To the Graduate Council:

I am submitting herewith a thesis written by Mohamed Baccar entitled "SURFACE CHARACTERIZATION USING A GAUSSIAN WEIGHTED LEAST SQUARES TECHNIQUE TOWARDS SEGMENTATION OF RANGE IMAGES." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

M. A. Abidi

M. A. Abidi, Major Professor

We have read this thesis
and recommend its acceptance:

Marshall O. Pace

Mohan M. Trivedi

Accepted for the Council:

C W Minkel

Associate Vice Chancellor
and Dean of The Graduate School

SURFACE CHARACTERIZATION USING A GAUSSIAN WEIGHTED LEAST SQUARES TECHNIQUE TOWARDS SEGMENTATION OF RANGE IMAGES

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Mohamed Baccar

May 1994

ACKNOWLEDGEMENTS

Sincere thanks are due to my wife Cindy H. Baccar for her day to day influence, and to my mother in law Betty W. Henson for providing her support to our efforts. I would like to express sincere thanks to Prof. M. A. Abidi for his guidance and contributions to this research. Sincere thanks are also due to Prof. R. C. Gonzalez for paying interest to the ideas, broadening the application, and for his tremendous influence on my reasoning and writing skills. Sincere thanks are also due to Prof. M. Trivedi and Prof. M. O. Pace for being part of my thesis committee, and for their tremendous contributions to my education.

ABSTRACT

Range images provide direct depth information of object points in the scene. They provide useful 3–D information useful for object location/recognition tasks. The primary focus of this research is to develop reliable and efficient algorithms for feature extraction, surface characterization, and segmentation.

The most effective technique, in feature enhancement, is to retrieve the derivatives from a second order polynomial locally fitted via an Equally Weighted Least Squares (EWLS) technique. This approach provides smoothing but has the disadvantage of providing poor approximations near the discontinuities. To gain higher accuracies, we develop a Gaussian Weighted Least Square (GWLS) technique that provides an additional focus element for the central pixel location. The computational efficiency was also maintained through the use of weighted sets of orthogonal polynomials. Two segmentation algorithms were developed that rely heavily on this development.

In the second segmentation algorithm, a general procedure was developed to characterize arbitrary shaped regions in range images. The proposed module minimizes the L_p norm, and its efficiency and robustess properties were evaluated against the Heavy Tailed Distribution, which is a natural extension to the normal distribution. The theoretical and experimental evidence shows the advantages of this approach over the least squares and Chebychev approximations in fitting arbitrary shaped contaminated regions.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

FIGURE **PAGE**

FIGURE **PAGE**

FIGURE		PAGE

FIGURE PAGE

CHAPTER 1

Introduction

Range images are valuable tools in scene analysis since they provide explicit geometry of objects in the scene. Each pixel intensity represents the distance of the corresponding surface point to a reference coordinate system determined by the camera position and orientation in a three–dimensional Euclidean space. Range Images can provide useful information for navigation and object manipulation tasks. The subject of this research is to partition a given range image into groups of homogeneous regions which can simplify higher level object location/recognition problems.

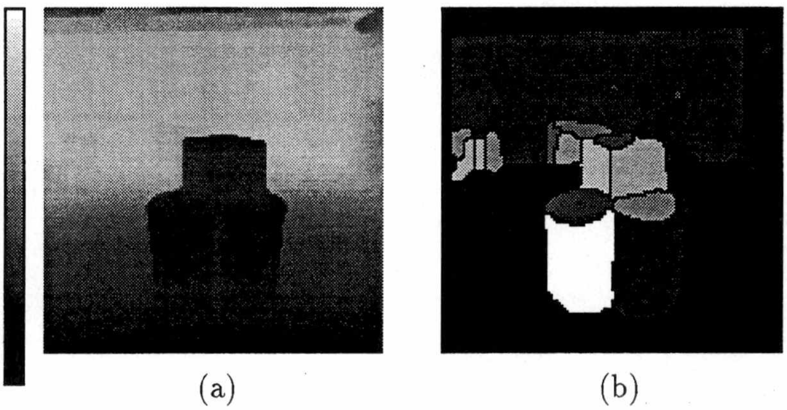
1.1 Statement of the Problem

Range scanners now have commercial applications due to the improved quality of range images, and an acquisition speed of more than one frame per second [29]. The laser scanner at the University of Tennessee Computer Vision and Robotics Laboratory was manufactured by Odetics [28]. This system has a 60×60 degree field–of–view, an azimuth/elevation resolution of 0.47 degree per pixel, and an ambiguity interval of 15.4 *ft*. It can generate a 128×128 eight–bit intensity/range pair with a resolution of 0.72 *in* per gray level. Range scanners are single–sensory devices. The acquired measurements represent angular distances from object points, (x, y, z), in the scene to the focal point of the scanner. Depth measurements correspond to distances along the Z axis, which is the distance perpendicular to the image plane. The problem of retrieving depth measurements from the angular measurements is known as the cosine–effect caused by the scanner–hardware. The transformation that generates cartesian coordinates (x, y, z), where x and y correspond to the image plane axis and z representing depth, from the angular measurements (r, θ, ϕ) is given by [29, 35]:

$$\begin{aligned} x &= \frac{r\tan(\theta)}{\sqrt{1+\tan^2(\theta)+\tan^2(\phi)}} \\ y &= \frac{r\tan(\phi)}{\sqrt{1+\tan^2(\theta)+\tan^2(\phi)}} \\ z &= \frac{r}{\sqrt{1+\tan^2(\theta)+\tan^2(\phi)}} \end{aligned}$$

Sweeney [36] analyzed the defects of commercially available range scanners, and identified two major sources of error: Systematic errors that depend on surface characteristics such as color, reflectivity, and smoothness of the objects viewed. The mechanical defects of the rotating mirrors which result in shifting of the location of measured intensities. Random errors caused by the heating effects of the electronic devices among other physical and optical effects. The error present in the measurements stems from a mixture of noise populations, which consequently excludes the normality assumption.

Figure 1.1 illustrates the source and objective of a typical range image segmentation algorithm. Note that each homogeneous surface patch of Fig. 1.1(a)



(a) (b)

Figure 1.1: *(a) Input Range Image (darker is closer). (b) Segmentation of (a).*

was identified with a unique label (shade of gray) in Fig. 1.1(b). Segmentation of range images has been a growing topic of research for more than a decade. It is a fundamental process in robotic vision, since it is at this stage that all higher level processing begins.

1.2 Range Image Segmentation Review

Methods involving the segmentation of range images can be divided into two major categories: edge–based methods, and region–based methods. In the first category, the segmentation relies on the extraction of point features such as boundaries. Two types of edges are sought: Step edges which represent the object's outer boundaries and can be defined as discontinuities in depth, and roof edges which represent the internal boundaries of the object and can be defined as discontinuities in surface orientation. These edges are then combined to yield a complete boundary description of the scene. Additional edge–linking steps are needed to ensure closed boundaries. The surface primitives, which occur as a group of pixels separated by closed boundaries, are identified with a connected component analysis [3] on the resulting edge–map.

Region–based Range Image Segmentation [17] divide a region R of the image into n subregions, $R_1, R_2, \cdots, R_n$, such that

- $\bigcup_{i=1}^{n} R_i = R.$
- R_i is a connected region, $i = 1, \cdots, n.$
- $R_i \cap R_j = \emptyset \;\; \forall\, (i,j), \;\; i = 1, \cdots, n.$
- $P(R_i) = TRUE, \;\; for\ i = 1, \cdots, n.$
- $P(R_i \cup R_j) = FALSE, \;\; \forall\, i \neq j.$

The subregions R_i form a partition of the region R. P is a pre–specified property that defines the criteria used in forming the partition of the discrete space. The partition of the image can be generated from an initial set of the seed points from which growing is initiated, or through a split–and–merge algorithm which starts with an arbitrary initial partition of the range image and splits the region when the property P is not satisfied, and merges two distinct regions when the union satisfies the property P. The process continues until no additional splitting or merging occurs. In the following, we provide a brief literature review of Edge– and Region–based segmentation algorithms.

1.2.1 Edge–based Segmentation Review

Haralick [20] defined a step edge to occur in a pixel *if and only if* there exists a point within the pixel area having a negatively sloped zero–crossing of the second directional derivative. The underlying image in an N×N window was represented using a facet model polynomial surface of order 3. The polynomial was defined as the tensor product of one–dimensional set of orthogonal polynomials, with respect to the equal weight, defined on a symmetric domain. Because nonedge pixels can contribute zero–crossings of the second directional derivative, another restriction was imposed on the gradient magnitude to be larger than a pre–specified constant threshold. Consequently, if the gradient exceeds a specified threshold and a zero–crossings of the second directional derivatives occurs in a direction of $+14.9$ to -14.9 degrees of the gradient direction within a circle of one pixel length centered in the pixel, the pixel was declared an edge pixel. Window sizes varied from 3×3 to 11×11, and the signal to noise ratio was increased using larger windows. This technique was shown to outperform the Prewitt operator and the Marr–Hildreth edge detector. Standard 3×3 window operators such as Roberts, Sobel, Prewitt, and Kirsch edge operators were shown to perform poorly in the presence of noise.

Fan *et al.* [38] applied a multiple scale feature extraction algorithm. The directional curvature extrema and zero–crossings were used to identify step and roof boundaries in range images. The range image was initially smoothed using a number of Gaussian smoothing kernels for $\sigma = 0.5, 1$, and 1.5. For each of the generated images, the first order difference was computed in four different directions (0, 45, 90, and 145 degrees) using four 5×5 directional masks. The second order difference was computed through convolution of the directional mask with the first order difference at that direction. The directional curvature was computed, and the corresponding extrema and zero–crossings were subsequently identified. Zero–crossings were used to identify step boundaries, and curvature extrema were used to identify roof boundaries. Scale–space tracking was employed to establish correspondence between the detected features. A second technique which computes the two principal curvatures at a single scale was introduced. The extrema and zero–crossings of the two principal curvatures were identified after application of a single smoothing step using a Gaussian kernel. This technique is sensitive to the proper selection of the scale, but eliminates the scale–space tracking problem. Using either technique, edge linking was subsequently applied to join open curves with a gap of 5 pixels or less. The preliminary segmentation was followed by surface fitting using second order quadrics to derive the properties of the identified surface patches.

Parvin [31] extracted features from range images through fitting a one–dimensional third–order regression polynomial in two orthogonal directions at each pixel location. Zero–crossings of the first and second directional derivatives were subsequently determined. Ridges, Ravines, and Creases create zero–crossings in the first derivative. These points were classified when the fit error is small, and the root of the zero–crossing falls within the pixel area. The distinction among *ridges* and *ravines* was decided by the sign of the second derivative. Because s-maller windows provide localization but perform poorly in the presence of noise, and larger windows are more robust with respect to noise but have poor localization properties, the size of the discrete window was chosen to be large enough with an added constraint imposed on the fit error. Once the optimal window was set, the localization was achieved by gradually decreasing the size of the window, and the merging was achieved after gradually increasing the size of the window. The author avoided the scale–space tracking problem which emerges from using a number of smoothing kernels before differentiation, since most existing solutions to this problem rely on heuristics in order to establish correspondence.

Mintz [13, 27] transformed the step edge profile to a binary profile using robust fitting techniques, namely the LMedS estimator, in the neighborhood of an edge. The fitting technique used can sustain 50% outliers. Up to half of the samples in the discrete window were assigned lower weights when fitted near an edge. This enabled the classification of the transition region between outlier and inlier pixels into a binary profile which represents a step edge profile of the original function. Once the edge was declared, more votes were accumulated using larger size windows containing the original window. The decision was made when more than 50% of the votes declare the profile as an edge profile. This technique was applied on simulated step and roof edges with added Gaussian noise and a signal to noise ratio close to 4. The probability of the detection was 84% and the probability of localization was 90%.

1.2.2 Region–based Segmentation Review

Besl [29] implemented a region–based segmentation algorithm which extracts intrinsic surface properties (geometric invariants [6]) to identify seed regions used for growing. The range image was initially smoothed, then the first and second order derivatives were computed at each pixel location using a second order regression polynomial locally fitted via an Equally Weighted Least Squares technique. The Gaussian curvature, K, and the Mean curvature, H, were subsequently computed at each pixel location. The Gaussian and Mean curvatures were used to classify each pixel into eight surface types based on the relative signs of H and K:

1. Peak: $(H < 0; K > 0)$,
2. Ridge: $(H < 0, K = 0)$,
3. Saddle Ridge: $(H < 0; K < 0)$,
4. Minimal:$(H = 0; K < 0)$,
5. Saddle Valley: $(H > 0; K < 0)$,
6. Valley: $(H > 0; K = 0)$,
7. Pit: $(H > 0; K > 0)$,
8. Flat: $(H = 0, K = 0)$.

A sign–map image was obtained using $\mathrm{sgn}_{HK} = 1+3(1+\mathrm{sgn}_{\epsilon}(H))+(1-\mathrm{sgn}_{\epsilon}(K))$, where ϵ is a zero–threshold obtained from an evaluation of the quality of the range image. Each region of the sign–map image, containing a large number of pixels of similar surface type, was eroded using a binary contraction algorithm until a seed region of "maximally interior, highly reliable, single surface seed region is isolated." The seed regions were subsequently grown via variable order surface fitting [7] until reaching a limit imposed by the surface fit error, and the order of the polynomial. Results of this segmentation were illustrated for synthetic and real range data.

Koivunen *et al.* [24] implemented a combined edge– and region–based segmentation algorithm. The authors started with noise attenuation using a 3×3 discrete window mask, and proceeded to compute the first and second order derivatives at each pixel location using 5×5 local convolution masks. The Gradient and the Gradient magnitude were computed using first order derivatives. The second order derivatives were used to compute the eigenvalues and eigenvectors of the matrix

$$\mathbf{M} = \begin{pmatrix} f_{uu} & f_{uv} \\ f_{uv} & f_{vv} \end{pmatrix},$$

where f_{uu}, f_{uv}, and f_{vv} are second partials with respect to the coordinate system variables u and v. The eigenvalues represent the extrema of the second directional derivatives, and the eigenvectors represent their directions. This information was used to generate a sign–map image. Pixels of the same label in the sign–map image were subsequently grouped into coherent regions. Edges were obtained after differentiating the image twice using four first derivative templates. The first iteration identifies step boundaries; the second iteration identifies roof boundaries. The sign–map image was overlayed with the detected edges for a final segmentation. In view of the influence caused by outliers on the estimated derivatives using the Equally Weighted Least Squares technique, Koivunen [25] implemented two robust estimation techniques in an attempt to achieve a more reliable segmentation. The first technique is iterative reweighing least squares (IRLS) proposed by Besl [25]. The second technique is the least trimmed squares (LTS) introduced by Rousseeuw [16, 25]. The IRLS computes the median of the neighborhood as an initial zero–order fit from which the residuals are retrieved. In the implementation of the IRLS, two iterations use Huber minimax weight function which is then followed by two more iterations using Hampel's weight function. The LTS is a standard least squares technique with outlier rejection. Outliers are identified when $|r_i|/\sigma > 2$, where r_i is the residual (difference between actual and estimated values), and σ is a scale estimate of the residuals. The classified outlier pixels are assigned zero weight during the next iteration. Robust estimation techniques show significant improvements to standard least squares techniques in noise removal tasks, surface parameter extraction, and segmentation. The LTS technique was superior in performance to the IRLS and EWLS techniques.

Yokoa [39] showed that the standard least squares technique was inadequate for computing accurate derivatives around surface boundaries. Among several 5×5 windows containing the pixel of interest, the window with the least fit error was employed to compute the derivatives. A curvature sign–map was generated from the estimated derivatives. Step edges were identified using a 3×3 discrete window operator; roof edges were identified using the discontinuities of computed surface normals. Standard thresholding techniques were used to isolate the two types of boundaries. These results were then overlayed to yield a final segmentation image.

Haralick [19, 21, 22, 37] developed a segmentation algorithm by which edge detection and region growing are achieved using a single process. The author assumed that the ideal model representing the noisy image is a collection of facets that can ideally be modeled using a polynomial surface. Two facets were experimented with and shown to converge: The flat facet model where an ideal homogeneous region is modeled by a flat surface. Each central pixel of a given

3×3 facet is element of 9 possible neighboring facets. The average gray tone intensity of each of these facets is computed, and the updated gray tone intensity takes the value of the nearest average gray tone among the 9 competing facets. The procedure is repeated until the image becomes a collection of flat facets of arbitrary shapes that correspond to homogeneous regions of the image. This method, however, yields an oversegmentation, and its convergence is slow. With the flat facet model assumption, edges are limited to depth discontinuities which correspond to boundaries between neighboring facets carrying very different gray tone intensities. In this context, edges correspond to a dissimilarity between neighboring facets, and homogeneous regions are identified by the similarity in gray tone intensities between neighboring facets. The second approach is the sloped facet model, where the ideal facets are represented by planar surfaces. Each 3×3 facet is locally fitted to a first order polynomial, and the updated gray tone for the central pixel is evaluated at one of the 9 competing facets that contains the pixel of interest, and carries the minimal least squares error. The procedure is repeated, until a set of arbitrary shaped facets with similar regression coefficients are isolated, and the boundaries are identified by means of the F–statistic between neighboring facets. This procedure has a higher convergence rate than the flat facet model, and provides less regions in the final results. Even though the sloped facet model provides more meaningful results, the segmentation was not ideal for a higher level process. The author extends the same concepts developed to grouping regions obtained from either method. A formal definition of the variance of a collection of neighboring regions called a neighborhood set was derived (two regions are neighbors if a 4– or 8– connectivity test shows labels from the corresponding distinct regions). Each region which is the central element of a neighborhood set, is also element of other neighboring sets. The region is identified with a property vector that contains descriptive information. The property vector of each region is iteratively updated as a function of the property vectors of the neighboring set of regions that contain the region of interest, and possess the minimal variance. Regions are merged on the basis of similarity between property vectors. The final segmentation is more meaningful, but still faces problems with regions containing boundary information. The result was improved considerably by excluding edge regions and processing the remaining regions. A simple filling test was consequently applied to extend the obtained regions in order to isolate thin boundaries that correspond to a useful segmentation for a higher level process.

Boulanger [9] identified step and roof edges in the input image. He partitioned the image into small rectangular regions satisfying the equation of a planar surface. The windows were chosen such that they do not contain step or roof boundaries. Bayesian decision theory was used as a predicate to allow grouping of these planar surfaces into second degree surfaces. Segmentation results show the ability of the algorithm to identify planar, cylindrical, and spherical objects.

1.3 Summary and Synopsis

Since the primary objective of an Edge–based segmentation lies in the extraction of reliable boundary information, differentiation of the range image is needed. A commonly used technique in derivative estimation relies on an Equally Weighted Least Squares polynomial approximation of the range image over small local windows [29]. Within a square grid, the range image is fitted to a two–dimensional polynomial from which the derivatives are retrieved. Equally Weighted Least Squares techniques, however, provide inadequate approximations of the range image around step boundaries (an important consideration in an edge–based algorithm). Based on experimental evidence and heuristics, we develop a Gaussian Weighted Least Squares technique to provide the additional focus needed for the central pixel location. The Gaussian weight will be shown experimentally to outperform the Equal weight in yielding accurate and reliable boundary information from real range images. Based on this development, two range image segmentation algorithms were achieved:

The first algorithm combines the depth and orientation discontinuities, obtained using the GWLS technique, to provide an edge magnitude image comprising these two types of discontinuities. The Morphological Watershed algorithm was subsequently applied to yield the desired scene segmentation. The second algorithm relied on edge detection and region growing principles. In the edge detection stage, the depth and orientation discontinuities of the range image were obtained using the GWLS technique at a number scales (different values of σ), which was intended to provide more reliabe detection. The resulting images were subsequently combined to yield a complete boundary description of the range image. In the region growing stage, the surface primitives acquired from the previous step were considered as seed regions, and were subsequently grown based on their polynomial surface representation. This approach has the potential of merging smaller seed regions into larger neighboring seed regions. The smaller seed regions frequently result from an oversegmentation due to false detection at preliminary stages of the detection. Different numerical techniques that provide polynomial approximations of arbitrary regions in range images were investigated upon:

There are two existing techniques [18], which can be applied. The first technique minimizes the L_1–norm (corresponds to a Chebychev approximation), and the second technique minimizes the L_2–norm (corresponds to a Least Squares approximation). We investigate the asymptotic behavior of intermediate estimates that minimize the L_λ–norm, where $1 \leq \lambda \leq 2$. The statistical evidence and conducted experiments both illustrate the superior performance of intermediate norms in providing more reliable regression estimates. These estimates also show superior performance in segmentation of real range images.

Chapter 2 introduces the GWLS technique for estimating accurate derivatives in range images. The surface characterization problem is also addressed. Chapter 3 describes a range image segmentation algorithm that is based on the Morphological Watersheds and the techniques developed in Chapter 2. The GWLS technique is used to transform a range image to a form suitable for the application of the Morphological Watershed algorithm. Chapter 4 is a description of the second segmentation algorithm which provides a reliable segmentation of range images. Chapter 5 is a theoretical and experimental evaluation of the L_λ–norms in regression. This chapter also analyzes the effects of these norms in segmentation of real range images. Finally, Chapter 6 presents a summary and conclusions.

CHAPTER 2

Surface Characterization Using A Gaussian Weighted Least Squares Technique

2.1 Introduction

Facet models were studied on numerous occasions by different authors mainly for the purpose of regression, image smoothing, and derivative estimation. The most commonly used facet models, in range analysis, rely on an Equally Weighted Least Squares (**EWLS**) approximation of the discrete measurements over small local windows. The earliest developments of this technique was due to Anderson and Houseman [2]. They introduced tables of orthogonal polynomials useful for fitting equally spaced one–dimensional data sets. The idea was developed for regression purposes; it provides a more meaningful interpretation of the discrete samples. Savitsky and Golay [34] applied the least squares technique to one–dimensional data sets for smoothing and differentiation purposes. They established correspondence between the application of general filters and the method of least squares and derived a set of convolute integers that can be applied to one–dimensional equally spaced discrete data sets. From the derived tables, smoothing and differentiation is achieved systematically after convolution of the desired window with the data set. Edwards [15] extended Savitsky's approach to two–dimensional data sets following the need of applying this technique in higher dimensions. Similar to Savitsky, Edwards applied this technique to equally spaced data in two orthogonal directions X and Y. The generated two–dimensional convolute integers, in the sense of least squares, share the same properties of their one–dimensional counterpart. Haralick [20] proved that the tensor product of two one–dimensional discrete orthogonal polynomials, over an equally spaced discrete space, is orthogonal in 2–D. The least squares approximation, in the sense of tensor product, is equivalent to its explicit representation, in canonical form, as a linear combination of the power terms $u^i v^j$, $i+j \leq n$ [8]. Beaudet [5] used the least squares technique for estimating the derivatives which were subsequently used to derive rotationally invariant image operators, such as the gradient magnitude. Bolle [8] investigated the performance of the least squares method in fitting spheres and cylinders. He proved the smoothing property of second degree polynomials in the presence of Gaussian noise.

In this chapter, we introduce a more general technique that has the same computational advantages of **EWLS** technique. The developed technique follows the need for more efficient and accurate algorithms in estimating derivatives in images. The results of experimentation has shown the utility and usefulness of the Gaussian Weighted Least Squares technique (**GWLS**) for estimating accurate derivatives. This chapter is organized as follows: Section 2 deals with the general theory of discrete and orthogonal polynomials in 1–D. Section 3 investigates the extensions to 2–D, and the consistency property of this approach (the added weights do not introduce any bias to the solution). Section 4 investigates the use of Gaussian weights in determining accurate derivatives. The correspondence between the window size and the standard deviation is established. Tables of convolute integers are subsequently derived to maximize efficiency and speed of the algorithm. Section 5 deals with surface characterization and comparisons among the **EWLS** and the **GWLS** techniques. Finally, a summary and conclusions are given in Section 6.

2.2 Polynomial Approximation of Range Images

Definition 1: The polynomials $\phi_i(u)$, where $i = 0, \ldots, n$ and $u \in [-M, M]$, are said to be orthogonal with respect to the weight function $W(u)$ if and only if:

$$< \phi_i, \phi_j >= \sum_{u=-M}^{M} W(u)\phi_i(u)\phi_j(u) = \alpha_i \delta_{ij} \ ,$$

where δ_{ij} is the Kronecker delta function, $\alpha_i =< \phi_i, \phi_i >$, and $W(u)$ is positive for all values of u and not identically zero. The following is a recursion formula that generates one–dimensional sets of orthogonal polynomials [11]:

The set of polynomials, $\{\phi_0, \phi_1, \ldots, \phi_n\}$, given by

$$\phi_0(u) = 1 \quad \phi_1(u) = u - B_1, \text{ for each } u \in [-M, M] \ ,$$

where

$$B_1 = \frac{< u,\ 1 >}{< 1,\ 1 >} \ ,$$

and for $k \geq 2$

$$\phi_k(u) = (u - B_k)\phi_{k-1}(u) - C_k\phi_{k-2}(u) \ ,$$

where B_k and C_k are given by

$$B_k = \frac{< u\ \phi_{k-1},\ \phi_{k-1} >}{< \phi_{k-1},\ \phi_{k-1} >}\ , \qquad C_k = \frac{< u\ \phi_{k-1},\ \phi_{k-2} >}{< \phi_{k-2},\ \phi_{k-2} >}\ ,$$

are orthogonal with respect to the weight function $W(u)$, $u \in [-M, M]$.

