Finding the Parts of Objects in Range Images

André Lejeune and Frank P. Ferrie

CIM-93-8 August 1993

McGill Research Center for Intelligent Machines

McGill University, McConnell Engineering Building

3480 Université, Montréal, Québec, CANADA, H3A 2A7

Email: andre@lightning.mcrcim.mcgill.edu

ferrie@lightning.mcrcim.mcgill.edu

Tel: (514) 398-6042 Fax: (514) 398-7348

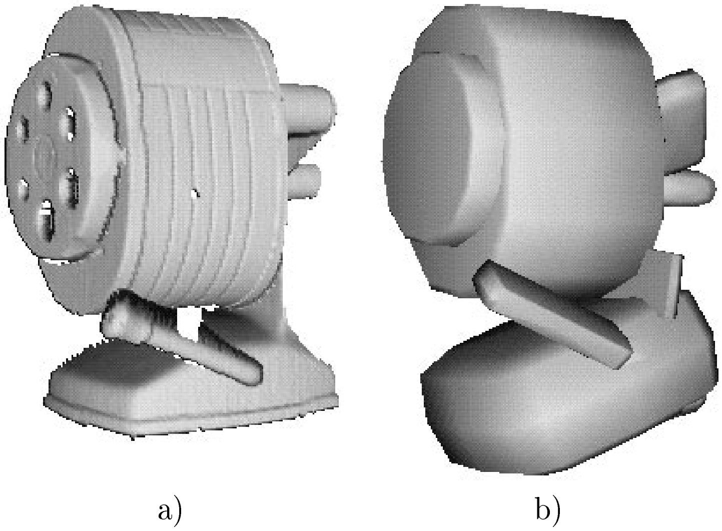
Abstract

A key problem in the interpretation of visual form is the partitioning of a shape into principal components that correspond to the parts of an object. This paper presents a method for partitioning a set of surface estimates obtained with a laser range finding system into subsets corresponding to such parts. The strategy employed makes use of two complementary representations for surfaces: one that describes local structures in terms of differential properties (e.g. edges, lines, contours) and the other that represents the surface as a collection of smooth patches at different scales. It is shown that by enforcing a consistent interpretation between these two representations, it is possible to derive a partitioning algorithm that is both efficient and robust. Examples of its performance on a set of range images are presented.

Section 1 Introduction

A key problem in the interpretation of visual form is the partitioning of a shape into principal components that correspond to the parts of an object. This paper presents a method for partitioning a set of surface estimates obtained with a laser range finding system into subsets corresponding to such parts. The strategy employed makes use of two complementary representations for surfaces: one that describes local structures in terms of differential properties (e.g. edges, lines, contours) and the other that represents the surface as a collection of smooth patches at different scales. Traditionally, parts decomposition has been posed as the problem of interpolating features associated with part boundaries. However, the locality of these features has made this a difficult problem. In this paper, we present a novel method that makes use of an auxiliary representation, a curvature scale space, which can simplify the process of interpolation in an efficient and robust manner. The key idea is that each of these representations can provide structural cues that may not be present in the other. Since each representation describes the same physical object, it is possible to exploit consistencies in each to arrive at a plausible interpretation, in this case a set of part boundaries.

The input to our algorithm consists of two maps, $x[m][n]$, a discrete set of range measurements sampled from a piecewise-continuous surface S, and $f[m][n]$, a feature map that identifies putative boundary points. For a given $f[m][n]$ the algorithm finds an optimal set of contours that best interpolates the set of boundary points while partitioning S into closed regions that are consistent with the scale-space representation. Estimation of $f[m][n]$ is beyond the scope of this paper, but is described elsewhere [17, 14].



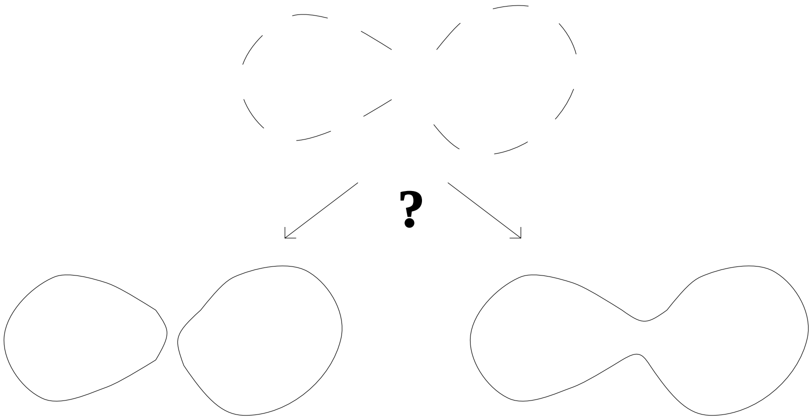
a) A shaded range image of a pencil sharpener, rendered from a depth map obtained with a laser range finding system. b) A representation of the sharpener using superquadric primitives, one for each part. To determine the description for each part (i.e. superquadric parameters), one must first partition the range data. The algorithm described in this paper was used to obtain the result shown.

Figure 1.1: Two representations of a pencil sharpener.

1.1 Motivation

The motivation for this research can be explained with the aid of Figure 1.1, which shows two different representations of a pencil sharpener. Figure 1.1a corresponds to a depth map obtained with a laser range finding system [30]. However, for tasks involving manipulation or recognition of the object, information regarding position, orientation and shape should be made more accessible [25] as is the case with the representation shown in Figure 1.1b. Here the position, orientation and approximate shape of each part is made explicit by using a superquadric shape primitive that best accounts for the corresponding range data [12, 13]. In addition to making essential features more accessible, the second representation also corresponds to a data reduction of approximately 80:1.

A fundamental problem in the process of object modelling is that of *parts decomposition*, the partitioning of the surfaces of an object into regions or segments corresponding to each part[1] [19]. Traditionally two general approaches have been considered. So-called edge-based approaches work by first identifying features corresponding to part boundaries and then interpolating to form smooth contours. Region-based approaches, by contrast, first identify clusters or regions that are continuous in specified features and then seek to merge smaller groupings until some continuity criteria are violated. Each strategy taken separately is often insufficient for lack of robustness, non-uniqueness or complexity. For example, the interpolation problem in edge-based methods is often complex and becomes very difficult when features are sparse [33, 12, 13], as shown in Figure 1.2. On the other hand, region-based approaches using minimal descriptor length strategies can easily be perturbed in the presence of noise [28]. We seek a method for partitioning that is robust to data perturbation and sparsity of features.



Top) A feature set to be interpolated. Left) Possible interpolation that can be obtained with an ad-hoc scheme; two regions are found. Right) Other possible interpolation obtained with another ad-hoc scheme; only one region is found. It is not possible to determine which interpolation best interpolates the feature set. Both are acceptable, but which one correctly represents the data?

Figure 1.2: The interpolation problem.

[1] This precludes the problem of figure/ground separation, which is not addressed in this paper.

1.2 Methodology

A key observation of this paper is that edge-based and region-based processes are, in fact, complementary to each other as the detected edges provide cues to validate region boundaries and these same region boundaries specify how the edges must be interpolated [15]. In this paper we define an object to be a solid mass composed of convex parts without holes of any kind[2]. Consequently in our model of a part, an edge can correspond to a discontinuity in depth, (C^0), a concave discontinuity in orientation, (C^1), or a negative local minimum in normal curvature [1, 16, 12, 13]. This definition complies with the notion of transversality regularity [19]. When two arbitrarily shaped surfaces are made to inter-penetrate, they meet in a contour of concave discontinuities of their tangent planes [18].

A region is defined as a subset of the surface of an object that is void of any edge features defined above. Regions are represented as parametric surface patches computed at different resolutions [2] to account for the appearance of features at different scales. We refer to the resulting representation as a curvature scale space [10] because discontinuities and negative local minima of curvature are implicit in the patch boundaries at each scale. The relation of this scale space to the interpolation problem now follows. A single region implicitly defines a closed contour. When two adjacent regions are merged, the resulting region defines yet another contour. The set of all possible surface contours can thus be determined by enumerating all possible merges. By construction, the resulting space must contain the true surface boundaries. The computational task is to search this space for the correct interpretation.

For lack of specific constraints, the minimal descriptor length paradigm has often been used to guide this search problem [28]. But in the present case there is a more substantial principle involved, that of consistency. Given that the edge and region descriptions must both correspond to the same object, the search problem can be precisely formulated as that of finding the smallest set of covers, i.e. unions of regions, that is most consistent with the edge-based description.

[2] While somewhat restrictive, this definition admits a large range of objects. It can be relaxed to a degree provided that holes can be ignored as in the pencil sharpener example.

Similar problems have been studied previously [26, 7, 15] in terms of sequential search strategies, but these can result in sub-optimal solutions from a global viewpoint. Backtracking is usually not considered in these algorithms, the search-space being simply too large. To obtain optimal solutions, what is required is a process that seeks a best covering but that can compromise locally for global benefit. It will be shown later that this can be achieved with the use of a relaxation labelling algorithm [20]. However, because there are many ways of combining adjacent regions, some means are required to reduce the search space involved.

1.3 Contribution

A curvature scale space was developed to reduce the dimension of the search space by organizing data in a precisely structured fashion. It has the property of smoothing regions of convex curvature to reduce noise while preserving the positions of features corresponding to part boundaries through scale.

Once the scale space representation is built, optimizing the covering of the surface so that the resulting boundaries are consistent with the detected cues becomes possible. We show that this optimization problem maps naturally into a relaxation labelling framework [20]. A label is assigned to each surface patch in the scale space representation and constraints are defined by the detected edges. These constraints will define which regions merge as the labels are allowed to change.

1.4 Paper Summary

This paper is organized as follows. Section 2 focuses on the curvature scale space with boundary conditions in 1-D and then shows how it can be adapted to 2-D. Next in Section 3, the resolution algorithm is presented and shows how relaxation labelling can be used to determine a set of part boundaries that is consistent with both the edge-based and surface-based representations. A number of experiments are presented in Section 4 in which range images from different sources are partitioned. Finally, some observations on the method and the results are presented in Section 5.

Section 2 Curvature Scale Space with Boundary Conditions

We now present a scale space that partitions a surface S into a collection of non-overlapping regions at different scales so that each region is void of C^0 discontinuities, concave C^1 discontinuities and negative local minima of curvature. By definition, any feature corresponding to a part boundary cannot be contained within any element of the resulting scale space. Scale is defined by a process of diffusion that progressively smoothes the surface S, yet preserves the positions of putative boundary features. Therefore, unlike a Gaussian scale space [32, 22], we avoid the problem of coarse-to-fine tracking, which is necessary to recover the true position of features. Accurate location of position is essential for further modelling. The process in one dimension is first considered and is then extended to two dimensions.

2.1 Scale Space in 1-D

In 1-D, scale spaces are comprised of a waveform undergoing progressive deformation. Through these deformation steps, the components of interest in the waveform are made explicit. Kimia [21] showed that any type of time dependent deformation of a parametrized curve $C_0(s)$ can be expressed by

$$\begin{array}{rcl} \frac{\delta C}{\delta t} & = & \beta(\kappa(s)))\ \vec{N}, \\ C(s,0) & = & C_0(s), \end{array} \tag{2.1}$$

where $\beta(.)$ is a function controlling the diffusion process, $\vec{N}$ is the normal to the curve, κ is the curvature of the curve, s is an arbitrary parameter (not necessarily arc-length) and t is the diffusion parameter, which takes place in time. To derive a scale space from this model, it is necessary to estimate $\vec{N}$ and κ for a given curve and to define a deformation function $\beta(.)$.

In this paper, we are concerned with curves of the form $y = f(x)$. For these curves, it is reasonable to assume that the normal to $f(x)$ is mostly vertical, i.e. parallel to the y-axis. It is also possible to approximate the curvature as $\tilde{\kappa}\,(x) = f''(x)$ [24].

