edges, breaklines can be found and the surface reconstruction process (Schenk et. al, 1991a, Schenk and Toth, 1992) becomes more robust.
- In many of the cases, edges correspond to object boundaries. Therefore, once the location of edges is known, they serve as building blocks for a symbolic description of the object space. Such descriptions can be matched with symbolic representations of "world" objects, stored in a library.

In order to use the edges for the symbolic description of the object space, we must segment and group them. In this paper, we focus on the segmentation aspect. The goal is to decompose the 3-D curves into primitives which are more explicit than a list of densely spaced points. Specifically, we want to segment the 3-D curves into straight lines, regular curves (circular arcs in our current implementation) and natural lines.

The curve segmentation problem has been addressed extensively in computer vision literature. A popular segmentation method is the Hough transform (Ballard and Brown, 1982). This method tries to find straight lines from a sparse set of points. In our application points are already organized along edges; Thus, the Hough transform would unnecessarily increase the computational complexity. Ramer (1972) presents a simple algorithm to approximate planar curves by polygons. He based his approximation on a maximum offset criterion. We have adopted this criterion in our approach. Pavlidis and Horowitz (1974) use a least-squares algorithm to fit straight lines to portions of the curve, and then iterate a split-merge procedure to refine the initial segmentation. Grimson and Pavlidis (1985) find the breakpoints of a curve by comparing the original and a smoothed version of the curve. Discontinuities are then easily detected, and regular curve fitting is performed only between discontinuities. Fischler and Bolles (1986) describe two methods, one of them passes a "stick" of a certain width and length over the curve, and the other looks at the curve from different "views" followed by a selection of breakpoints according to the maximum votes obtained from these views. Both methods are based on segmenting the curve over different scales, and on perceptual organization. Grimson (1989) suggests an approach which is a combination of split-and-merge and $\psi - s$ algorithms. Wuescher and Boyer (1991) describe an algorithm based on a constant curvature criterion.

Except for the Hough transform, all the proposed methods consider a plane curve as the input for the process. Grimson (1989) mentioned that the segmentation can also be performed in 3-D, but did not elaborate it any further. While curve segmentation in 2-D may be sufficient for many applications, it has some disadvantages:

- Features which appear in one image only are also segmented, although they do not lend themselves to edges in the object space.

- The segmentation of edges performed individually in both images, does not necessarily produce corresponding breakpoints. Therefore, the identification of the same feature in both images becomes a nontrivial task.
- Although straight lines in the object space are also straight lines in all projections, the converse does not hold. Circular arcs appear as elliptic arcs, which are more difficult to detect.

To avoid these disadvantages we propose to segment the 3-D curves in the object space. In the next section we present two methods for segmenting a 3-D curve into its basic primitives.

2 METHODS

Here we describe two methods suitable for segmenting 3-D curves. These methods are not necessarily the best curve segmentation methods known, rather they demonstrate the concept of segmentation in 3-D space. The first method is a 3-D version of a split-and-merge concept, based on the offset from a straight line. In this method, the curve is segmented into straight lines only. The second method is an extension of the $\psi - s$ concept (see Li and Schenk, 1991 for a 2-D description) into 3-D. With this method, straight lines and circular arcs are detected.

The input for both methods are lists of densely spaced points of 3-D edges. It should be noted that the points are not evenly spaced. They are represented by real 3-D coordinates. We are presently investigating another representation, where the edge points are resampled into a 3-D discrete space (voxels).

2.1 Split-and-merge method

As the name indicates, the split-and-merge method consists of the two phases split and merge. In the split phase, the input data is segmented to assure that each segment fulfills a certain condition. In our case, the condition is that all the points contained in a segment are likely to correspond to a straight line. In the merge phase, redundant breakpoints that have been produced during the split phase are eliminated.

The criterion for deciding whether a group of small line fragments can be represented as a longer straight line is the maximum offset. We have chosen the maximum offset and not a least-squares criterion as suggested in (Pavlidis and Horowitz, 1974). The reason is that the computational cost for applying the least-squares criterion is much higher than the one for the maximum offset, especially in the 3-D case. In addition, we only perform one split and one merge phase because the initial segmentation criterion is strong. A refinement of the breakpoints and a fitting of a straight line to a list of points can be performed once the segmentation has been achieved. In general, the offset criterion is superior to other criteria, since it is not very noise sensitive. Other criteria, such as the orientation of a line, are quite sensitive to noise, as we will see in the next section.