2.3 Range Image Approximation Using Weighted Polynomials

Consider the polynomials ψ_{ij} given by the tensor product of two orthogonal polynomials, ϕ_i and ϕ_j, with respect to the arbitrary weight $W(u)$: $\psi_{ij}(u,v) = \phi_i(u)\phi_j(v)$. These polynomials ψ_{ij}, $i+j \leq n$, are in effect orthogonal with respect to the two–dimensional weight function $W(u,v) = W(u)W(v)$:

proof: In order to prove orthogonality in 2–D, it suffices to show that the dot product of a given pair of polynomials (ψ_{ij}, ψ_{kl}) is identically zero unless these two polynomials are identical. The dot product is given by

$$< \psi_{ij},\ \psi_{kl} > = \sum_{u=-M}^{M} \sum_{v=-M}^{M} W(u)W(v)\psi_{ij}(u,v)\psi_{kl}(u,v)\ .$$

Substituting for the corresponding tensor products yields

$$< \psi_{ij},\ \psi_{kl} > = \sum_{u=-M}^{M} \sum_{v=-M}^{M} W(u)W(v)\phi_i(u)\phi_j(v)\phi_k(u)\phi_l(v)\ .$$

Upon separating the variables u and v, we get

$$< \psi_{ij},\ \psi_{kl} > = \left[\sum_{u=-M}^{M} W(u)\phi_i(u)\phi_k(u) \right] \left[\sum_{v=-M}^{M} W(v)\phi_j(v)\phi_l(v) \right]\ .$$

Since the elements of the set ϕ_i are orthogonal with respect to the one–dimensional weight, it follows that

$$< \psi_{ij},\ \psi_{kl} > = [\alpha_i \delta_{ik}]\, [\alpha_j \delta_{jl}] = \alpha_i \alpha_j \delta_{ik} \delta_{jl}\ .$$

Using the properties of the δ function, we conclude that the previous expression is different from zero only when $i = k$ and $j = l$, which means that both polynomials ψ_{ij} and ψ_{kl} are identical. It follows that the two-dimensional set of polynomials, ψ_{ij}, $i+j \leq n$, are orthogonal with respect to the two-dimensional weight function $W(u)W(v)$. This concludes the proof.

The importance of this result lies in the fact that the derived set of two-dimensional orthogonal polynomials, ψ_{ij}, $i + j \leq n$, are independent and form a basis set of the n'th order polynomial-space of two dimensions. They can, therefore, be used to provide a piecewise approximation of range images over small local windows. The approximation to a range image, $f(u,v)$, is of the form:

$$g(u,v) = \sum_{i+j \leq n} a_{ij}\psi_{ij}(u,v) \ ,$$

where the polynomials ϕ_i are orthogonal with respect to the weight function, $W(u)$, defined and positive in $[-M, M]$. The approximation of $f(u,v)$ by $g(u,v)$ is sought to minimize the weighted least square error given by

$$\epsilon^2 = \sum_{u=-M}^{M} \sum_{v=-M}^{M} W(u)W(v)\left[f(u,v) - g(u,v)\right]^2 \ .$$

After expansion of the weighted least squares error and substitution for $g(u,v)$ defined above, leads

$$\epsilon^2 = \sum_{u,v} W(u)W(v)f^2(u,v)$$

$$-2 \sum_{i+j \leq n} a_{ij} \sum_{u,v} W(u)W(v)f(u,v)\psi_{ij}(u,v)$$

$$+ \sum_{i+j \leq n} \sum_{k+l \leq n} a_{ij}a_{kl} \sum_{u,v} W(u)W(v)\psi_{ij}(u,v)\psi_{kl}(u,v) \ .$$

The solution coefficients a_{ij} are chosen such that $\frac{\partial \epsilon^2}{\partial a_{ij}} = 0$, which implies

$$-2 \sum_{u,v} W(u)W(v)f(u,v)\psi_{ij}(u,v)$$

$$+2 \sum_{k+l\leq n} a_{kl} \sum_{u,v} W(u)W(v)\psi_{ij}(u,v)\psi_{kl}(u,v) = 0 \ .$$

Since the polynomials ψ_{ij} are orthogonal with respect to the two–dimensional weight, $W(u)W(v)$, the previous expression becomes

$$\sum_{u,v} W(u)W(v)f(u,v)\psi_{ij}(u,v) = \sum_{k+l\leq n} a_{kl}\alpha_i\delta_{ik}\alpha_j\delta_{jl} \ .$$

Using the properties of the Kronecker delta function, δ, yields

$$\alpha_i\alpha_j a_{ij} = \sum_{u,v} W(u)W(v)f(u,v)\psi_{ij}(u,v) \ .$$

Therefore, the solution coefficients, after substitution for ψ_{ij}, are given by

$$a_{ij} = \frac{1}{\alpha_i\alpha_j} \sum_{u,v} W(u)W(v)f(u,v)\phi_i(u)\phi_j(v) \ , \tag{2.1}$$

where

$$\alpha_i \ = < \phi_i, \ \phi_i > = \sum_{u=-M}^{M} W(u){\phi_i}^2(u) \ .$$

Consider the function $K_i(u)$ such that

$$K_i(u) = \frac{W(u)\phi_i(u)}{\alpha_i} \ . \tag{2.2}$$

If the discrete window is fixed ($N \times N$), this function is constant and can be substituted for its numerical discrete value in the solution to the weighted least squares problem. The equivalent formulation of a_{ij} in terms of K_i and K_j is as follows:

$$a_{ij} = \sum_{u=-M}^{M} \sum_{v=-M}^{M} f(u,v)K_i(u)K_j(v) \ . \tag{2.3}$$

This simplified expression results in significant gains in terms of execution time.

One important property to show is that the solution coefficients to the weighted least squares problem is *consistent* for any arbitrary weight:

proof: Suppose that the input discrete signal we want to approximate is sampled from an n'th order polynomial $P(u,v)$, which can be represented as a linear combination of the orthogonal set ψ_{ij}, $i+j \leq n$, with respect to the arbitrary weight $W(u)W(v)$. If the input signal is formulated as $P(u,v) = \sum_{i+j\leq n} a_{ij}\psi_{ij}(u,v)$, the approximation using our described procedure will be a polynomial $g(u,v) = \sum_{k+l\leq n} b_{kl}\psi_{kl}(u,v)$, where u and $v \in [-M, M]$. Consistency of the estimate means that the approximation is identical to the input signal for any arbitrary choice of the weight function (provided , of course, a sufficient number of observations are available), or $b_{ij} = a_{ij}$ for all i, j. Following Eq. (2.1), the coefficients b_{ij} are determined by

$$b_{ij} = \frac{1}{\alpha_i \alpha_j} \sum_{u,v} W(u)W(v)P(u,v)\psi_{ij}(u,v) \ .$$

Substituting for $P(u,v)$, and using the orthogonality property of the ψ_{ij}'s, yields

$$\begin{aligned} b_{ij} &= \frac{1}{\alpha_i \alpha_j} \sum_{u,v} W(u)W(v)\psi_{ij}(u,v) \left[\sum_{k+l\leq n} a_{kl}\psi_{kl}(u,v) \right] \\ &= \frac{1}{\alpha_i \alpha_j} \sum_{k+l\leq n} a_{kl}\alpha_i\delta_{ik}\alpha_j\delta_{jl} \ . \end{aligned}$$

Using the property of the delta function, the previous expression is simplified further to yield $b_{ij} = a_{ij}$ for all i, j. The solution coefficients are therefore *consistent.* This concludes the proof.

2.4 The Gaussian Weighted Least Squares technique: GWLS

As indicated in the introduction, the motivation behind this study is to gain more accuracy at the central point of the discrete window, because the contributions of the observations in the sense of **EWLS** do not allow for discrimination among the observations. Based on this, a good choice for $W(u)$ is a Gaussian function with unit amplitude, zero mean, and standard deviation σ: $W(u) = \exp\{-u^2/2\sigma^2\}$. The product $W(u)W(v)$ is therefore a two–dimensional Gaussian kernel focused at the center of the discrete window. Because the error in the weighted least squares problem is graded more importantly when the weight is large, and less importantly when the weight is small, the result of the applied Gaussian weight can be summarized with the additional focus introduced at the central pixel location of the discrete window. With reference to the consistency property established, the **GWLS** technique is able to maintain the same accuracy of the **EWLS** technique when the signal is homogeneous. The advantages of the **GWLS** technique over **EWLS** lie in the fact that the latter can damp the effects of deviant pixels at the edges of the discrete window. This occurs especially when the discrete window approaches a step discontinuity. The one–dimensional set of Gaussian Weighted Discrete and Orthogonal Polynomials are computed as follows:

$$\phi_0(u) = 1 \qquad \phi_1(u) = u - B_1 \ ,$$

where

$$B_1 = \frac{< u,\ 1 >}{< 1,\ 1 >} = \frac{\sum_{u=-M}^{M} u\ \exp\{-u^2/2\sigma^2\}}{\sum_{u=-M}^{M} \exp\{-u^2/2\sigma^2\}} = 0\ ,$$

since the numerator term, $u \exp\{-u^2/2\sigma^2\}$, is odd. Therefore, $\phi_1(u) = u$. Polynomial $\phi_2(u)$ is given by $(u - B_2)\phi_1(u) - C_2\phi_0(u)$, where

$$B_2 = \frac{< u\phi_1,\ \phi_1 >}{< \phi_1,\ \phi_1 >} \quad = \quad \frac{\sum_{u=-M}^{M} u^3 \exp\{-u^2/2\sigma^2\}}{\sum_{u=-M}^{M} u^2 \exp\{-u^2/2\sigma^2\}}\ ,$$

$$C_2 = \frac{< u\phi_1,\ \phi_0 >}{< \phi_0,\ \phi_0 >} \quad = \quad \frac{\sum_{u=-M}^{M} u^2 \exp\{-u^2/2\sigma^2\}}{\sum_{u=-M}^{M} \exp\{-u^2/2\sigma^2\}}\ . \tag{2.4}$$

The term $B_2 = 0$ since, as before, the numerator term is odd. Therefore $\phi_2(u) = u^2 - C_2$, where C_2 is as defined above. Polynomial $\phi_3(u)$ is given by $(u - B_3)\phi_2(u) - C_3\phi_1(u)$, where

$$B_3 = \frac{< u\phi_2,\ \phi_2 >}{< \phi_2,\ \phi_2 >} \quad = \quad \frac{\sum_{u=-M}^{M} u(u^2 - C_2)^2 \exp\{-u^2/2\sigma^2\}}{\sum_{u=-M}^{M} (u^2 - C_2)^2 \exp\{-u^2/2\sigma^2\}}\ ,$$

$$C_3 = \frac{< u\phi_2,\ \phi_1 >}{< \phi_1,\ \phi_1 >} \quad = \quad \frac{\sum_{u=-M}^{M} u^2(u^2 - C_2) \exp\{-u^2/2\sigma^2\}}{\sum_{u=-M}^{M} u^2 \exp\{-u^2/2\sigma^2\}}\ .$$

The first term $B_3 = 0$, because the numerator is odd, as well. The second term, after simplification, becomes

$$C_3 = \frac{\sum_{u=-M}^{M} u^4 \exp\{-u^2/2\sigma^2\}}{\sum_{u=-M}^{M} u^2 \exp\{-u^2/2\sigma^2\}} - C_2\ ,$$

which implies

$$C_2 + C_3 = \frac{\sum\limits_{u=-M}^{M} u^4 \exp\{-u^2/2\sigma^2\}}{\sum\limits_{u=-M}^{M} u^2 \exp\{-u^2/2\sigma^2\}} \ . \tag{2.5}$$

Therefore, Polynomial $\phi_3(u) = u(u^2 - C_2) - C_3 u = u^3 - (C_2 + C_3)u$. Concluding, the Gaussian Weighted Orthogonal Polynomials are given by:

$$\phi_0(u) = 1 \ , \tag{2.6}$$

$$\phi_1(u) = u \ , \tag{2.7}$$

$$\phi_2(u) = u^2 - C_2 \ , \tag{2.8}$$

$$\phi_3(u) = u^3 - (C_2 + C_3)u \ , \tag{2.9}$$

where the constant C_2 is given by Eq. (2.4), and $C_2 + C_3$ is given by Eq. (2.5). The constants α_i are computed as: $\alpha_i = \sum\limits_{u=-M}^{M} \phi_i^2(u) \ \exp\{-u^2/2\sigma^2\}$, where polynomials ϕ_i's are given by Eqs. (2.6) through (2.9). Furthermore, the functions $K_i(u)$ are computed using Eq. (2.2): $K_i(u) = [\exp\{-u^2/2\sigma^2\} \ \phi_i(u)] / \alpha_i$. Note that the functions K_i need to be computed only once for a given set of parameters (window size $N \times N$, where $N = 2M + 1$, and the standard deviation σ of the Gaussian kernel). Finally, the solution coefficients, a_{ij}, are computed using Eq. (2.3), or

$$a_{ij} = \sum_{u=-M}^{M} \sum_{v=-M}^{M} f(u,v) K_i(u) K_j(v) \ .$$

2.4.1 Computation of Derivatives in Range Images

As mentioned previously, the range image is approximated locally using

$$g(u,v) = \sum_{i+j \leq n} a_{ij} \phi_i(u) \phi_j(v) \ .$$

For a second order polynomial approximation, $n = 2$, the expansion of $g(u,v)$ yields

$$g(u,v) = a_{00} - C_2(a_{20} + a_{02}) + a_{10}u + a_{01}v + a_{11}uv + a_{20}u^2 + a_{02}v^2 \ . \qquad (2.10)$$

The derivatives of the range image $f(u,v)$ are computed using the approximating polynomial function $g(u,v)$ [ref. Eq. (2.10)]. The first and second order partial derivative estimates are:

$$g_u(u,v) = a_{10} + a_{11}v + 2a_{20}u \quad , \quad g_v(u,v) = a_{01} + a_{11}u + 2a_{02}v \ ,$$

$$g_{uv}(u,v) = a_{11} \quad , \quad g_{uu}(u,v) = 2a_{20} \quad , \quad g_{vv}(u,v) = 2a_{02} \ .$$

At the center of the discrete window, $(u,v) = (0,0)$, the estimated derivatives of $f(u,v)$ then become

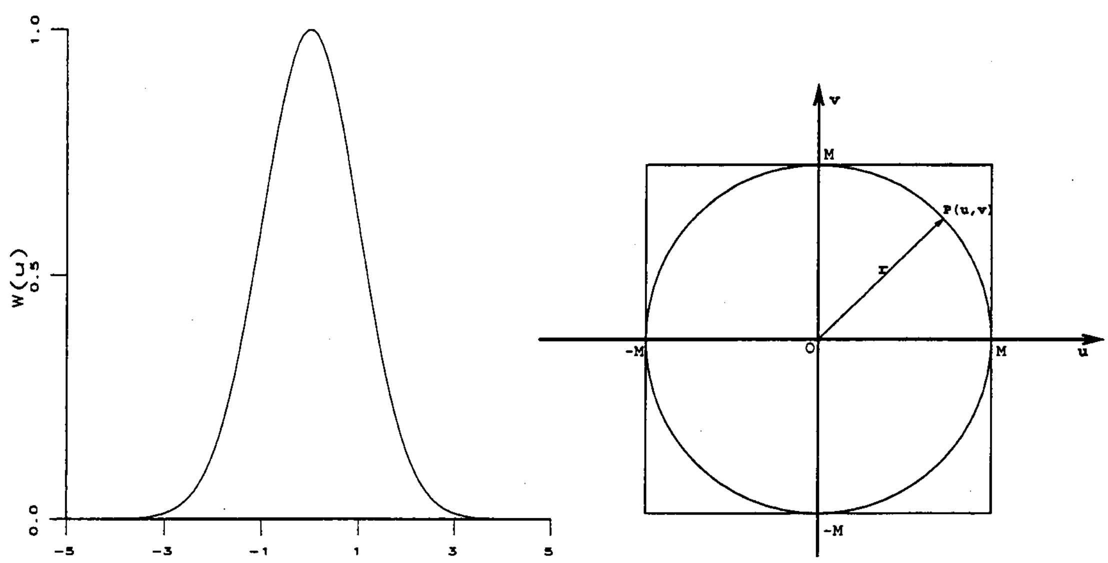
$$f = a_{00} - C_2(a_{20} + a_{02}) \quad , \quad f_u = a_{10} \quad , \quad f_v = a_{01} \ , \qquad (2.11)$$

$$f_{uv} = a_{11} \quad , \quad f_{uu} = 2a_{20} \quad , \quad f_{vv} = 2a_{02} \ , \qquad (2.12)$$

where the coefficients a_{ij} are computed using Eq. (2.3).

2.4.2 Convolute Integers for Smoothing and Differentiation

Our main interest in the development of the **GWLS** technique lies in the estimation of the derivatives at each pixel location of the range image that can be used for surface characterization. Consider the solution coefficients to the weighted least squares problem [Eq. (2.1)]. At the center of the discrete window (the point of interest), the computed first and second order derivatives are direct functions of these coefficients [Eqs. (2.11) and (2.12)]. The solution coefficients, after



(a) (b)

Figure 2.1: *(a) One–dimensional profile of a Gaussian function. (b) Domain of a two–dimensional Gaussian kernel.*

---

substitution for the weight function in Eq. (2.1), become

$$a_{ij} = \frac{1}{\alpha_i \alpha_j} \sum_{u=-M}^{M} \sum_{v=-M}^{M} \exp\{-(u^2+v^2)/2\sigma^2\} f(u,v)\phi_i(u)\phi_j(v) \ . \tag{2.13}$$

The constant term $\alpha_i \alpha_j = \sum_{u=-M}^{M} \phi_i^2(u) \ \exp\{-u^2/2\sigma^2\} \sum_{u=-M}^{M} \phi_j^2(u) \ \exp\{-u^2/2\sigma^2\}$ which, after changing the second summation variable to v and grouping, yields the following form for the solution coefficients:

$$a_{ij} = \frac{\sum_{u=-M}^{M} \sum_{v=-M}^{M} \exp\{-(u^2+v^2)/2\sigma^2\} f(u,v)\phi_i(u)\phi_j(v)}{\sum_{u=-M}^{M} \sum_{v=-M}^{M} \exp\{-(u^2+v^2)/2\sigma^2\} \phi_i^2(u)\phi_j^2(v)} \ . \tag{2.14}$$

The numerator and denominator of the previous equation involve a convolution with a two–dimensional Gaussian kernel. Since the Gaussian is known to decrease rapidly away from its maximum, it is expected that away from the center of the kernel, additional terms of each summation become negligible. Figure 2.1(a) represents a one–dimensional profile of a Gaussian kernel. Consider that we have a two–dimensional profile of the same kernel, and Fig. 2.1(a) is a cross–section of the actual profile. The domain of the two–dimensional profile, in the (u, v)–plane, is shown in Fig. 2.1(b). With reference to Fig. 2.1(b), the distance between the origin O of the coordinate system and the point $P(u, v)$ is $r = \sqrt{u^2 + v^2}$. Since

$$W(u)W(v) = \exp\{-(u^2 + v^2)/2\sigma^2\} = \exp\{-r^2/2\sigma^2\} \ , \tag{2.15}$$

we conclude that the assigned weight carries an equal magnitude for pixels along the circle with radius r [Fig. 2.1(b)]. The objective is to tailor the Gaussian kernel to the $N \times N$ discrete window, $N = 2M + 1$, such that additional terms outside the given window carry a weight smaller or equal to a pre–specified constant threshold (ϵ). With the aid of Fig. 2.1(b) and Eq. (2.15), we let $r = M$, and choose σ such that $\exp\{-(M+1)^2/2\sigma^2\} \leq \epsilon$ for some $\epsilon > 0$. Since $M = (N-1)/2$, the previous inequality yields $(1/\epsilon)\exp\{-(N+1)^2/8\sigma^2\} \leq 1$. After applying the natural log, we get the following upper–bound for the standard deviation σ:

$$\sigma \leq \frac{N+1}{2\sqrt{2\ln(1/\epsilon)}} \ .$$

A crisp value of σ is obtained using the upper–bound of the standard deviation for $\epsilon = e^{-7} \approx 10^{-3}$. Based on this, the correspondence between N and σ is established:

$$\sigma = \frac{N+1}{2\sqrt{14}} \approx 0.134 \times (N+1) \ . \tag{2.16}$$

Table 2.1: *Table of $K_i(u)$ for $(N = 5, \sigma = 0.80)$.*

u	$44 \times K_0(u)$	$14 \times K_1(u)$	$92 \times K_2(u)$	$12 \times K_3(u)$
−2	1	−1	9	−1
−1	10	−5	10	2
0	22	0	−38	0
1	10	5	10	−2
2	1	1	9	1

Table 2.2: *Table of $K_i(u)$ for $(N = 7, \sigma = 1.07)$.*

u	$136 \times K_0(u)$	$50 \times K_1(u)$	$77 \times K_2(u)$	$226 \times K_3(u)$
−3	1	−1	2	−4
−2	9	−6	6	−3
−1	33	−11	−1	18
0	50	0	−13	0
1	33	11	−1	−18
2	9	6	6	3
3	1	1	2	4

Table 2.3: *Table of $K_i(u)$ for $(N = 9, \sigma = 1.34)$.*

u	$289 \times K_0(u)$	$129 \times K_1(u)$	$248 \times K_2(u)$	$380 \times K_3(u)$
−4	1	−1	2	−2
−3	7	−5	7	−4
−2	28	−14	9	3
−1	65	−16	−7	13
0	86	0	−22	0
1	65	16	−7	−13
2	28	14	9	−3
3	7	5	7	4
4	1	1	2	2

As mentioned previously, the functions $K_i(u)$ [Eq. (2.2)] are uniquely defined for a given (N, σ) pair. Integer approximations of these kernels are shown for $N = 5, 7$, and 9 in Tables (2.1) through (2.3). The coefficients of the fitting polynomial are retrieved using these kernels following Eq. (2.3), which were shown to be direct functions of the derivative estimates at the center of the local window [Eqs. (2.11) and (2.12)]. Smoothing of range images is achieved by evaluating the polynomial surface representation at the center of the discrete window (shown as f in Eq. (2.11)). Note that smoothing can also be achieved using the derived kernels K_i, since the constant C_2 of Eq. (2.11) can be written as $\sum_{u=-M}^{M} u^2 K_0(u)$. The derived kernels are therefore a set of convolute integers useful for range image smoothing and differentiation.

2.5 Surface Characterization of Range Images

In this section, we use the estimated derivatives to compute more complex measures to provide descriptive boundary information of the range image. The range image represents a two-dimensional manifold embedded in a 3-D space. Each point of the range image is assigned three coordinates (u, v, w), where (u, v) are the curvilinear coordinates of the surface. An equivalent vector representation of the surface as follows:

$$\mathbf{r} = (u, v, g(u, v))^T .$$

- The unit normal vector, $\mathbf{n}$, to the range image at the coordinates (u, v) is the unit vector perpendicular to the tangent-plane. Since the vectors $\mathbf{r}_u$ and $\mathbf{r}_v$ are tangent vectors to the surface, the vector given by the cross-product $\mathbf{r}_u \times \mathbf{r}_v$ is perpendicular to $\mathbf{r}_u$ and $\mathbf{r}_v$. Therefore, a vector normal to the surface is also collinear with the cross-product vector. The unit normal vector is given by

$$\mathbf{n} = \frac{\mathbf{r}_u \times \mathbf{r}_v}{||\mathbf{r}_u \times \mathbf{r}_v||} = \frac{(-f_u, -f_v, 1)^T}{\sqrt{1 + f_u^2 + f_v^2}} ,$$

which is a three-dimensional vector. A gray scale representation of the normal is computed using the angle, β, between the two projections of the unit normal onto the (u, w) and (v, w) planes. The angle representation is given by

$$\beta = \tan^{-1}(f_v / f_u) \ .$$

Roof boundaries represent the orientation discontinuities of the range image. The scalar β is used to enhance the internal discontinuities of the range image. False edges generally result when dealing with spherical shapes. As indicated by Besl [29], one way to deal with this problem is to apply folding on the resulting image.

- The gradient magnitude image which is an edge magnitude image, known to detect step discontinuities, is computed as follows

$$GM = \sqrt{f_u^2 + f_v^2} \ .$$

- The quadratic variation is computed as follows

$$Q = f_{uu}^2 + 2f_{uv}^2 + f_{vv}^2 \ .$$

Integrating the quadratic variation over a region of the range image results in a measure of flatness (or smoothness) of that region [29].

- The Gaussian curvature K, and the mean curvature H are given by [29]

$$\begin{aligned} K &= \frac{f_{uu} f_{vv} - f_{uv}^2}{(1 + f_u^2 + f_v^2)^2} \ , \\ H &= \frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{2(1 + f_u^2 + f_v^2)^{3/2}} \ . \end{aligned}$$

The mean curvature can be used jointly with the Gaussian curvature to classify each pixel of the range image into 8 surface types [29].

- The two principal curvature k_1 and k_2 can be computed using the Gaussian and the mean curvature as follows

$$\begin{aligned} k_1 &= H + \sqrt{H^2 - K}\ , \\ k_2 &= H - \sqrt{H^2 - K}\ . \end{aligned}$$

The principal curvature, k_1 and k_2, are associated with the principal directions, whereas the Gaussian and mean curvature are direction–independent [29].

- The images GM, $|k_1|$, $|H|$, and $CM = \sqrt{H^2 - K}$ can be interpreted as edge magnitude images [29]. A non–maxima suppression algorithm can be applied to isolate the relevant edges of the range image (step and roof discontinuities).

