ABSTRACT

DETECTION OF PLANAR FACETS IN NOISY RANGE IMAGES

by

Ajey S. Atre

Segmentation of the image is one of the major tasks of a machine vision system designed for constructing a three-dimensional representation of the object being imaged. A robust approach for segmenting planar surfaces from range images is presented in this paper. An algorithm based on clustering through fuzzy covariance matrices, which has been proposed by Gustafson and Kessel is considered for planar segmentation. However this algorithm performs poorly if the data is noisy, which is usually the case in real life applications. In order to handle noisy data, a robust modification, based on the "noise clustering" concept, is introduced to the algorithm. This modification is found to work very well in noisy data. Another alogrithm called the Adaptive Fuzzy c-Elliptotypes has been used by Dave for detecting lines in 2-D digital images, this algorithm is also considered for range image segmentation. The robust modification of this algorithm is used for planar segmentation of 3-D range images and is found to perform well. Examples of range image data are included to show the effectiveness of the algorithms proposed.

DETECTION OF PLANAR FACETS IN NOISY RANGE IMAGES

by
Ajey S. Atre

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Department of Mechanical and Industrial Engineering

October 1993

APPROVAL PAGE

DETECTION OF PLANAR FACETS IN NOISY RANGE IMAGES

Ajey S. Atre

Dr. Rajesh Dave, Thesis Adviser (Date)
Associate Professor of Mechanical Engineering, NJIT

Dr. Nouri Levy, Committee Member (Date)
Associate Professor of Mechanical Engineering, NJIT

Dr. Zhiming Ji, Committee Member (Date)
Assistant Professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Ajey Shashikant Atre

Degree: Master of Science in Mechanical Engineering

Date: October 1993

Undergraduate and Graduate Education:

- Master of Science in Mechanical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1993
- Bachelor of Engineering (Mechanical Engineering),
  College of Engineering, Pune, India, 1991

Major: Mechanical Engineering

This thesis is dedicated to

my parents

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my adviser Dr. Rajesh Dave for his guidance and motivation throughout this research. I also would like to thank him for leading me through my Masters program.

Special thanks to Dr. Nouri Levy for his help and advice which made it possible for me to reach higher standards. My thanks go to Dr. Zhiming Ji for providing help in the technical documentation of this thesis.

Thanks to Kurra Bhaswan for his helpful tips on coding of the algorithms and Shailesh Jadav for his editorial support.

Finally I would like to thank Anand Atre for his moral support and advice during this effort.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Introduction to Segmentation of 3-D Range Images

Generating a computer compatible mathematical model of an object is the first step in the CAD/CAM process which is normally achieved by using solid modeling softwares. This can become a time consuming and complicated step if the object to be modeled is intricate. In these situations it is possible to use 3-D information about the object to generate the required mathematical model. In some cases like reverse engineering and orthopedic biomechanics 3-D information is readily available and can be used. The use of range images for this purpose also looks promising.

Range images and other types of 3-D data sets are being widely used in the fields of machine vision, computer aided design & computer aided manufacturing (CAD/CAM) and robotics. Range cameras are currently used in space and defense applications and are expected to be used for manufacturing applications in the near future. Other devices like coordinate measuring machines are already being used in the industry for generating 3-D data sets. Another form of 3-D data is the CAT scan image which is widely used in orthopedic biomechanics.

A range images must first be processed to generate the desired mathematical model. The first step in this processing is segmentation of the image into surface patches of complete surfaces, which can then be further processed for 3-D object recognition. The purpose of this work is to propose and evaluate methods to obtain planar segmentation of range images. Planar segmentation is considered since it is easy to represent an object with planar facets and is a standard method to do so in most of the CAD/CAM softwares.

The usual format of range image is such that for each point on the surface of the object, there is a real vector of XYZ coordinates. These surface points have to be grouped into meaningful clusters which can be used in further processing. Therefore, segmenting is essentially a clustering problem of unlabeled data points.

1.2 Introduction to Cluster Analysis

Clustering of a data set X means the identification of c clusters, $2 \leq c < n$, where n is the number of data points in X. The data points are clustered according to some common mutual relationship, so the clusters identified should be such that all the points in a single cluster share a common relationship which is stronger that their relationships with points classified into other clusters. The structure of the clusters sought dictated the relationship between points. A measure has to be established to enable the algorithm to classify the data set into clusters of a specific structure. This measure also called as the clustering criterion is based on some mathematical property of the points is the data set e.g distance, angle, curvature, symmetry, intensity etc.

The importance of the right choice of clustering criterion is discussed by Bezdek[1]. The specification of a clustering criterion has to accompany a good clustering method for successful cluster analysis. The main classes of clustering methods are Hierarchical, Graph-Theoretic and Objective Function. The final issue to be considered is the type of memberships to be used in the analysis. The memberships can be hard or fuzzy. In the hard approach a data point can be assigned to one cluster only where as in the fuzzy approach a data point can be given a membership with respect to each cluster. Hence the fuzzy membership of a data point can have a value between 0 and 1, where as a hard membership has a value of 0 or 1.

1.3 Literature Review

Various techniques for range image segmentation have been reported. These techniques can be broadly classified into region based and edge based techniques. Region based techniques attempt to group data points into surface regions based on homogeneity or similarity of surface properties [2,3,4]. This approach assumes that parts of the object surface can be well approximated by a particular function. The edge based techniques try to extract discontinuities in the properties of the object surface to detect the closed boundary of the object [5,6,7]. A hybrid approach combining both these techniques has been reported by Yokoya and Levine [8].

Besl and Jain[2] have developed an algorithm that approximates the image data with bivariate functions to compute complete noiseless reconstruction. The algorithm first generates an initial coarse segmentation and this is refined using variable order surface fitting. This algorithm works well for images that can be represented by piecewise-smooth surfaces but the methods used for noise estimation need to be improved. The hybrid approach proposed by Yokoya and Levine[8] is successful in identification of distinct surface regions that are adjacent. It provides a rich description of the surfaces detected. This approach requires several steps of processing and its performance in presence of noise is not documented.

The region based approach is very popular, but Hoffmann and Jain [5] have used the edge approach over the region approach as the region approach tends to merge surfaces connected by smooth edges and the parameters involved are affected by noise. They have used a three stage procedure that detects surface patches in the first stage, classifies these patches as planar, convex or concave in the second stage and classifies boundries between these patches as crease or non-crease. Clustering techniques are used for surface patch detection but the effect of noise is not considered.

Jolion, Meer and Bataouche [9] have recently introduced a algorithm based on minimum volume ellipsoid robust estimators. This algorithm requires no *apriori* information about the number of clusters, it is successful in segmentation of range images and has a good tolerance to noise. But the clusters tend to loose their clarity as the amount of noise increases. This approach has managed to address all the issues involved in segmentation of range images and produce impressive results for simple objects.

Various clustering techniques based on fuzzy objective functions have been reported [10],[11],[12]. These techniques work well when clusters are spherical (or hyper-spherical) in shape e.g fuzzy c-means (FCM) clustering algorithm reported by Bezdek[1]. Gustafson and Kessel [13] proposed an algorithm, (here after called GK algorithm) that used covariance matrices to detect clusters of different geometrical shapes in the same data set. This approach was applied by Krishnapuram and Ferg[14] to detect linear and planar clusters. The algorithm provides good results on noise free range images.

Edge detection techniques have been used on 2-D images. Petrou and Kittler[20] have used ramp filters for edge detection. The Adaptive Fuzzy c-Elliptotype Clustering (AFC) algorithm proposed by Dave has been shown to work for segmenting lines from 2-D digital images[15,16].

Several researchers have addressed this problem of noise in cluster analysis and have suggested methods to improve the performance of clustering algorithms in such cases. Jain and Dubes[11] suggested a method for removing noise points from data sets before applying clustering algorithms. This, however, is very hard to achieve in many cases, because the structure of noise is generally unknown, and if there are many outliers, the task of identifying them apriori is very difficult. Jolion and Rosenfeld[17] proposed an approach that assigned weights to data points based on the relative local data density. Therefore, in principle, points belonging to good clusters are assigned higher weights than noise points. The problem, however, in this approach is that it may be difficult to compute a meaningful value of local data density for clusters other than spherical shapes. There are also several methods based on the principles from robust statistics [9], but in general the problem of noisy data remains rather difficult to handle.

Recently, Dave[18] presented a novel method that assigned noise points to a cluster called the *noise cluster* thus reducing their effect on the clustering algorithm. The idea is based on defining the noise as a prototype, and is applicable to all the objective function based clustering algorithms.

1.4 Objective of this thesis

The objective of this work is to propose a robust approach for segmentation of noisy range images into planar facets. This is to be achieved by developing robust algorithms for segmentation of noisy range data into planar clusters. The use of fuzzy objective functions as a method of clustering is proposed since these do not require multiple step processing of the data. An attempt is made to develop such algorithms by modifying existing algorithms.

The GK algorithm has been proved to work for detecting planar clusters in range images. However, its performance in the presence of noise is very poor. This work will use the method proposed by Dave[18] to improve the performance of the GK algorithm.

The AFC algorithm has been used for segmentation of lines in 2-D digital images[16]. This algorithm will be extended to 3-D to detect planar clusters in range images. The issue of noise will be addressed here also.

CHAPTER 2

FUZZY OBJECTIVE FUNCTION ALGORITHMS

2.1 Introduction

To partition a data set into clusters we should first know the clustering criterion. Thus we have to use some mathematical property like distance, curvature, intensity etc. of the points in the data set that will enable us to detect clusters of a particular structure. Using objective functions allows a precise formulation of the clustering criterion [12]. So the extrema of the objective function will give us the optimal partitioning of the data. Since our approach is fuzzy these functions are called fuzzy objective functions. For example if one considers as the similarity measure the Euclidean distance of data points from the cluster center, and as a measure of cluster quality the overall within-group sum of squared errors, then the objective function is the sum of squared errors. This clustering criterion is called minimum variance objective.

In this chapter the GK and AFC algorithms which use fuzzy objective functions will be discussed. The FCM functional is described first as a stepping stone to explain the GK algorithm. The FCV functional is discussed next, followed by the AFC algorithm.

2.2 The FCM Functional

The generalization of the minimum variance objective mentioned earlier, leads to many infinite families of fuzzy clustering algorithms that have been developed and used by a number of investigators. The initial generatization of this squared error function and an algorithm akin to hard c-means was reported by Dunn[19]. The fuzzy c-means functional has been defined as

Let $J_m : M_{fc} \times R^{cp}$ be

$$J_m(U,v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 \tag{2.1}$$

where:

M_{fc} is the fuzzy partition space,

R^{cp} is the cp-tuples of real numbers,

$U \in M_{fc}$ is a fuzzy c-partition of data set X,

$(v = (v_1, v_2, \ldots, v_c)) \in R^{cp}$ are the cluster centers of U,

$(d_{ik})^2 = \|x_k - v_i\|^2$ and $\| \bullet \|$ is any inner product induced norm on R^p,

c is the number of clusters,

n is the number of data points in X and

$m \in [1, \infty)$ is a weighting exponent.

Examination of J_m reveals that the dissimilarity measure d_{ik}, is the distance between each data point x_k and a cluster center v_i; the squared distance is then weighted by $(u_{ik})^m = (u_i(x_k))^m$, the mth power of x_k's membership in the cluster u_i. Thus the minimization of J_m will yield the least-squared error stationary points of J_m. The following theorem has been by Bezdek [12] for the minimization of J_m with respect to the memberships and distances calculated.