Let us now define the maximum offset criterion. Consider a string of n small line fragments $l_1 \ldots l_n$ which are formed by a corresponding set of densely spaced points $p_0 \ldots p_n$. This string of fragments can be considered as one longer straight line if the distance from each point $p_1 \ldots p_{n-1}$ to the straight line connecting p_0 and p_n does not exceed a predefined threshold value.

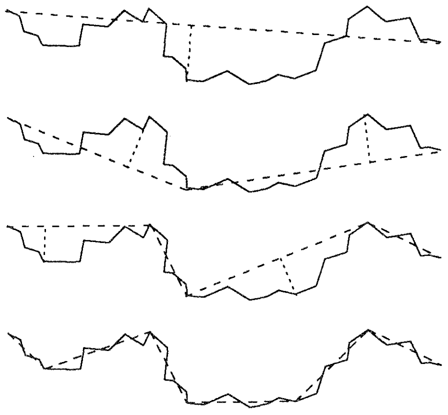

Figure 1: Splitting a curve according to the maximum offset criterion

The following pseudo code describes the split phase of the split-and-merge method. It processes an edge $E = \{p_1 \ldots p_n\}$ or a part of it. The input to the function in the first call is the indices 1 and n of the first and last points of the edge. The function works recursively, and in general its input is the indices of the first and last points of a sub-edge. The set s (which is a global parameter) is initialized to contain the numbers 1 and n.

Split(f,l)

```
1. $m := -1$
2. $\forall i,\ f \le i \le l$
    2.1 $o :=$ the offset of point $p_i$ from the line $\overline{p_f p_l}$
    2.2 if $o > m$ then $m := o$; $t := i$
3. if $m >$ MAXOFFSET
   then $s := s \cup \{t\}$; Split($f,t$); Split($t,l$)
```

Once the above algorithm terminates, s contains an unsorted list of indices of the potential breakpoints. The merge algorithm, as specified in the following pseudo code, attempts to merge two neighboring segments, according to the same maximum offset criterion described earlier.

Merge()

```
1. Sort the $s$ list by ascending order
2. $k :=$ the number of breakpoints in $s$
3. $\forall i,\ 2 \le i \le k-1$
    3.1 $o :=$ the offset of point $p_{s[i]}$ from the line $\overline{p_{s[i-1]} p_{s[i+1]}}$
    3.2 if $o >$ MAXOFFSET
        then $s := s - \{s[i]\}$; $k := k - 1$
```

Figure 1 demonstrates a 2-D curve splitting. The 3-D case is similar, except for the calculation of the offsets. These are calculated in a 3-D coordinate system.

Another aspect is the analysis of the segmented line. Although the split-and-merge method aims at segmenting straight lines, some lines cannot be classified, be it because of noise, short segments or simply because no straight line segments are present.

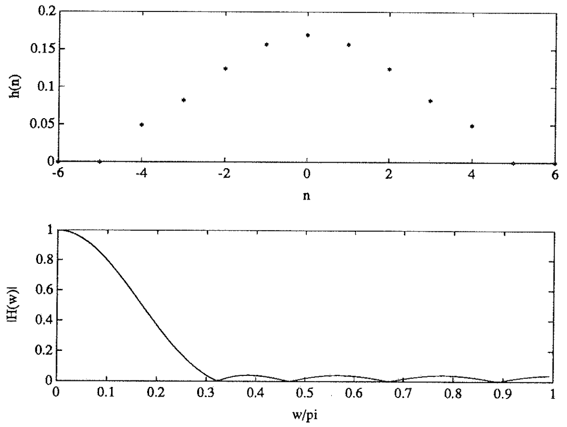

Figure 2: Low-pass filter for the $\psi - s$ curve: impulse response and magnitude of the frequency response

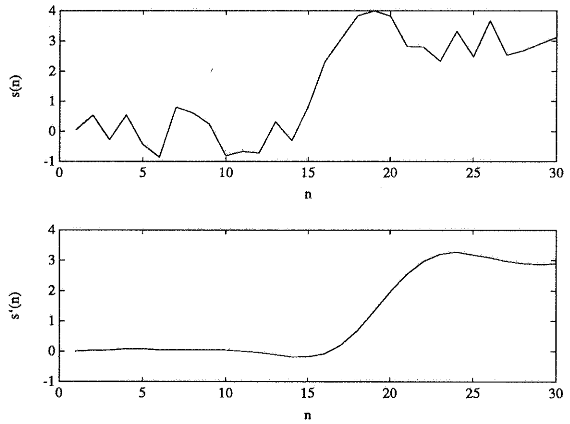

Figure 3: An example of filtering a 1-D sequence by the equiripple low-pass filter