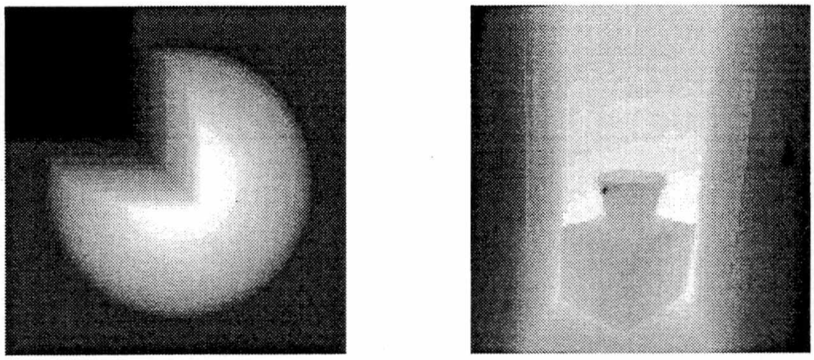
- The curvature can also be computed separately for each direction θ [38] as follows

$$k_\theta = \frac{f''_\theta}{(1 + f'^2_\theta)^{3/2}} \sqrt{\frac{1 + (f_u \cos\theta - f_v \sin\theta)^2}{1 + f_u^2 + f_v^2}}\ .$$

It follows that the curvature evaluated in the u– and v–directions (k_u and k_v) are given by

$$\begin{aligned} k_u &= \frac{f_{uu}}{\sqrt{(1 + f_u^2)(1 + f_u^2 + f_v^2)}}\ , \\ k_v &= \frac{f_{vv}}{\sqrt{(1 + f_v^2)(1 + f_u^2 + f_v^2)}}\ . \end{aligned}$$

2.5.1 Comparisons of EWLS and GWLS Techniques



(a) (b)

Figure 2.2: *(a) Synthetic image* SPHERE, *(b) Real image* ROOM.

In order to illustrate the effects of the standard deviation on the estimated derivatives, we compute the gradient magnitude image, GM, and the curvature magnitude image, CM, for the synthetic image of Fig. 2.2(a), SPHERE, using the **GWLS** and **EWLS** techniques for $N = 5, 7$, and 9 [Fig. 2.3 and Fig. 2.4].

The results of the generated gradient magnitude images, GM, are shown in Fig. 2.3. This image clearly illustrates the boundary thickening problem of the **EWLS** technique. As the size of the discrete window is increased, from $N = 5$ to $N = 9$, we note a severe expansion of the boundaries at the base of the sphere for the case of **EWLS** [Fig. 2.3(f)]. The presence of boundary pixels at the edges of the discrete window caused the transition region of the detected features to be very wide. As clearly shown in Figs. 2.3(a),(c), and (e), the boundary expansion problem was minimized through the application of a Gaussian weight. Figure (2.3) shows the smearing effect, of **EWLS**, on the resulting gradient magnitude image

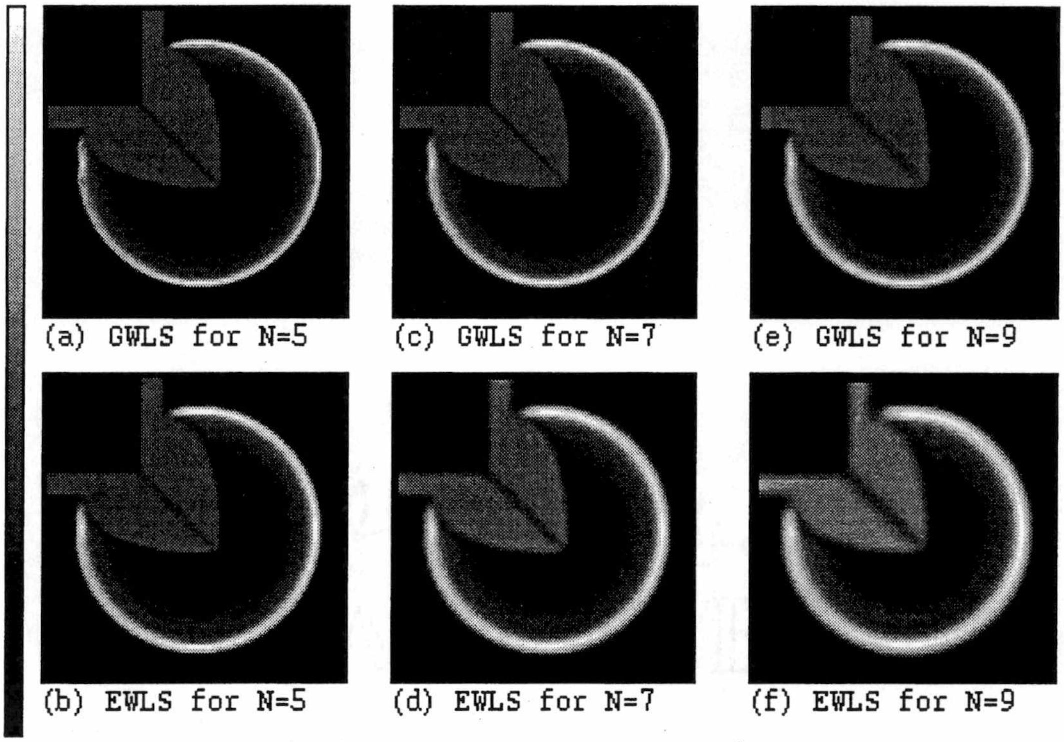


Figure 2.3: *Gradient magnitude images of synthetic image* SPHERE.

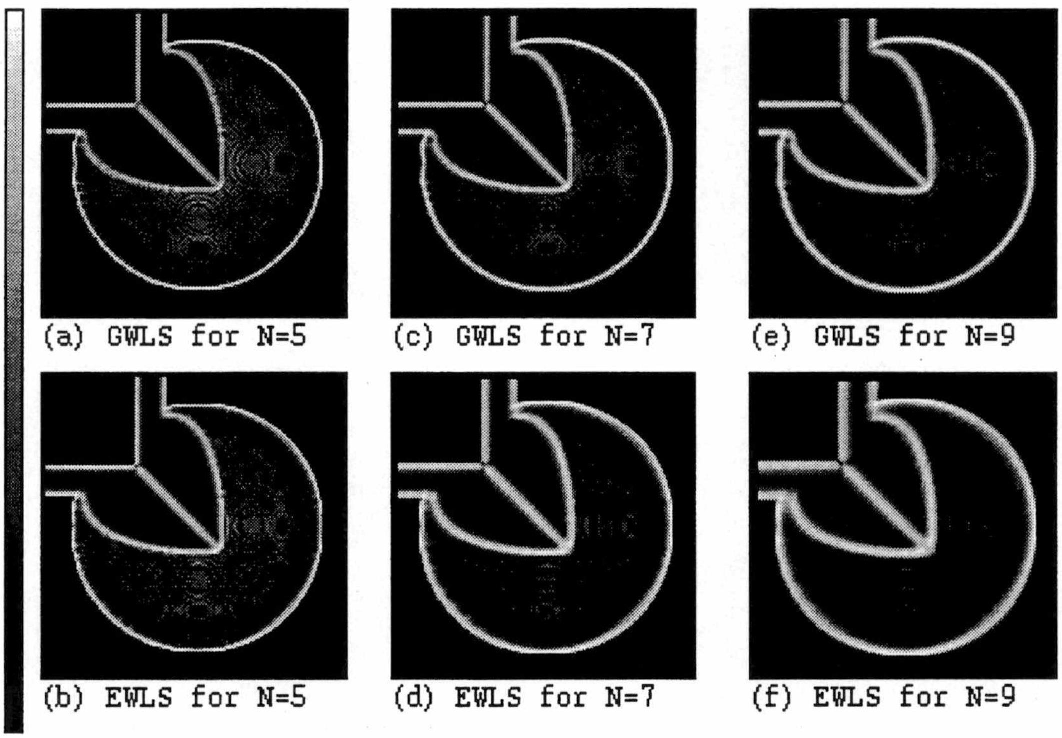


Figure 2.4: *Curvature magnitude images of synthetic image* SPHERE.

as the size of the window is increased.

The results of the generated curvature magnitude images, CM, are shown in Fig. (2.4). This image illustrates the same problem for **EWLS** technique in generating thick roof edges. The roof edges, which divide the spherical object into partial hemispheres, exhibit wider transition regions for the case of **EWLS**, and for all window sizes used [Figs. 2.4(b), (d), and (f)]. We also note the additional smoothing which results from the use of larger windows for the case of **GWLS**. Even with the use of the third order polynomials, the obtained second order derivatives are unchanged. It requires a minimum of a fourth order polynomial to acquire somewhat thinner roof boundaries. The additional cost in terms of the number of unknowns a_{ij}, 16 vs. 6, and the negative effects of using higher order polynomials in terms of the higher sensitivity to spurious pixel values makes the **GWLS** technique more suitable for derivative estimation. The advantages of the second order polynomial approximations, as was previously indicated, lies in the additional smoothing [8], which indicates the robustness of this approach. In conclusion, we have obtained similar results when computing second order differences [Fig. 2.4], which clearly demonstrates the superiority of the **GWLS** technique over the **EWLS** technique.

2.6 Summary

This chapter investigated the utility of the Gaussian Weighted Least Squares technique in providing accurate derivative estimates. The approach was compared to the classical Equally Weighted Least Squares technique and shown to be superior for all window sizes commonly used. The importance of this algorithm lies in its ability to provide edge maps in a variety of scales. This module will therefore be employed in an effort to simplify the segmentation problem.

CHAPTER 3

Range Segmentation via Data Fusion and Watersheds

3.1 Introduction

Morphology is a powerful classical tool in image processing. The application of the morphological watershed in segmentation has been focused on electrophoresis gel images [14]. A typical gel image can be decomposed into ridges and valleys. The distinction among ridges and valleys yields the desired segmentation of the gel image. The morphological watershed has been applied also to intensity images after the appropriate transformations, to the original image, have taken place. Examples of transformations are the morphological gradient, or any existing technique that computes the gradient image. Range images, however, are characterized by two principle types of discontinuities (step and roof discontinuities). The procedure that transforms the range image to a form suitable for the application of the morphological watershed is therefore not trivial. This Chapter discusses how to apply the **GWLS** technique in order to transform a range image to a form suitable for the application of the morphological watershed. The algorithm combines information from the range image and the surface normal image using the appropriate fusion methodologies. A block diagram of the algorithm is shown in [Fig. (3.1)].

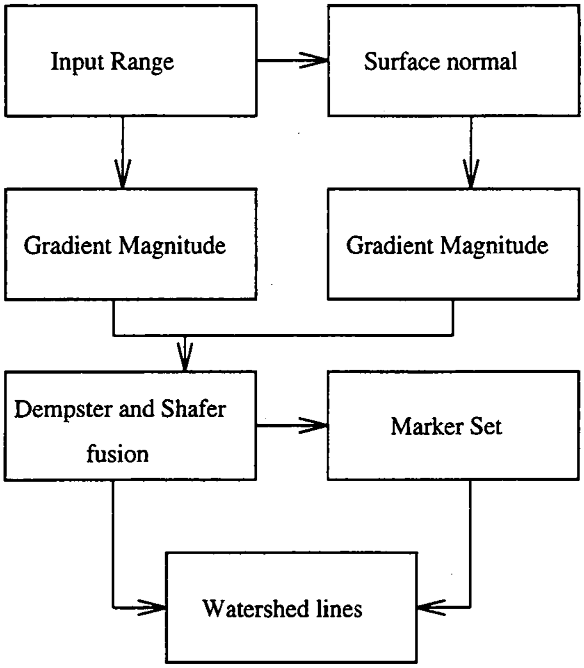


Figure 3.1: *Block diagram of gray-scale watershed segmentation algorithm.*

3.2 Transforming Range Images to a Form Suitable for Watersheds

First we apply the cosine correction, which is described in Chapter 1, in order to transform the angular measurements of the range image into direct depth measurements. A 3×3 median filter is subsequently applied to the resulting image to eliminate shot noise, which is common in range images [36]. The gradient magnitude, $GM = \sqrt{f_u^2 + f_v^2}$, and the surface normal image [1, 29, 35]

$$\Theta = \tan^{-1}(f_v/f_u)$$

are subsequently determined using the **GWLS** technique for $N = 5$. An upper and a lower threshold is applied to the resulting surface normal image, Θ, to eliminate pixels at which one the computed partials, f_u or f_v, is very small. The values used are 5% of the maximal value of Θ for the lower threshold, and 95% for the upper threshold. A 5×5 median filter is subsequently applied to the resulting image to provide a smooth surface normal image. The gradient magnitude of the resulting surface normal image, Θ, is computed using the **GWLS** technique for $N = 5$. As can be foreseen, the gradient magnitude of the input range image, which we call g_s, is used to highlight the depth discontinuities of the range image, and the gradient magnitude of the surface normal image, which we call g_r, is used to highlight the orientation discontinuities of the range image. Data fusion techniques are subsequently applied to both images, g_s and g_r, to yield a complete gray scale boundary description of the range image. Experimental evidence has shown the utility of Dempster and Shafer rule of combination in combining these two images. This fusion methodology was compared to and shown experimentally to outperform standard averaging techniques and the SuperBayesian rule of combination [1, 4, 26]. The primary reason for this superior performance is the inherent tendency of Dempster and Shafer rule of combination to over–estimate the probabilities and subsequently yield distinct valleys in the resulting image (an important consideration in a gray scale watershed algorithm). Given the belief images g_s and g_r, the Dempster and Shafer rule of combination yields an combined belief image g_{DS} given by [4]

$$g_{DS} = 1 - (1 - g_s) \times (1 - g_r) \ .$$

This image, g_{DS}, is used as input to the higher level morphological watershed algorithm, which is presented in the next section.

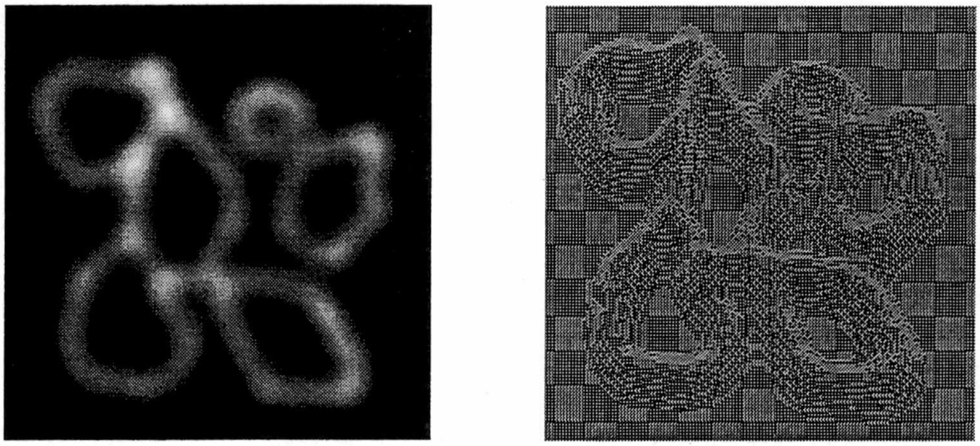


Figure 3.2: *3–D perspective of a typical edge magnitude image.*

3.3 Segmentation via Morphological Watershed

The morphological watershed is a powerful tool that helps eliminate or simplify ill–posed problems of thresholding and linking to achieve a binary image, often called an edge map image [4]. The input to the watershed algorithm is an edge magnitude image, similar to g_{DS}, where the intensity conveys the edginess of a particular point (or pixel) in the range image. The edge magnitude image, in 3–D perspective, is composed of a group of valleys (points with low intensity) surrounded by ridges (points with high intensity) [Fig. (3.2)]. These points, of the edge magnitude image, are classified into three categories [Fig. (3.3)]:

1. Points at which a drop of water, if placed on top, would not drift in any direction. These points are called minima points, or regional minima points.

2. Points at which a drop of water, if placed on top, would drift to a single minima. These points belonging to a single minima form the catchment

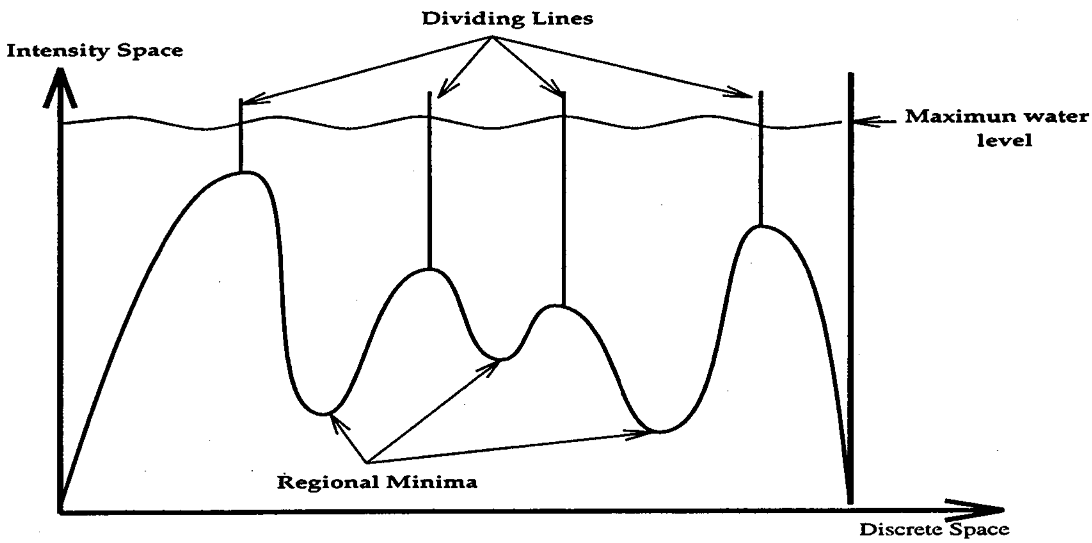


Figure 3.3: *One–dimensional illustration of gray scale watershed algorithm.*

basin of that minima.

3. Points at which a drop of water, if placed on top, would fall to more than one minima. These points form the dividing lines.

To put the problem in perspective, the valleys of the edge magnitude image are catchment basins. The point (or points) with the lowest gray tone intensity in each catchment basin represent the regional minima. The points that separate two or more distinct valleys are the dividing lines, which represent the objective of the watershed algorithm.

The idea used to find the dividing lines is very simple. If we punch a hole in each regional minima and allow water levels to rise gradually in the entire topography of the edge magnitude image, we reach a level where collected water of one catchment basin merges with another. In this context, the allowed water levels for the case of an edge magnitude image are discrete numbers. For an 8-bit edge magnitude image, the water level ranges from 0 to 255. As a direct result, raising the water level refers to incrementing a value threshold by one. Points below the threshold are flooded points, and points above the threshold are also above the current water level. The dividing lines are built such that, during the most recent iteration, the set of points which resulted in merging of two or more distinct catchment basins are transformed into dams. The transformation is achieved simply by projecting the gray tone intensity of these points to the maximum gray level allowed (255 in this case). This procedure continues until the entire image is flooded, for a water level of 254, and only the dams lie above the water level. These dams are separated, which yields an edge map image of the original edge magnitude image. A mathematical formulation of this algorithm is found in [4, 14, 33].

For practical considerations, the primary problem with this approach is the high dependability on an edge magnitude image, the result of differentiation of the original range image. Because of the higher sensitivity of differentiation to noise, a typical edge magnitude image contains a large number of minima, which generally results in an over-segmentation. A better result is achieved when the regional minima of interest are seed regions provided by a higher level process. A simple technique was shown experimentally to provide reliable seed regions is to threshold g_{DS} at 10% of its maximal value. The seed regions are the set of points of the edge magnitude image that lie below this threshold. The success of this technique is primarily due to the effectiveness of the **GWLS** technique in providing reliable derivative estimates, and the contribution of the Dempster–Shafer fusion methodology in yielding an adequate edge magnitude image. With this implemented, the primary task of the watershed algorithm is to grow the seed regions and ignore the minima of the edge magnitude image. A mathematical formulation of this algorithm is as follows [4]:

Let g_{DS} be an edge magnitude image, and $T[n]$ be the threshold set image containing the set of points (u, v) such that $g_{DS}(u, v) \leq n$, where n is the current water level. Let $R[n-1]$ be the catchment basin image at stage $n-1$. Suppose also that $R[n-1]$ is known, $T[n]$ is known, and the objective is to find $R[n]$. Let $R[n-1] = \cup C_i, \quad i = 1, \cdots, N$, where C_i and C_j are distinct for all $i, j \leq N$, and $T[n] = \cup B_i, \quad i = 1, \cdots, M$. Then, at stage n, given a region of $T[n]$, namely B_k, there are three possibilities:

1. $B_k \cap R[n-1] = \emptyset$. In this case, a new regional minima has occurred that is not present in the catchment basin image. $R[n-1]$ is not updated (case ignored).

2. $B_k \cap R[n-1]$ forms a single connected component. In this case, an existing region of the catchment basin image has expanded without causing any merging. The catchment basin image is updated with B_k, and $R[n] = R[n-1] \cup B_k$.

3. $B_k \cap R[n-1]$ forms more than one connected component. In this case,

merging has occurred, and the dam is built. The dam is the Geodesic Skiz of the intersection image, $B_k \cap R[n-1]$, inside B_k.

The primary consideration remaining lies in the method used to compute the Skiz at any point where merging of two or more distinct catchment basins occurs; this allows for one pixel wide dams, which is a desirable property. The technique used to find the Skiz is to *conditionally dilate* $B_k \cap R[n-1]$ inside B_k, and append the pixels which do not cause the merging of the distinct catchment basins of $R[n-1]$ [33].

3.4 Range Image Segmentation Results

Figures 3.4(a)–(f) illustrate the different steps required to transform a range image into a form suitable for the application of the morphological watershed. Figure 3.4(a) shows the input synthetic range image, and Fig. 3.4(b) shows the surface normal image. Figure 3.4(c) shows the gradient magnitude image of Fig. 3.4(a). This image clearly shows the range discontinuities present in the outer rim of the spherical object, but does not show any significant response to the edges at the base of the pyramid. Figure 3.4(d) shows the gradient magnitude of the surface normal image [Fig. 3.4(b)]. The orientation discontinuities, at the base of the pyramid, are clearly brought out in this image. Neither of the two gradient images provides a complete edge feature description of the range image. The Dempster–Shafer formalism successfully combined both gradient magnitude images to yield a complete gray scale boundary description of the original range

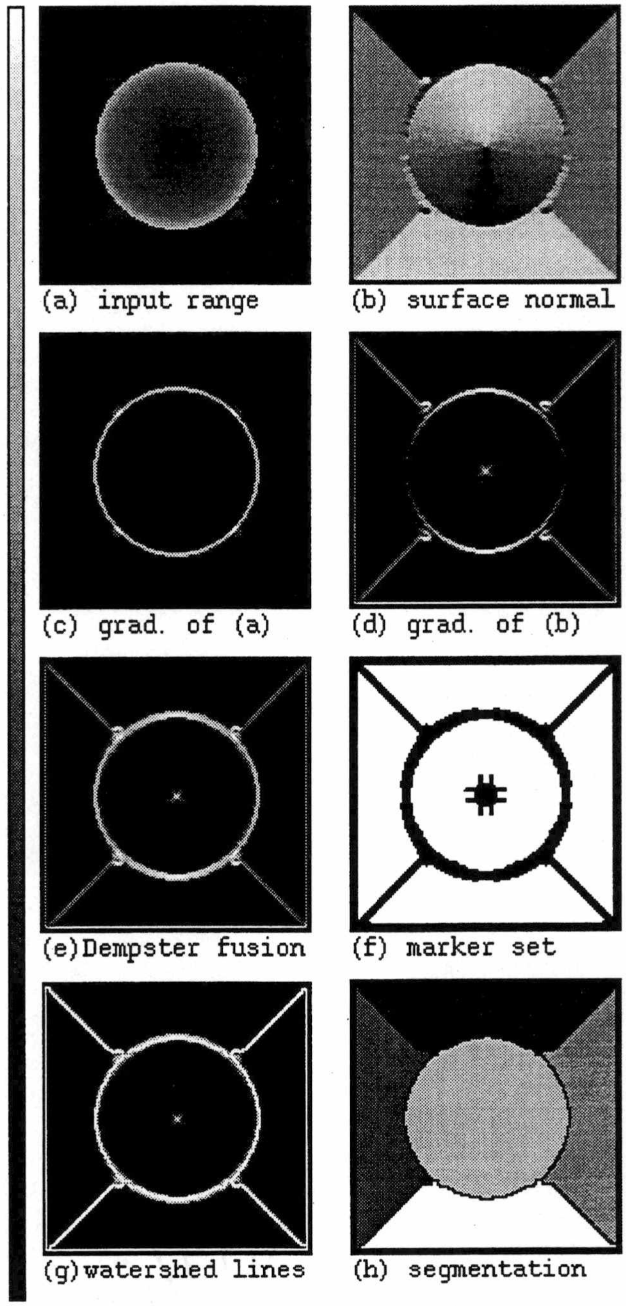


Figure 3.4: *Segmentation Algorithm using Gray Scale Watershed (a) Synthetic range image (b) The surface normal of (a), (c) Gradient Magnitude Image of (a), (d) Gradient Magnitude Image of (b) (e) Dempster–Shafer fusion of (c) and (d), (f) Marker Set Image (10% threshold on (e)), (g) Dividing line Image overlapping (e), (h) Final Segmentation via Watersheds*

image [Fig. 3.4(e)]. Figure 3.4(f) shows, in white, the marker set used as starting regions for the watershed segmentation algorithm. This image was obtained by thresholding the gradient image at 10% of its maximum possible value. We found experimentally that a low threshold, in this approximate range, yielded successful results over a large set of range images. In fact, all results discussed below were obtained using a 10% threshold. This relative insensitivity is due in no small part to the effectiveness of the GWLS technique and the fusion methodology in providing reliable edge maps that required minimal post processing prior to watershed segmentation. Figure 3.4(g) shows the watershed segmentation lines superimposed on the edge map of Fig. 3.4(e). Note the thinness and accuracy of these lines as compared to the lines obtained by simply fusing the gradient images. Finally, Fig. 3.4(h) shows the resulting watershed segmentation image.