According to our model, parts are convex and part boundaries are manifest in three ways: i) as C^0 discontinuities where the boundary is an occluding contour, or ii) as concave C^1 discontinuities or iii) negative local minima of curvature depending on whether the intersection between parts is discontinuous or smooth. What is common to each of these features is that they correspond to concave local neighbourhoods where $\tilde{\kappa}$ is negative[1].

Therefore, the safest way therefore to preserve the positions of putative boundary features is to apply no deformation where $\tilde{\kappa}$ is negative. This translates into a $\beta(.)$ that is zero for a negative (.) argument. When $\tilde{\kappa}$ is positive, the section of the curve is convex and diffusion is required. The level of diffusion can be chosen to be linearly proportional to the estimate of the curvature. This means that $\beta(.)$ is linearly proportional to its argument when it is positive.

As a result, the curvature scale space for a curve $C_0(s)$ can be defined as

$$\begin{aligned} \frac{\delta C}{\delta t} &= \begin{cases} 0 & \text{if } \tilde{\kappa} < 0 \\ \tilde{\kappa}\vec{N} & \text{if } \tilde{\kappa} \geq 0 \end{cases}, \\ C(s,0) &= C_0(s), \end{aligned} \tag{2.2}$$

where $C_0(s)$ is a parametrization of the original $y = f(x)$ curve, $\tilde{\kappa}$ is an approximation of the curvature computed as the second derivative f_{xx} of the original curve and $\vec{N}$ is assumed to be parallel and in the direction of the negative y-axis.

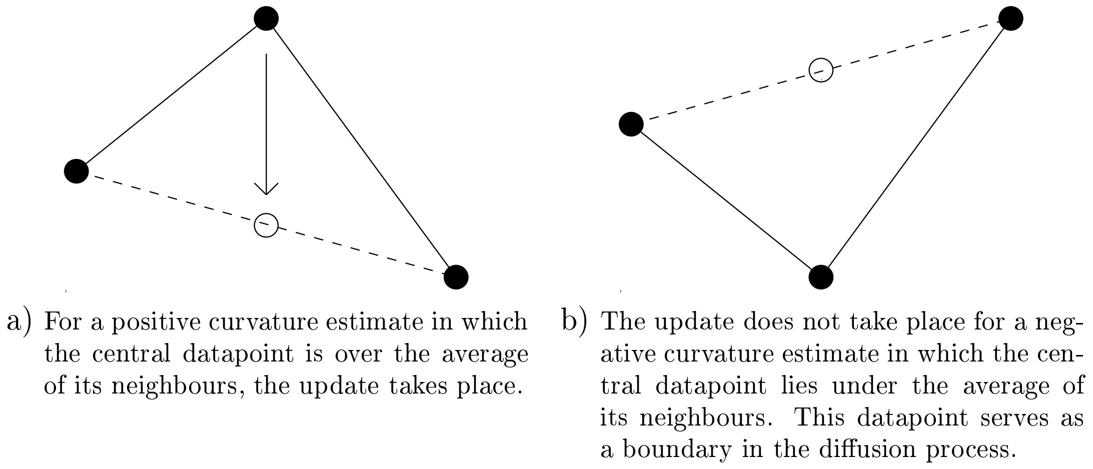
[1] Note that C^0 discontinuities always have a concave side where $\tilde{\kappa}$ is negative.

2.2 Discretization of the Scale Space in 1-D

The extension of this scale space to discretized data follows. The curve $y = f(x)$ is discretized into a sampled set $x[n] | n = 1, \ldots, N$. The second derivative $\ddot{x}[n]$ on this set is obtained with the kernel $[-\frac{1}{2}, 1, -\frac{1}{2}]$ and the scale parameter that was time now becomes a sampled scale, denoted as k. The diffused signal after k iterations $x_k[n]$ is given by

$$\begin{aligned} \ddot{x}_{k-1}[n] &= -\frac{1}{2}x_{k-1}[n-1] + x_{k-1}[n] - \frac{1}{2}x_{k-1}[n+1] \\ x_k[n] &= \begin{cases} x_{k-1}[n] & \text{if} \quad \ddot{x}_{k-1}[n] < 0 \quad \text{(concave region)} \\ x_{k-1}[n] - \ddot{x}_{k-1}[n] & \text{if} \quad \ddot{x}_{k-1}[n] \geq 0 \quad \text{(convex region)} \end{cases} \quad . \end{aligned} \tag{2.3}$$

Equation 2.3 can be explained with the aid of Figure 2.1. In the case of a positive curvature, Figure 2.1a, $\ddot{x}[n]$ is positive. Hence local smoothing is applied to this convex region which serves to displace the central point towards its neighbours. For the concave region shown in Figure 2.1b, $\ddot{x}[n]$ is negative in which case no smoothing is applied.



a) For a positive curvature estimate in which the central datapoint is over the average of its neighbours, the update takes place.

b) The update does not take place for a negative curvature estimate in which the central datapoint lies under the average of its neighbours. This datapoint serves as a boundary in the diffusion process.

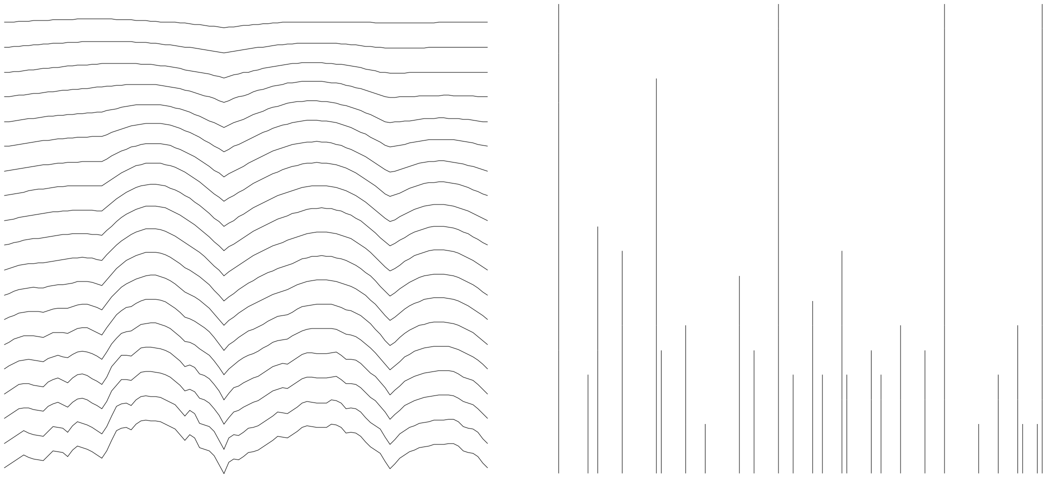
Figure 2.1: Two cases defining the diffusion process.

The level of diffusion is controlled by the number of times the operator is reapplied. Boundary conditions are set by the second case where the curve is concave, in which case no diffusion occurs. By design, this diffusion function can displace neither negative minima of curvature, nor discontinuities in depth (C^0), nor concave discontinuities in orientation (C^1). The signal is forced to pass through these points, hence they serve as boundary conditions. This condition remains in force until neighbouring points have diffused sufficiently to change their normal direction.

The implementation of our diffusion process, in Equation 2.3, may look similar to median or average filtering. By definition (Equation 2.2), it is different because the update value of each datapoint is function of the local curvature estimates of the curve to be diffused. Moreover the update only takes place at locations where the curvature estimate is positive, thus ensuring that feature locations are preserved. This is not the case in median and average filtering, in which such features would be displaced.

The effect of this diffusion process on a waveform that consists mostly of two large convex regions and two smaller ones is shown in Figure 2.2a. As the sequence begins at the bottom, the original waveform is not diffused. As it progresses to the top, the waveform becomes increasingly diffused. The intuitive effect of this operator is to flatten noise quickly, texture slowly and shapes that are mostly convex very slowly while leaving concave ones untouched. The overall effect is to bring out mostly convex shapes. Notice how the separation between each convex shape really stands out. Figure 2.2b shows the curvature scale space image, which for this diffusion process also corresponds directly to what Witkin called an interval tree [32]. At each scale, the signal could be segmented along the vertical lines.

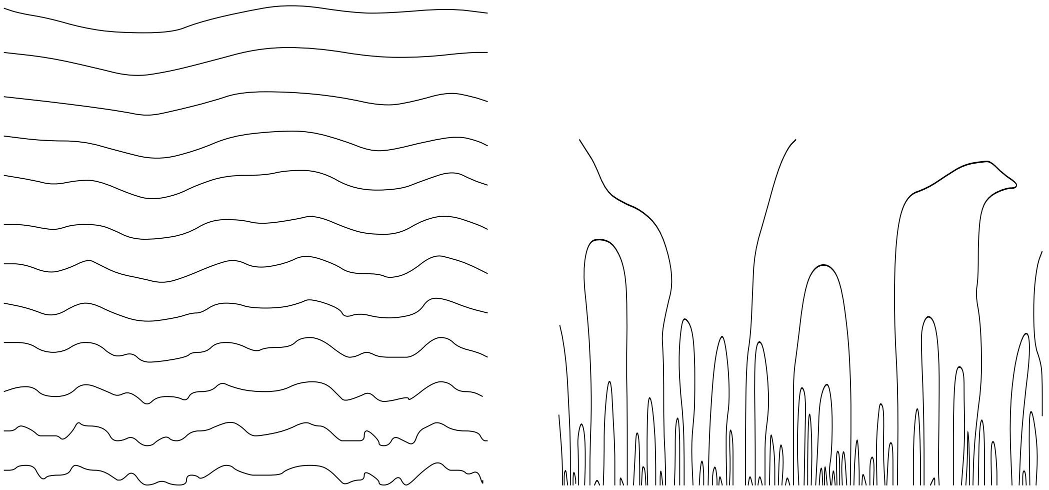
It can be noticed that Gaussian diffusion as in Figure 2.3a, is not tuned to convex shapes and indiscriminately smoothes the regions and their boundaries. As the diffusion progresses, the features of interest become more difficult to identify, they do not stand out as in Figure 2.2a. The scale space image for Gaussian diffusion in Figure 2.3b does not consist of vertical lines, but of curves. To properly locate features that survive to coarser scales, a coarse-to-fine tracking must be performed.



a) A sequence of waveforms from the non-diffused ones at the bottom to the most diffused ones at the top. As the waveform becomes diffused, noise and texture disappear, unveiling the more important structure.

b) The scale space image is the same as the interval tree for this diffusion process since the boundaries do not change position. Each line corresponds to a detected boundary through scale.

Figure 2.2: Scale space for an arbitrary waveform with noise and texture.



a) A sequence of Gaussian diffused waveforms. This diffusion process is not tuned to convex shapes and indiscriminately smoothes the regions and their boundaries.

b) Contours of zeros of the second derivative of the waveforms in a scale space image. These contours are curved (features change position) and a coarse-to-fine tracking algorithm is necessary to properly locate features at coarse scales and obtain an interval tree as in Figure 2.2b.

Figure 2.3: Scale space produced with Gaussian diffusion, adapted from Witkin [32].

2.3 2-D Case

In two dimensions, implementation of the diffusion operator is moderately more complex, given that curvature is a directional property of a surface [9]. The basic elements comprising the differential geometry can be described with the aid of Figure 2.4, which shows the local neighbourhood of a point P_i on a surface S. Let T_{P_i} be the plane tangent to P_i and $\Pi_{N_{Pi}}$ a plane orthogonal to T_{P_i} containing the unit normal vector to P_i, N_{P_i}. As $\Pi_{N_{Pi}}$ is rotated about N_{P_i}, it intersects S in a contour called a normal section. The curvature of the normal section at P_i is called the normal curvature, $\kappa_{N_{Pi}}$. There are two special directions in T_{P_i} for which $\kappa_{N_{Pi}}$ takes on maximum and minimum values, $\kappa_{M_{Pi}}$ and $\kappa_{\mathcal{M}_{Pi}}$, and are referred to as the principal directions M_{P_i} and $\mathcal{M}_{P_i}$ respectively [9]. The scalar quantities $\kappa_{M_{Pi}}$ and $\kappa_{\mathcal{M}_{Pi}}$ are similarly referred to as the principal curvatures at P. According to Sander [31], the set $(P_i, M_{P_i}, \mathcal{M}_{P_i}, N_{P_i}, \kappa_{M_{Pi}}, \kappa_{\mathcal{M}_{Pi}})$ is referred to as the *augmented Darboux frame* at P_i, i.e. $\mathcal{D}(P_i)$.