2.2 Segmentation in $\psi - s$ domain

In order to easily detect circular arcs in addition to straight lines, the $\psi - s$ domain can be used. In this domain, straight lines appear as horizontal lines, and circular arcs as arbitrary straight lines. Since both straight lines and circular arcs appear in the $\psi - s$ domain as straight lines, we can use the split-and-merge algorithm described in section 2.1 to segment the $\psi - s$ curve and get as a result both the straight lines and the circular arcs.

As described earlier, the input is a list of points in a 3-D continuous coordinate system. Since this input is derived from a discrete 2-D representation, noise effects that were produced during the scanning of the original aerial photographs cannot be avoided. The $\psi - s$ method is very sensitive to noise. A point in the spatial domain that is displaced by approximately the distance between two neighboring points will cause $\sim 45°$ "offset" in the $\psi - s$ curve. Therefore, the original data should be filtered by a low-pass filter. Since the breakpoints we try to detect are also high frequency phenomena, they will be affected too. In order to compromise between noise removal and information preservation, a filter with few coefficients should be used. We used the Parks-McClellan equiripple algorithm (Rabiner et. al, 1975) to design such a filter. The equiripple method minimizes the maximum error between an ideal (infinite length) low pass filter and a filter with a truncated number of coefficients. By this, an optimal filter can be achieved for a given set of specifications. This set includes the cutoff frequency, the transition band, a weighting function for the errors in the pass and the stop bands, and the number of coefficients. Recursive or nonlinear filters (Wuescher and Boyer, 1991) are alternate solutions to the filtering problem. The impulse and frequency responses of the filter are shown in figure 2. Figure 3 shows a noisy 1-D sequence before and after filtering. In the case of 3-D edges, all three coordinates are convolved separately with this filter.

The $\psi - s$ domain in 2-D space consists basically of a plot of the orientation (ψ) versus length (s) of the original spatial curve. In this representation, the slope of the line corresponds to the curvature of the original curve. Therefore, it can be easily shown that a straight line in the spatial domain appears as a horizontal line (parallel to the s axis) in the $\psi - s$ domain, and a circular arc (which has a constant curvature) appears as an arbitrary straight line. The $\psi - s$ curve for a nonanalytical spatial curve is constructed by computing the directions between points. In order to overcome some residual noise effects, we calculate the direction at a certain point not between the point and its neighbor, but between its predecessor and successor. In cases of more extreme noise residuals, a larger interval can be used for calculations.

In order to segment the $\psi - s$ curve, discontinuities should appear only at breakpoints. An artificial discontinuity is present when the original curve orientation goes from 360° to 0° or vice versa. Hence, after representing the curve in the $\psi - s$ domain, this artificial discontinuity is eliminated. The procedure is described by the following pseudo code, where c is a parameter which compensates for the discontinuity:

Discontinuity_elimination1()

```
1. let $p_1 \ldots p_n$ be the list of points of the $\psi - s$ curve
2. $c := 0°$
3. $\forall i,\ 2 \leq i \leq n$
    3.1 $\psi_i := \psi_i + c$
    3.2 if $|\psi_i - \psi_{i-1}| \geq 180°$ then
        • if $\psi_i > \psi_{i-1}$
          then $\psi_i := \psi_i - 360°$; $c := c - 360°$
          else $\psi_i := \psi_i + 360°$; $c := c + 360°$
```

With this procedure no changes in orientation of more than 180° will occur.

We have extended the $\psi - s$ approach to 3-D. A horizontal angle α and a vertical angle ϕ are used to express the spatial direction. Again, a straight line in the spatial domain appears as a line which is parallel to the distance axis of the $\psi - s$ domain. A circular arc, contained in an arbitrary plane in the 3-D space, appears as an arbitrary straight line in the 3-D $\psi - s$ space.