Figure 3.5 shows the watershed segmentation process at various stages of flooding (threshold values). The one–pixel–thick lines that persist throughout the remaining steps of iteration are dams (dividing lines) built to prevent flooding between distinct catchment basins. The remaining black pixels correspond to pixels that are either above the current water level or below the current water level that were not allowed to form new catchment basins. Only the regions associated with the initial marker set are giving rise to new potential catchment basins. If these were not ignored, the eventual result would be oversegmentation. The final segmentation result, shown in Fig. 3.5(h), demonstrates the capability of the watershed algorithm to extract all relevant edges with a high degree of precision.

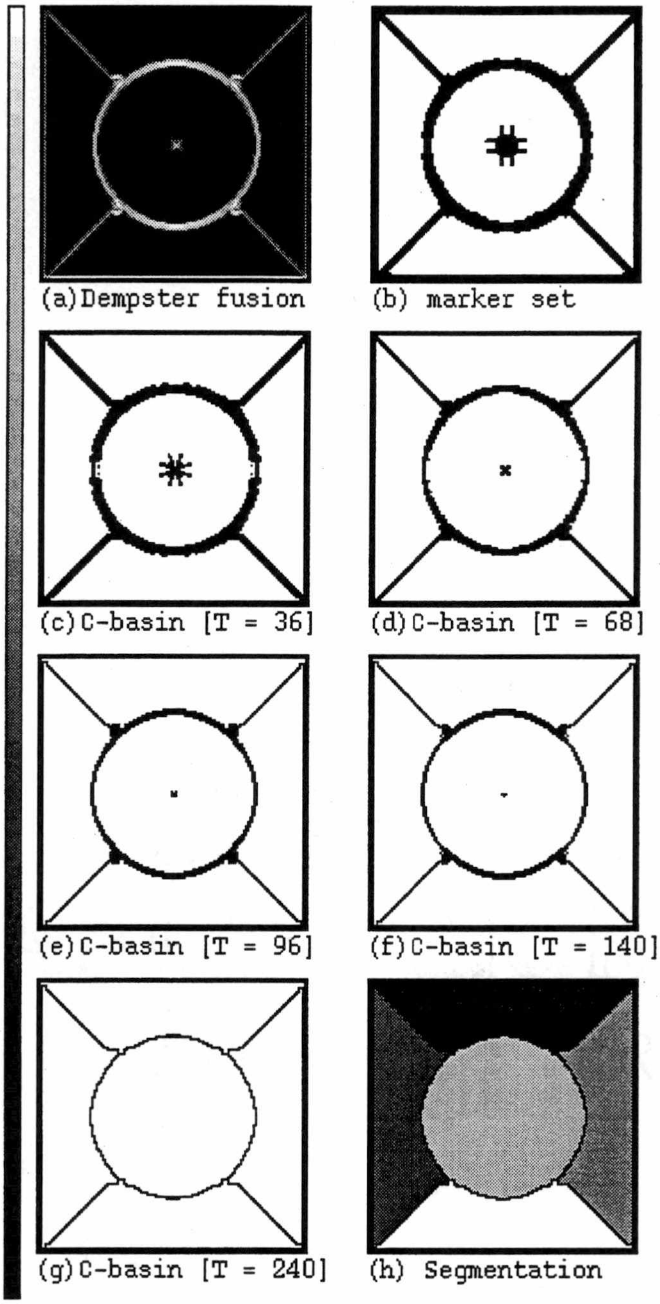


Figure 3.5: *Catchment-basin image at various flooding levels (a) Dempster–Shafer Fusion of Step and Roof Edge-maps for the Synthetic Range Image in Fig. 2(a), (b) 10% threshold of (a) Used as a Marker Set, (c) Flooding level = 36, (d) Flooding level = 68, (e) Flooding level = 96, (f) Flooding level = 140, (g) Flooding level = 240, (h) Final Segmentation Resulting in One-pixel Wide Dams*

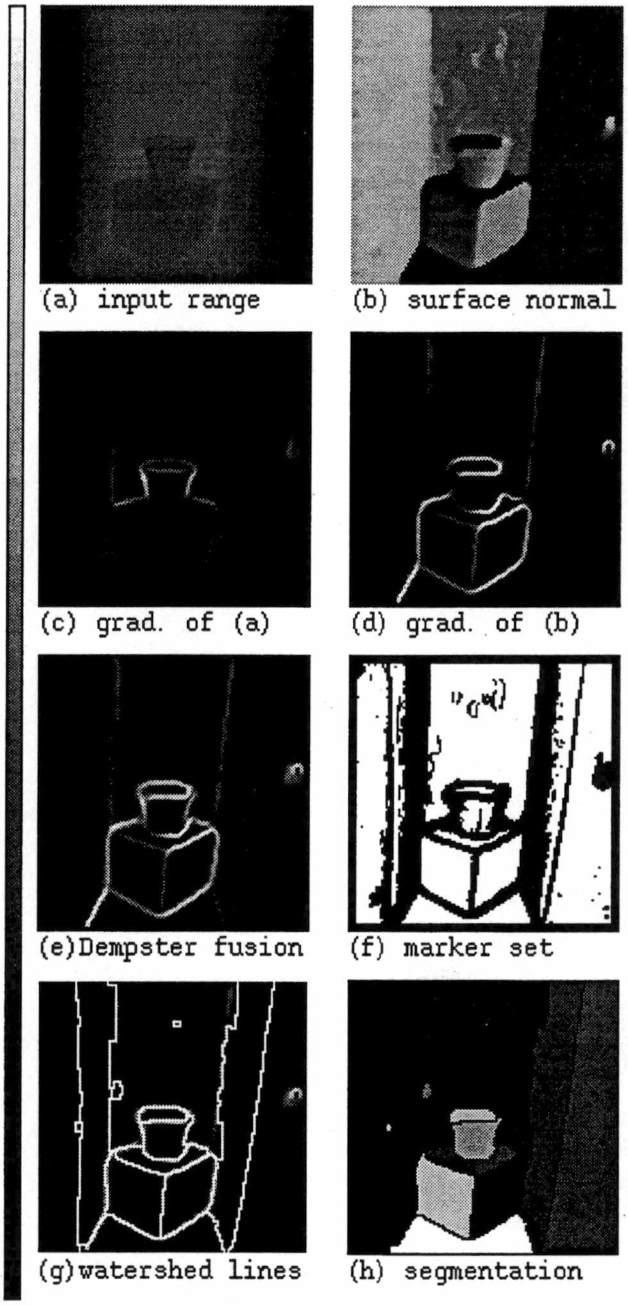


Figure 3.6: *Segmentation Algorithm using Gray Scale Watershed (a) Real range image (b) The surface normal of (a), (c) Gradient Magnitude Image of (a), (d) Gradient Magnitude Image of (b) (e) Dempster–Shafer fusion of (c) and (d), (f) Marker Set Image (10% threshold on (e)), (g) Dividing lines Image overlapping (e), (h) Final Segmentation via Watersheds*

Figure 3.6(a) shows a real range image obtained using an Odetics laser scanner [28]. Figure 3.6(b) shows the surface normal image that highlights the internal boundaries of the box. The gradient image shown in Fig. 3.6(c) displays all depth discontinuities with a high degree of precision, which also indicates the robustness of the GWLS technique. Figure 3.6(e) shows the fused image using the Dempster–Shafer formalism. Figure 8(f) shows the marker set, obtained using a 10% of the maximum value threshold on Fig. 3.6(e). Note that the shapes of the seed regions are rather coarse in comparison to the edges themselves, a condition that is remedied by the final watershed result shown superimposed on the edge map in Fig. 3.6(g). The final segmentation is seen to be a highly descriptive representation of an original image that is difficult to analyze by visual inspection.

Figure 3.7(a), depicting a real operating environment, was obtained at the Oak Ridge National Laboratory using an Odetics scanner. Figure 3.7(b) shows the power of computing a normal surface representation using the GWLS technique, since all of the major surfaces are clearly highlighted in this image. As in all of the results shown, step edges (Fig. 3.7c) and roof edges (Fig. 3.7d) were reliably extracted using the GWLS technique for derivative estimation from range images. As mentioned earlier, neither edge image constitutes a reasonably complete edge map of the original. In particular, note that the top edge of the left barrel is incomplete in both edge images. This provides a vivid demonstration of the usefulness of fusion which, as shown in Fig. 3.7(e), remedied this situation. The watershed algorithm produced a highly accurate scene segmentation of the original range image.

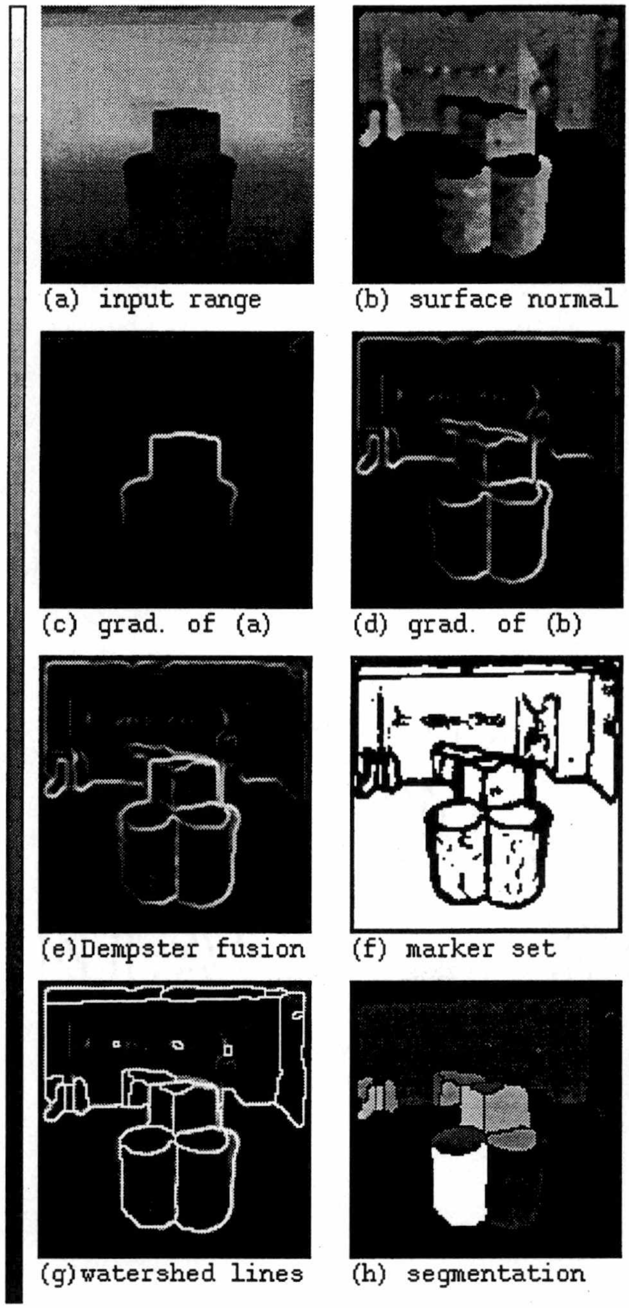


Figure 3.7: *Segmentation Algorithm using Gray Scale Watershed (a) Synthetic range image (b) The surface normal of (a), (c) Gradient Magnitude Image of (a), (d) Gradient Magnitude Image of (b) (e) Dempster–Shafer fusion of (c) and (d), (f) Marker Set Image (10% threshold on (e)), (g) Dividing lines Image overlapping (e), (h) Final Segmentation via Watersheds*

3.5 Summary

It has been demonstrated [4] that rugged, practical segmentation of range images can be achieved by a process that incorporates three fundamental steps: (1) An edge extraction procedure that takes into account discontinuities in depth as well as discontinuities in the direction of surface normals; (2) a data fusion approach capable of producing a single edge map that combines these two types of discontinuities; and (3) a segmentation algorithm flexible enough to capture the principal features of these edge maps under a meaningful variety of conditions. The experimental results presented show that this three-step process is an effective approach for segmenting range images of practical interest.

CHAPTER 4

Fusion of Multi–Scale Edge Maps for Robust Segmentation

4.1 Introduction

In this chapter, the **GWLS** technique is used to extract boundary information from range images at a number of scales. Edge extraction is then followed by a Surface–based Region Splitting algorithm in order to generate a more complete partition of the image. The result is fed into a Surface–based Region Growing algorithm which yields the final segmentation image.

4.2 Edge Detection from Multiple Scales Using the GWLS technique

A Multi–Scale Gaussian Weighted Least Squares technique is developed for extracting the two types of edges from range images. The algorithm computes three gradient magnitude images, GM, and three curvature magnitude images, CM, using the **GWLS** technique for $N = 5, 7$, and 9. The definitions of GM and CM were provided in Chapter 2. Using the computed GM, we generate three binary images, STEP, that contain the desired step boundaries, through the

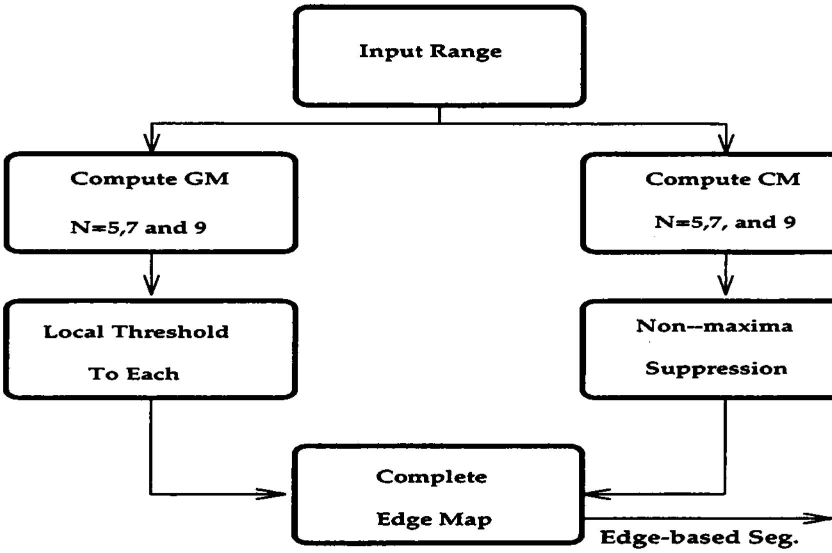


Figure 4.1: *Multi–Scale Feature Extraction Using the GWLS technique.*

---

application of a local threshold [Fig. (4.1)]:

$$STEP = \begin{cases} 1, & \text{if } GM > A_{3\times 3} + \delta \text{ and } GM \geq T \\ 0, & \text{otherwise} \end{cases},$$

where $A_{3\times 3}$ is the local average over a 3×3 window centered at the pixel of interest. The constant δ is chosen experimentally as 5% of the global average of GM. Finally, T is chosen as 10% of the maximum value of GM. Naturally, the gradient magnitude image, GM, for $N = 9$ is the least sensitive to noise and is used to detect boundary pixels. The localization of boundary pixels relies on the smallest scale which corresponds to $N = 5$. The gradient magnitude image for $N = 7$ provides additional evidence to the presence of boundary pixels [26] and, consequently, enhances the reliability of the detection. The merging of the resulting STEP images is done through a binary **AND** operator; no scale–space tracking is needed:

proof: Without loss of generality, since the gradient magnitude GM is rotationally symmetric, we let $f(u,v)$ be a vertical unit step edge given by $f(u,v) = \text{step}(v - v_0)$. The gradient magnitude $\nabla = GM$ is computed using the estimated derivatives

$$\nabla = \sqrt{f_u^2 + f_v^2} = \sqrt{a_{10}^2 + a_{01}^2}\ .$$

The objective is to determine for which value of v_0 does the gradient magnitude ∇ attain its maximum value. The coefficient a_{10} can be shown to equal 0, which, after substitution in ∇ and expansion, gives

$$\nabla = |a_{01}| = \left| \frac{1}{\alpha_1} \sum_{v=-M}^{M} v \exp\{-v^2/2\sigma^2\} \text{step}(v - v_0) \right| \ .$$

The partial derivative of ∇ with respect to v_0 is

$$\begin{aligned} \frac{\partial \nabla}{\partial v_0} &= \text{sign}(a_{01}) \times \frac{-1}{\alpha_1} \sum_{v=-M}^{M} v \exp\{-v^2/2\sigma^2\}\ \delta(v - v_0) \\ &= \frac{-v_0 \exp\{-v_0^2/2\sigma^2\}}{\alpha_1} \times \text{sign}(a_{01}) \quad \text{if } v_0 \in [-M, M], \text{ 0 otherwise .} \end{aligned}$$

Assuming that $v_0 \in [-M, M]$, which means that the step edge passes through the discrete window. The maxima of ∇ are solutions of $\frac{\partial \nabla}{\partial v_0} = 0$, or $v_0 \exp\{-v_0^2/2\sigma^2\} = 0$. Therefore, the value of v_0 at which the gradient magnitude attains its maximum value is $v_0 = 0$, or when the step edge passes through the center of the discrete window. In conclusion, for any arbitrary window size, the gradient magnitude computed using the **GWLS** technique provides localized step edges. The Multi–Scale Gaussian Weighted Least Squares technique overcomes the scale–space tracking problem [38] resulting from the use of a number of Gaussian smoothing kernels before applying differentiation. Because of the use of the **GWLS** technique, we obtain the desirable smoothing for a larger σ while maintaining the location of the features intact.

Using the curvature magnitude images, CM, we generate three binary images, ROOF, that contain the desired roof boundaries, through a non–maxima suppression algorithm [29, 38]. In the current implementation, a pixel is considered a boundary pixel *if and only if* its actual curvature magnitude, CM without scaling, exceeds 0.125 ($CM > 0.125$). The merging of the generated three roof edge maps is, again, through a binary **AND** operator. The resulting step edge discontinuities, STEP, and roof edge discontinuities, ROOF, are combined using a logical **OR** operator yielding a complete boundary description of the input image. Because of the nature of differentiation, when computing a gradient magnitude image using a window of size $N = 9$, the border pixels of the range image are not included in the computation. Four border rows and columns surrounding the combined edge map are filled with 1's to close this gap. The resulting image is negated which leads a binary image we call EDGES. This image is labeled using a four–connectedness algorithm [3], and the connected components of the resulting image that total less than 10 pixels are subsequently eliminated. This yields a preliminary edge–based segmentation we reference by INITIAL. The input range image and INITIAL are then fed into a Region Splitting algorithm [Fig. 4.2].

4.3 Surface–based Region Splitting Algorithm

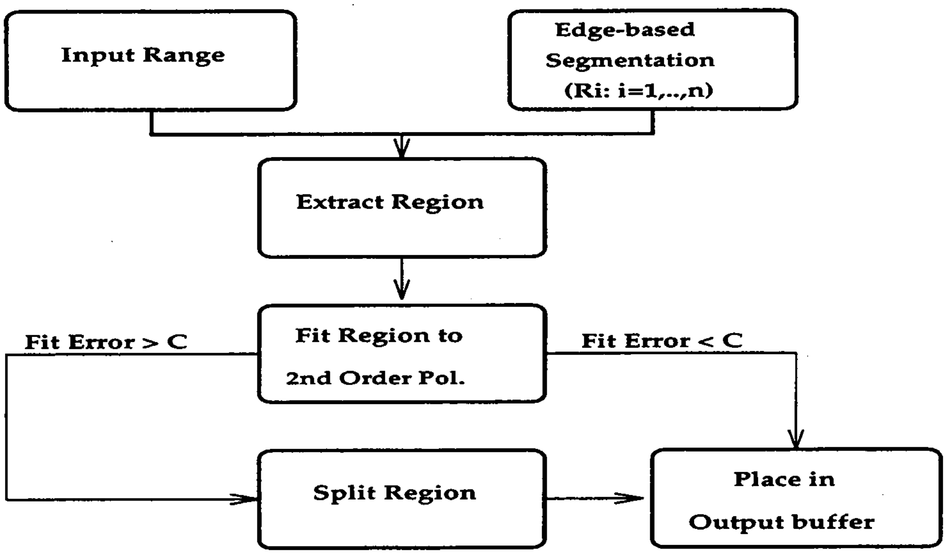


Figure 4.2: *Region splitting algorithm generating* SPLIT *image.*

Each region, $\mathbf{R}$, of the range image corresponding to a connected component of INITIAL is fitted to a second order polynomial surface, $P(u,v)$, using the Equally Weighted Least Squares surface fitting algorithm of arbitrary shaped regions [29]:

$$P(u,v) = a_{00} + a_{10}u + a_{01}v \ + a_{11}uv \ + a_{20}u^2 + a_{02}v^2.$$

If the absolute deviation error norm,

$$\eta = 1/N \sum_{(u,v)\in\mathbf{R}} |f(u,v) - P(u,v)| \ ,$$

between the fitting polynomial, $P(u,v)$, and the original data set, $f(u,v)$, is lower than a predefined maximum tolerance error, η_0, the connected component is placed into a new image buffer we call SPLIT. If $\eta > \eta_0$, the connected component is split into its homogeneous subcomponents in the following manner:

The set of pixels of the connected component in a 3×3 window are retained in SPLIT image *if and only if* the discrete window contains exactly 9 pixels with intensity 1 [14, 17, 33]. The initial region is decomposed, in this fashion, if the gap between disconnected edges is less than 3 pixels (this has been justified experimentally; however, larger windows can be applied to deal with larger ruptures in the edge map). At this step, elongated shapes might become fragmented, which generally results in an over partitioning of the non–homogeneous region. This problem is dealt with during the next region growing phase which allows for these fragmented regions to gain their original shape. The generated connected subcomponents are subsequently expanded inside the initial non–homogeneous connected component of INITIAL via *conditional dilation* [14, 33], without rupturing the linkage established from the previous step. This procedure, as can be foreseen, augments the size of the generated seed regions, and subsequently maximizes the reliability of the region growing algorithm, since the larger seed region infers a more accurate interpretation about the nature of the actual region. A detailed description of this technique, in the most general case, is as follows:

A square window of size $N \times N$, is moved around the binary non–homogeneous region in a sequential fashion (left to right and top to bottom). At each of these steps, we compute the total number of pixels inside the $N \times N$ window that possess the intensity 1. The entire window is copied into the output binary image, LINK, *if and only if* the computed total number of pixels is exactly N^2. As can be foreseen, this process allows us to isolate the homogeneous portions inside the non–homogeneous region, which systematically excludes edge regions or regions where the gap between broken edges is less than N pixels. The output image, LINK, is labeled using an 8–connectivity connected component algorithm [3]. LINK then is *conditionally dilated* inside the non–homogeneous region using the 8–connectivity mask structuring element. During each iteration, we append to LINK the pixels at which a 3×3 window, if centered in the pixel, would contain labels from a single region. The procedure stops when no additional pixels are appended to LINK, which yields the desirable edge linking of the broken edges in the non–homogeneous region.

This procedure continues until all the connected components of SPLIT correspond to a homogeneous region that can be fitted to a second degree polynomial surface within the bounds of the maximum tolerance allowed, η_0. The resulting image is labeled using a four–connectedness algorithm. The labels of the image SPLIT are ordered with respect to the size of each region (from larger seeds to smaller). Regions that total less than 10 pixels are eliminated because they form unreliable seeds for region growing. The input range image, and the labeled image, SPLIT, are fed into the following Region Growing algorithm [Fig. 4.3].

4.4 Surface–based Region Growing Algorithm

Each seed region of the input image corresponding to a connected component of SPLIT is grown based on its polynomial surface representation generated using a variable order surface fitting algorithm of arbitrary shaped regions: In this algorithm, the initial assumption about the seed region is a planar surface.