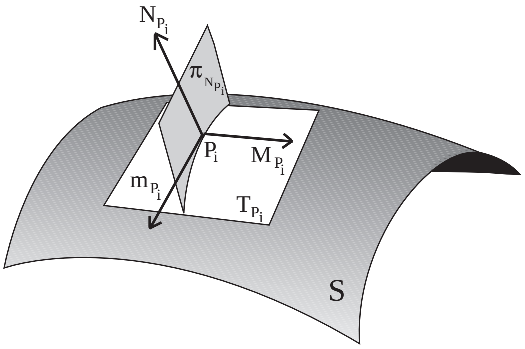


Figure 2.4: Local surface representation: the augmented Darboux frame.

For our diffusion process, it is sufficient to determine the signs of the two principal curvatures at each P_i. If both are positive, P_i will likely correspond to a convex surface where diffusion should be applied. Otherwise, the likelihood is that P_i corresponds to a putative part boundary where no diffusion should be applied, such as in the 1-D case. Accurate recovery of the elements of $\mathcal{D}(P_i)$ over S is possible with surface reconstruction algorithms[12, 13, 11]. However, when only the signs of $\kappa_{M_{Pi}}$ and $\kappa_{\mathcal{M}_{Pi}}$ are of interest, estimation of second directional derivatives at P_i can suffice.

It was found that a reasonable estimate of the signs of the principal curvatures can be determined from four estimates of the second directional derivative (*sdd*) in the vertical, horizontal and two diagonal directions.

As a result, the curvature scale space in 2-D is built from these four second directional derivatives. The diffusion process to produce this scale space is now described. The reader should be aware that surface samples as given by conventional range imaging systems are usually given in terms of the distance from the camera to the object. Accordingly, the z-axis is reversed with respect to usual conventions, as in the 1-D case in Section 2.1. In the following equations, this will be reflected by the direction of inequalities. Therefore, given a 2-D array of data $x[m][n]$, the diffused signal after k iterations $x_k[m][n]$is given by

$$\begin{aligned}
a &= \frac{1}{2}(x_{k-1}[m][n-1] + x_{k-1}[m][n+1]) \\
b &= \frac{1}{2}(x_{k-1}[m-1][n] + x_{k-1}[m+1][n]) \\
c &= \frac{1}{2}(x_{k-1}[m-1][n-1] + x_{k-1}[m+1][n+1]) \\
d &= \frac{1}{2}(x_{k-1}[m+1][n-1] + x_{k-1}[m-1][n+1]) \qquad (2.4)
\end{aligned}$$

$$e = \min(a, b, c, d) \qquad (2.5)$$

$$x_k[m][n] = \begin{cases} e & \text{if } x_{k-1}[m][n] < e \quad \text{(convex surface)} \\ x_{k-1}[m][n] & \text{if } x_{k-1}[m][n] \geq e \quad \text{(other types)} \end{cases} . \qquad (2.6)$$

Figure 2.5 shows how Equations 2.4, 2.5 and 2.6 are applied to a convex set of datapoints. Notice the direction of the z axis in the image.

Each of the a, b, c and d values in Equation 2.4 determines a possible update for each datapoint being processed, $x_k[m][n]$. They are equivalent to those in the 1-D case however, this time, their value is computed in the four directions previously discussed, i.e. horizontal, vertical and two diagonal directions. If each of a, b, c and

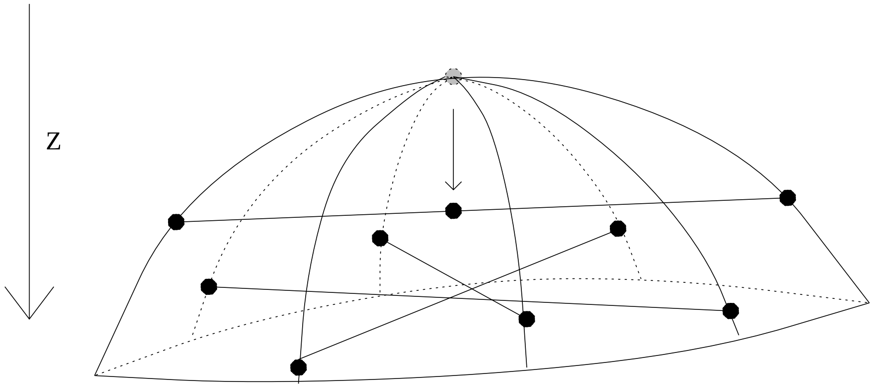


Figure 2.5: Effect of the diffusion equation on a convex surface.

d is larger than $x_k[m][n]$, the surface is convex at this datapoint and diffusion should take place. Otherwise, the surface is not convex and no diffusion should take place. Indeed, the second directional derivatives are given by

$$\begin{array}{rcll} SDD_h & = & a - x_k[m][n] & \text{(horizontal case)} \\ SDD_v & = & b - x_k[m][n] & \text{(vertical case)} \\ SDD_{d1} & = & c - x_k[m][n] & \text{(first diagonal)} \\ SDD_{d2} & = & d - x_k[m][n] & \text{(second diagonal)} \end{array} . \tag{2.7}$$

A convex surface would have all SDD's positive, which is equivalent to each of a, b, c and d being larger than $x_k[m][n]$. Since the value e computed in Equation 2.5 equals the smallest of a, b, c and d, all that is required to test for a convex surface is that e be larger than $x_k[m][n]$. Ergo, e serves as a test for convexity and can also be used as an update value for the datapoint being processed, provided the surface is convex. This update value is the most conservative of all a, b, c and d, ensuring that the region stays convex even after being diffused.

In the 2-D case, the level of diffusion is also controlled by the number of times the diffusion process is applied. The positions of boundaries marked by negative minima of curvature, (C^0) and concave (C^1) discontinuities are preserved as in the 1-D case, in this way avoiding the problem of coarse-to-fine tracking, a technique that is both complicated and expensive [8, 29].

At any scale, a region segmentation is easily obtained by labelling every connected set of datapoints where the surface is found to be convex (from Equation 2.6). Because the whole image cannot be made only of convex surfaces, there will be datapoints left unlabelled. Since our definition of a boundary consists of C^0, concave C^1 discontinuities and negative local minima of curvature, the labelled regions can be grown until they reach these boundaries and all datapoints are labelled. Contours of the resulting labelled regions are places where the image could be segmented. These contours exist at the finest scale and slowly disappear as the scale is made coarser. While in 1-D, the interval tree was a 2-D image containing lines, the equivalent in a 2-D scale space is a volume containing cylinder-like surfaces, as shown in Figure 2.6.

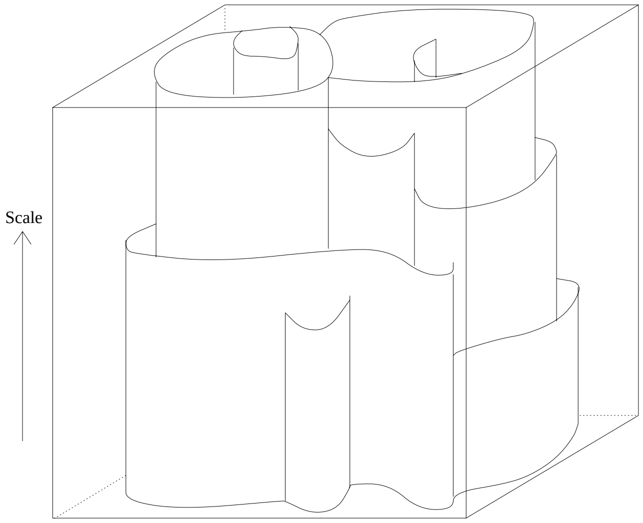
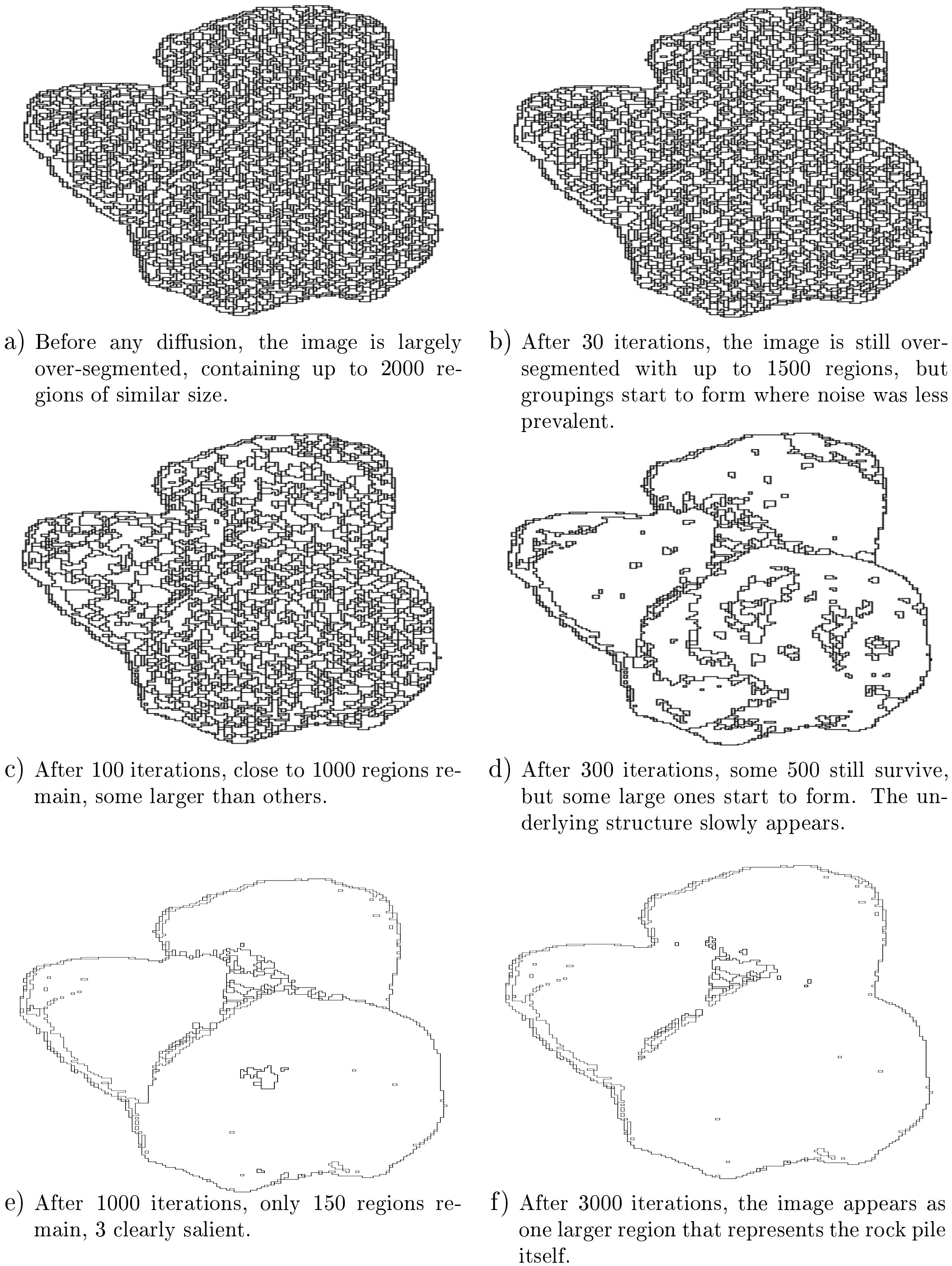


Figure 2.6: Interval tree in 2-D.