Theorem 1:

Assume $\| \bullet \|$ to be inner product induced norm, fix $m \in [1, \infty)$ and let X have $n > c$ distinct points,define the sets

$$I_k = \{ i | 1 \le i \le c; d_{ik} = 0 \}$$

$$\tilde{I}_k = \{1, 2, \ldots, c\} - I_k$$

Then $(U, v) \in M_{fc} \times R^{cp}$ may be globally minimal for J_m only if

$$I_k = \varnothing \Rightarrow u_{ik} = \frac{1}{\left[\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{(m-1)}}\right]} \tag{2.2}$$

or

$$I_k \neq \varnothing \Rightarrow u_{ik} = 0 \forall \tilde{I}_k \text{ and } \sum_{i \in I_k} u_{ik} = 1 \tag{2.3}$$

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \forall i \tag{2.4}$$

The proof of this theorem can be obtained in [12]. The FCM algorithm[1] that originates from the functional described above has been widely used for detection of well separated spherical or round clusters.

2.3 The GK Algorithm

Gustafson and Kessel [13] proposed a modification of the FCM algorithm in an attempt to recognize the fact that different clusters in the same data set may have different geometrical form. The norm controls the shape of all c clusters identified with $J_m(U,v)$. An algorithm can be able to detect clusters of different shape in the same data set if the norm is varied for each individual cluster. As explained by Bezdek [12] mathematical realization of this idea is accomplished by considering the class of inner product norms induced on R^p by symmetric positive definite matrices in vectors space p, where p is the dimension of the data set. Let us denote by A a c-tuple of matrices, so $A = (A_1, A_2, \ldots A_c)$. Let the weighted inner product induced by R^p by A_i be $\langle x,x \rangle A_i = \|x\|_A^2 = x^T A_i x$. Thus the distance between x,y in R^p in the weighted norm is $\|x-y\|_A$.

Extending this idea to J_m we can define

$$J_{mgk}(U,v,A) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m \|x_k - v_i\|^2_{A_i} \tag{2.5a}$$

The clustering criterion employed by J_{mgk} and J_m are the same. The basic difference being that in J_m all the distances in $\{d_{ik}\}$ are measured by a prepre-scribed norm where as in J_{mgk} c different norms are sought by the functional. These distances are given by

$$(d_{ik})^2 = \langle x_k - v_i\rangle^T A_i \langle x_k - v_i\rangle \tag{2.5b}$$

The minimization of J_{mgk} with respect to A will yield the least-squared-error stationary points of J_{mgk}. To render the minimization of J_{mgk} with respect to A trac-table, each A_j is constrained by requiring the determinant of A_j to be fixed. Specifi-cation of $\det(A_j) = \rho_j > 0$ for each j = 1 to c amounts of constraining the volumes of cluster u_j along the jth axis. Allowing A_j to vary while keeping its determinant fixed corresponds to seeking the optimal shape fitting the data points for a fixed volume for each cluster.The following theorem has been presented by Gustafson and Kessel for calculating the norm inducing matrix A.

Theorem 2:

If PD^c represents the c-fold cartesian product of a set of symmetric positive defi-nite matrices in vector space p x p.

Let $\eta : PD^c \rightarrow R, \eta(A) = J_{mgk}(U, v, A)$

Where (U,v) satisfy equations 2.2, 2.3 and 2.4, under the hypothesis of theorem 1. If m > 1 and for each j, $\det(A_j) = \rho_j > 0$ is fixed, then A is a local minimum of η only if

$$A_j = [\rho_j \det(S_{fi})]^{\left(\frac{1}{p}\right)}(S_{fi}^{-1}),\ 1 \le j \le c \tag{2.6}$$

Where,

$$S_{fi} = \sum_{k=1}^{n} (u_{ik})^m (x_k - v_i)(x_k - v_i)^T \tag{2.7}$$

is the fuzzy scatter matrix of u_j.

The proof of this theorem can be obtained in [1].

The GK algorithm based on the above mentioned theorem is as follows

Algorithm 1 (GK).

1. Fix c the number of clusters, $2 \le c \le n$, where n is number of data points;

   Fix fuzzifier $m \in (1, \infty)$

   Fix c volume constraints $\rho_j \in (0, \infty)$, $1 \le j \le c$

   Initialize membership matrix $U^{(0)}$

   At steps l, l = 0,1,2,......

2. Calculate the c fuzzy cluster centers $\{v_i\}^l$ using memberships $U^{(l)}$
3. Calculate c fuzzy scatter matrices $\{S_{fi}\}^{(l)}$. Calculate their determinants and inverses.
4. Calculate the norm inducing matrix $\{A_j\}^{(l)}$
5. Update $U^{(l)}$ to $U^{(l+1)}$
6. If $(U^{(l+1)} - U^{(l)}) \le \varepsilon$ stop, else return to 2 with l = l+1.

In step 2 we use equations 2.4 to calculate the cluster centers. In step 3 the fuzzy scatter matrices are calculated from equation 2.7 and the norm inducing matrices are calculated using equation 2.6. In step 5 we use equation 2.2 and 2.5B to update the membership matrix. The GK algorithm is a simple Picard iteration and it is stoped if the change in memberships in not significant in step 6.The ability of GK to detect clusters of different shapes in the same data set and its comparision to other algorithms is documented in [12].

The GK algorithm when used for clustering of 3-D range data yields very good results. This has been shown by Krishnapuram and Ferg[14] and verified in this work. The clusters detected are essentially surface patches. In this case the data is 3-Dimensional and hence the value of p is 3. The norm inducing matrices and the scatter matrices will be of order 3 x 3. The eigenvector of each scatter matrix give the orientation of the cluster and the eigenvalues give the length of the cluster in direction of the corresponding eigenvector.

2.4 The Fuzzy c-Varieties Functional

Gustafson and Kessel attempted to improve the ability of J_m to detect different cluster shapes in a fixed data set by locally varying the metric topology around a fixed kind of prototype namely the cluster centers which are prototypical data points in the real space of dimension p. Another attempt by Bezdek [12] to enhance the ability of J_m to detect nonhyperelliptically shaped substructures takes an approach which is in some sense opposite to that embodied by the GK algorithm. In the fuzzy c-varieties functional defined by Bezdek the norm inducing matrix A is fixed globally, but allows the c prototypes to be r-dimensional linear varieties, $0 \le r \le p - 1$, rather than just points (cluster centers).

This type of objective functional is most amenable to data sets which consist essentially of c clusters, all of which are drawn from linear varieties of the same dimension.

Definition 1 (Linear Variety).

The linear variety of dimension r, $0 \le r \le p$, Through point $v \in R^p$ spanned by the linearly independents vectors $\{s_1, s_2,...,s_r\}$, is the set

$$V_r(v; (s_1, s_2 ... s_r)) = \left\{ y \in R^p \middle| y = v + \sum_{j=1}^{r} t_j s_j; (t_j \in R) \right\} \quad (2.8)$$

In 2.8 if v is a zero vector, then V_r is just the linear hull or span of the $\{s_i\}$, an r-dimensional linear subspace through the origin parallel to the set in 2.8. Linear varieties of all dimensions are thought of as "flat" sets in real space of dimension p and r is the number of directions in which this "flatness" extends. Certain linear varieties have special names and notations:

$$V_0(v;\varnothing) = v \qquad \text{"points"} \qquad (2.9)$$

$$V_1(v;s) = L(v;s) \qquad \text{"lines"} \qquad (2.10)$$

$$V_2(v;(s_1, s_2)) = P(v;(s_1, s_2)) \qquad \text{"planes"} \qquad (2.11)$$

So we call V_o a point; V_1 a line parallel to s; V_2 a plane through v parallel to the plane spanned by $\{s_1,s_2\}$.

A fuzzy clustering criterion which recognizes varietal shapes can be based on distances from data points to prototypical linear varieties. Specifically the orthogonal(OG) distance (in the A norm of real space) from x to V_r, when $\{s_r\}$ are an orthonormal basis for their span, and the distance is given by

$$D_A(x, V_r) = \left[\|x - v\|_A^2 - \sum_{j=1}^{r} (\{x - (v, s_j)\} A)^2 \right]^{\frac{1}{2}} \qquad (2.12)$$

The weighted objective function to be optimized in this case is the natural extension of the functional J_m which measures the total weighted sum of squared OG errors from each point in the data set to each of the c r-dimensional linear varieties.

$$J_{Vrm}(U,v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})(D_{ik})^2 \qquad (2.13)$$

Where U is the fuzzy c-partition of X and D_{ik} is same as the distance given by 2.12. An examination of 2.13 reveals that for r = 0, J_{V0m} reduces to J_m the FCM functional; for r = 1 we get J_{V1m} which we will call J_1 the fuzzy c-lines (FCL) functional. Similarly for r = 2 we will get the fuzzy c-planes functional (FCP) J_2 and further values of r will give functionals for fuzzy hyperplanes.

2.5 The FCE Functional

Bezdek [12] has shown that the fuzzy c-varieties algorithms are reliable for the detection and characterization of substructure in a data set when all c clusters are essentially r-dimensional and linear varietal in shape. There may be, however, a disadvantage of the FCV which is intrinsic to the variety itself, namely, "size". For example, lines (varieties of dimension 1) have infinite length, and it may be the case that colinear clusters that are widely separated would be identified by this functional as one cluster. An example of this can be obtained in [12].

Bezdek[12] has suggested that the utility of J_1 can be considerably increased by forcing each cluster to contain a center of mass in or near its convex hull. Although the natural supposition stemming from there remarks would be to form a convex combination of J_m and J_1, it is a remarkable fact that arbitrary convex combinations of all the J_{Vm}'s are minimized over choice of dimensionally different varieties simply by using the linear varieties which are necessary for minimization of the individual term of highest dimension.

If we consider the convex combination of J_m and J_1, the new functional resulting called fuzzy c-elliptotype functional (FCE) can be defined as

$$J_e(U, V_0, V_1) = (1-\alpha) J_m + \alpha J_1 \tag{2.14}$$

Where J_m if the FCM functional and J_1 is the FCL functional. This equation can be seen as a combination of two equations and can be written as

$$J_e = \sum\sum (u_{ik})^m (Z_{ik})^2 \tag{2.15}$$

Where Z_{ik} is the modified distance given by

$$(Z_{ik})^2 = \alpha (D_{ik})^2 + (1-\alpha)(d_{ik})^2 \tag{2.16}$$

Where D_{ik} is the distance of data points from the linear prototype (obtained by substituting r=1 in equation 2.12) and d_{ik} is the distance of the data points from the cluster center(obtained by substituting r=0 in equation 2.12). The mixing coefficient α, then defines the proportion in which the distance is measured from a point and a line for a 2-D case, thus defining the elliptotype clusters.

2.6 The AFC Algorithm

The main short-coming of the algorithms based on the concept presented above is that one must have some knowledge about the cluster shapes in order to choose a value of the mixing coefficient. Since the same value of α is used for all the clusters, the algorithm will seek clusters of the same elliptotypical shape. Dave[16,21] and Gunderson[22] have addressed the problem of selection of α. In the approach used by Dave[17] each cluster can have a different value of α and can have any value between 0 and 1. This modification will change equation 2.16 as follows

$$(Z_{ik})^2 = \alpha_i (D_{ik})^2 + (1 - \alpha_i)(d_{ik})^2 \tag{2.17}$$

Where α_i is defined by Dave as

$$\alpha_i = 1 - \left(\frac{\beta_i}{\gamma_i}\right) \tag{2.18}$$

Where $\beta_i = \min(\lambda_{1i}, \lambda_{2i})$, $\gamma_i = \max(\lambda_{1i}, \lambda_{2i})$, $i = 1,2,..,c$.

In the above equations, λ_{ij}'s are the eigen values of the scatter matrix. The above definition utilizes the information from scatter matrices to derive different distances for each cluster.

The idea behind use of equation 2.18 is derived by Dave from the following argument. If we consider a 2-D case then the value of the mixing coefficient for a round cluster must be 0 and for a linear cluster it should be 1. In case of a round cluster the eigenvalues of the scatter matrix given by equation 2.7 should be equal, while for a linear cluster one eigenvalue should be zero or close to it. The above definition of the mixing coefficient will give the desired effect. This modification of the FCE functional was called adaptive fuzzy c-elliptotpye (AFC) functional. The algorithm proposed by Dave is presented below.

Algorithm 2 (AFC).