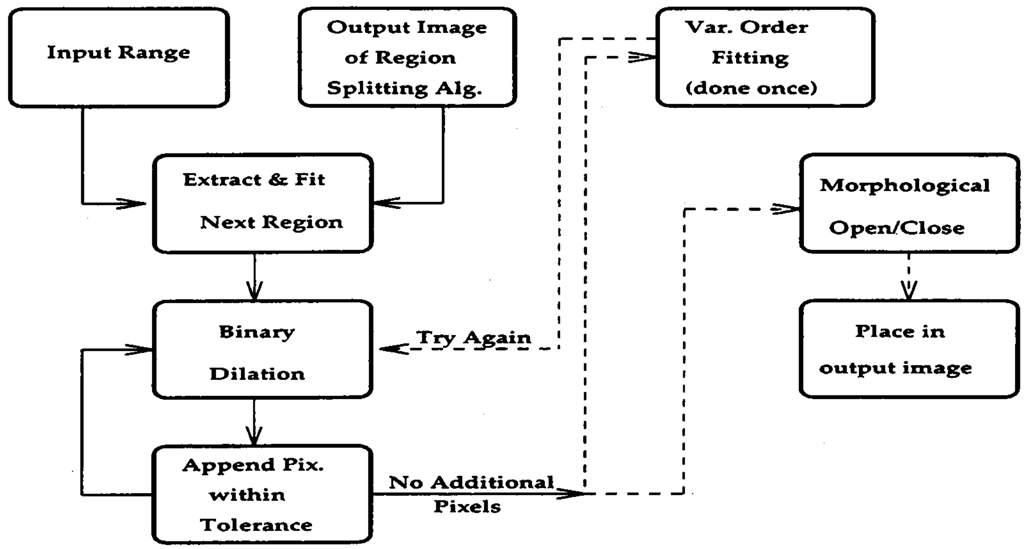


Figure 4.3: *Block diagram of Region Growing algorithm generating* FINAL *segmentation image.*

If the absolute deviation error norm is larger than a predefined tolerance error, η_0, the order of the fitting polynomial is increased. The maximum polynomial order allowed is 3. The connected component is iteratively dilated using the 8–connectedness mask structuring element. At each of these steps, among the added pixels to the connected component we retain those, (u, v), which are not element of a previously grown seed region, and at which the property $|f(u, v) - P(u, v)| \leq \eta_0$ is satisfied. This procedure is applied recursively until no additional pixels to the connected component satisfy the stated conditions, in which case, the surface representation of the newly formed seed region is updated and the order of the polynomial is increased if $\eta > \eta_0$. The procedure stops either when we reach the maximum order allowed, 3, and the condition $\eta \leq \eta_0$ fails; or, if the updated surface representation of the seed region, where $\eta \leq \eta_0$, does not allow for additional pixels to be merged. In this case, the region has grown to its full natural dimensions. The resulting connected component is subsequently *opened* and *closed* (standard morphological *opening* and *closing* operators, which refers to the sequence {*dilation; erosion; erosion; dilation*}) using the 8–neighborhood mask structuring element. The morphological *opening* of the resulting connected component ensures that the obtained region is homogeneous, therefore, it allows the inclusion of deviant pixels which occur inside the connected component. The morphological *closing* operator ensures the convexity of the obtained region. Convexity of the region implies that for any given pair of points inside the region, the line segment drawn between the two points forms a new set of points which are also element of the region. The result, after *opening* and *closing*, is placed into a new image, FINAL, and assigned a unique label. Furthermore, a seed region of SPLIT is totally eliminated if it becomes a subset of a previously grown seed region. In the case of partial occlusion, the new seed region is the component that is not enclosed in the grown seed region. If this component contains more than 10 pixels, it is subsequently grown using the same concepts discussed earlier. Otherwise, it is rejected and the next seed region of SPLIT is evaluated (the first seed region is more reliable because its initial dimension is larger). The process continues until all the seeds of SPLIT are either grown or rejected based on the stated conditions, which yields the final segmentation image, FINAL.

4.5 Range Image Segmentation Results

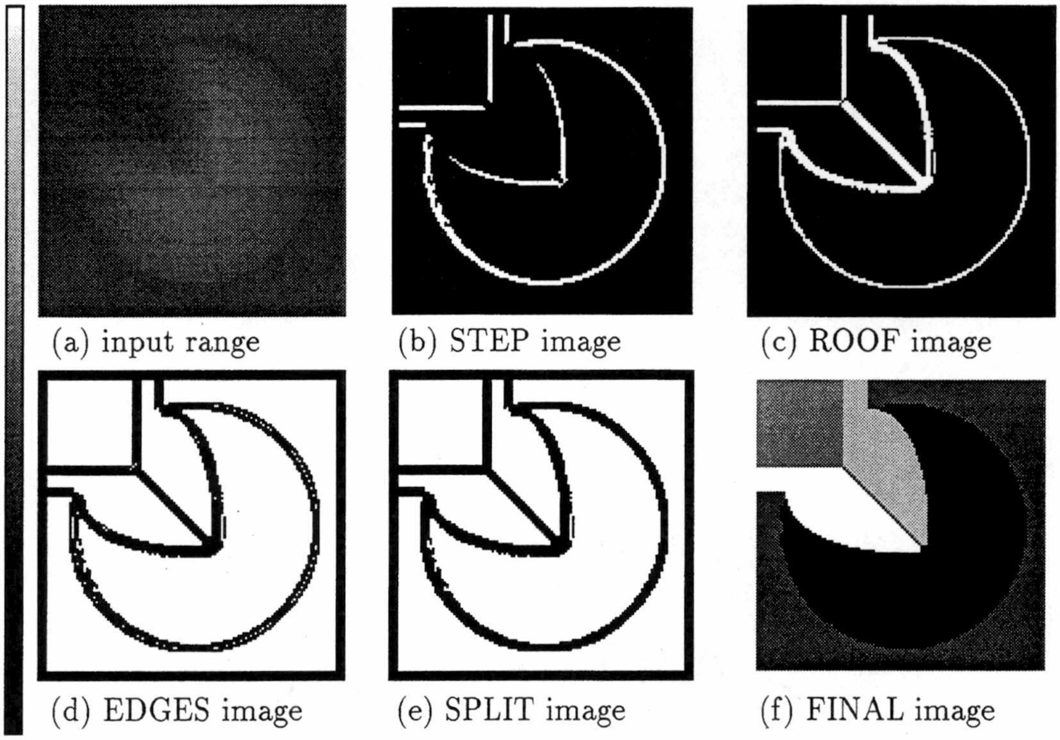


Figure 4.4: *Segmentation of synthetic image* SPHERE.

Figures 4.4(b),(c) and (d) illustrate the effectiveness of the described Feature Extraction algorithm in providing a complete unbroken edge map for the image SPHERE. Figure 4.4(f) shows the high degree of accuracy of the resulting scene segmentation. There is a rationale behind the selection of the value of η_0 based on the generated set of fit error values from the Region Splitting algorithm. The obtained set, for the image SPHERE, is the following: $\{0.0, 0.0, 0.48, 0.48, 1.22\}$. The value 1.22 is clearly deviant from all other values obtained and was, subsequently, interpreted to correspond to a non–homogeneous region (although it was due to the limitation of the variable order surface fitting algorithm in describing spherical shapes accurately). Therefore, the value of η_0 was selected small enough to invoke

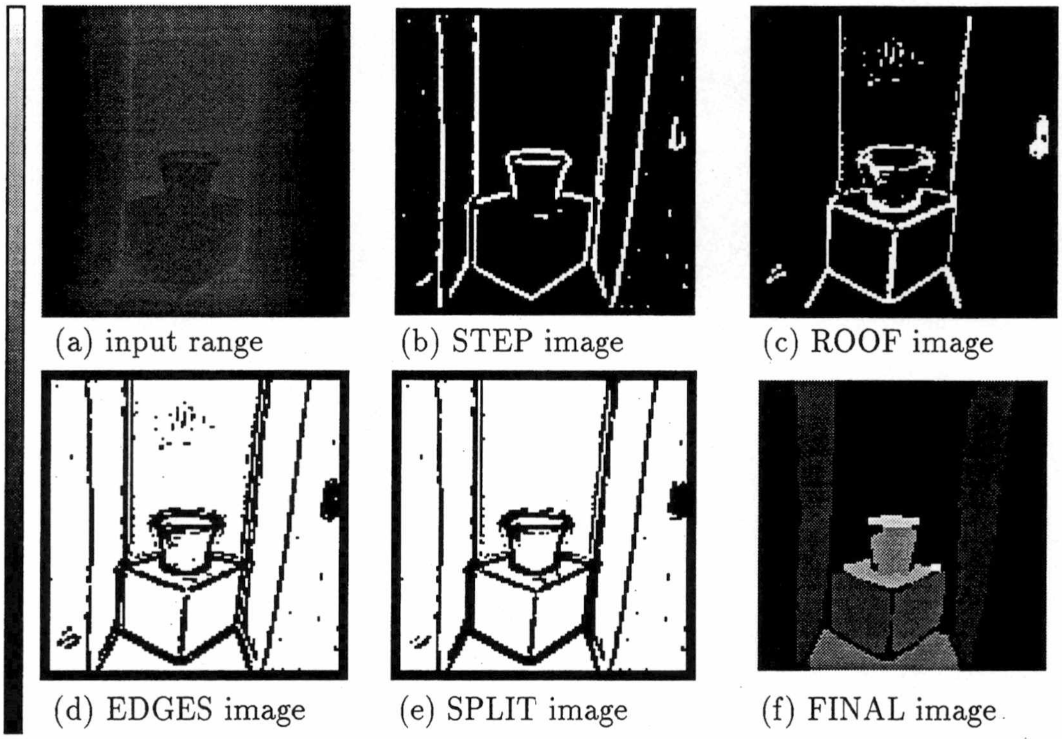


Figure 4.5: *Segmentation of real image* ROOM.

the splitting of the spherical region (1.0) which, since no edges are found within the region, did not lead to a decomposition. The value of η_0 used for growing is not as critical to select; it is chosen always larger than, 1.0 which corresponds to an error magnitude of a single gray level, but is not to exceed the splitting constant ($\eta_0 = 1.0$). For these reasons, the value of η_0 used to grow the seed regions of SPLIT was chosen to be 1.0 ($\eta_0 = 1.0$).

Figure 4.5(b) and (c) illustrate the reliability of the Feature Extraction algorithm in identifying all relevant edges of ROOM with minor breaks in the resulting edge map and very little noise contents [Fig. 4.5(d)]. The obtained fit error set for the image ROOM, based on the connected components of the initial segmentation EDGES of Fig. 4.5(d), is as follows: {0.27,0.30,0.33,0.37,0.41,0.42,0.58,1.28,1.99}. The values 1.28 and 1.99 are clearly deviant from the remaining values of the generated fit error set. Based on this, we select $\eta_0 = 1.0$ to invoke the splitting of these two non-homogeneous regions. Figure 4.5(e) shows that we obtained the desirable splitting of the left side–wall, which corresponds to the fit error value of 1.28, and of the front side of the box, which corresponds to the fit error value of 1.99, that was decomposed into 5 new connected components. This resulted in an oversegmentation to the top surface of the box (3 connected components instead of 1). Figure 4.5(f) shows that the obtained oversegmentation was eliminated, to a reasonable level, after the application of Region Growing to the largest of the original 3 seed regions (with $\eta_0 = 1.0$ as previously mentioned).

The image SHAPES [Fig. 4.7(a)] was corrupted during the acquisition phase, as clearly shown for the frontal side of the box in Fig. 4.7(a). Figure 4.7(b) illustrates a very descriptive STEP image of the original image SHAPES. However, the computation of CM, Fig. 4.7(c), amplified the noise present in this image, and subsequently resulted in false detection. As can be foreseen, the computed second order derivative estimates have a much higher sensitivity to noise than first order differences. Figure 4.7(f) illustrates the ability of the algorithm to circumvent such high levels of noise, which are common for real range data [36], yielding a very accurate and descriptive scene segmentation image. The generated set of fit error values for the image SHAPES, based on the initial partition EDGES of Fig. 4.7(d), is as follows: {1.31,1.38,1.82,2.26,3.05,3.21,4.27,5.55,10.08}. The values 5.55 and

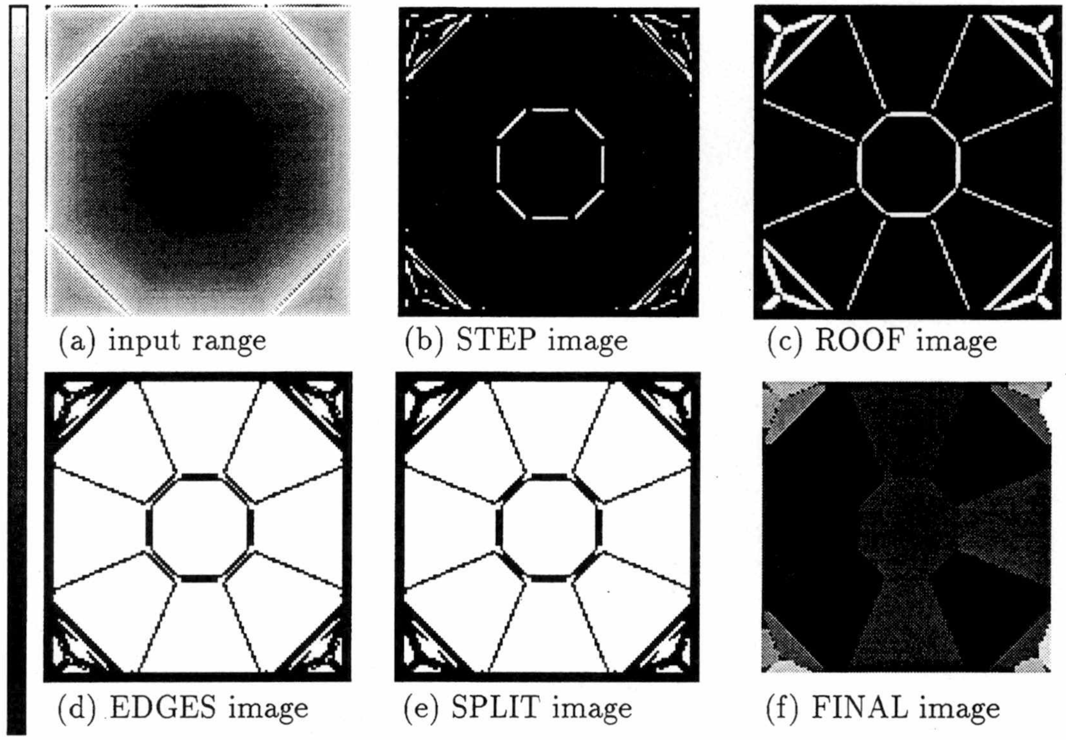


Figure 4.6: *Segmentation of synthetic image* PLANES.

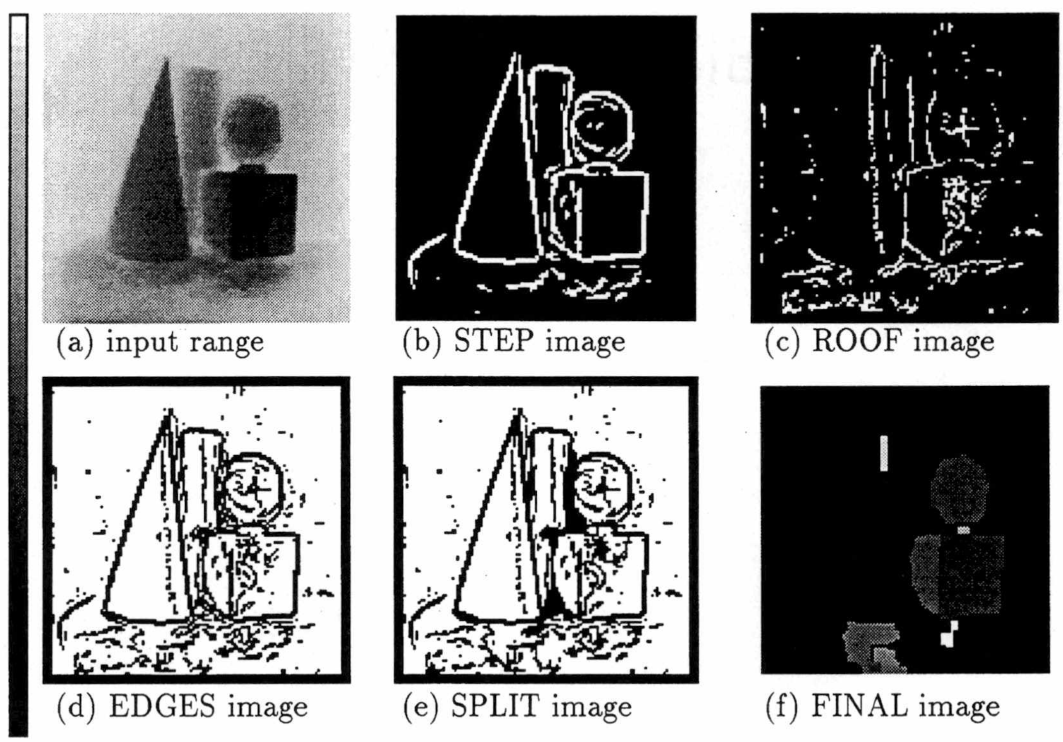


Figure 4.7: *Segmentation of real image* SHAPES.

10.08 were interpreted to correspond to non–homogeneous regions, and $\eta_0 = 5.0$ was chosen to invoke splitting of these non–homogeneous regions that correspond to the background and box regions. The value of $\eta_0 = 4.0$ was used to grow the seed regions of the generated SPLIT image [Fig. 4.7(e)], which is based on the 4.27 generated fit error value that is element of the set. The importance of this two–step algorithm can be seen with the aid of Fig. 4.7(d), where linking of broken edges can cause severe oversegmentation of already existing homogeneous and closed regions, such as the conical, cylindrical, and spherical regions. In terms of computation, there are significant advantages in checking first before linking broken edges that may, in reality, be the result of noise in the obtained edge map.

4.6 Summary

In this chapter, we investigated the utility of the **GWLS** technique in providing reliable step and roof edge maps from multiple scales. As illustrated using synthetic and real range images, which have practical value and various complexity degree, the algorithm provides a very descriptive and reliable scene segmentation. The turning question is on the selection of η_0 for any given image. One way to select the splitting constant, η_{0S}, is to perform surface fitting on all seed regions of INITIAL, and clustering the obtained values of the fit error into two groups: acceptable and unacceptable. Region Splitting is subsequently applied to non–homogeneous regions. The value of the region growing constant, η_{0G}, should be large enough, and is not to exceed η_{0S}, in order to allow for significant improvements in the results. The described segmentation algorithm provides additional information about each region such as the center of the region, the polynomial surface representation with respect to the center of the region, the size of each region in number of pixels, and a unique label for each of these regions that allows us to isolate the homogeneous components of the range image in order to facilitate further analysis and interpretation of the original range image.

CHAPTER 5

Robust Regression Estimates for Range Data Processing

This chapter investigates the asymptotic behavior of location estimates intermediate to the sample mean and the sample median. The location estimates are extended to provide solutions of multi–parameter regression problems applied to fitting arbitrary shaped regions in range images. The algorithm is illustrated with extensive experimental evidence that show the usefulness of intermediate estimates in identifying accurate surface parameters.

5.1 Introduction

Many physical factors contribute noise to the discrete measurements in range images [36], which lead us to consider alternative techniques for estimating accurate surface parameters from raw range data. In classical statistics, an assumption is often made about the nature of the error distribution, such as the Normality or the Laplacian distribution assumption, then proceed to estimate the desired surface parameters. However, the true distribution rarely satisfies the model assumption since it is only an idealized approximation of reality. The discrete measurements are in most cases contaminated where contamination means that the actual error distribution deviates from the assumed model. The effects of contamination on the actual error distribution are often identified with an elongated tail as a mixture of noise populations. This is where classical procedures fail to provide reliable estimates even under very small contamination levels, which indicates their limited use in real applications.

The motivation behind robustifying classical statistical techniques has been the inability of classical statistics to provide reliable estimates when the assumptions are slightly violated. There are many approaches in robust statistics, such as the Huber minimax theory [30], and the Infinitesimal approach of Hampel [16]. Other approaches are simply non–parametric relying on the empirical measure, but imposing constraints such as global smoothness on the distribution. Such methods try to approximate the actual distribution with a function before solving for the unknown parameters. The goal of robust statistics, as defined by Hampel, is to study the effects of small departures from the model on the estimate. Switching from the mean to the median is a step towards robustifying the estimate in view of long–tailed probability distributions [16]. Huber [30] introduced the gross–error model given by the set $\{F = (1-\epsilon)G(x-\theta) + \epsilon H(x-\theta),\ 0 < \epsilon < 1.\}$, where G represents the assumed model distribution, H represents the deviation from the assumed model, and ϵ represents the percent contamination. Symmetry of H was maintained for location estimates. Huber estimators were derived through placing a bound on the influence curve, IC, and minimizing the asymptotic variance against all possible contaminations, H. The philosophy behind this technique is to fight nature. Nature chooses a given H, the statistician selects IC, and the loss and gain is determined by the asymptotic variance of the estimate, which under mild and general conditions is determined by $V(T,F) = \int IC^2 dF$, where T represents the solution estimate. At the normal distribution, we obtain the commonly known Huber minimax estimators [30].

In this chapter, we consider a family of probability distributions, F_λ, parametrized by λ. The maximum likelihood estimate, assuming the distribution F_λ, varies from the sample median estimate to the sample mean estimate of location as we vary the value of the parameter. The sample median is known to be B–robust but highly inefficient under the normality assumption. The opposite holds for the sample mean estimate. Based on this observation, we consider intermediate values of the parameter λ which yield intermediate estimates of location that combine the advantages of the sample mean and the sample median estimates. We consider extending the applicability of this technique to solving the explicit polynomial surface representation of range data, which is useful for fitting arbitrary shaped regions. This chapter is organized as follows: Section 2 presents the family F_λ of probability distributions needed to derive a variety of maximum likelihood estimates. In Section 3, we derive the properties of the estimates at the heavy tailed error distribution (HTD), a mixture of two normal populations. The HTD is a gross error model that introduces deviations from the normal distribution. A qualitative evaluation and comparisons of the estimates are provided. Section 4 deals with numerical solutions of single–parameter and multi–parameter regression problems. Section 5 is an experimental evaluation using synthetic images with added noise. Section 6 analyzes the effects of the parameter λ on the segmentation of real range images. Finally, a summary is provided in Section 7.

5.2 Statement of the Problem

M–estimation techniques have been applied, on many occasions, to solve range data processing problems. The application of the median filter for eliminating noise from range images is an example of applying M–estimates to solve range data processing problems. The Least Squares techniques, frequently used to provide a surface approximation of discrete range data, is another application of M–estimates in range data processing. M–estimators are important to study because they are the most straightforward and the easiest to implement types of estimators [30]. They can be systematically applied to multi–parameter problems by introducing a cost function of the form $\rho(x_i, \theta_1, \ldots, \theta_n)$ where θ_i represent the desired regression parameters.

Given a set of n *independent observations* X_i and assuming a probability density $f(x,\theta) = f(x-\theta)$, the Maximum Likelihood Estimate of the location parameter θ is one that maximizes the joint probability density of the observations [10]

$$f(x_1, \cdots, x_n/\theta) = \prod_{i=1}^{n} f(x_i/\theta) \rightarrow max! \,. \tag{5.1}$$

In an $N \times N$ window of a range image, the discrete observations represent measurements taken at different time intervals; therefore, we can assume independence. A statistical model for the observations is needed to provide solutions for Eq. (5.1). The outcome is the location parameter θ, which can be viewed as a more reliable estimate of the actual range (distance) from the focal plane of the scanner to the corresponding surface point in the scene.

5.2.1 The Family F_λ of Probability Distributions

The functional $f_\lambda(x-\theta)$, $\lambda \in [1,2]$, given by $f_\lambda(x-\theta) = K_\lambda \exp\{-\alpha_\lambda \left|\frac{x-\theta}{\sigma}\right|^\lambda\}$ is a probability density function with mean θ, and standard deviation σ^2. The parameters α_λ and K_λ are to be determined:

The two unknowns α_λ and K_λ must satisfy the following two conditions:

$$\int_{-\infty}^{\infty} f_\lambda(x-\theta)\, dx = 1 \;, \quad \text{and} \quad \int_{-\infty}^{\infty} (x-\theta)^2 f_\lambda(x-\theta)\, dx = \sigma^2 \;.$$

After changing the integration variable to $y = (x-\theta)/\sigma$, the previous system of equations becomes

$$\begin{aligned} K_\lambda \sigma \int_{-\infty}^{\infty} \exp\{-\alpha_\lambda |y|^\lambda\}\, dy &= 2K_\lambda \sigma \int_0^{\infty} \exp\{-\alpha_\lambda y^\lambda\}\, dy = 1 \;, \\ K_\lambda \sigma^3 \int_{-\infty}^{\infty} y^2 \exp\{-\alpha_\lambda |y|^\lambda\}\, dy &= 2K_\lambda \sigma^3 \int_0^{\infty} y^2 \exp\{-\alpha_\lambda y^\lambda\}\, dy = \sigma^2 \;. \end{aligned}$$

Let $t = \alpha_\lambda y^\lambda$, then $y = (t/\alpha_\lambda)^{(1/\lambda)}$ and $dy = \frac{1}{\lambda\alpha_\lambda}(t/\alpha_\lambda)^{(1/\lambda-1)}$. The resulting system of equations after substitution becomes

$$\frac{2K_\lambda\sigma}{\lambda(\alpha_\lambda)^{1/\lambda}} \int_0^{\infty} t^{1/\lambda-1} \exp\{-t\}\, dt = 1 \;, \quad \text{and} \quad \frac{2K_\lambda\sigma^3}{\lambda(\alpha_\lambda)^{3/\lambda}} \int_0^{\infty} t^{3/\lambda-1} \exp\{-t\}\, dt = \sigma^2 \;.$$

Therefore, the solution to this system of equations in terms of the Gamma function, $\Gamma(x) = \int_0^\infty t^{x-1}\exp\{-t\}dt$, is

$$\alpha_\lambda = \left[\frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)}\right]^{\lambda/2}, \text{and } K_\lambda = \frac{\lambda}{2\sigma\Gamma(1/\lambda)}\sqrt{\frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)}} \;.$$

5.2.2 M–estimates of the Location Parameter

The distribution F_λ depends on the arbitrary parameter λ, it is therefore a family of probability distributions. We choose this model to identify solutions to Eq. 5.1.