A few slices of such a volume are shown for the rock pile in Figure 2.7 as contour maps. These slices were taken at regular intervals on a logarithmic scale. In Figure 2.7a, the region partitioning algorithm can detect a structure in the image. For this reason, the image shows a multitude of contours forming tiny regions. As diffusion is applied, the same region partitioning algorithm finds fewer regions, but these are larger, as shown by the subsequent images. In Figure 2.7e, three large regions are found, corresponding to the three rocks. Finally, in the last image, only the pile itself is seen, as expected at very coarse scales.



a) Before any diffusion, the image is largely over-segmented, containing up to 2000 regions of similar size.

b) After 30 iterations, the image is still over-segmented with up to 1500 regions, but groupings start to form where noise was less prevalent.

c) After 100 iterations, close to 1000 regions remain, some larger than others.

d) After 300 iterations, some 500 still survive, but some large ones start to form. The underlying structure slowly appears.

e) After 1000 iterations, only 150 regions remain, 3 clearly salient.

f) After 3000 iterations, the image appears as one larger region that represents the rock pile itself.

Figure 2.7: Contours of segmented regions through the scale space of a rock pile.

2.4 Traversing through the scale space

In general, features in an image exist at many different scales. The problem of automatically selecting the appropriate scale for each feature is a difficult one. We consider a simpler problem by assuming that all the features of interest can be represented at a single scale. These are represented by the feature map $f[m][n]$. This condition is true for many man-made objects like those considered in this report. The problem which remains is that of choosing the best single scale to interpolate the feature set. Because the region contours in our scale space do not change position in our scale space, as shown in Figures 2.2, 2.6, 2.7, it is always possible to match every feature point with the appropriate region contour. Thus, to interpolate a given feature set, we choose the coarsest scale in which every feature $f[m][n]$ still matches with a region contour.

As an example of this problem, Figure 2.8 represents a feature set to be interpolated. Each feature in this image is represented by a small line or a small square. Figure 2.7 shows contours at different scales that can be used to interpolate this feature set. The first five samples, Figures 2.7a through e contain contours that completely overlap with the feature set, while the last sample, Figure 2.7f does not. Therefore, the sample that should be used to interpolate the feature set is the one shown in Figure 2.7e because it consists of the coarsest scale in which the contours completely overlap the feature set.

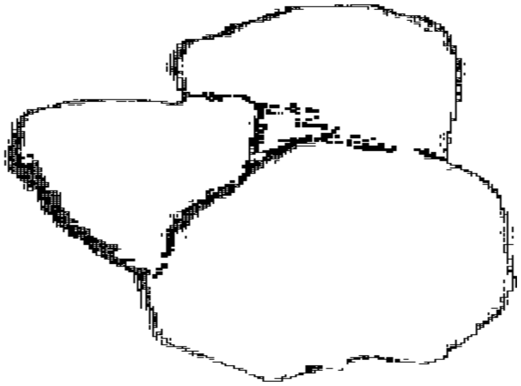


Figure 2.8: Feature set to be interpolated.

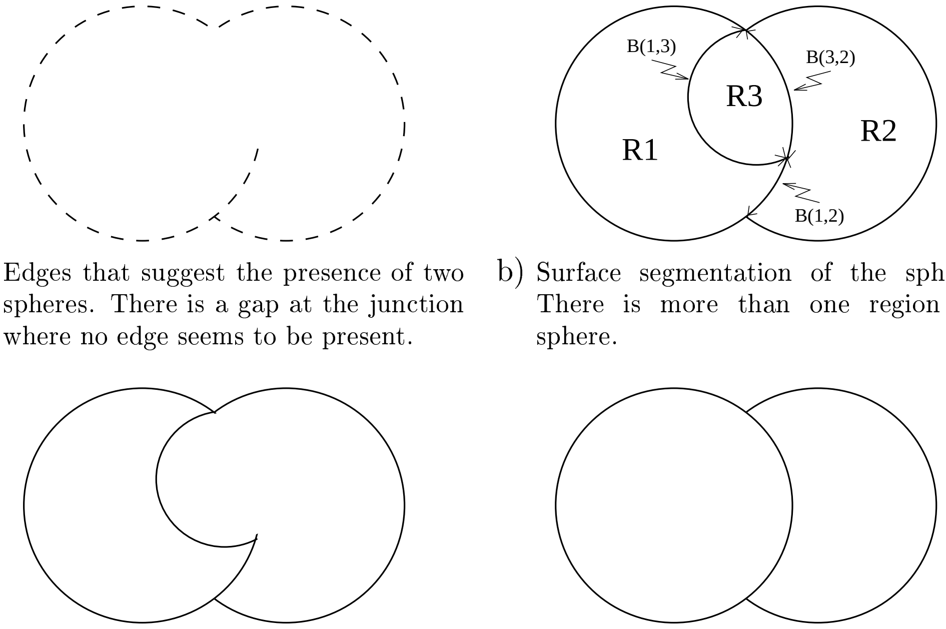
Section 3 Resolution Algorithm

Given a set of features, $f[m][n]$, corresponding to putative boundary points and a surface segmentation, taken from the scale space, that implicitly defines the space of boundary contours, the task is now to determine how regions can be merged so that the resulting boundary contours are consistent with this feature set. In fact, the process of merging two surface regions is also one of interpolation where union of the two boundaries results in a new contour that extends further over the surface. For some optimal set of surface region combinations, there exists a set of contours that best interpolates (is most consistent with) the set of discrete features. This section describes how the determination of this set can be formulated as an optimization problem.

3.1 Merging as an Optimization Problem

In Figure 3.1a a set of putative boundary points is shown that suggest either two overlapping spheres or a peanut shape with a crease in the lower half. Figure 3.1b shows a segmentation of the corresponding surfaces — notice that three distinct regions are present. Different interpretations are possible through different merges, e.g., R3 ∪ R2 (Figure 3.1c), R1 ∪ R3 (Figure 3.1d), R1 ∪ R2 (not shown) and of course R1 ∪ R2 ∪ R3.

Local overlap between region contours and feature points serves to guide the merging process. For example, the merging of regions R1 and R3 is encouraged because there is no support for their common boundary B(1,3) in the feature set shown in Figure 3.1a. Similarly, R2 ∪ R3 is plausible since there is no support for B(3,2). On the other hand, R1 ∪ R2 is discouraged because B(1,2) is supported by overlapping features.



a) Edges that suggest the presence of two spheres. There is a gap at the junction where no edge seems to be present.

b) Surface segmentation of the spheres. There is more than one region per sphere.

c) Possible covering that is consistent with both a) and b). The contours overlap with all edges in a) and are made from a subset of b).

d) A second consistent covering that is optimal with respect to c) because shorter contour segments were used to overlap with the edges.

Figure 3.1: Possible coverings for two spheres.

The term *covering* is used to represent the union of regions that is consistent with a feature set. Therefore, Figures 3.1c and d constitute possible coverings of the feature set in Figure 3.1a. The decision on whether to merge regions is given by a global cost function that determines the convergence of the merging algorithm. Such a global cost function is based on the length of the unsupported segments to keep, shorter segments being preferred. Figure 3.1d consists of the optimal covering with respect to the global cost function because shorter boundary segments are used to interpolate the feature set.

We use the term *resolution* to underline the fact that at convergence the process has resolved the region and edge-based representations into a single consistent interpretation of an object or scene. Continuous relaxation labelling, as defined by Hummel and Zucker in 1983 [20] applies directly to this problem, as it is a global optimization process based on local interactions.

3.2 Overview of the Relaxation Labelling Algorithm

An optimization problem can be formulated as a continuous relaxation labelling process by constructing a network in which the nodes correspond to the entities for which an optimal set of properties or parameters must be determined. The latter are represented by a set of discrete labels, one for each instance of a property or parameter value. To each label is affixed a numerical value that represents its associated likelihood. In a problem with n nodes and m properties, this likelihood is denoted as $p_i(\lambda)$ where $i = 1, ..., n$ and $\lambda = 1, ..., m$. Thus, $p_i(\lambda)$ is a matrix whose elements represent the likelihood that node i possesses property λ.

In this network, arcs connecting nodes are implicitly represented by constraints that relate how these likelihoods should change given the current states (labels + likelihoods) of neighbouring nodes. These constraints, denoted by $r_{ij}(\lambda, \lambda')$ define how a node j with label λ' should influence the node i with label λ. This influence is called a support function and is simply computed as

$$s_i(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'}^{m} r_{ij}(\lambda, \lambda') p_i(\lambda). \qquad (3.1)$$

Once the support is computed, an update for each node can be computed by projecting the support vector onto the valid likelihood space [24]. The projection scheme used in this paper was proposed by Mohammed, Hummel and Zucker [27] and is called *Gradient Projection Algorithm.*

After updating of all nodes has taken place, a new labelling is found. The network is allowed to iterate until the likelihoods at each node converge to a steady-state, at which point the ordering of the labels at each node as a function of their likelihoods represents the optimal solution. The global function A that is being optimized is a measure of consistency between the nodes as defined by the constraints, i.e.,

$$A = \sum_{i,j,\lambda,\lambda'} r_{ij}(\lambda,\lambda')p_i(\lambda)p_j(\lambda'). \tag{3.2}$$

3.3 Merging as a Relaxation Labelling Network

Formulation of the resolution (merging) problem follows directly. Nodes in the relaxation network correspond to each region in the scale space representation and labels are unique numbers that identify each single region. Constraints between nodes are defined in terms of the feature set. The process of merging two regions can be represented by increasing the likelihoods of labels that are common to both regions and decreasing those corresponding to labels that are mutually exclusive. Details are elaborated in the following sections.

3.4 Labels

Given an array $I[m][n]$ completely covered by n non-overlapping regions S_i

$$I = \sum_{i=1}^{N} S_i,\ \ S_i \cap S_j = \emptyset\ , \forall i,j\ i \neq j. \tag{3.3}$$

A label λ belonging to the positive integer set N^* is assigned to each region and associated elements so that each has a different label and all elements belonging to the same region have the same label

$$\lambda(S_i) \neq \lambda(S_j) \text{ iff } i \neq j, \tag{3.4}$$

$$\lambda(I[m][n]) = \lambda(I[s][t]) \text{ iff } I[m][n],\ \ I[s][t] \in Si. \tag{3.5}$$

3.5 Constraints

The constraints between labels have four dimensions: $r_{ij}(\lambda, \lambda')$ is used to denote how the likelihood of label λ' at region S_j influences the likelihood of label λ at region S_i. Four important considerations reduce this 4-Dimensional array to a 2-Dimensional matrix symmetric along the $i = j$ diagonal with very sparse non-zero elements.

First, there is no need to consider the effect of two regions having different labels, λ and λ'. We are only interested in the case where two regions could have the same label λ. If two regions have different labels, there is no need to encourage or discourage a specific label more than another. Thus,

$$r_{ij}(\lambda, \lambda') = 0 \; \forall \lambda \neq \lambda'. \tag{3.6}$$

Second, when considering the case where two regions could have the same label, there is no need to either encourage or discourage a specific label more than another. That is, the particular label with which the two regions end up is unimportant. Therefore,

$$r_{ij}(\lambda, \lambda) = const \; \forall \lambda. \tag{3.7}$$

However, $r_{ij}(\lambda, \lambda)$ does vary with respect to i and j. It is only necessary to compute and retain r_{ij}, resulting in a 2-D matrix.

Third, it should be noted that constraints exist only between adjacent regions. The r_{ij} are non-zero only when regions touch each other. For a given region, there is only a limited number of adjacent ones, much less than the total number of regions in general. This leads to a very sparse matrix.

Finally, constraints are reciprocal. The attraction or repulsion of a region S_i towards S_j is the same as that of S_j towards S_i. Thus, the matrix is symmetric along the $i = j$ diagonal. The resulting constraints can now be represented by a symmetric

2-D sparse matrix with approximately $\frac{5n}{2}$ elements, where n is the total number of labels and 5 the average number of adjacent regions to a given one.