Special attention must be paid when the tangent of a circular arc at a certain point becomes vertical. This situation is described by the following:

$$|\alpha_i - \alpha_{i-1}| \approx 180°$$

$$|\phi_i| \approx |\phi_{i-1}| \approx 90°$$

$$\phi_i \approx \phi_{i-1}$$

Careful examination reveals the gradient of ϕ changes its sign leading to a discontinuity of the vertical angle. In

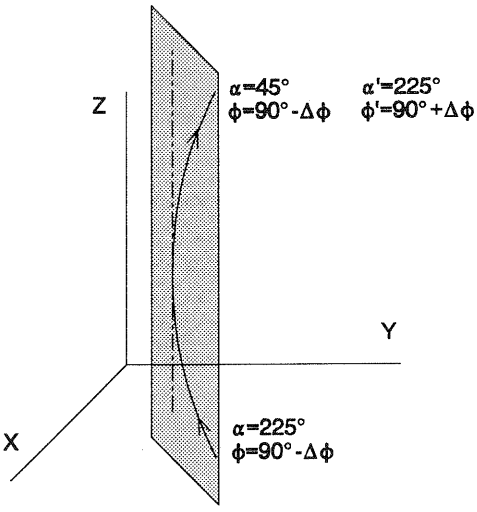

Figure 4: Compensation for an artificial discontinuity

order to eliminate this discontinuity problem, the following procedure is added to the transformation of a spatial curve into the $\psi - s$ domain:

Discontinuity_elimination2()

```
1. let $p_1 \dots p_n$ be the points of the $\psi - s$ curve, $c$ be a
   compensation factor for the horizontal angle, $z$ be a
   zero elevation base for the vertical angle, and $s$ a sign
   factor
2. $c := 0°$; $z = 0°$; $s = 1$
3. $\forall i,\ 2 \leq i \leq n$
   3.1 $\alpha_i := \alpha_i + c$; $\phi_i := \phi_i + z * s$
   3.2 if $(|\alpha_i - \alpha_{i-1}| \approx 180°)$ &
       $(|\phi_i - z| \approx |\phi_{i-1} - z| \approx 90°)$ &
       $(\phi_i \approx \phi_{i-1})$ then
     3.2.1. if $\alpha_i > \alpha_{i-1}$
            then $c := c - 180°$; $\alpha_i := \alpha_i - 180°$
            else $c := c + 180°$; $\alpha_i := \alpha_i + 180°$
     3.2.2. if $((\phi - z \approx 90°)$ & $(s = 1))$ or
            $((\phi - z \approx -90°)$ & $(s = -1))$
            then $z := z + 180°$; $\phi_i := \phi_i + 180°$
            else $z := z - 180°$; $\phi_i := \phi_i - 180°$
```

Figure 4 shows a case where the compensation is necessary. α' and ϕ' are corrected angles.

The disadvantage of this approach is that the restoration of the original spatial curve from the $\psi - s$ curve is no longer possible. However, the conversion into the $\psi - s$ domain is done for approximating the location of the breakpoints on the curve. We can certainly store the indices of the found breakpoints, go back to the original spatial domain, and segment the original curve according to these breakpoints.

Once we have a $\psi - s$ curve which does not contain representation related discontinuities, the simplest way to segment it into straight lines is by the split-and-merge algorithm described in section 2.1. The result of this operation is a list of straight lines in the $\psi - s$ domain. Each of these straight lines is examined and classified into one of three spatial domain categories, namely, straight line, circular arc or "other," i.e., natural lines or noise effects, according to the following order of criteria:

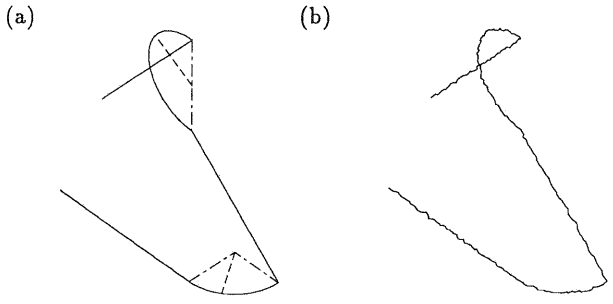

Figure 5: Synthetic data: (a) clean; (b) noisy

- If the line is shorter than a predefined threshold value, it is classified as "other."
- If the slope of the line is less than a predefined threshold value, it is classified as straight line.
- The radius, arrow and angle of a circular arc are estimated from the slope and first and last points of a $\psi - s$ segment. If these parameters are within a predefined interval, the segment is classified as a circular arc.
- In other cases, the segment is classified as "other."