After substitution for f_λ ($\sigma = 1$, without loss of generality) in Eq. (5.1), and applying the natural log, we get

$$n \ln(K_\lambda) - \alpha_\lambda \sum_{i=1}^{n} |x_i - \theta|^\lambda \rightarrow max! \ .$$

The latter equation is equivalent to minimizing the second term

$$\sum_{i=1}^{n} \rho(x_i - \theta) = \sum_{i=1}^{n} |x_i - \theta|^\lambda \rightarrow min! \ , \tag{5.2}$$

where $\rho(x_i - \theta)$ is the cost of assigning the value θ. The solutions to Eq. (5.2) are the commonly known L_p estimates [30]. The minima can be found by differentiation with respect to θ. It follows that

$$\sum_{i=1}^{n} \psi(x_i - \theta) = \sum_{i=1}^{n} \frac{|x_i - \theta|^\lambda}{x_i - \theta} = 0 \ . \tag{5.3}$$

Figure 5.1 illustrates the effects of the parameter λ on the shape of the corresponding density function, f_λ:

For $\lambda = 1$, the model density function is

$$f_1(x - \theta) = \frac{1}{\sqrt{2}\sigma} \exp\{-\sqrt{2} \left| \frac{x - \theta}{\sigma} \right| \} \ .$$

The probability distribution is a *Laplacian* distribution, and the maximum likelihood estimate of the location parameter θ is solution of [Eq. (5.3)]

$$\sum_{i=1}^{n} \frac{|x_i - \theta|}{x_i - \theta} = \sum_{i=1}^{n} \operatorname{sign}(x_i - \theta) = 0 \ .$$

The solution estimate is the sample median, which is is obtained through the sorting of the discrete measurements and assigning the mid–order statistic for an estimate of the location parameter θ.

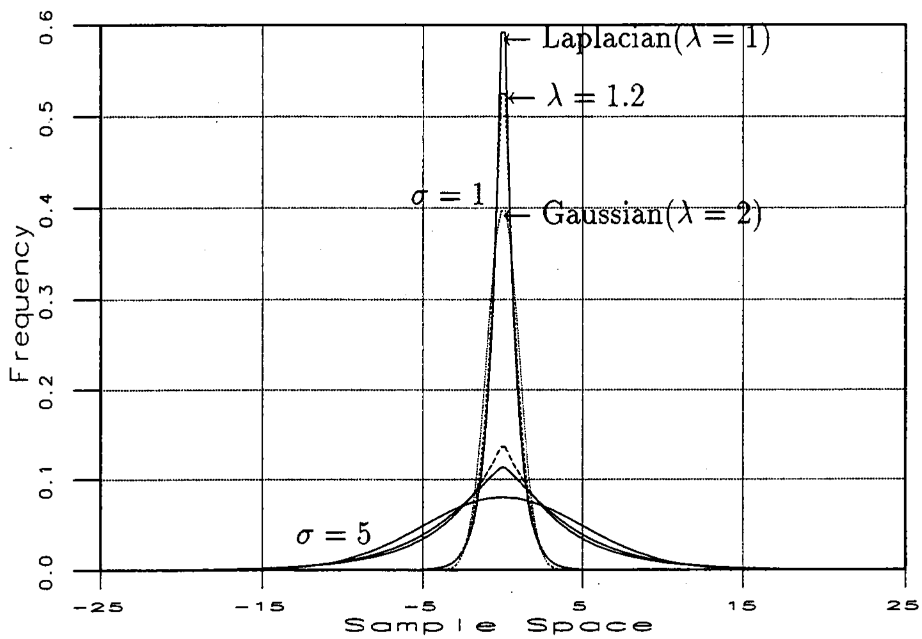


Figure 5.1: *Effects of the parameter λ on the shape of the probability density function. These two clusters correspond to $\lambda = 1, 1.2, 2.0$ for a standard deviation $\sigma = 1$ (top three curves), and 5 (bottom three curves). Note that the distribution for $\lambda = 1.2$ is intermediate to the normal and the laplacian distributions.*

For $\lambda = 2$, the model density function is

$$f_2(x-\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{1}{2}(\frac{x-\theta}{\sigma})^2\} \ .$$

The probability distribution is the *Gaussian*: $\eta(\theta, \sigma)$, and the maximum likelihood estimate of the location parameter θ is solution of [Eq. (5.3)]

$$\sum_{i=1}^{n} \frac{|x_i - \theta|^2}{x_i - \theta} = \sum_{i=1}^{n}(x_i - \theta) = 0 \ .$$

The solution estimate is the sample mean:

$$\theta = \frac{1}{n}\sum_{i=1}^{n} x_i \ ,$$

which is obtained through the application of a pure average on the observations.

The median estimate can sustain very large error; it is a robust estimate of location. It is sensitive to round–off errors, where the error distribution is about the median. The median filter has the capability of eliminating shot noise from range images while preserving the discontinuity features [33]. On the other hand, the sample mean estimate is very sensitive to outlier measurements as opposed to the median. The sample mean is a minimum variance estimate when the error distribution is normal. It is therefore less sensitive to round–off errors or to the sorting of the measurements as opposed to the median estimate. The sample mean estimate, applied to an $N \times N$ discrete window of a range image, results in smearing of the discontinuity features. The sample mean and the sample median estimates are in opposite ends of the spectrum. The questions that need to be addressed at this stage are as follows:

1. Would an estimate for an intermediate value λ have the combined advantages of the mean and the median estimates?

2. If so, to what extent would the advantages of these two estimators be distributed for an an intermediate estimate corresponding to an intermediate value of λ?

3. Is there an optimal λ parameter that provides maximum reliability estimates?

5.3 Qualitative Evaluation and Comparisons of Location Estimates

The most common assumption made on the error distribution in range images is the normality assumption. The deviation from the normal distribution was formalized by Tukey [30] with the heavy tailed distribution, which is a reasonable extension of the normal distribution given by the mixture distribution, F_m, of two normal populations

$$F_m(x-\theta) = (1-\epsilon)\ \Phi(x-\theta) + \epsilon\ \Phi(\frac{x-\theta}{\tau})\ ,$$

where ϵ is the percent contamination, and τ is the error factor. The normal distribution corresponds to a contamination $\epsilon = 0$, in which case, 68% of the measurements lie in the confidence interval, $(-\sigma, +\sigma)$. If $\epsilon > 0$, a percentage of the measurements will have a larger variance (for $\tau > 1$), and the number of measurements falling in the $(-\sigma, +\sigma)$ range becomes smaller. The heavy tailed distribution, with $\tau > 1$, can consequently be described as a normal distribution with an elongated tail. Huber [30] describes real case applications to deal with contamination levels higher than 10% ($\tau = 3$ was used), in which case, the mean estimate looses its high efficiency properties to the median estimate.

5.3.1 Equivariance and Invariance of Location Estimates

The Equivariance and Invariance properties describe the way by which an estimator responds to the shifting and scaling of the sample. An estimator, or statistic, $T_n = T(x_1, \ldots, x_n)$ is called location–and–scale equivarient *iff* [12]

$$T_n(ax_1+b, \ldots, ax_n+b) = aT_n(x_1, \ldots, x_n) + b\ .$$

For location estimates solution of Eq. (5.3), we have

$$\sum_{i=1}^{n} \frac{|ax_i + b - \theta|^{\lambda}}{(ax_i + b - \theta)} = 0 \ , \tag{5.4}$$

which yields

$$\sum_{i=1}^{n} \frac{|x_i - (\theta - b)/a|^{\lambda}}{(x_i - (\theta - b)/a)} = 0 \ .$$

Therefore, the statistic $T_n = T(x_1, \ldots, x_n)$ is given by $T_n = (\theta - b)/a$, where $\theta = T(ax_1 + b, \ldots, ax_n + b)$ [Eq. (5.4)]. It follows that

$$T(ax_1 + b, \ldots, ax_n + b) = aT_n + b = aT(x_1, \ldots, x_n) + b \ ,$$

and the solution estimates are location–and–scale equivarient, which concludes the proof. In summary, location estimates solution of Eq. (5.3) do not require solving dual problems of location and scale simultaneously, which simplifies the procedure tremendously.

5.3.2 Fisher Consistency at The Heavy Tailed Distribution

An important property to establish is the Fisher Consistency at the assumed model since the influence curve, which formalizes the resistance of the estimator to outlier measurements, is a direct consequence of Fisher Consistency [16, 30].

The estimator, $T_n = T[F_n]$, is a real valued statistic of the observations X_i, where F_n is the empirical distribution of the sample $(X_1, \cdots, X_n)$. For each sample size n, the estimator provides an estimate of location. As the sample size becomes infinite, assuming the true distribution F_θ, the empirical distribution converges to the true distribution. Fisher consistency of the estimate states that at the model

F_θ, the estimator T_n asymptotically measures θ, the true value of the location parameter.

Definition: An estimator $T[F_n] = \theta_n$ of a set of observations (X_i, *i.i.d*, of size n), where F_n is the empirical distribution, is said to be Fisher Consistent at the model distribution F_θ (F_θ is arbitrary) *iff* [30]

$$\lim_{n\to\infty} T[F_n] = T[F_\theta] = \theta \quad \text{or} \quad \int \psi(x-\theta)\, dF_\theta = 0 \; .$$

Staudte [32] formalized three conditions under which the statistic given implicitly by Eq. (5.3) is Fisher Consistent at the model it is trying to estimate:

Let $\beta(\theta) = \int \psi(x-\theta)\, dF_m(x-\theta_0)$, then

$$\beta'(\theta) = (1-\lambda) \int |x-\theta|^{\lambda-2}\, dF_m(x-\theta_0) \; .$$

Since the integrand is positive for all θ and $1-\lambda \leq 0$, then $\beta'(\theta) \leq 0$ which implies that $\beta(\theta)$ is monotonically decreasing over θ. At θ_0, we get

$$\beta(\theta_0) = \int \psi(x-\theta_0)\, dF_m(x-\theta_0) = 0 \; .$$

Therefore, $\beta(\theta_0)$ exists, which implies that $\beta(\theta)$ is continuous in the neighborhood of θ_0. This concludes the Fisher Consistency property of the M–estimate at the Heavy Tailed Distribution F_m.

5.3.3 The Influence Curve at the Heavy Tailed Distribution

The influence curve, which formalizes the bias caused by an added outlier observation, thereof measures the effects of deviant pixels on the estimate, is given by [16]

$$\mathrm{IC}(x, F_m, T) = \frac{\psi(x-\theta)}{\int \psi'(x-\theta)\; dF_m} \; . \tag{5.5}$$

Let $\mathbf{I} = \int \psi'(x-\theta)\; dF_m$. After substitution for F_m and ψ, we get

$$\begin{aligned} \frac{\sqrt{2\pi}\,\mathbf{I}}{(\lambda-1)} &= (1-\epsilon) \int_{-\infty}^{\infty} |x-\theta|^{(\lambda-2)}\; \exp\{-\frac{1}{2}(x-\theta)^2\}\; dx \\ &\quad + \frac{\epsilon}{\tau} \int_{-\infty}^{\infty} |x-\theta|^{(\lambda-2)}\; \exp\{-\frac{1}{2}(\frac{x-\theta}{\tau})^2\}\; dx \; . \end{aligned}$$

Substituting for $y = (x-\theta)$ in the first term, and $y = (x-\theta)/\tau$ in the second term, leads

$$\begin{aligned} \frac{\sqrt{2\pi}\,\mathbf{I}}{(\lambda-1)} &= 2(1-\epsilon) \int_{0}^{\infty} y^{(\lambda-2)}\; \exp\{-\frac{1}{2}\, y^2\}\; dy \\ &\quad +2\epsilon\; \tau^{(\lambda-2)} \int_{0}^{\infty} y^{(\lambda-2)}\; \exp\{-\frac{1}{2}\, y^2\}\; dy \; . \end{aligned}$$

Grouping both integrals and substituting for $t = \frac{1}{2}\, y^2$, and $dy = (2t)^{-1/2}\; dt$, the previous identity becomes

$$\mathbf{I} \;=\; \frac{2^{(\lambda-2)/2}\; (\lambda-1)}{\sqrt{\pi}} \; \Big[\, (1-\epsilon) + \epsilon\; \tau^{(\lambda-2)} \,\Big] \int_0^{\infty} t^{(\frac{\lambda-1}{2}-1)}\; \exp\{-t\}\; dt \; ,$$

Using the Gamma function, $\Gamma(x)$, yields

$$\begin{aligned} \mathbf{I} &= \frac{2^{(\lambda-2)/2}\; (\lambda-1)}{\sqrt{\pi}} \Big[(1-\epsilon) + \epsilon\; \tau^{(\lambda-2)}\Big]\, \Gamma(\frac{\lambda-1}{2}) \\ &= \frac{2^{\lambda/2}\Gamma\{(\lambda+1)/2\}}{\sqrt{\pi}} \Big[(1-\epsilon) + \epsilon\; \tau^{(\lambda-2)}\Big] \; . \end{aligned}$$

Substituting $\mathbf{I}$ in Eq. (5.5) yields the influence curve at the heavy tailed distribution

$$\mathrm{IC}(x, F_m, T) = \frac{\sqrt{\pi}}{2^{\lambda/2}\Gamma\{(\lambda+1)/2\}\,[(1-\epsilon) + \epsilon\; \tau^{(\lambda-2)}]}\; \frac{|x-\theta|^{\lambda}}{(x-\theta)} \; . \tag{5.6}$$

At the normal distribution, $\tau = 1$, the influence curve is given by

$$\mathrm{IC}(x, \Phi, T) = \frac{\sqrt{\pi}}{2^{\lambda/2}\Gamma\{(\lambda+1)/2\}} \frac{|x-\theta|^{\lambda}}{(x-\theta)} .$$

The influence curve is proportional to $\psi(x-\theta)$, which is bounded only when $\lambda = 1$. Since an M–estimator is B–robust only when $\psi(x-\theta)$ is bounded, the sample median is the only B–robust estimator. B–robustness is important especially when the added outlier measurement can carry extremely large error. In this case for the sample mean estimate where $\psi(x-\theta) = x - \theta$, this implies that a single outlier measurement has the ability to drive the estimate all over bounds (for any size sample where the observations are ideally normal). The sample mean therefore fails to provide reliable estimates in the presence of outlier observations, which is to say nearly always, knowing the different sources contributing error in range images.

However, one might argue as to what degree is B–robustness important when the sample space is finite. The sample space is bounded, since we are dealing with 8–bit images (255 maximum gray levels allowed). The error $(x-\theta)$ is consequently bounded by the interval $[-255, +255]$. Figure 5.2 shows the influence caused by an added outlier measurement for $\lambda = 1, 1.3$, and 2.0 where the error is bounded by the interval $[-255, +255]$. It is difficult to distinguish among the influence curves for the case $\lambda = 1$ and $\lambda = 1.3$; whereas the influence curve for $\lambda = 2$ is linear. Therefore, the sample mean estimate contributes the largest amount of bias, and the sample median contributes the least amount of bias to the estimate. Intermediate values of the parameter λ, such as $\lambda \leq 1.3$, yield estimators of

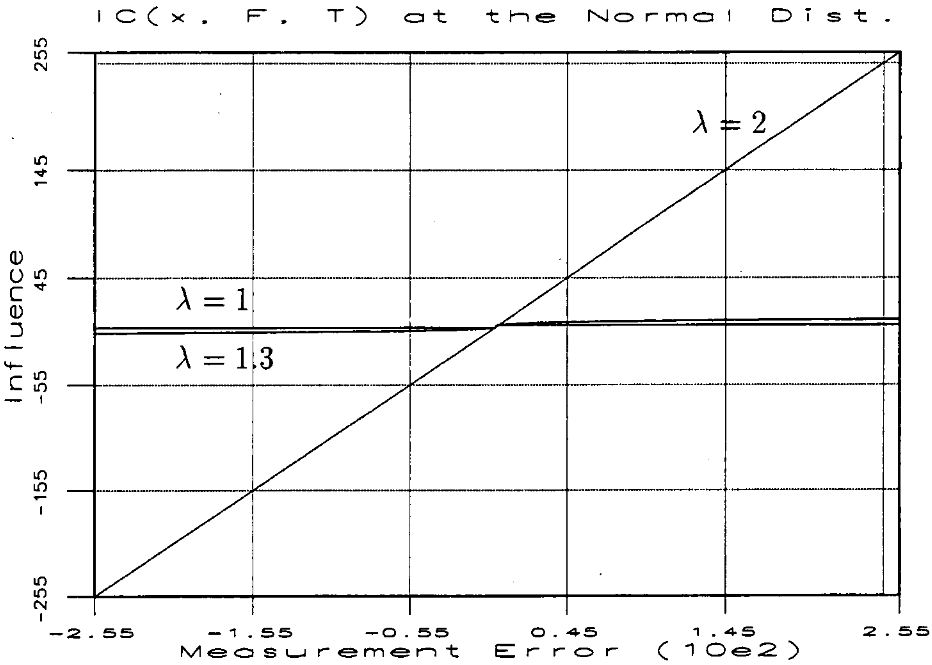


Figure 5.2: *Influence curve at the Normal Distribution for* $\lambda = 1.0, 1.3,$ *and* 2.0 *M–estimators. Note the wide difference in robustness to measurement error of the M–estimator for* $\lambda = 1.3$ *as opposed to the sample mean(*$\lambda = 2$*).*

---

location with reasonable robustness to outliers.

5.3.4 The Asymptotic Variance at the Heavy Tailed Distribution

Assuming asymptotic normality of the estimate [16], the asymptotic variance at the heavy tailed distribution F_m is given by

$$V(T, F_m) = \int \mathrm{IC}^2(x, F_m, T)\, dF_m \,. \tag{5.7}$$

Let

$$\mathbf{I} = \frac{\sqrt{\pi}}{2^{\lambda/2}\Gamma\{(\lambda+1)/2\}\left[(1-\epsilon) + \epsilon\, \tau^{(\lambda-2)}\right]}\,, \tag{5.8}$$

then, using Eq. (5.6) and (5.7) yields

$$V(T, F_m)/\mathbf{I}^2 = \int_{-\infty}^{\infty} |x-\theta|^{(2\lambda-2)} \left[(1-\epsilon)\ d\Phi(x-\theta) + \epsilon\ d\Phi(\frac{x-\theta}{\tau})\right] .$$

Substituting for Φ and splitting the integrals (for $\sigma = 1$), leads

$$\begin{aligned} \sqrt{2\pi}\ V(T, F_m)/\mathbf{I}^2 &= (1-\epsilon) \int_{-\infty}^{\infty} |x-\theta|^{(2\lambda-2)} \exp\{-\frac{1}{2}(x-\theta)^2\}\ dx \\ &\quad + \frac{\epsilon}{\tau} \int_{-\infty}^{\infty} |x-\theta|^{(2\lambda-2)} \exp\{-\frac{1}{2}(\frac{x-\theta}{\tau})^2\}\ dx . \end{aligned}$$

Let $y = x - \theta$ in the first term, and $y = (x-\theta)/\tau$ in the second term, the previous identity in terms of the new variable y leads

$$\begin{aligned} \sqrt{2\pi}\ V(T, F_m)/\mathbf{I}^2 &= 2(1-\epsilon) \int_{0}^{\infty} y^{(2\lambda-2)} \exp\{-\frac{1}{2}\ y^2\}\ dy \\ &\quad + 2\epsilon\ \tau^{(2\lambda-2)} \int_{0}^{\infty} y^{(2\lambda-2)} \exp\{-\frac{1}{2}\ y^2\}\ dy . \end{aligned}$$

Grouping the integrals and substituting for $t = \frac{1}{2}\ y^2$, and $dy = (2t)^{-1/2}\ dt$, the previous identity in terms of the new variable t leads

$$V(T, F_m)/\mathbf{I}^2 = \frac{2^{(\lambda-1)}}{\sqrt{\pi}} \left[\ (1-\epsilon) + \epsilon\ \tau^{(2\lambda-2)}\ \right] \int_{0}^{\infty} t^{(\lambda-1/2)-1} \exp\{-t\}\ dt .$$

Using the Gamma function, $\Gamma(x)$, leads

$$V(T, F_m)/\mathbf{I}^2 = \frac{2^{(\lambda-1)}}{\sqrt{\pi}} [(1-\epsilon) + \epsilon\ \tau^{(2\lambda-2)}]\ \Gamma(\lambda - 1/2) .$$

Substituting for $\mathbf{I}$, Eq. (5.8), yields the asymptotic variance at the heavy tailed distribution

$$V(T, F_m) = \frac{\sqrt{\pi}}{2}\ \frac{\Gamma(\lambda - 1/2)}{\Gamma^2(\frac{\lambda+1}{2})}\ \frac{(1-\epsilon) + \epsilon\ \tau^{2\lambda-2}}{[(1-\epsilon) + \epsilon\ \tau^{\lambda-2}]^2} . \tag{5.9}$$

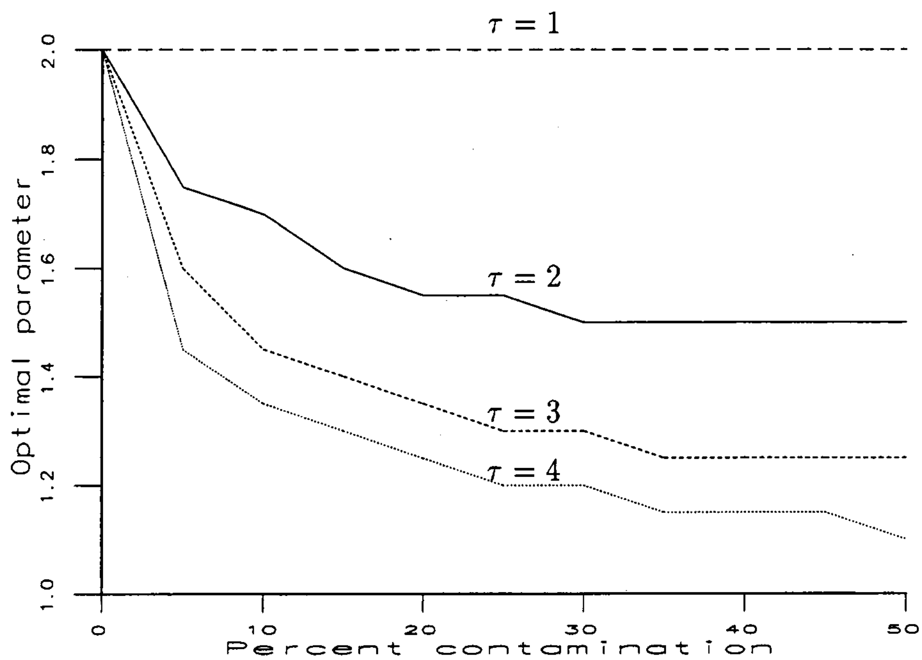


Figure 5.3: *Minimum variance estimate vs. contamination level ϵ for $\tau = 1, 2, 3$, and 4.*

If we fix τ, we can analyze the effects of ϵ and λ on the variance of the estimate. The results are shown in Fig. 5.4.

For $\tau = 1$, the heavy tailed distribution is the normal distribution for any level of the contamination, ϵ, since both probability distributions of the mixture distribution have identical variance. Figure 5.4(a) shows that the mean estimate, $\lambda = 2$, is a minimum variance solution when the error distribution is normal. On the other hand, the median estimate, $\lambda = 1$, has the largest asymptotic variance against the normal distribution. This indicates the sensitivity of the sample median and the immunity of the sample mean to round–off errors (which can be modeled by the normal distribution).

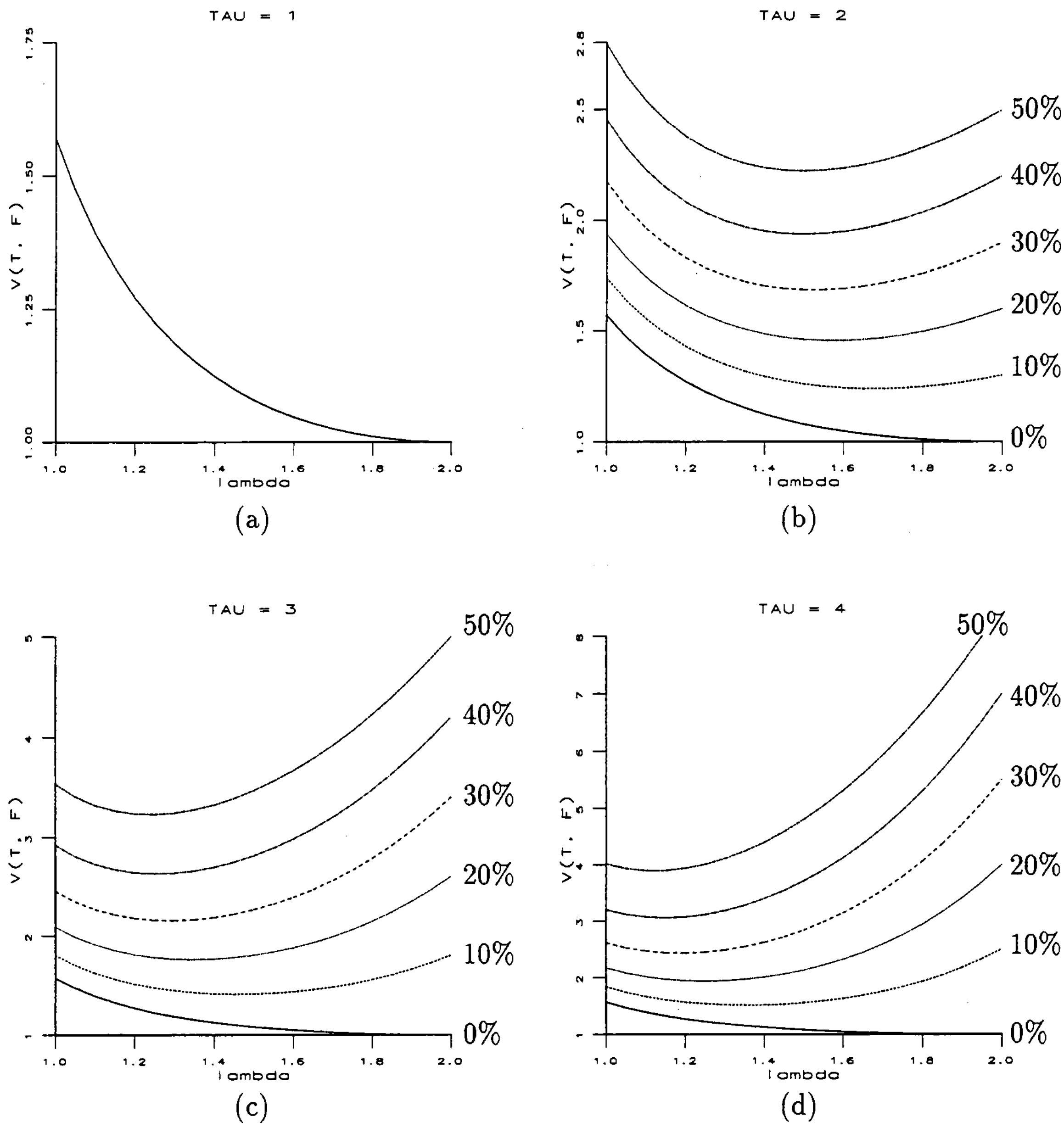


Figure 5.4: *Effects of Contamination ϵ on $V(T,F)$; (a) $\tau = 1$ for all ϵ. (b) $\tau = 2$ and ϵ incremented by 10% from bottom to top. (c) $\tau = 3$ and ϵ incremented by 10% from bottom to top. (d) $\tau = 4$ and ϵ incremented by 10% from bottom to top.*

For $\tau = 2$, Fig. 5.4(b), the mean estimate maintains its efficiency properties against the median estimate at all levels of the contamination, ϵ. Figure 5.4(b) also shows that for very small deviations from the normal distribution, $\tau = 2$, the variance of the estimate is optimum (smallest) for intermediate values of the parameter λ. The minimum variance solution depends on the level of the contamination, ϵ [Fig. 5.3]. The parameter λ that corresponds to the minimum variance solution is monotonically decreasing as a function of ϵ. This indicates the existence of favorable efficiency properties of estimators intermediate to the sample mean and the sample median estimates. These favorable properties become more acute as we increase the magnitude of the error, τ, in the contaminated samples.