The problem remains as to how to assign values to constraints. Various schemes are possible as there are few restrictions on their magnitudes. In general, positive values are used to encourage the merging of adjacent regions whose common boundary is not supported by any cue (i.e. element of the feature set). Negative values do exactly the opposite, inhibit merging where the common boundary is supported by such cues. The following consists of a possible constraint assignment, but many other assignments are acceptable.

Given cues $H[m][n]$ and $V[m][n]$ that lie directly on the boundary of adjacent regions where

$$H[m][n] = \begin{cases} 1 & \text{if a horiz. edge exists between } I[m][n] \text{ and } I[m+1][n] \\ 0 & \text{otherwise} \end{cases}, \quad (3.8)$$

$$V[m][n] = \begin{cases} 1 & \text{if a vert. edge exists between } I[m][n] \text{ and } I[m][n+1] \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

and given the boundary path B, the length of the path L and the overlap C, which consists of the number of detected cues that match with a given boundary,

$$B(S_i, S_j) = \text{The boundary path of adjacent regions } S_i \text{ and } S_j, \quad (3.10)$$

$$L_{ij} = \text{length of } B(S_i, S_j), \quad (3.11)$$

$$C_{ij} = \sum_{B(S_i,S_j)} (H[m][n] + V[m][n]). \quad (3.12)$$

The constraints r_{ij} are given by

$$r_{ij} = \begin{cases} L_{ij} & \text{if } C_{ij} = 0 \\ -C_{ij} & \text{if } C_{ij} > 0 \end{cases}. \quad (3.13)$$

3.6 Determining Initial Conditions

Because relaxation labelling is in fact a gradient ascent optimization method, it is necessary to provide an initial condition to the labelling $p_i(\lambda)$. The closer the initial condition is to the final optimal result, the less chance of becoming stuck at a local maximum. On the other hand, it is possible to start far away from the optimal solution by using a method analogous to simulated annealing.

The matrix $p_i(\lambda)$ is initialized by assigning a constant value to each element and then adding noise to avoid numerical instabilities. The type of noise, Gaussian or uniform, is not critical. The level σ is kept small with respect to the initial constant value. Also, a bias is added to the $i = \lambda$ diagonal. This bias is used to bring the initial condition closer to the final solution. Finally, each vector $\overline{p}_i$ is normalized for each i. Therefore,

$$p_i(\lambda)_1 = 1 + \sigma * noise \; \forall i, \lambda, \tag{3.14}$$

$$p_i(\lambda)_2 = \begin{cases} p_i(\lambda)_1 + bias. & \text{if } i = \lambda \\ p_i(\lambda)_1 & \text{otherwise} \end{cases}, \tag{3.15}$$

$$p_i(\lambda) = p_i(\lambda)_2 / \sum_{\lambda} p_i(\lambda)_2 \; \forall i. \tag{3.16}$$

Now that the labels, the constraints and the initial conditions have been assigned, continuous relaxation labelling can be applied as explained in Section 3.2. Upon convergence, the labels assigned to each node are ordered according to their updated likelihood values. The label assigned to each node is that with the highest likelihood. Merges are detected simply by noting adjacent nodes with the same label.

Section 4 Implementation and Results

4.1 Implementation

To evaluate the algorithms described in the previous sections, test bed environments were implemented on SUN SPARC and Silicon Graphics platforms. Range images were typically of the order of 256×256 at 10-bits/pixel of depth resolution. The data was processed in four steps: a pre-processing step to correct gross measurement errors from the range finding system; then the diffusion step, as presented in Chapter 2, to bring out the underlying structure in the image; then a region segmentation and part boundaries detection step; and finally the resolution step, as presented in Chapter 3, to validate region contours with part boundaries.

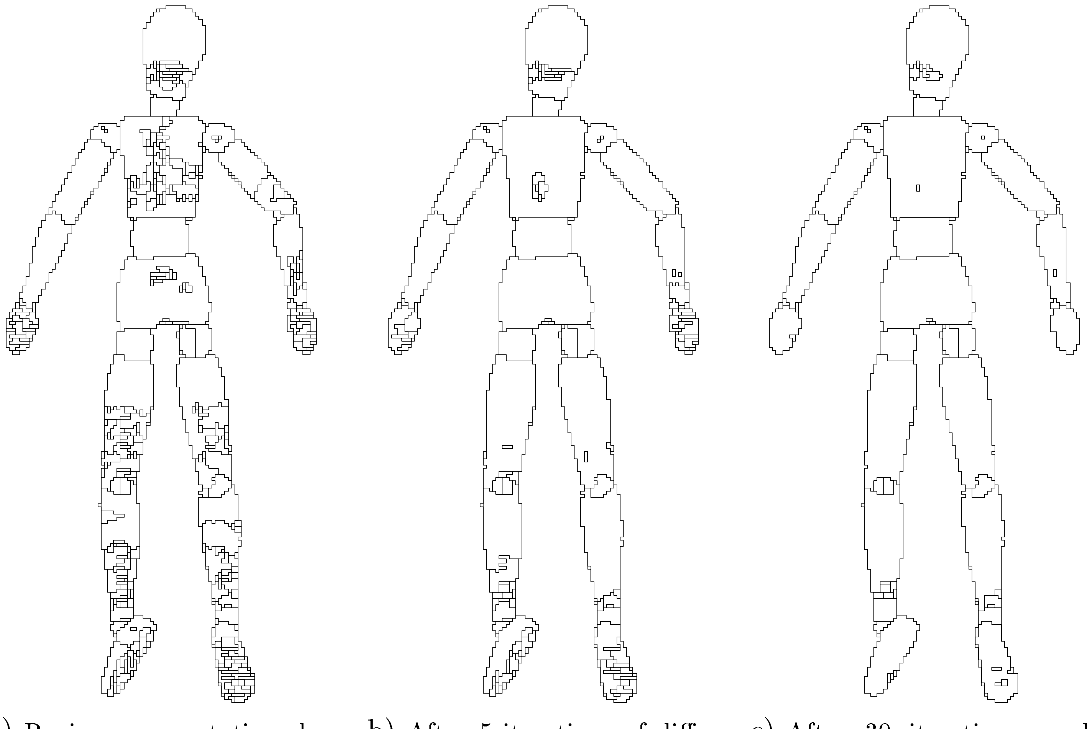
4.2 Results

Fifteen range images were successfully partitioned with very little parameter adjustments [24]. These images came from three different sources: The Canadian National Research Council in Ottawa [30], the Pattern Recognition and Image Processing Laboratory in Michigan [23] and the McGill Research Center for Intelligent Machines in Montreal. The images span a wide variety of object types in terms of noise, texture and number of constituting parts. The results from the scale space and the resolution algorithms will now be presented for some of the images.

4.3 Scale Space Results

Samples of the scale space representation for two different images will now be presented. For each of these images, the samples of the scale space were taken at regular intervals on a logarithmic scale. The images shown consist of the closed contours of the segmented regions at each sampled scale.

In Figure 4.1, the samples of the scale space for a wooden doll were taken at four different scales. Figure 4.1a shows the contours of the region segmentation before any diffusion was applied. The over-segmentation of the doll over the limbs, chest and head is caused by noise measurement and quantization error. For example the chest, which is mostly convex, locally contains non-convex features that obscure its general structure. The second sample in Figure 4.1b is taken after five iterations of the diffusion algorithm. It shows that noise is rapidly eliminated to make room for larger regions that correspond better to each part. The chest is less segmented. Figure 4.1c is the third sample in the scale space, taken after 30 iterations. Here, there is almost a one-to-one correspondence between the parts and the regions. Some parts, such as the legs, are still over-segmented, these parts not being entirely convex.



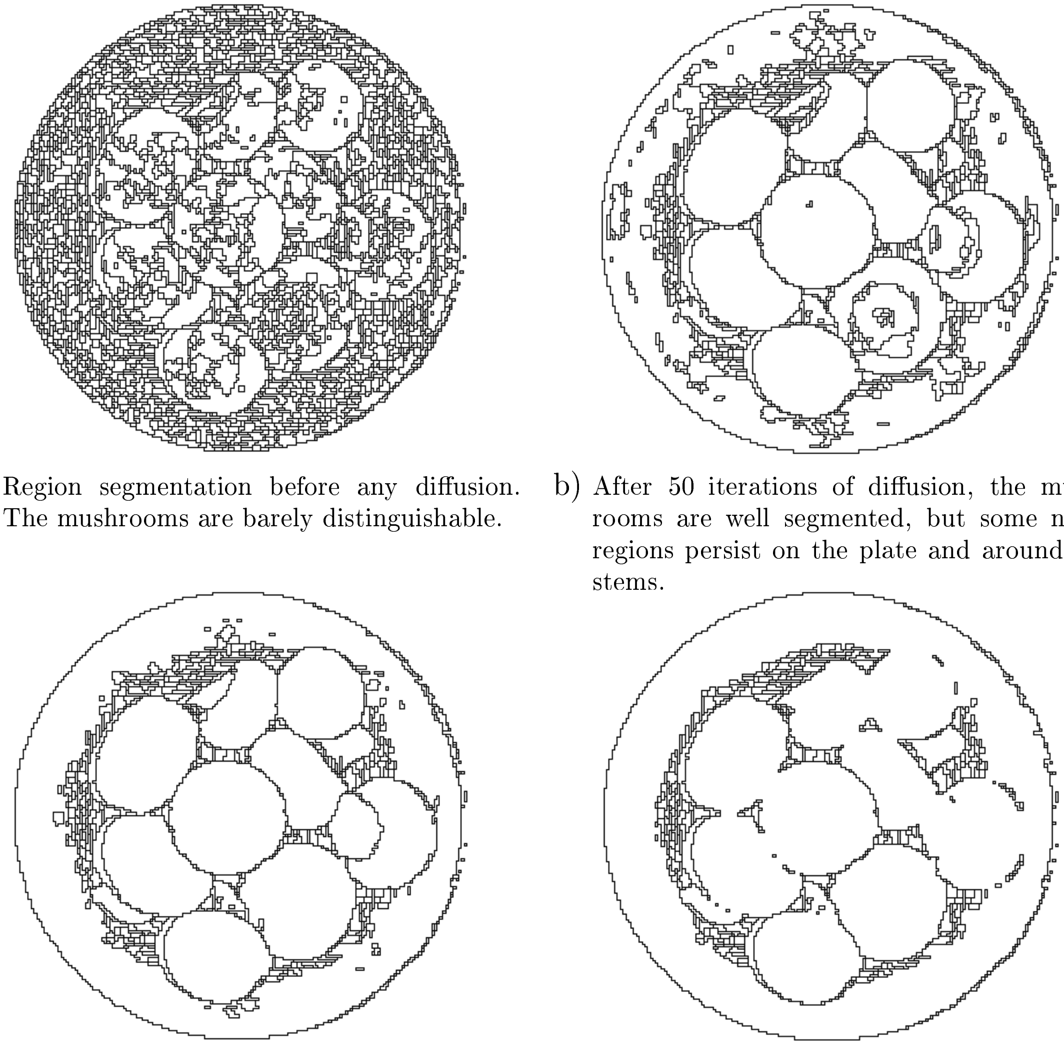
a) Region segmentation before any diffusion. The parts of the doll are over-segmented because of noise on the surface.

b) After 5 iterations of diffusion, some noise still causes over-segmentation on some parts.

c) After 30 iterations, each part consists mostly of a single region.

Figure 4.1: Curvature scale space of the CNRC doll.

A similar scale space is presented in Figure 4.2 for the range images of a plate of mushrooms. Also, samples of the scale space for a rock pile were presented in Chapter 2, Figure 2.7. For these two scale spaces, the same type of behavior as with the doll can be observed: noise initially causes an over-segmentation; after a few iterations of diffusion, the texture of the object is made explicit; after more diffusion, there is almost a one-to-one correspondence between the regions and the parts that constitutes the objects.



a) Region segmentation before any diffusion. The mushrooms are barely distinguishable.

b) After 50 iterations of diffusion, the mushrooms are well segmented, but some noisy regions persist on the plate and around the stems.

c) After 300 iterations, small regions still survive on the plate because it is a smooth concave region rather than a convex one.

d) After 1500 iterations, more regions have merged together. We approach the figure-ground (plate-background) separation.