3 EXPERIMENTAL RESULTS

Both the split-and-merge and the $\psi - s$ methods were implemented and tested with synthetic and real data. Not all the experiments have been completed yet, leading to more results with real data.

The synthetic data were produced by combining a set of straight lines and circular arcs in 3-D space, which were then corrupted by noise that was produced by a pseudo-random number generator. The magnitude of the noise was chosen in a way that mimics the behavior of real data. Figure 5 shows the clean and noisy synthetic data as 3-D curves. The real data were taken from the results of the matching process, consisting of lists of 3-D points. Figure 6a shows the left image of the stereo pair which was used for the production of these edges. The 3-D edges are shown in figure 6b in an orthogonal projection.

3.1 Split-and-merge results

The split-and-merge algorithm was implemented according to the description in section 2.1. In general, the offset threshold can be derived directly from the scale and the scanning resolution of the aerial images, and it should be larger than the size of a pixel in object space. The aerial photographs we used have a scale of approximately 1/4000, and the scanning pixel size is approximately 60 μm. Therefore, a pixel size in object space is $\sim 0.25\ m$. We selected a value which is slightly higher, taking into account also other noise effects. The threshold was the same for both the split and the merge phases of the algorithm.

Synthetic data: Testing the split-and-merge procedure on the synthetic data did not present any troubles in the segmentation, just as we anticipated. The straight lines

