

# Energy-Based Segmentation of Very Sparse Range Surfaces

Terrance E. Boulton and Mark Lerner  
Department of Computer Science, Columbia University  
NYC NY 10027.  
tboulton@cs.columbia.edu lerner@cs.columbia.edu

CUCS-422-89

## Abstract

This paper describes a new segmentation technique for very sparse surfaces which is based on minimizing the energy of the surfaces in the scene. While it could be used in almost any system as part of surface reconstruction/model recovery, the algorithm is designed to be usable when the depth information is scattered and very sparse, as is generally the case with depth generated by stereo algorithms. We show results from a sequential algorithm that processes laser rangefinder data or synthetic data. We then discuss a parallel implementation running on the parallel Connection Machine.

The idea of segmentation by energy minimization is not new. However, prior techniques have relied on discrete regularization or Markov random fields to model the surfaces to build smooth surfaces and detect depth edges. Both of the aforementioned techniques are ineffective at energy minimization for very sparse data. In addition, this method does not require edge detection and is thus also applicable when edge information is unreliable or unavailable. Our model is extremely general; it does not depend on a model of the surface shape but only on the energy for bending a surface. Thus the surfaces can grow in a more data-directed manner.

The technique presented herein models the surfaces with reproducing kernel-based splines which can be shown to solve a regularized surface reconstruction problem. From the functional form of these splines we derive computable bounds on the energy of a surface over a given finite region. The computation of the spline, and the corresponding surface representation are quite efficient for very sparse data. An interesting property of the algorithm is that it makes no attempt to determine segmentation boundaries; the algorithm can be viewed as a classification scheme which segments the data into collections of points which are "from" the same surface. Among the significant advantages of the method

is the capacity to process overlapping transparent surfaces, as well as surfaces with large occluded areas.

## 1 Segmentation: Introduction and Background

Segmentation is one of the most pervasive and most difficult problems in computer vision. It rears its ugly head in such subareas as: edge/region detection, motion detection, determination of textures, shape-from-X (for almost all X), calculation of disparity fields (stereo matching), model recovery, surface reconstruction and medical imaging. Unfortunately, the segmentation problem in each of these areas will not, in general, be solvable by the same techniques. One reason for the failure of the methods to extend to different the segmentation problems in the various subareas is because the assumptions about the data vary dramatically:

- In edge/region detection, the data is the intensity values of the image irradiance and assumptions used for segmentation must be related to the process of image formation.
- In motion detection, the data can be either spatio-temporal intensity images or spatio-temporal surface information, and the segmentation assumptions must be related either to the flow of intensities as objects/self undergo motion (for spatio-temporal intensities images) or to object models and the physics of motion of said objects.
- In surface reconstruction, the assumptions used for segmentation must be related to models of world surfaces.
- In the recovery of disparity fields, assumptions must be tied to either a model of disparity fields, or a combination of models of world surfaces and the pair of image formation equations used to obtain the disparity field.
- etc. ...

Of these, the segmentation tasks in surface reconstruction, disparity field recovery and certain classes of motion detection problems, have been approached using segmentation coupled with recovery using an energy-based smoothness assumption, for example see [Terzopoulos 84], [Hoff and Ahuja 85], [Anandan and Weiss 85], [Blake and Zisserman 86], [Chou Brown 88], [Kanade 88]. In each of these cases, the attempt at segmentation can be roughly described as follows:

- Step 1** Do a initial smoothness-based reconstruction (this is generally a minimal energy surface or configuration).
- Step 2** Mark those parts of the reconstruction which are "not locally smooth" (generally with a gradient like operator) as discontinuities.
- Step 3** Adjust the reconstruction mechanism to deal with the newly marked discontinuities and go to Step 1.

Other segmentation approaches for surface reconstruction have incorporated smooth measures implied by volumetric models, for example, see [Pentland 86],

[Bajcsy and Solina 87], [Rao, Nevatia and Medioni 87], or local smoothness properties such as planarity or curvature consistency, see [Besel and Jain 86].

## 2 Motivation and a Different Formulation

This paper proposes a different model of segmentation which has some fundamental differences in the formulation of the problem. This section discusses this model, and also provides some motivation for it. In defining this model, the section introduces the energy-based segmentation approach wherein the energy of reconstructed surface(s) is *directly* used to segment the data.

The traditional view of the surface segmentation problem is one of determining the "discontinuity boundaries" in surface depth, surface orientation and/or surface curvature. This approach usually requires some reconstruction of the surface, which is related to an *a priori* chosen measure of surface energy. Unfortunately, in order to correctly generate a surface reconstruction, knowledge of data segmentation is generally required. This results in a difficult chicken-and-egg problem. Thus, researchers may assume the scene consists of a specific class of surfaces (such as planar or convex). To make matters worse, the quality of the reconstruction in the neighborhood of an unmarked (i.e. as yet, undetected) discontinuity is generally poor. Thus the localization of the discontinuity of iterative reconstruct/segment approaches, see e.g. [Terzopoulos 84] or [Hoff and Ahuja 87], will be questionable. Moreover, data from scenes with transparent surfaces cannot easily fit to these models.

A second shortcoming of the traditional approach is that it will require considerable post processing to handle extended multiply connected objects (say an object behind a picket fence) and may never be able to handle transparent surfaces where locally there are only a few points on any one surface.

A final remark about traditional segmentation is related to the definition of "boundaries". It is well known that the perceived "boundaries" of surfaces in depth share many characteristics with subjective contours, see [Julesz 71], [Marr 81]. This suggests that a definition of "boundaries" in depth might be accomplished by some secondary processing which is shared with "boundary" detection from other visual modalities. Therefore the energy-based method of this paper can be combined with a secondary boundary-detection process to obtain both the shape and the outline of each surface.

To accommodate the above mentioned problems, this paper proposes that segmentation of 3D information should not attempt to determine boundaries. Rather, the segmentation should simply classify points as belonging to the same surface. The determination of boundaries will be relegated to some secondary process which is not discussed here. Of course this view cannot be taken too far, as there are limits both to the number of possible "transparent" surfaces and the number of times a background object is blocked by foreground objects. Psychological experiments have already demonstrated that such limits exist in the human visual system.

As mentioned above, what is desired is some measure which can be applied

to groups of points to determine if they are part of the same physical surface. This section presents one such measure, surface bending energy, and discusses its appropriateness. The authors acknowledge that other measures might be used and the end of this section touches upon some of these alternatives.

Segmentation has been extensively studied in the context of image segmentation, and one might wonder if the algorithm herein is new / applicable to that domain. The crucial part of our algorithm is the use of "surface energy" to heuristically (though reliably) determine if two points are part of the same "extended region". Unfortunately, such measures have proven illusive for intensity images.

In computer vision, as well as other domains, researchers have used minimal surface bending-energy as an assumption to aid in surface recovery, for example see [Grimson 81], [Franke 82], [Terzopoulos 84], [Wahba 84], [Hoff and Ahuja 85], [Choi and Kender 85], [Lee 85], [Blake and Zisserman 86], [Boult 86]. Bending energy thus seems a natural choice for the "measure" to determine if a group of points belong to the same surface. The bending energy of a thin-plate surface  $f$  is given by:

$$\left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \cdot \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right) dx dy \right\}^{\frac{1}{2}} \quad (1)$$

Of course, the use of bending energy can only be a partial basis for a practical measure for segmentation. Other issues that must be also be addressed include:

- What is the allowable class of functions for bending energy.
- How to determine the effect of surface size, or the area over which the energy is measured.
- What is the relationship between the energy and the number of data points and surface area.
- How to set the threshold for separation of a group, or alternatively to define the tradeoff between the number of surfaces and sum of the energies of these surface.
- When is the energy of a collection of points "too big".
- How does one locate the fewest number of "culprit" point(s) in the group, i.e., the credit assignment problem.
- What is the relationship between "depth" discontinuities and "orientation" orientations, and how do these characteristics effect the energy measure discontinuities.