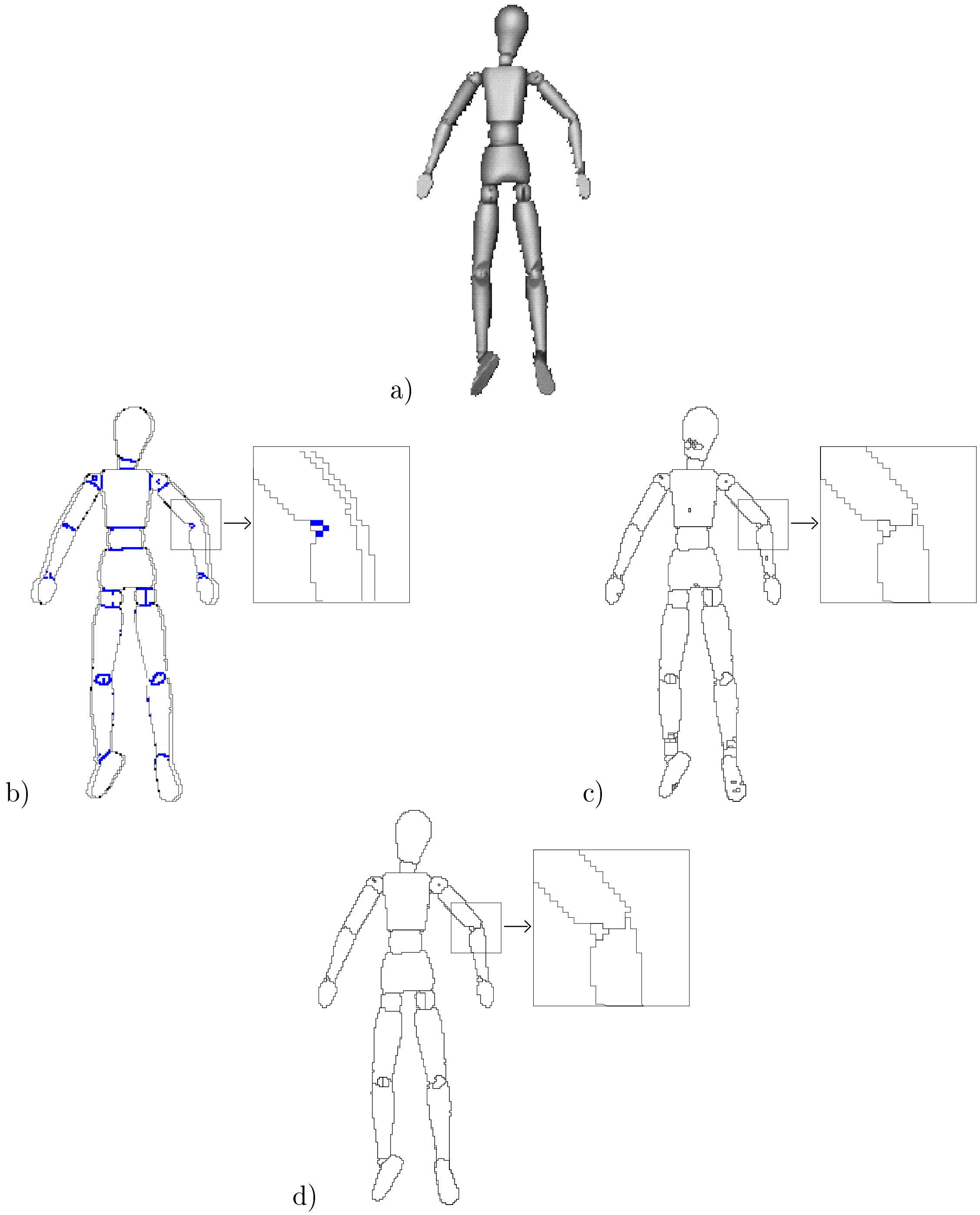
Figure 4.2: Curvature scale space of the CNRC mushrooms.

4.4 Results from the Resolution Algorithm

After diffusion, edge and region extraction, the resolution algorithm presented in Chapter 3 was applied to the range images. In the two examples that follow, it will be shown how the extracted edges support the presence of part boundaries and how non-supported boundaries are removed through the merging of adjacent regions.

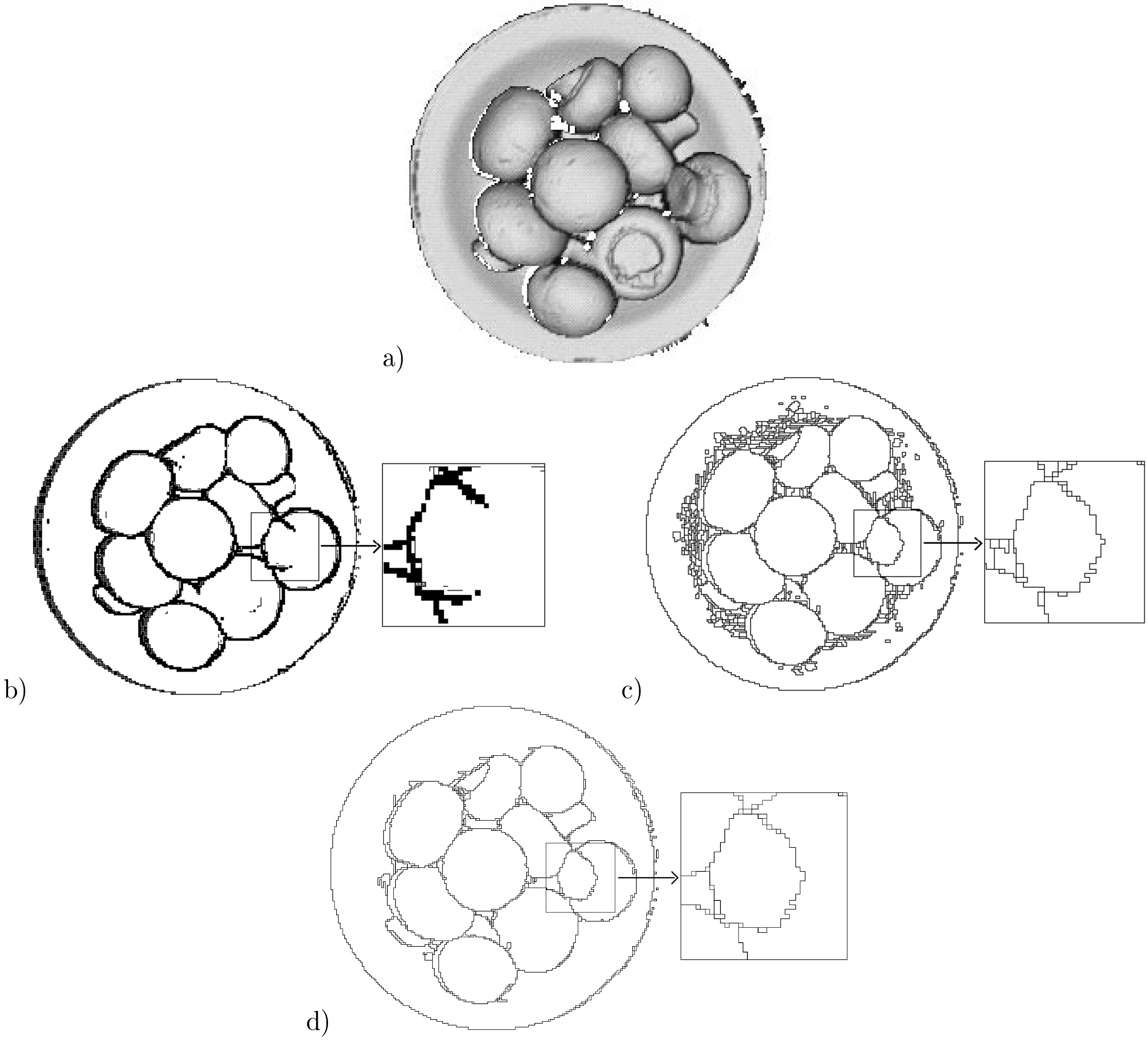
For the wooden doll of Figure 4.3a, edges are extracted (4.3b) and are used to validate the region segmentation (4.3c). The resulting partition is shown in Figure 4.3d. The blowup in 4.3b shows extracted edges in the doll's elbow. On each side of the arm, enough edges have been detected for a simple interpolation algorithm to connect them, but only a few edges separating the forearm from the arm have been detected. An interpolation algorithm would fail completing this segment. The blowup in 4.3c shows the contours of the region segmentation for the same elbow. These contours are connected segments and will serve in interpolating the edges on each side of the arm and on the arm/forearm separation. The blowup of 4.3d shows the partition that is most consistent with both 4.3b and c. In this example, all previously extracted regions are kept as they are all supported by edge features. The arm/forearm separation has been properly interpolated. A small region has also been identified: it is the elbow section that is made of a very small wood sphere.

Figure 4.4 shows that the resolution algorithm not only interpolates detected edges but also merges regions whose common boundary is not supported by any edge feature. The shaded range image of a plate of mushrooms is presented in Figure 4.4a. The blowup in 4.4b shows that extracted edges suggest the presence of a stem, but here also, interpolation is required. The segment that will perform this interpolation will be provided by region segmentation as shown by 4.4c in the blowup. On the other hand, this image clearly consists of an over- segmentation. Many region boundaries are not supported by any edge feature. The blowup in Figure 4.4d shows that the stem has been properly interpolated while the regions whose common boundary was not supported have been merged to form a single region.



a) Rendered original. b) Detected edges after 30 iterations of diffusion. Blowup of the elbow shows enough cues to suggest segmentation of the forearm, but interpolation is necessary to obtain two closed regions. c) Region segmentation taken from the scale space. It provides the necessary contour to interpolate the cues in the elbow. Also note that there are contours in each leg that are not supported by any cues. d) Merging, 39 groups found, 28 groups had over 5 datapoints. The expected interpolation occurred in the elbow and regions have been properly merged in each leg.

Figure 4.3: Merging results of CNRC doll, 256x256.



a) Rendered original. b) Detected edges after 300 iterations of diffusion. Blowup shows cues that support the presence of a stem. c) Region segmentation taken from the scale space. Boundary of the stem is present along with other boundaries not supported by the cues. d) After merging, 149 groups found, 18 groups had over 5 datapoints. The stem is properly identified and most unsupported boundaries have been removed.

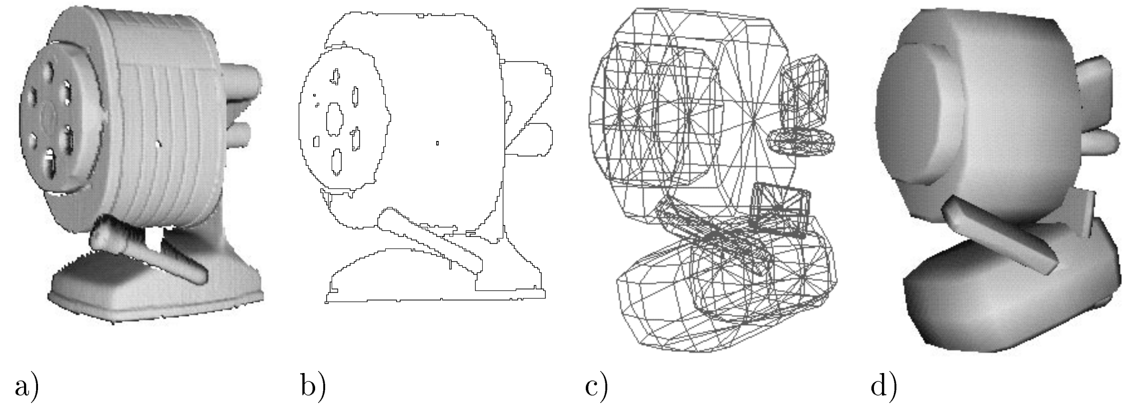
Figure 4.4: Merging results of the CNRC mushrooms, 256x256.