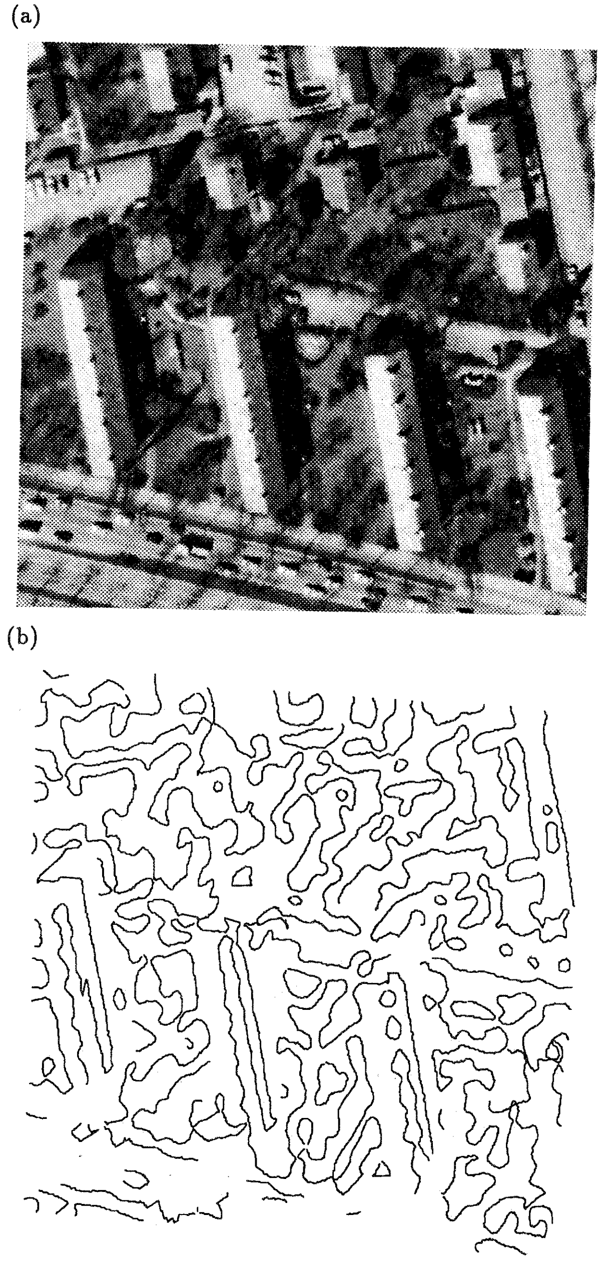

Figure 6: Real data: (a) left stereomate; (b) 3-D edges in orthogonal projection

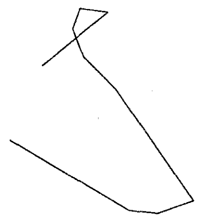

Figure 7: Results of applying the split-and-merge method to the synthetic data

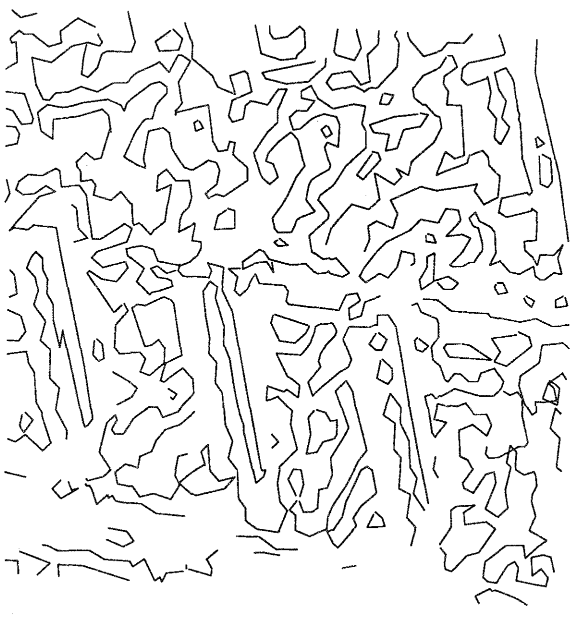

Figure 8: Results of applying the split-and-merge method to the real data (orthogonal projection)

were extracted completely, and the circular arcs were segmented into small straight lines. The results are shown in figure 7

Real data: The results of the split-and-merge segmentation for the real data are shown in figure 8 as an orthogonal projection of the 3-D segments received. These results are very encouraging. Many straight segments were detected. The noisy parts of the curves, which are interpreted as such also by a human observer, remained unchanged. A comparison between the results and the image shows correspondence between straight lines and man-made features.

3.2 $\psi - s$ results

The $\psi - s$ segmentation algorithm was implemented according to the description in section 2.2. The selection of threshold values is more crucial than it is for the split-and-merge case. The main reason for this problem is the fact that we deal with angular parameters, while the real physical perturbations are linear. Therefore, the threshold value for a certain line length will not necessarily be suitable for other lengths. Despite this, we used values which are acceptable for the synthetic data, as described below. The offset threshold for the $\psi - s$ curve was set to 10. We also limited the accepted circular arcs radii to the interval $2 - 200$ m. We have not limited the arc angle and arrow at this stage.

Synthetic data: The results of executing the $\psi - s$ algorithm with the synthetic data are presented in a 3-D view in figure 9. The results need some explanations.

1. Longer segments (either straight lines or circular arcs) were segmented into shorter ones. However, it can be seen that most of the segments were classified correctly.

2. Small segments, which were characterized as noise effects were created near the discontinuity points.

The phenomenon of breaking an expected segment into a small number of shorter segments can be resolved in the

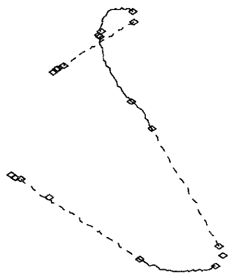

Figure 9: Results of applying the $\psi - s$ method to the synthetic data: straight segments are represented by dashed lines, circular arcs by solid lines, and noise effects are not presented. The breakpoints are represented by squares.

spatial domain. For example, the second arc in the synthetic example was detected as two smaller arcs and a noise segment. In the spatial domain, these shorter arcs can be combined into a larger arc by applying a least-squares adjustment, and eliminating possible blunders. The noise effects near the breakpoints can be resolved as well. If we eliminate any "short" phenomena, we can intersect neighboring longer phenomena, and by that close the gaps produced by the elimination of the short segments.

Real data: Experiments with the $\psi - s$ method were also performed with real data. We found that the limitations of the $\psi - s$ method, in terms of predefined thresholds, are quite critical. The selection of the threshold values is application dependent, i.e., the approximate size of features should be known.

4 SUMMARY AND CONCLUSIONS

The paper describes curve segmentation in 3-D object space. Although the two methods described for that purpose are not necessarily the best available segmentation methods, the results are encouraging and show that 3-D segmentation is possible.

The split-and-merge method segments the data into straight lines only. Circular arcs are segmented into a list of short straight line segments. The offset criterion used reduces the sensitivity to noise. In other words, the split-and-merge method is quite robust and detects line segments even if they are very noisy.