1. Fix c the number of clusters, $2 \leq c \leq n$, where n is number of data points;

   Fix fuzzifier $m \in (1, \infty)$

   Initialize value of mixing coefficient $\alpha_i \in ((0, 1), 1 \leq i \leq c)$

   Initialize membership matrix $U^{(0)}$

   At steps l, l = 0,1,2,......
2. Calculate the c fuzzy cluster centers $\{v_i\}^l$ using memberships $U^{(l)}$
3. Calculate c fuzzy scatter matrices $\{S_{fi}\}^{(l)}$. Calculate its eigen values and the corresponding mixing coefficient.
4. Update $U^{(l)}$ to $U^{(l+1)}$
5. If $(U^{(l+1)} - U^{(l)}) \leq \varepsilon$ stop, else return to 2 with l = l+1.

In step 2 we use equations 2.4 to calculate the cluster centers.In step 3 the fuzzy scatter matrices are calculate from equation 2.7 and the mixing coefficients are calclulate from equation 2.18. In step 5 the memberships are calculated using 2.2, the distances are given by equation 2.17.

This algorithm has been shown[17] to work for detecting clusters of different shapes in the same data set. Its does have a tendency to pick longer clusters as explained by Dave.

CHAPTER 3

NOISE IN CLUSTERING

3.1 Introduction

The presence of noise in the data set to be analysed is a common problem in cluster analysis. Noise arising due to the statistical distribution of the data from the measuring instrument can be tackled but the noise that appears completely arbitrarily is of real concern. In some cases even a few noise points can severly deteriorate the performance of the algorithm. Several researchers[9,11,17] have addressed the problem of noise in cluster analysis and various techniques have been recommended for tackling this problem.

The presence of noise in range data is very common. Hence the algorithms used for their segmentation should be able to handle noise. The techniques currently used do not perform well in noisy data and in some cases fail completely. The squared error type clustering algorithms are extremely susceptible to noise. Dave [18] has recommended a novel approach to improve the performance of these type of algorithms. This approach requires no preprocessing of the data set and has been shown to give excellent results for various type of 2-D data sets with multiple clusters. The approach will be used in this work by extending it to 3-D data sets to improve the performance of the existing GK algorithm for planar segmentation in noisy data. This technique will also be used on the modified AFC algorithm to evaluate its performance. The approach proposed by Dave is explained in this chapter.

3.2 Noise Prototype

The performance of squared error type algorithms is highly susceptible to noise, this is because in these type of algorithms each point in the data set including the noise points have to be assigned to a cluster. This causes some of the noise points to be assigned to good clusters hence deteriorating the performance of these algorithms. To eliminate the necessity of classification of noise points into good clusters Dave has introduced a concept of noise cluster, a cluster in addition to the clusters being sought, into which all the noise points can be dumped.

This approach is based on first defining a noise cluster and then defining a similarity (or dissimilarity) measure for the noise cluster. This measure will gauge the belonging of a point to the noise cluster. A scheme that allows the definition of noise as a prototype cluster is proposed in [18] where the noise prototype is defined as

Definition 2 (Noise Prototype).

Noise prototype is a universal entity such that it is always at the same distance from every point in the data set. Let v_c be the noise prototype, and x_k be the point in feature space, $v_c, x_k \in R^p$. Then the noise prototype is such that the distance d_{ck}, of point x_k from v_c is

$$d_{ck} = \delta, \forall k \qquad (3.1)$$

The definition tells us that all the points in the data set are at a distance δ from the noise prototype, it does not however tell us about the value of δ.The fact that all the points in the data set are at the same distance from the noise prototype is explained by Dave as that it indicates that all the points in the data set have an equal apriori probability of being assigned to the noise cluster, and as the algorithm progresses the good points increase their probability of being assigned to good clusters. Physically this means that the distance of good points from good cluster prototypes decreases below δ as the algorithm progresses.

If we consider that there are c - 1 clusters in a data set and let the cth cluster be the noise cluster. Then the functional J_N can be defined as

$$J_N(U,v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 \qquad (3.2)$$

Where the distance d_{ik} is defined as

$$(d_{ik})^2 = \langle x_k - v_i \rangle^T A_i \langle x_k - v_i \rangle \qquad (3.3)$$

for all k and i = 1 to c-1, and,

$$(d_{ik})^2 = \delta^2 \qquad (3.4)$$

for i = c.

Thus for a specified value of δ the minimization of J_N can proceed in the same manner as that of the functional described earlier.

The issue of selection of a value for δ was addressed in [18] and the following discussion is offered. An examination of equation 2.2 shows that the membership u_{ik} of a point x_k with respect to a cluster i depends not only on the distance of this point from cluster i , but also on the distance from all other clusters. Thus a point will have the highest membership for the cluster that is closest. Thus if δ is chosen very small, then most of the points will be classified to the noise cluster, while if δ is chosen too large, then most of the point will be classified into clusters other than the noise cluster. A proper selection of δ will result in a classification where the points that are close to good clusters will be classified into good clusters, while the noise points that are away from good clusters will get classified into the noise cluster.

Prespecification of δ is not easy since information necessary for fixing the value is not available and that this value would be different for different problems. Taking this into consideration, a scheme based on average interpoint distances is proposed in [18]. Interpoint distances reflect structural realtionship among the feature points. The proposed formula is

$$\delta = \lambda \left[\frac{\sum_{i=1}^{c-1} \sum_{k=1}^{n} (d_{ik})^2}{n(c-1)} \right] \tag{3.5}$$

where λ is the value of multiplier used to obtain δ from the average distances.

3.3 Noise Clustering Algorithm

The algorithm based on the functional above with the fixed point iteration scheme is presented as follows [18].

Algorithm 3 (Noise).

1. Fix c the number of clusters, $2 \le c \le n$, where n is number of data points;

   Fix fuzzifier $m \in (1, \infty)$

   Initialize value of δ

   Initialize membership matrix $U^{(0)}$

   At steps l, l = 0,1,2,......
2. Calculate the c fuzzy cluster centers $\{v_i\}^l$.
3. Calculate the fuzzy memberships $U^{(l)}$ and value δ
4. Update $U^{(l)}$ to $U^{(l+1)}$
5. If $(U^{(l+1)} - U^{(l)}) \le \varepsilon$ stop, else return to 2 with l = l+1.

In step 2 we use equations 2.4 to calculate the cluster centers for i = 1 to c - 1 cluster. In step 3 memberships can be calculated using equation 2.2 the distance is given by equation 3.3 and 3.4, δ is calculated using 3.5.

The algorithm mentioned above can be applied to a variety of clustering algorithms for cluster detection in noisy data. The proper selection of λ is essential and can be generally achieved by trial and error. The process of integration of this algorithm with another algorithm is discussed in the next chapter.

CHAPTER 4

DEVELOPMENT OF NEW ALGORITHMS

4.1 The NGK Algorithm

The GK algorithm was used by Krishnapuram and Ferg[14] for planar segmentation of range data. This algorithm like other algorithms of its class perform very poorly in the presence of noise and hence some kind of techniques have to be used to improve the performance of this algorithm. The approach propose by Dave[18] to handle noise in cluster analysis is used. By integrating the GK algorithm with the noise algorithm a new algorithm called the noise-GK algorithm (NGK) is developed. The process of integration is discussed in the following paragraphs.

As discussed earlier, the GK algorithm used a norm inducing matrix A_i to identify clusters of different shapes in the same data set. If we consider that there are c-1 clusters in a range image and using Dave's approach allow the cth cluster be the noise cluster, then the objective function for this algorithm can be defined as

$$J_{NGK}(U, v, A_i) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})(d_{ik})^2 \tag{4.1}$$

Where the distance d_{ik} is defined as

$$(d_{ik})^2 = \langle x_k - v_i \rangle^T A_i \langle x_k - v_i \rangle \tag{4.2}$$

for all k and i = 1 to c-1, and,

$$(d_{ik})^2 = \delta^2 \tag{4.3}$$

for i = c.

In equation 4.2 A_i is the norm inducing matrix given by equation 2.6.The algorithm corresponding to the minimization of this functional is as follows

Algorithm 4 (NGK).

1. Fix c the number of clusters, $2 \le c \le n$, where n is number of data points;

   Fix fuzzifier $m \in (1, \infty)$

   Initialize value of δ

   Fix c volume constraints $\rho_j \in (0, \infty)$, $1 \le j \le c$

   Initialize membership matrix $U^{(0)}$

   At steps l, l = 0,1,2,......

2. Calculate the c fuzzy cluster centers $\{v_i\}^l$ using memberships $U^{(l)}$

3. Calculate c fuzzy scatter matrices $\{S_{fi}\}^{(l)}$. Calculate their determinants and inverses. Calculate the value of δ

4. Calculate the norm inducing matrix $\{A_j\}^{(l)}$

5. Update $U^{(l)}$ to $U^{(l+1)}$

6. If $(U^{(l+1)} - U^{(l)}) \le \varepsilon$ stop, else return to 2 with l = l+1.

In step 2 we use equations 2.4 to calculate the cluster centers for i = 1 to c - 1 clusters . In step 3 the fuzzy scatter matrices are calculate from equation 2.7 and δ is calculated using 3.5.The norm inducing matrix is calculated in step 4 using 2.6. In step 5 the membership matrix is updated using 2.2 and the distances are given by equations 3.3 and 3.4.

This algorithm presented above is expected to perform well in the presence of noise. Its performance is discussed in the next chapter.

4.2 The NAFC Algorithm

The AFC Algorithm has been used by Dave[16] for detecting lines in 2-D digital images and was shown to provide results than the FCL algorithms. Moreover it was observed that if the mixing coefficient was fixed, the algorithm(i.e. FCE) failed to detect the correct lines in the images. These results indicate that the mixing coefficient has to be selected in an adaptive manner. As recommended by Dave, the AFC algorithm can be extended to the 3rd dimension to detect planes in 3-D range data. In the AFC algorithm for 2-D, elliptotypical clusters were detected using a convex combination of the FCM and the FCL functional, where the mixing coefficient was selected in an adaptive manner. For 3-D range image segmentation. A functional which is a convex combination of FCM and the fuzzy c-planes (FCP) functional can be used as suggested by Dave. Using such a functional we should be able to detect elliptotypical clusters in 3-D data sets, thus enabling us to detect planar clusters in 3-D range images. This section discusses the extension of AFC to 3-D and its integration with the noise algorithm.

Equation 2.17 is used in the AFC algorithm for distance calculations. In this equation D_{ik} is the distance of a data point from the linear prototype and d_{ik} is the distance of the data point from the cluster center. The linear varieties used in this case are V_1 and V_0 given by equations 2.10 and 2.9 respectively. In our case for 3-D we will be using V_2 (equation 2.11) and V_0 as prototypes for our distance calculations. At this point we can write the distance formula as

$$(Z_{ik})^2 = \alpha_i (D_{ik})^2 + (1 - \alpha_i) (d_{ik})^2 \qquad (4.4)$$

where D_{ik} is the distance of a data point from the planar prototype (obtained by substituting r = 2 in 2.12) and d_{ik} is the distance of the data point from the cluster center. The next issue to be addressed it the method for calculating the mixing coefficient α_i. It should be noted that now we have three eigen values compared to two in the 2-D case. When α_i is equal to 0 the cluster detected will be spherical and the cluster will be planar when α_i is very close to or equal to 1. This can be achieved by using the equation

$$\alpha_i = 1 - \left(\frac{\beta_i}{\gamma_i}\right) \qquad (4.5)$$

where $\beta_i = \min(\lambda_{1i}, \lambda_{2i}, \lambda_{3i})$, $\gamma_i = \max(\lambda_{1i}, \lambda_{2i}, \lambda_{3i})$, i = 1,2,..,c.