4.5 Results Summary

To summarize this chapter, results for ten range images are presented in a condensed format. For each range image, four figures are presented. The first figure consists of the original rendered range image. The second is the final segmentation, after the resolution algorithm has been applied and regions with few datapoints have been filtered out. The next two figures show how the resulting representations can serve as a basis for computing object models.

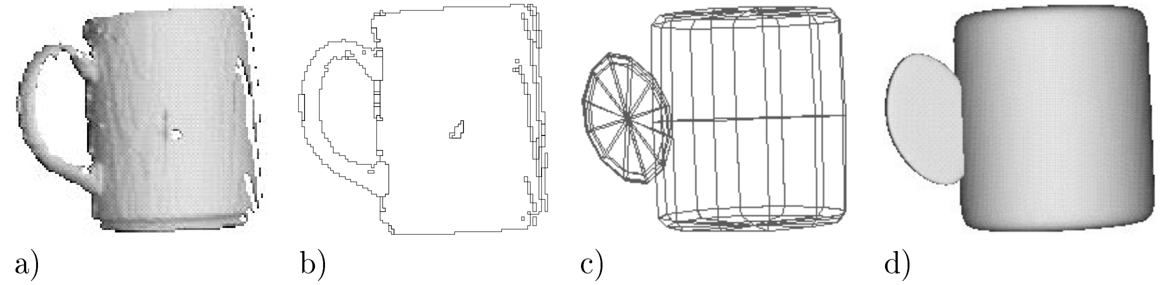
For each final partition, a new rendering is determined by fitting superquadric primitives to the data corresponding to each surface region. Superquadric fitting algorithms are often very sensitive to the distribution of datasets to be fitted. They converge with small residual error when the input dataset corresponds to a sampling of the superquadric model surface to which Gaussian noise is added. The residual error is large when the partition is wrong and the datasets are non-convex. The badly partitioned datapoints become outliers in the dataset and cause large residual errors in the fit. Unless robust fitting algorithms are used for modelling the parts in a range image, the partitioning algorithm must determine the exact location of the part boundaries. The last two figures that are presented for each range image show a mesh of the fitted superquadrics and the rendered equivalent. The resemblance of the rendered models to the original object can be used as a qualitative test of the partitioning algorithm.

Four types of range images are presented. Common objects are shown first in Figures 4.5 to 4.8: a pencil sharpener, a cup, a bowl of mushrooms and a fruit assortment. In this last example, bananas are properly partitioned, but superquadric models are not suitable for their bent shape. Deformable superquadrics should be considered.



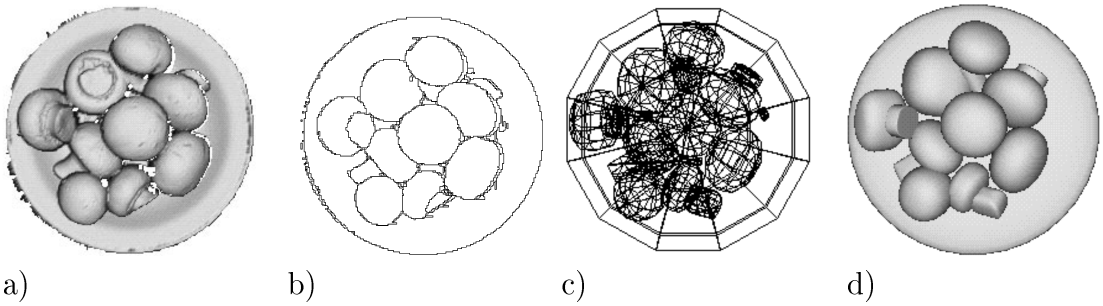
a) Rendered original. b) After 100 iterations of diffusion and merging, 20 groups were found, 8 groups had over 25 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

Figure 4.5: Partition of the CNRC sharpener, 256x256.



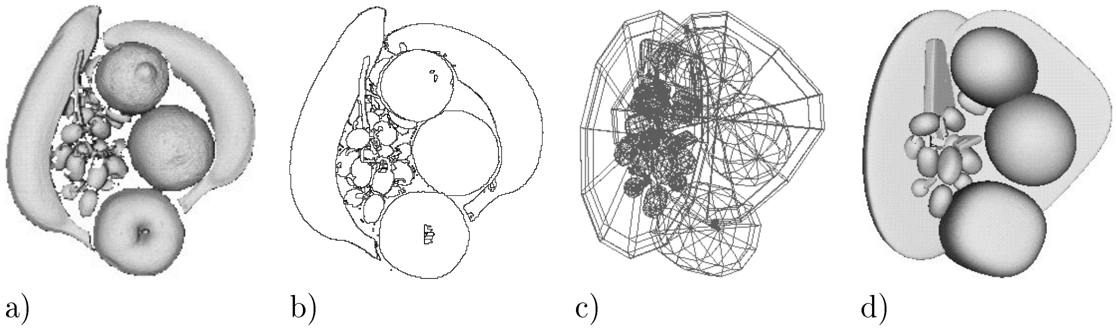
a) Rendered original. b) After 100 iterations of diffusion and merging, 32 groups were found, 2 groups had over 25 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

Figure 4.6: Partition of the Michigan PRIP cup, 240x240.



a) Rendered original. b) After 300 iterations of diffusion and merging, 149 groups were found, 18 groups had over 25 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

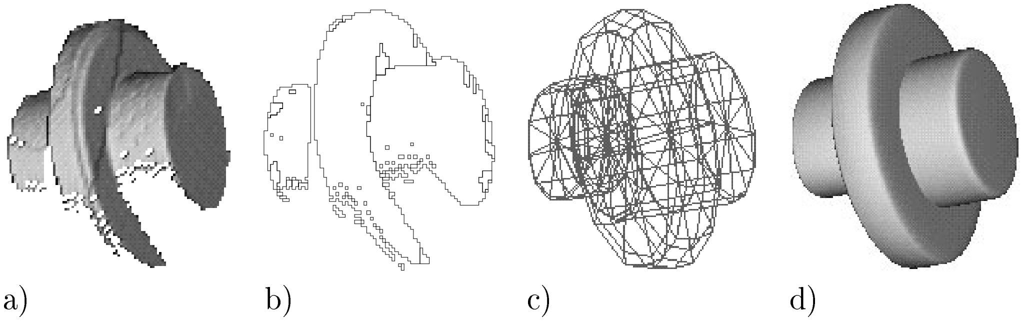
Figure 4.7: Partition of the CNRC mushrooms, 256x256.



a) Rendered original. b) After 30 iterations of diffusion and merging, 322 groups were found, 42 groups had over 25 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

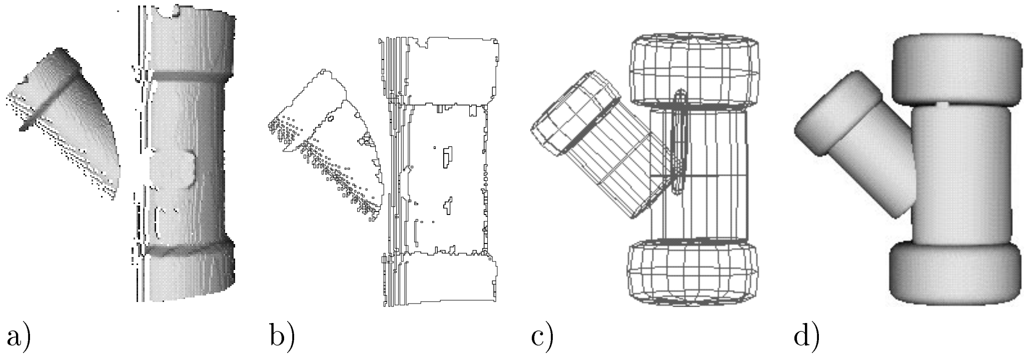
Figure 4.8: Partition of the CNRC fruit assortment, 256x256.

The second set of range images consists of mechanical parts that could be encountered along an assembly line. The final superquadric models could be used for recognition and grasping of the objects. These are shown in Figures 4.9 to 4.10. They consist of adaptors and a Y-shaped pipe.



a) Rendered original. b) After 100 iterations of diffusion and merging, 38 groups were found, 3 groups had over 25 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

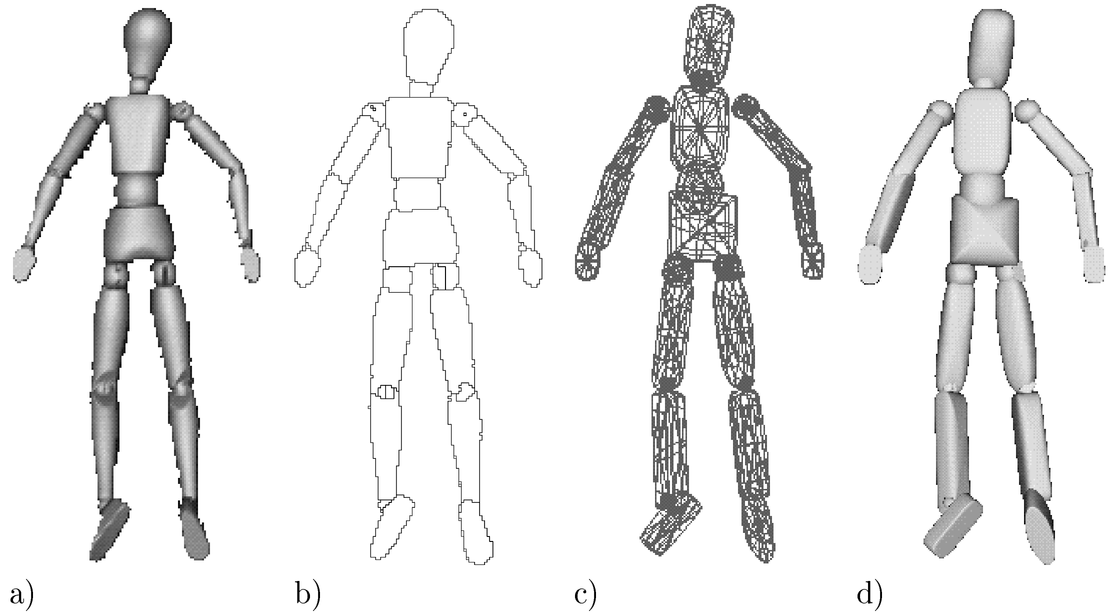
Figure 4.9: Partition of a third Michigan PRIP component, 240x240.



a) Rendered original. b) After 100 iterations of diffusion and merging, 200 groups were found, 6 groups had over 25 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

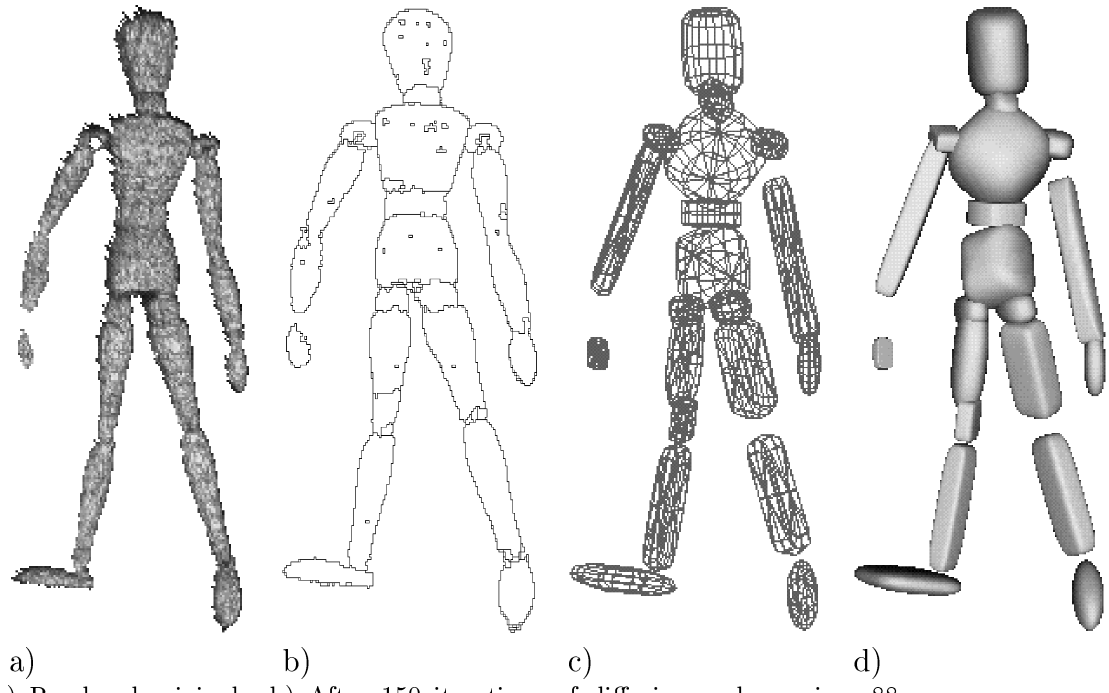
Figure 4.10: Partition of a Michigan PRIP Y-pipe, 240x240.