For $\tau = 3$, Fig. 5.4(c), if 10% of the observations are contaminated, the mean estimate loses its efficiency properties to the median estimate [23, 30]. For $\tau = 4$, Fig. 5.4(d), contamination levels as low as 5% can offset the mean, and the median becomes a smaller variance estimate. Even for these high error levels, neither the mean nor the median are minimum variance solutions [Figs. 5.4(c) and (d)]. The minimum variance estimate is again a function of the contamination, ϵ [Fig. 5.3].

5.3.5 The Asymptotic Relative Efficiency

The Asymptotic Relative Efficiency (ARE) is used to compare two statistics $T_1[F]$ and $T_2[F]$, and to provide a qualitative evaluation of both statistics. It is defined as the ratio of the asymptotic variances against the underlying F [30]:

$$\mathrm{ARE}(T_1, T_2) = \frac{V(T_2, F)}{V(T_1, F)} \, . \tag{5.10}$$

Choosing $\tau = 1$ means that the observations are normally distributed. Fig-

Table 5.1: *Asymptotic Relative Efficiency at the Normal Distribution.*

λ	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
$ARE(T_\lambda, T_2)$	63.7	71.8	78.7	84.4	89.0	92.7	95.5	97.6	99.0	99.8	100.0

ure 5.4(a) is a measure of the asymptotic variance of the estimate for an arbitrary λ. Let T_1 be the solution estimate for an arbitrary λ, $T_1 = T_\lambda[F]$, and $T_2 = T_2[F]$ be the sample mean estimate ($\lambda = 2$). Consequently, the condition ARE$(T_\lambda, T_2) \leq 1$ indicates that $T_\lambda[F]$ is less efficient than the mean estimate due to its larger asymptotic variance and visa-versa [23]. Since in this case ($\tau = 1$) $T_2[F]$ corresponds to the minimum variance estimate, the ARE becomes a percentage that grades the efficiency of the estimate for an arbitrary λ relative to the sample mean estimate.

Table 5.1 shows that the ARE at the normal distribution is monotonically increasing with respect to λ. For $\lambda = 1.3$, the estimate reaches 84% the efficiency of the sample mean, which is a reasonable efficiency at the normal distribution. The sample median, however, has a relative efficiency of 63% which is quite inefficient when the observations are normally distributed.

Figure 5.4(d), described earlier, is a measure of the asymptotic variance of the estimate for $\tau = 4$ for various contamination levels, ϵ. The chosen values of τ and ϵ specify the exact model distribution. We use this information to construct a table of ARE values for each level of the contamination, ϵ. Table (5.2) shows that 10% contamination increases the efficiency of the sample median relative to

Table 5.2: *The Asymptotic Relative Efficiency,* ARE(T_λ, T_2), *at the Heavy Tailed Distribution* ($\tau = 4$, *and* ϵ *ranging from 10% to 50%)*

λ	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
$\epsilon = 10\%$	1.36	1.50	1.59	1.64	1.65	1.61	1.53	1.43	1.30	1.15	1.0
$\epsilon = 20\%$	1.84	1.98	2.06	2.05	1.99	1.88	1.72	1.55	1.36	1.17	1.0
$\epsilon = 30\%$	2.10	2.23	2.26	2.21	2.10	1.94	1.75	1.55	1.35	1.17	1.0
$\epsilon = 40\%$	2.18	2.28	2.27	2.20	2.06	1.89	1.70	1.51	1.32	1.15	1.0
$\epsilon = 50\%$	2.11	2.18	2.16	2.07	1.93	1.77	1.60	1.43	1.28	1.13	1.0

the sample mean from 63% to 136%, which indicates the poor performance of the sample mean with respect to small departures from the normal distribution. However, at $\epsilon = 10\%$, the ARE at $\lambda = 1.3$ is 164%, showing a significantly higher efficiency relative to the sample median. The most efficient estimator at the 10% contamination level roughly corresponds to $\lambda = 1.35$ [Fig. (5.3)]. This estimator ($\lambda = 1.35$) presents very small gains in terms of efficiency. In addition, it introduces additional bias to the estimate, since it is naturally closer to the sample mean estimate (the estimator that contributes maximum bias to the estimate). Note that the ARE for $\lambda = 1.2$ is 160% at the 10% contamination level. This relative efficiency (160% for $\lambda = 1.2$) is comparable to that for $\lambda = 1.3$. Naturally, this estimator introduces less bias to the estimate (as opposed to $\lambda = 1.3$), since it approaches the sample median estimate (the estimator which contributes the least amount of bias to the estimate). At the normal distribution, the ARE at $\lambda = 1.2$ is 79%, showing a reasonable efficiency at the normal distribution [Table 5.1]. As ϵ increases, the estimates in the range 1.2 and 1.3 for λ maintain a high efficiency relative to the sample mean, and also relative to the sample median. They represent very efficient estimates of location that can outperform the efficiency of the median estimate at all contamination levels (which includes the normal distribution where the median estimate is quite inefficient). We can conclude that for certain intermediate values of the paramater (λ in the range $[1.2, 1.3]$), the location estimates have robustness of efficiency properties.

There is always a compromise to be made between robustness and efficiency. For example, Huber minimax estimator of location introduces a robust control parameter that has direct effects on robustness and efficiency [13, 30]. Our objectives can be summarized as follows: Maximize the efficiency against all possible deviations from the normal, and minimize with respect to the bias. It is evident that for λ in the range $[1.2, 1.3]$, we achieve robustness of efficiency and a reasonably small bias. These estimators are more reliable than the sample mean and the sample median estimates of location.

5.4 Numerical Solutions

In this section, numerical solutions are presented to the location estimate problem. In addition, the extension to multi–parameter (regression) problems is also presented, which is the primary motivation behind this study because of its direct application to the segmentation algorithm described in Chapter 4.

5.4.1 Numerical Solutions of Location Estimates

We have, so far, identified estimators intermediate to the sample mean and the sample median estimates of location, and determined their relative robustness and efficiency properties. We have also identified reliable estimates obtained through a proper assignment of the parameter λ. The objective is to provide numerical solutions for the location parameter problem. The solution estimate satisfies the following identity [Eq. (5.3)]:

$$\sum_{i=1}^{n} \psi(x_i - \theta) = \sum_{i=1}^{n} \frac{|x_i - \theta|^{\lambda}}{x_i - \theta} = 0 \ .$$

After Taylor series expansion of $\psi(x_i - \theta)$ about an arbitrary estimate $\tilde{\theta}$, we obtain the following recursive solution:

$$\theta = \tilde{\theta} + \frac{\sum_i \left|x_i - \tilde{\theta}\right|^{\lambda} / (x_i - \tilde{\theta})}{(\lambda - 1) \sum_i \left|x_i - \tilde{\theta}\right|^{\lambda-2}} \ .$$

The previous equation is similar to a gradient descent technique for solving non-linear systems of equations. The parameter $(\lambda - 1)$ controls the size of the increment for each iteration. This parameter is set to a fixed value of 1 (for legitimate reasons which are explained later). After assignment, the solution estimate is determined by

$$\theta = \tilde{\theta} + \frac{\sum_i \left|x_i - \tilde{\theta}\right|^{\lambda} / (x_i - \tilde{\theta})}{\sum_i \left|x_i - \tilde{\theta}\right|^{\lambda-2}} = \frac{\sum_i \left|x_i - \tilde{\theta}\right|^{\lambda-2} x_i}{\sum_i \left|x_i - \tilde{\theta}\right|^{\lambda-2}} \ . \qquad (5.11)$$

The solution estimate is a weighted average where the weights are relative to the residual of each observation. Only one power term needs to be computed for each residual magnitude $|x_i - \theta|$, which offers computational advantages. The form of the solution is also equivalent to the W–estimator solution [16], which confirms the chosen assignment of $\lambda - 1$. This assignment will also be encountered in the numerical solution to the multi–parameter regression problem. In real applications, such as the smoothing of range images, a maximum of 2 iterations is implemented starting with the median estimate as an initial value estimate. The stopping criteria used is $|\theta - \tilde{\theta}| \leq 0.0005$.

5.4.2 Extension to Non–linear Regression Problems

As was earlier indicated, the extension of single–parameter estimates to multi–parameter has its direct application in region growing segmentation algorithms. The general regression problem is to solve the n unknowns $(\theta_1, \cdots, \theta_n)$ constrained by the cost function $\rho(z_k, \theta_1, \cdots, \theta_n)$, where z_k are the discrete observations. The M–estimate described earlier suggests a particular cost function, ρ. We will be using this cost function to identify polynomial coefficients, a_{ij}, such that

$$\sum_k \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots) = \sum_k \left| z_k - \sum_{i+j<=n} a_{ij} u^i v^j \right|^\lambda \rightarrow min! \; , \qquad (5.12)$$

where the variables (u, v) represent the spatial coordinates of the discrete measurements z_k in the range image. The order of the fitted surface is defined by the parameter n, where $i + j \leq n$. As a simple example, fitting the discrete measurements to a planar surface ($n = 1$) is equivalent to solving for the three unknown coefficients a_{00}, a_{01} and a_{10}. The planar surface which best represents the discrete measurements in the sense of the chosen cost function, ρ, is given by $P(u,v) = a_{00} + a_{10}u + a_{01}v$. One important observation regarding the arbitrary parameter $\lambda \in [1,2]$ is that the two special cases for $\lambda = 1$ and $\lambda = 2$

correspond to two well known cost functions. For $\lambda = 1$, we obtain a Chebychev approximation of the discrete measurements [18]. For $\lambda = 2$, we obtain a Least Squares approximation of the discrete measurements. The analysis of this cost function for an arbitrary λ will be experimentally evaluated against two commonly used approaches. The theoretical framework established for the proper selection of the arbitrary parameter λ is also applicable in this case. The solution is obtained in a similar fashion to location estimates. The first step is to expand $\rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)$ with respect to an arbitrary estimate $(\tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots)$. The Taylor series expansion is given by

$$\begin{aligned}
&\rho(z_k, a_{00}, \cdots, a_{lm}, \cdots) = \rho(z_k, \tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots) \;\; + \\
&\sum_{i+j \leq n} \frac{\partial \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{ij}} \Big|_{(\tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots)} (a_{ij} - \tilde{a}_{ij}) \;\; + \\
&\frac{1}{2} \sum_{i'+j' \leq n} \; \sum_{i+j \leq n} \frac{\partial^2 \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{ij} \partial a_{i'j'}} \Big|_{(\tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots)} (a_{ij} - \tilde{a}_{ij})(a_{i'j'} - \tilde{a}_{i'j'}) \; + \; R \; .
\end{aligned}$$

After substitution for ρ in Eq. (5.12) and differentiating with respect to a_{lm} (while ignoring the reminder term R), the non–linear regression problem becomes solution of

$$\begin{aligned}
&\sum_k \frac{\partial \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{lm}} = \sum_k \frac{\partial \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{lm}} \Big|_{(\tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots)} + \\
&\sum_k \sum_{i+j \leq n} \frac{\partial^2 \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{ij} \partial a_{lm}} \Big|_{(\tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots)} (a_{ij} - \tilde{a}_{ij}) = 0 \;, \quad \forall \, l + m \leq n \; .
\end{aligned}$$

Re–ordering the second term of the summation yields

$$\begin{aligned}
&\sum_k \frac{\partial \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{lm}} = \sum_k \frac{\partial \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{lm}} \Big|_{(\tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots)} + \\
&\sum_{i+j \leq n} (a_{ij} - \tilde{a}_{ij}) \sum_k \frac{\partial^2 \rho(z_k, a_{00}, \cdots, a_{lm}, \cdots)}{\partial a_{ij} \partial a_{lm}} \Big|_{(\tilde{a}_{00}, \cdots, \tilde{a}_{lm}, \cdots)} = 0 \;, \quad \forall \, l + m \leq n \; .
\end{aligned}$$

After substitution for the partials in the previous equation,

$$\frac{\partial \rho}{\partial a_{lm}} = -\lambda\, u^l\, v^m \, \frac{\left| z_k - \sum\limits_{i'+j'\leq n} a_{i'j'} u^{i'}\, v^{j'} \right|^{\lambda}}{z_k - \sum\limits_{i'+j'\leq n} a_{i'j'} u^{i'}\, v^{j'}} ,$$

$$\frac{\partial^2 \rho}{\partial a_{ij} \partial a_{lm}} = (\lambda^2 - \lambda) u^{i+l}\, v^{j+m} \left| z_k - \sum_{i'+j'\leq n} a_{i'j'} u^{i'}\, v^{j'} \right|^{\lambda-2} ,$$

the system of equations becomes solution of

$$-\lambda \sum_k u^l\, v^m \, \frac{\left| z_k - \sum\limits_{i'+j'\leq n} \tilde{a}_{i'j'} u^{i'}\, v^{j'} \right|^{\lambda}}{z_k - \sum\limits_{i'+j'\leq n} \tilde{a}_{i'j'} u^{i'}\, v^{j'}} +$$

$$(\lambda^2 - \lambda) \sum_{i+j\leq n} (a_{ij} - \tilde{a}_{ij}) \sum_k u^{i+l}\, v^{j+m} \left| z_k - \sum_{i'+j'\leq n} \tilde{a}_{i'j'} u^{i'}\, v^{j'} \right|^{\lambda-2} = 0 .$$

The latter is an $N \times N$ linear system of equations (where $N = (n+2)(n+1)/2$, and n is the order of the polynomial used) which can be solved using the standard Gaussian elimination method. Pivoting techniques [18] will also be used to avoid singularity problems. The parameter $(\lambda - 1)$ is assigned the value 1, similar to the location problem. Therefore, the $N \times N$ system of linear equations used, in our experimentation, is given by:

$$\sum_{i+j\leq n} (a_{ij} - \tilde{a}_{ij}) \sum_k u^{i+l}\, v^{j+m} \left| z_k - \sum_{i'+j'\leq n} \tilde{a}_{i'j'} u^{i'}\, v^{j'} \right|^{\lambda-2}$$

$$= \sum_k u^l\, v^m \, \frac{\left| z_k - \sum\limits_{i'+j'\leq n} \tilde{a}_{i'j'} u^{i'}\, v^{j'} \right|^{\lambda}}{z_k - \sum\limits_{i'+j'\leq n} \tilde{a}_{i'j'} u^{i'}\, v^{j'}} , \quad l + m \leq n.$$

In this experiment, the initial estimates are set to zero (for all coefficients). If we provide a better set of initial parameters, as we have done for the location estimate, this additional step would help speed–up convergence of the algorithm. We chose this initial set in order to prove the convergence of the numerical solution without any additional guidance. Given the initial set of parameters, the algorithm proceeds to evaluate an updated set of parameters which would serve as initial estimates for the next iteration. In order to simplify notation, we will discuss this two step process in detail and define the stopping criteria used to terminate the recursion.

Once we are given the set of coefficients $\tilde{a}_{ij}$, the entire region is evaluated with respect to those coefficients, and the approximation $\tilde{z}_k$ of z_k at the spatial coordinates (u, v) is given by

$$\tilde{z}_k = \sum_{i'+j'\leq n} \tilde{a}_{i'j'} u^{i'} \; v^{j'} \; .$$

Substituting for $\tilde{z}_k$ leads

$$\sum_{i+j\leq n} (a_{ij} - \tilde{a}_{ij}) \sum_k u^{i+l} \; v^{j+m} \; |z_k - \tilde{z}_k|^{\lambda-2} = \sum_k \; u^l \; v^m \; \frac{|z_k - \tilde{z}_k|^{\lambda}}{z_k - \tilde{z}_k}, \qquad l + m \leq n \; .$$

The solution can be rewritten in matrix form: $\mathbf{F}\mathbf{x} = \mathbf{G}$, where

$$\mathbf{F} \;=\; \begin{pmatrix} \sum_k |z_k - \tilde{z}_k|^{\lambda-2} & \sum_k u \; |z_k - \tilde{z}_k|^{\lambda-2} & \sum_k v \; |z_k - \tilde{z}_k|^{\lambda-2} & \cdots \\ \sum_k u \; |z_k - \tilde{z}_k|^{\lambda-2} & \sum_k u^2 \; |z_k - \tilde{z}_k|^{\lambda-2} & \sum_k u \; v \; |z_k - \tilde{z}_k|^{\lambda-2} & \cdots \\ \sum_k v \; |z_k - \tilde{z}_k|^{\lambda-2} & \sum_k u \; v \; |z_k - \tilde{z}_k|^{\lambda-2} & \sum_k v^2 \; |z_k - \tilde{z}_k|^{\lambda-2} & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix},$$

$$
\mathbf{x} = \begin{pmatrix} a_{00} - \tilde{a}_{00} \\ a_{10} - \tilde{a}_{10} \\ a_{01} - \tilde{a}_{01} \\ \vdots \end{pmatrix}, \qquad \text{and}
$$

$$
\mathbf{G} = \begin{pmatrix} \sum_{z_k \geq \tilde{z}_k} (z_k - \tilde{z}_k)^{\lambda-1} - \sum_{z_k \leq \tilde{z}_k} (\tilde{z}_k - z_k)^{\lambda-1} \\ \sum_{z_k \geq \tilde{z}_k} u\,(z_k - \tilde{z}_k)^{\lambda-1} - \sum_{z_k \leq \tilde{z}_k} u\,(\tilde{z}_k - z_k)^{\lambda-1} \\ \sum_{z_k \geq \tilde{z}_k} v\,(z_k - \tilde{z}_k)^{\lambda-1} - \sum_{z_k \leq \tilde{z}_k} v\,(\tilde{z}_k - z_k)^{\lambda-1} \\ \vdots \end{pmatrix}.
$$

The final solution coefficients are determined by

$$
\begin{pmatrix} a_{00} \\ a_{10} \\ a_{01} \\ \vdots \end{pmatrix} = \begin{pmatrix} \tilde{a}_{00} \\ \tilde{a}_{10} \\ \tilde{a}_{01} \\ \vdots \end{pmatrix} + \mathbf{F}^{-1}\,\mathbf{G}\,.
$$

The inversion of the matrix **F** was achieved using the standard Gaussian elimination. As mentioned previously, pivoting techniques where also used in order to prevent singularity problems of the matrix **F**. This procedure is performed recursively with a maximum of 10 iterations. The process is terminated when $\|\, \mathbf{x} \,\| \leq 0.005$.

5.5 Experimental Evaluation

5.5.1 The Need for Robust Location Estimates

The location parameter is computed for $\lambda = 1, 1.1, 1.2, \ldots, 2.0$ using the Cushny and Peebles data set [16]: $\{0, 0.8, 1.0, 1.2, 1.3, 1.3, 1.4, 1.8, 2.4, 4.6\}$. The observations corresponding to this sample, representing the effects of two drugs on the elongation of sleep performed on 10 individuals, were originally used by others as an illustration of a normally distributed sample. Hampel [16] after examining the sample carefully, noticed that the 4.6 measurement was deviant from all other measurements, and therefore represents an outlier measurement. With reference to Hampel, a robust estimate of location should be element of the interval [1.24, 1.4]. As an illustration of this, he saw that the 10%–trimmed mean provides a location estimate of 1.4, the 20%–trimmed mean leads 1.33, the 50%–trimmed mean (i.e., the sample median) leads 1.3. If the 4.6 observation was removed, by means of some outlier rejection procedure, the sample mean estimate becomes 1.24. Without performing outlier rejection, the sample mean estimate is 1.58. This value of the sample mean is clearly deviant from all other estimates as a result of the presence of an outlier measurement in the Cushny and Peebles data set. The results of this experiment applied to our M–estimator at those specified values of the parameter λ are shown in Table (5.3).

Since According to Hampel, robust estimates of location are constrained by the interval [1.24, 1.4], and with reference to the results found [see Table (5.3)], the estimators are robust with respect to this error magnitude of the outlier ob-

Table 5.3: *Experimentation with Cushny and Peebles Data Set.*

λ	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
T_{peeble}	1.300	1.300	1.302	1.320	1.352	1.385	1.412	1.454	1.497	1.539	1.580

Table 5.4: *Experimentation with Cushny and Peebles Data Set for an Outlier Observation of 254.*

λ	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
T_{peeble}	1.300	1.308	1.400	1.793	2.440	4.356	7.584	11.747	16.484	21.484	26.520

servation when the parameter $\lambda \leq 1.5$. We propose to run the same experiment after modifying the magnitude of the outlier observation to 254. The results of the conducted experiment are shown in Table (5.4). From the results shown, we conclude that location estimates for $\lambda \leq 1.2$ are robust with respect to the maximum allowed infinitesimal deviation from the normal, since all other estimates ($\lambda \geq 1.2$) resulted in unreliable estimates of location (not bound by the interval [1.24, 1.4]). Note also the breakdown of the sample mean, and the capability of the sample median estimate in providing the same location estimate with respect to this larger error. This is a simple illustration investigating the robustness of the M–estimator in the presence of outlier observations, and showing the need for robust statistical procedures to achieve reliable estimates of location. These experiments clearly show that the computed estimates for intermediate values of the parameter λ are in fact intermediate (linearly increasing with respect to the parameter used) to the sample mean and sample median estimates of location.

5.5.2 Experimental Evaluation of Non–linear Regression Estimates

Our main motivation behind the development of a robust M–estimator is its direct application to multi–parameter problems. In this experiment, we use an ideal synthetic range image, $f_i(u,v)$, to which we add contaminated noise of various magnitudes. The contaminated surface is fitted to a 4th order polynomial, and an output image, $\tilde{f}(u,v)$, is generated through evaluation of this smooth function at each pixel location of the fitted region. The surface fitting is done using various λ–estimators, and the regression coefficients of the fitting polynomial surface are determined for $\lambda \in [1,2]$. A goodness of fit measure P_f, namely the absolute deviation error norm, is used to provide us with comparisons:

$$P_f = \frac{1}{N} \sum_{(u,v)\in R} \left| f_i(u,v) - \tilde{f}(u,v) \right| \,, \tag{5.13}$$

where R represents the set of all pixels element of the fitted region, and N is the total number of pixels in R.

Spherical shapes have practical significance in real applications. Figure 5.5 represents samples of contaminated images used in the testing of the algorithm. The Heavy Tailed Distribution may not be the exact representative of the noise sources present in range scanners, as one might argue. However, if we take a quick scan over the images used for testing, we realize that an extremely high level of noise was appended to the synthetic data. The noise that was added follows a Heavy Tailed distribution. Figure 5.5 corresponds to a standard deviation $\sigma = 20$, an error factor $\tau = 4$, and a contamination $\epsilon = 10, 20, 30, 40$ and 50%. The ideal

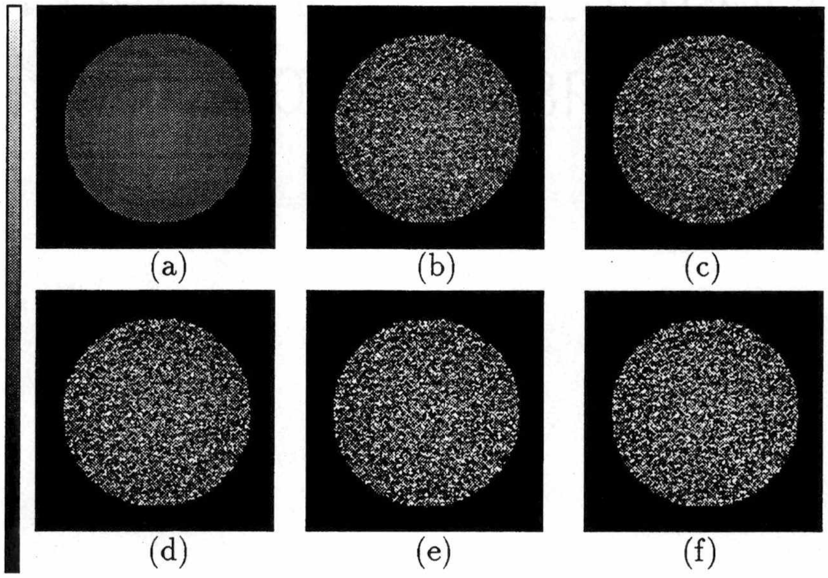


Figure 5.5: *Contaminated images ($\sigma = 20$, and $\tau = 4$) used of a sphere for surface fitting: (a) ideal spherical region (b) $\epsilon = 10\%$ applied to (a). (c) $\epsilon = 20\%$ contamination applied to (a). (d) $\epsilon = 30\%$ contamination applied to (a). (e) $\epsilon = 40\%$ contamination applied to (a). (f) $\epsilon = 50\%$ contamination applied to (a).*

---

image is shown in Fig. 5.5(a). Through simple visual inspection and the results of the conducted experiments with real range scanners, range images were never shown to reach the contamination best illustrated with the aid of Fig. 5.5(f). Therefore, if the algorithm is robust with respect to this level of contamination, it should perform in a similar fashion in real applications.