In the above equations, λ_{ij}'s are the eigen values of the scatter matrix. A closer look at this equation reveals that this is just the extension of equation 2.18 to 3-D. The last issue to be discussed is the integration of the noise algorithm with the new proposed algorithm. This is achieved by using the same approach as that used for the GK algorithm. Thus we consider the range image to have c - 1 good clusters and the cth cluster to be the noise cluster. Based on the discussion presented above we can now define the functional J_{NAFC} as

$$J_{NAFC}(U, V_0, V_2) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik}) (Z_{ik})^2 \qquad (4.6)$$

Where the distance Z_{ik} given by 4.4 and 4.5 for all k and i = 1 to c-1, and,

$$(Z_{ik})^2 = \delta^2 \qquad (4.7)$$

for i = c.

The algorithm for the minimization of the functional presented above can be achieved by the following algorithm.

Algorithm 5 (NAFC).

1. Fix c the number of clusters, $2 \le c \le n$, where n is number of data points;

   Fix fuzzifier $m \in (1, \infty)$

   Initialize value of mixing coefficient $\alpha_i \in ((0, 1), 1 \le i \le c)$

   Initialize value of δ

   Initialize membership matrix $U^{(0)}$

   At steps l, l = 0,1,2,......
2. Calculate the c fuzzy cluster centers $\{v_i\}^l$ using memberships $U^{(l)}$
3. Calculate c fuzzy scatter matrices $\{S_{fi}\}^{(l)}$. Calculate its eigen values and the corresponding mixing coefficient. Calculate value of δ.
4. Update $U^{(l)}$ to $U^{(l+1)}$
5. If $(U^{(l+1)} - U^{(l)}) \le \varepsilon$ stop, else return to 2 with l = l+1.

In step 2 we use equations 2.4 to calculate the cluster centers.In step 3 the fuzzy scatter matrices are calculate from equation 2.7 and the mixing coefficients are calculate from equation 4.5. In step 5 the memberships are calculated using 2.2, the distances are given by equations 4.4 and 4.7.

The above mentioned algorithm was coded and used for range image segmentation. The results are presented in the next chapter.

CHAPTER 5

EXAMPLES

5.1 Introduction

The algorithms presented in the earlier chapter were coded using the "c" programming language. Graphical interface was based on the HOOPS [23] graphics library subroutines. Linpack and Eispack [24] subroutines were used for eigen value calculations and matrix operations like calculating the inverse and determinant. These codes were used for the segmentation of range images obtained from PRIP (Pattern recognition and Image Processing) lab at the Michigan State University. Random artificially generated noise was introduced into these data files to accentuate the effect of noise. The code used for introducing the noise is presented in the appendix. Results of the segmentation are presented in the following sections. Each cluster detected by the algorithms is shown in a different shades of grey in the figures. The value of m (weighing exponent or fuzzifier) is taken as 2 for all the cases.The results obtained by using AFC with a fixed mixing coefficient (i.e. FCE) are not included to conserve space.

5.2 Images with Planar Surfaces

Figures 1,2,3 and 4 show the range image of a section of a staircase with four planar surfaces. The range image has 8477 data points. The noise points make up 5% of the data points. The outliers can be seen distinctly in the image while the noise points closer to the surface points cannot be distinguished by the eye. Figure 1 shows the segmentation obtained by using the conventional GK algorithm. The number of clusters c is taken as 4. It can be seen that one planar surface is detected correctly but the algorithm fails to detect the other planar clusters. The algorithm detects three clusters on the remaining data points and noise.

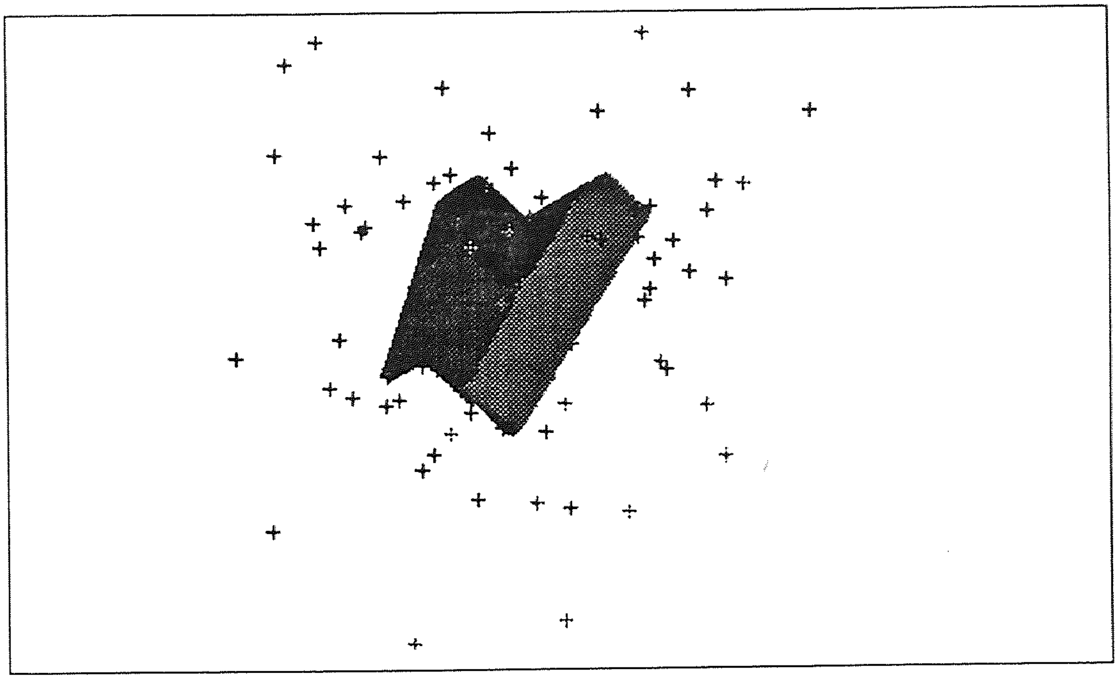

Figure 1 Segmentation of “Staircase” image using GK algorithm.

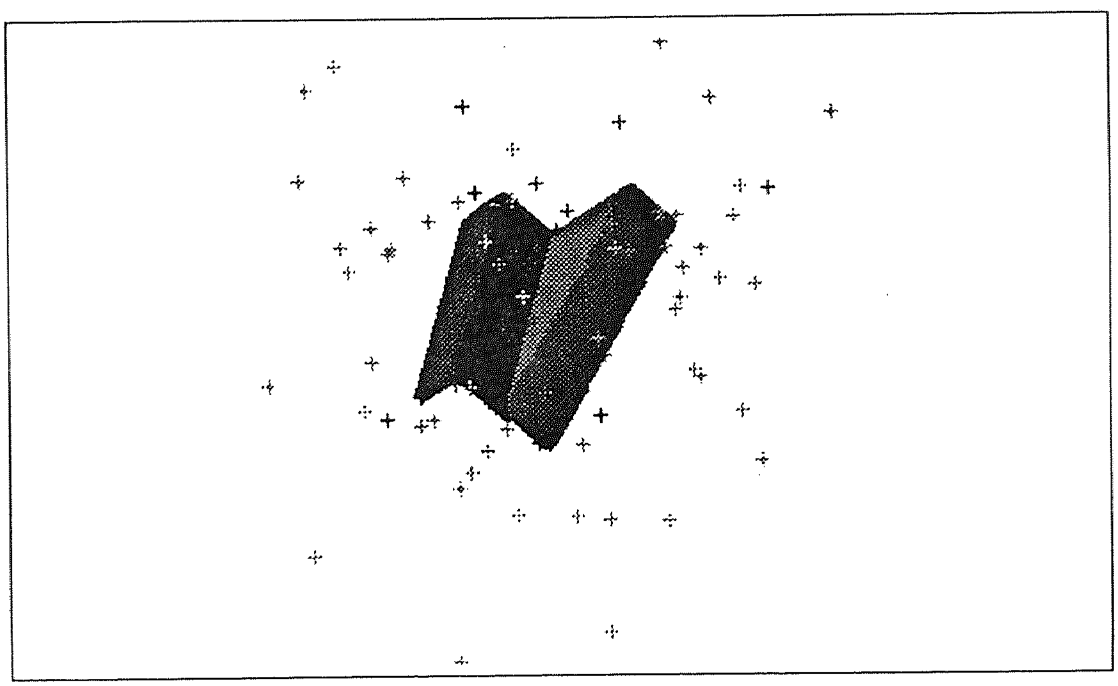

Figure 2 Segmentation of “Staircase” image using NGK algorithm.

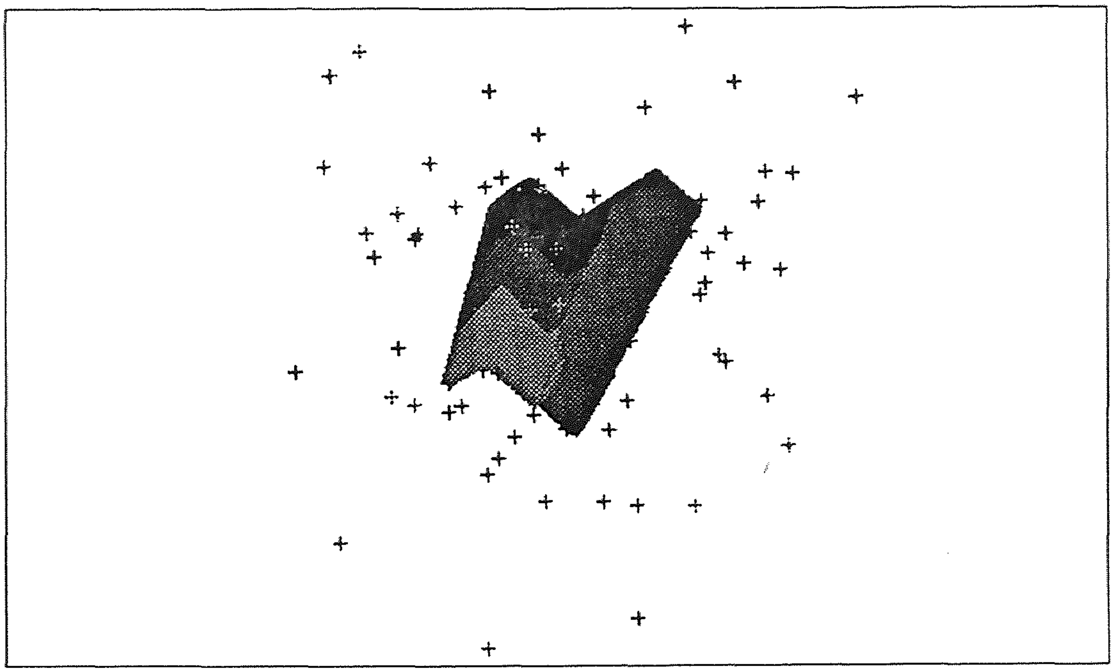

Figure 3 Segmentation of "Staircase" image using AFC algorithm.