### 3 An Energy-based Segmentation Algorithm

This section describes one way to realize an energy based segmentation algorithm given that one accepts the assumption of minimizing bending energy.

However, the algorithm is easily modified to handle many related measures such as those described in [Boult 86] and [Boult 87]. The section describes the mathematical background of the algorithm. The discussion of advantages and disadvantages of this approach are relegated to a separate section following this one.

The algorithm constructs initial approximations of the surfaces from the local data cluster 3-space.\* These approximate surfaces are updated by subsequent processing. The algorithm then heuristically determines which point to add next (more below) and points are added to a surface as long as the addition does not cause the energy (see below) of said surface to exceed a certain threshold. If the surface cannot accept any remaining points, a "new" surface is created and the process repeats itself until all data points have been processed. A variation of the algorithm develops several surfaces in parallel by placing a point on several surfaces that can accept it without exceeding the threshold.

In its basic form, each pass of the algorithm computes the surface and corresponding energy that would result if each remaining point were separately added to the current surface. The system then adds the point which would cause the minimal rise in surface bending energy, assuming it would not exceed the specified threshold. Because of the monotonic and commutative nature of the energy measure, this approach will generally find the surface of minimal energy given the starting basis and the threshold, though it can be computationally expensive. This expense is being addressed in two ways. First, through a parallel implementation. Second, we recognize that the true minima is not always necessary, and we develop heuristics that develop the segmentation at lower cost.

The discussion of algorithmic details can be divided into seven smaller conceptual components which appear as separate subsections. These components are:

1. Definition of the model of world surfaces.
2. Definition of the reproducing kernel-based spline which is used to reconstruct the surfaces.
3. Definition of the energy measure.
4. Calculation of bounds on the energy of a reproducing kernel-based spline surface.
5. Heuristics used for (a) basis point selection, (b) point selection, and (c) culprit point selection, to remove some points and decrease the surface energy.
6. Method to merge similar surfaces into one surface.
7. A short discussion of the ongoing parallel implementation.

---

\*The size of the cluster is user definable from 4-30 data-points. In general, the number must be at least 1 more than the size of the dimension of the null space of the energy measure. A number of heuristics have been developed to pick the basis points. These generally select an initial point along with a the subset of its neighbors in 3 space that produce the lowest energy surface.

### 3.1 Definition of the Model of World Surfaces

The assumed model of world surface is intimately related to techniques for regularized surface reconstruction, see [Boult 86]. An important set of these classes can be parameterized formally as those functions with their  $m^{th}$  derivative in  $H^\eta$ , where  $H^\eta$  is the Hilbert space of functions such that their tempered distributions  $\nu$  in  $\mathbb{R}^2$  have Fourier transform  $\tilde{\nu}$  that satisfy

$$\int \int_{\mathbb{R}^2} \left( |\tau|^{2\eta} \cdot |\tilde{\nu}(\tau)|^2 d\tau \right) < +\infty.$$

This class of functions, referred to as  $D^m H^\eta$ , is equipped with the  $m^{th}$  Sobolev semi-norm,

$$\|\cdot\|_{D^m} = \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \sum_{i+j=m} \binom{m}{i} \left( \frac{\partial^m f}{\partial x^i \partial y^j} \right)^2 \right) dx dy \right\}^{\frac{1}{2}} \quad (2)$$

which, if  $1 > \eta > 1 - m$ , results in a semi-Hilbert space. Note that if one chooses  $m = 2, \eta = 0$ , then using the properties of Fourier transforms, the above definitions yield exactly the space  $D^2 L^2$  which was used by Grimson and Terzopoulos.

Let us now give an intuitive definition of these classes of functions. First, note that the spaces of “functions” assume the existence of the  $m^{th}$  derivative of the function, in the *distributional sense*. This means, roughly, that the  $m^{th}$  derivative of the functions exists except on sets of measure of zero (e.g., at isolated points or lines). Then the classes  $D^m H^0$ , which are also known as  $D^m L^2$ , simply assume that the power of these functions is bounded. For the classes  $D^m H^\eta, \eta > 0$  we have that the square of the spectrum of the derivatives goes to zero (as the frequency goes to  $\infty$ ) faster than a specified polynomial of the frequency. Thus, these functions have less high frequencies and are “smoother” than functions which simply have  $m$  derivatives. For the classes  $D^m H^\eta, \eta < 0$  we see that the spectrum of the derivatives is bounded away from zero, and that as the frequency goes to  $\infty$ , the derivatives go to infinity no faster than a given polynomial. Thus, these functions have less low frequencies and are less “smooth” than most functions with  $m$  derivatives.

In the work reported here, we consider the class  $D^2 H^{\frac{1}{2}}$  which, intuitively is the space of functions which are smooth (almost everywhere) up to derivatives of order approximately 1.5, i.e., they are significantly smoother than membrane surfaces but are not as smooth as thin-plate splines. The motivation for this choice of functions (i.e., this “intermediate” level of smoothing) is supported by the results of [Boult 87].

### 3.2 Definition of Reproducing Kernel-Based Spline

An essential ingredient of the current algorithm, at least from the point of view of efficient serial implementation, is the use of the reproducing kernel-based

spline reconstruction as described in [Boult 86]. This section introduces some aspects of that algorithm necessary for later discussions.

We do not choose to interpolate the data, instead we follow the "regularization" approach of minimizing a smoothness term (the  $m^{th}$  Sobolev semi-norm) a weighted sum of squares of the distance of the surface from the data, i.e. we find the surface from our class of surfaces which minimizes:

$$\lambda \cdot \sum_{i=1}^n \frac{(\sigma(x_i, y_i) - z_i)^2}{\delta_i} + \|\sigma\|_{D^m} \quad (3)$$

where the data  $z_i$  at point  $(x_i, y_i)$ ,  $i = 1, \dots, n$  is assumed to be on one surface. The *global smoothing parameter*,  $\lambda$ , should depend on the overall error in the initial data, and the factors  $\delta_i$  allow for individual points to have greater "noise"; the factor  $\lambda$  effects the overall tradeoff between surface smoothness (as measured by the norm  $\|\cdot\|_{D^2}$ ) and the fidelity to the data points  $z_i$  while the factor  $\delta_i$  effects the contribution of a single data point so as not to penalize the surface as much (or to penalize it more, depending on the value of  $\delta_i$ ) for not closely approximating the data at that point. Techniques for choosing these parameters have been discussed by other researchers, see [Bates and Wahba 82].

One solution to the above reconstruction problem is a reproducing kernel-based spline. It can be shown, see [Meinguet 83], that for the above model of world surfaces, the appropriate reproducing kernel here is

$$K(x, y; u, v) = \gamma((x - u)^2 + (y - v)^2)^{\frac{3}{2}}$$

for some constant  $\gamma$

Given the above kernel, the spline (i.e., the reconstruction of the  $2\frac{1}{2}$ D sketch) which approximated the data

$$z = z_1, \dots, z_k = \{f(x_1, y_1), f(x_2, y_2), \dots, f(x_k, y_k), \quad i = 1, \dots, k\}$$

can be developed as:

$$\sigma_s = \sum_{i=1}^k \alpha_i K(x, y; x_i, y_i) + \sum_{i=1}^3 \beta_i p_i(x, y) \quad (4)$$

where  $p_1(x, y) = 1$ ,  $p_2(x, y) = x$ ,  $p_3(x, y) = y$ . The constants  $\alpha_i$  and  $\beta_i$  are the solution to the system of linear equations:

$$\begin{array}{ccccccc}
A_{1,1} & \dots & A_{1,k} & B_{1,1} & B_{1,2} & B_{1,3} & z_1 \\
\vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\
A_{k,1} & \dots & A_{k,k} & B_{k,1} & B_{k,2} & B_{k,3} & z_k \\
C_{1,1} & \dots & C_{1,k} & D_{1,1} & D_{1,2} & D_{1,3} & 0 \\
C_{2,1} & \dots & C_{2,k} & D_{2,1} & D_{2,2} & D_{2,3} & 0 \\
C_{3,1} & \dots & C_{3,k} & D_{3,1} & D_{3,2} & D_{3,3} & 0
\end{array}$$

where

$$\begin{aligned}
A_{i,i} &= \delta_i / (\alpha_i \lambda 8\pi) \quad i = 1, \dots, k; \\
A_{i,j} &= \alpha_j (K(x_j, y_j; x_i, y_i)), \quad i = 1, \dots, k \quad j = 1, \dots, k \quad i \neq j; \\
B_{i,j} &= C_{j,i} = \beta_j p_j(x_i, y_i) \quad i = 1, \dots, k, \quad j = 1, \dots, 3; \\
\text{and} \quad D_{i,j} &= 0 \quad i = 1, \dots, 3 \quad j = 1, \dots, 3;
\end{aligned}$$