The third set of images presents toys: a wooden doll obtained from the CNRC and another one obtained from the McRCIM laboratory. Although the pose and amount of noise differ noticeably for the two examples, the final representations do not differ drastically. These images are shown in Figures 4.11 to 4.12.



a) Rendered original. b) After 30 iterations of diffusion and merging, 39 groups were found, 28 groups had over 5 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

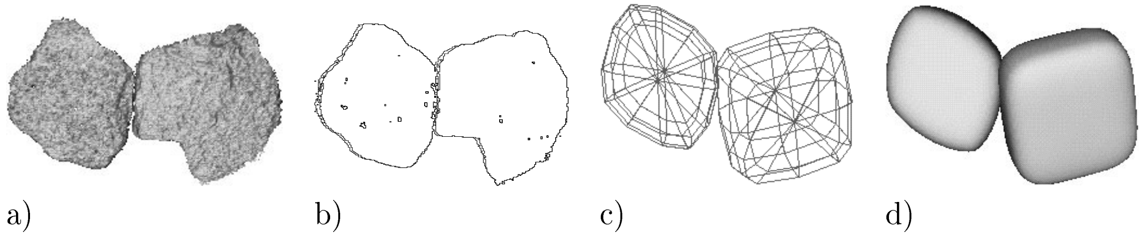
Figure 4.11: Partition of the CNRC doll, 256x256.



a) Rendered original. b) After 150 iterations of diffusion and merging, 88 groups were found, 20 groups had over 5 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

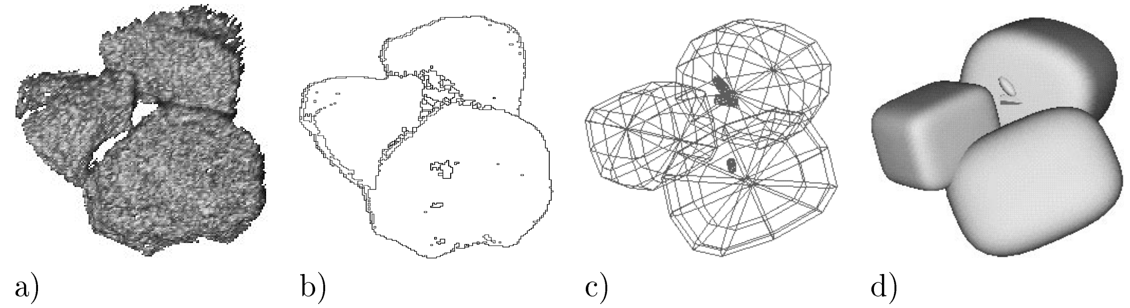
Figure 4.12: Partition of the McRCIM doll, 255x256.

The last set consists of mineral samples. In these, the resolution algorithm did not improve the partitioning significantly. A filtering algorithm, which retains the larger regions from the region segmentation, would have sufficed. Some of these examples could not easily be partitioned by previous algorithms based on a Gaussian scale space [6, 4, 5, 3] however they posed no problem to the curvature scale space with boundary conditions presented in this paper. They are presented in 4.13 to 4.14.



a) Rendered original. b) After 1000 iterations of diffusion and merging, 28 groups were found, 2 groups had over 50 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

Figure 4.13: Partition of two McRCIM rocks, 256x256.



a) Rendered original. b) After 1000 iterations of diffusion and merging, 77 groups were found, 6 groups had over 50 datapoints. c) Mesh of fitted superquadrics. d) Rendered superquadrics.

Figure 4.14: Partition of three McRCIM rocks, 256x196.

Section 5 Conclusion

The task of partitioning surface information into part-oriented descriptions of object geometry is fundamental to problems of object recognition and model building. This paper presented a partitioning method, given objects that can be characterized as conjunctions of convex parts. The key idea borrowed from early work in computer vision, that is, the concurrent application of edge-based and region-based methods of image segmentation.

The contribution in this paper is a re-casting of this paradigm in terms of surface differential geometry and the development of a solution using constraint propagation networks (a.k.a. relaxation labelling). The results obtained demonstrate that the resulting algorithms work very well and can provide a sound basis for the generation of object models as was shown with the superquadric renderings.

Future work will be directed at finding how to use multiple scales to perform an interpolation because different parts of objects are likely to be best represented at different scales.

Given the complexity of the overall processing, the results obtained are both stable and robust. This is largely due to the involvement of two complementary processes. In cases where the scale space was fragmented because of noise in the input data, the resolution algorithm was still able to determine a consistent interpretation, albeit at a significant increase in computation time. Errors in the detection of boundary features and the resulting errors in constraint functions were often accommodated by cues provided by the scale space description. This experience suggests that there is still much to be gained in exploiting computation based on multiple representations as advocated back in the seventies [34].

Bibliography

[1] H. Asada and M. Brady. The curvature primal sketch. *IEEE Tr. on PAMI*, PAMI-8:2–14, 1986.

[2] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Tr. on PAMI*, PAMI-10(2):167–192, Mar. 1988.

[3] W. Cheung, F. Ferrie, G. Carayannis, and J. Edwards. Rockpile surface decomposition: Machine vision in mining. In *PROC. Canadian Conference on Industrial Automation*, Montréal, Québec, June 1-3, 1992 1992.

[4] W. Cheung, F. Ferrie, R. Dimitrakopoulos, and G. Carayannis. Towards computer vision driven rock modelling. In *2nd Canadian Conference on Computer Applications in the Mineral Industry*, Vancouver, B.C., Sept. 15-18 1991.

[5] W. Cheung, F. Ferrie, R. Dimitrakopoulos, and G. Carayannis. Computer vision-based rock modelling. *Computer Systems in Engineering*, 1992. accepted for publication.

[6] W. Cheung, M. Lin, F. Ferrie, G. Carayannis, and J. Edwards. Pilot studies on computer vision techniques for an automated mine environment. In *Australian Robot Association: The Third National Conference on Robotics*, pages 142–153, Melbourne, Australia, June 3-6 1990.

[7] C. Chu and J. Aggarwal. The integration of region and edge-based segmentation. *Proceedings of the 3RD International Conference on Computer Vision*, 1:117–120, Dec. 1990.

[8] J. Clark. Singularity theory and phantom edges in scale space. *IEEE Tr. on PAMI*, PAMI-10(5):720–727, 1988.

[9] M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Inc., Englewood Cliffs,New Jersey, 1976.

[10] G. Dudek and J. Tsotsos. Shape representation and recognition from curvature. In *Proceedings of the IEEE CS Conf. on CVPR*, pages 35–41, Maui, Hawaii, June 1991. Computer Society of the IEEE, IEEE Computer Society Press.

[11] F. Ferrie and J. Lagarde. Curvature consistency improves local shading analisys. In *Proceedings of the 10TH International Conference on Pattern Recognition*, pages 70–76, Atlantic City, New Jersey, jun 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[12] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom-up. In *Proceedings IEEE Workshop on Interpretation of 3D Scenes*, pages 170–176, Austin, Texas, Nov. 27-29 1989. Computer Society of the IEEE, IEEE Computer Society Press. IEEE Trans. PAMI - accepted for publication.

[13] F. Ferrie, J. Lagarde, and P. Whaite. Recovery of volumetric object descriptions from laser rangefinder images. In *Proc. of the 1ST Eur. Conf. on CV*, Antibbes, France, April 1990.

[14] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom-up. *IEEE Tr. on PAMI*, 15(8), Aug. 1993. to appear.

[15] F. Ferrie, A. Lejeune, and D. Baird. Curvature, scale, and segmentation. In *SPIE App. of AI X: Mach. Vision & Rob.*, volume 1708, pages 240–250, Orlando, Florida, April 20-24 1992.

[16] F. Ferrie and M. Levine. Deriving coarse 3d models of objects. In *Proceedings of the IEEE CS Conf. on CVPR*, pages 345–353, University of Michigan, Ann Arbor, Michigan, June 1988. Computer Society of the IEEE, IEEE Computer Society Press.

[17] F. Ferrie, S. Mathur, and G. Soucy. *3D Object Recognition Systems*, chapter Feature Extraction for 3-D Model Building and Object Recognition. Elsevier, Amsterdam, 1993.

[18] V. Guillemin and A. Pollack. *Differential Topology.* Prentice-Hall, Englewood Cliffs, NJ.

[19] D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.

[20] R. Hummel and S. Zucker. On the foundation of relaxation labeling processes. *IEEE Tr. on PAMI*, PAMI-5(3):267–287, 1983.

[21] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations, I: The components of shape and the reaction-diffusion space. Technical Report LEMS 105, LEMS, Brown University, May 1992.

[22] J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[23] G. Lee and G. Stockman. Obtaining registered range and intensity images using the technical arts scanner. Technical Report CPS-91-08, Pattern Recognition and Image Processing Laboratory, Dept. Computer Science, Michigan State U., East Lansing, MI 48824, 1991.

[24] A. Lejeune. Partitioning range images using curvature and scale. Master's thesis, Department of Electrical Engineering, McGill University, 1993.

[25] D. Marr. *Vision.* W.H. Freeman & Co., San Francisco, 1982.

[26] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, Sept. 1990.

[27] J. Mohammed, R. Hummel, and S. Zucker. A gradient projection algorithm for relaxation methods. *IEEE Tr. on PAMI*, PAMI-5(3):330–332, 1983.

[28] A. Pentland. Recognition by parts. In *Proceedings of the 1ST International Conference on Computer Vision*, London,UK, June 1987. Computer Society of the IEEE, IEEE Computer Society Press.

[29] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Tr. on PAMI*, PAMI-12(7):629–639, 1990.

[30] M. Rioux. Laser range finder based on synchronized scanners. *Applied Optics*, 23(21):3837–3844, Nov. 1984.

[31] P. Sander and S. Zucker. Inferring differential structure from 3-d images: Smooth cross sections of fiber bundles. *IEEE Tr. on PAMI*, PAMI-12(9):833–854, 1990.

[32] A. P. Witkin. Scale-space filtering. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 1019–1021, 1983.

[33] S. Zucker, C. David, A. Dobbins, and L. Iverson. The Organization of Curve Detection: Coarse Tangent Fields and Fine Spline Coverings. In *Proceedings of the 2ND International Conference on Computer Vision*, Tampa,Florida,USA, Dec. 1988. Computer Society of the IEEE, IEEE Computer Society Press.

[34] S. W. Zucker. *Computer Vision Systems*, chapter Vertical And Horizontal Processes In Low Level Vision, pages 187–195. Academic Press, 1978.