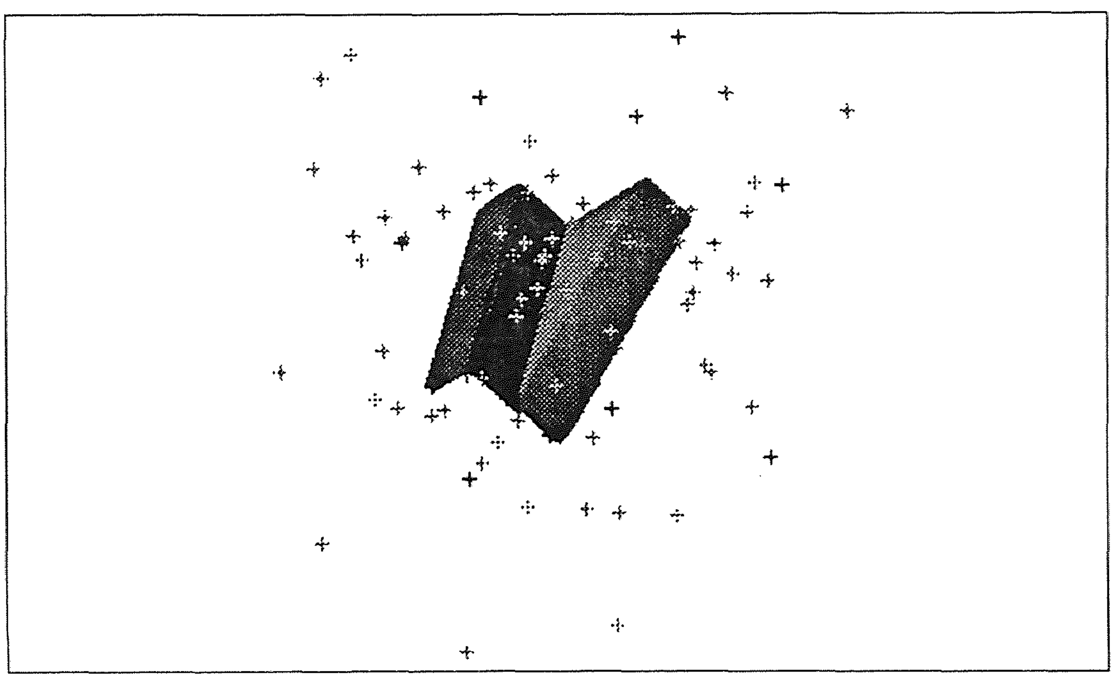

Figure 4 Segmentation of "Staircase" image using NAFC algorithm.

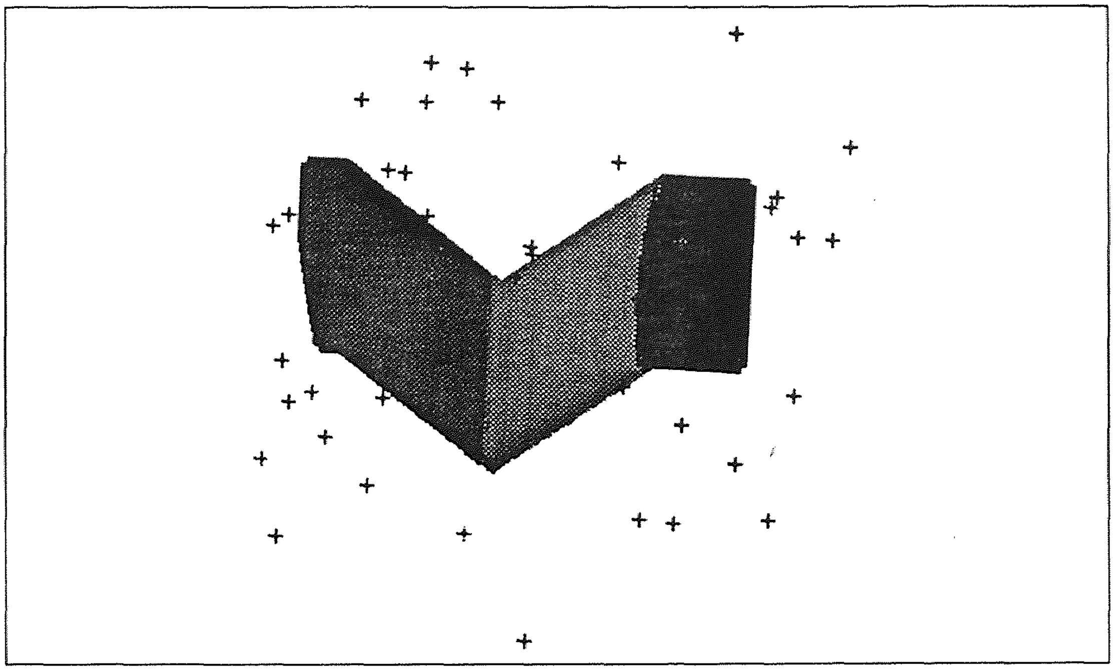

Figure 5 Segmentation of "Jig" image using GK algorithm.

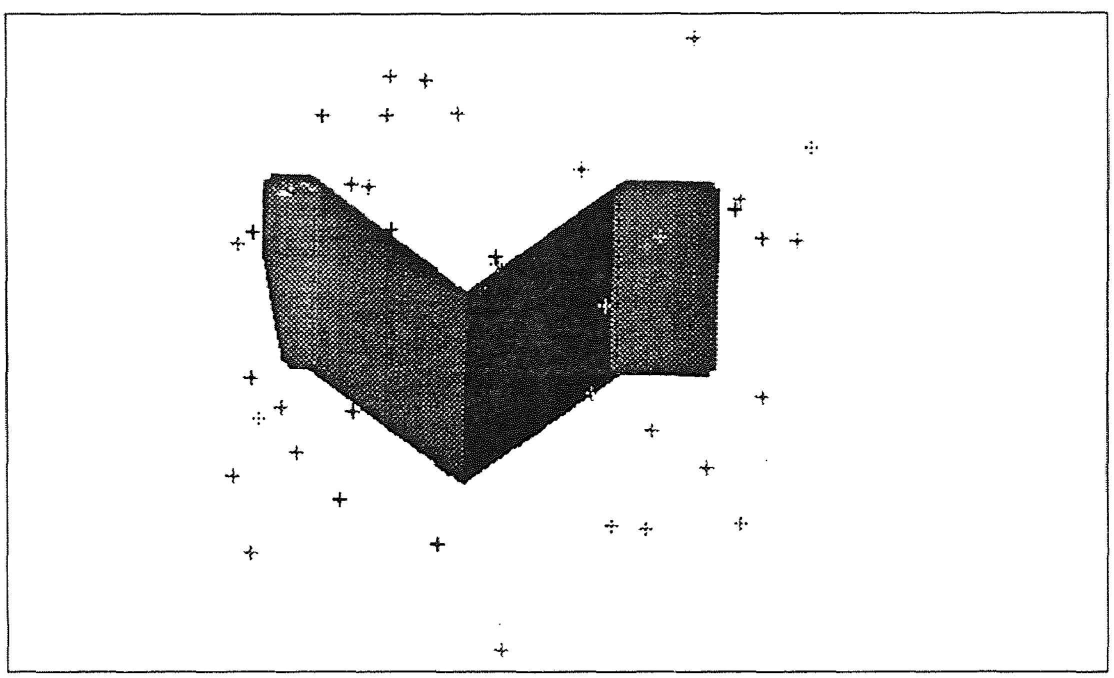

Figure 6 Segmentation of "Jig" image using NGK algorithm.

Figure 2 shows the segmentation obtained by using the NGK algorithm. In this case all four facets of the object are clearly detected and the noise points are dumped into the noise cluster. Figure 3 shows the segmentation of the same image using the AFC (for 3-D) algorithm, it can be seen that the algorithm fails to detect even one cluster clearly. Figure 4 shows the segmentation form the NAFC algorithm and excellent segmentation is obtained as the algorithm picks all the 4 facets correctly.

Figures 5,6,7 and 8 have the image of a section of a jig showing the "V" with flat surfaces at the two ends of the "V". This range image has 21148 data points, 5% of which are noise points. The image has 4 planar facets and all the algorithms used a value of c equal to 4. The segmentation obtained by the conventional GK algorithm is shown in figure 5 and as can be seen the algorithm detects two surfaces correctly though the edge between them is not clearly distinguished. The two remaining planes are classified as one cluster and the fourth cluster is lost in the noise points.Figure 6 shows the segmentation obtained by the NGK algorithm. All the facets are clearly detected and all the edges can be clearly distinguished. Figure 7 shows the segmentation given by the AFC algorithm and as can be seen it fails to detect a single facet clearly. Figure 8 shows the segmentation of the same range image obtained by using the NAFC algorithm, all the facets are detected clearly and the outliers are classified into the noise cluster.

5.3 Image with Cylindrical and Planar Surfaces

Figures 9,10,11 and 12 show the image of a cylinder with one cylindrical (partial) surface and one planar surface. This image has 7975 data points with roughly 5% noise points. Figure 9 shows the segmentation obtained using the conventional GK algorithm and as can be seen the algorithm fails completely to detect the correct clusters. Figure 10 shows the segmentation given by the NGK algorithm.

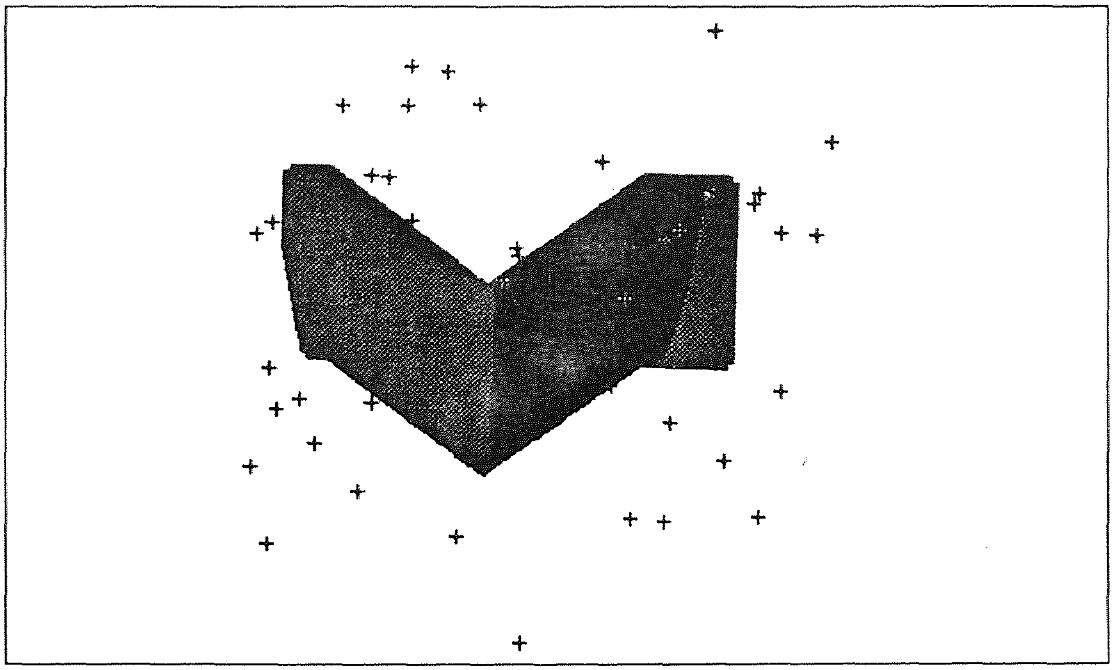

Figure 7 Segmentation of "Jig" image using AFC algorithm.

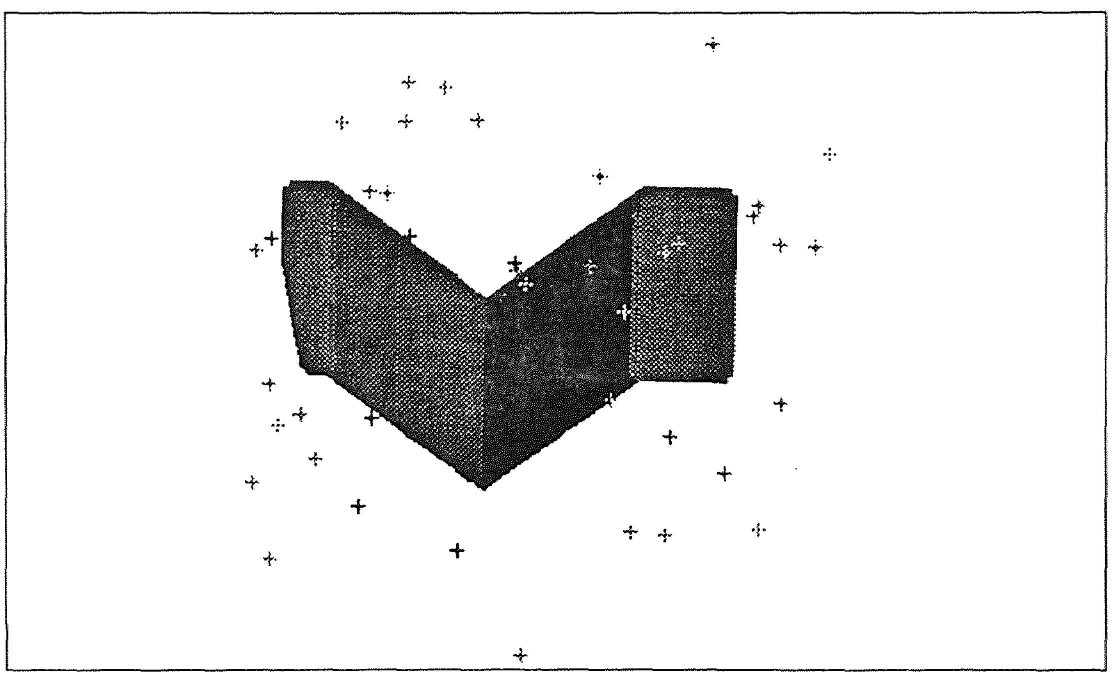

Figure 8 Segmentation of "Jig" image using NAFC algorithm.