(5)

The important properties of the above solution to the surface reconstruction problem are:

1. The algorithm is efficient for very sparse data (anything more than 3 non-collinear points will do, and the fewer the number of points, the faster the surface can be computed).
2. The surface is defined by the solution to a linear system which depends only on the location of the data. If the solution to this system can be updated quickly, the surface can also be updated quickly.
3. The surface is given in a functional form, thus the evaluation of derivatives is trivial, and bounds on the energy of the surface can be computed analytically.
4. The surfaces are independent of the "boundaries" of discontinuities, and depend only on the data values. However, the actual surface will change if the number/value of data points on the boundary are changed.

**3.2.1 Short discussion of the updatable QR algorithm** This section briefly discusses the way the algorithm updates the linear system to allow for an efficient update of the surface for serial implementations. The algorithm begins by doing a QR decomposition of the initial linear system. Then, using Givens rotations, the algorithm can allow for the addition or deletion of any row/column (in fact, it can handle any rank one modification). The computation of the initial QR decomposition for  $k$  data points requires time  $O(\frac{1}{3}k^3)$ <sup>1</sup>. The addition/deletion then costs  $O(k^2)$ , and the recomputation of the solution with the new system costs  $O(k^2)$ . The algorithm is numerically stable which is important since the condition number for reproducing kernel-based spline problems can be moderately large.

<sup>1</sup>While the decomposition could in fact be computed by adding every row one at a time, this doubles the cost of the decomposition, and may slightly effect the quality of the solution.



A secondary advantage of using the QR decomposition is that changing the values of the data (while leaving the position unchanged) requires only  $O(k^2)$  to recompute the solution. While the algorithm is not new, the authors hope that this brief disclosure will alert the vision community to its potential uses. The algorithm is precisely defined in [Daniel et al. 76] and widely available implementations can be found in various mathematical libraries, e.g. IMSL.

### 3.3 Definition of the Energy Measure

The basic form of the energy measure is given by equation 2 except that the region of integration may be different than that expressed therein.

The energy of the surface will depend on the size of the region in  $\mathbb{R}^2$  over which the energy norm is computed. The two most natural choices are  $\mathbb{R}^2$  itself, and the convex hull of the data defining the "current" surface. Unfortunately, neither is appropriate. For the class used in the initial tests, the integral over  $\mathbb{R}^2$  is not necessarily finite.<sup>†</sup> While the energy norm over the convex hull of the data defining the "current" surface is obviously finite,<sup>\*\*</sup> this choice has two difficulties:

1. The convex hull would continuously change as new data points were added to a surface.
2. The use of a domain which ends near the data points will allow the addition of new points to lower the surface energy, thus the energy will no longer be monotonically increasing, making a region growing method less stable.<sup>††</sup>

### 3.4 Derivation of Bounds on Energy of Surface

Given the definition of the spline as in equation 4, one can symbolically compute bounds on the energy. To begin, the exact form of the energy integral is manipulated to explicitly expand the squaring operation and move the differentiation

---

<sup>†</sup>If one considers the class  $D^2H^0$ , then the energy norm is, by definition, finite. However, the energy value may be very large and thus may be numerically unstable.

<sup>\*\*</sup>Intuitively one can assume a Lebesgue integral which may ignore sets of measure zero. More formally, the definition of the class was in terms of distributions and the energy measure can be defined in a distributional sense as well.

<sup>††</sup>Intuitively this anomaly can be understood by noting the following:

- (a) Surface energy measures the "bending" of the surface in the domain of integration.
- (b) The definition of the class of surfaces insures that the value of "the surface" must go zero as the point of evaluation approaches infinity, thus outside the convex hull of the data the surface will approach zero, and this may cause some "ringing" in the surface if the data density is "low" near the edge of the region.
- (c) By increasing the data density, the value of the surface inside a bounded region can be forced to approach a planar surface which has zero energy. Thus after initially building a surface using points on some region boundary, adding new data points within the region can "push" the high energy portions (i.e. bent areas) outside the convex hull. This results in a nonmonotonic energy measure.

and integration inside the sum, to wit:

$$\|\sigma\|_{D^2} = \left\{ \int_{X_1}^{X_2} \int_{Y_1}^{Y_2} \left( \left( \frac{\partial^2 \sum_{i=1}^k \alpha_i K(x, y; x_i, y_i)}{\partial x^2} \right)^2 + 2 \cdot \left( \frac{\partial^2 \sum_{i=1}^k \alpha_i K(x, y; x_i, y_i)}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 \sum_{i=1}^k \alpha_i K(x, y; x_i, y_i)}{\partial y^2} \right)^2 \right) dx dy \right\}^{\frac{1}{2}} \quad (6)$$

By letting the integration be over the square  $[-B, B]^2$  we have can obtain

$$\|\sigma\|_{D^2} = \left\{ \sum_{i=1}^k \sum_{j=1}^k \alpha_i \cdot \alpha_j \cdot \int_{-B}^B \int_{-B}^B (K_{xx}(x, y; x_i, y_i) \cdot K_{xx}(x, y; x_j, y_j) + 2 \cdot K_{xy}(x, y; x_i, y_i) \cdot K_{xy}(x, y; x_j, y_j) + K_{yy}(x, y; x_i, y_i) \cdot K_{yy}(x, y; x_j, y_j)) dx dy \right\}^{\frac{1}{2}} \quad (7)$$

While it would be most appropriate to integrate the terms symbolically in the last of the above equations, the authors (and MACSYMA) have been unable to obtain a solution. Fortunately, a symbolic solution can be obtained for the upper bound using the fact that we can compute:

$$I(B, s, t) = \int_{-B}^B \int_{-B}^B \{ K_{xx}(x, y; s, t)^2 + 2 \cdot K_{xy}(x, y; s, t) + K_{yy}(x, y; s, t)^2 \} dx dy \quad (8)$$

While we will only report on the classes  $D^2 H^{-\frac{1}{2}}$  and  $D^3 H^{-\frac{3}{2}}$ , we have obtained closed form solutions for  $I(B, s, t)$  for various values of  $m$  and  $\eta$ . These are given in the following table:

$D^2 H^{-\frac{1}{2}}$	$180B^2(t^2 + s^2) + 120B^4$
$D^3 H^{-\frac{3}{2}}$	$252B^2$
$D^2 H^{\frac{1}{2}}$	$\frac{1}{7} \cdot (11900B^2t^6 + (35700B^2s^2 + 71400B^2)t^4 + (35700B^2s^4 + 142800B^4s^2 + 66640B^6)t^2 + 11900B^2s^6 + 71400B^4s^3 + 66640B^6s^2 + 8160B^6)$
$D^3 H^{-\frac{3}{2}}$	$17100B^2(s^4 + t^4) + 34200B^2t^2s^2 + 45600B^4(s^2 + t^2) + 10640B^6$
$D^4 H^{-\frac{5}{2}}$	$87300B^2(s^2 + t^2) + 58200B^4$

Table 1: Table of  $I(B, s, t)$  for different classes of functions.