Note that the added contamination can cause a severe damage to the original image. This is illustrated with Fig. 5.5(f) where the contamination is clearly visible with the drastic changes made to the pixel values of the original image.

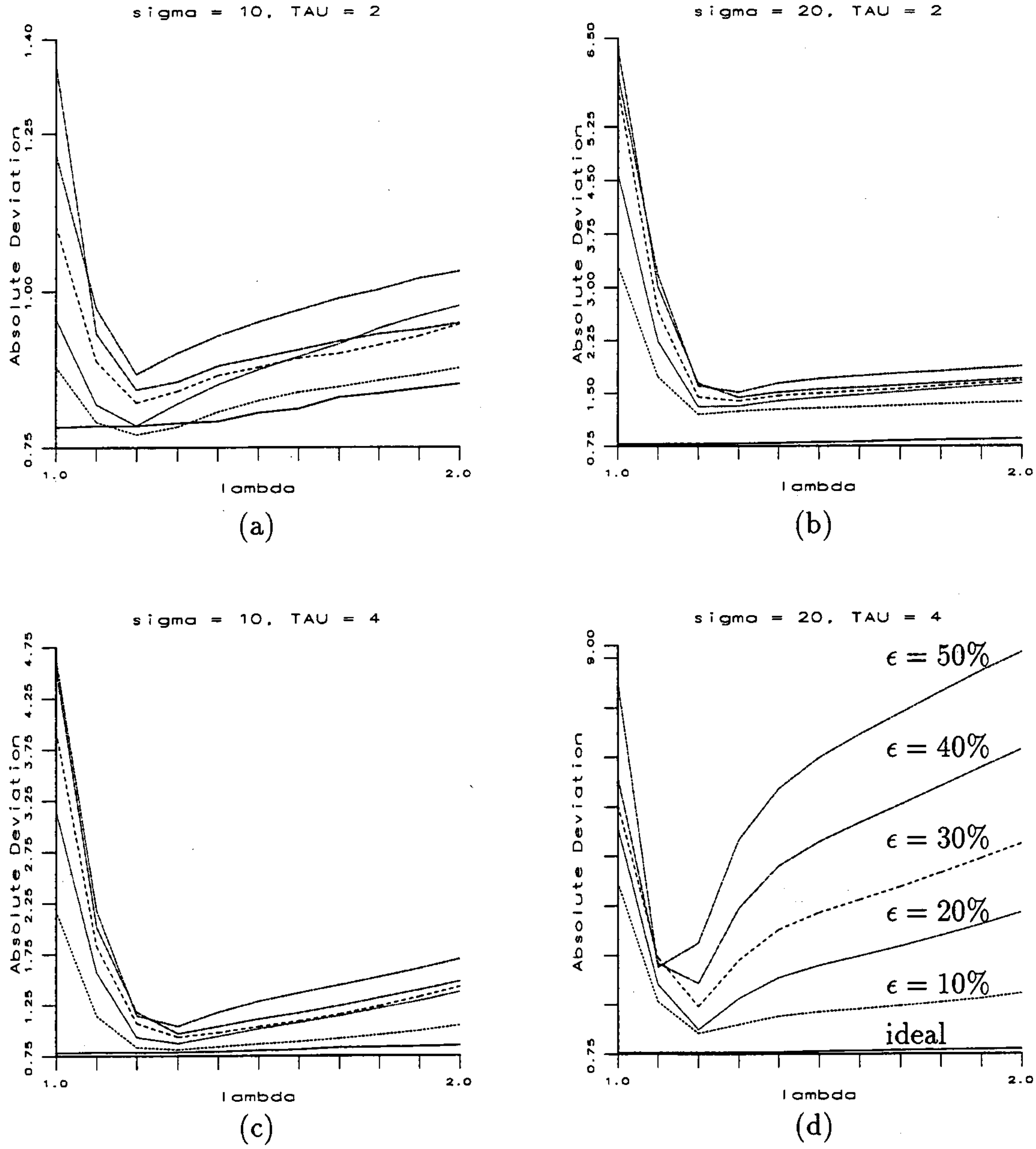


Figure 5.6: *Effects of increasing the amount of contamination* ϵ, *and the standard deviation* σ *on the absolute deviation error norm* P_f *for the synthetic spherical image: (a)* $\tau = 2$, $\sigma = 10$ *and* $\epsilon = 10, 20, 30, 40$ *and* 50%. *(b)* $\tau = 2$, $\sigma = 20$ *and* $\epsilon = 10, 20, 30, 40$ *and* 50%. *(c)* $\tau = 4$, $\sigma = 10$ *and* $\epsilon = 10, 20, 30, 40$ *and* 50%. *(d)* $\tau = 4$, $\sigma = 20$ *and* $\epsilon = 10, 20, 30, 40$ *and* 50%.

The results of fitting the set of images in Fig. 5.5 are those shown in Fig. 5.6(d). Note that the absolute deviation error norm found is consistently lower when the regression parameter λ is near 1.2. The error found for this particular value of λ is also very close to the ideal case (fitting the noise free synthetic image to a 4th order polynomial, which corresponds to the curve found nearest to the λ–axis). The generated results show that the estimator corresponding to this value of the parameter ($\lambda = 1.2$) results in a consistent minimal absolute deviation error norm for all levels of contamination.

5.6 Effects of Regression Parameter on Segmentation

In this section, we will apply our Edge– and Region–based segmentation algorithm presented in Chapter 4 in order to analyze and illustrate the effects of this regression algorithm on final segmentation. The region splitting and growing modules which relied on Equally Weighted Least Squares will now be using the developed surface fitting technique with the arbitrary parameter λ. The comparisons in terms of segmentation results will be established using the real image SHAPES shown in Fig. 4.7(a).

In the first experiment, we provide the same set of parameters to all segmentation modules except for the regression parameter λ which is allowed to vary. The results are shown in Fig. 5.7 which indicates a superior result for intermediate values of the regression parameter. Note for instance that for $\lambda = 1$, the cylinder has grown beyond its actual boundaries. The primary problem with Chebychev

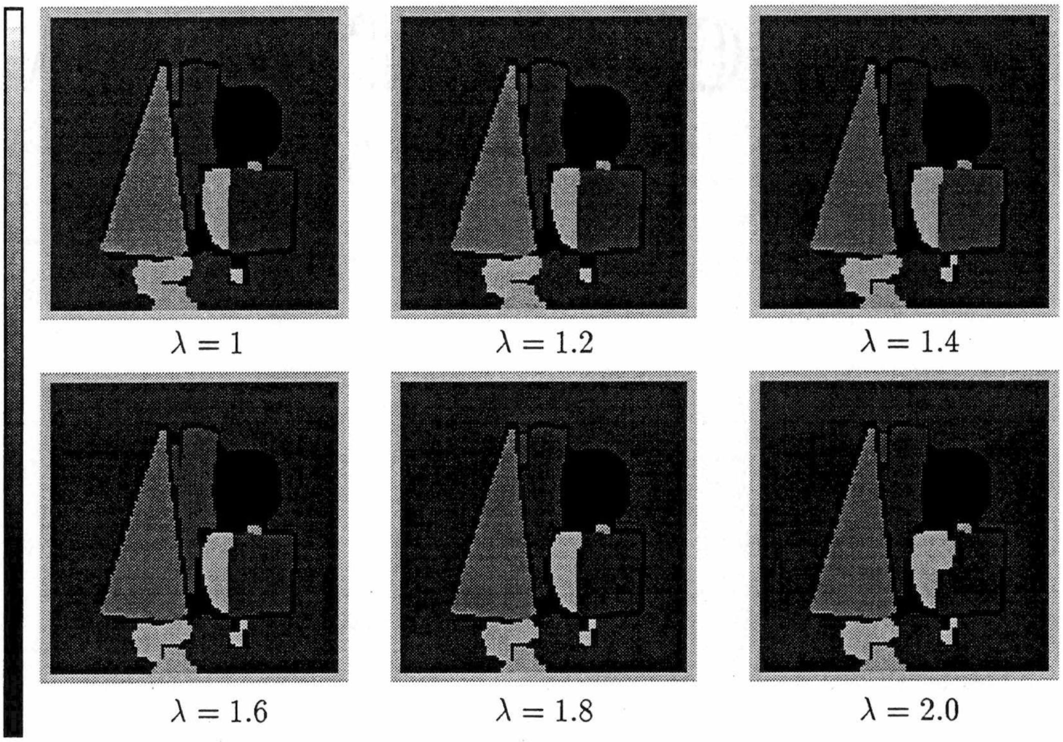


Figure 5.7: *Effects of λ on segmentation of image* SHAPES.

approximation lies in its slow convergence rate. For larger values of λ, the results show a more refined cylinder, which indicates a better polynomial representation of the discrete sample within the cylindrical region. Figure 5.7 also shows problems within the left side of the box. This region is progressively growing on the count of an existing flat surface. For $\lambda = 2$, the results are more severe in this case. The region growing constant is set to 3.0 in this experiement. In the next experiment, we allow this constant to vary from low to high in order to analyse its effects on the final result. However, a conclusion can be made as to the previously indicated problem relating to the box region:

From the conducted study, we found that a lower Absolute Deviation Error Norm results from the use of intermediate values of λ (other than 2.0 which is the Least Squares method). The problem is that the value of the growing constant, $\eta = 3$, was not optimal for the least squares method to yield accurate results. In fact, this value of η is too low for the least squares method ($\eta = 4$ provides better results). When this happens, the left side of the box takes a second order polynomial representation instead of a planar surface. This results in a lower error, and the region is therefore allowed to grow considerably. The problem with this fitting algorithm lie is the fact that the left side of the box has occluded an existing independent flat region. In addition, the polynomial representation which is a plane in reality has taken a higher order polynomial representation. This may be misleading when we attempt to provide an interpretation for this region.

Figure 5.8 illustrates some of the problems of the least squares technique in terms of its effects on final segmentation. Note the inaccuracies in final segmentation caused by the least squares technique for all levels of η (the region growing constant). Note for ($\lambda = 2.0$, $\eta = 2$) parameters that the box region is fragmented. This is due to the inadequate representation of this region by the least squares method. Note also that for lower values of λ, we get a more accurate surface representation of the box region. For higher values of η, $\eta = 3$, we note that the least squares method still yields inaccurate segmentation. The results of using lower values of λ illustrate the advantages of using intermediate values of λ over the least squares method. The differences among $\lambda = 1.2$ and $\lambda = 1$ are not as severe. However, the convergence rate is much faster for $\lambda = 1.2$. Also, there are some small differences in the cylindrical region among the ($\lambda = 1, \eta = 3$) and ($\lambda = 1.2, \eta = 3.0$) final results, which was noted and commented upon in the first experiment. The same remark holds for the least squares technique with regard to the cylindrical region. This experiment also shows the higher flexibility of the segmentation algorithm in terms of choosing a suitable set of parameters such as the value of η.

5.7 Summary

In this chapter, we have shown the importance of dealing effectively with outlier observations in order to achieve reliable location estimates. This chapter also shows the effect of the M–estimate in providing reliable regression polynomial coefficients from contaminated synthetic data. This algorithm also provides descriptive information about homogeneous regions in range images. A highly recommended λ–parameter, for regression and smoothing of range data, would be in the neighborhood of 1.2. From the experiement conducted, we can conclude the following: A very slow convergence of Chebychev approximation for the solution coefficients ($\lambda = 1$, and the number of iterations employed is 10 for all images without attaining complete convergence for the solution estimates). A very fast convergence of the least square approximation ($\lambda = 2$, and the number of iterations employed is a maximum of 3 for all images with a sure convergence of the solution estimates), but with the additional property of being most sensitive to noise. This effect is clearly illustrated with Fig. 5.6(d) where it is evident that least square estimates were unreliable. For an intermediate λ–estimator, such as $\lambda = 1.2$, we are gaining considerable speed in convergence, and maintaining a reasonable level of robustness to outliers. This is best illustrated with the aid of Fig. 5.6(d) where the estimator ($\lambda = 1.2$) clearly outperforms the Chebychev and the least square approximations.

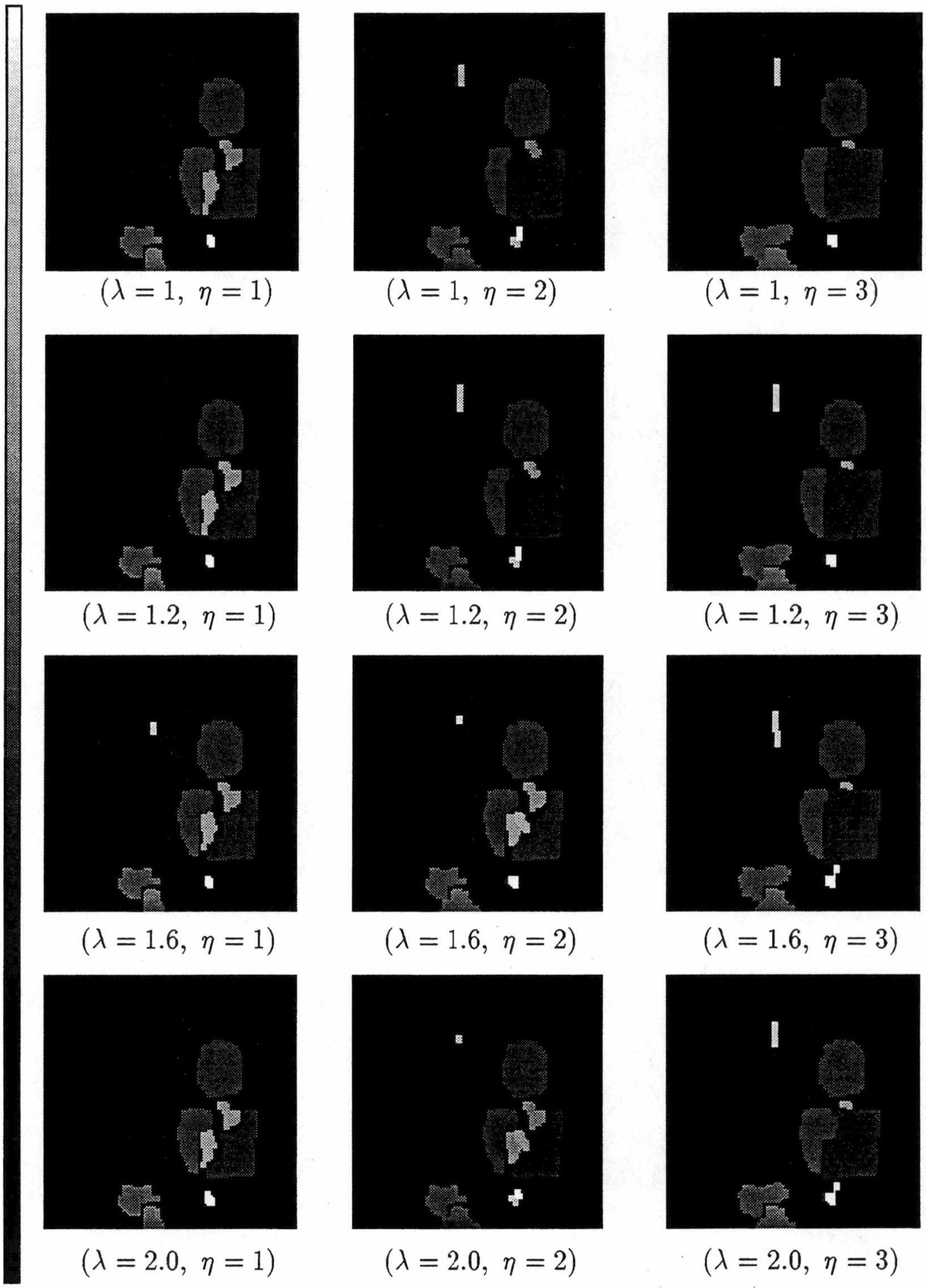


Figure 5.8: *Effects of (λ, η) pair on segmentation of image* SHAPES.

CHAPTER 6

Conclusions

It has been demonstrated with extensive experimental evidence the high utility of the **GWLS** technique in providing accurate derivative estimates in range images. The approach was compared to the classical **EWLS** technique and shown to be superior for all window sizes used. This technique is also useful for other images such as intensity images. The execution time of computing the derivatives from range images was minimized with the generated tables of convolute integers. Two segmentation algorithms were presented to provide a partition of the range image into its homogeneous subcomponents. Both of the algorithms rely heavily on this technique, **GWLS**, as a starting point.

The first algorithm extracts the edges in the range image in a number of scales in order to maximize the reliability of the detection. This demonstrates the ability of the **GWLS** technique in providing information about the range image in a variety of scales. The described algorithm is robust, since the larger window used ($N = 9$) is more robust with respect to noise in range images. The smaller window, $N = 5$, was used to localize the features in the range image. The procedure eliminates the scale–space tracking problem for which existing solutions rely solely on heuristics. A highly useful technique for linking broken edges was developed. The technique was based on morphology concepts, and simple connected component analysis routines. We have shown also the importance of complementing the edge–based module with a region based module in order to refine the result. This provides additional information about each region of the range image that can be used in a higher level process. The algorithm is capable of providing an accurate segmentation of real range images which contain all shapes of practical interest such as spheres, cylinders, planes, and cones.

The second algorithm computes the gradient magnitude of the range image, and the gradient magnitude of the surface normal image at a single scale using the **GWLS** technique. This provides a vivid demonstration of the usefulness of the **GWLS** technique in providing reliable feature enhancement techniques in real range images which have practical value and various complexity degrees. The Dempster–Shafer formalism was used to combine the two resulting feature images. The result of fusion was shown experimentally to provide a more complete gray–scale boundary description of the original range image. The Morphological Watershed algorithm was subsequently invoked under a meaningful set of conditions. The results of this segmentation were highly accurate for synthetic and real range data.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. Abdulghafour. *Data Fusion Through Fuzzy Reasoning Applied to Feature Extraction From Multi–Sensory Images.* PhD thesis, The University of Tennessee, Knoxville, U.S.A., 1992.

[2] R. L. Anderson and E. E. Houseman. Tables of Orthogonal Polynomial Values Extended to $n = 104$. *Iowa State College of Agriculture and Mechanics Arts, Ames, Iowa*, Reaserch Bulletin 297:594–672, April 1942.

[3] B. Klaus and P. Horn. *Robot Vision.* The MIT Press, Cambridge, MA, 1986.

[4] R. C. Gonzalez, M. Baccar, and M. A. Abidi. "Segmentation of Range Images via Data Fusion and Morphological WaterSheds.". In *Proc. of the 8th Scandinavian Conf. on Image Anal., Norway*, May 1993.

[5] P. R. Beaudet. Rotationally Invariant Image Operators. *Proc. of 4th Int. Conf. Patt. Rec., Kyoto, Japan*, pages 579–583, November 1978.

[6] P. J. Besl and R. C. Jain. Invariant Surface Characteristics for 3–D Object Recognition in Range Images. *Computer Vision, Graphics, and Image Processing*, 33:33–80, June 1985.

[7] P. J. Besl and R. C. Jain. Segmentation Through Variable–Order Surface Fitting. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI–10, no. 3:167–192, March 1988.

[8] R. M. Bolle and D. B. Cooper. Bayesian Recognition of Local 3–D Shape by Approximating Image Intensity Functions with Quadric Polynomials. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI–6, no. 4:418–429, April 1984.

[9] P. Boulanger and G. Godin. Multiresolution Segmentation of Range Images Based on Bayesian Decision Theory. *Sub. IEEE Trans. Patt. Anal. Mach. Intell.*, 1991.

[10] H. D. Brunk. *An Introduction to Mathematical Statistics.* Blaisdell Publishing Company, Waltham, MA, 1965.

[11] R. L. Burden. *Numerical Analysis, Fourth Edition.* PWS–KENT, Boston, MA, 1986.

[12] D. C. Hoaglin, F. Mosteller, and J. W. Tukey. *Understanding Robust and Exploratory Data Analysis.* John Wiley, New York, NY, 1983.

[13] P. Meer, D. Mintz, and A. Rosenfeld. Analysis of the Least Median of Squares Estimator for Computer Vision Applications. *Proc. 1992 IEEE Com. Soc. Conf. on Computer Vision and Pattern Recognition*, pages 621–623, June 1992.

[14] E. R. Dougherty. *An Introduction to Morphological Image Processing.* SPIE, Bellingham, WA, 1992.

[15] T. R. Edwards. Two–Dimensional Convolute Integers for Analytical Instrumentation. *Anal. Chem.*, 54:1519–1524, August 1982.

[16] P. J. Rousseeuw, F. R. Hampel, E. M. Ronchetti, and W. A. Stahel. *Robust Statistics, The Approach Based on Influence Functions.* John Wiley, New York, NY, 1986.

[17] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, Second Edition.* Addison Wesley, Reading, MA, 1992.

[18] M. Gutknecht. *Lectures on Numerical Mathematics.* Birkhauser, Boston, MA, 1990.

[19] R. M. Haralick. Edge and Region Analysis for Digital Image Data. *Computer Vision, Graphics, and Image Processing*, 12:60–73, 1980.

[20] R. M. Haralick. Digital Step Edges from Zero Crossings of Second Directional Derivatives. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-6, No. 1:58–68, 1984.

[21] R. M. Haralick and L. G. Shapiro. Survey: Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, 29:100–132, 1984.

[22] R. M. Haralick and L. Watson. A Facet Model for Image Data. *Computer Vision, Graphics, and Image Processing*, 15:113–129, 1981.

[23] I. Pitas and A. N. Venetsanopulos. *Nonlinear Digital Filters: Principles and Applications.* Kluwer Academic Publishers, Norwell, MA, 1990.

[24] V. Koivunen and N. Pietikainen. A Combined Edge and Region–Based Method for Range Image Segmentation. *SPIE Proc., Vol. 1381, Intelligent Robots and Computer Vision IX*, November 1990.

[25] V. Koivunen and N. Pietikainen. Improving the Robustness of Edge and Region–based Range Image Segmentation. *SPIE Proc., Vol. 1608, Intelligent Robots and Computer Vision X*, November 1991.

[26] M. A. Abidi and R. C. Gonzalez (Eds.). *Data Fusion in Robotics and Machine Intelligence.* Academic Press, Boston, MA, 1992.

[27] D. Mintz. Robust Consensus Based Edge Detection. *Proc. 1992 IEEE Com. Soc. Conf. on Computer Vision and Pattern Recognition*, pp. 651–653, June 1992.

[28] Odetics, Inc. *3–D Laser Imaging System User's Guide.* Anaheim, CA, 1990.

[29] P. J. Besl. *Surfaces in Range Image Understanding.* Springer–Verlag, New York, NY, 1986.

[30] P. J. Huber. *Robust Statistics.* John Wisley, New York, NY, 1981.

[31] B. Parvin and G. Medioni. Adaptive Multiscale Feature Extraction from Range Data. *Computer Vision, Graphics, and Image Processing*, 45:346–355, October 1989.

[32] R. G. Staudte and S. J. Sheather. *Robust Estimation and Testing.* John Wisley, New York, NY, 1990.

[33] J. C. Russ. *Image Processing Handbook.* CRC Press, Boca Raton, FL, 1992.

[34] A. Savitsky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.*, 36:1627–1639, July 1964.

[35] J. C. Sluder and M. A. Abidi. "Parallel Range Data Processing: A Real Case Study". *SPIE Intelligent Robots and Computer Vision X*, vol. 1608, pp. 248–261, November 1991.

[36] F. J. Sweeny. Random and Systematic Errors in Laser Range Camera Data. *Preceedings of the 39th Conference on Remote Systems Technology*, November 1991.

[37] L. T. Watson T. C. Pong, L. G. Shapiro and R. M. Haralick. Experiments in Segmentation Using a Facet Model Region Grower. *Computer Vision, Graphics, and Image Processing*, 25:1–18, 1984.

[38] T. J. Fan. *Describing and Recognizing 3–D Objects Using Surface Properties.* Springer–Verlag, New York, NY, 1990.

[39] N. Yakoa and M. D. Levine. Range Image Segmentation Based on Differential Geometry: A Hibrid Approach. *McRCIM-TR-CIM*, pages 1–48, September 1987.

VITA

Mohamed Baccar was enrolled in the graduate program at the University of Tennessee, Knoxville since January 1990. He has held a Teaching Assistant position and monitored the electronics laboratory for a year. Subsequently, he joined the research group with Prof. M. A. Abidi and been involved since. He has 2 publications dealing with Range Data Analysis and Segmentation. His main contribution lies in the introduction of the Gaussian Weighted Least Squares technique to the feature extraction problem in range images. He has also demonstrated the applicability of the L_p norms in regression.