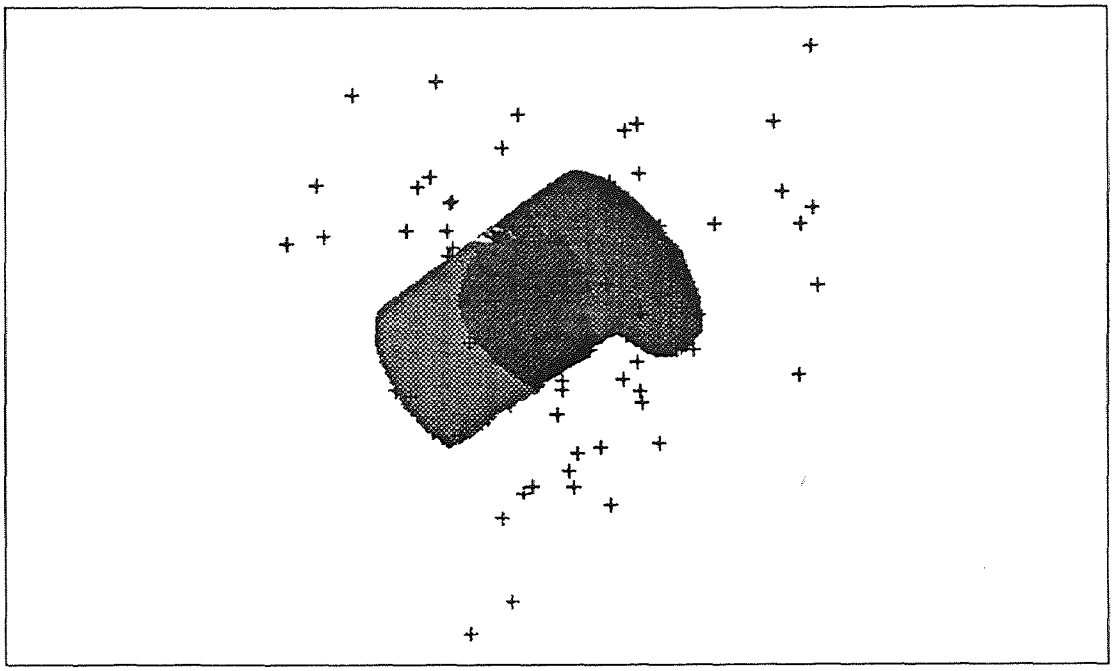

Figure 9 Segmentation of "Cylinder" image using GK algorithm.

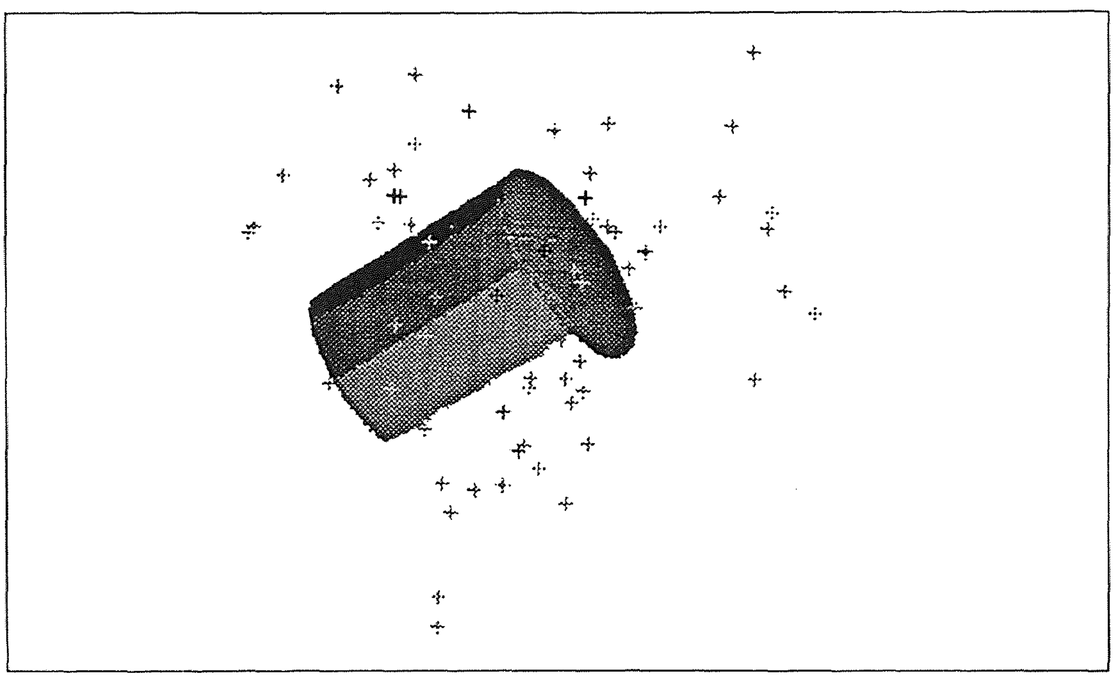

Figure 10 Segmentation of "Cylinder" image using NGK algorithm.

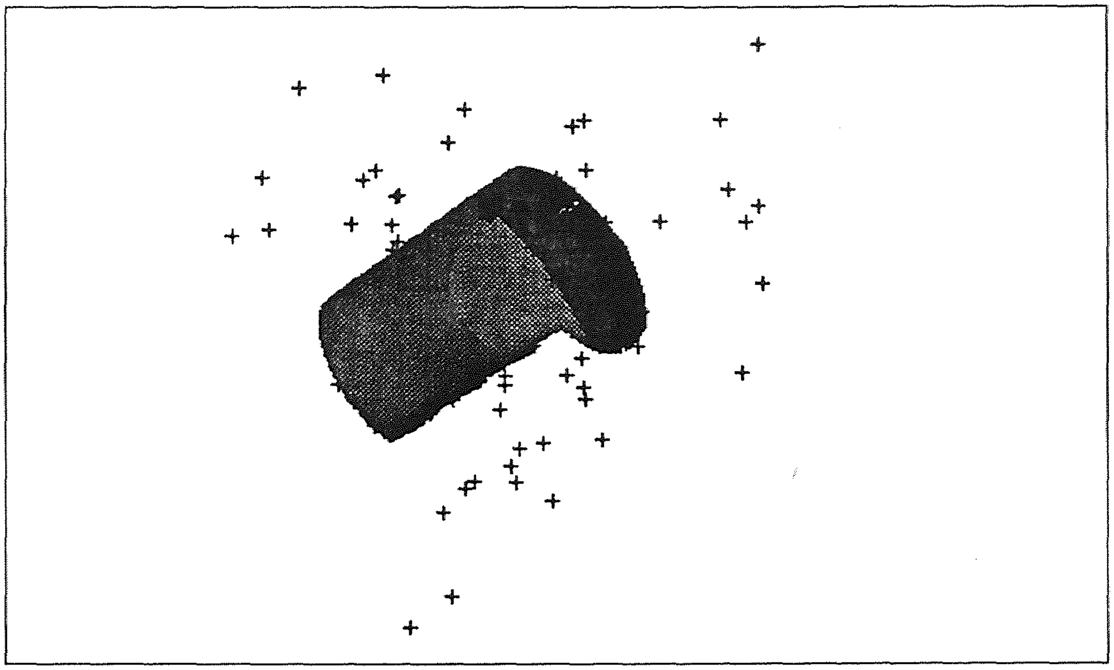

Figure 11 Segmentation of "Cylinder" image using AFC algorithm.

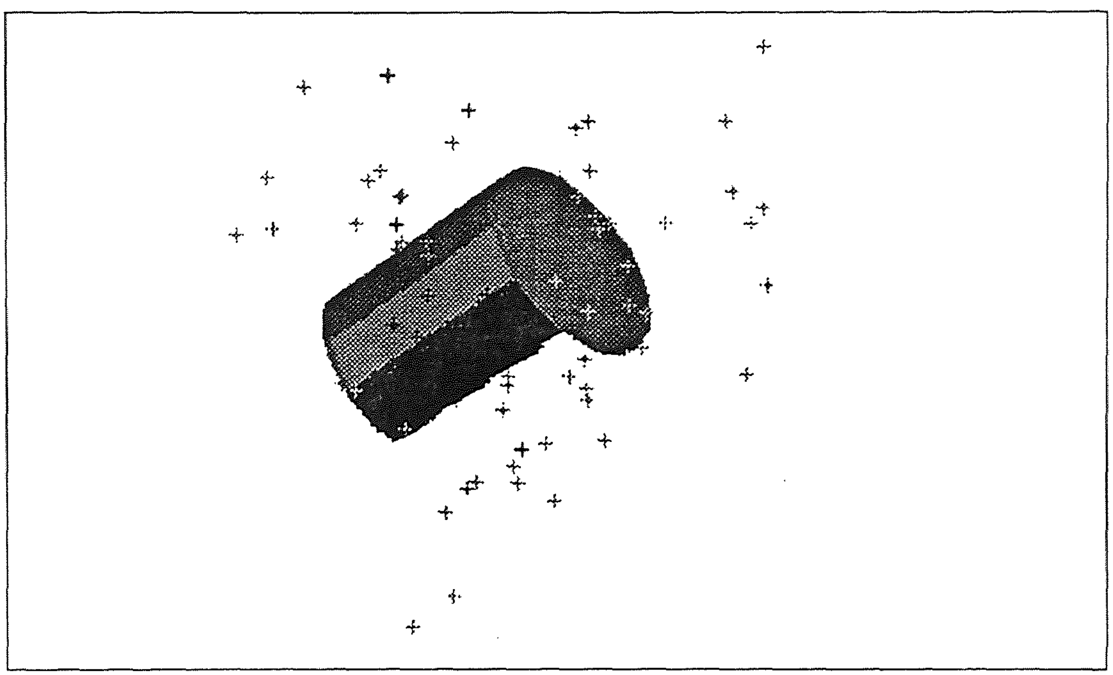

Figure 12 Segmentation of "Cylinder" image using NAFC algorithm.

The algorithm detects one cluster that is the planar surface of the cylinder, but more interestingly it detects 3 clusters on the cylindrical surface such that each defines a facet of the cylindrical surface. It can also be seen that the edge separating the two facets on the cylindrical surface (away from the observer) is not a straight edge.

Figure 11 shows the segmentation obtained using the AFC algorithm. The algorithm fails to detects any cluster clearly. Figure 12 shows the segmentation obtained using the NAFC algorithm. It can be seen that one cluster is detected as the planar surface of the cylinder and the other clusters are detected as facets on the cylindrical surface. This segmentation is better than the segmentation obtained from the NGK algorithm (figure 10) since here all the edges separating the facets on the cylindrical surface are straight.