The choice of norm is significant. The second norm gives zero energy to planar surfaces, but large energy to conic sections. Thus it is appropriate when the world model is known to be mostly planar. This is exactly the case for many man-made objects, such as bent sheet metal, desks, and other rectangular scenes. In contrast, the third norm gives zero energy to conic sections, and thus it is appropriate when many round objects are in the scene.

Once the norm is selected we set the maximum energy thresholds according to the precomputed energy for prototype shapes in the scenes. A number of intermediate thresholds are also used to slowly increase the surface energy. This prevents the erroneous segmentations that occur if only the final threshold values occur, since it bends the surface slowly. The algorithm therefore places points onto the surface where they most easily fit. The next larger threshold is used when no more points can be placed within the current one.

### 3.5 Heuristics

Heuristics reduce the cost of the energy-based method by selecting points to be considered for the various surfaces. Good heuristics should focus the system by selecting the points that should be added to each surface. The points selected by a heuristic can then be tested for the minimum energy criteria, and only the low-energy points are used. We now discuss several subproblems where heuristics help, in particular building the initial surfaces, adding points to them, and merging or eliminating duplicate surfaces.

The system combines two kinds of heuristics. *Domain independent* heuristics are general enough to provide reasonable results for many scenes, including scenes with occluding surfaces, low-energy surfaces and high-energy surfaces. We have developed several such heuristics, and describe them below. *Domain specific* heuristics increase efficiency by selecting points according to the expected scene characteristics, for example if information is available on the curvature of a surface the program can construct an initial solution from the points that are "likely" to be on the surface. This provides a good basis for adding more points by the energy-based method. Domain specific information also can be used to set the thresholds for the surfaces.

*Construction of the initial "basis" surfaces.* Each surface is initially described by a small number of points, between 4 and 30. Good starting surfaces are essential for accurate segmentation, and therefore the program must find points that all lie on the same actual surface. This is done in two steps. First a significant single point is found as a "seed" for the basis. Second, a subset of its neighbors in 3-space are selected. Ideally each "seed" point should be from a different surface. Since we cannot achieve this without knowing the segmentation, we instead select the seed according to the following heuristics. A physical explanation of why it is a good heuristic follows each one:

- Pick the nearest point (physically, this is a point on the closest object).
- Pick the farthest point (physically, a point on the most distant object).
- Pick randomly, but not near any seed value that has already been chosen.

- Pick a point near to a specific  $XY$  coordinate on a grid (appropriate when approximate object locations or distributions are known).
- Pick the point that is farthest from one picked previously (span as much of the scene as possible).
- Use a clustering technique that selects the point that has the most neighbors which are within the average inter-point distance, since these are significant as the center of an object. Use the method to obtain peripheral points by picking the point with the fewest number of neighbors.

The lowest energy surface is then built from the seed and a subset of the neighboring points. This entails a combinatorial search of the  $M$  near neighbors for the lowest energy  $N$  element surface. The starting surfaces are identified by selecting an "interesting" point and its neighbors in 3-space. We then perform an exhaustive search of the  $M$  nearest neighbors to build  $N$ -point surfaces. If  $M$  is too small the constructed surface may span several actual surfaces. Likewise, if  $N$  is too small the resulting surface will not be sufficiently descriptive to permit accurate selection of points.

It is important to restrict the search to the near neighbors because it is otherwise quite easy to build large nearly-planar surfaces that cut through several surfaces. Moreover, the points must not be too close to each other because the resulting reconstruction would be highly distorted by surface noise. Otherwise the points are not distinguishable in  $XY$  coordinates but the resulting surface can have huge energy due to the variation in  $Z$ , which will be quite relative to the difference in  $XY$  values. The  $Z$  values could be due to noise or to multiple surfaces.

The algorithm can recover from incorrect starting surfaces. First, the smoothness assumption identifies as erroneous any surface that has a large energy with a small number of points. These surfaces can be pruned and the points reused on other surfaces.

*Point selection: what point to add?* Once the initial basis surface has been constructed, points must be selected and then added to the surface which can accept them with minimum increase in energy. The exhaustive method requires  $N^2m$  update operations (for  $N$  points and  $m$  surfaces) at a cost of  $N^2$  per update. This  $N^4$  cost is exorbitant, and therefore we first estimate the energy. We have been successful with an estimation method that considers only the points which are near to some point already on the surface (we will describe this shortly). The nearness criteria also prevents the erroneous addition of a non-surface point that just "happens" to fit onto a partially developed surface, for example with occluding transparent surfaces.

A point is near a surface if the distance to some point on the surface is less than half the diagonal of the box that bounds the points already on the surface. The point is temporarily ignored if it is too far from the surface, and is considered at a later iteration after the surface has accepted closer points (and the bounding box is larger). The near points are then ranked by a weighted distance formula.

This combines the distance to the nearest point already on the surface, with the proximal distance from the point to the partially reconstructed surface. The points are then added in order of increasing cost subject to a global threshold.

*Culprit identification: when and how to pick points to delete?* Misclassifications of data during the incremental addition of points to surfaces results in excessively large surface energy. It is then important to delete some point(s) from the surface. In general the point which makes the largest contribution to the energy does not belong to the surface and is the classification error. Thus it should be deleted. The exhaustive method to find this point (i.e. try each possible point) is uneconomical. Therefore we need some way to predict which point is the "culprit" responsible for the excess energy.

A good indicator of the "culprit" is the value of  $\alpha$  in the linear system which recovers the spline parameters. Intuitively this is because a large  $\alpha$  makes a large contribution to the energy. We have shown experimentally that an incorrect data point is associated with a large alpha value. We use culprit deletion by removing the point with largest  $\alpha$  after every 10 insertions. If a point is erroneously removed it can still be added back in a subsequent add operation. This is because the increase in energy at each step is a good predictor of the correct segmentation.

*Segmentation: when to create a new surface?* We have investigated two methods: first of completely building one surface at a time, and second of building all surfaces simultaneously by gradually increasing the global energy threshold. The second method gives significantly better results than the first, since it completes all low-energy classifications before it increases the energy. The first method can misclassify when a point can be on either  $S_1$  with large energy or on  $S_2$  with lower energy. The approach of building all surfaces may also improve performance of the parallel implementation. However, the simultaneous building of multiple surfaces will construct redundant surfaces, and also place co-surface points onto different reconstructions. This problem is solved by merging similar solutions as described below.

*Merging: when and how to combine surfaces into one?* Physically close surfaces should be merged. For example, two surfaces are "close" if both interpolate to the same  $Z$  values at all  $XY$  values. The reason we construct redundant copies is because we begin with many initial surfaces, and several of these initial surfaces may be from the same actual surface. They should be merged because the points are actually from the same surface and should be classified as such.

We efficiently test if two surfaces should be merged by interpolating the surfaces  $S_1$  and  $S_2$  on a sparse and fixed  $X - Y$  grid, and then constructing a new surface from these samples. We use precomputation to efficiently compute the energy of such a system, which reduces the computational expense to simple data interpolation (at the grid values) and back-substitution. This is much cheaper than building a new surface by adding each point to it.

A surface is considered for merging only if it has at least the median number

of data values on it. The merge cost of all such pairs is computed, and the merges are performed in order of increasing energy. Each point of the less-dense surface is considered for the denser surface, and is added if the point does not exceed  $k$  times the median energy of the less-dense surface. In this manner erroneous points will not be merged, but instead are returned to the pool of unprocessed data and will be placed onto a surface in a subsequent iteration.