The $\psi - s$ method offers the advantage of representing circular arcs as straight lines. This property allows detection of circular arcs by using the split-and-merge approach. However, determining threshold values becomes a crucial issue. Due to noise effects, it is dependent on the lengths of lines to be classified. The noise is reduced significantly by a proper filtering of the original data. However, filtering also blurs the breakpoints. Current research focuses on a 3-D Freeman code (Freeman, 1974) representation. That is, the object space is discretized, thus reducing some of the noise caused by the scanning process.

The experience gained leads to the following conclusions:

1. Since the $\psi - s$ method allows easy detection of circular arcs, it can be used for a rough segmentation of the 3-D curve into straight lines and circular arcs. Once such approximations exist, other methods can be used to refine the segmentation.
2. Other segmentation methods should be investigated and eventually extended to 3-D.

The segmentation of the 3-D curves is an important clue for man-made features, which are usually composed of 3-D straight lines and other regular curves that provide information which is much more explicit than the original densely spaced points resulting from stereo matching.

References

[1] Ballard, D. H. and C. M. Brown, 1987. *Computer Vision.* Prentice-Hall, Inc., Englewood, NJ.

[2] Cho, W., M. Madani, and T. Schenk, 1992. Resampling digital imagery to epipolar geometry. In *International Archives of Photogrammetry and Remote Sensing.*

[3] Fischler, M. A. and R. C. Bolles, 1986. Perceptual organization and curve partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 8(1):100–105.

[4] Freeman, H., 1974. Computer processing of line-drawing images. *Computing Surveys,* 6(1).

[5] Grimson, W. E. L., 1985. Computational experiments with a feature-based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 7(1):17–34.

[6] Grimson, W. E. L., 1989b. *Object Recognition by Romputer: The Role of Geometrical Constraints.* MIT Press, Cambridge, Massachusetts; London, England.

[7] Grimson, W. E. L., 1989a. On the recognition of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 11(6):632–642.

[8] Grimson, W. E. L. and T. Pavlidis, 1985. Discontinuity detection for visual surface reconstruction. *Computer Vision, Graphics and Image Processing,* 30:316–330.

[9] Li, J. C. and T. Schenk, 1991. Stereo image matching with sub-pixel accuracy. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Photogrammetry and Primary Data Acquisition,* pp. 228–236.

[10] Marr, D. and E. C. Hildreth, 1980. Theory of edge detection. In *Proceedings, Royal Society London, B 207,* pp. 187–217.

[11] Pavlidis, T. and S. L. Horowitz, 1974. Segmentation of plane curves. *IEEE Transactions on Computers,* 23(8):860–870.

[12] Rabiner, L. R., J. H. McClellan, and T. W. Parks, 1975. FIR digital filter design techniques using weighted Chebyshev approximations. In *Proceedings, IEEE 63,* pp. 595–610.

[13] Ramer, U., 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing,* 1:244–256.

[14] Schenk, T., 1989. Application of the zero-crossings method to digital mapping. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Surveying and Cartography*, pp. 325–333.

[15] Schenk, T., Li J. C., and C. Toth, 1991. Towards an autonomous system for orienting digital stereopairs. *Photogrammetric Engineering and Remote Sensing*, 57(8):1057–1064.

[16] Schenk, T., J. C. Li, and C. Toth, 1991. On using warped images in a hierarchical approach to reconstruct visible surfaces from aerial imagery. In *Proceedings, IEEE Computer Vision and Pattern Recognition Conference.*

[17] Schenk, T. and C. Toth, 1992. Conceptual issues of softcopy photogrammetric work stations. *Photogrammetric Engineering and Remote Sensing*, 58(1):101–110.

[18] Stefanidis, A., P. Agouris, and T. Schenk, 1991. Aspects of accuracy in automatic orientation. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Photogrammetry and Primary Data Acquisition*, pp. 334–343.

[19] Wuescher, D. M. and K. L. Boyer, 1991. Robust contour decomposition using a constant curvature criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):41–51.

[20] Zong, J., Li J. C., and T. Schenk, 1991. Application of förstner interest operator in automatic orientation systems. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Photogrammetry and Primary Data Acquisition*, pp. 440–448.

[21] Zong, J. and T. Schenk, 1992. Aerial image matching pased on zero-crossings. In *International Archives of Photogrammetry and Remote Sensing.*

ACKNOWLEDGEMENTS

Funding for this research was provided by the NASA Center for the Commercial Development of Space Component of the Center for Mapping at The Ohio State University.