5.4 Image with a Conical Surface

Figures 13,14,15 and 16 show the image of a cone with only a partial conical surface (only the visible side of the cone was scanned into the range image). The image has 4578 points with about 5% noise points. Figure 13 shows the segmentation obtained by using the conventional GK algorithm. The value of c was taken as 4. The algorithm is not able to detect four facets on the cone. Three clusters are detected on the conical surface but neither defines a facet clearly. The fourth cluster is not detected on the surface but is lost in the outliers.

The NGK algorithm was used to segment the same data set for c equal to 4. The algorithm detected 2 facets clearly but the other two clusters failed to represent facets clearly. Figure 14 shows the segmentation obtained by using the NGK algorithm for c equal to 3, the segmentation is fair and the three clusters detected roughly define one facet each on the conical surface.

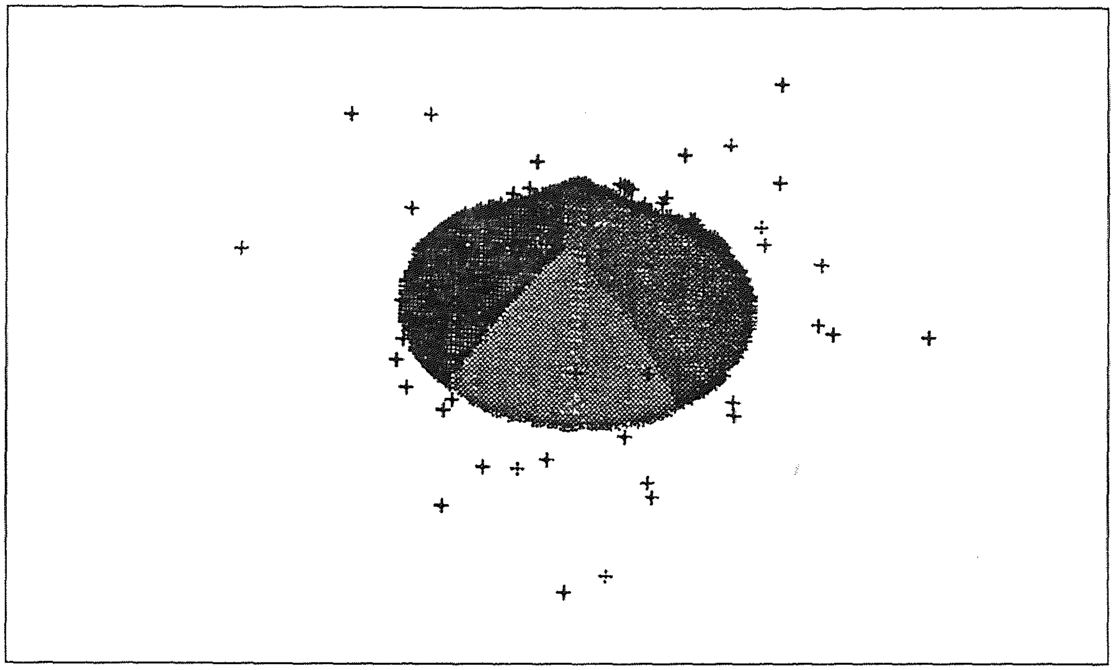

Figure 13 Segmentation of "Cone" image using GK algorithm.

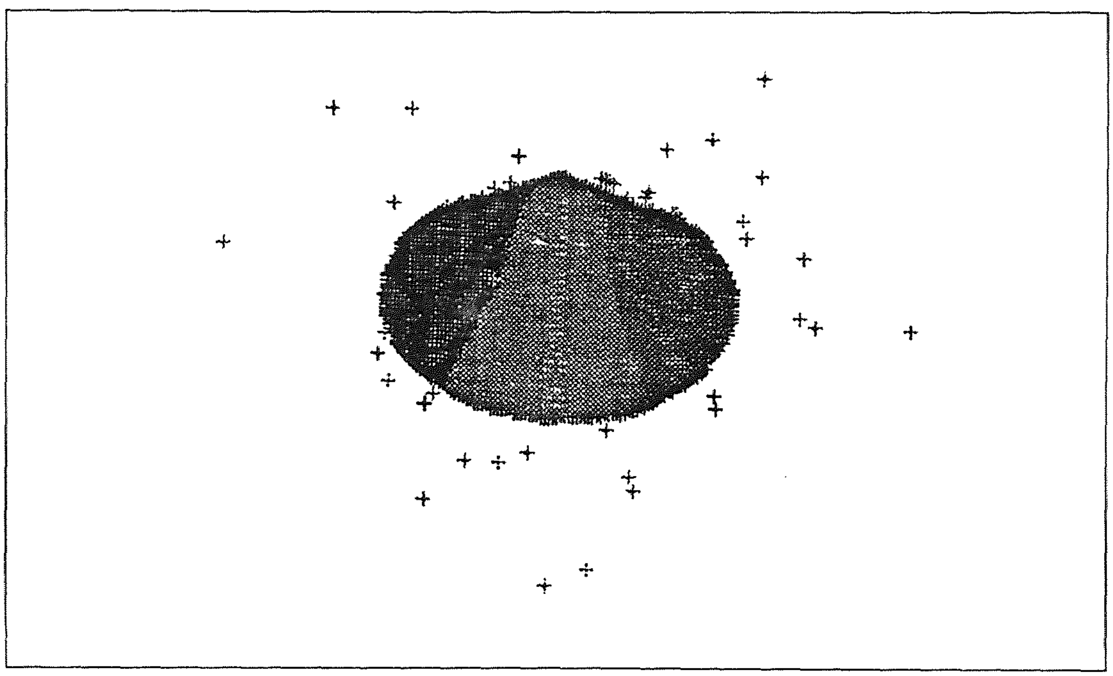

Figure 14 Segmentation of "Cone" image using NGK algorithm.

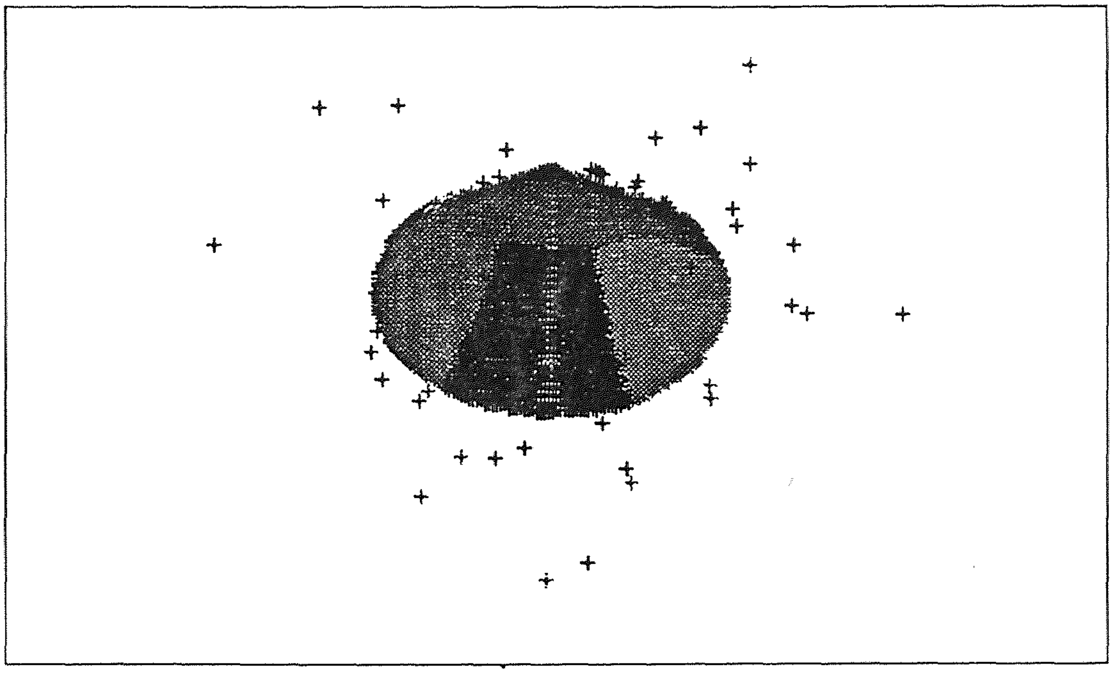

Figure 15 Segmentation of "Cone" image using AFC algorithm.

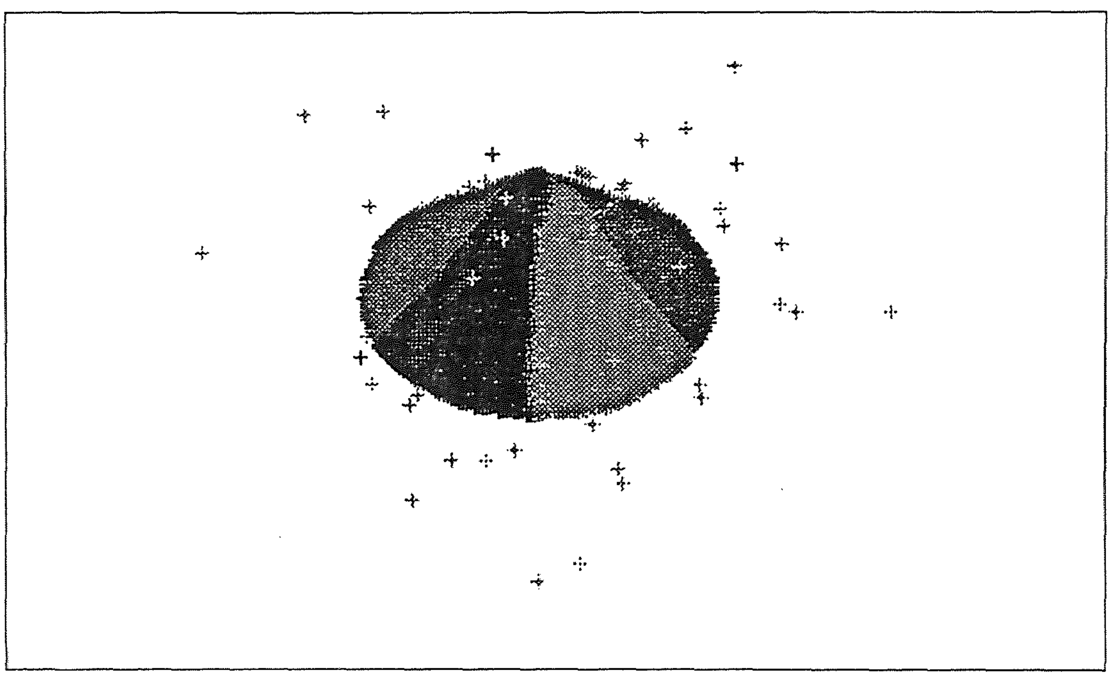

Figure 16 Segmentation of "Cone" image using NAFC algorithm.

Figure 15 shows the segmentation obtained by using the AFC algorithm for c equal to 4. The algorithm detects all four clusters on the conical surface but none of them represent a facet clearly. Figure 16 shows the segmentation obtained using the NAFC algorithm, and as can be seen four facets are clearly detected on the conical surface.

CHAPTER 6

CONCLUSIONS

6.1 Discussion

Krishnapuram and Freg[14] have used the GK algorithm to detect a large number of surface patches. The surface patches are then merged, using properties (eigenvectors and eigenvalues) of the clusters as merging criterion, to construct the surface of the object. This work has used a different approach where an attempt is made to detect a planar surface as a single cluster and detect a curved surface as number of facets. The eigenvalues and the eigenvectors of these planar clusters can be directly used for mathematical modeling of the object. This is of particular importance since many CAD packages today use faceted descriptions of objects to be modeled. In both the cases the number of clusters to be detected have to be decided apriori (step 1 of the algorithms). The approach used by this work eliminates the need for merging of clusters but makes the task of cluster number selection more difficult.