*Pruning: when should a surface be eliminated?* Sometimes a surface is incorrectly started from the data on several different scene surfaces. The energy of such surfaces is generally high, and thus the surface does not accept many additional data points. These sparse surfaces are discarded and the points are reused on some other surface. The program thereby recovers from false starts by resegmenting data that does not produce a surface.

### 3.6 Parallel Implementation

An prototype of segmentation is operational on the massively parallel Connection Machine. The CM-2 has 65,536 processors with 512 megabytes of memory and a 300 gigabyte/sec memory bandwidth. This gives 9K bytes RAM per processor. Although the processors are small 1-bit PEs, the typical aggregate operation speed is 2500 mflops for double precision on a 4Kx4K matrix multiplication, and 5000 mflops for a dot product. For a complete technical summary see [Thinking Machines].

We have implemented for the CM-2 the updatable QR algorithm for solving linear systems, construction of the linear system (equation 5), evaluation of surface energy in parallel, and adding points to a surface. This software has processed data from a number of synthetic surfaces including spheres and planes. The software is written in the \*Lisp language.

A major bottleneck is the combinatorial search for the initial basis surfaces. Since the amount of sensor data exceeds the PE capacity, the algorithm distributes a different subset of the data to each processor. In one data-parallel subroutine each processing element simultaneously constructs the minimum energy surface that fits its subset. The lowest energy solutions are retained as the starting-bases that accept additional data points.

We need the parallel implementation to process data much faster than we can do with a uniprocessor implementation. This should allow us to process thousands of points, as occurs in non-sparse data, and also to process many surfaces. The heuristics can also be processed in parallel, in particular the local properties of all points can be explored simultaneously.

## 4 The pluses and minuses

This section critically reviews the algorithm described in this paper pointing out some of the major advantages +, major problems -, and some aspects which can be viewed as either a pro or a con  $\pm$ .

- + The segmentation process is based on surfaces having low bending energy, a heuristic which can be directly related to the physical process of surface

formation. Since this heuristic is often used in surface reconstruction, it is a natural for segmentation as well. Because we do not attempt to detect discontinuities, the algorithm is well behaved for very sparse data, and even handles transparent and occluded objects with few problems.

- + The functional form of the reproducing kernel-based spline allows for analytic computation of bounds on the surface energy, thus making the segmentation process reasonably computationally efficient. The functional form of the energy bounds are quite simple.
- + While not presented here, the algorithm can easily be extended to handle the case of derivative information (e.g. surface orientation or curvature) in addition to depth data. The extension is accomplished by allowing a more complex surface reconstruction scheme, see [Kender Lee Boulton 85].
- ± The complexity of the algorithm, with the selection heuristic of minimal energy addition, is  $O(n^4)$  for  $n$  points, and with the other heuristics it is  $O(n^3)$ . For very sparse data, this is a significant saving over discrete regularization costs, but as the data densities grow, the algorithm becomes too costly.
- ± The algorithm does not recover "boundaries" for the segmented data. This is advantageous because it allows for transparent and/or occluding surfaces, and because data is generally sparse (and often noisier) near the boundary resulting in a poor boundary definition. This is a disadvantage because it requires a secondary processes (possibly using ideas borrowed from work on subjective contour perception or Gestalt psychology) to determine the actual boundary.
- ± The algorithm can easily be adapted to different measures of surface smoothness. This is advantageous because it allows for greater flexibility, but disadvantageous because determination of the most appropriate measure is difficult. The measure used in the experiments presented herein has proved to be a reasonable one.
- ± The algorithm is based on reproducing kernel-based splines which are essentially a global surface reconstruction algorithm and provide for efficient serial implementation for sparse data (say  $< 500$  points per surface on a  $512 \times 512$  grid). If there are more points, the algorithm can be extended to use local reproducing kernel-based splines (loosely based on [Franke 82]), at the cost of making the surface definition localized to patches.
- ± The reproducing kernel-based spline with updates to the QR factorization provide an efficient serial implementation. However, they do not suggest an efficient parallel implementation other than the trivial extension of doing the matrix computations in parallel. Future work may explore the possibility of using parallel multi-grid techniques to solve a discretized version of the surface reconstruction problem which can then be used in the segmentation algorithm. Unfortunately how one evaluates energy in this case is still unclear.

- ± The order of processing of points currently effects the resultant segmentation in all but the exhaustive heuristic. This is especially true when two surfaces come into direct contact and join in a rather smooth fashion (e.g. a wedge). This may actually be used to help in the segmentation process by processing the data in multiple orders and using any difference in data labeling to suggest a refined segmentation.
- The algorithm currently uses a threshold on the energy bounds for surfaces in the scene. A schedule of surface thresholds is currently manually set, and future work (in progress) attempts to redress this issue. Luckily, this threshold for energy-based segmentation does not seem too sensitive as say thresholds for segmentation of an image based on intensity, e.g. (variations of 10% of the threshold are generally indistinguishable).
- The algorithm assumes one is interested in smooth surfaces and will most likely fail when this assumption is not satisfied. Unfortunately, the algorithm cannot even determine if the assumptions are satisfied (For example, consider a rough surface similar to a plane covered with a large number of small densely packed cones. If the data supplied to the algorithm are points on the background and the peak values of the cones, the algorithm is hopelessly doomed to predict two planar surfaces.)
- The algorithm is surface based, and cannot deal with data from multiple views of a volumetric object. Additionally, it will often fail if noise is such that a single  $x, y$  location is assigned multiple data values (of the same type).

## 5 Experimentation

This section describes some of the initial experimentation with the segmentation algorithm. The reader should remember that the experimentation involves some human interaction (to determine thresholds) and most of the examples highlight the systems best behavior. The experiments we run on a Sun Sparc1 workstation configured with 12M of memory.

We processed three sets of data. First is scene of *synthetic* data for three superquadric surfaces which are both overlapping and with very similar shapes. The results show excellent differentiation between the surfaces. However, this is sensitive to the heuristics which pick the initial basis surfaces. It is otherwise possible to hypothesize an initial surface which spans several actual surfaces. Such surfaces will eventually have excessive energy and be pruned, but there is a cost in lost processing time in this event. Second, several sets of laser range finder data have been processed from; laser data from the Purdue Vision Lab is due to Avi Kak, and also laser data from the University of Utah.

For the synthetic data we show a needle plot of the initial data, a needle plot of the segmented data, the reconstructed surfaces, and statistics on the number of wrong segmentations. For the laser data we show the same plots but give statistics based on hand-segmentation of the data.





Figure 1:

The data shows needle plots of the segmentation of actual laser range-finder data. Data is from the Utah range database [Wan 85]. The upper-right shows the unsegmented data consisting of a cube, a cylinder, and a sphere on a table. The lower right is the top of the sphere, the lower left is the top of the cube (with a piece of the table), and the upper left is the rim of the cylinder. Not shown here are other pieces of the segmentation, such as the side of the cylinder. Note the algorithm makes no assumption about the surface shape or curvature, and does not recover planar patches!

The discussion of each example appears as the caption to the figures showing the initial data and the reconstructed surfaces. The initial data is presented as perspective views of the depth points represented as needles in space.

## 6 Conclusions and future work

This paper has presented a new algorithm for segmentation of depth data. The algorithm is based on adding points to a surface only when doing so does not increase the surface bending energy above a user determined threshold. The algorithm has been experimentally tested and in most cases correctly labels all data points. The algorithm does *not* determine boundaries between segmented surfaces which allows it to handle extended objects occluded by other objects and transparent objects.