The results presented earlier showed that the GK algorithm failed to perform well in the presence of noise. The NGK algorithm performed very well in the presence of noise especially when the objects have only planar clusters. The NGK algorithm was capable of detecting planar facets on objects with planar and cylindrical surfaces. As expected the AFC was not able to perform well on noisy data on its own. The NAFC algorithm managed to perform well not only in the presence of noise but also in the presence all types of surfaces, namely planar, cylindrical and conical. Though the fourth cluster tends to get lost in the noise points for GK and AFC algorithms, it should be evident to the reader that by using c=5 is not going to solve the problem.

The actual runtime for the algorithms was not very different from each other and was highly affected by the initialization scheme used. The key factor in determining the runtime was the number of points in the data sets. The algorithms required 10 to 15 minutes for segmentation of the "staircase" image (8477 data points) and "cylinder" image (7975 data points), on the SUN SPARC station 10. The "jig" image (21148 data points) required 20 to 30 minutes for segmentation on the same machine, while the "cone" image (4578 data points) required 5 to 10 minutes. The memory requirement for the proposed algorithms was low compared to the Hough Transform based algorithms.

The NGK algorithm when used for segmenting the "Cylinder" image detected the facets correctly but the edge seperating two facets was not a straight edge. This algorithm did not provide a good segmentation of the "Cone" image compared to the NAFC algorithm. This can be attributed to the fact that the two algorithms use very different approaches for clustering. The NGK uses "points" or cluster centers as prototype for measuring distances and the shape of the cluster is controlled by the norm inducing matrix which is varied locally. Thus the NGK algorithm tends to detect hyper-ellipsoidal clusters (shape of a football rounded at the two sharp points). The NAFC on the other hand used "planes" and "points" as prototypes for measuring distance and the norm inducing matrix is an identity matrix for all clusters and hence tries to detect clusters that are "disc" shaped. The mixing coefficient plays an important role in the shape of these "discs", for lower values of α these tend to be round. As the value of α increases the "discs" are elliptical in shape and for value equal to or close 1 the algorithm detects planes (which is usually the case).

It is important to mention at this point that for a fixed value of α (close to or equal to 1) the algorithm (which is essentially FCP or FCE) fails to detect the correct planar facets, which suggests that the value of α has to be selected in a adaptive manner. This is in agreement to a similar observation made by Dave in detecting lines in 2-D digital images, where a fixed α failed to detect the correct lines in the image.

6.2 Conclusions

A robust approach for range image segmentation based on clustering is presented by proposing two algorithms (NGK and NAFC). This approach is capable of performing very well in the presence of noise and is successful in detecting planar facets on objects with planar, cylindrical and conical surfaces. It is established that the concept of "noise clustering" can be applied to any objective function based algorithms to yield very good results. The success of the proposed algorithms suggests that clustering coupled with "noise clustering" is a very useful tool for range image segmentation. It is shown that the extension of AFC is equally successful in 3-D as it was in 2-D. The NAFC algorithm appears to perform equally well or better, compared to the NGK algorithm, this though is still an open issue of research.

6.3 Scope for Future Research

The work has concentrated on presenting a robust approach for segmentation that works well for noisy data and for simple objects. The issues of detecting unknown number of clusters in a data set still needs to be addressed. The performance of the two proposed algorithms need to be compared by a more extensive study on different range images. The extension of this approach to more complicated surfaces is also an area for further investigation.

APPENDIX

NOISE GENERATION PROGRAM

```
/*PROGRAM FOR INTRODUCING NOISE POINTS IN PRIP DATA SETS*/
#include <math.h>
#include <stdio.h>
/*** VARIABLES ****/
#define P 3
#define MAX_PTS 30000
#define RAND_MAX 99800
#defineFACTOR.2

int n,c,i,k,j,a,b,p=3,num[50],count,flag[240][240];
int iter_flag=0;
int noise,outlier;
double u[50][MAX_PTS], x[MAX_PTS][P], d[50][MAX_PTS] ,v[50][P],s[P][
],fv1[P];
double rho[50],an[P][P],temp,clx[50][MAX_PTS],cly[50][10000],z[P][P];
doublemax[P],min[P],clz[50][MAX_PTS];
float frac;
FILE *filename;
char str[30],out[30],junk[60],sel[20];

double tem1;
int numb = 1;
double noise_per, out_per, noise_dev, out_dev;
doublexmin,xmax,ymin,ymax,zmax,zmin,yvalmin[P],yvalmax[P],zi[P][P][50];
```

```
main()
{
 /**** GETTING THE FILE NAME**********/
L0:
 printf("Give filename\n");

 scanf("%s",str);

 filename=fopen(str,"r");

 if (filename == NULL) {

 printf("File not found \n");

 goto L0;
 }

 read_file();
 generate();
 write_file();
}

 /*****SUBROUTINE TO READ DATA FILE*********/
read_file()
{
```

```
/*READING THE DATA FILE*/
n=0;
fscanf(filename,"%s %s\n",junk,junk);
fscanf(filename,"%s %s\n",junk,junk);
fscanf(filename,"%s %s %c %c %c %s\n",junk,junk,junk,junk,junk,junk);
for(j=0;j<240;j++)
{
for(i=0;i<240;i++)
{
if(i<239){
fscanf(filename,"%d ",&flag[j][i]);
}
else{
fscanf(filename,"%d \n",&flag[j][i]);
}


}
}
/****READING THE X COORDINATES*******/
for(j=0;j<240;j++)
{
for(i=0;i<240;i++)
{
if(i<239){
fscanf(filename,"%G ",&tem1);
```

```
}
else{
fscanf(filename,"%G \n",&tem1);
}
if(flag[j][i]==1){
x[n][0]=tem1;
n++;
}


}
}
/********READING THE Y COORDINATE ******/
a=0;
for(j=0;j<240;j++)
{
for(i=0;i<240;i++)
{
if(i<239){
fscanf(filename,"%G ",&tem1);
}
else{
fscanf(filename,"%G \n",&tem1);
}
if(flag[j][i]==1){
x[a][1]=tem1;
a++;
}
```

```
}
}
/***** READING THE Z COORDINATE *******/
a=0;
for(j=0;j<240;j++)
{
for(i=0;i<240;i++)
{
if(i<239){
fscanf(filename,"%G ",&tem1);
}
else{
fscanf(filename,"%G \n",&tem1);
}
if(flag[j][i]==1){
x[a][2]=tem1;
a++;
}

}
}

printf("No. of points read %d\n",n);
fclose(filename);

}
```

```
write_file()
{

/***** WRITING THE OUTPUT FILE *****/

printf("Give name of output file\n");
scanf("%s",junk);
filename=fopen(junk,"w");
fprintf(filename,"%d \n",(n+noise+outlier));
for(i=0;i<(n+noise+outlier);i++){
fprintf(filename,"%G %G %G \n",x[i][0],x[i][1],x[i][2]);
 }
printf("No. of total points written %d + %d + %d = %d\n",n,noise,outlier,n+
noise+outlier);
fclose(filename);
}

generate()
{
/**** GENERATING THE NOISE POINTS *****************/

printf("Give percentage of noise points in decimals\n");
scanf("%G",&noise_per);
printf("Give deviation ( 0.05 to 0.5) \n");
scanf("%G",&noise_dev);
printf("Give percentage of outlier points in decimals\n");
```

```
scanf("%G",&out_per);
printf("Give deviation for outliers ( 1.0 to 5.0 )\n");
scanf("%G",&out_dev);

 noise = n * noise_per;
 outlier = n * out_per;



 for(i=0;i<noise;i++){
 for(j=0;j<3;j++){
 gen_rand();
 numb = 500 - numb;
 frac = numb / 500.0;
 temp = (frac * noise_dev);
 gen_rand();
 k = (int) (n * (float)(numb /1000.0)) ;
 x[n+1+i][j] = x[k][j] + temp;
 }
 }

 for(i=0;i<outlier;i++){
 for(j=0;j<3;j++){
 gen_rand();
 numb = 500 - numb;
 frac = (float)numb / 500.0;
 temp = (frac * out_dev);
 gen_rand();
```

```
 k = (int)(n * (float)(numb / 1000.0));
 x[n+1+i][j] = x[k][j] + temp;
 }
 }
}

gen_rand()
{
/********** GENERATING A RANDOM NUMBER ********/
srand(numb);
numb = rand();
numb = numb%1000 + 1;
}
```

WORKS CITED

1. Bezdek, J. *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York and London, Plenum Press (1987): 65-71.

2. Besl, P., and Jain, R. "Segmentation Through Variable-Order Surface Fitting." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* VOL .10, No. 2 (1988): 176-189.

3. Faugeras, O., Herbert, M., and Pauchon, P. "Segmentation of Range Data into Planar and Quadratic Patches."*Proc. IEEE Conf. Computer Vision and Pattern Recognition* (1983): 8-13.

4. Muller Y., and Mohr, P. "Planes and Quadratics Detection Using Hough Transforms." *Proc. 7th Int. Conf. Pattern Recognition* (1984): 1101-1103.

5. Hoffmann, R., and Jain, K. "Segmentation and Classification of Range Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* VOL. 9, No. 5 (1987): 608-619.

6. Bhanu, B., and Lee,S. "Range Data Processing: Representation of Surfaces by Edges." *proc. 8th Int. Conf. Pattern Recognition* (1986): 77-85.

7. Tomita, F., and Kanade, T. "A 3D Vision System: Generating and Matching Shape Description in Range Images." *Proc. 1st Int. Conf. Artificial Intelligence Applications* (1984): 186-191.

8. Yokoya, N., and Levine, M. "Range Image Segmentation Based on Differential Geometry: A Hybrid Approach." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* VOL 11, No. 6 (1989): 643-649.

9. Jolion, J., Meer, P., and Bataouche, S. "Robust Clustering with Applications in Computer Vision." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* VOL. 13, No. 8 (1991): 791-801.

10. Duda, R., and Hart, P. *Pattern Classification and Scene Analysis.* Wiley, New York (1973).

11. Jain, A. K. and Dubes R. *Algorithms for Clustering Data.* Prentice-Hall, Englewood Cliffs, New Jersey (1988).

12. Bezdek J. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York and London (1987).

13. Gustafson, D., and Kessel, W. "Fuzzy Clustering with a Fuzzy Covariance Matrix." *Proc. IEEE-CDC.* VOL. 2 (1979): 761-766.

14. Krishnapuram, R., and Freg, C. "Fitting An Unknown Number of Lines and Planes to Image Data Through Compatible Cluster Merging." *Pattern Recognition* (1992).

15. Dave, R. "An Adaptive Fuzzy c-Elliptotype Clustering Algorithm." *Proceedings of NAFIPS 90: Quarter Century of Fuzziness,* ed. I. B. Turksen. VOL.I (1990): 9-12.

16. Dave, R. "Use of The Adaptive Fuzzy Clustering Algorithm to Detect Lines in Digital Images." *Intelligent Robots and Computer Vision VIII.* VOL. 1192(2) (1989): 600-611.

17. Jolion, J., and Rosenfeld, A. "Cluster Detection in Background Noise." *Pattern Recognition* (1989): 603-607.

18. Dave, R., "Characterization and Detection of Noise in Clustering." *Pattern Recognition Letters*, VOL. 12 (1991): 657-664.

19. Dunn, J. "A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well Separated Clusters," *Journal of Cybern.* VOL. 2 (1974): 32-57.

20. Petrou, M., and Kittler, J. "Optimal Edge Detectors for Ramp Edges." IEEE *Transactions on Pattern Analysis and Machine Intelligence.* VOL. 13, No. 5 (1991): 483-491.

21. Anderson, I., Bezdek, J. and Dave, R. "Polygonal Shape Description of Plane Boundaries." *Systems Science and Science 1.* SGSR Press, Louisville, Kentucky (1982): 295-301.

22. Gunderson, R. "An Adaptive FCV Clustering Algorithm." *Int. J. Man-Machine Studies* (1983): 97-104.

23. Wiegand, G. and Covey, R, *HOOPS Graphics System Reference Manual.* Ithaca, Alameda, California.

24. Cleve, m, *Linpack and Eispack Subroutines.* Argonne National Lab, University of New Mexico.