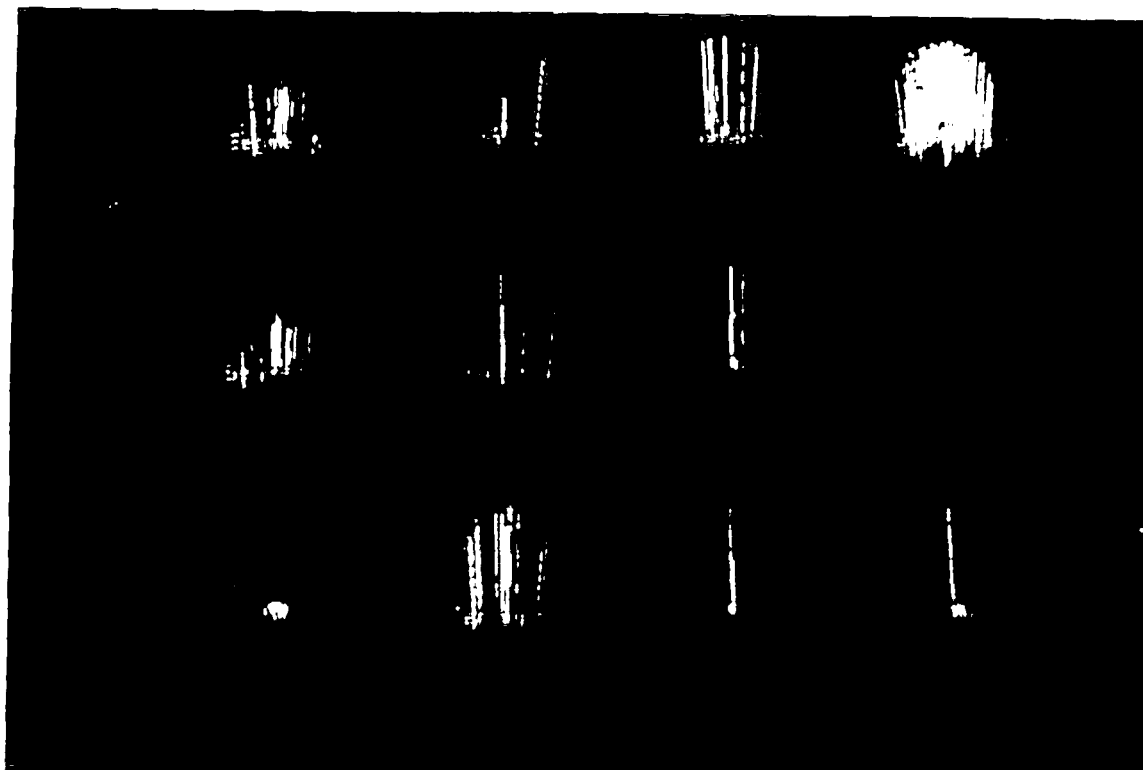


Figure 2: This shows a needle plot of three overlapping spheres. The data was generated with radius and center as follows. The upper right shows the unsegmented data. The recovered segments are listed to the left. The parameters of the spheres are: Top surface:  $r = 0.8$ ,  $c = (0.0, 0.0, 1.0)$ . Middle surface:  $r = 0.25$ ,  $c = (0, 0, 0)$ . Bottom surface:  $r = 0.25$ ,  $c = (-0.5, -0.5, -0.5)$ . Each surface is sampled at 150 points on a spherical coordinate system and randomly shifted from the grid. Uniform random noise in the range  $[-0.05, 0.05]$  was added to the  $z$  values of the data.

"Segment 0" is the smallest sphere, with one erroneous point. "Segment 1" is the second sphere, with 5 erroneous points. "Segment 3" is the outer sphere. Note that segments 5 and 6 are clearly errors and can be partitioned into groups of nearly uniform length.

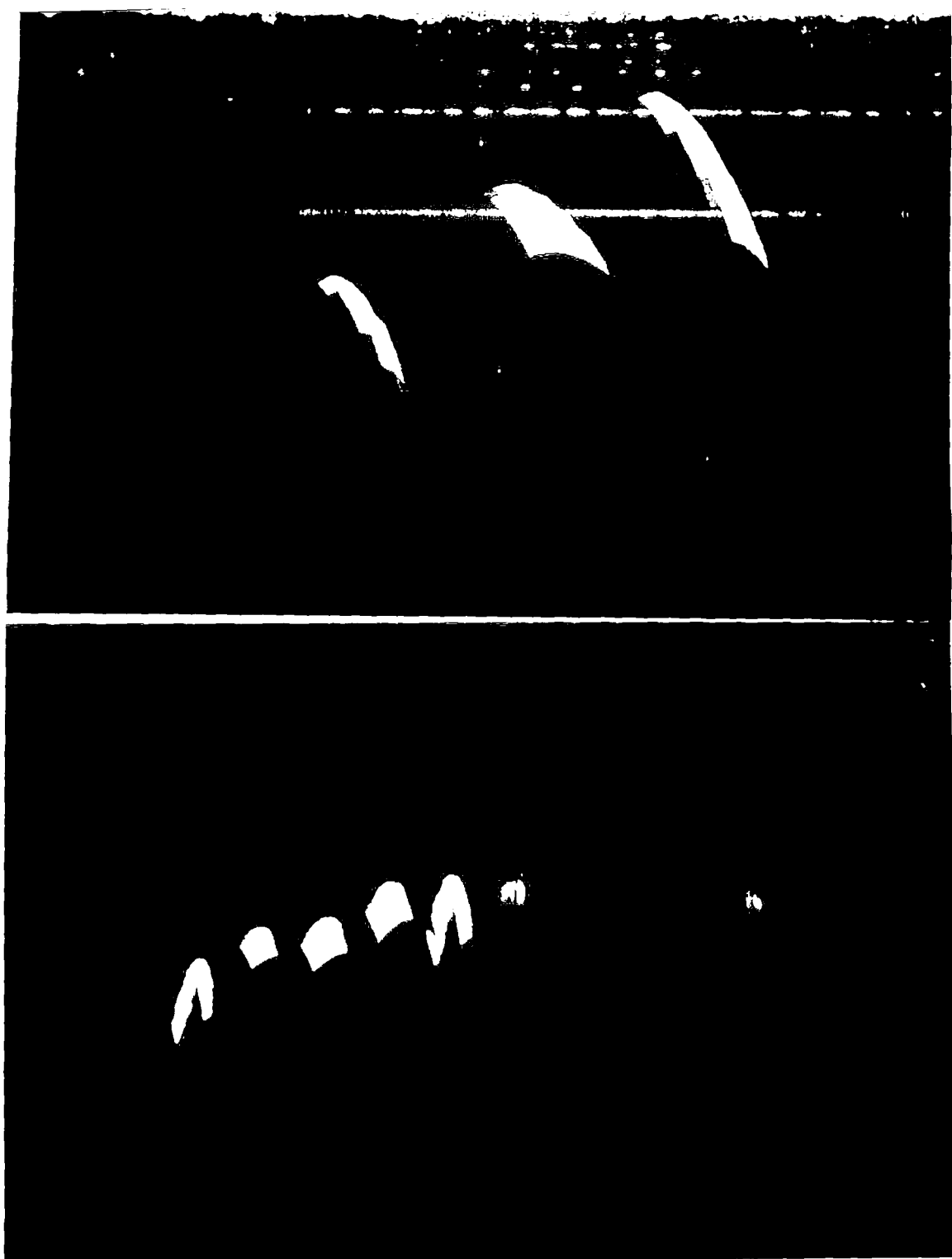


Figure 3:

This shows all the recovered surfaces from the above example. The upper image shows the reconstructions of segments 0, 1 and 3. The lower image shows all the images. The five leftmost surfaces correspond to correct segmentation. The sixth surface started with the erroneous zero-energy basis surface of a horizontal plane. Points from the other surfaces then deformed it at the lower edges. The peaks are due to extrapolation error in the graphic rendering. This corresponds to "segment 5" in the needle plot in the previous figure; note the needles can be partitioned into two groups of nearly-equal height.

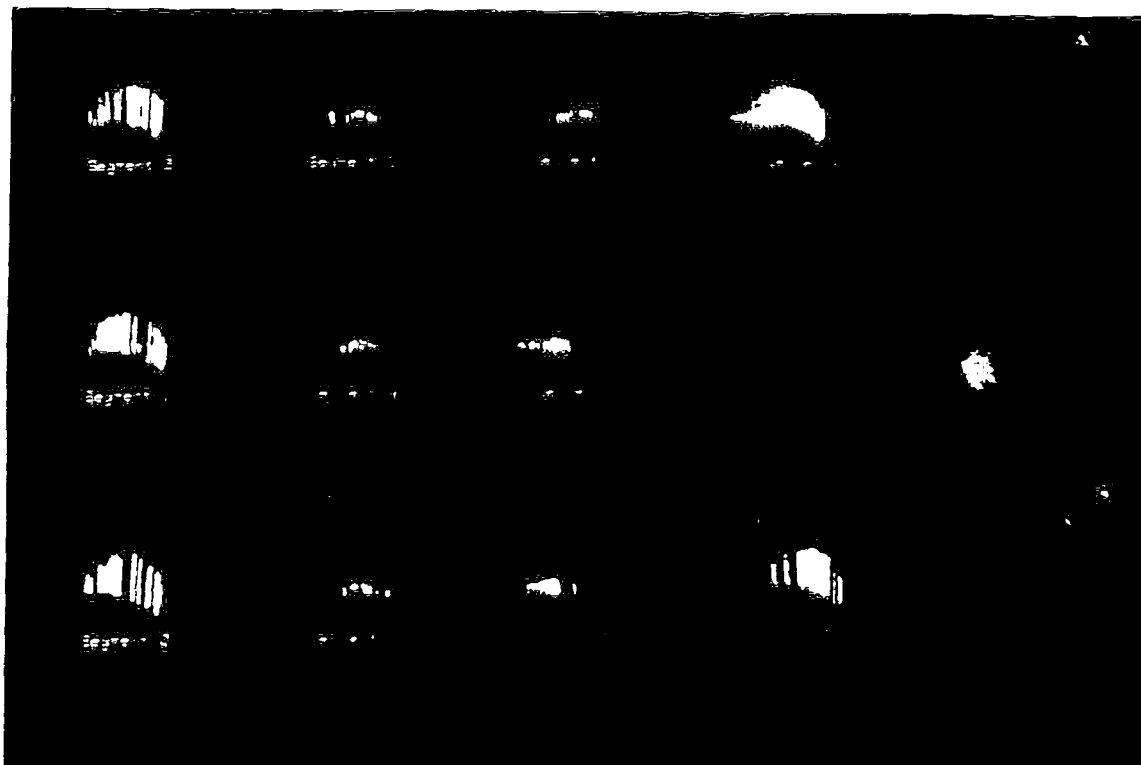


Figure 4: This shows laser range finder data for three round surfaces (Purdue Vision Lab data). The unsegmented data is in the upper right, and the segments are to the left.

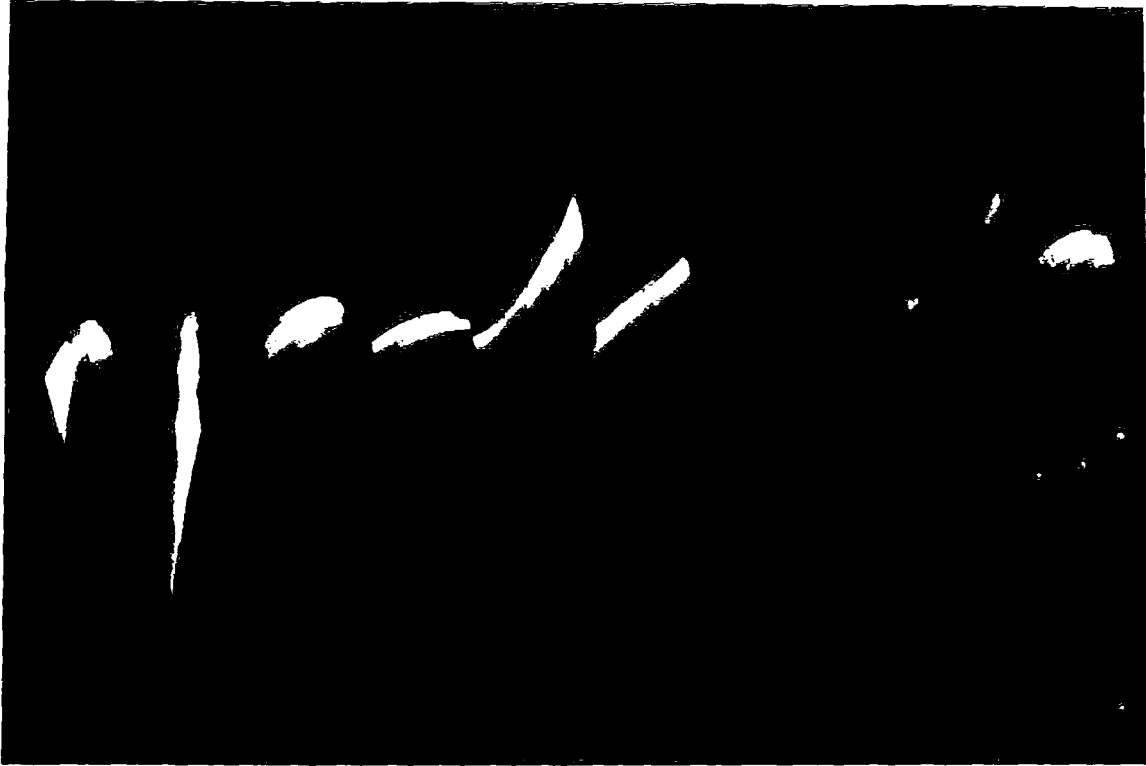


Figure 5: This shows the reconstruction and segmentation of the laser data for three round surfaces without culprit elimination. The first, second, third, and last are the top sphere. The fifth, sixth and ninth correspond to a middle cylinder shape. The seventh and eighth are a planar section in the initial data.

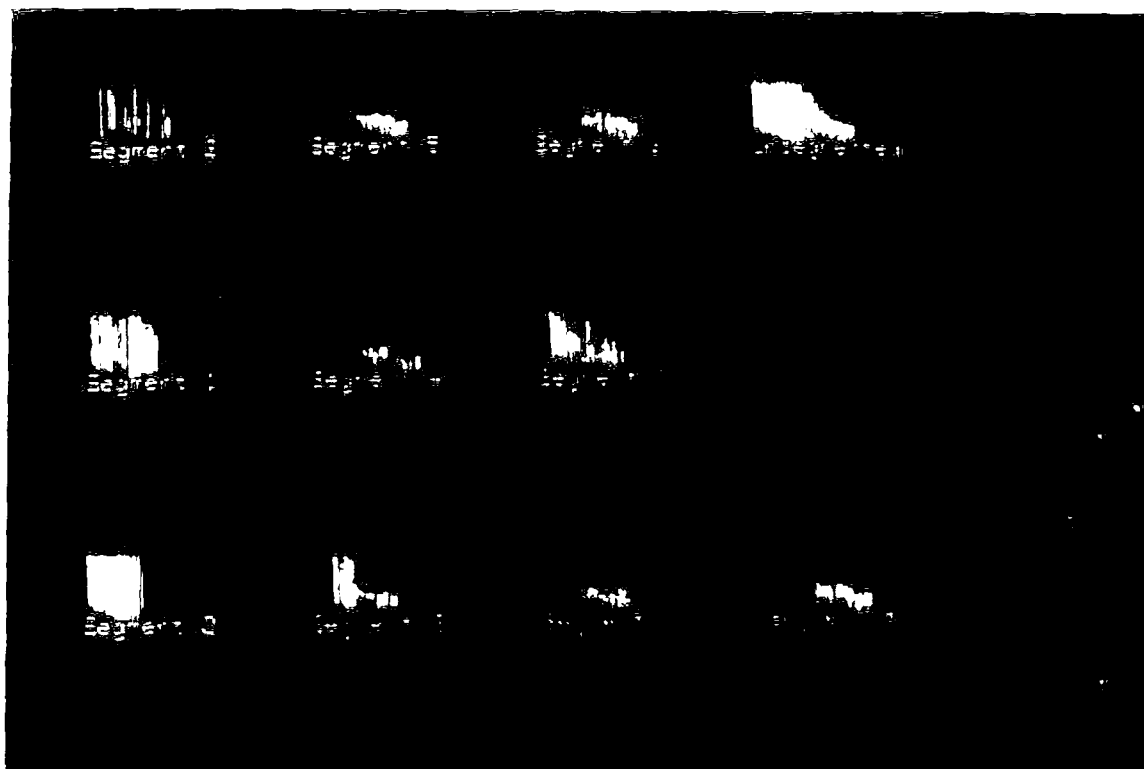


Figure 6: This shows several planar and slanted surfaces. Note there is a circular cutout in the slanted surface. The original data is in the upper right, and the segmentations to the left. (Purdue Vision Lab data)



Figure 7: This shows reconstruction of the planar and slanted surfaces.

A problem with the current algorithm is that the energy of a surface may become large not because of the last point added, but because of some point previously considered. Future work will investigate additional ways of identifying the "culprit" data (probably using a local energy measure) and removing that point rather than the most recently added point. This will also help alleviate problems caused when the "initial basis" for the surface contains data from different surfaces.

One of the most obvious failings of the approach is the current dependence on a global thresholding technique to realize the segmentation. Such a process is doomed to be troublesome unless a systematic determination of the threshold is possible. Future work will address this issue and will also investigate the use of adaptive thresholding (depending on the actual data) and the use of other properties, say rate of change of energy, as the means of realizing segmentation.

The algorithm as presented has little theoretical basis for the use of energy as an indicator for segmentation. Of course, in a worst case setting, segmentation is an unsolvable problem. However, on the average there is still hope of determining a theoretical basis for segmentation. Future work will explore this idea, and attempt to show that surface energy can be related to the probability of all data-points being on the same surface. This will likely borrow from the work of [Kimeldorf and Wahba 70], where a fundamental relationship is shown between reproducing kernel-based splines and optimal Bayesian estimators.

While the method, as presented, is limited to depth data, future work will also address the implementation issues which remain before derivative information can also be incorporated. (The related theoretical issues are already being resolved).

### Acknowledgments

This work was supported in part by Darpa Contract #N00039-84-C-0165 and in part by the National Science Foundation Grant #IRI8800370. Thanks to Cliff Beshers for his help with the graphics displays. Part of coding for an earlier version of this approach was done by Liang-Hua Chen.

### References

- [Thinking Machines] Thinking Machines. Connection Machine (R) Model CM-2 Technical Summary. Thinking Machines Incorporated, Cambridge Mass., 1987.
- [Marr 81] D. Marr. Vision: A Computational investigation into the human representation and processing of visual information. W.H. Freeman and Company, San Fransisco, 1981.
- [Allen 85] P. K. Allen. *Object Recognition Using Vision and Touch*. PhD thesis, University of Pennsylvania, Department of Computer Science., 1985.
- [Anandan and Weiss 85] P. Anandan and R. Weiss. Introducing a smoothness constraint in a matching approach for the computation of displacement



- fields. In *Proceedings of the DARPA Image Understanding Workshop*, pages 186-195, DARPA, 1985.
- [Bajcsy and Solina 87] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proceedings of the IEEE Computer Society International Conference on Computer Vision*, pages 231-240, IEEE, June 1987.
- [Bates and Wahba 82] D. Bates and G. Wahba. Computational methods for generalized cross-validation with large data sets. In C.T.H. Baker and G.F. Miller, editors, *Treatment of Integral Equations by Numerical Methods*, pages 283-296, Academic Press, New York, 1982.
- [Besel and Jain 86] P. J. Besel and R. C. Jain. Segmentation through symbolic surface description. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 77-85, IEEE, 1986.
- [Blake and Zisserman 86] A. Blake and A. Zisserman. Invariant surface reconstruction using weak continuity constraints. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 62-68, IEEE, 1986.
- [Boult 86] Terrance E. Boult. *Information Based Complexity in Non-Linear Equations and Computer Vision*. PhD thesis, Department of Computer Science, Columbia University, 1986.
- [Boult 87] T.E. Boult. What is regular in regularization? In *Proceedings of the IEEE Computer Society International Conference on Computer Vision*, pages 457-462, IEEE, June 1987.
- [Kanade 88] Matthias L., R. Szeliski, T. Kanade. Incremental estimation of dense depth maps from image sequences. In *Proceedings of Computer Vision and Pattern Recognition 1988*, pages 366-374.
- [Choi and Kender 85] D.J. Choi and J. R. Kender. Solving the depth interpolation problem with adaptive chebyshev acceleration method on a parallel computer. In *Proceedings of the DARPA Image Understanding Workshop*, pages 219-223, DARPA, 1985.
- [Daniel et al. 76] J.W. Daniel, W.B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. *Math. Comp.*, 30:722-795, 1976.
- [Franke 82] R. Franke. Smooth interpolation of scattered data by local thin plate splines. *Comp. & Math. with Applications*, 8(4):273-281, 1982.
- [Grimson 81] W. E. L. Grimson. *From Images to Surfaces: A Computational Study of the Human Visual System*. MIT Press, Cambridge, MA, 1981.
- [Hoff and Ahuja 85] W. Hoff and N. Ahuja. Surfaces from stereo. In *Proceedings of the DARPA Image Understanding Workshop*, pages 98-106, DARPA, 1985.

- [Kender Lee Boulton 85] Information-based complexity applied to optimal recovery of the  $2\frac{1}{2}$ D sketch. In *Proceedings of the IEEE Computer Society Workshop on Computer Vision: Representation and Control*, pages 157-167, IEEE, October 1985.
- [Chou Brown 88] Chou P., C. M. Brown The theory and practice of bayesian image labeling. In *International Journal of Computer Vision* (to appear).
- [Hoff and Ahuja 87] W. Hoff and N. Ahuja. Extracting surfaces from stereo images: an integrated approach. In *Proceedings of the IEEE Computer Society International Conference on Computer Vision*, pages 284-294, IEEE, 1987.
- [Julesz 71] B. Julesz. *Foundations of Cyclopean Perception*. University of Chicago Press, Chicago, IL, 1971.
- [Kimeldorf and Wahba 70] G.S. Kimeldorf and G. Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *Annals of Math. Stat.*, 41(2):495-502, 1970.
- [Lee 85] D. Lee. *Contributions to Information-based Complexity, Image Understanding, and Logic Circuit Design*. PhD thesis, Department of Computer Science, Columbia University, 1985.
- [Meinguet 83] J. Meinguet. Surface spline interpolation: basic theory and computational aspects. *Institut de Mathematique Pure et Appliquee, Universite Catholique de Louvain*, 35, 1983.
- [Pentland 86] A. Pentland. *Recognition by Parts*. Technical Report 406, SRI International, December 1986.
- [Rao, Nevatia and Medioni 87] K. Rao, R. Nevatia, and G. Medioni. Issues in shape description and an approach for working with sparse data. In *Proceedings of the AAAI Workshop on Spatial Reasoning and Multi-sensor Fusion*, pages 168-177, St. Charles, IL, October 1987.
- [Terzopoulos 84] D. Terzopoulos. *Multiresolution Computation of Visible-Surface Representations*. PhD thesis, MIT, 1984.
- [Utah 85] Utah. *The University of Utah range database manual*. Technical Report, University of Utah, 1985.
- [Wahba 84] G. Wahba. Surface fitting with scattered noisy data on euclidean d-space and on the sphere. *Rocky Mountain Journal of Mathematics*, 14(1):281-299, 1984.